CoNaIISI 2014

Actas del 2° Congreso Nacional de Ingeniería Informática/Sistemas de Información

13 y 14 de Noviembre de 2014
San Luis, Argentina

Consejo Federal de Decanos de Facultades de Ingeniería

Red de Carreras de Ingeniería en Informática / Sistemas de Información del CONFEDI

Departamento de Informática Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de SanLuis, Argentina

Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina

Universidad Nacional de San Luis, Argentina

ISSN: 2346-9927

Actas de CoNaIISI 2014 - ISSN: 2346-9927
Red de Carreras de Ingeniería en Informática / Sistemas de Información (RIISIC) perteneciente al CONFEDI. Ayacucho 132 1º Pso.- CABA, Argentina.
Tel. +54 (011) 4952 4466 - e-mail: info@confedi.org.ar
Editores: Daniel Riesco, Germán Montejano, Fabiana Piccoli, Andrés Bursztyn.
Editorial: Universidad Nacional de San Luis, Facultad de Ciencias Físico-Matemáticas y Naturales, Departamento de Informática,
Ejercito de Los Andes 950 - D5700HHW – San Luis – Argentina.
Tel. +54 (0266) 4520300, int. 2112. e-mail: secinfo@unsl.edu.ar

Organizadores CoNaIISI 2014 San Luis – Argentina

Red de Carreras de Ingeniería en Informática / Sistemas de Información del CONFEDI (RIISIC)

Comité Ejecutivo (Período Noviembre 2014 a Noviembre 2015)

Coordinador:	Ing. Andrés Bursztyn	Universidad Tecnológica Nacional, Facultad Regional Buenos Aires
Coordinador Suplente:	Dr. Daniel Riesco	Universidad Nacional de San Luis
Coordinador Saliente:	Ing. Luis Perna	Universidad Tecnológica Nacional, Facultad Regional Delta
Secretarios Titulares:	Dr. Aldo Vecchietti	Universidad Tecnológica Nacional, Facultad Regional Santa Fé
	Ing. Beatriz P. De Gallo	Universidad Católica de Salta
Secretarios Suplentes:	Lic. Claudia Herrera	Universidad Nacional de Catamarca
	Ing. Osvaldo Marcovecchio	Universidad de la Marina Mercante

Comité Organizador General

Ing. Andrés Bursztyn	Universidad Tecnológica Nacional, Facultad Regional Buenos Aires Coordinador Red RIISIC
Dr. Daniel Riesco	Universidad Nacional de San Luis. Coordinador de CoNaIISI
Dr. Germán Montejano	Universidad Nacional de San Luis Coordinador de CoNaIISI
Dra. Fabiana Piccoli	Universidad Nacional de San Luis Directora del Departamento de Informática

Comité Organizador Local

Mg. Ing. Corina Natalia Abdelahad
Dra. Gisela Dorzán
Mg. Ing. Norma Beatriz Pérez
Lic. Guillermo Aguirre
Sr. Leonardo Escudero
Prof. Gabriela Palacio
Prof. María Clauda Albornoz
Mg. Augusto Campos Farnese
Lic. Cecilia Palacios
Lic. Javier Apolloni
Lic. Jackelin Fernández
Lic. Lucrecia Rescaldani
Dra. Victoria Soledad Aragón
Dr. Edgardo Ferreti
Lic. Luis Roque
Mg. Ing. Lorena Soledad Baigorria
Lic. Mónica Fuentes
Mg. Ing. Carlos Humberto Salgado
Prof. Mercedes Barrionuevo
Mg. Ing. Ana Funes
APU. Enrique Sánchez
APU. Edgardo Bernardis
Dr. Mario Groppo
Lic. Dario G. Funez
Mg. Ing. Alejandro Sánchez
Lic. Hernán Bernardis
Mg. Ing. Ana Gabriela Garis
Mg. Cs. María Teresa Taranilla
Dr. Mario Marcelo Berón
Lic. Claudia Ruth Gatica
Dr. Cristian Tissera
D.Graf. Verónica Cacace
Lic. Verónica Ludueña
Lic. Matías Truglio
Dra. Leticia Cagnina
APU. Mariano Luzza
Mg. ing. Gabriel Domingo Vilallonga
Lic. Alicia D. Castro
Lic. Enrique Alfredo Miranda
Esp. Mariela E. Zuñiga
Mg. Ing. Arístides Juan Dasso
Mg. Cs. Silvia Molina
Lic. Mónica Daza
Esp. Ing. Mario Gabriel Peralta
APU. Javier Azcurra

Comité Académico

Aplicaciones Informáticas y de Sistemas de Información

Moderadores:

Dr. Sebastián Rodriguez	Universidad Tecnológica Nacional – F.R. Tucumán – GITIA
Dr. Elías Todorovich	Universidad Nacional del Centro de la Provincia de Buenos Aires

Dr. Julio Dondo	Universidad de Castilla La Mancha – España
Dr. Horacio Leone	Universidad Tecnológica Nacional – F.R. Santa Fe
Mg. Juan Giró	Universidad Tecnológica Nacional – F.R. Córdoba Universidad Nacional de Córdoba
Dr. Eduardo Destéfanis	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Ing. Ana Funes	Universidad Nacional de San Luis
Mg. Marcelo Marciszack	Universidad Tecnológica Nacional – F.R. Córdoba
Dra. Verónica Gil Costa	Universidad Nacional de San Luis
Dra. María Laura Caliusco	Universidad Tecnológica Nacional – F.R. Santa Fe
Dr. Pedro Colla	Instituto Universitario Aeronáutico
Dr. Omar Chiotti	Universidad Tecnológica Nacional – F.R. Santa Fe
Dra. Mariela Rico	Universidad Tecnológica Nacional – F.R. Santa Fe
Dr. Sebastián Rodriguez	Universidad Tecnológica Nacional – F.R. Tucumán – GITIA
Dr. Nestor Calvo	Universidad Nacional del Litoral
Dr. Marcelo Karanik	Universidad Tecnológica Nacional – F.R. Resistencia
Mg. Guillermo Sampallo	Universidad Tecnológica Nacional – F.R. Resistencia
Ing. Luis H. Perna	Universidad Tecnológica Nacional – F.R. Delta
Mg. Carlos Alberto López	Universidad Tecnológica Nacional – F.R. La Plata
Dr. Marcelo De Vincenzi	Universidad Abierta Interamericana
Dr. Elías Todorovich	Universidad Nacional del Centro de la Provincia de Buenos Aires
Cal. Liliana Estela Rathmann	Universidad Atlántida Argentina
Ing. Marta Lasso	Universidad Nacional de la Patagonia Austral, U.A. Caleta Olivia
Mg. Gustavo Maigua	Universidad Tecnológica Nacional – F.R. Tucumán
Mg. Mario M. Figueroa de la Cruz	Universidad Tecnológica Nacional – F.R. Tucumán
Mg. Raúl Oscar Klenzi	Universidad Nacional de San Juan
Lic. Natalia Carolina Mira	Instituto Universitario Aeronáutico
Mg. Marta Cristina Fennema	Universidad Tecnológica Nacional – F.R. Tucumán Universidad Nacional de Santiago del Estero Universidad Católica de Santiago del Estero
Mg. Sebastián García	Universidad Nacional de San Juan
Mg. Miguel Solinas	Universidad Nacional de Córdoba
Lic. Alejandra Malberti	Universidad Nacional de San Juan
Dr. Marcelo Frias	Instituto Tecnológico de Buenos Aires
Dr. Nelson Acosta	Universidad Nacional del Centro de la Provincia de Buenos Aires
Ing. Anibal Carmona	UNITECH
Ms.C. Ariel Leiva	Mercado Libre

Aspectos Legales y Profesionales

Moderadores:

MBA. H. Beatriz P. de Gallo	Universidad Católica de Salta
Mg. Ing. Ana Garis	Universidad Nacional de San Luis

Dr. Daniel Riesco	Universidad Nacional de San Luis

Esp. María Alejandra Barrera	Universidad Nacional de Catamarca
Dra. Laura Spina	Universidad Tecnológica Nacional – F.R. Santa Fe
Mg. Paula Angeleri	Universidad de Belgrano
Mg. Ing. Ana Garis	Universidad Nacional de San Luis
Ab. Esp. Carlos Dionisio Aguirre	Uiversidad Nacional de Córdoba
Lic. Guillermo A. Montenegro	Consejo Profesional Ciencias Informáticas de la Provincia de Córdoba
MBA. H. Beatriz P. de Gallo	Universidad Católica de Salta
Dr. Ab. Carlos Acquistapace	Universidad Tecnológica Nacional – F.R. Buenos Aires

Bases de Datos

Moderadores:

Esp. Roberto Muñoz	Universidad Tecnológica Nacional – F.R. Córdoba
Dr. Aldo Vecchietti	Universidad Tecnológica Nacional – F.R. Santa Fe

Dr. Daniel Romero	Universidad Nacional de Asunción – Paraguay
Dr. Mario Divan	Universidad Nacional de La Pampa
Dra. Olinda Gaggliardi	Universidad Nacional de San Luis
Mg. Marcelo Castro	Universidad Nacional de Jujuy
Dr. Marcelo Falappa	Universidad Nacional del Sur
Dr. Aldo Vecchietti	Universidad Tecnológica Nacional – F.R. Santa Fe
Esp. Roberto Muñoz	Universidad Tecnológica Nacional – F.R. Córdoba
Dra. Mariana del Fresno	Universidad Nacional del Centro
Mg. Anabella Cecilia De Battista	Universidad Tecnológica Nacional – F.R. Concepción del Uruguay
Esp. Mariano García Mattio	Instituto Universitario Aeronáutico
Esp. Andrés Pascal	Universidad Tecnológica Nacional – F.R. Concepción del Uruguay
Mg. Gustavo Juarez	Universidad Tecnológica Nacional – F.R. Tucumán
Mg. Osvaldo Spositto	Universidad Nacional de La Matanza

Revisores Adicionales:

Dra. Gisela Dorzán	Universidad Nacional de San Luis
MCs. Maria Teresa Taranilla	Universidad Nacional de San Luis

Educación en Ingeniería

Moderadores:

Ing. José Luis Cabrera	Universidad de Buenos Aires
Mg. Claudia Russo	Universidad del Noroeste de Buenos Aires

Mg. Uriel Cukierman	Universidad de Palermo
	Universidad Tecnológica Nacional – F.R. Buenos Aires
Esp. Alejandra Jewsbury	Universidad Tecnológica Nacional – F.R. Córdoba
Esp. Carlos Bartó	Universidad Nacional de Córdoba
Mg. Calixto Maldonado	Universidad Siglo XXI
	Universidad Tecnológica Nacional – F.R. Córdoba
Ing. Lic. José Luis Cabrera	Universidad de Buenos Aires
Mg. Natalia Andriano	Universidad Tecnológica Nacional – F.R. Córdoba

Dr. Jorge Marcelo Montagna	CONICET
	Universidad Tecnológica Nacional, F.R. Santa Fé
Mg. Marcela E. Daniele	Universidad Nacional de Río Cuarto
Mg. Claudia Russo	Universidad del Noroeste de Buenos Aires
Dr. Gustavo Chiodi	Universidad Católica Córdoba
	Universidad Tecnológica Nacional – F.R. Córdoba
Esp. Santiago Igarza	Universidad Nacional de La Matanza
Dr. Oscar R. Bruno	Universidad Tecnológica Nacional – F. R. Buenos Aires
Ing. Anibal Romandetta	Unuversidad Nacional tres de Febrero – Red RUNIC
Esp. Marta Castellaro	Universidad Tecnológica Nacional – F. R. Santa Fe
Esp. Ing José Oscar Mugetti Mare	Universidad Tecnológica Nacional – F.R. San Francisco
Dra. Ines Casanovas	Universidad Tecnológica Nacional – F.R. Buenos Aires
	Universidad Tecnológica Nacional – F.R. General Pacheco
Mg. José Sergio Fernández	Universidad Nacional del Chaco Austral
Ing. María Eugenia De San Pedro	Universidad Nacional de la Patagonia Austral, U.A. Caleta Olivia
Mg. Virginia Mauco	Universidad Nacional del Centro de la Provincia de Buenos Aires
Mg. Carmen Leonardi	Universidad Nacional del Centro de la Provincia de Buenos Aires
Ing. Jorge Steifensand	Universidad del Norte Santo Tomás de Aquino
	Universidad Nacional de Tucumán
Ing. Horacio Loyarte	Universidad Nacional del Litoral
Lic. Alberto Cortez	Universidad Tecnológica Nacional – F.R. Mendoza
	Universidad Nacional de Mendoza
Lic. Sergio Daniel Conde	Universidad de la Marina Mercante
Ing. Andres Bursztyn	Universidad Tecnológica Nacional – F.R. Buenos Aires
Mg. Alicia Guillermina Salamon	Instituto Universitario Aeronáutico
Esp. Sandra Valla	Universidad Tecnológica Nacional – F.R. Tucumán
Lic. Natalia Carolina Mira	Instituto Universitario Aeronáutico
Ing. María Alejandra Boggio	Instituto Universitario Aeronáutico
Mg. Ing. Claudia Inés Inchaurrondo	Universidad Nacional de Catamarca
Mg. Ing. Carlos Salgado	Universidad Nacional de San Luis
Mg. Ing. Ana Gabriela Garis	Universidad Nacional de San Luis
Dr. Daniel Riesco	Universidad Nacional de San Luis
Dr. Germán Montejano	Universidad Nacional de San Luis
Mg. Ing. Alejandro Sanchez	Universidad Nacional de San Luis
Esp. María Alejandra Barrera	Universidad Nacional de Catamarca
Dr. Luis Canali	Universidad Tecnológica Nacional – F.R Córdoba
Dra. Zulma Cataldi	Universidad Tecnológica Nacional – F.R. Buenos Aires
Esp. Waldo Geremía	Universidad Blas Pascal
Mg. Liliana Cuenca Pletch	Universidad Tecnológica Nacional – F.R. Resistencia
Esp. Rosana Hadad Salomon	Universidad Tecnológica Nacional – F.R. Tucumán

Gestión de Proyectos

Moderadores:

Mg. Carlos Alberto López	Universidad Tecnológica Nacional – F.R. La Plata
Dra. Gabriela Robiolo	Universidad Austral

Esp. Iris Gastañaga	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Alejandro Vázquez	Universidad Tecnológica Nacional – F.R. Mendoza
Mg. Marcela E. Daniele	Universidad Nacional de Río Cuarto

Dr. Pablo Villarreal	Universidad Tecnológica Nacional – F.R. Santa Fe – CONICETT
Esp. Adriana Echeverría	Universidad de Buenos Aires
Dra. Mariel Alejandra Ale	Universidad Tecnológica Nacional – F.R. Santa Fe – CIDISI
Mg. Carlos Alberto López	Universidad Tecnológica Nacional – F.R. La Plata
Ing. Jorge Buabud	Universidad Tecnológica Nacional – F.R. Tucumán
Ing. Osvaldo Marcovecchio	Universidad de la Marina Mercante
Dr. Ing. José Domingo Cuozzo	Instituto Universitario Aeronáutico
Alejandro Aramburu	NEC

Ingeniería de Sistemas y de Software

Moderadores:

Dr. Luis Olsina	Universidad Nacional de La Pampa
Dr. Silvio Gonnet	Universidad Tecnológica Nacional – F.R. Santa Fe – INGAR

Dra. Elsa Estevez	United Nations University
Dra. Hernán Astudillo	Universidad Técnica Federico Santa María – Chile
Dr. Carlos Kavka	ESTECO – Italia
Dr. Luis Olsina	Universidad Nacional de La Pampa
Dr. Daniel Riesco	Universidad Nacional de San Luis
Dr. María de los Ángeles Martín	Universidad Nacional de La Pampa
Dr. Nazareno Aguirre	Universidad Nacional de Río Cuarto – CONICET
Dr. Hernán Molina	Universidad Nacional de La Pampa
Dr. Germán Montejano	Universidad Nacional de San Luis
Mg. Paula Angeleri	Universidad de Belgrano
Dr. Roberto Uzal	Universidad Nacional de San Luis
Esp. Alvaro Ruiz de Mendarozqueta	Universidad Tecnológica Nacional – F.R. Córdoba Fundación Sadosky
Dr. Mario Berón	Universidad Nacional de San Luis
Dra. Gabriela Robiolo	Universidad Austral
Ing. Liliana Favre	Universidad Nacional del Centro de la Provincia de Buenos Aires
Mg. Ing. Lorena Soledad Baigorria	Universidad Nacional de San Luis
Mg. Marcelo Castro	Universidad Nacional de Jujuy
Mg. Ing. Carlos Salgado	Universidad Nacional de San Luis
Esp. Lic. Jorge Finochietto	Universidad CAECE – Sede Mar del Plata
Dra. Luciana Roldán	Universidad Tecnológica Nacional – F.R. Santa Fe – CONICET
Dr. Andres Díaz Pace	Universidad Nacional del Centro – CONICET
Esp. Ing. Mario Peralta	Universidad Nacional de San Luis
Ing. Alejandro Bianchi	Liveware, Argentina
Dr. Daniel Giulianelli	Universidad Nacional de La Matanza
Mg. Ing. Alejandro Sánchez	Universidad Nacional de San Luis
Mg. Marcela E. Daniele	Universidad Nacional de Río Cuarto
Mg. Liliana Cuenca Pletch	Universidad Tecnológica Nacional – F.R. Resistencia
Dr. Pablo Villarreal	Universidad Tecnológica Nacional – F.R. Santa Fe – CIDISI – CONICET
Dr. Silvio Gonnet	Universidad Tecnológica Nacional – F.R. Santa Fe – INGAR

Dra. Marcela Vegetti	Universidad Tecnológica Nacional – F.R. Santa Fe – INGAR – CONICET
Mg. Santiago Ceria	Universidad de Buenos Aires – Fundación Sadosky
Dr. Ricardo F. Orozco	Universidad Argentina de la Empresa
Esp. Santiago Igarza	Universidad Nacional de La Matanza
Dr. Pablo Fillottrani	Universidad Nacional del Sur
Mg. Gustavo Lopez	Universidad de Buenos Aires
Dr. Cesar Acuña Borobach	Universidad Tecnológica Nacional – F.R. Resistencia
Ing. Edgardo Belloni	Instituto Universitario Gaston Dachary – Misiones
Mg. Carmen Leonardi	Universidad Nacional del Centro de la Provincia de Bs. As. (UNICEN)
Dr. Luciana Ballejos	Universidad Tecnológica Nacional – F.R. Santa Fe – CIDISI
Mg. Marcelo Estayno	Universidad Nacional Lomas de Zamora
Dra. Claudia Pons	Universidad Nacional de La Plata Universidad Abierta Interamericana
Dr. Leandro Antonelli	Universidad Nacional de La Plata – LIFIA
Lic. Alejandro Oliveros	Universidad Nacional de Tres de Febrero
Dr. Jorge Roa	Universidad Tecnológica Nacional – F.R. Santa Fe – CONICET
Dra. Alejandra Cechich	Universidad Nacional del Comahue
Lic. Sergio Zapata	Universidad Nacional de San juan
Dr. Claudio Ochoa	INTEL
Mg. Carlos Arroyo Arzubi	IUE/EST

Revisores Adicionales:

Dra. Agustina Buccella	Universidad Nacional del Comahue

Redes – Sistemas Operativos – Fundamentos Informáticos

Moderadores:

Dr.David La Red Martinez	Universidad Tecnológica Nacional – F.R. Resistencia
Esp. Ing. Martín Morales	Universidad Nacional Arturo Jauretche

Ing. Juan Carlos Vazquez	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Marcelo Arroyo	Universidad Nacional de Río Cuarto
Esp. Ing. Martín Morales	Universidad Nacional Arturo Jauretche
Ing. Juan Calloni	Universidad Tecnológica Nacional – F.R. San Francisco
Ing. Hernán Vitri	Universidad Tecnológica Nacional – F.R. Rosario
Esp. Cynthia Corso	Universidad Tecnológica Nacional – F.R. Córdoba
Esp. María Alejandra Paz Menvielle	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Ulises Mario Alberto Rapallini	Universidad Tecnológica Nacional – F.R. Concepción del Uruguay
Dra. María Lorena Bergamini	Universidad Abierta Interamericana
Dra. Paola Caymes Scutari	Universidad Tecnológica Nacional – F.R. Mendoza
Mg. Jorge Eterovic	Universidad de Morón
Mg. Cecilia Sánchez	Universidad Tecnológica Nacional – F.R. Córdoba
Esp. Patricia Nazar	Universidad Tecnológica Nacional – F.R. Tucumán
Dr. Fabiana Piccoli	Universidad Nacional de San Luis
Dr. Germán Bianchini	Universidad Tecnológica Nacional – F.R. Mendoza
Dr. David La Red Martinez	Universidad Tecnológica Nacional – F.R. Resistencia Universidad Nacional del Noreste
MSc. Antonio Castro Lechtaler	Universidad Tecnológica Nacional

Mg. Abel Crespo	Universidad Nacional de La Pampa
Ing. Alejandro Arroyo Arzubi	IUE/EST

Seguridad Informática

Moderadores:

Mg. Alejandro Vázquez	Universidad Tecnológica Nacional – F.R. Mendoza
Dr. Roberto Uzal	Universidad Nacional de San Luis

Dr. Mario Groppo	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Pablo Pessolani	Universidad Tecnológica Nacional – F.R. Santa Fe
Mg. Francisco Bavera	Universidad Nacional de Río Cuarto
Ing. Norberto Gaspar Cena	Universidad Tecnológica Nacional – F.R. Villa María
Mg. Ing. Arístides Dasso	Universidad Nacional de San Luis
Mg. Jorge Eterovic	Universidad de Morón
Lic. Julio César Liporace	IUE/EST

Trabajos Estudiantiles

Moderadores:

Mg. Marcelo Castro	Universidad Nacional de Jujuy
Mg. Lorena Soledad Baigorria	Universidad Nacional de San Luis

Mg. Ing. Lorena Soledad Baigorria	Universidad Nacional de San Luis
Mg. Ing. Arístides Dasso	Universidad Nacional de San Luis
Lic. Guillermo Aguirre	Universidad Nacional de San Luis
Esp. Alejandra Jewsbury	Universidad Tecnológica Nacional – F.R. Córdoba
Dr. Mario Berón	Universidad Nacional de San Luis
Dr. Germán Montejano	Universidad Nacional de San Luis
Dr. Daniel Riesco	Universidad Nacional de San Luis
Dr. Rubén Apolloni	Universidad Nacional de San Luis
Esp. Iris Gastañaga	Universidad Tecnológica Nacional – F.R. Córdoba
Mg. Claudia Necco	Universidad Nacional de San Luis
Esp. Agustín Brau	Universidad Tecnológica Nacional – F.R. Concepción del Uruguay
Mg. Ing. Carlos Salgado	Universidad Nacional de San Luis
Esp. Ing. Mario Peralta	Universidad Nacional de San Luis
Lic. Jorge Arroyuelo	Universidad Nacional de San Luis
Ing. María Concepción Aparicio	Universidad Nacional de Jujuy
Lic. Berta García	Universidad Nacional de San Luis
Esp. Ing. Martín Morales	Universidad Nacional Arturo Jauretche
Lic. Javier Apolloni	Universidad Nacional de San Luis
Ing. Nilda María Pérez Otero	Universidad Nacional de Jujuy
Lic. Jackelin Fernández	Universidad Nacional de San Luis
Mg. Marcelo Castro	Universidad Nacional de Jujuy
Dr. Natalia Miranda	Universidad Nacional de San Luis
Mg. Ing. Ana Garis	Universidad Nacional de San Luis
Mg. Alejandro Vázquez	Universidad Tecnológica Nacional – F.R. Mendoza
Mg. Ing. Ana Funes	Universidad Nacional de San Luis
Lic. Verónica Ludueña	Universidad Nacional de San Luis
Lic. Hernan Bernardis	Universidad Nacional de San Luis

Ing. Víctor David Sánchez Rivero	Universidad Nacional de Jujuy
Mg. Ing. Norma Pérez	Universidad Nacional de San Luis
Ing. Javier Saldarini	Universidad Tecnológica Nacional – F.R. San Francisco
Mg. Ing. Corina Abdelahad	Universidad Nacional de San Luis
Mg. Silvia Molina	Universidad Nacional de San Luis
Dr. Edgardo Ferreti	Universidad Nacional de San Luis
Prof. Claudia Albornoz	Universidad Nacional de San Luis
Mg. María Beatriz Rosas	Universidad Nacional de San Luis
Lic. Rafael Montesinos	Universidad Tecnológica Nacional – F.R. Tucumán
Lic. Alicia Catro	Universidad Nacional de San Luis
Lic. Enrique Miranda	Universidad Nacional de San Luis
Dr. Leticia Cagnina	Universidad Nacional de San Luis
Lic. Luis Roque	Universidad Nacional de San Luis
Esp. Mariela Zuñiga	Universidad Nacional de San Luis
Mg. Alberto Sánchez	Universidad Nacional de San Luis
Lic. Mario Silvestri	Universidad Nacional de San Luis
Mg. Ing. Alejandro Sánchez	Universidad Nacional de San Luis
Ing. Esteban Pelaez	Universidad Nacional de San Luis

Tabla de Contenidos

Disertantes Invitados (Resúmenes) **1**

Software Development Research – Current and Future 1
Narayan C. Debnath, Ph. D, D. Sc.

Ciber Paz: Protegiendo la Infraestructura Crítica Nacional ante Ciber Ataques y minimizando el Ciber Espionaje entre Estados Naciones 3
Prof. Dr. Roberto Uzal

Métodos Ágiles y Modelos de Madurez para el Desarrollo de Empresas de Software 5
Jorge Boria - Viviana Rubinstein

Artículos de investigación **6**

Aplicaciones Informáticas y de Sistemas de Información 6

Artículos completos **6**

Reconocimiento de Patrones en Registro de Pozos 6
Manuel Romero Salcedo, Elizabeth Santiago del Ángel, Juan Carlos Ontiveros Neri

Módulo adaptador Universal para Simulación Inmersiva 11
Alejandro Arroyo Arzubi, Carlos Federico Blanc, Francisco Brandolini, Damián Alejandro Camarasa, Antonio Castro Lechtaler, César Daniel Cicerchia, Enrique Jorge Matias Luzuriaga, Alexis Gabriel Polak

Una Arquitectura de un Sistema de Búsqueda de Respuestas 20
Alejandra Carolina Cardoso, Agustina Bini, M. Alicia Pérez Abelleira

Agentes Virtuales en Idioma Español 28
Bender, Adrián; Mazza, Néstor; García Calabria, Federico; Nicolet, Santiago

Tres Dimensiones de los Sistemas de Información 37
García, Sebastián José

Desarrollo de un Agente Inteligente Basado en el Estándar ANSI/ISA-95 47
Melina C. Vidoni, Aldo R. Vecchietti

Qript, Herramienta para automatizar la carga de documentos usando códigos QR que contienen datos en formato JSON 60
Armando, Silvana; Bianciotti, Andrés; Bonino, María Belén; Calloni, Juan Carlos; Gioino, Mauro; Knüssel, Federico; Mulassano, Micaela; Ponce, Martín; Rossi, Martín

Estrategia para la implementación de aplicaciones móviles basadas en servicios de geolocalización y crowdsourcing 67
María Roxana Martínez, Rocío Rodríguez, Pablo Vera

Modelo de Vigilancia Tecnológica e Inteligencia Competitiva (VTeIC) Basado en Minería de Datos 79
Marcelo Karanik, Leonardo Wanderer, Fabián Levin

Una Estrategia de Aprendizaje Autónomo para Sistemas Expertos, con Aplicación Concreta al Proyecto – Generador Automático de Modelos de Datos Normalizados en Bases de Datos Relacionales 89
Luis Esteban Damiano, Juan Carlos Cuevas, María Alejandra Paz Menvielle, María Soledad Romero, Roberto Miguel Muñoz, Carlos Bartó

Desarrollo de una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica 98
Garbarini, Ramiro; Gaspes Ezequiel; Macri Matías; Osuna Gustavo; Larran Germán

Rediseño E Implementación De Herramientas Informáticas Para La Auditoria De Sistemas En Red 106
Ciolli María Elena, Porchietto Claudio, Rossi Roberto

Optimización de Constantes Numéricas en Regresión Simbólica utilizando un Framework de Tree-Based Genetic Programming 114
Elli, Soledad, Jimenez, Victor A., Will, Adrian, Rodríguez, Sebastián

Diseño eficiente de redes de radio frecuencia con algoritmos CHC en comunicaciones inalámbricas 125
D. Molina, D. Pandolfi, A. Villagra, G. Leguizamón

Generador de Modelos de Programación Lineal para la optimización del cursado de la carrera de Ingeniería de Sistemas de Información dirigida a estudiantes con disposiciones horarias condicionadas por trabajo 134
Rojas, Cristina; Lescano, Gonzalo Emmanuel; Khouri, Ramiro Federico; Ibarra, Gerardo Daniel

Robot Educativo con Visión Desarrollado como Sistema Embebido de Arquitectura Abierta 141
Rafael Ignacio Zurita, José Riquelme, Rodolfo del Castillo

Análisis y procesamiento de imágenes para la detección del contorno labial en pacientes de odontología 147
César Osimani

Experiencia sobre la Incorporación de Herramientas de Software Libre a la Extracción de Conocimiento en Datos Astronómicos 153
Raúl Oscar Klenzi, Alejandra Malberti, Georgina Codwell

Sistemas de Información Geográfica en la gestión de Instituciones Universitarias 160
M. Soledad Retamar, Juan Pablo Nuñez, Anabella De Battista, Andrés Pascal, Francisco Savoy, Norma Edith Herrera

Detección de Patrones Circulares en Imágenes 168
Facundo Solano, José María Maffei, Francisco G. Gutiérrez, Agustín Martino, Juan Eduardo Picco

Análisis Comparativo en Múltiples Plataformas de los Principales Métodos Holístico Usados para la Identificación de Rostros 174
Pablo Santiago Pugliani, Elizabeth Vera

Análisis de Decisiones Multi Criterio Utilizando Agregación de Expertos basada en Calificaciones 185
Jorge Roa, Rubén Bernal, Marcelo Karanik

Modelo Semántico Basado en una Meta-Ontología de Información Geográfica 195
Tolaba, Ana Carolina; Caliusco, Ma. Laura; Galli, Ma. Rosa

Desarrollo de una Infraestructura de Datos Espaciales Académica 204
Martín, Horacio; Linares, Santiago; Suarez, Marcos; Magliocchetti, Agustín; Sajewicz, Brian; Gil, Guillermo; Ortmann, Mauro

Evaluación de la Usabilidad en sistemas E-Cultura 211
Gallo, Fabio Rafael; Palavecino, Rosa Adela; Herrera, Susana Isabel

Utilización del sistema SIG para el diseño de políticas de Vinculación Tecnológica 222
de San Pedro María Eugenia, Lasso Marta, Serón Natalia, Ramos Luis, Montenegro Cristian, Carrizo Alejandra

Configuración Eficiente de Controladores Fuzzy 228
Martínez Sergio L.; Tarifa Enrique E.; Gutiérrez Jorge J.

Un lenguaje para el modelado y simulación de procesos de morfogénesis de sistemas biológicos 236
José Oscar Angelini, Ernesto Martínez, Carlos H. G. Ramírez, María de los Milagros Gutiérrez

Técnicas metaheurísticas para resolver una variante de aplicación industrial del problema de flowshop 247
Gabriela Minetti, Carolina Salto

Visualización de imágenes médicas de alta resolución mediante una aplicación zero footprint 255
Emanuel J. C. Arguiñarena, Mariana del Fresno, Jose M. Massa, Pedro P. Escobar, Martin A. Santiago

Artículos cortos **261**

Automatización del modelo electoral de participación ciudadana 261
Istvan, Romina Mariel; Antonini Sergio Andrés

Una Aproximación al Estado del Desarrollo de Tecnologías Móviles en la Región de La Plata 267
Mirta del Carmen Peñalva

La Arquitectura de Aplicaciones en un Modelo de Servicios Reutilizables para Gobierno Electrónico 277
Marcelo Antonio Castro, Víctor David Sánchez Rivero, Luis Alejandro Vargas, José Humberto Farfán, María Concepción Aparicio & Daniel Castro

TDD - ¿Qué se le cuestiona? - Algunas respuestas a los principales cuestionamientos a la metodología 283
Ing. Pablo Andrés Vaca, Ing. Calixto Maldonado, Ing. Claudia Inchaurrondo, Ing. Juan Peretti, Ing. María Soledad Romero, Ing. Matías Bueno, Lic. Marcelo Cagliolo

Aspectos Legales y Profesionales **289**

Artículos completos **289**

Una Guía Para El Investigador Sobre Derechos De Autor 289
Litvak, Claudia S.; De Giusti, Marisa

El problema de la cifra negra en los Delitos Informáticos en Latinoamérica y el Proyecto ODILA 301
Temperini, Marcelo Gabriel Ignacio; Borghello, Cristian Fabian; Macedo, Maximiliano

Caracterización de los perfiles biográficos digitales en Facebook de adolescentes de Rafaela y Sunchales 312
María Eugenia Balbiano, Noelia Castillo, Román Pablo Zenobi, Dario Karchesky, Fernando Bircher, Micaela Bressan

Los Derechos Personalísimos en las Redes Sociales Rastros Digitales y Captura de Evidencia 324
María del Carmen Becerra, Pedro Daniel Zárate

RSE, SISTEMAS DE GESTION COMPUTARIZADOS Y CALIDAD DE VIDA 333
López, Carlos Alberto

Internet de las cosas y RFID: ¿políticas públicas o auto-regulación de privados? Un análisis a la luz de la participación de los gobiernos en Net mundial 343
Analía Aspis

Relevamiento de los factores que afectan al pleno desarrollo de la industria del software y servicios informáticos en el interior de la Provincia de Córdoba 352
Luciano Báez Moyano, Fernanda Barbero, Ana Carolina Ferreyra, Ricardo Medel, Mariano Emilio Rodríguez, Silvana Sánchez Fernández

Bases de Datos **358**

Artículos completos **358**

Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación para la Formalización de Necesidades en Proyectos de Almacenes de Datos 358
Verónica Nathalí Avalos, Mario José Diván

Conectando Gambas 3 con Firebird/Interbase SQL Server 368
Guillermo Ruben Cherencio

Construcción y carga de un DataMart y su aplicación para la caracterización de incidentes informáticos. 377
Corso, Cynthia Lorena; García, Mario Alejandro; Ciceri, Leonardo Ramón; Romero, Fernando

Modelos para la Estimación de una Cota Superior del Número Cromático de Grafos Aleatorios Usando Regresión Lineal 386
Gustavo Dejean

An Extension to Hierarchical Distance-based Conceptual Clustering 392
A. Funes, M. J. Ramírez-Quintana, R. Uzal

Artículos cortos **402**

Aplicación de técnicas y estrategias de Inteligencia de Negocio (BI) Sobre datos académicos integrados de la Facultad de Informática de la UNLP. Análisis la deserción 402
Díaz, Javier; Osorio, Alejandra; Amadeo, Paola; Kruzylko Ostojic, Claudia

Educación en Ingeniería 408

Artículos completos **408**

Diseño de un taller de introducción a la programación de robots 408
M. Lorena Talamé, Norberto Aramayo, Ana E. Gardel, M. Alicia Pérez Abelleira

Estudio de la Modalidad Atencional en Educación a Distancia en una Plataforma Learning Management System 416
Patricia Etcheverry, Calixto Maldonado

Avances en el uso del Aula Virtual con nuevas propuestas de mejora 423
Verónica L. Vanoli, M. Lucrecia Lavirgen

Software Educativo y Estrategia Didáctica para la Enseñanza de Conceptos y Métodos en Simulación de Eventos Discretos 429
Darío Weitz

La educación con TICs en un trabajo colaborativo sobre calorimetría entre docentes y alumnos de los niveles secundario y universitario 438
Mirta Susana Velazque, Horacio José Martínez

Análisis de los modelos de datos de Moodle y LAMS 448
M. Brachetta., J. Monetti, A. Cortez, C. Navas, O. León

Las nuevas realidades educativas a través de las TICS 454
Antonieta Kuz, Mariana Falco, Roxana Giandini

E-curriculum, Flexibilización Curricular: Un Caso en Matemática 463
Nadal, Jorgelina.C., Martinelli, Mercedes B., Poco, Adriana N., Haudemand, Norma Y., Constantino, Gustavo D., Brunner, Gisela P., Monetta, Agustín I., Villanueva Rottini, Juan P.

LAS COMPETENCIAS EN INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD DE LA MARINA MERCANTE. ANÁLISIS Y APLICACIÓN DE COMPETENCIAS. 471
Osvaldo Donato Marcovecchio,Sergio Daniel Conde

La importancia de las TIC's en las instancias de Evaluación: Análisis de una experiencia educativa 483
María Delia Grossi

La educación problemática o la problemática de la educación 488
Andrés Bursztyn, Santiago Ferreiros Cabrera, Susana Alcira Granado Peralta, Marcela Guerrero, Alejandra Laura Isola, Marcela Eugenia Portillo, Alejandro Prince

Bases para el Diseño Curricular de Carreras de Ingeniería en Informática / Sistemas de Información 496
Gustavo López, Arturo Servetto, Adriana Echeverría, José Luis Cabrera, Patricia Calvo, Ismael Jeder

Metodologías activas y corrección por rúbricas en el proceso de enseñanza-aprendizaje de programación 500
Verónica Aubin, Leonardo Blautzik, Renata Guatelli, Federico Pafundi, Matías Salica, Zulema Nisi, Mabel Zanga,

Aplicación de Experimentos Controlados en la Enseñanza de la Ingeniería de Software 508
Sergio Gustavo Zapata, Gustavo Antonio Sevilla, Estela Liliana Torres

Una Propuesta Pedagógica de Adecuación a Tendencias Emergentes del Uso de las TICs en la Universidad 518
Mirta Peñalva, Stella Maris Calderón, Ernesto Girbal, Carlos Gardella

Avances en la aplicación de Mundos Virtuales en Educación 526
Calixto Alejandro Maldonado, Patricia Etcheverry, Mario Groppo, Manuel Pérez Cota

Accesibilidad en el Ambiente Educativo Virtual Moodle 2.7 536
Diana Lorena Salinas Gomez, Liliana Raquel Cuenca Pletsch, Franco Escobar, María Antonieta Lopez Lotero, Valeria Celeste Sandobal Verón

Metodología para determinar la exactitud de una respuesta, escrita en forma textual, a un interrogatorio sobre un tema específico 542
María Alejandra Paz Menvielle, Karina Ligorria, Analía Guzmán, Martín Casatti, Mario A. Groppo, Seiyu Ricardo Higa Tamashiro, Juan Pablo Gimenez

Análisis de stakeholders en el desarrollo de proyectos para e-learning en contextos universitarios 549
Lucila Romero, Marcela Andrea Vera

Estrategias de evaluación y motivación del aprendizaje para la materia Simulación 557
Rosana M. Portillo, Mariana Cóccola, Ana Rosa Tymoschuk.

Entornos virtuales de aprendizaje colaborativos aplicados a Sistemas y Organizaciones bajo el marco de trabajo de la Metodología Scrum 565
Susana Rey, Silvia Lanza Castelli

Yo Virtual 573
Rosana Hadad Salomón, Jorge Buabud, Fernando Rubén Araujo, María Concepción Caporale, Elizabeth M. A. Dufour, Mario A. Paredi

Las Competencias Estudiantiles de Programación y las Tecnologías, como nuevos medios para la Formación e Integración 578
Marta Castellaro, Malva Alberto, Daniel Ambort

La educación popular en la universidad obrera 586
Alfredo Sanzo, Lucas Spigariol

Definición y medición de indicadores específicos para aulas virtuales de apoyo al dictado presencial 596
Marcela Andrea Vera, Lucila Romero

Estudio del perfil de rendimiento académico: un abordaje desde Data Warehousing 604
David La Red Martinez, Marcelo Karanik, Mirtha Giovannini, Noelia Pinto

Herramienta de seguimiento del rendimiento académico en la Facultad de Ingeniería (Sede Puerto Madryn) 613
Romina Stickar, Laura Biscayart, Sandra Alvarez, Sergio Salvatierra

Simulador de Redes Virtuales para Laboratorios de Sistemas RVL 621
Fabian Alejandro Gibellini, Ignacio José Sanchez Balzaretti; Germán Parisi

La visualización como estrategia pedagógica para la enseñanza del Análisis Matemático con GeoGebra 626
Roberto Daniel Lamas, Adelina García

Análisis de los Estilos de Aprendizajes de alumnos de Ingeniería Informática a través del Cuestionario CHAEA 636
Adelina García, Roberto Daniel Lamas

Programa de Tutorías Para Primer Año: Una experiencia con la intervención de monitores alumnos y las tecnologías como soporte comunicacional. 644
María Alejandra Odetti, Amalia Inés Haefeli, Alejandra María Jewsbury, Roberto Miguel Muñoz, Marcelo Martín Marciszack

Artículos cortos **650**

Construcción de Mapas conceptuales como estrategia didáctica en la enseñanza de la asignatura Paradigmas de Programación 650
Cynthia Lorena Corso

Estrategias en la enseñanza de Paradigmas de Programación: de la Programación Estructurada a la Programación Orientada a Objetos 657
Gustavo Eduardo Juarez, Luis Eduardo Nieto Peñalver, Mariana Sánchez, Jorge Federico Steifensand

Integración de Contenidos mediante Programación de Agentes Virtuales en Plataformas de Juegos 661
Valerio Frittelli, Felipe Steffolani, Romina Teicher, MarcelaTartabini, Gustavo Bett, Diego Serrano, Ana Strub, Julieta Fernández, Eduardo Destéfanis, Alexis Cuadrado, Mauricio Domenech

El Mundo Virtual de Matemática Discreta de la Facultad Regional Tucumán de la UTN 667
Rosana Hadad Salomón, Gladys Mónica Romano, Julia Beatriz Marisel Bedrán, Elizabeth María Alexandra Dufour, Mario Alberto Paredi

Propuesta de Mejora para la Enseñanza de Herramientas Informáticas en la UNAJ 671
Jésica Vanesa Guzmán, Diego Miguel Montezanti, Daniel Martín Morales

Entornos virtuales y herramientas de software libre como apoyo a la enseñanza de programación 677
Jesús Francisco Aguirre, Hugo José Viano, Berta García

Educación en Ingeniería 683
Cecilia Acevedo, Raúl Moralejo, Franco Catena

Gestión de Proyectos **689**

Artículos completos **689**

Simulación Dinámica de Gestión de Tareas en Proyectos Desarrollados con Scrum 689
Diego Alberto Godoy, Henry Kotynski, Edgardo Belloni, Eduardo O. Sosa

Un Enfoque Integrado para la Gestión de Riesgos y Cambios en la (Re)Planificación de Proyectos en Organizaciones Fractales 700
Laura Tosselli, Verónica Bogado, Ernesto Martínez

Proceso para Asistir la Migración de Información a Arquitecturas Cloud Computing en PyMEs 712
L. Bernal Tomadoni, P. Pytel, M. Pollo-Cattaneo

SPRABI: Una Herramienta para la Planificación, Control y Evaluación de Proyectos 722
Edgardo Bernardis, Germán Montejano, Luis Roqué, Hernán Bernardis, Mario Berón

Artículos cortos **733**

Mecanismos Institucionales de Transferencia Tecnológica 733
Patricia Zachman, Walter López, Andrés Redchuk

Metodologías y Guías de Gestión de Proyectos de TICs en el Sector Público: Enfoque Tradicional vs. Enfoque Ágil 739
Patricia R. Cristaldo, Luciana C. Ballejos, Mariel A. Ale

Gestión de los Stakeholders en un Proyecto destinado a que la ONU pueda dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones 744
Uzal, R., Riesco, D., Montejano, G., Sanchez, A., Agüero, W., Baieli, C.

Ingeniería de Sistemas y de Software

758

Artículos completos

758

Una Plataforma para Facilitar la Comprensión del Vocabulario y las Reglas de Negocio en el Desarrollo de Software 758
Ing. Guillermo Pantaleo, Ing. Ignacio Vazquez

Sintonización Estática de Parámetros Operacionales para un Modelo Paralelo de Evolución Diferencial 770
María Laura Tardivo, Paola Caymes Scutari, Miguel Méndez Garabetti, Germán Bianchini

Análisis de consistencia de Casos de Uso con simuladores de autómatas finitos 780
Marcelo Martín Marciszack, Oscar Carlos Medina, Claudia Castro, Enrique Humberto Moyano

Perspectivas de Usabilidad para Aplicaciones Móviles: Un Caso Práctico de Evaluación de Facebook 787
Lucas Santos, Pablo Becker, Luis Olsina

Herramientas de Análisis de Bordes para Reconocimiento y Clasificación de Objetos en Imágenes Digitales 799
Jorge Kamlofsky, María Lorena Bergamini

The Role of Stories in Software Engineering 805
Manuel Imaz

Generador de Código basado en GRAPHML 816
Cherencio, Guillermo Ruben

Automatización de la Selección de Cloud Applications Invocadas por un Workflow 826
Marcela Daniele, Fabio Zorzan, Ariel Arsaute y Ariel Gonzalez

Estrategia Multipropósito de Medición y Evaluación Orientada a Cambio y Mejora Continua 833
Belén Rivera, Pablo Becker, Luis Olsina

Aspectos para la definición de Reglas de Negocio 845
Castiblanco Sparano, Jose Luis; Baigorria, Lorena Soledad.

Towards Counterexample-Guided Computation of Validated Stability Certificates for Hybrid Systems 853
Eike Möhlmann and Oliver Theel

Factores que Determinan la Adopción de la Trazabilidad de Requerimientos Un Análisis Cualitativo 861
Juan C. Vázquez, Leticia E. Constable, Brenda E. Meloni, Juan F. Giró

Modelo de referencia para trazar procesos Scrum: Moodle como un caso de estudio 870
Roberto Nazareno, Silvio Gonnet, Horacio Leone

AId: Uma Ferramenta para Análise de Identificadores de Programas Java 880
Javier Azcurra, Augusto Farnese, Mario Berón, Pedro Rangel Henriques, Germán Montejano, Maria J. Varanda Pereira

QuVi: Un enfoque estadístico para la obtención de sistemas Web de Calidad 893
Nicolás Tortosa, Noelia Pinto, Liliana Cuenca Pletsch, César J. Acuña, Bruno Demartino

Ingeniería Inversa en el Contexto de ADM 900
Claudia Pereira, Liliana Martinez, Liliana Favre

Una Ontología para la Interoperabilidad de Modelos de Simulación 912
Mónica del Carmen Gil, Germán A. Montejano, Mario Berón

Detección de Conceptos en Programas Escritos con Lenguajes Multiparadigma 924
José L. Albanés, Mario M. Berón

Una Taxonomía de Atributos de Calidad para la Evaluación de Arquitecturas de Software por medio de Simulación 936
María Julia Blas, Silvio Gonnet, Horacio Leone

Un Modelo para Soporte de la Investigación Asistido por Técnicas de Visión Artificial 945
Jorge Alejandro Kamlofsky, María Lorena Bergamini

Priorizar Requisitos: un Estudio sobre sus Propósitos 953
Graciela D.S. Hadad, Gerardo A. Riera, Jorge H. Doorn

Extracción de información y cálculo de métricas en WSDL 1.1 y 2.0 963
Hernán Bernardis, Mario Marcelo Berón, Edgardo Bernardis, Daniel Edgardo Riesco, Pedro Rangel Henriques

Un Sistema de Anotaciones para la Especificación de Componentes de una Línea de Productos de Software 975
Matias Pol'la, Maximiliano Arias, Agustina Buccella and Alejandra Cechich

Detección de Agrupamientos en Glosarios del Universo de Discurso 987
Marcela Ridao, Jorge H. Doorn

TracEDaaS: Captura y Trazabilidad de Artefactos del Proceso de Diseño 996
Federico Hernandez, Luciana Roldán, Marcela Vegetti, Silvio Gonnet, Horacio Leone

Ontología para el Análisis Forense de Correo Electrónico 1008
Beatriz P. de Gallo, Marcela Vegetti, Horacio Leone

Representación de la evolución y refactoring de arquitecturas de software mediante la aplicación y captura de operaciones arquitectónicas 1019
María Luciana Roldán, Silvio Gonnet, Horacio Leone

Modelado de Procesos en el Ministerio de Educación, Ciencia y Tecnología de Catamarca 1031
Acosta Parra, Carlos Alberto; Manfredi, Silvia Elisa; Mazzurco, María de los Angeles; Ovejero, Marcelo Alejandro; Vilallonga, Gabriel Domingo; Zurita Perez, Franco

Modelo de calidad de productos de software 1043
Paula M. Angeleri, Alejandro Oliveros, Amos Sorgen, Rolando Titiosky, Jaquelina Wuille Bille

Una Estrategia Orientada a la Comprensión de Programas para Extracción y Análisis de Información Estática en Sistemas OO 1052
Miranda Enrique A.,Berón Mario M., Riesco Daniel E.

Web Application Frameworks Evaluation 1063
Ana Funes, Aristides Dasso

Análisis del comportamiento de los eventos de cambios a través de una ontología para gestionar versiones de una familia de producto 1071
Sonzini M. Soledad, Vegetti Marcela, Leone Horacio

Medición y Evaluación de Transformaciones a Nivel Metamodelo: un Enfoque Orientado al Mantenimiento 1083
Abdelahad Corina N.,Miranda Enrique A.,Perez Norma B.

Caracterización de los Riesgos inherentes a la Ingeniería Reversa 1095
Cuevas, Juan Carlos; Gastañaga, Iris; Gimenez Zens, Inés; Mana Franco

Testing en el Desarrollo de Software Científico en el Marco de la Integración Continua 1106
Salamon Alicia, Maller Patricio, Mira Natalia, Boggio Alejandra, Pérez Sofía, Coenda Francisco

Implementación de Sistemas de Información Basados en Ontologías: Análisis de Tecnologías 1114
Agustín Martínez, Santiago Sosa, Emiliano Reynares, Ma. Laura Caliusco

Framework Basado en Streams y Grafos para la Extracción de Modelos desde Código Fuente 1119
Paul Mendoza del Carpio

JInsight: Herramienta de Análisis Estático para la Comprensión de Programas 1129
Arnaldo Ceballos, Hernán Bernardis, Mario Berón, Daniel Riesco

Aceleradores de hardware para funciones de software en SoCKIT bajo Linux 1141
Esteban Peláez, Sergio D. Calderón R., Mario Berón

Artículos cortos **1145**

Transformación y obtención de Modelos Conceptuales mediante Léxico Extendido del Lenguaje y Escenarios 1145
Fernández Taurant, Juan Pablo – Marciszack, Marcelo Martín

Integrando SCRUM y CMMI en PyMES 1151
Sosa Zitto, Rossana; Pralong, Lourdes; Blanc, Rafael; Álvarez, Claudia.

Diseño de GUI: características deseables para su construcción 1157
M. Claudia Albornoz, Mario Berón, Germán Montejano

Aprendizaje Móvil para el Aprendizaje Organizacional en el Proceso de Desarrollo de Software 1164
Figueroa, Liliana; Rios, Miriam; Maldonado, Marilena; López, Gustavo

Análisis para la identificación de clusters en información recopilada de empresas de desarrollo de software sobre técnicas de priorización de requerimientos 1171
De Federico, Sara, Sincosky, Noelia, Lascano, Alfredo, Avogradini, Mariela, Moschetti, Diana

Cuadrantes de Testing Ágil 1177
Fernando Martín Córdoba

Un Análisis sobre la Asignación de Recursos en Procesos de Negocio 1182
Claudio Barua, Daniel Romero y Cynthia Villalba

Desarrollo de software robusto y flexible en el Paradigma de la Programación Orientada a Objetos 1188
Passerini, Nicolás; Spigariol, Lucas; Casanovas, Inés

A Graph Suite Generator for Real-World Quorum Protocol Analysis 1194
Robert Schadek and Oliver Theel

Aplicación de Mapa Cognitivo Compartido en Equipos de Desarrollo de Software Científico-Técnico 1200
Salamon Alicia, Maller Patricio, Mira Natalia, Boggio Alejandra, Pérez Sofía, Coenda Francisco

Redes – Sistemas Operativos – Fundamentos Informáticos **1206**

Artículos completos **1206**

Validando aplicaciones para Ciudades Inteligentes - Recolección de Residuos Urbanos 1206
Raúl Luft, Darío Sosa, Diego A. Godoy, Eduardo O. Sosa, Juan D. Benitez

Análisis del Impacto del Tráfico, de las Configuraciones de Red y de Seguridad en Redes Wi-Fi 1216
H. Facchini, S. Pérez, L. Bisaro, F. Hidalgo, G. Cangemi, Nora Costa

Virtualization Extensions into a Microkernel based Operating System 1225
Pablo Pessolani

Optimizando JavaScript a través de la reescritura de código 1237
Julieta Alvarez, Alexis Ferreyra, Ricardo Medel, Emanuel Ravera

Despliegue de una arquitectura de Cloud Computing híbrida Open Source 1242
Maria A. Murazzo, Nelson R. Rodríguez, Susana B. Chávez, Francisca A. Valenzuela, Adriana E. Martin, Miguel J. Guevara

Sistema de Comunicación para Renderización Paralela de Volúmenes en Tiempo Real 1250
C. F. Perez-Monte, M. F. Piccoli, C. Luciano, S. Rizzi

Impacto de la implementación de 802.11e para tráfico heterogéneo en MANET 1260
Maria A. Murazzo, Nelson R. Rodríguez, Miguel J. Guevara

Simulación de Algoritmos de Control de Congestión en TCP bajo User Mode Linux 1268
Rattalino Damián José, Crespo Aldo Abel

DNSCurve, una alternativa para la seguridad en el Sistema de Nombres de Dominio. 1278
Ernesto Sánchez, Daniel Arias Figueroa, Sergio Rocabado, Verónica Agüero, Gustavo Molina

Implementación y Operación de un Cluster HPC Utilizando Laboratorios de Computadoras en Horarios de Inactividad 1284
Ernesto J. Sale, Sebastián Rodríguez

Técnicas de Metaprogramación para el Diseño de EDSLs en C++0x11 1293
Marcelo Arroyo

Servidor de Comunicaciones Convergentes 1305
Antonio Castro Lechtaler, Rafael Mario Olivieri, Luis Fumagalli, Osaba Mariano, Francisco Echazú, Matías Bönke

Evaluación de la Interferencia para Esquemas de Radio Cognitiva mediante Plataformas de Radio Definida por Software 1312
Enrique Mariano Lizárraga, Gastón Peretti, Víctor Hugo Sauchelli, Walter Edgardo Herrera

Artículos cortos **1318**

Implementación de un servicio Peer to Peer Nativo en IPv6 1318
Isabel Marko, Víctor Fernández, Federico Valles, Pablo Vera, Rocío Rodríguez, Daniel Giulianelli, Graciela Cruzado

Seguridad Informática **1324**

Artículos completos **1324**

Specifying Security Characteristics, Attributes, and Metrics for Evaluating Web Applications 1324
Alexander Dieser, Guillermo Covella, Luis Olsina

Benchmarking y optimización de algoritmos criptográficos 1336
Miguel Montes, Daniel Penazzi

On the Interleaving Process Applied to the Trivium Algorithm 1344
Antonio Castro Lechtaler, Marcelo Cipriano, Edith García, Julio Liporace, Ariel Maiorano, Eduardo Malvacio

Implementation and Analysis of Opensource Honeypots Solutions 1351
Sebastián Norberto Mussetta, Norberto Gaspar Cena, Ignacio Daniel Favro

Aspectos Optimizables en un Protocolo Non - Interactive Dining Cryptographers 1356
Pablo García, Germán Montejano, Silvia Bast

Redes anónimas como instrumento para la conformación de un ciberataque sobre una infraestructura crítica 1365
Eduardo Casanovas, Maximiliano Delfino

Desarrollo de una guía de auditoria para la verificación de la calidad del software crítico en sistemas ferroviarios 1372
Jorge Esteban Eterovic, Domingo Donadello

Software Development Research – Current and Future

Narayan C. Debnath, Ph. D, D. Sc.

Department of Computer Science
Winona State University - Winona, MN 55987, USA
ndebnath@winona.edu
Director and Past-President
International Society for Computers and Their Applications (ISCA)
Minnesota, USA
http://www.isca-hq.org

Abstract

This keynote presentation provides a brief overview of software development research and outlines potential future research directions and challenges. Such an overview may help encourage young scholars and professionals, interested in their personal and professional growth in various topics including computer science and engineering, software engineering and developments, multimedia computing and systems, software and business managements, information technology and engineering, information and management science, and discrete mathematics and structures, to enhance possible collaborations among the scholars, professionals, and practitioners across the globe.
Computer software typically follows the lifecycle that begins with a *requirements specification* and analysis, followed by a *design* phase, followed by an *implementation* phase, and is ultimately *validated* for acceptance. Finally, the software enters a *maintenance* phase, where problems with the delivered product are corrected and further enhancements to the product are made. The primary focus of the presentation includes a brief outline of the current progress on each of the phases along with possible future research ideas and initiatives.
During the specification phase, we develop a careful specification for the problem to be solved. This includes specifications for the required inputs, outputs, and desired tasks. During the design phase, we decide how to perform these tasks and develop algorithms to perform the desired processing, and this is considered the most difficult phase. In the implementation phase, a programmer is given a complete specification of all inputs, all outputs, and all required algorithms/design. In coding, the programmer is expected to follow the detailed design and transform it into a program using a particular programming language.
The goal of validation is to establish confidence that the software should perform as intended. Software testing is considered as the most common, widely accepted and practiced method of validating computer software. During the testing phase, each module is tested using a wide range of data and finally we test the whole system with all modules combined. Software testing, a practical engineering activity, is essential for producing high-quality software and to perform quality assessment. Software Testing has been extensively studied in the literature. Researchers have been trying to come up with efficient testing theory, tools and techniques. Unfortunately, we are making very little progress and still much more work needs to be done in this area.
The maintenance phase involves additional testing to verify correctness as well as further enhancement of the system functionalities. In software maintenance research, attention has been directed toward reducing software cost. To this end, researchers have attempted to find relationships between the characteristics of programs and the difficulty of performing tasks. The objective has been to develop abstract models and measures of software complexity that can be used for cost projection, manpower allocation, and program and programmer evaluation. In the software models and metrics literature, a number of abstract models, essentially directed graph representations, of imperative language software have been proposed and analyzed to derive useful properties that may help understanding computer programs and thus allow better maintainability of software. In this presentation, a family of abstract models for representing imperative language software is described. The basic concepts, definitions, and some applications of these graph models are presented. These graphs may have wide range of applications in software engineering and technology. These graphs may serve as potential tools for (a) the study interconnection or integration complexity, an area that needs serious attention, (b) the study of compiler optimization, a topic to help compiler design and code optimization, (c) the study and analysis of concurrent programs, an area that may be useful to study the characteristics of concurrent programs, (d) the study and research in expression and statement complexity, an important topic in metrics research and may help to estimate program clarity, and (e) the study of the characteristics of software involving recursion, an area that needs to be addressed to compute the complexity of recursive programs. Once these models are defined with a sound theoretical basis and accurately validated, the entire family of graphs may serve as potential tools for further study and research in mathematics, computer science, software engineering and technology.

Evidently, a long road lies ahead towards the software development research progress and dealing with the current challenges. Therefore, a serious attention to all phases of software life cycle and careful execution of research activities in all directions may help widen the road to future research and quality software development.

Short Biography:

Dr. Narayan C. Debnath has been a Full Professor of Computer Science since 1989 and currently the Chairman of Computer Science at Winona State University, Minnesota, USA. He is also serving as the Director and Past-President of the International Society for Computers and Their Applications (ISCA). Dr. Debnath is a recipient of a Doctorate degree in Computer Science and a Doctorate degree in Applied Physics (Electrical Engineering). In the past, he served as the President, Vice President, and Conference Coordinator of the International Society for Computers and Their Applications (ISCA), and has been a member of the ISCA Board of Directors since 2001. He served as the Acting Chairman of the Department of Computer Science at Winona State University and received numerous Honors and Awards. During 1986-1989, Dr. Debnath was a faculty of Computer Science at the University of Wisconsin-River Falls, USA, where he was nominated for the National Science Foundation Presidential Young Investigator Award in 1989.

Dr. Debnath has made original research contributions in software engineering and development, software testing and quality assurance, software models, metrics and tools, and information science, technology and management. He is an author or co-author of over 300 publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering.

Professor Debnath has made numerous teaching and research presentations at various national and international conferences, industries, and teaching and research institutions in Asia, Australia, Europe, North America, and South America. He has been serving as an international teaching and research advisor/coordinator of the Master of Software Engineering Program at the National Universities in Argentina, South America, since 2000. He has offered courses and workshops on Software Engineering and Software Testing at the universities in South America, Asia, and Middle East.

Dr. Debnath served as the General Chair, Program Chair, invited Keynote Speaker, Tutorial Chair, and technical Track or Session Organizer and Chair of the international conferences sponsored by various professional societies including the IEEE, IEEE Computer Society, the Society of Industrial and Applied Mathematics (SIAM), International Association of Computer and Information Science (ACIS), International Association for Science and Technology in Education (IASTED), Arab Computer Society, and the International Society for Computers and Their Applications (ISCA). Dr. Debnath is a member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.

Ciber Paz: Protegiendo la Infraestructura Crítica Nacional ante Ciber Ataques y minimizando el Ciber Espionaje entre Estados Naciones

Prof. Dr. Roberto Uzal

Universidad Nacional de San Luis – Argentina
Ejército de los Andes 950
San Luis
Argentina

Agenda

1. Estados Naciones y agresiones en el Ciberespacio: Distintos posicionamientos
2. Ciber Agresiones entre Estados Naciones ¿Tiene solución tecnológica el "Problema de la Atribución"?
3. Defensa Cibernética desde un enfoque tecnológico: Principio, Métodos, Herramientas, Técnicas y Efectos
4. Derechos y Obligaciones de los Estados Naciones en el Ciberespacio.
5. El Derecho a la Legítima Defensa de los Estados Naciones ante Ciber Agresiones; el Artículo 51 de la Carta de las Naciones Unidas aplicado en el Ciberespacio
6. Libertad y Seguridad en el Ciberespacio
7. Las Naciones Unidas interviniendo en Ciber Conflictos entre Estados Naciones
8. Recapitulación y propuestas

Acerca del Conferencista:

- Director del Doctorado en Ingeniería Informática – Universidad Nacional de San Luis – Argentina
- Director de la Maestría en Ingeniería de Software – Universidad Nacional de San Luis – Argentina
- Director de la Maestría en Calidad del Software – Universidad Nacional de San Luis – Argentina
- Director de la Especialización en Ingeniería de Software – Universidad Nacional de San Luis – Argentina
- Investigador Categoría I - Programa de Incentivo a la Investigación
- Director del Laboratorio de Calidad e Ingeniería de Software de la Universidad Nacional de San Luis Argentina
- Coordinador, por Argentina, del Programa CAFES BA de integración de la enseñanza de IV nivel con Brasil (acuerdo de co titulación – co tutela de tesis de maestría entre la Universidad Nacional de San Luis y la Universidad Federal de Minas Gerais
- **Actividades recientes relacionadas con Ciber Defensa**
- Conferencia en el Centro de Oficiales de las FFAA: "Nuevos escenarios, nuevos conflictos, nuevas armas, nuevos soldados: Cyber War"; 24 de abril de 2012
- Conferencia en el Consejo Profesional de Ingenieros Electrónicos y en Telecomunicaciones – COPITEC – "Guerra Cibernética", 4 de julio de 2012
- Artículo de opinión sobre "Cyber War" en la Revista DEF - Agosto de 2012
- Reportaje sobre Cyber War - Canal de TV C5N el 19 de Agosto de 2013 a las 23 horas.
- Reportaje sobre "Guerra Cibernética" realizado por la Revista DEF ON LINE – Agosto de 2012 http://www.defonline.com.ar/?p=9064
- Consejo Argentino de Relaciones Internacionales, seminario sobre "Crimen Organizado Transnacional y Ciberamenazas", 11 de noviembre de 2012
  http://www.cari.org.ar/pdf/crimenorganizado-uzal-2013.pdf
- Sociedad Científica Argentina. Seminario sobre Guerra Cibernética dirigido a los Oficiales del Ejército de la Especialidad Informática; 21 de noviembre de 2012.
- Exposición en la Universidad Nacional de San Luis en el contexto de las Jornadas por un Ciberespacio más seguro y confiable: "Ciberamenazas, escenarios y propuestas"; 18 de diciembre de 2012. Esta actividad fue organizada en forma conjunta con la Cancillería de Argentina http://webfmn.unsl.edu.ar/boletines/boletin418/boletin418.html
- Una síntesis de la ideas desarrolladas en las actividades detalladas más arriba se plasmó en un "paper" titulado "Trust in Cyberspace: New Information Security Paradigm" el cual fué presentado en la Conferencia IEEE que se realizó los días 22 y 23 de marzo de 2013 en Hangzhou, China. El trabajo fue aceptado arbitraje internacional mediante

- Artículo en la Revista “Visión Conjunta” de la Escuela Superior de Guerra Conjunta de las FFAA de Argentina, “Guerra Cibernética: ¿Un Desafío para la Defensa Nacional?” http://esgcffaa.mil.ar/numero7/index.html
- Asistencia al Estado Mayor Conjunto de las FFAA de Argentina (Ministerio de Defensa) en el Planeamiento Estratégico de la Ciber Defensa. Primer semestre de 2013
- Seminarios realizados en la Escuela Superior de Guerra Conjunta de las FFAA de Argentina en 2012 y 2013 (Coroneles, Capitanes de Navío y Comodoros de la Fuerza Aérea de Argentina). Algunos de dichos seminarios constituyeron el “kick off” de ejercitaciones realizadas en dicho instituto superior.
- Seminarios organizados por The Armed Forces Communications and Electronics Association (AFCEA) / Facultad de Ingeniería del Ejército (Escuela Superior Técnica):
    - Ciber Seguridad en Dispositivos Móviles 23 y 24 de Octubre de 2013 http://afcea.org.ar/?p=1336
    - Ciber Defensa: Estado actual y tendencias 21 y 22 de Mayo de 2014 (ver apuntes, texto y transparencias) http://afcea.org.ar/?p=1649
- Propuestas de extensión del trabajo de tesis del Doctor Mario Berón, “Doctor en Ciencias de la Computación” otorgado por la Universidad Nacional de San Luis - Argentina / “Doutor en Ciencias da Computação” otorgado por la Universidad de Minho, Portugal (Reingeniería y Comprensión de Programas). La mencionada extensión posibilitará encarar la Reingeniería de
- Ciber Armas. Supervisores de la doble titulación del Dr. Berón: Roberto Uzal (Argentina) / Pedro Rangel Henriques (Portugal)
- Co supervisor (UNSL - Argentina) con el Prof. Dr. Daniel Macedo (UFMG – Brasil) de la Tesis de Maestría del Lic. Walter Agüero. La tesis está orientada a dotar a países de la Región de instrumentos que permitan un efectivo ejercicio del derecho enunciado en el Artículo 51 de la Carta de las Naciones Unidas (tesis a ser defendida en los próximos meses).
- Co supervisor (UNSL - Argentina) con el Prof. Dr. Italo Cunha (UFMG – Brasil) de la Tesis de Maestría del Lic. Claudio Baieli. La tesis está orientada a dotar a países de la Región de instrumentos que permitan detectar Ciber Agresiones / Ciber Espionaje (tesis a ser defendida en los próximos meses)
- Co Autor del Capítulo 19 del libro Managing Trust in Cyberspace - CRC Press Book (Diciembre de 2013) ISBN: 978-1-4665-6844-0. eBook ISBN: 978-1-4665-6845-7
  http://www.amazon.com/Managing-Trust-Cyberspace-Sabu-Thampi/dp/1466568445
- Presentaciones de trabajos en las Jornadas Argentinas de Informática (arbitraje / evaluación internacional)
    - 2013 (Universidad Nacional de Córdoba) “Planeamiento Estratégico Informático: Planeamiento Basado en Capacidades aplicado al Planeamiento Estratégico de la Ciber Defensa”
    - http://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/SIE/22.pdf
    - 2014 (Universidad de Buenos Aires / Universidad de Palermo) “Conflictos en el Ciber Espacio entre estados naciones: Potenciales aportes para la eventual actuación de las Naciones Unidas y de la Unión Internacional de las Telecomunicaciones” a ser expuesto del 1 al 5 de Septiembre de 2014.
    - 2014 Consejo Federal de Decanos de Ingeniería (CONFEDI) – Congreso Nacional de Ingeniería Informática y Sistemas de Información (CoNaIISI - UNSL) “Gestión de los Stakeholders en un Proyecto destinado a que la ONU pueda dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones”
- Participante de NETmundial, la reunión de múltiples partes interesadas para la gobernanza de Internet (Sao Paulo – Brasil, 23 y 24 de Abril de 2014). Convocado por el Comité Académico. Interviene cinco veces intentando refutar la postura “anti acuerdo de no proliferación de Ciber Armas” y “anti acuerdo de Control de Ciber Espacio para evitar agresiones entre estados naciones” sostenidas por la delegación de EEUU. Produjo dos informes para el Gobierno Argentino.
- Expositor en la “Jornada sobre Ciberdefensa y Seguridad de Redes de Comunicaciones” organizada por el Ministerio de Defensa en el Aula Magna de la Escuela Superior Técnica del Ejército el 2 de septiembre de 2014. Título de la exposición: Análisis de Flujo de Redes para la detección de Amenazas Cibernéticas

Métodos Ágiles y Modelos de Madurez para el Desarrollo de Empresas de Software

Jorge Boria - Viviana Rubinstein

LiveWare Inc. Software Engineering
Austin, Texas, USA

Abstract

En la medida en que los métodos ágiles y los modelos de madurez de procesos han sido considerados en las antípodas uno del otro, esta charla se propone como una alternativa que los reune. Se habla principalmente de cuatro implementaciones de ágil: Kanban, Scrum, XP y FDD (Feature Driven Development) y del modelo de madurez de mejora de procesos del software del Brasil, el MPS-SW.
En torno de la cuestión del cambio organizacional, seguiremos el camino que marca la metodologia Lean, que se concentra principalmente en la resolución de problemas a través de la mejora de procesos. Esa constituye nuestra estrategia. Como parte integral de la solución táctica utilizaremos Kanban como el paso inicial en un proyecto de mejora de procesos; luego Scrum para iniciar actividades de gestión más tradicionales; XP (Extreme Programming) para lo que constituye la categoría de las áreas de ingeniería de productos tratadas en los modelos CMMI-DEV y el MPS- SW. Cuando una empresa crece y creccen sus proyectos en volumen, los métodos anteriores comienzan a mostrar limitaciones. La coordinación de muchos grupos en Scrum de Scrums presenta mayores limitaciones que los métodos tradicionales. XP es difícil de controlar. Para mantenerse en los métodos ágiles, una empresa que crece puede utilizar Feature Driven Development (FDD).

Acerca de los Conferencistas:

Jorge Boria: *Master of Engineering in Computer Science* de la Universidad Cornell, Estados Unidos. *Visiting Scientist* del Software Engineer Institute de la Universidad Carnegie-Mellon por cinco años. *Senior Advisor* del Modelo MPS. madurez DEV y SVC. *Instructor certificado* de los cursos del modelo CMMI DEV y SVC. Fue *Profesor Titular* de las Universidades UBA, UNICEN, UNSL, USal, UB y otras de la Argentina y la ROU.

Viviana Rubinstein: *Licenciada en Computación Científica* de la FCEyN de la UBA. *Certified Project Manager*, de la Universidad de Texas - SQI. *Certified SCAMPI Lead Appraiser* para alta madurez DEV y SVC. *Instructora certificada* de los cursos del modelo CMMI DEV y SVC. Fue *Profesora Titular* en la UNSur - Ushuaia, UBA, UNICEN y varias otras de la Argentina y la ROU. Conferenciante en numerosos congresos del tema.

Reconocimiento de Patrones en Registro de Pozos

Manuel Romero Salcedo[1], Elizabeth Santiago del Ángel[1]
Juan Carlos Ontiveros Neri[2]

1.- Matemáticas Aplicadas y Computación
Instituto Mexicano del Petroleo
México, D.F., Méx. 07730
mromeros@gmail.com ; esangel@imp.mx
2.- Academia de Ingeniería en Sistemas Computacionales
Instituto Tecnológico Superior Zacatecas Occidente
Sombrerete, Zac, Méx. 99100
ontiverosneri@yahoo.com.mx

Abstract
"En este trabajo se presenta una metodología para determinar la mejor topología de una red neuronal perceptrón con la finalidad de aplicarla en el reconocimiento de patrones en registro de pozos petroleros de exploración, además del desarrollo de una herramienta de software basada en el TOOLBOX de matlab, para generar y entrenar redes neuronales"

Palabras Clave
Redes Neuronales, Algoritmo backpropagation, Registro de Pozos, Matlab.

Introducción

Se presenta el desarrollo de una herramienta de software en MATLAB, para generar y entrenar redes neuronales, con el fin de aplicarla en estudios del subsuelo, es decir, en el conocimiento de las características y composición de las capas del subsuelo mediante la perforación de pozos conocidos como exploratorios [1]. Esta herramienta busca reconocer patrones en datos extraídos de los pozos exploratorios a partir de bases de datos generadas de pozos analizados anteriormente.

El uso de métodos tradicionales a menudo limita las soluciones de los problemas de las industrias y mediante varios trabajos de investigación [2-4], se ha demostrado que la aplicación de nuevas tecnologías, específicamente las redes neuronales, permite modelar, analizar y optimizar los procesos industriales.

Elementos del Trabajo y metodología

Herramienta de Software:
Programa 1

```
%DESCRIPCIÓN DE VARIABLES
% Train: matriz de los datos de entrenamiento
% Test: matrix de los datos de prueba o validación
% R : modelo de ajuste obtenido
%NOTA: los datos de entrada son introducidos por el prompt de Matlab
%introduciendo las matrices completas (Train y Test)
function [R]= NetFF1 (Train,Test)
In=Train(:, [1,2,3,4,5,6]);  Out=Train(:,7);
IN=In';  OUT=Out';  TEST=Test';
    %EJECUCIÓN DE LA ANN
    NET=newff(IN,OUT,[8 8]);    %CREACION DE LA TOPOLOGIA DE LA ANN
    for i=1:10                  %REPITE PROCESO DE ENTRENAMIENTO 10 VECES
    net=train(NET,IN,OUT);              %FASE DE ENTRENAMIENTO
    [PP, Pf, Af, E, perf] = sim(net,TEST);
    %FASE DE VALIDACION O PRUEBA
    % PP : SALIDA DE LA ANN
    % Pf: CONDICIÓN DE RETRASO EN LA ENTRADA FINAL DE DATOS
    % Af: CONDICION DE RETRASO EN LA CAPA FINAL
    % E : ERROR DDE LA RED
    % perf : DESEMPEÑO DE LA RED
    P=PP';   M(:,i)=P;
    end
    N=M';
    R=sum(N)/10;        %CÁCLULO DE PROMEDIO DE LAS 10 ITERATIONES
    Matrizres(:,cont)=R';
end
xlswrite('C:\Program Files\MATLAB\R2008a\work\salidaModelo1.xls', Matrizres,'Hoja1','A3');
%GENERA ARCHIVO EXCEL CON EL MODELO DE AJUSTE RESULTANTE
```

En las primeras líneas del código se muestran las instrucciones para cargar los datos con los cuales funciona el programa.

El código ejecuta la misma topología de la red 10 veces (en este caso una red con dos capas, cada una con 8 neuronas) y al finalizar saca un promedio de los 10 resultados arrojados con el fin de sacar un vector promedio representativo de las mismas, posteriormente se crea un archivo en Excel para su recuperación.

Programa2

```
%DESCRIPCIÓN DE VARIABLES
% Train: matriz de los datos de entrenamiento   Test: matrix de los datos de prueba o validación
% R : modelo de ajuste obtenido   NOTA: los datos de entrada son introducidos por el prompt de Matlab
%introduciendo las matrices completas (Train y Test)
function [R]= NetFF1 (Train,Test)
In=Train(:, [1,2,3,4,5,6]);   Out=Train(:,7);   IN=In';   OUT=Out';   TEST=Test';
       %EJECUCIÓN DE LA ANN
cap1=2;   cont=0;   dimensionco=10; % TOPE DE NEURONAS
while cap1<=dimensionco, %VARÍA LAS NEURONAS DE LA PRIMER CAPA OCULTA
    cap2=cap1;   cont=cont+1;
   while cap2<=dimensionco, %VARÍA LAS NEURONAS DE LA SEGUNDA CAPA OCULTA
     NET=newff(IN,OUT,[cap1 cap2]);  %CREACION DE LA TOPOLOGIA DE LA ANN
       for i=1:10              %REPITE PROCESO DE ENTRENAMIENTO 10 VECES
          net=train(NET,IN,OUT);       %FASE DE ENTRENAMIENTO
          [PP, Pf, Af, E, perf] = sim(net,TEST);
          %FASE DE VALIDACION O PRUEBA
          % PP : SALIDA DE LA ANN   Pf: CONDICIÓN DE RETRASO EN LA ENTRADA FINAL DE DATOS
          % AfF: CONDICION DE RETRASO EN LA CAPA FINAL   E : ERROR DDE LA RED
          % perf : DESEMPEÑO DE LA RED
          P=PP';   M(:,i)=P;
       end
       N=M';   R=sum(N)/10;  %CÁCLULO DE PROMEDIO DE LAS 10 ITERATIONES
       Matrizres(:,cont)=R';   cap2=cap2+2;   cont=cont+1;   M=zeros(size(M));
   end
cap1=cap1+2;
end
for k=1:(Rank(Matrizres))          %DEPURA LA MATRIZ SI HAY COLUMNAS CEROS
   if (Matrizres(1,k)==0)
       Matrizres(:,k)=[];
   end
end
xlswrite('C:\Program Files\MATLAB\R2008a\work\salidaModelo1.xls',Matrizres,'Hoja1','A3'); %GENERA ARCH
```

Con el fin de explorar diferentes topologías se incorporaron al código dos ciclos (ver Programa 2), los cuales varían el número de neuronas de las capas ocultas de la red. Los vectores promedio resultantes de cada una de las topologías se recuperan en un archivo de Excel.

Programa 3

```
function [R]= NetFFNORERRITER (Train,Test,VectorR)
errormin=1000;
iter=1;
while (errormin>1.12)
for l=1:(length(Train(1,:)))            %NORMALIZA LA MATRIZ Train
    menor=min(Train(:,l));
    mayor=max(Train(:,l));
    TrainN(:,l)=(Train(:,l)-menor)/(mayor-menor);
end
 for p=1:length(Test(1,:))            %NORMALIZA LA MATRIZ Test
    menor=min(Test(:,p));
    mayor=max(Test(:,p));
    TestN(:,p)=(Test(:,p)-menor)/(mayor-menor);
end
In=TrainN(:, [1,2,3,4,5,6]);
Out=Train(:,7);
IN=In';
OUT=Out';
TEST=TestN';
       %EJECUCIÓN DE LA ANN
dimensionco=10;
cap1=2;
cont=0;
while cap1<=dimensionco,     %VARÍA LAS NEURONAS DE LA PRIMER CAPA OCULTA
    cap2=cap1;
    cont=cont+1;
    while cap2<=dimensionco,%VARÍA LAS NEURONAS DE LA SEGUNDA CAPA OCULTA
       NET=newff(IN,OUT,[cap1 cap2]); %CREACION DE LA TOPOLOGIA DE LA ANN
       for i=1:iter             %REPITE PROCESO DE ENTRENAMIENTO 10 VECES
       net=train(NET,IN,OUT);      %FASE DE ENTRENAMIENTO
       [PP, Pf, Af, E, perf] = sim(net,TEST);%FASE DE VALIDACION O PRUEBA
       % PP : SALIDA DE LA ANN
       % Pf: CONDICIÓN DE RETRASO EN LA ENTRADA FINAL DE DATOS
       % AfF: CONDICION DE RETRASO EN LA CAPA FINAL
       % E : ERROR DDE LA RED
       % perf : DESEMPEÑO DE LA RED
            P=PP';
            M(:,i)=P;
        end
        N=M';
        R=(sum(N)/iter);       %CÁCLULO DE PROMEDIO DE LAS 10 ITERATIONES
        Matrizres(:,cont)=R';
    cap2=cap2+2;
    cont=cont+1;
    M=zeros(size(M));               %LIMPIA LA MATRIZ M
    end
cap1=cap1+2;
end
cont=1;
for k=1:length(Matrizres(1,:))           %DEPURA LA MATRIZ SI HAY COLUMNAS CEROS
    if (Matrizres(1,k)~=0),
        Matrizresaux(:,cont)=Matrizres(:,k);
        cont=cont+1;
    end
end
Matrizres=zeros(size(Matrizres));
Matrizres=Matrizresaux;
cont=1;
for k=1:(length(VectorR))          %DEPURA EL VECTOR
    if (VectorR(k)>(-1)),
        VectorRaux(cont)=VectorR(k);
        cont=cont+1;
    end
end
for k=1:(length(Matrizres(1,:)))          %CALCULA EL ERROR
    VectorAux=abs(Matrizres(:,k)-VectorR);
    MatrizDif(:,k)=VectorAux;
    y = VectorR
    yhat = Matrizres(:,k);
    RMSE(k) = sqrt(mean((y - yhat).^2));
    %Error(k)=sum(MatrizDif(:,k))/length(VectorR);
end
errormin=min(RMSE);
xlswrite('C:\Program Files\MATLAB\R2008a\work\salidaModelo1.xls', Matrizres,'Hoja1','A3');
%GENERA ARCHIVO EXCEL CON EL MODELO DE AJUSTE RESULTANTE
xlswrite('C:\Program Files\MATLAB\R2008a\work\salidaModeloMATRIZDIF.xls' ,MatrizDif,'Hoja1','A3');
%GENERA ARCHIVO EXCEL CON EL MODELO DE AJUSTE RESULTANTE
xlswrite('C:\Program Files\MATLAB\R2008a\work\salidaModeloNORMERR1.xls', RMSE,'Hoja1','A3');
%GENERA ARCHIVO EXCEL CON EL MODELO DE AJUSTE RESULTANTE
Matrizres=[];
MatrizDif=[];
Matrizresaux=[];
VectorRaux=[];
iter=iter+1;
end
```

finalmente se incorporó al código el cálculo del RMSE para determinar el error medio entre los vectores promedio resultantes de cada una de las topologías y el vector esperado (ver Programa 3), además de la opción de fijar el error permitido para optimizar la búsqueda de la topología apropiada.

METODOLOGÍA

Se trabajaron dos casos de estudio:

- Caso1: 2 parámetros
- Caso2: 1 parámetro

Para los dos casos se realiza el siguiente procedimiento:

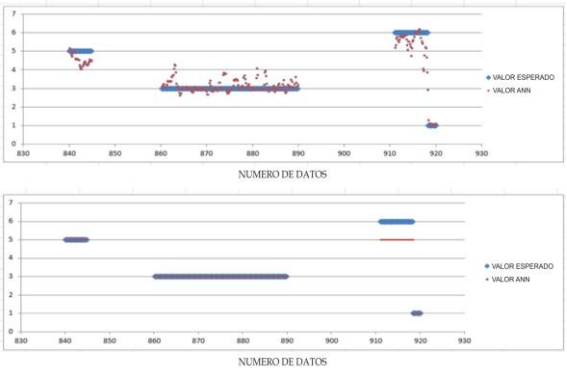
Primera Etapa: Tomar intervalos al azar en tamaño y posición de los datos de un pozo analizado y con éstos entrenar y probar la red, una vez que obtengamos los resultados podremos determinar si efectivamente la red aprendió, en este proceso la matriz a probar tendrá que ser igual a la matriz de entrenamiento y con ello esperamos que la red responda de una forma aceptable.
Segunda etapa: Tomar la matriz de datos del pozo que no se utilizaron en la primera etapa y probarlos en la red neuronal para garantizar que la red aprendió y que puede responder de una forma aceptable a datos desconocidos para ella.
Tercera Etapa: Tomar todos los datos del pozo para realizar el entrenamiento de la red e introducir los datos a probar de un pozo de exploración nuevo que aún no haya sido analizado para intentar inferir la información del mismo.

En todas las etapas se aplica el siguiente algoritmo:
1.- Cargar los datos en la red.
2.- Con el código de Matlab entrenar varias topologías de red.
3.- Determinar cuál topología es la adecuada.
4.- Utilizar la red con la mejor topología para probar los datos.

Resultados
Aplicando la metodología y el software tenemos:

Caso 1

Primer Etapa

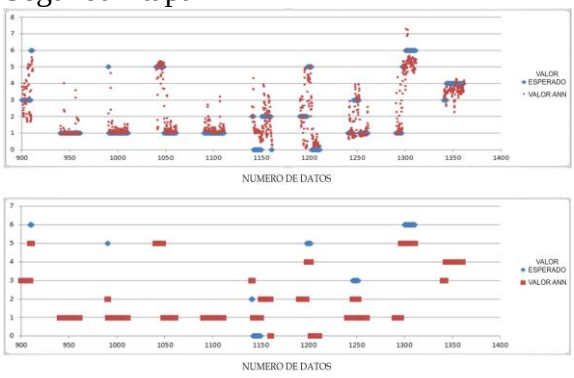


Resultados Primer Etapa (Caso 1)

Se puede apreciar en la gráfica como los valores arrojados por la red (puntos), se aproximan al valor esperado en cada uno de los bloques de profundidad. Para concentrar los resultados arrojados por la red, obtuvimos el promedio de los datos por bloques de profundidades y posteriormente redondeamos.

De los cuatro valores esperados en la primera etapa, la red coincide en tres y en el que falló existe una diferencia de uno con respecto al esperado, es decir se tuvo una efectividad del 75%.

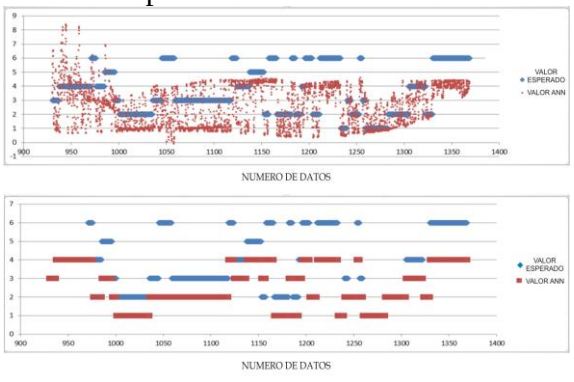
Segunda Etapa



Resultados Segunda Etapa (Caso 1)

En la segunda etapa la red coincidió en quince valores de veintidós, se tuvo una efectividad del 68%, cabe destacar que los valores arrojados por la red tienen una diferencia de una unidad con respecto al valor esperado, por lo que se puede apreciar que la red predice de buena manera aun cuando trabaja con datos completamente desconocidos por ella, datos que no fueron utilizados para su entrenamiento.

Tercer Etapa



Resultados Tercer Etapa (Caso 1)

En la tercera etapa de treinta y dos valores la red no logra coincidir en veintiséis, se tuvo una efectividad del 19%, la red no responde adecuadamente en datos extraídos de pozos nuevos, sin embargo su comportamiento es de acuerdo a la distribución de los datos esperados.

Caso 2

En el caso 2 seguimos aplicando la técnica del caso 1 con una leve mejora, se declara una función denominada SUMA la cual se define como la suma del error promedio y la varianza, el error promedio mide la distancia de los resultados arrojados de la red con respecto a los esperados, mientras que la varianza nos indica la estabilidad de la red al arrojar los datos.

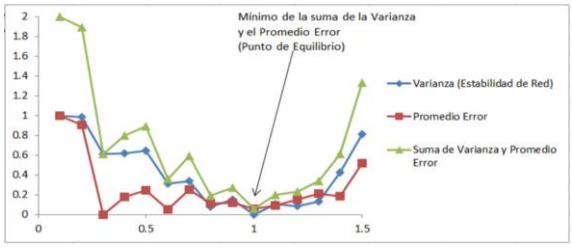
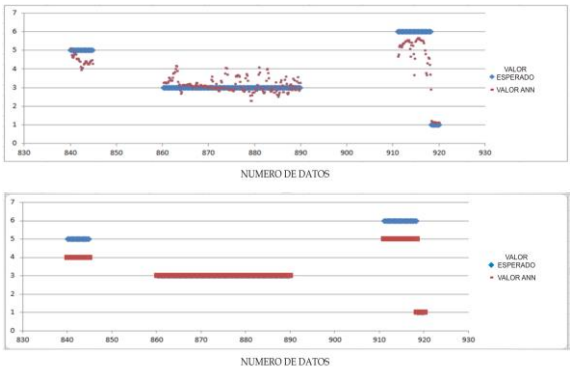
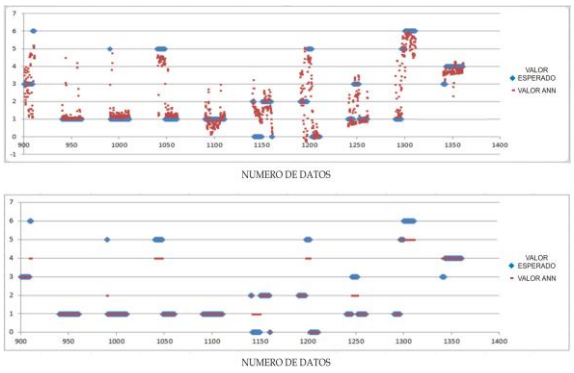


Figura 1

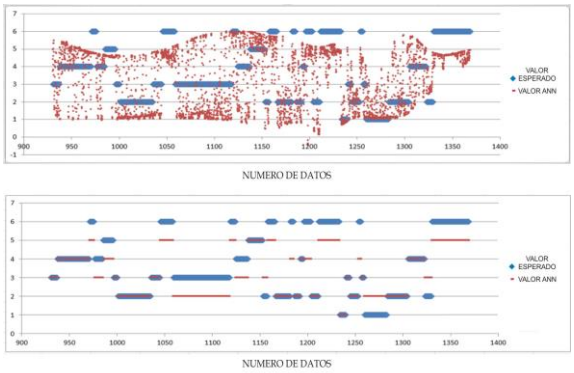
La topología que se utiliza para probar los datos es seleccionada a partir del mínimo de la función SUMA (Figura 1), si se minimiza dicha función se obtiene un valor de equilibrio entre el error y la estabilidad de la red, ya que si perdemos error, ganamos estabilidad y viceversa.



Resultados Primer Etapa (Caso 2)



Resultados Segunda Etapa (Caso 2)



Resultados Tercer Etapa (Caso 2)

En la primera etapa los resultados arrojados por la red coinciden en dos valores esperados y falla en los otros dos con una diferencia de una unidad, se obtiene una efectividad del 50% en la red. Cabe mencionar que cuando el promedio de los valores de la red está dentro del rango (“valor esperado” - 0.5) ó (“valor esperado” + 0.5) se puede considerar un punto crítico, ya que una pequeña variación puede hacer que coincida con el valor o que se aleje una unidad, por esta razón los resultados de la red se deben de interpretar a partir de la técnica usada para redondear.

Existen diferentes métodos para el análisis y mejoramiento del redondeo, por ejemplo se puede ver como algoritmos recursivos[6] o como Series y sucesiones[7], ambos proporcionan herramientas matemáticas que hacen que se logren mejores resultados a la hora de redondear.

En la segunda etapa la red arroja resultados satisfactorios al coincidir en quince de veintitrés valores esperados, se tiene una efectividad del 65%.

Al hacer el análisis en los resultados de la segunda etapa se observa que cuando la red tiene que predecir en intervalos pequeños de profundidad, no es confiable, mientras que en bloques grandes los resultados son más exactos, esto nos lleva a pensar que a medida que se le proporcionen más datos a la red, aprende mejor.
En la tercera etapa, de un total de treinta y dos valores esperados la red arrojó dieciséis correctos, se tiene una efectividad del 50%. De los resultados erróneos trece se encuentran como valores críticos al haber una diferencia de una unidad con respecto al valor esperado y en tres valores hay una diferencia de dos unidades.

Los resultados obtenidos en el caso 2 mejoraron con respecto al caso 1, más no lo suficiente, dado que la efectividad de la red sigue siendo baja.

Conclusiónes

Las redes neuronales aplicadas según la metodología descrita en este artículo, muestran resultados alentadores en el reconocimiento de patrones en registro de pozos de exploración, sin embargo no son óptimos.

Una ventaja que se puede apreciar en los casos que se experimentaron es que al corregir la red neuronal desde su entrenamiento, puede ser suficiente para obtener buenos resultados en la predicción de patrones en pozos de exploración no analizados.

Se pueden explorar redes perceptrón con tres o más capas ocultas y un número mayor a diez neuronas por cada capa, ya que en este trabajo solo se probaron topologías de red con dos capas y un número menor o igual a diez neuronas.

Otra opción es probar con diferentes funciones de activación en las neuronas y con diferentes algoritmos de entrenamiento ya que solo se probó el algoritmo backpropagation, además de que existen otros tipos de redes neuronales que no se abordaron en este trabajo [5].

La técnica de redondeo tal vez no sea la más apropiada para concentrar los resultados arrojados por la red, se puede utilizar otro tipo de método, por ejemplo la función mayor o menor entero o el truncamiento.

Como trabajo futuro se está buscando la aplicación de esta metodología y esta herramienta tecnológica en la minería.

Referencias

[1] PEMEX, “Base de datos institucional: Términos usados en la industria petrolera”, © 2014 Petróleos Mexicanos
[2] T NABIL AL-BULUSHI, PETER R. KING, MARTIN J. BLUNT, MARTIN KRAAIJVELD. Development of artificial neural network models for predicting water saturation and fluid distribution, Journal of Petroleum Science and Engineering 68 (2009) 197–20
[3] FRED K. BOADU. Predicting oil saturation from velocities using petrophysical models and artificial neural network , Journal of Petroleum Science and Engineering 30 (2001) 143–154
[4] E.M.EL-M. SHOKIR, A.A. ALSUGHAYER, A. AL-ATEEQ. Permeability Estimation from Well Log Responses, Canadian International Petroleum Conference Journal of Petroleum Science and Engineering 012 (2005)
[5] MATHWORKS. Accelerating the pace of engineering and science, Neural Network ToolBox 012 (2005)
[6] M. BONACINA, A. HAIDAR, C. TETI. Las Nuevas Tecnologías y el Tratamiento del Error, Facultad de Ciencias Bioquímicas f Farmacéuticas – U.N.R.
[7] A. CASARES M. Estimación y Control del Error, Escuela de Sistemas, 2do semestre 2005-2006. (2006)

Módulo adaptador universal para Simulación Inmersiva

[1] Alejandro Arroyo Arzubi; [1,2] Carlos Federico Blanc; [1] Francisco Brandolini; [1,2] Damián Alejandro Camarasa; [2,3] Antonio Castro Lechtaler; [2] César Daniel Cicerchia; [1,2] Enrique Jorge Matías Luzuriaga; [1] Alexis Gabriel Polak [1]

[1] CIDESO - Centro de Investigación y Desarrollo de Software, DIGID; [2] Escuela Superior Técnica - Facultad de Ingeniería del Ejército, IUE - Cabildo 15, Ciudad Autónoma de Buenos Aires, C1426; [3] Universidad Nacional de Chilecito - 9 de julio 22, Chilecito, La Rioja, F5360

aarroyo_arzubi@hotmail.com, ing.carlos.blanc@gmail.com, franbrandolini@gmail.com, dcamarasa@ejercito.mil.ar, acastro@iese.edu.ar; cdcicerchia@iese.edu.ar, mluzuriaga@ejercito.mil.ar, apolak@ejercito.mil.ar

Resumen

La industria de los videojuegos siempre intentó mejorar su negocio, elevando la calidad de los gráficos y del sonido, así como la manera en la que se utiliza un juego. InmerSim es un proyecto que se desarrolla con la finalidad de alcanzar una mayor ventaja competitiva, incorporando tecnologías de realidad virtual para optimizar la experiencia del juego, e integrando software libre y de código abierto, así como hardware de diseño público, para disponer de independencia tecnológica a bajo costo.

1. Introducción

El proyecto InmerSim[1] [1] [2] [3] busca desarrollar componentes de software de simulación "inmersiva" para el usuario de juegos o simuladores "en primera persona"[2].

Una de las técnicas para lograr un escenario de simulación con elevado realismo es la que se ha denominado "*inmersiva*". Este método utiliza una computadora para crear un ambiente tridimensional y producir una realidad virtual con la cual el jugador interactúa por medio de guantes, cascos u otros dispositivos. Mediante estos sensores se capturan la posición y los movimientos de diferentes partes del cuerpo humano, que producen "entradas" en el ambiente virtual y lo modifican; o también, reciben "respuestas" que se perciben por los sentidos del jugador.

Existen diversos periféricos que asisten a tener una simulación más inmersiva [4] [5]. Entre ellos podemos destacar ~~a~~ los dispositivos de video como los proyectores 3D, los anteojos de realidad virtual[3] y los elementos de detección de movimiento, como son el *Kinect* de *Xbox* de la empresa *Microsoft*®, el *PS Move* de *Sony*® *Playstation* o el *Wiimote* de *Nintendo*®, entre otros.

Desde nuestro punto de vista, estos últimos periféricos tienen dos problemas importantes:

- Los controles son propietarios de la empresa que las fabricó y sólo funcionan para una consola específica.
- Los juegos/simuladores deben estar específicamente desarrollados para este tipo de dispositivos.

Por otra parte, el bajo costo de la tecnología de sensores y de las placas con micro-controladores, como *Arduino*, *BeagleBone*, *Raspberry Pi* y *PCDuino* entre otros, permite la incorporación de la tecnología de realidad virtual a la industria de los videojuegos, de una manera económica, simple y empleando *hardware* denominado "libre[4]".

El proyecto InmerSim creará un prototipo que soluciona estos problemas, mediante la construcción de un módulo - *hardware* más *software* - que es independiente de la consola utilizada, y además, puede ser adaptado a cualquier juego o simulador ya existente.

2. Objetivo del prototipo

El objetivo del trabajo es construir un prototipo compuesto de un módulo adaptador universal para

[1] Financiado y ejecutado por la Escuela Superior Técnica - Facultad de Ingeniería del Ejército – Grl Div Manuel N. Savio

[2] Esta técnica utiliza una vista desde la cual el mundo simulado se muestra desde la perspectiva del protagonista. Es de uso frecuente en juegos de disparo o de aventuras.

[3] Por ejemplo, se puede mencionar ***Oculus Rift*** de la empresa *Oculus VR*® (adquirida por *Facebook*®).

[4] Por similitud al *software* libre, sus especificaciones y modelos de diseño son públicos, disponibles totalmente gratis o mediante un pago de menor valor.

simulación inmersiva que permita mejorar la experiencia del juego/simulador.

Dicho módulo estará compuesto por una plaqueta embebida que brindará una interfaz para adaptar cualquier juego de consola[5] o de una PC a un ambiente inmersivo, mejorando la experiencia del usuario.

Para el prototipo, se utilizará una consola *Xbox* 360 desde la cual correrá el juego deseado (en esta etapa no se desarrollará ningún simulador y se empleará un videojuego comercial). Este deberá aceptar juegos tanto de la consola *Xbox 360* como de la PC. En este último caso, se necesitará un receptor *joystick*[6].

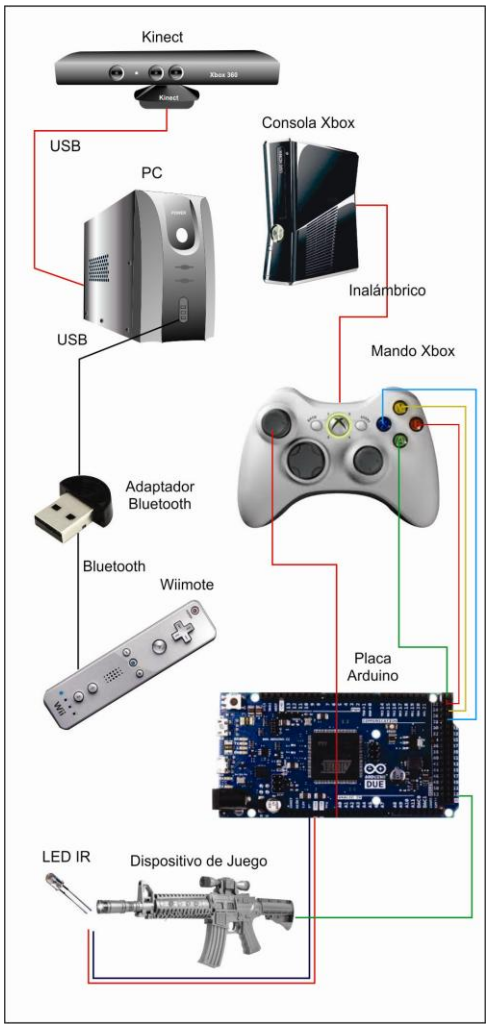


Figura 1. Arquitectura del prototipo

Para interactuar con el juego se utilizará un control inalámbrico, el cual será modificado para que pueda ser controlado por una placa *Arduino*, en este caso la *Arduino Mega 2560 Rev3*. La placa controladora interpretará los comandos de la PC y emulará cualquiera de los 8 botones del control o el movimiento de alguno de sus 3 joysticks (2 de ellos son analógicos y uno es digital).

La PC tendrá una interfaz USB con la cual se conectará al controlador *Kinect* y según las acciones detectadas se le ordenará a la placa *Arduino* para que accione los comandos correspondientes del control.

Debe tenerse en cuenta que el software que interpretará los movimientos del *Kinect* será desarrollado especialmente para este prototipo, así como también el firmware de la placa *Arduino* y el software de interfaz con la misma, instalado en la PC.

Además, se utilizará el control de la consola *Wii*, el cual posee una cámara infrarroja (IR), y servirá para detectar la posición de un LED IR ubicado en el cañón del dispositivo para juego que tendrá el usuario [6].

En la Figura 1, se puede observar la arquitectura que guiará el desarrollo.

3. Placa controladora

La placa controladora será la encargada de controlar el *joystick*, haciendo de interfaz entre la PC y dicho control. Luego, la PC interpretará los datos de los distintos sensores[7] y enviará los datos a la placa controladora para que la misma actúe sobre el mando de la *Xbox* en este caso.

Para simular la acción sobre los botones digitales del mando, se utilizó un transistor actuando al corte y a la saturación, logrando así un interruptor o *switch*, ya que cuando el transistor está en corte se puede comparar con un interruptor abierto y cuando está en saturación es equivalente a un interruptor cerrado.

La Figura 2 muestra la similitud existente entre un transistor y un interruptor.

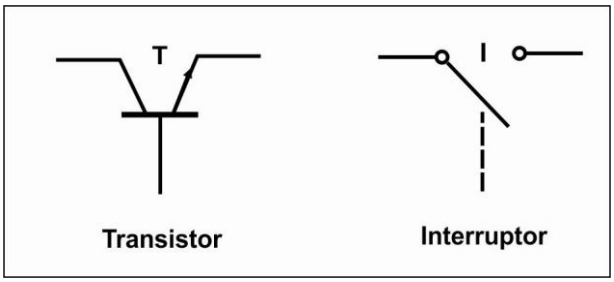


Figura 2. Similitud entre el transistor y un interruptor

4. Control de movimiento

Para este prototipo se hará uso de un controlador *Kinect*, que permite a los usuarios controlar e interactuar

[5] Por ejemplo: *Xbox*, *PlayStation*, *Wii*, etc.

[6] Se ha elegido por su facilidad de obtención en el mercado local un receptor *Microsoft Xbox 360* inalámbrico para PC.

[7] *Kinect*, *Wiimote*, etc.

con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, objetos e imágenes [7] [8].

Un problema que tiene el controlador, y que el prototipo intenta solucionar es que los juegos que utilizan el Kinect son especialmente desarrollados para dicho periférico. En este caso, lo que se pretende es que cualquier juego (o simulador) pueda hacer uso de las facilidades que el controlador ofrece.

Para que el juego o simulador sea independiente del controlador *Kinect*, se utilizará una PC, la cual interpretará los gestos que produzca el usuario y se lo transmitirá a la placa controladora, en este caso la interfaz *Arduino*, conectada al *joystick* de la *Xbox*.

El prototipo se desarrollará de manera evolutiva, incrementando las funcionalidades en sucesivas versiones. Para esta primera versión, se controlará el movimiento de una persona, detectando si está de pie, en cuclillas o con el cuerpo en tierra. En el futuro se extenderá a poder dar órdenes gestuales y por voz, reconocimiento facial para la identificación automática de los usuarios, etc.

5. Control del dispositivo para juego

El dispositivo para juego tendrá en su cañón un LED IR, que emitirá luz IR[8] en un foco colocado al extremo de éste, de forma que ***dependiendo de cómo ve el mando la posición de este punto podrá calcular dónde se ubica el dispositivo para juego.*** Este control, requerirá de una calibración previa.

La Figura 3 muestra los componentes del mando de la consola *Wii* [6].

6. Gafas

Existen métodos adicionales para incrementar el realismo, además de los aspectos meramente físicos. Uno de ellos es el desarrollo de la visión subjetiva, caracterizada por:

- Ilusión de profundidad (estereoscopía).
- Ilusión de orientación.

Para ello se utilizan unas gafas estereoscópicas y un sensor de posición y orientación (giróscopo).

La ilusión de profundidad se crea suministrando a cada ojo la información que, por su posición, debería ver.

La conciencia debe recibir datos que le permitan armar la profundidad necesaria. Si los datos son suficientes, se configura una percepción que completa lo que no hay entre las dos imágenes planas que llegan a los ojos, y se obtiene la ilusión de una tercera dimensión. Para lograr esto, se tienen dos mecanismos principales: polarización y ***side-by-side***.

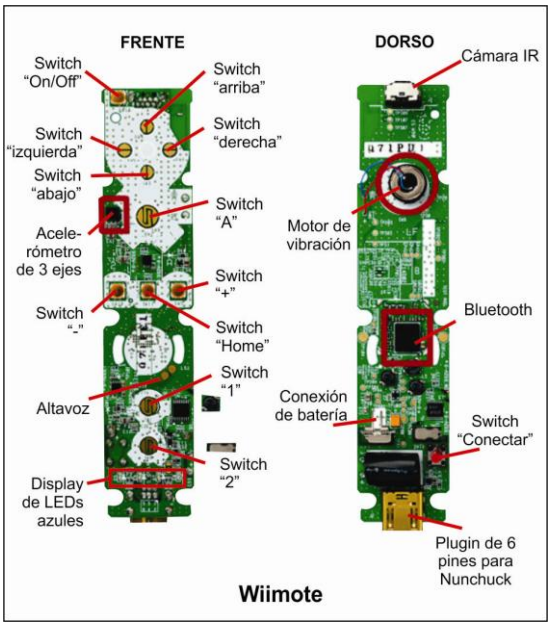


Figura 3. *Wiimote*

En el mecanismo de ***polarización*** se utilizan gafas estereoscópicas cuyo comportamiento se asemeja al de un sistema de monitores duales, donde se transmiten a las gafas ambas imágenes por ciclo, que son entrelazadas en una única imagen, tal como lo muestra la Figura 4.

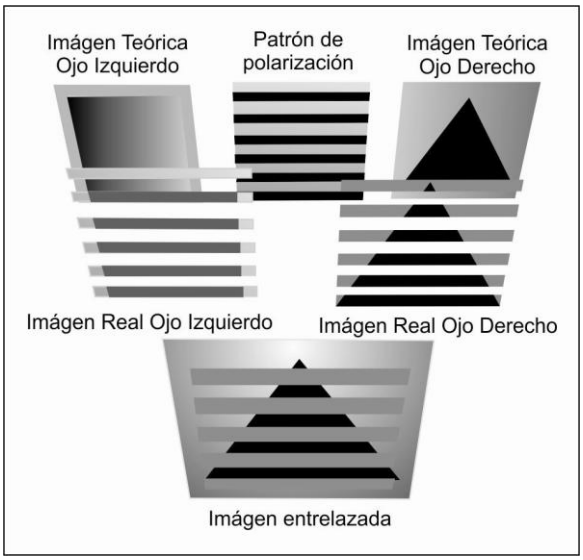


Figura 4. Imágenes entrelazadas

Luego, las gafas decodifican la imagen enviada y dibujan la imagen correspondiente para cada ojo.

Por ej., si se tienen 1080 pixeles en el sentido vertical de la imagen, cada línea de pixeles se verán con

[8] En una frecuencia de onda no visible para el ojo humano con la que trabaja el mando de la *Wii* para saber la posición del puntero.

diferentes lados de los lentes y si enumeramos los pixeles de arriba hacia abajo del 1 al 1080 veremos que, con el lado derecho de los lentes se ven solamente los pixeles 1, 3, 5, 7, 9, etc. y con el lado izquierdo se verán los pixeles 2, 4, 6, 8, etc. Las desventajas son que la resolución de la imagen disminuye en un 50% y que requiere que se dibuje dos veces la escena para cada instante, pero en contraparte, da la sensación de profundidad y respuesta a los movimientos del usuario y la representación del entorno no es muy costosa en tiempo de CPU con relación al cálculo dinámico. Además, al ser un método de 3D pasivo (en contraparte las gafas activas, usan batería y parpadean muy rápidamente casi imperceptible) no marean al usuario y como no utiliza pilas ni sincronizador son mucho más económicos.

Utilizando el método ***side-by-side***, la pantalla se divide para mostrar dos imágenes al mismo tiempo. Cuando el observador se concentra produce la fusión de las imágenes creando la ilusión de profundidad.

La Figura 5 muestra un esquema de este método.

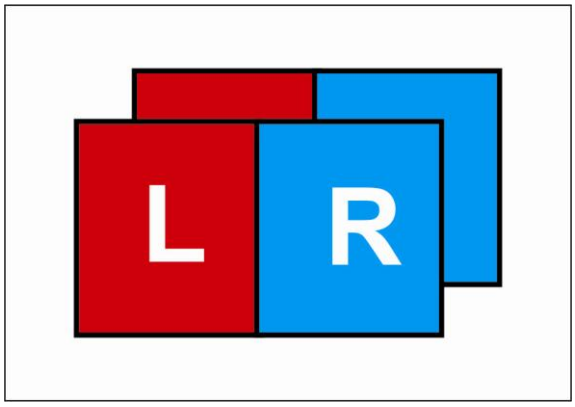


Figura 5. Imágenes *side-by-side*

Tomando el mismo ejemplo que el descripto en el método anterior, en este caso se utilizarán los 1080 píxeles para ambas imágenes, pero la imagen izquierda ocupará la primera mitad horizontal de la pantalla entera, mientras que la derecha ocupará la mitad restante.

El sensor de posición y orientación es una IMU[9] que consta de un giróscopo y un acelerómetro, ambos de 3 ejes, los cuales sensan el movimiento de la cabeza del usuario. En este caso, es fundamental tener en cuenta el tiempo de refresco de píxeles del panel, para poder reducir la latencia y el desenfoque de movimiento al girar la cabeza de una forma rápida.

Debe tenerse en cuenta que es fundamental cubrir casi todo el campo visual del usuario, para crear un fuerte sentido de la inmersión.

Las gafas tendrán dos interfaces, las cuales en principio serán cableadas. Una interfaz HDMI, la cual se utilizará para la entrada de los datos de vídeo y la otra será la salida de los datos de la IMU a la placa controladora *Arduino*, que interpretará los datos para ser transmitidos al juego o simulador. En el futuro el proyecto desarrollará comunicaciones *wireless* que reemplazarán a las cableadas.

7. *Feedback*

Para obtener más realismo, se incorporará un ***feedback*** desde el juego o simulador hacia el usuario. Para ello, en una primera instancia se hará uso de los motores de vibración que ya tiene el mando de la *Xbox* y los mismos serán adaptados para que el usuario lo lleve puesto, por ejemplo en formato brazalete o cinturón.

Además, se agregará sonido envolvente completamente en 3D, de forma de seguir adicionando factores que sumen a la inmersión del usuario a la simulación.

8. Resultados Obtenidos

Actualmente, ya se han confeccionado prototipos que han funcionado correctamente, respecto a los siguientes módulos

8.1. Gafas

Se construyó un prototipo inicial consistente en elementos de hardware con su correspondiente software embebido.

Respecto al hardware, se determinó la necesidad de utilizar una pantalla con buena resolución, alta densidad de píxeles y, además, lentes asféricas de mayor calidad.

El hardware utilizado tiene las siguientes características:

- Teléfono celular Smartphone con Android 4.4.2 y con un procesador Qualcomm Snapdragon S4 Pro MSM8960 de doble núcleo funcionando a 1.7 GHz, un procesador gráfico Adreno 320, 2Gb de memoria RAM y una pantalla de 4,7 pulgadas con resolución de 720x1028 y con una densidad de pixeles de 316 ppi[10]. Además posee un giróscopo que es utilizado para capturar el movimiento rotacional de la cabeza.
- Gafas de Realidad Virtual creadas para que puedan utilizarse con el efecto *side-by-side* 3D.

  Para ello se utilizó un armazón que sostuviera dos lentes (preferentemente asféricas) de 2.5 cm de diámetro y aumento de 10x, separadas por aproximadamente 1.5 cm. Las lentes actuales son genéricas, de plástico, sacados de unos binoculares genéricos, por lo que no se garantiza un aumento de 10x. Cabe destacar que escogió este método frente al

[9] Del inglés Unidad Inercial de Movimiento

[10] Del inglés Píxeles por Pulgada

de polarización, ya que éste requiere anteojos polarizados o con filtro y del doble de fps[11] que se desea, es decir, si el simulador es de 60 fps, la frecuencia de las pantallas debe ser del doble, en este caso 120. Además el método de polarización hubiera sido más difícil de implementar que el Side-by-side.

- Un joystick *DualShock* 4[12] para el movimiento de traslación de la cámara.

Las Figuras 6 y 7, muestran algunas imágenes del prototipo creado.

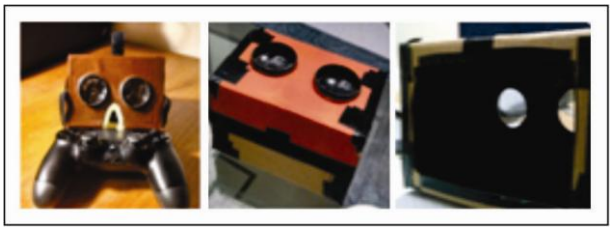


Figura 6. *Hardware* del prototipo gafas

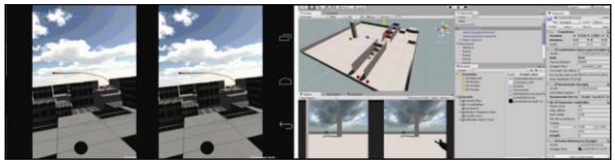


Figura 7. Software con Unity 3D del prototipo gafas.

Con respecto al *software*, al utilizar la técnica *side-by-side*, es necesario contar con dos imágenes que estén levemente desfasadas, esto se logra con técnicas especiales de cámara.

El software utilizado tiene las siguientes características:

- Es un desarrollo propio, realizado especialmente para demostrar el funcionamiento de las gafas, creado con el motor gráfico Unity3D 4.3.4[13] con scripts propios. El mismo posee dos grandes ventajas: es gratuito y además deja exportar los proyectos que realices para que los ejecutes desde casi cualquier dispositivo, ya sea PC, iOS o Android como en este caso.
- Entorno SDK de Android para poder crear la interfaz .APK que se instala en el Smartphone con el fin de acceder a los dispositivos del teléfono, como ser el acelerómetro.

El funcionamiento correcto de las gafas de realidad virtual se logra creando una distorsión en las imágenes al usuario, mezclando ambas en una sola, y como éstas muestran lo mismo pero de distinto ángulo, se genera una tercera imagen central con el efecto de tres dimensiones.

El software se divide en dos capas, la primera está destinada a generar los objetos que van a participar de la misma, lo que en *Unity* se conoce como "*Scene*" o escena; para este proyecto se ubicaron dos cámaras, cada una separada levemente de la otra y apuntando a un punto en común, para poder generar el efecto de 3D. La segunda parte son los *scripts* o programas que vamos a utilizar; en particular, se usaron para el giróscopo del *smartphone* y para moverse por la escena utilizando un control de la *Playstation 3* (todos ellos se desarrollaron en C#).

La Figura 8 muestra una sección del *script* utilizado para controlar el giróscopo del dispositivo.



```
ControladorGiroscopio.cs
No selection
using UnityEngine;

public class ControladorGiroscopio : MonoBehaviour {

    //Atributos Publicos
    public GameObject bala;
    public int fuerzaDisparo = 2000;
    public Texture2D imagenMira;
    public int correctorDeAltura = 5;
    public GameObject deDondeSaleLaBala;
    /*public float valorDelSlider = 0.0f;
    public Camera camaraIzquierda;*/
    //public AudioClip disparo;
    //public GameObject casquillo;
    //public GameObject miraLaser;

    //Atributos Privados
    public bool estaHabilitadoElGiroscopio = true; //deberia ser private
    private const float filtro = 0.201f;  //deberia ser 0.2f

    private readonly Quaternion paradoNormal =  Quaternion.Euler(90, 0, 0);
    private readonly Quaternion acostadoParaLaDerecha =  Quaternion.Euler(0, 0, 90);
    private readonly Quaternion acostadoParaLaIzquierda =  Quaternion.Euler(0, 0, -90);
    private readonly Quaternion paradoAlrevez =  Quaternion.Euler(0, 0, 180);

    private Quaternion camera =  Quaternion.identity;
    private Quaternion calibracion =  Quaternion.identity;
    private Quaternion orientacionParado =  Quaternion.Euler(90, 0, 0);
    private Quaternion orientacionDeLaRotacionParado =  Quaternion.identity;

    private Quaternion referenciaDeRotacion = Quaternion.identity;
    private bool debug = true;
```

Figura 8. Sección del *script* utilizado para controlar el giróscopo del *smartphone*

Además, se realizaron pruebas sobre el sensor de movimiento de 9 ejes (giróscopo, acelerómetro y brújula) MPU-9150, el cual es ideal para diseños de tecnología *wearable* debido a su poco consumo, poco costo y alto rendimiento. Dicho sensor fue controlado utilizando una placa *Arduino Uno* [9]. A continuación, en la Figura 9, se muestra un esquemático de la conexión utilizada.

Las mejoras que se tendrán en cuenta para un próximo prototipo serán las siguientes:

- Calidad de los lentes:
  - aumento 10x asegurado.
  - lentes asféricas.
  - lentes de vidrio o algún material que no se raye fácilmente.
  - lentes de mayor diámetro

[11] Del inglés Cuadros por Segundos
[12] Podría utilizarse otro modelo equivalente.
[13] *Unity* es un motor de renderizado gratuito, integrado con un conjunto de herramientas para crear contenido 3D interactivo. http://unity3d.com

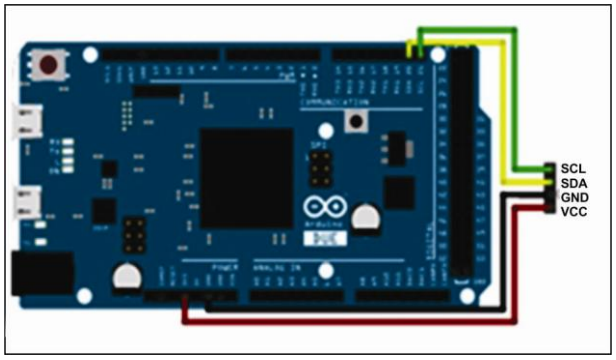


Figura 9. Modelo esquemático de la conexión de la placa Arduino con el sensor MPU-9150

- Calidad de la carcasa:
  - modelo de la carcasa impreso en 3D.
  - mejor diseño para que se adapte al rostro, balancee el peso y no deje entrar luz externa.
- Calidad de la pantalla:
  - Se pueden utilizar smartphone con pantallas que tengan mayor resolución o mayor densidad de píxeles.
  - Se puede utilizar una pantalla conectada a una computadora o consola por HDMI. En este caso, se debería incluir un giróscopo para la captura de la rotación.

8.2. Placa controladora

Para este prototipo en particular se utilizó la placa *Arduino Mega 2560 Rev. 3*. Esta es una placa electrónica basada en la *Atmega2560. Cuenta con:* 54 pines de entradas/salidas digitales (de los cuales 14 se pueden utilizar como salidas PWM[14] y 16 entradas analógicas, 4 *UARTs*[15], un oscilador de cristal de 16 MHz y un puerto de conexión USB, lo que lo hace ideal para el proyecto planteado

Para la simulación de los botones digitales se utilizaron transistores BC337.

8.3. Control de movimiento

Se realizó un prototipo utilizando el sensor *Kinect* utilizando el *SDK*[16]. Mediante el mismo actualmente se logra reconocer la figura humana y si la misma se encuentra de pie o agachado [7] [10] [11]. En el futuro se le agregará la capacidad de detectar si el mismo se encuentra en posición cuerpo a tierra y además tendrá la capacidad de detectar gestos, como por ej.: el lanzamiento de granadas.

8.4. Control del dispositivo para juego

Para esta sección del prototipo, se replicaron dos proyectos publicados por Johnny Chung Lee[17] donde se conecta el mando de la Wii a la PC utilizando la conexión *bluetooth* y aprovechando la cámara IR que posee el mando se reconoce la emisión de un LED IR.

Para la conexión *bluetooth*, se utilizó como driver la aplicación *Bluesoleil*[18].

El primero de los proyectos que se realizó es el de *Head-Tracking* [12] [13], mediante el cual se puede simular una imagen 3D moviendo la cámara de la imagen (se entiende que la cámara es el objeto de software que está mostrando lo que el usuario percibe) solidario al movimiento de la cabeza del usuario. Cuando una persona mira la imagen en una pantalla y se mueve hacia un costado, la imagen que percibe es exactamente la misma pero en otro ángulo, pero esta imagen podría modificarse, si se cambia el ángulo de la cámara, simulando así una imagen 3D.

En la Figura 10, la (a) muestra como el usuario percibe la misma imagen a pesar de su movimiento, pero la (b) muestra la nueva imagen que debería percibir al sensar su desplazamiento.

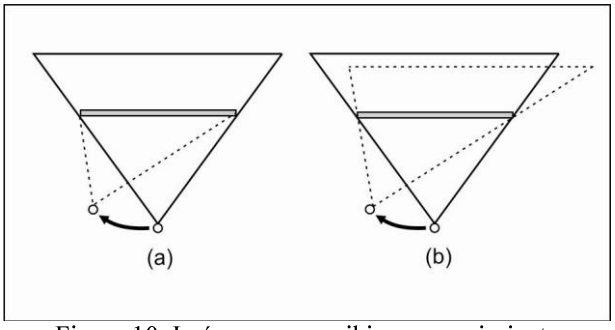


Figura 10. Imágenes a percibir por movimiento tangencial

Lo mismo ocurre cuando el usuario se acerca o se aleja de la pantalla, tal como lo muestra la Figura 11.

Luego, se entiende por *head-tracking* al hecho de cambiar la perspectiva de lo que se muestra en pantalla dependiendo de donde tengamos la cabeza. Es decir, en un entorno 3D se obtiene una perspectiva mucho más real, con vistas distintas dependiendo desde donde se esté mirando a la pantalla.

[14] del inglés Modulación por Ancho de Pulso.
[15] *Universal Asynchronous Receiver-Transmitter* - puertos de hardware de serie
[16] El mismo ha sido publicado por *Microsoft* en su sitio de la Red http://www.microsoft.com/en-us/download/details.aspx?id=43661

[17] http://johnnylee.net/
[18] http://www.bluesoleil.com/bssoftware/BSoftware.aspx.

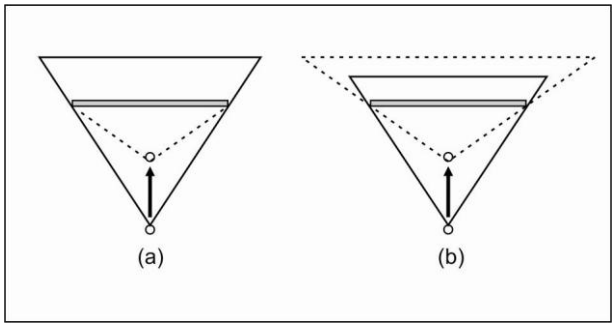


Figura 11. Imágenes a percibir por movimiento radial**.**

Para ello, se dota al usuario de un par de lentes con 2 (dos) LEDs emisores de IR, cada uno ubicado al costado de la cabeza. Cabe recordar que se utiliza al mando de la Wii como cámara IR.

La Figura 12 muestra un esquema de lo detallado anteriormente.

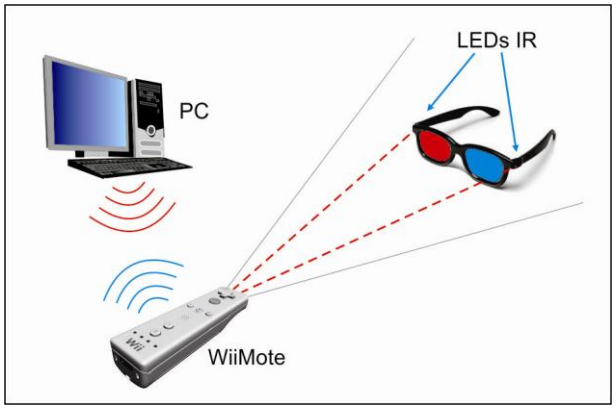


Figura 12. Esquema *Head-Tracking*

Luego midiendo la distancia entre ambas fuentes de luz se puede determinar si la persona se aleja o se acerca de la cámara. Y además se puede sensar si la persona se mueve hacia un lado o el otro, midiendo con respecto al centro de la cámara.

La Figura 13 muestra los cálculos, tanto para la posición horizontal como para la vertical.

De esta forma, se puede obtener la localización y orientación de la cabeza y representar en pantalla, por lo tanto, la vista que se desee. El resultado es una forma de convertir una simple pantalla en un portal a un entorno virtual. La escena en la pantalla reacciona a los movimientos de la cabeza como si se tratara de una ventana real, creando una ilusión bastante realista de profundidad y espacio.

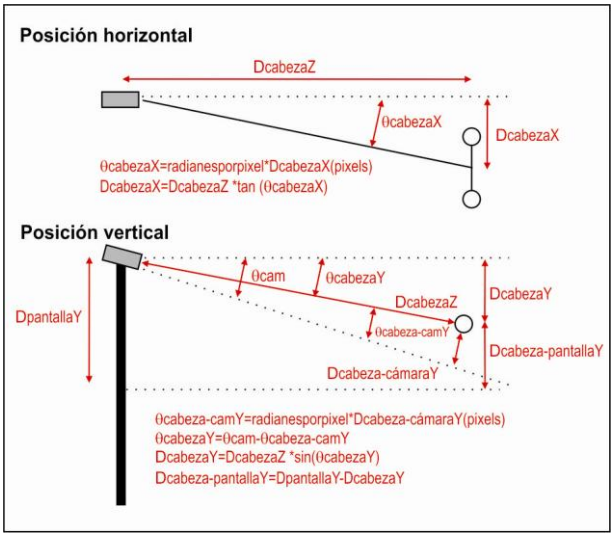


Figura 13. Cálculo para determinar la posición de la cabeza del usuario basado en la distancia y orientación con respecto al *Wiimote*

El otro proyecto replicado es el de pizarra, mediante el cual se simula una pizarra digital. Dado que el *Wiimote* puede rastrear fuentes de luz infrarroja, se pueden hacer punteros con LEDs infrarrojos y usarlos para manejar una interfaz. Apuntando el *Wiimote* a una pantalla proyectada o a un monitor LCD se pueden crear pizarras interactivas por muy poco dinero. También funciona con retroproyectores. La Figura 14 muestra el esquema utilizado.

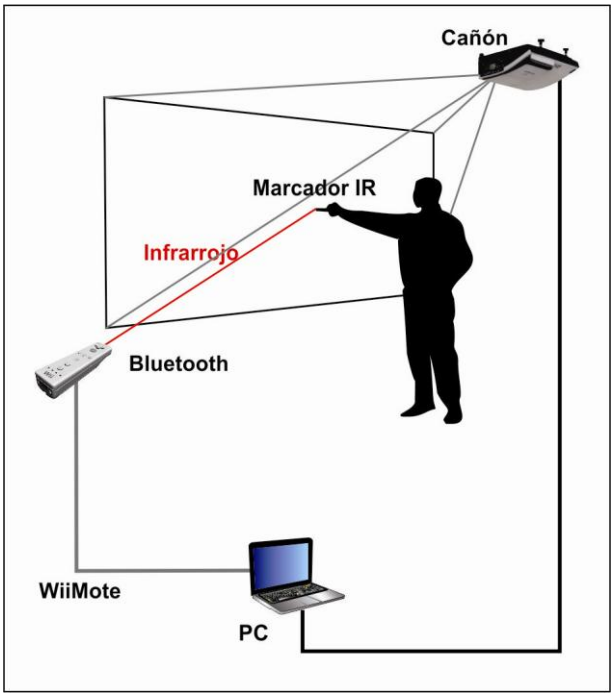


Figura 14. Esquema *Whiteboard*

El marcador IR utilizado es como el que se detalla en la Figura 15, el cual consta de un LED IR, una batería y un pulsador, para indicar cuándo se está escribiendo en la pantalla.

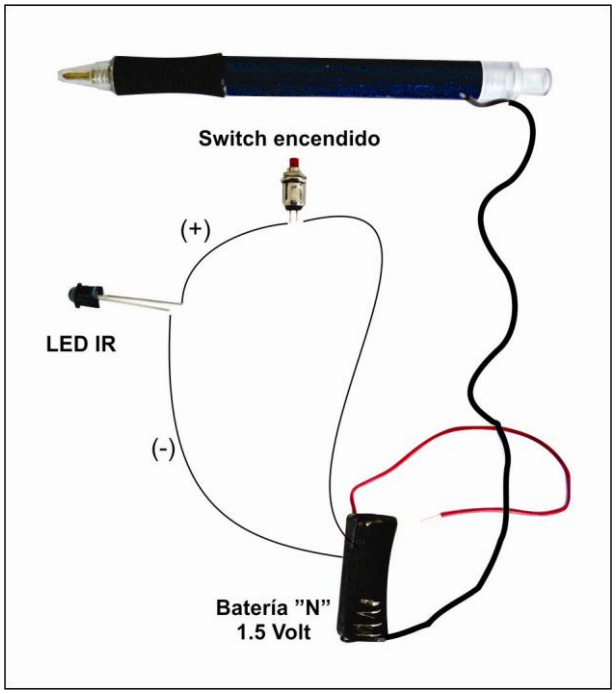


Figura 15. Marcador infrarrojo

9. Futuras versiones

El equipo de trabajo está planificando las futuras versiones del producto. En las mismas se tendrán en cuenta las siguientes nuevas funcionalidades:

- Control por gestos [14] y por voz.
- Sistema para que el usuario pueda moverse en el espacio con su propio esfuerzo físico, caminando, e incluso corriendo. La idea es que el usuario pueda moverse en todas las direcciones libremente sobre una superficie, y que el sistema reconozca sus movimientos y los interprete en el juego.
- Generación de imágenes 3D partiendo de una 2D, utilizando procesamiento de imágenes.
- Seguimiento del iris para visión periférica.
  Para ello se le incorporará a las gafas la posibilidad de poder sensar el movimiento del iris y de esta forma poder mostrar la nueva visión del usuario, la cual se logra sin el movimiento de la cabeza. Esta funcionalidad, se piensa lograr con la incorporación de una cámara IR - Infra-Roja y un procesamiento de imágenes para poder sensar el movimiento de los ojos. Cabe destacar que se deberá tener una pantalla más grande para poder cubrir todo el campo de visión del usuario. Para ello, se investigarán las pantallas OLED[19], ya que algunas de ellas tienen la capacidad de tener una estructura flexible, lo que ya ha dado lugar a desarrollar pantallas plegables o enrollables.
- *Kinect* y *Wii* para reconocimiento en 360º. Actualmente el sensado de ambos dispositivos solo cubre el ángulo propio de cada uno, pero se desarrollará una plataforma o se agregarán más sensores para poder cubrir los 360º completos.
- Controles inalámbricos en vez de cableado. Tanto para el dispositivo para juego, como para la *IMU* y el video de las gafas.
- Reacción del simulador hacia el usuario. En una primera instancia se utilizarán los motores de vibración que ya vienen incorporado en el control de la *Xbox*.
- Reconocimiento facial para la identificación automática de los usuarios.
- Se le incorporará a las gafas una cámara externa, para que las mismas puedan ser utilizadas con tecnología de Realidad Aumentada, definiendo una visión, directa o indirecta, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

10. Referencias

[1] D. A. Camarasa y O. M. Bianchi, *Desarrollo de Software de Simulación Inmersiva para Fracciones Heterogéneas,* Presentado en WICC 2012. http://sedici.unlp.edu.ar/bitstream/handle/10915/19214/Documento_completo.pdf?sequence=1, 2012.

[2] J. N. Templeman, L. E. Sibert and R. C. Page, *Immersive Simulation to Train Urban Infantry Combat,* Washington, DC: Naval Research Laboratory. http://www.researchgate.net/publication/228727384_Immersive_simulation_to_train_urban_infantry_combat/file/32bfe50fd612966169.pdf.

[3] W. Wilkerson, D. Avstreih, L. Gruppen, K. P. Beier and J. Woolliscroft, *Using immersive simulation for training first responders for mass casualty incidents,* Academic emergency medicine official journal of the Society for Academic. http://www.academypublisher.com/ijrte/vol01/no01/ijrte0101529531.pdf.

[4] I. Hernández Gutiérrez, *Wii-immersion: Aprovechando las capacidades del Wiimote para la mejora de la sensación de inmersión,* Universitat Pompeu Fabra. https://repositori.upf.edu/bitstream/handle/10230/11314/PFC%20Wii-immersion_Hern%C3%A1ndez.pdf?sequence=1,

[19] Del inglés Diodo Orgánico de Emisión de Luz

2010.

[5] P. Toharia, *Realidad Virtual e Interacción. Dispositivos Entrada/Salida,* Universidad Rey Juan Carlos. http://dac.escet.urjc.es/rvmaster/rvmaster/asignaturas/rvi/TemaIII.Dispositivos_de_ES__1314.pdf.

[6] J. C. Lee, *Hacking the Nintendo Wii remote,* Carnegie Mellon. https://www.cs.cmu.edu/~15-821/CDROM/PAPERS/lee2008.pdf, 2008.

[7] J. Han, L. Shao, D. Xu and J. Shotton, *Enhanced Computer Vision with Microsoft Kinect Sensor: A Review,* IEEE Transactions on Cybernetics, Vol. 43, No. 5, October 2013.

[8] Y. Chow, *A Cost-Effective 3D Interaction Approach for Immersive Virtual Reality,* Wollong, Australia: School of Computer Science and Software Engineering, University of Wollongong. http://www.academypublisher.com/ijrte/vol01/no01/ijrte0101529531.pdf, 2009.

[9] O. González, *Sistema de medición inercial con Arduino y Processing.,* http://blog.bricogeek.com/noticias/arduino/sistema-de-medicion-inercial-con-arduino-y-processing/, 2009.

[10] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, *Real-Time Human Pose Recognition in Parts from Single Depth Images.,* Microsoft Research Cambridge & Xbox Incubation. http://research.microsoft.com/pubs/145347/bodypartrecognition.pdf, 2011.

[11] K. Abhishek, *Skeletal Tracking using Microsoft Kinect,* Department of Computer Science and Engineering, IIT Kanpur. http://www.cs.berkeley.edu/~akar/cs397/Skeletal%20Tracking%20Using%20Microsoft%20Kinect.pdf

[12] K. Hejnkhejn and J. P. Rosenkvist, *Headtracking using a Wiimote.,* http://image.diku.dk/projects/media/rosenkvist.hejn.08.pdf, 2008.

[13] M. J. Ubilla Carvajal y D. Mery, *Head Tracking en el espectro infrarrojo utilizando el Nintendo Wii Remote.,* http://web.ing.puc.cl/~dmery/Prints/Supervised-Theses/2009-MScMUbilla.pdf, 2009.

[14] T. Schlomer, B. Poppinga, N. Henze and S. Boll, *Gesture Recognition with a Wii Controller,* University of Oldenburg & OFFIS Institute for Information Technology. http://www.wiigee.org/download_files/gesture_recognition_with_a_wii_controller-schloemer_poppinga_henze_boll.pdf.

Una Arquitectura de un Sistema de Búsqueda de Respuestas

Alejandra Carolina Cardoso, Agustina Bini y M. Alicia Pérez Abelleira
Facultad de Ingeniería e IESIING
Universidad Católica de Salta
Campo Castañares, Salta A4400
{acardoso,aperez}@ucasal.net, agubini@hotmail.com

Abstract

Dada la gran cantidad de información no estructurada disponible en todo tipo de organizaciones, la minería de textos va creciendo en importancia, y en particular el tipo de sistemas que pretenden responder a preguntas expresadas en lenguaje natural, los sistemas de búsqueda de respuestas. Este trabajo describe la arquitectura de un sistema con esas características que responde a preguntas de usuarios cuyas respuestas están en un corpus de más de ocho mil documento que contienen resoluciones rectorales.

1. Introducción

La búsqueda de respuestas (BR) tiene como objetivo dar respuestas en lenguaje natural a preguntas también en lenguaje natural. Extendiendo esta definición, puede considerarse como un proceso interactivo entre un usuario y el sistema en el que el segundo pretende comprender la necesidad de información del primero expresada como una consulta en lenguaje natural, recuperar datos o información relevantes para responderla, priorizar entre éstos y finalmente presentar las respuestas relevantes al usuario [1].

Aunque el problema de BR ha sido estudiado desde hace más de diez años, continúa siendo un desafío que incorpora varias tareas del ámbito de la minería de textos, del procesamiento del lenguaje natural y otras técnicas para poder (a) comprender adecuadamente las necesidades de información de la pregunta, (b) obtener una lista de respuestas candidatas a partir de los documentos, y (c) filtrarlas en base a evidencia que justifique que cada una de esas respuestas es la correcta.

En este trabajo presentamos una primera aproximación a un sistema de BR que puede contestar preguntas factoides sobre un corpus de más de 8000 documentos que contienen 9 años de resoluciones rectorales de la Universidad Católica de Salta en distintos formatos (Word, texto plano, PDF). Este trabajo es continuación de una línea de investigación en minería de textos que ha desarrollado una arquitectura para búsqueda semántica [2] [3]. Tras una introducción a los sistemas de búsqueda de respuestas en la Sección 2, la sección 3 describe dicha arquitectura de búsqueda semántica.

El sistema de BR propuesto está formado de los siguientes componentes: análisis de la pregunta, que incluye su categorización y la construcción de la consulta; recuperación de documentos en base a la consulta; y extracción de las respuestas candidatas y presentación al usuario. Estos componentes se describen a partir de la Sección 4. El artículo concluye evaluando el desarrollo actual de este sistema y enunciando algunas conclusiones.

2. Sistemas de búsqueda de respuestas

Ya en los años 70 Wendy Lehnert [4] propuso la primera aproximación a un sistema de BR y sus características ideales, según las cuales el sistema debía entender la pregunta del usuario (comprensión del lenguaje natural), buscar la respuesta en una base documental (búsqueda de conocimiento), y componer la respuesta y mostrarla al usuario (generación de lenguaje natural). Esta estructura de tres componentes se ha preservado en la mayoría de los sistemas actuales.

Los sistemas de BR comenzaron a desarrollarse en el ámbito de la inteligencia artificial y después en el de los sistemas de recuperación de la información. Así se consiguieron avances importantes en la fase de recuperación de documentos o pasajes susceptibles de contener la respuesta correcta, pero no tanto en la comprensión de las preguntas y extracción de las respuestas. Los desarrollos posteriores incluyeron diversos recursos lingüísticos procedentes del ámbito del procesamiento del lenguaje natural, para ayudar a la comprensión de los textos y de las preguntas. La cantidad y complejidad de los recursos lingüísticos utilizados ha sido variable, incluyendo etiquetadores POS (part-of-speech), analizadores sintácticos, extractores de entidades

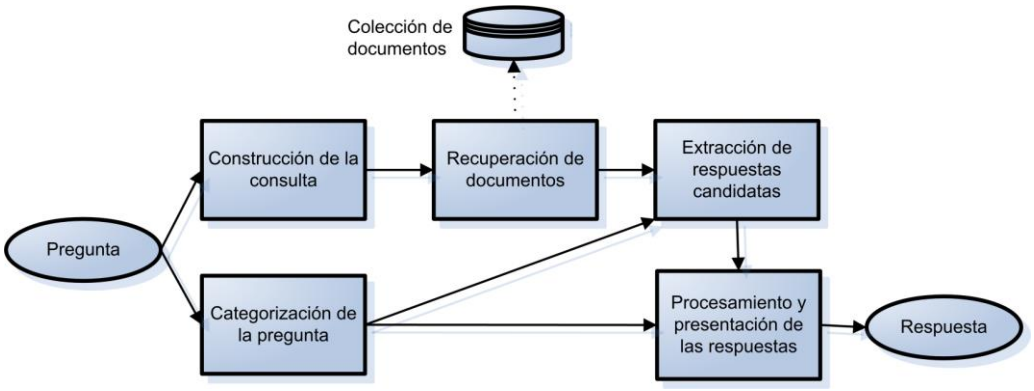


Figura 1. Arquitectura básica para la búsqueda de respuestas [3].

con nombre (NER), diccionarios, recursos externos como bases de datos léxico semánticas y ontologías, y hasta técnicas de análisis semántico y contextual. Algunas técnicas, especialmente las correspondientes a los primeros niveles del análisis de lenguaje natural, es decir los niveles léxico y sintáctico, han sido efectivas. Sin embargo, construir recursos más sofisticados es una tarea compleja y no necesariamente llega a mejores resultados, con lo cual a menudo no se justifica la mejora de rendimiento con el esfuerzo empleado en su aplicación y con los tiempos de ejecución empleados en el proceso completo [1] [5] [6].

Un sistema típico de búsqueda de respuestas supone una serie de procesos que comienzan tomando la pregunta del usuario como entrada y terminan respondiendo con una respuesta o una lista de respuestas priorizadas, con indicaciones de la fuente de la información. Pasca [7] caracterizó el paradigma de-facto para la búsqueda de respuestas como: recuperar documentos potencialmente relevantes, extraer las respuestas potenciales, devolver las respuestas con mayor ranking. Para procesar las preguntas suelen realizarse dos pasos: identificar qué tipo de información se busca (categorización de la pregunta), y elaborar la consulta que ubique en la colección de documentos fragmentos de texto con posibilidades de contener la respuesta. La Figura 1 muestra una arquitectura genérica para un sistema de BR con los componentes mencionados.

3. Antecedentes

La línea de trabajo en la que se encuadra este proyecto ha estado centrada en el uso de técnicas de minería de texto para la explotación de información no estructurada. Una aplicación de gestión de la información no estructurada (UIM por sus siglas en inglés) típicamente es un sistema que analiza grandes volúmenes de información no estructurada con el fin de descubrir, organizar y entregar conocimiento relevante al usuario final. La información no estructurada puede ser mensajes de correo electrónico, páginas web o documentos generados con una variedad de procesadores de texto, como en el caso de las resoluciones rectorales de nuestra universidad. Estas aplicaciones utilizan para el análisis una variedad de tecnologías en las áreas del procesamiento del lenguaje natural, recuperación de la información, aprendizaje automático, ontologías y hasta razonamiento automático.

El resultado del análisis generalmente es información estructurada que se hace accesible al usuario mediante aplicaciones adecuadas. Un ejemplo puede ser la generación de un índice de búsqueda y la utilización de un buscador que facilita el acceso a documentos de texto por tema, ordenados según su relevancia a los términos o conceptos de la consulta del usuario [2]. Los sistemas de búsqueda de respuestas son una instancia de este tipo de aplicaciones.

Conceptualmente las aplicaciones de gestión de información no estructurada suelen organizarse en dos fases. En la fase de análisis se recogen y analizan colecciones de documentos y los resultados se almacenan en algún lenguaje o depósito intermedio. La fase de entrega hace accesible al usuario el resultado del análisis, y posiblemente el documento original completo mediante una interfaz apropiada. La Figura 2 muestra la aplicación de este esquema a nuestro dominio [2], en el que partimos de más de 8000 resoluciones rectorales en archivos de texto de distinto tipo: Word, PDF, texto plano. Previo al análisis, se procede a la extracción del texto de cada archivo utilizando herramientas de software libre (*poi.apache.org* y *tm-extractors*). El texto se normaliza eliminando acentos para facilitar los procesos de búsqueda y equiparación de cadenas. También se divide en partes la resolución extrayendo el encabezado (texto que contiene el número y la fecha de la resolución) y el cuerpo con la mayor parte de la información, y descartando en lo posible el texto “de forma”.

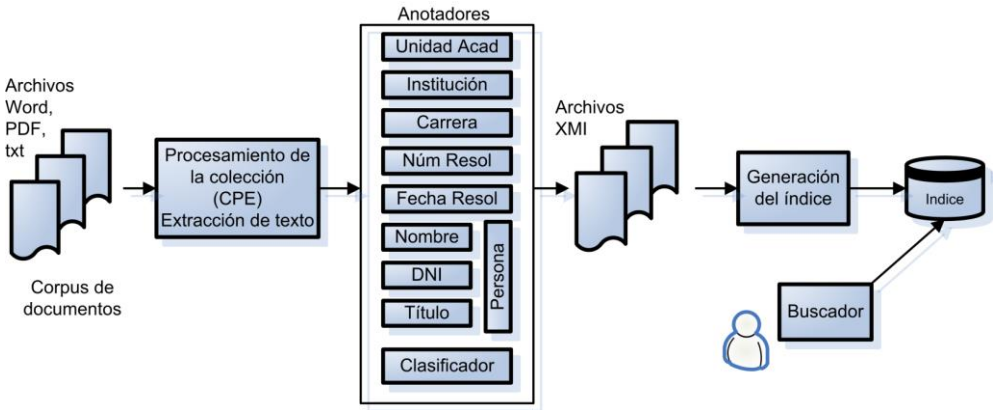


Figura 2. Arquitectura del sistema de gestión de información no estructurada.

La fase de análisis incluye tokenización y detección de entidades en documentos individuales tales como personas, fechas, organizaciones, unidades académicas y metadatos de la resolución (fecha y número). Además con la ayuda de un clasificador aprendido automáticamente del corpus de resoluciones se anota cada documento con una categoría [2]. Existen 21 categorías que fueron obtenidas del personal especializado en la elaboración de resoluciones. Algunos ejemplos son: designación de planta docente, convenio de pasantías, convenio de colaboración, o llamado a concurso docente.

El resultado de la fase de análisis es un conjunto de archivos en formato XMI. Estos archivos contienen, además de las partes relevantes del texto original, metadatos en forma de anotaciones correspondientes a las entidades existentes y a la categoría de documentos. Estos archivos serán procesados para construir el índice de un motor de búsqueda que contiene los tokens (en nuestro caso, las palabras que aparecen en el texto) y las entidades y categorías extraídas automáticamente.

En la fase de entrega existe una interfaz para hacer búsquedas en el índice. El usuario puede buscar documentos que contengan combinaciones booleanas de entidades, categorías y tokens mediante un motor de búsqueda semántica.

Las dos fases están desarrolladas sobre UIMA (Unstructured Information Management Architecture), una arquitectura basada en componentes para construir sistemas de procesamiento de información no estructurada [8]. En UIMA, el componente que contiene la lógica del análisis se llama anotador. Cada anotador realiza una tarea específica de extracción de información de un documento y genera como resultado anotaciones, que son añadidas a una estructura de datos denominada CAS (common analysis structure). A su vez, esas anotaciones pueden ser utilizadas por otros anotadores. Los anotadores pueden ser agrupados en anotadores agregados. La mayoría de nuestros anotadores realizan reconocimiento de entidades con nombre (NER), a saber:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmi:XMI … xmi:version="2.0">
…
  <cas:Sofa xmi:id="1" sofaNum="2" sofaID="encabezado" mimeType="text" sofaString="ResoluciOn N 470/08 En el Campo Castanares, sito en la Ciudad de Salta, Capital de la Provincia del mismo nombre, Republica Argentina, sede de la Universidad Catolica de Salta, a los veintisiete dias el mes de mayo del ano dos mil ocho:" />
  <cas:Sofa xmi:id="13" sofaNum="1" sofaID="_InitialView" mimeType="text" sofaString="la presentacion efectuada por las autoridades de la Escuela Universitaria de Educacion Fisica, dependiente de la Facultad de Artes y Ciencias en virtud de la cual se propone las modificaciones de designaciones docentes Planta Transitoria, para la carrera Licenciatura en Educacion Fisica…" />
…
  <examples:SourceDocumentInformation xmi:id="25" sofa="13" begin="0" end="0" uri="file:/D:/UIMA/docs/RES.%20%20Nº0470-08.txt" offsetInSource="0" documentSize="2280" />
  <ucs:UA xmi:id="48" sofa="13" begin="177" end="208" confidence="30.0" componentId="de.julielab.jules.lingpipegazetteer.GazetteerAnnotator" specificType="UnidadAcademica" />
  <ucs:Carrera xmi:id="72" sofa="13" begin="398" end="430" confidence="0.0" componentId="de.julielab.jules.lingpipegazetteer.GazetteerAnnotator" specificType="Carrera" />
  <ucs:NumeroResol xmi:id="33" sofa="1" begin="0" end="19" nroResol="RESOLUCION N 470/08" numero="470" anio="2008" />
  <ucs:FechaResol xmi:id="40" sofa="1" begin="196" end="248" anio="2008" mes="MAYO" dia="27" fechaResolCompleta="VEINTISIETE DE MAYO DE DOS MIL OCHO" />
  <ucs:Clase xmi:id="28" sofa="1" begin="0" end="0" valor=" DesigDocPlanta" />
…
</xmi:XMI>
```

Figura 3. Ejemplo de texto anotado.

personas, unidades académicas, carreras, instituciones; además hay otros que extraen fechas, número y año de las resoluciones. Para detectar entidades correspondientes a personas se agregan otras (nombres propios, DNIs y títulos) obtenidas por los anotadores correspondientes. Un último anotador asigna la categoría de documento utilizando el modelo aprendido automáticamente. Inicialmente los anotadores para NER fueron codificados a mano; luego fueron reemplazados o complementados con modelos aprendidos automáticamente usando campos aleatorios condicionales (CRFs) [3].

El trabajo descrito en este artículo construye sobre esta arquitectura, añadiendo, en la fase de entrega, un sistema de búsqueda de respuestas mediante el cual el corpus anotado de documentos sirve para contestar preguntas en lenguaje natural.

3.1. Formato de la información estructurada

Un importante principio de arquitectura en UIMA es que todos los analizadores operan sobre una estructura de datos estándar, el CAS, que incluye el texto y las anotaciones. El CAS es el principal repositorio de información estructurada y utiliza como lenguaje el estándar XMI (*XML Metadata Interchange*) [9]. XMI proporciona un formato en XML para el intercambio de modelos UML, lo que además hace posible la interoperabilidad con otras aplicaciones. En XMI el texto del documento se incluye (opcionalmente) en uno o más SofAs (*Subject of Analysis*) del CAS como atributo *sofaString*. Las anotaciones tienen prefijos particulares en el espacio de nombres de XMI, como por ejemplo *<ucs:Carrera>*. El espacio de nombres queda definido al declarar un sistema de tipos como parte del dominio en UIMA. La Figura 3 muestra un ejemplo del resultado del proceso de análisis. Dos *SofAs* recogen el encabezado, del que sólo se extrae la fecha y el número de resolución, y el cuerpo de la misma. La primera anotación del tipo *SourceDocument Information* guarda metadatos sobre el archivo, para poder recuperarlo posteriormente, por ejemplo, para mostrarlo al usuario como resultado de una búsqueda. A continuación aparecen varias anotaciones, algunas de ellas con varios campos. La anotación *Clase* contiene la categoría *DesigPlanta* asignada a esta resolución por el modelo aprendido.

4. Categorización de las preguntas

Para una computadora, el método más sencillo de interpretar una pregunta hecha en lenguaje natural sería eliminar las denominadas palabras vacías o *stopwords*, incluyendo quién, cuál o dónde, y convertir el resto de la pregunta en una consulta booleana. Sin embargo, al hacer esto se está desperdiciando información de gran utilidad para reducir el alcance de la pregunta. Por ejemplo, eliminando la palabra "cuándo" de la frase "¿Cuándo murió Güemes?" puede llevar a recuperar respuestas sobre cómo y dónde, en lugar de recuperar solo respuestas con la fecha solicitada. Por ello es que en los sistemas de BR se definen categorías de preguntas. La categorización de la pregunta tiene como objetivo asociar a la misma una categoría que indique el tipo de información que se busca, por ejemplo, el nombre de una persona que cumpla una serie de características, una fecha, una definición, etc. En general las entidades con nombre identificadas en la colección de documentos son candidatos típicos como objetivo de una pregunta. Por tanto, estas categorías asociadas a las preguntas proporcionan restricciones semánticas a la hora de extraer las respuestas candidatas.

Durante el análisis de la pregunta se ubican identificadores del tipo de pregunta, que después pueden compararse con patrones típicos para cada una de las categorías definidas. De hecho, en la literatura queda claro que determinar el dominio de la pregunta y caracterizar el tipo de la respuesta buscada son pasos esenciales en los sistemas de búsqueda de respuestas. No obstante existen una variedad de enfoques que varían en el número de categorías determinadas, la estructura (plana o jerárquica) de la clasificación y la elección de las categorías propiamente dichas [10] [5].

Como se indicó en la Seccion 1, el sistema descrito en este trabajo tiene como objetivo contestar preguntas factoides sobre un corpus de más de 8000 documentos que contienen resoluciones rectorales de la Universidad Católica de Salta. Las preguntas factoides requieren normalmente un hecho sencillo como respuesta, por ejemplo una fecha, el nombre de alguien o algo, o una cantidad. En nuestro caso, para establecer las posibles categorías de preguntas nos guiamos por el conjunto de entidades con nombre y otros elementos que los anotadores son capaces de detectar en el corpus, y que en general son el foco de las respuestas esperadas. Es en base al tipo de anotaciones que se estableció la clasificación. Para cada una de las anotaciones se determinó el patrón de comienzo de la pregunta, que en general incluye un pronombre interrogativo, ya que por el momento se trata en todos los casos de preguntas factoides.

Una vez establecida la clasificación, para la categorización de una pregunta dada en la literatura suele utilizarse uno de estos dos enfoques: reglas construidas manualmente, por ejemplo, "si la pregunta comienza con Quién, el tipo de la pregunta es PERSONA"; y clasificadores aprendidos automáticamente que predicen la *P(TipoDePregunta | Pregunta)* donde la pregunta se representa como un conjunto de atributos. Estos clasificadores son aprendidos en base a conjuntos de preguntas categorizadas manualmente. En nuestro caso, hicimos un análisis exhaustivo de ejemplos de preguntas

significativas en nuestro corpus, y dada la limitada variedad de las mismas el primer enfoque es suficiente, evitando además la construcción del conjunto de preguntas necesario para entrenar un clasificador.

En base a lo expuesto, la Tabla 1 muestra las categorías de preguntas relevantes en nuestro corpus particular, organizadas en base al tipo de respuesta esperada.

Tabla 1. Clasificación de las preguntas según el tipo de respuesta esperada.

Respuesta esperada	Pregunta
Persona	¿Quién … ¿A quién … ¿Quiénes …
Institución	¿ A qué … ¿Qué institución … ¿Qué empresa …
Unidad Académica	¿Qué facultad … ¿Qué unidad académica … ¿En qué unidad académica … ¿Qué escuela … ¿En qué escuela … ¿Qué sede … ¿En qué sede … ¿Qué delegación …¿En qué delegación...
Carrera	¿Qué carrera … ¿En qué carrera …
Fecha	¿Cuándo … ¿En qué mes … ¿En qué año … ¿En qué fecha …
Resolución	¿En qué resolución… ¿Cuál es el n° de la resolución en que…
Titulo	¿Qué titulo …

Cada una de las categorías de preguntas lleva asociados patrones típicos de la pregunta, como se explica a continuación.

5. Construcción de la consulta

Después de haber determinado el tipo de pregunta, es necesario generar una consulta que recupere documentos del índice, que deberán contener una o más anotaciones correspondientes al tipo de la pregunta, y los conceptos relevantes de la pregunta del usuario, y todo ello dentro de una ventana de texto de longitud limitada. En esta sección se describe la construcción de las consultas, explicando previamente el lenguaje utilizado para expresar las mismas.

5.1. Formato de las consultas

Para construir las consultas hemos adoptado el lenguaje de consultas *XML Fragments* [11] debido a su expresividad y a la disponibilidad de un motor de búsquedas que acepta consultas en este lenguaje. El motor acepta en las consultas la sintaxis de las anotaciones de UIMA y además texto libre y operadores estándar como AND, OR, NOT para combinar los fragmentos que componen las consultas.

Una consulta en el lenguaje *XML Fragments* consta de una estructura XML sub-especificada que combina consultas de palabras con consultas de información anotada. Esto permite buscar conceptos más específicos, e incluso relaciones entre objetos (por ejemplo, "la persona y la unidad académica deben aparecer en la misma frase" o "una persona que pertenece a cierta unidad académica" o "una unidad académica vinculada a una institución" porque aparecen cerca en el documento).

Las operaciones que se pueden realizar con *XML Fragments* incluyen [11]:

- conceptualización, que generaliza un literal (string) a un concepto del sistema de tipos. Por ejemplo, la consulta "institución" devuelve documentos en que aparezca esa palabra, mientras que *<Institución>.<Institución/>* busca ocurrencias de la anotación institución, aunque la palabra "institución" misma no aparezca en el documento.
- restricción: restringe las ocurrencias de etiquetas XML indicando qué palabras deben aparecer. Por ejemplo *<Institución> ingenier </Institución>* devuelve ocurrencias de la institución Consejo Profesional de Ingenieros mientras que *<UnidadAcadémica> ingenier </UnidadAcadémica>* devuelve documentos sobre la Facultad de Ingeniería.
- relación: una anotación representa la relación entre los términos de la consulta. Ej. *<FechaResol año=2007 mes=Septiembre><FechaResol>* encuentra las resoluciones de septiembre del 2007 (y no simplemente resoluciones de 2007 en que aparezca "septiembre").

5.2. Construcción de las consultas

El presente trabajo describe el prototipo inicial del sistema de BR, en que solo se han considerado preguntas que tienen como respuesta una persona y en particular que comienzan con el pronombre interrogativo "Quién". En esta sección se describe el proceso de armado de la consulta en el lenguaje XML Fragments a partir de la consulta en lenguaje natural del usuario. Por comenzar con "Quién", como se indicó en la Sección 4, se espera que la respuesta sea una persona, en particular un fragmento de algún documento que haya sido anotado como persona. Por tanto, parte de la consulta será la expresión *<Persona>.</Persona>*, que indica que el texto devuelto debe contener una anotación de tipo persona.

Para construir la consulta es necesario analizar la pregunta. Hay diversos enfoques más o menos complejos de "comprensión" de la pregunta, pero para nuestro caso un enfoque sencillo, basado en características léxico-sintácticas es suficiente. Este análisis sintáctico realizado se denomina *chunking*. Podría considerarse una variante superficial de un análisis sintáctico completo (*parsing* o

deep parsing), y ahorra tiempo de procesamiento al mismo tiempo que se centra en los elementos claves de la frase, en este caso la consulta, como son el verbo y los sintagmas nominales, ignorando otras partes de la frase, lo cual es suficiente para nuestra tarea. No necesitamos un análisis sintáctico completo, sino lo suficiente para obtener las partes claves de la pregunta [12].

En el procesado de la pregunta se procedió a anotar la misma de dos formas diferentes:
(a) utilizando el etiquetador POS de FreeLing, una colección de herramientas de código abierto para el análisis del lenguaje (http://nlp.lsi.upc.edu/freeling/).
(b) utilizando los anotadores desarrollados o aprendidos automáticamente en UIMA (sección 3) que detectan Personas, Títulos, Instituciones, Unidades Académicas, Carreras y Fechas. Frecuentemente las preguntas tienen referencias a entidades con nombre, tales como nombres propios, organizaciones, lugares, fechas etc. Es de utilidad que estas entidades sean gestionadas como un solo token, en lugar de como varias palabras.

Una vez anotada la pregunta, se utilizan patrones para extraer los componentes relevantes de la misma. En el caso de las preguntas "Quién" el foco es el sintagma nominal que aparece después del verbo. De forma heurística detectamos en la pregunta componentes que podríamos llamar grupos nominales que aparecen después del verbo, afines a los sintagmas nominales pero sin la garantía de realizar un análisis sintáctico completo. Cada uno de estos grupos nominales puede ser un nombre común, un nombre común seguido de un adjetivo, un nombre propio. Además, si alguno de estos grupos va separado de otro nombre por una sola palabra, en general una preposición, se anexa éste al grupo. Para cada uno de estos grupos nominales *gn*, se añade a la consulta la cadena *<>"gn"</>*

A estos términos obtenidos mediante la aplicación del patrón se añaden los devueltos por los anotadores, que ya están en el lenguaje *XML Fragments*. Las Tablas 2 y 3 presentan dos ejemplos.

En el prototipo actual es preciso que la pregunta se formule de manera correcta sintáctica y ortográficamente para poder obtener los resultados correctos. Por otro lado, este enfoque da pie a dificultades. En el segundo ejemplo si el evaluador es una mujer, no se obtendrá respuesta, ya que el texto del documento buscado probablemente incluya la palabra "evaluadora" en lugar de "evaluador". Un uso adecuado de la lematización puede resolver algunos de estos problemas.

6. Recuperación de documentos relevantes

La indexación y recuperación de documentos está basada en el paquete *SemanticSearch 2.1* [13] el cual añade a UIMA un motor de búsqueda semántica.

Tabla 2. Construcción de la consulta: Ejemplo 1

Pregunta: *¿Quién fue designado decano de la Facultad de Ingeniería e Informática en el 2008?*
Tipo de pregunta: *Quién*. Respuesta esperada: *Persona*
Anotaciones: *<UA>Facultad de Ingenieria e Informatica</UA>*
Etiquetas proporcionadas por Freeling: fue → VERBO designado → VERBO decano → NOMBRE_COMUN de → OTRO la → OTRO Facultad_de_Ingenieria → NOMBRE_PROPIO e → OTRO Informatica → NOMBRE_PROPIO en → OTRO el → OTRO 2008 → NUMERAL
Consulta resultante en XML Fragments: *+<Persona> . </Persona> +<>decano</>* *+<UA>Facultad de Ingenieria e Informatica</UA>* *+<FechaResol anio="2008"></FechaResol>*

Tabla 3. Construcción de la consulta: Ejemplo 2

Pregunta: *¿Quién ha sido evaluador externo de proyectos del Consejo de Investigaciones?*
Tipo de pregunta: *Quién*. Respuesta esperada: *Persona*
Anotaciones: *<UA>Consejo de Investigaciones</UA>*
Etiquetas proporcionadas por Freeling: ha → VERBO sido → VERBO evaluador → NOMBRE_COMUN externo → ADJETIVO de → OTRO proyectos → NOMBRE_COMUN de → OTRO el → OTRO Consejo_de_Investigaciones → NOMBRE_PROPIO
Consulta resultante en XML Fragments: *+<Persona> . </Persona>+<>evaluador externo de proyectos</>+<UA>Consejo de Investigaciones</UA>*

La búsqueda semántica es una búsqueda en que se especifica la intención semántica de la consulta, no solo los términos que se desean encontrar. Para ello se incluyen en la consulta entidades con nombre, anotadas como tal, o hasta sus atributos. Por ejemplo, se podría especificar que se está buscando una carrera de "ingeniería" (<Carrera>ingeniería</Carrera>), y tal consulta no devolvería resultados sobre otros usos de la palabra ingeniería.

Nuestra implementación de búsqueda semántica usando *SemanticSearch* incluye un CAS Consumer que llena un índice con el contenido del documento y con las anotaciones añadidas por los anotadores ya descriptos implementados en UIMA. Las facilidades de búsqueda incluyen, además de la búsqueda de términos en el contenido del documento, la búsqueda por anotaciones.

Ambas se realizan mediante el lenguaje *XML Fragments* ya descrito. Este paquete es software libre aportado por IBM y para acceder al índice existe una API que puede ser utilizada en una aplicación en Java.

7. Extracción y presentación de respuestas

La consulta al índice devuelve una lista de *d* archivos XMI candidatos a contener la respuesta buscada. Estos documentos contienen al menos una anotación del tipo de la respuesta, además de las anotaciones y palabras clave detectadas a la hora de armar la consulta. Los *d* documentos están ordenados en un *ranking* en base a factores tales como la capacidad de los términos de distinguir un documento de otros (ej. un término que aparece en la mayoría de los documentos contribuye menos a la puntuación, que otro que aparece solo en un conjunto pequeño de documentos), el número de ocurrencias de los términos buscados en el documento, o la proximidad entre los mismos. La métrica utilizada no está documentada en *Semantic Search.* En general la respuesta buscada, si existe, aparece en los primeros documentos del ranking, y en los primeros experimentos con nuestro corpus para preguntas"Quién" el valor *d*=15 es más que suficiente.

A continuación se procesa esta colección de documentos para extraer hasta un máximo de *r* respuestas candidatas (con valor *r*=5 adecuado para nuestro corpus).

Cada documento se divide previamente en fragmentos. Un fragmento en nuestro corpus es la secuencia de caracteres más larga entre caracteres punto o punto y coma. Para esto utilizamos la herramienta *OpenNLP Sentence Detector* que detecta oraciones o fragmentos de textos en inglés, adaptada a nuestro caso. Dadas las características de nuestro corpus, en que en general las oraciones son largas y a menudo con varias oraciones subordinadas, los fragmentos suelen ser largos.

En cada documento se ubica una anotación del tipo de respuesta buscada, y alrededor de ésta se va a centrar el análisis. Se busca una ventana de *c* que incluya dicha anotación y además las anotaciones y palabras claves que componen la consulta, y se devuelve el fragmento o fragmentos de texto que incluyen todos estos elementos. En los experimentos un valor de *c*=150 obtiene resultados adecuados.

En nuestro corpus es frecuente que aparezca la misma respuesta más de una vez en el documento (por ejemplo como parte de los considerandos de la resolución y como parte de las decisiones resueltas). Por ello de cada documento solo se devuelve una respuesta.

La anotación del tipo buscado en el contexto del fragmento que la incluye constituye una respuesta. En ese fragmento la anotación respuesta aparece resaltada en un color, y los términos de la consulta aparecen en negrita. En los documentos de nuestro corpus los fragmentos suelen ser largos, y el fragmento en que aparece la anotación que sirve de respuesta provee suficiente contexto para dar confianza al usuario y que éste decida si la respuesta obtenida es adecuada para la consulta realizada. La sección siguiente justifica la elección de fragmentos de esta manera como resultado a mostrar al usuario.

Finalmente se devuelven las primeras *r* respuestas obtenidas (las anotaciones en su contexto), que corresponden a *r* documentos diferentes. El orden de las respuestas es el mismo en que aparecieron sus respectivos documentos. Por cada respuesta se devuelve el fragmento en que está incluida.

Las Figuras 4 y 5 muestran los resultados de las consultas de las Tablas 2 y 3 respectivamente.

7.1 Consideraciones sobre la presentación de respuestas

Lin et al [14] exploraron los diversos tipos de interfaces para los sistemas de BR, partiendo del paradigma foco-más-contexto. Dado que la mayoría de los sistemas extraen las respuestas de documentos de texto, el texto que rodea a la respuesta sirve como proveedor natural de contexto. Los autores realizaron un experimento para decidir cuánto contexto debería devolver el sistema, en particular:

- la respuesta exacta
- respuesta-en-oración: la respuesta exacta con la frase de la que se extrajo
- respuesta-en-párrafo: la respuesta exacta con el párrafo del que se extrajo, en el que se resalta la frase que contiene la respuesta.
- respuesta-en-documento: la respuesta exacta con el documento completo del que se extrajo, en el que se resalta la frase que contiene la respuesta.

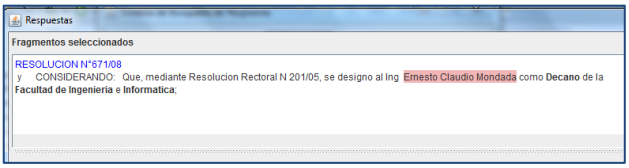


Figura 4. Resultados de la consulta del ejemplo 1

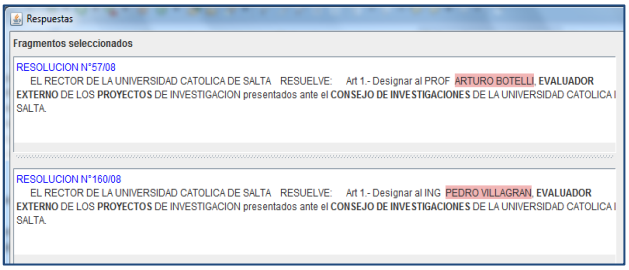


Figura 5. Resultados de la consulta del ejemplo 2

El estudio mostró que los usuarios prefirieron la opción respuesta-en-párrafo, mientras que la opción respuesta exacta fue la menos preferida. En particular, los pronombres causaron problemas, especialmente en la opción respuesta-en-oración al tomar las frases con pronombres fuera de contexto. Además se evaluó la percepción de confiabilidad de las respuestas: los participantes necesitaron en promedio al menos un párrafo para formarse un juicio en cuanto a la confiabilidad de la respuesta.

8. Evaluación y trabajo futuro

Dado que el sistema de BR está en etapa preliminar no tenemos un mecanismo formal de evaluación. No obstante en nuestros experimentso utilizamos como punto de referencia para la evaluación el buscador semántico descrito en la Sección 3, que consta de una interfaz tipo formulario para realizar consultas en el lenguaje XML Fragments y devuelve los documentos relevantes. Para evaluar la construcción de las consultas a partir de las preguntas en lenguaje natural se comparan los documentos obtenidos por el sistema BR con los del buscador semántico. Para evaluar si las respuestas obtenidas (la respuesta resaltada dentro de su contexto) son adecuadas, realizamos inspección ocular para verificar que las respuestas obtenidas son correctas en base a los contenidos del corpus. Los resultados preliminares, para preguntas de tipo "Quién" son satisfactorios. A partir de estos resultados y la arquitectura básica descripta se están extendiendo los tipos de preguntas que se pueden contestar, y las características de las respuestas (por ejemplo, en el caso en que la respuesta no sea solo una entidad sino un conjunto de ellas).

9. Conclusiones

Los sistemas de búsqueda de respuestas son una de las áreas de investigación más activas en la minería de textos. En este trabajo hemos presentado una arquitectura preliminar para un sistema de búsqueda de respuestas en un corpus de más de 8000 documentos que contienen resoluciones rectorales, que por el momento responde solamente a preguntas factoides sencillas, pero que está sirviendo de plataforma para atacar tanto preguntas más complejas como tipos de respuestas también más complejos.

Agradecimientos

Este trabajo ha sido financiado en parte por el Consejo de Investigaciones de la Universidad Católica de Salta (Resol Rect 839/13).

Referencias

[1] M. Á. García Cumbreras, "BRUJA: Un Sistema de Búsqueda de Respuestas Multilingüe", *Colección de Monografías de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN)*, nº 9, Septiembre 2010.

[2] A. Pérez y A. C. Cardoso, "Categorización automática de documentos", de *Simposio Argentino de Inteligencia Artificial*, *40 JAIIO*, Córdoba, 2011.

[3] A. Pérez y A. C. Cardoso, "Extracción de entidades con nombre", de *Simposio Argentino de Inteligencia Artificial*, *42 JAIIO*, Córdoba, 2013.

[4] W. Lehnert, "Human and computational question-answering", *Cognitive Science*, nº 1, pp. 47-73, 1977.

[5] B. Webber y N. Webb, "Question Answering", de *The Handbook of Computational Linguistics and Natural Language Processing*, A. Clark, C. Fox y S. Lappin, Edits., Wiley-Blackwell, 2010, pp. 630- 654.

[6] M. Montes-y-Gómez, L. Villaseñor-Pineda y A. López-López, "Mexican Experience in Spanish Question Answering", *Computación y Sistemas*, vol. 12, nº 1, Julio-Septiembre 2008.

[7] M. Pasca, "Lightweight Web-Based Fact Repositories for Textual Question Answering", de *CIKM'07*, Lisboa, 2007.

[8] D. Ferrucci y A. Lally, "Building an example application with the Unstructured Information Management Architecture", *IBM Systems Journal*, vol. 45, nº 3, 2004.

[9] OMG, "XML Metadata Interchange (XMI), v 2.1.1", 2007.

[10] C. Amaral, D. Laurent, A. Martins, A. Mendes y C. Pinto, "Design and Implementation of a Semantic Search Engine for Portuguese", de *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 2004.

[11] J. Chu-Carroll, J. Prager, K. Czuba, D. Ferrucci y P. Duboue, "Semantic Search via XML Fragments: a High-Precision Approach to IR", de *Proceedings of the 29th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, New York, 2006.

[12] G. Ingersoll, T. Morton y A. Farris, *Taming Text*, Shelter Island: Manning, 2013.

[13] Apache UIMA Development Community, "UIMA Tutorial and Developers' Guides, version 2.6.0", May 2014. Disponible: https://uima.apache.org/d/uimaj-2.6.0/index.html.

[14] J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huunh, B. Katz y D. Karger, "The Role of Context in Question Answering Systems", de *CHI '03: Extended Abstracts on Human Factors in Computing Systems*, Fort Lauderdale, Florida, 2003.

Agentes Virtuales en Idioma Español

Bender, Adrián
Universidad Del Salvador,
Facultad de Ingeniería
adrian.bender@gmail.com

García Calabria, Federico
Universidad Del Salvador,
Facultad de Ingeniería
federico.garciacalabria@gmail.com

Mazza, Néstor
Universidad Del Salvador,
Facultad de Ciencias de la Administración
nestor.mazza@sustentum.com

Nicolet, Santiago
Universidad Del Salvador,
Facultad de Ingeniería
santiago@nicolet.com.ar

Abstract

Los agentes virtuales son programas informáticos capaces de interactuar con seres humanos en lenguaje natural. La tecnología de agentes virtuales está considerada entre las más prometedoras en la interacción humano - máquina, con aplicaciones en múltiples campos: servicio al cliente, soporte, entretenimiento, educación, marketing, ventas, etc. El español es uno de los idiomas más hablados en el mundo y uno de lenguajes oficiales de las Naciones Unidas. Sin embargo, comparado con otros lenguajes, los agentes virtuales de habla hispana están empezando a surgir lentamente. En esta investigación se revisa la aplicación actual de agentes virtuales para usuarios de habla hispana, con especial foco en el área de servicios al cliente. Adicionalmente, y en base a una encuesta, se determinan las características que hacen que el desempeño de un agente virtual sea similar al que su contraparte humana ofrecería.

Virtual agents are software capable of human-like communication, using natural languages. Virtual agents' technology is considered among the most promising in the interaction human - machine, with applications in multiple fields: customer service, support, entertainment, education, marketing, sales, etc. Spanish is among the most spoken in the world and one of the United Nations' official languages. Although, compared with other languages, Spanish capable virtual agents are slowly starting to emerge. In this research we review the current application of virtual agents for Spanish speaking users with special focus on Customer Services. Additionally, and based on a survey, several approaches are applied to determine the characteristics that make a virtual agent performance similar to its human counterpart.

1. Introducción

La Inteligencia Artificial es relativamente reciente en la historia de la humanidad, y su origen se remonta a 1950 [1].

La interacción con los seres humanos a través de lenguaje natural (Mandarín, Ruso, Español, Inglés, Francés, etc.) es un gran desafío para las Ciencias de la Computación, específicamente para una rama de la Inteligencia Artificial: el Procesamiento del Lenguaje Natural (PLN) [2].

Está generalmente aceptado que el primer agente o bot (de robot) con capacidad razonable para interactuar con seres humanos utilizando lenguaje natural fue Eliza, quien simuló a un psicoterapeuta aplicando el método humanista de Roger. Eliza fue creado por Joseph Weizenbaum en el MIT durante 1964-1966 [3].

El aumento de la potencia de las computadoras, tanto a nivel servidor como cliente, la amplia infraestructura de comunicaciones proporcionada por Internet, los avances del PLN y de la síntesis de voz entre otros, han permitido el desarrollo de robots de conversación, ahora conocidos como agentes virtuales, en muchos campos: atención al cliente, entretenimiento, educación, marketing, ventas, etcétera [4].

Probablemente, y debido al valor económico asociado, los agentes virtuales en el campo de servicio al cliente son unos de los más atractivos.

Hasta el momento, más de mil agentes virtuales se han publicado en todo el mundo [5]. Menos de cien son de habla hispana, a pesar del hecho de que el Español es,

después del Mandarín, la segunda lengua nativa en el mundo, y ocupa el cuarto lugar de los lenguajes más hablados [6].

En esta investigación, se revisa la aplicación actual de agentes virtuales en el servicio al cliente para los usuarios de habla hispana, tanto en España como en América Latina, con el objetivo de comprender que aspectos de un agente virtual son considerados importantes a la hora de valorar su utilidad.

En las subsecciones siguientes se describen los hitos más importantes que ha definido la tecnología aplicada al servicio de atención al cliente, centrándonos finalmente en los agentes virtuales. La sección 2 describe nuestro foco de interés y el método empleado para relevar datos sobre agentes virtuales de habla hispana en atención al cliente en la actualidad. La sección 3 despliega los resultados obtenidos. En la sección 4 se plantea una discusión y el análisis de los resultados. Finalmente, en la sección 5 se resumen los principales aportes del trabajo y se presentan futuras líneas de investigación.

1.1. Tecnología y Atención al Cliente

La tecnología ha sido aplicada al área de servicio al cliente de forma ininterrumpida y creciente. Uno de los resultados de esta relación son las tecnologías de autoservicio (SST: Self Service Technologies), que son "*las interfaces tecnológicas que permiten a los clientes consumir servicios sin la participación directa de empleados*" [7].

Los cajeros automáticos (ATM: Automatic Teller Machine) en los bancos son unos de los primeros y más importantes ejemplos de SST [8]. Un cliente bancario puede insertar su tarjeta, introducir su número de PIN, seleccionar la cuenta de la cual extraer, introducir la cantidad que desea retirar y recibir el dinero en la mano. Hace décadas, estas acciones fueron realizadas por un cajero humano.

Los primeros dispositivos eran independientes, aunque poco flexibles. Los avances en la tecnología han añadido nuevas capacidades: auto-depósito, pago de facturas y transferencias de dinero, por nombrar unos pocas.

Estos avances liberaron al personal en las sucursales de banca minorista para dedicarlos a atraer a clientes en los servicios de más alto valor, tales como seguros, hipotecas y el comercio del mercado de valores [9]. Mientras que los cajeros automáticos se convirtieron en la columna vertebral del sistema de pagos; y abrieron la puerta a la atención telefónica automatizada y a la banca por Internet [10].

A través del tiempo, el uso de las SST se hizo más generalizado y, han reemplazado o complementado las interacciones tradicionales, cara a cara, de servicios con la intención de hacer las transacciones de servicios en forma más rápida, más cómoda y más precisa [11].

Otro uso de la tecnología en el servicio de atención al cliente son los sistemas de respuesta de voz interactiva (IVR: Interactive Voice Response). Los primeros casos de éxito IVR aparecieron en la década de 1970, y han sido ampliamente utilizados en los centros de llamadas desde la década de 1980. Los sistemas IVR permiten a las organizaciones ofrecer información y servicios a sus clientes durante todo el día, siete días a la semana. Oradores humanos, entre los primeros IVR, fueron reemplazados por guiones interactivos pre-grabados de alta calidad a los que el usuario daba respuestas presionando las teclas de un teléfono [12].

Con el tiempo, los IVR han incluido el reconocimiento de voz. Estas aplicaciones permiten a las personas que llaman, usar su propia voz en lugar de un teclado para completar las transacciones [13].

1.2. Agentes Virtuales

Los agentes virtuales (VA: Virtual Agents) son programas informáticos que permiten tener una comunicación natural, similar a la humana, con una máquina [14].

En servicio al cliente, los agentes virtuales han sido considerados como una herramienta de auto-ayuda para orientar a los consumidores. Pero últimamente la tecnología de agente virtual se ha convertido en una alternativa proactiva de resolución. La capacidad de los agentes para resolver problemas cada vez más complejos de los consumidores a través de conversaciones personalizadas y datos provistos por distintas aplicaciones de la empresa está transformando a los agentes virtuales en una herramienta muy potente para solucionar problemas. Esto hace que los agentes virtuales sean priorizados tanto por consumidores, que ansían velocidad y autosuficiencia en su demanda de servicios, como por los profesionales del negocio electrónico, que están tratando de menguar las interacciones asistidas por personal y brindar una alternativa en línea más importante [15].

Los seres humanos somos agentes sociales, interactuando unos con otros. Los agentes sociales poseen características humanas, como la voz humana, la apariencia de la cara humana y el movimiento antropomorfo del cuerpo. Debido a esto, los agentes virtuales diseñados para interactuar con los seres humanos de una manera similar a la humana suelen estar equipados con caras, con movimiento y capacidades de habla [16].

Picard [17] dijo "*Una máquina, aunque limitada a la comunicación de texto, será un comunicador más eficaz si se le da la capacidad de percibir y expresar emociones*". Así, los agentes virtuales deben tener

mecanismos que permitan percibir y manifestar emociones trabajando en conjunto con su sistema basado en reglas para ser realmente eficaces [18].

Las principales formas para expresar la emoción son: animación, lenguaje hablado y gestos [19].

El generar animaciones eficientes y realistas de criaturas virtuales ha sido siempre un gran desafío en el campo de la animación por computadora [20]. Por otra parte, en los últimos años, una de las áreas de investigación más activa en el área de animación gráfica ha sido la de agentes virtuales [21].

Para expresarse en lenguaje hablado, los agentes virtuales utilizan voz sintetizada (TTS: Text-To-Speech Engine) [19]. Para un resultado realista, los movimientos de labios deben estar perfectamente sincronizados con el audio [22].

Se han adoptado numerosos enfoques para proporcionar gestos a los agentes virtuales [23]. Entre los primeros, el kit de herramientas de animación de expresión comportamental (BEAT: Behavior Expression Animation Toolkit) empleó un conjunto de generadores de comportamiento para seleccionar y programar comportamientos conversacionales, tales como: gestos de manos, asentimiento con la cabeza, etcétera [24].

Algunas aplicaciones de agentes virtuales en educación por ejemplo, incluyen dominios específicos de conocimiento y rutinas para facilitar los procesos relacionados a la enseñanza-aprendizaje.

Dentro de las primeras aplicaciones de la informática a la educación con reconocimiento del lenguaje natural (inglés), se encuentra el proyecto Scholar, de Jaime Carbonel en 1970 [25], el cual utilizaba una red semántica para un dominio de conocimiento limitado: geografía sudamericana, e introducía un nuevo paradigma permitiendo que el alumno recorra la red libremente, en contraposición con los CAI (Computer Aided Instruction) tradicionales.

Sofía de Harvard, especializada en matemáticas, es probablemente uno de los desarrollos más interesantes en idioma inglés [26].

Las experiencias en el campo de la educación en idioma español son muy pocas, siendo Ariel, especializado en Gestión de Recursos Informáticos el más elaborado y estudiado. El mismo cuenta con un avatar 3D y TTS y se encuentra en operación desde 2009 en la Universidad de Buenos Aires, con resultados prometedores [27].

2. Elementos del Trabajo y Metodología

Con el objeto de analizar el estado del arte en cuanto al desarrollo de agentes virtuales para atención al cliente en español, elegimos doce de los agentes existentes. Para seleccionarlos, utilizamos como fuente a Chatbots.org, un sitio que contiene un catálogo de todos los chatbots (asistentes virtuales, chatbots, agentes conversacionales, agentes virtuales, etc.) disponibles en el mundo [5].

El criterio utilizado para la selección fue el tomar dos agentes de cada uno de los seis principales proveedores, definidos éstos según la cantidad de agentes desarrollados y publicados al mercado. La elección de los dos agentes a utilizar de cada proveedor fue aleatoria.

Todos los agentes tienen una interfaz web y son accesibles a través de Internet. Al ser de código privado desconocemos la tecnología utilizada.

Enfocamos nuestro trabajo en las siguientes características de los agentes: presencia de avatar, síntesis de voz y técnica de reconocimiento de texto.

Encontramos que siete de los doce agentes tienen avatar, el cual les brinda una apariencia de un cara humana en tres dimensiones. Todos los agentes estudiados que poseen un avatar, tienen TTS y gestos. Los cinco agentes restantes no tienen ni avatar ni TTS.

En cuanto al reconocimiento de texto definimos una tipificación de los agentes entre las siguientes tres clases:

- Buscador por palabras clave (BPC): el agente reconoce palabras clave en el texto ingresado por el usuario, hace una búsqueda en la base de datos interna del sitio y responde con enlaces a diferentes partes del sitio.
- Buscador de tipo IVR (BIVR): el agente reconoce palabras clave en el texto ingresado por el usuario y devuelve respuestas predefinidas. Si las palabras clave no coinciden con ningún registro en la base de datos interna, ofrece cambiar la solicitud a una de las dos o tres posibles entradas más cercanas en la base y, eventualmente, responde enlaces a diferentes partes del sitio.
- Procesador de lenguaje natural (PLN): el agente reconoce lenguaje natural. Mantiene una conversación similar a la de un agente humano. Citando al agente virtual Inés (Ibanesto) "*Creo que me ves como un motor de búsqueda, imagíname como una persona*".

Para llevar a cabo nuestra investigación hemos diseñado una encuesta para recoger opiniones de los usuarios acerca de agentes virtuales. Nuestra intención fue evaluarlos en función a las siguientes tres dimensiones:

- Resolución de problemas: evalúa si el agente virtual logra satisfacer los requerimientos del usuario, ya sea usando conocimiento especializado o no.
- Interacción con seres humanos: determina si el agente tiene éxito en la comunicación con el usuario a través del reconocimiento de texto, siendo capaz de entender oraciones escritas correctamente o no.
- Reconocimiento de emociones: define si el agente virtual logra reconocer emociones dentro de las oraciones y responder en consecuencia.

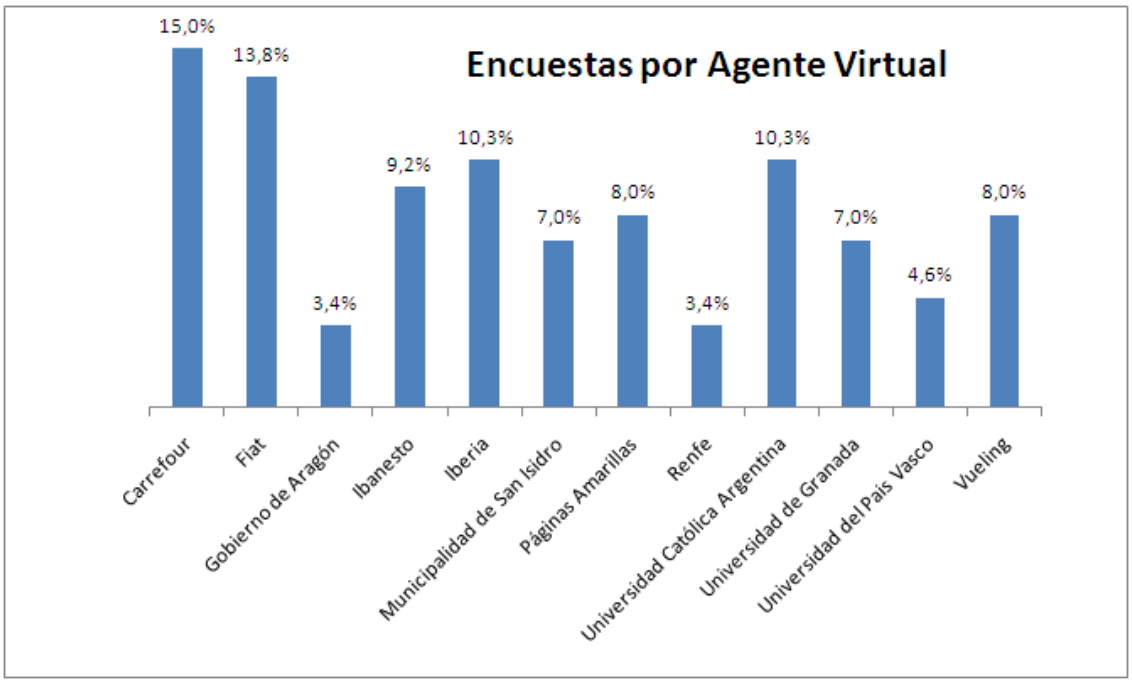


Figura 1. Distribución de encuestas por agente.

2.1. Metodología de la Encuesta

Diseñamos una encuesta con dieciséis puntos:

- Tres preguntas demográficas solicitando el género de la persona, la generación a la que pertenece, y el nivel de uso de la tecnología.
- Un punto requiriendo el ingreso del agente evaluado desde una lista de opciones.
- Nueve puntos contemplando tres preguntas por cada una de las tres dimensiones definidas.
- Una pregunta solicitando al usuario que evalúe la atención brindada por el agente virtual en comparación con el servicio que le hubiese brindado un agente humano en circunstancias similares.
- Dos solicitudes de comentarios donde el usuario se podía explayar libremente. Una acerca de los aspectos más positivos y otra de los más negativos de la experiencia con el agente.

Las nueve preguntas correspondientes a las dimensiones y la solicitud de comparación con el agente humano fueron incluidas en el formulario mediante una escala Likert de cinco puntos [28] en la que los encuestados debían responder según su grado de acuerdo o desacuerdo con la afirmación enunciada.

La encuesta se realizó durante mayo y junio de 2013 y se obtuvieron 87 respuestas. Involucró principalmente a estudiantes de la carrera Ingeniería en Sistemas de Información de la Universidad Del Salvador (Argentina) y a estudiantes de Administración de Recursos Informáticos de la Universidad de Buenos Aires (Argentina).

A cada participante se le pidió interactuar con uno o dos agentes asignados al azar, y completar la encuesta. Antes de la interacción con el agente, los voluntarios tenían que familiarizarse con la empresa u organización a la cual representa el agente virtual. El proceso de interacción consistió en:

- Presentarse ante el agente virtual.
- Preguntar acerca de la organización.
- Preguntar acerca de un producto o servicio.
- Realizar una queja acerca de un producto, de un servicio, o bien del nivel de la atención al cliente.
- Finalizar la conversación.

Las respuestas fueron guardadas para su análisis. Los resultados se presentan en el capítulo siguiente.

3. Resultados

La Figura 1 muestra el porcentaje de encuestas según el agente virtual evaluado.

Observamos los siguientes aspectos a partir de las respuestas obtenidas de las preguntas demográficas (que se visualizan en la Figura 2):

- El 64% de las personas que interactuaron con los agentes eran hombres y 36% eran mujeres.

- Casi el 68% eran nativos digitales o pertenecientes a la Generación Y.
- El 94% de los individuos manifestaron hacer uso intensivo de la tecnología.

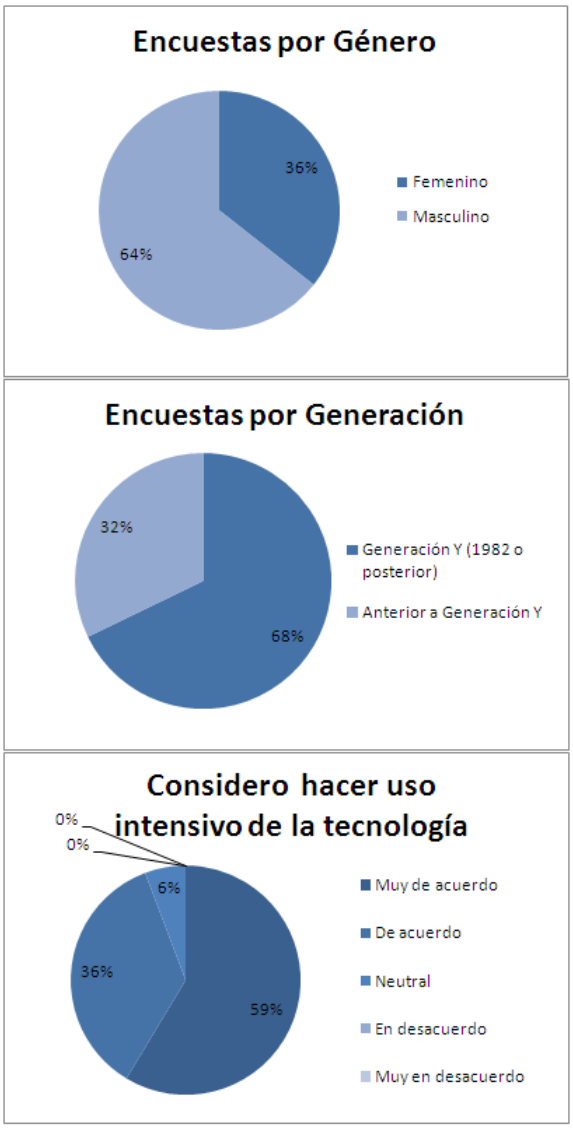


Figura 2. Perfil demográfico de los encuestados.

Las respuestas a las tres preguntas pertenecientes a la dimensión Resolución de problemas se resumen en la Figura 3.

Las respuestas a las tres preguntas correspondientes a la dimensión Interacción con seres humanos se grafican en la Figura 4.

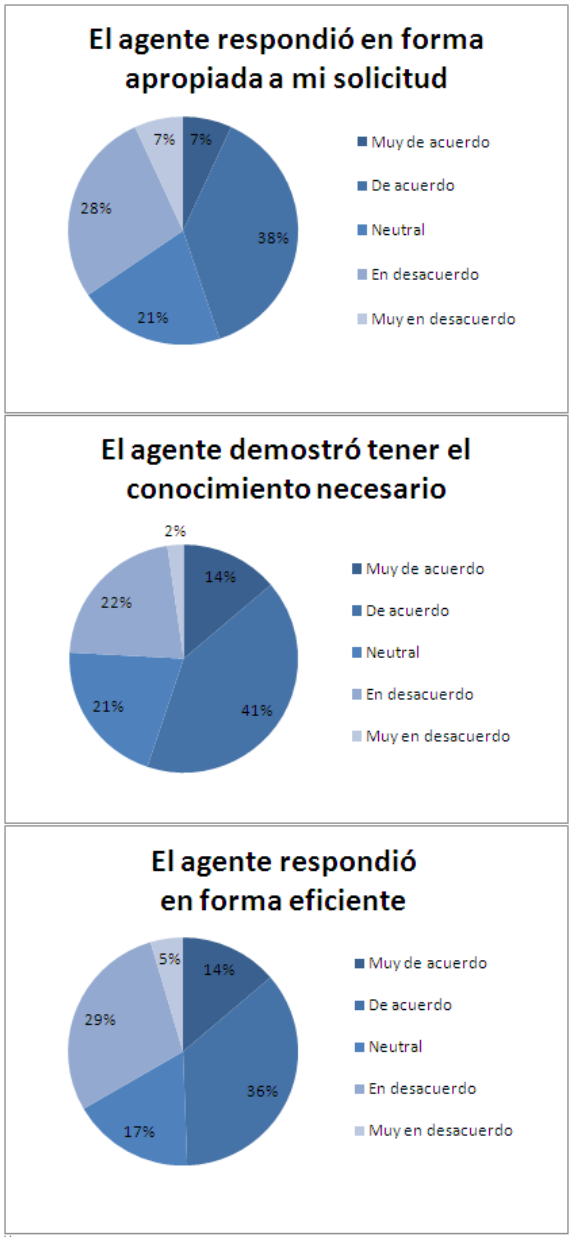


Figura 3. Respuestas de la dimensión Resolución de problemas.

Las respuestas a las tres preguntas correspondientes a la dimensión Reconocimiento de emociones se resumen en la Figura 5.

Las respuestas a la solicitud de comparación de la atención ofrecida por el agente evaluado en función al servicio que un agente humano habría proporcionado en circunstancias similares se muestran en la Figura 6.

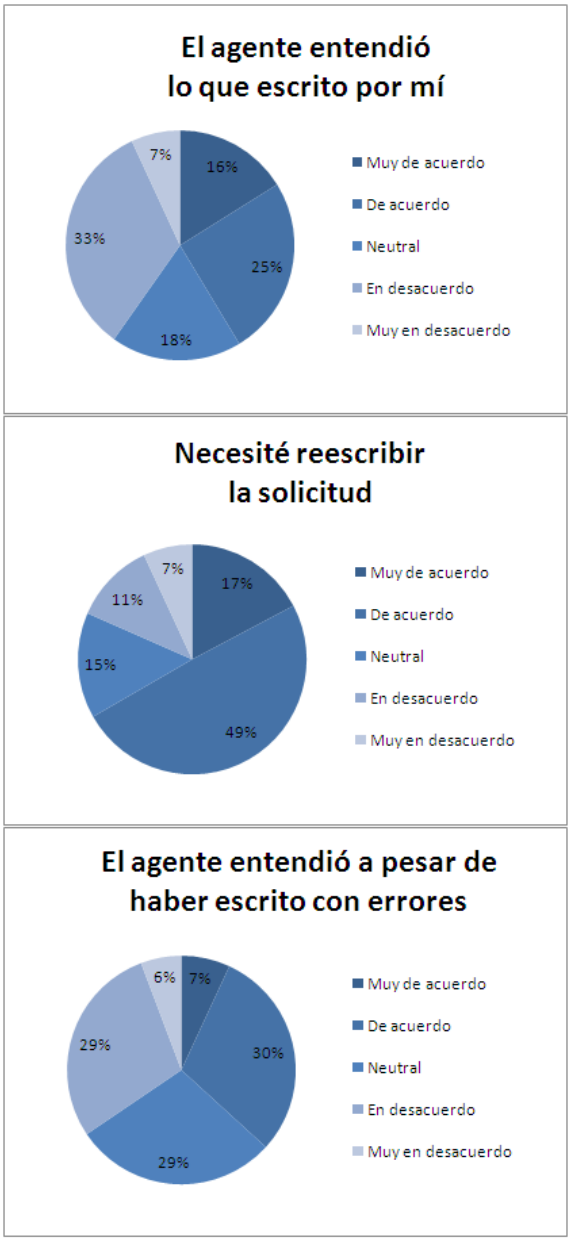


Figura 4. Respuestas de la dimensión Interacción con seres humanos.



Figura 5. Respuestas en la dimensión Reconocimiento de emociones.

Las dos últimas preguntas de la encuesta pedían los aspectos más positivos y negativos de la experiencia. Las respuestas fueron escritas en texto libre y luego fueron clasificadas en las tres dimensiones descritas anteriormente en este documento. Las descripciones que no coincidían con las categorías básicas no fueron tomadas en cuenta.

En la Figura 7 se comparan los porcentajes de los aspectos positivos y negativos por agente. La Figura 8 muestra los porcentajes de los comentarios tanto positivos como negativos por dimensión.



Figura 6. Respuestas a la pregunta que pedía contrastar la atención del agente virtual con la de un agente humano.

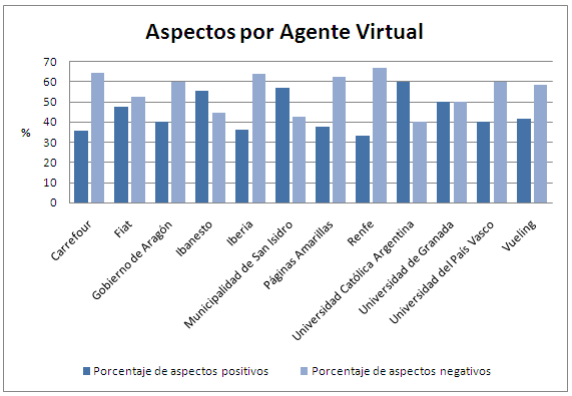


Figura 7. Porcentajes de aspectos positivos y negativos por agente.

4. Discusión

Sólo el 30% de los encuestados respondió que la atención ofrecida por el agente evaluado es similar, superior o muy superior al servicio que un agente humano habría proporcionado en similares circunstancias.

La impresión resultante al analizar los aspectos positivos y negativos de los agentes, es que la mayoría de los agentes virtuales tienen aún más errores que aciertos, excepto tres de los agentes bajo análisis: los iBanesto, la Municipalidad de San Isidro y la Universidad Católica Argentina. Además, resulta importante mencionar que al analizar los aspectos comentados según las dimensiones estudiadas:

- El 76% de los comentarios positivos estuvieron relacionados con la resolución de problemas.
- El 73% de los comentarios negativos estuvieron relacionados con la interacción con seres humanos.
- Sólo el 2% de los comentarios positivos estuvieron relacionados con el reconocimiento de emociones.

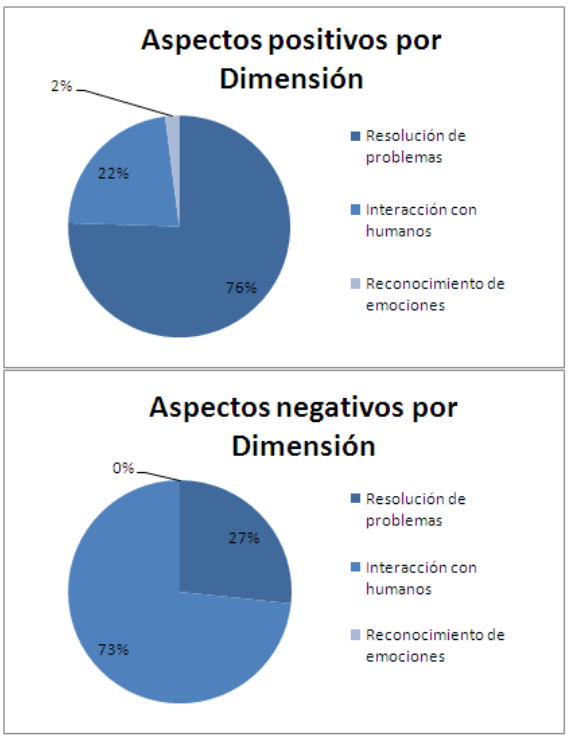


Figura 8. Distribución de aspectos positivos y negativos según dimensiones.

Analizando las respuestas en la dimensión Resolución de problemas observamos que los agentes muestran eficiencia y conocimiento, pero la respuesta que brindan no termina de satisfacer a los usuarios. A continuación algunos porcentajes que avalan dicha situación:

- Sólo el 33% de los encuestados no consideró eficiente la interacción con el agente.
- El 55% estuvo de acuerdo o muy de acuerdo en afirmar que el agente disponía de los conocimientos necesarios del producto o servicio ofrecido por la empresa.
- Sólo el 45% estuvo de acuerdo o muy de acuerdo en que el agente pudo responder sus requerimientos en forma apropiada.

Avalando lo observado en los comentarios negativos, ninguna de las respuestas en la dimensión Interacción con seres humanos alcanzó siquiera el 50% de acuerdo en las afirmaciones presentadas en la encuesta:

- El 67% estuvo de acuerdo o muy de acuerdo en que tuvo que reescribir su solicitud al considerar que el agente no entendió la misma.
- El 60% no estuvo de acuerdo en considerar que el agente entendió lo escrito por el usuario.

Respecto a la dimensión Reconocimiento de emociones:

- Sólo el 14% estuvo de acuerdo o muy de acuerdo en considera que el agente reconoció el estado emocional del usuario.

Finalmente, analizamos la dispersión de las respuestas a las 10 preguntas mencionadas precedentemente. Debido a la naturaleza ordinal de la escala de Likert, utilizamos la medida consenso [29] cuya fórmula está dada por:

$$\mathrm{Cns}(X) = 1 + \sum_{i=1}^{n} p_i \log_2 \left(1 - \frac{|X_i - \mu_X|}{d_X}\right)$$

Un nivel de consenso de 0 (cero), indica una amplia polarización, mientras que un nivel consenso de 1 (uno) indica un acuerdo total. En todos los casos relevados, el nivel de consenso fue bajo (entre 0,48 y 0,67), evidenciando una alta dispersión, Esta dispersión puede ser causada tanto por diferentes expectativas por parte de los encuestados, como por el diferente nivel de madurez alcanzado por los distintos agentes virtuales.

5. Conclusiones

Los datos recolectados muestran que, en líneas generales, los usuarios consideraran que el servicio ofrecido por un agente virtual es aún inferior al que un humano ofrecería en circunstancias similares.

Aún así, los agentes virtuales analizados resultaron relativamente buenos en la resolución de problemas pero, en conjunto, carecen de las habilidades lingüísticas necesarias para una fluida interacción con humanos.

Por otro lado, el escaso reconocimiento de emociones parecería afectar la calidad de la experiencia.

Finalmente, identificamos las siguientes líneas de investigación futura:

- Analizar el desempeño de los agentes según la tipificación realizada en este documento, en la sección 2. Esto permitiría profundizar el conocimiento del estado actual según tipos de agentes virtuales, y poder minimizar la dispersión encontrada.
- Construir un modelo inferencial que permita encontrar relaciones entre las respuestas dadas a las preguntas de las distintas dimensiones y la percepción que el servicio ofrecido es comparable al que un ser humano podría ofrecer en similares condiciones.

6. Agradecimientos

Queremos expresar nuestro sincero agradecimiento a todos aquellos que colaboraron con este trabajo: a la Universidad del Salvador, en especial a Marcelo Zanitti y a José María Samman por su apoyo; y a los alumnos de la materia Sistemas Inteligentes de la Universidad del Salvador y de Administración de Recursos Informáticos de la Universidad de Buenos Aires, por su participación en la experiencia.

Un especial agradecimiento a Javier Herrera, profesor de Investigación Operativa en la Universidad Del Salvador, por su ayuda y valiosas sugerencias sobre análisis estadístico.

7. Referencias

[1] Nilsson, N., *The Quest for Artificial Intelligence* por Cambridge University Press, 2009.

[2] Jurafsky, D. & Martin, J., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* por Prentice-Hall, 2000.

[3] Weizenbaum, J., "ELIZA—a computer program for the study of natural language communication between man and machine" en *Magazine Communications* de ACM, Vol. 9, pp. 36-45, 1966.

[4] Gilbert, M., Wilpon, J., Stern, B. & Di Fabbrizio, G., "Intelligent virtual agents for contact center automation" en *IEEE Signal Processing Magazine*, Vol. 22 Issue 5, 2005.

[5] Chatbots, Recuperado el 01 de Junio 2013 de http://www.chatbots.org, s/f.

[6] Nationalencyclopedin, Recuperado el 01 de Junio 2013 de http://www.ne.se/spr%C3%A5k/v%C3%A4rldens-100-st%C3%B6rsta-spr%C3%A5k-2010, 2010.

[7] Meuter, M., Ostrom, A., Roundtree, R. & Bitner, M., "Self-Service technologies: Understanding customer satisfaction with technology-based service encounters" en *Jornal de Marketing*, 2000, Vol. 64, No. 3, pp. 50-64.

[8] Cho, H., & Fiorito, S., "Self-Service technology in retailing. The case of retail kiosks" en *Symphonya Emerging Issues in Management*, Issue 1, Marketing Channels and Global Markets, 2010.

[9] Cluckey, S., "Branch transformation, brought to you by the ATM" en *ATM Marketplace*. Recuperado de http://www.atmmarketplace.com/article/206039/Branch-transformation-brought-to-you-by-the-ATM, 2010.

[10] Batiz-Lazo, B., "How the ATM Revolutionized the Banking Business" [Comentario en Web blog]. Recuperado de http://www.bloomberg.com/news/2013-03-27/how-the-atm-revolutionized-the-banking-business.html, 2013.

[11] Barnes, J., Dunne, P. & Glynn, W., "Self-service and technology: Unanticipated and unintended effects on customer relationships" en *Handbook of services marketing & management*, pp. 89-103, por T. Swartz, & D. Iacobucci (Eds.), 2000.

[12] Corkrey, R. & Parkinson, L., "Interactive voice response: Review of studies 1989–2000" en *Behavior Research Methods, Instruments, & Computers*, pp. 342-353, 2002.

[13] Lee, K. & Lai, J., "Speech versus touch: A comparative study of the use of speech and DTMF keypad for navigation" en *International Journal of Human–Computer Interaction*, Vol. 19, pp. 343–360, 2005.

[14] Niewiadomski, R., Demeure, V. & Pelachaud, C., Warmth, "Competence, Believability and Virtual Agents" en *Proceedings of the 10th International Conference on Intelligent Virtual Agents (IVA'10)* , 2010, pp. 272-285.

[15] Clarkson, D., Johnson, C., Stark, E. & Van Geldern, K., "It's Time To Give Virtual Agents Another Look" en *Forrester report10 © 2012*, Forrester Research Inc, 2009.

[16] Dalibard, S., Magnenat-Thalmann, N. & Thalmann, D., "Anthropomorphism of artificial agents: a comparative survey of expressive design and motion of virtual characters and social robots" en Workshop on Autonomous Social Robots and Virtual Humans de *25th Annual Conference on Computer Animation and Social Agents (CASA '12)*, 2012.

[17] Picard, R., *Affective Computing* por MIT Press, 1997.

[18] Lim, M. & Aylett, R., Feel the difference: "A guide with attitude" en *Proceedings of the 7th international conference on Intelligent Virtual Agents (IVA '07)* , 2007, pp. 317-330.

[19] Barbat, B.E. & Cretulescu, R., "Affordable affective avatars. Persuasion, emotions and language(s)" en *Proceedings of the 1st Balkan Conference in Informatics (BCI'2003)*, 2003.

[20] Huang, J. & Pelachaud, C., "Expressive body animation pipeline for virtual agent" en *Proceeding of the 12th international conference on Intelligent Virtual Agents (IVA'12)*, 2012, pp. 355-362.

[21] Gillies, M. & Ballin, D., "A model of interpersonal attitude and posture generation" en *Workshop on Intelligent Virtual Agents*, 2003, Vol. 2792, pp. 88-92.

[22] Zoric, G., Smid, K. & Pandzic, I., "Automated gesturing for embodied agents" en *Proceedings of the 2005 International Conference on New Frontiers in Artificial Intelligence (JSAI'05)* , 2006, pp. 401-410.

[23] Bergmann, K. & Kopp, S., "Increasing the expressiveness of virtual agents: autonomous generation of speech and gesture for spatial description tasks" en *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)* , 2009, Vol. 1 , pp. 361-368.

[24] Cassell, J., Vilhjalmsson, H. & Bickmore, T., Beat: "The behavior expression animation toolkit" en *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)* , 2001, pp. 477-486.

[25] Carbonell, J., "AI in CAI: An artificial intelligence approach to computer aided instruction" en *Science*, 1970, Vol. 167, pp: :190-202.

[26] Knill ,O. , Carlsson, J., Chi , A. & Lezama , M., "An artificial intelligence experiment in college math education" disponible en http://www. math. harvard. edu/~ knill/preprints/sofia. Pdf, 2004.

[27] Dorfman , M. et al. "Asistentes Virtuales de Clase como complemento a la educación universitaria presencial" en *40 JAIIO*, Agosto 2011.

[28] Likert, R., *A Technique for the Measurement of Attitudes* en Archives of Psychology, 1932, Vol. 140, pp. 1–55.

[29] Tastle, W. & Wierman, M., "Consensus and dissention: A measure of ordinal dispersion" en *International Journal of Approximate Reasoning*, 2006.

Tres Dimensiones de los Sistemas de Información

García, Sebastián José
Universidad Nacional de San Juan, Facultad de Ciencias Exactas, Físicas y Naturales
sgarcia@iinfo.unsj.edu.ar

Abstract. There is a considerable debate about Information Systems (IS) discipline crisis. In this work, three dimensions of IS to help understand that discipline is exposed. They are: data processing, representation and adaptation. Through showing the results of a research line is to provide an academic contribution to improve the concept of SI.

Resumen. Existe un importante debate acerca de la crisis en la disciplina Sistemas de Información (SI). En este trabajo se exponen tres dimensiones de los SI que ayudan a entender dicha disciplina. Ellas son: el procesamiento de datos, la representación y la adaptación. A través de mostrar los resultados de una línea de investigación se pretende entregar un aporte académico para mejorar el concepto de SI.

1 Introducción

Este trabajo está dirigido a los académicos y profesionales del área interesados en la disciplina de los SI, su historia y sus implicaciones para su futuro (http://history.aisnet.org./). Es interesante ya que se trata de esclarecer dicha disciplina, su crisis de identidad y la escasez de fundamentos conceptuales. El propósito del mismo es presentar tres dimensiones de los SI, que es el último resultado de una línea de investigación en SI que se ha llevado adelante durante los últimos 20 años. El trabajo entrega una nueva taxonomía de SI que permitirá realizar futuros aportes en la disciplina.

Los Sistemas de Información (SI) y las Tecnologías de Información (TI) continúan evolucionando significativamente y teniendo hoy un impacto profundo en la sociedad a nivel global que ha sido prometido y esperado durante años. Los avances son muy rápidos y sus capacidades están cambiando la forma en que vivimos y trabajamos.

Sin embargo, el debate sobre la crisis en la disciplina de los SI presentado por [2], [3], [9] y muchos otros autores, se atribuye principalmente a que es indefinida su identidad. Miles de conceptualizaciones sobre la disciplina han dado lugar a una plétora de recomendaciones no estructuradas y desconectadas que ponen restricciones en el crecimiento del campo de SI. Por tanto, es esencial contar con un marco teórico que explique cuál es la identidad de la disciplina SI.

Esta crisis disciplinaria se ha visto reflejada en tres aspectos: el primero de ellos es el académico. Ha disminuido significativamente el interés en su estudio, reduciendo la cantidad de alumnos, tanto a nivel global como local. Existe a nivel internacional significativos esfuerzos en dar un marco teórico que identifique a la disciplina y de formular un modelo global de curricula en SI [1]. En nuestro país el esfuerzo está reflejado en la Resolución 786/2009 del Ministerio de Educación [19] que pretende ser una referencia para solucionar la crisis curricular a nivel nacional [9] [15]. Esta resolución lleva a que las Universidades Nacionales estén en proceso de adaptación de sus curriculas, lo que implica que se vienen cambios radicales en la enseñanza de SI.

El segundo aspecto en crisis de los SI es el organizacional ya que existe una gran confusión entre los SI y la TI [10]. Las organizaciones y los usuarios confunden tareas y actividades de SI con TI produciendo una fuerte subutilización de los recursos de la organización.

Y por último, un tercer aspecto en el que se manifiesta esta crisis es en el ámbito de los desarrolladores de SI, quienes tienen una gran responsabilidad en el entendimiento del dominio del problema a solucionar.

Teniendo en cuenta la problemática planteada en estos diferentes aspectos surge la necesidad de buscar algún mecanismo que permita generar conceptualizaciones ordenadas bajo un criterio formal y con un respaldo teórico asociado en la disciplina de SI.

En [10] se muestra el estado del arte acerca de qué son los SI, para qué sirven y cuál es la diferencia con las TI. Se ha reunido información acerca de cuáles son los fundamentos conceptuales de SI. En [15] y [21] se plantean casos de estudio de la problemática curricular de carreras en SI a nivel nacional. En esos trabajos se encontró que predominan dos orientaciones de los SI: I) La orientación como artefacto TI [4] y II) La orientación a los negocios [18]. Este trabajo mejora esa clasificación de SI proponiendo una taxonomía de tres dimensiones de SI.

La organización del trabajo es la siguiente. En el punto 1 (Introducción) se detalla el estado del arte y la investigación personal y resultados obtenidos en la temática. En el punto 2 se describen las tres dimensiones en SI. En el punto 3 se enuncia los resultados, seguido por el punto 4 acerca de la discusión de la temática. La conclusión se expone en el punto 5.

2 Las Tres Dimensiones de los SI

A continuación se exponen tres dimensiones de SI que se resumen en la Tabla 1.

Dimensiones de SI			
Denominación	Procesamiento de Datos	Representación	Adaptación
Dominios Principales	Ciencias de la Computación, Inteligencia Artificial, Tecnologías de Información	Ciencias de la administración de organizaciones.	Sistemas auto-generados, auto-referenciados.
Objetivo	Incrementar la capacidad de manipulación de datos	Potenciar el modelamiento de la realidad	Viabilidad de los sistemas. Aplicable en ambientes ambiguos e impredecibles
Teorías Dominantes	Teoría Computacional y Teoría de la Información	Ninguna	Autopoiesis y Cibernética
Décadas principales de auge	1960 y 1970	1980 y 1990	2000 y 2010
Significado de Información	Es un dato	Es una señal de un objeto para un observador	Es una creación (síntesis) del observador para adaptar sistemas.
Observador	No existe	Es Pasivo	Es activo

Tabla 1: Principales Atributos de las tres dimensiones

2.1 Los SI como Procesamiento de Datos

Aquí los SI son vistos como procesos que manipulan datos. Esta fue la primera dimensión de los SI que surgió con la invención de la computadora. Fue casi la única forma de sistemas durante varias décadas.

Los datos son las entradas y salidas de procedimientos en mentes, máquinas u organizaciones. Por ejemplo, el envío de un registro de una transacción de compra a una base de datos implica una secuencia de procesos del registro de compra, validarlo, codificarlo, almacenarlo en la memoria y recuperarlo para el análisis posterior. En cada caso, un proceso manipula los símbolos; por ejemplo, cuando el “14 de

septiembre" se codifica, se convierte en "09/14" o a un equivalente binario u otra forma.

La ventaja de la dimensión de procesamiento de datos es su aplicación robusta y las metodologías de desarrollo en los últimos 50 años. Otras ventajas de esta dimensión incluyen su concisión y su capacidad para mejorar nuestra comprensión de los procesos humanos fundamentales, los computacionales y los organizacionales.

La limitación del procesamiento de datos es que no hace distinción entre los datos y la información. Por ejemplo, los profesionales de SI a menudo usan indistintamente el procesamiento de la información y el procesamiento de los datos. Para evitar confusiones con otros usos del término información, se recomienda el rotulo de datos, no información, para los símbolos de procesamiento. El rotulo de procesamiento de información es un nombre poco apropiado, y debería ser abandonado. Esto haría más claro que los símbolos están siendo manipulados por procesos, y que otros conceptos de información, tales como el significado, no están implicados. Por ejemplo, habría que decir que los datos pueden ser procesados y manipulados, resumidos y almacenados, pero que la información desde los datos debe surgir de alguna otra manera. Se debe decir que SQL procesa datos, y que la salida de una instrucción SQL sigue siendo datos y no información. El dato se transforma en información cuando el lector le asigna un significado.

Otra limitación del procesamiento de datos es que no hace distinción entre SI y TI. Existe una confusión entre SI y TI [10] y los SI están siendo remplazados por la TI [7]. La TI también es vista como procesos que usan datos como entradas y salidas.

2.2 Los SI como Representación

Los SI como representación son un modelo de algo para alguien. Una representación incluye una trilogía; una señal, un objeto y un observador. Por ejemplo, cuando las nubes oscuras (señal) auguran una tormenta (objeto) a un caminante (observador), las nubes oscuras son la información para el caminante. Del mismo modo, un analista (observador) examina un registro de una operación financiera (señal) de una compra (objeto); el registro es información acerca de la compra para el analista.

Las representaciones pueden ser externas o internas al observador. Un ejemplo de una representación externa es una puesta del sol (señal) que predice la venida de oscuridad (objeto). Un ejemplo de una representación interna es cuando Juan (objeto) actúa como un adolescente agresivo (señal) frente a su padre (observador). Las representaciones externas utilizan señales que se originan fuera del observador y son independientes de él, mientras que las representaciones internas tienen señales que se crean por el observador y por lo tanto dependen de él.

Los SI en la dimensión de representación modelan una realidad. La representación de la realidad puede ser idealista, escéptica o constructivista (Barry Smith [24], [25]). Las representaciones siempre simplifican la realidad, literalmente la "representa". Los SI como representación es la dimensión más ampliamente utilizada de los SI en los negocios, la ciencia de la computación, la psicología, etc. En esta dimensión, las representaciones más comunes son modelos conceptuales, tipologías y los diagramas de entidad- relación.

Para Weber [22], la representación es la única razón de ser de los sistemas de información. Sirven para hacer seguimiento de estados y cambios de estados en otros

sistemas. Por medio de observar el comportamiento de un SI, obviamos la necesidad de observar el comportamiento del sistema que él representa. La utilidad de la dimensión es evidente en el clásico ejemplo de un sistema de información contable. Este SI representa cuentas financieras para que los administradores no tengan que inspeccionar cada transacción. Por medio de no hacer el seguimiento de cada una de las transacciones, y en cambio hacer la gestión de la "representación" de transacciones, la empresa reduce los costos y aumenta la coordinación.

Debido a que es la dimensión más conocida, las implicaciones y las limitaciones de la dimensión de representación también son las más ampliamente desarrolladas.

Los profesionales de SI y los negocios a menudo tratan las señales externas y las internas como iguales. Al utilizar señales externas, las diferencias entre los observadores acerca de qué objeto la señal representa pueden conciliarse por la comunicación. Por ejemplo, un usuario de los estados financieros podría pensar que la línea 14 en un balance de situación es la suma de los números por encima de ella, mientras que otro usuario podría pensar que la línea 14 representa el promedio de los números por encima de ella. Estas representaciones diferentes pueden ser reconciliadas con la comunicación. En cambio, las representaciones sobre la base de señales internas, no pueden nunca reconciliarse ya que nadie tiene la representación "verdadera" u "objetiva" de la representación señal - objeto. Por ejemplo, un individuo conoce las ventajas (señales) de su diagrama E-R (objeto), pero estas ventajas no pueden conciliarse con lo que otros analistas creen que son las ventajas del diagrama. Ninguna comunicación reconcilia las representaciones basadas en las señales internas.

Una limitación de la dimensión de representación es que depende de la especificación de la señal, el objeto y el observador. Pero cuando los profesionales de SI crean una representación, el objeto y el observador suelen ser no especificados o implicados. Esta falta de especificidad con respeto a la señal, el objeto y el observador puede conducir a importantes malentendidos. Por ejemplo, se dice a menudo, "los números hablan por sí mismos." Sin embargo, los números (señales) pueden representar diferentes objetos para diferentes observadores. Este es un ejemplo de pensamiento que los números son la información y no los objetos que representan para un observador particular. Esta fijación de señales también puede conducir a equiparar las representaciones y los símbolos. Es común que los usuarios creen que pueden compartir información a través de los intercambios de símbolos. Sin embargo, una representación no se puede compartir o leer; sólo las señales (símbolos) pueden ser compartidas. El observador receptor luego colocará la señal a un objeto. Dado que los usuarios (observadores) sólo pueden intercambiar señales, a menudo se sorprenden cuando se comparten los resultados de datos en diferentes representaciones.

Otra limitación de la dimensión de representación es que no proporciona una base sólida para discriminar entre la representación verdadera y la representación falsa (distorsión). ¿Cuál es el grado de error de la diferencia entre una representación verdadera y una falsa? La Contabilidad ha desarrollado el concepto de materialidad para hacer frente a la distinción entre menores y mayores distorsiones, pero el campo SI no tiene tal concepto. El problema de la distorsión se vuelve mucho más difícil de resolver cuando están involucradas distorsiones internas. Es bastante fácil de

reconocer y corregir una mala representación externa a través de la observación y la comparación. Pero, ¿cómo puede un fenómeno interno abstracto como un plan a largo plazo o un nivel de seguridad de la red ser valuados como una falsedad? ¿Qué se puede observar o comparar para determinar si estas representaciones son distorsiones?

Otra limitación más de la dimensión de representación es que cada representación puede variar en maneras significativas, intratables, e inesperadas de individuo a individuo en base al carácter subjetivo de la experiencia y la interpretación [17]. Por ejemplo, tres modeladores de datos pueden generar diferentes modelos del mismo diagrama E-R debido a las diferentes experiencias e interpretaciones. O pensemos en una hoja de balance. Cada usuario de la hoja de balance accede a diferentes experiencias para interpretar cómo los datos fueron desarrollados, medidos, o combinados, y lo que significan. Estas diferencias individuales se convierten en un problema significativo cuando los analistas de TI subestiman lo diferente que son los usuarios informados de lo que parece ser la misma representación, ya sea que se trate de un formulario, un informe, un diagrama o un gráfico. Este aspecto de la intersubjetividad es particularmente importante para entender por qué los SI son confundidos y reemplazados por la TI [10]. El desafío de la disciplina SI es la entrega de servicios efectivos a los usuarios con una amplia variedad de experiencia que enfrente los efectos de la intersubjetividad del usuario, que obviamente no son considerados en la TI.

Otra fuente persistente de la intersubjetividad, en relación con la experiencia y la interpretación, es el conocimiento, un tema de creciente interés en SI. Cada observador trae una configuración diferente de conocimiento para comprender una representación. Por ejemplo, la representación de un eclipse solar (señal) es información acerca de la luna en relación con el sol (objetos), pero sólo para los observadores que tengan conocimiento de una teoría astronómica que conocen a objetos como la luna y el sol, y conocen sus configuraciones posibles. Del mismo modo, la representación que el ingreso neto es de 100.000 pesos requiere el conocimiento de la contabilidad y la representación puede significar algo diferente para el contador y el accionista por sus diferencias en el conocimiento. La representación de un diagrama de proceso de la cadena de suministro requiere conocimiento de modelado de procesos y de la cadena de suministro que produce información diferente para los profesionales en SI y usuarios. Esta mirada de las representaciones sugiere a la comunidad de SI cambiar su visión tradicional que la información conduce al conocimiento. Desde la dimensión de la representación, el conocimiento conduce a la información.

2.3 Los SI como Adaptación

Los SI como adaptación usan la información para adaptación [17]. Según McKinney la dimensión de adaptación requiere de un cambio de paradigma ontológico, pasar de la creencia de que existe una realidad objetiva independiente de la percepción, al reconocimiento de que la realidad depende de la percepción [24] y [25].

La información en la dimensión de la adaptación viene de la percepción. La información es cualquier "diferencia que hace la diferencia" percibida por un sujeto. Los sujetos de esta dimensión de adaptación se conocen como sistemas. Un sistema puede ser un mecanismo, un organismo, o una organización, tal como una máquina, una mente, o una empresa. Si una diferencia percibida se traduce en una adaptación de un

sistema, entonces existe información. Por ejemplo, si un conductor percibe un exceso de velocidad (una diferencia percibida existe entre la velocidad actual y la velocidad límite), y esta diferencia percibida hace una diferencia para el conductor, el conductor cambia a una velocidad diferente (más lenta), entonces se adapta. La información no es la diferencia percibida entre la velocidad y la velocidad límite, o el cambio a la velocidad diferente; más bien, información es la síntesis de las dos diferencias, es una creación. Un segundo ejemplo proviene de un contexto de negocios. Si una empresa percibe disminución de las ventas (una diferencia), y la percepción de la disminución de las ventas marca la diferencia para la empresa, entonces hay información. De lo contrario, si el individuo o la empresa no percibe, o la percepción no hace ninguna diferencia para el individuo o la empresa, no hay información.

En la dimensión de la adaptación, el individuo nunca puede estar completamente informado, ya que muchas diferencias pasan desapercibidas. Además, cada individuo sólo está informando sobre lo que él está actualmente notando. Una vez que una persona saca su atención de una diferencia que hace la diferencia, el individuo ya no es un ser informado. Estar informado es creativo y dinámico; toda la información en la dimensión de la adaptación es fabricada por el individuo. Por lo tanto, la información en la dimensión de la adaptación depende por completo de la percepción del individuo y lo que el individuo haga con su percepción, si es que hace algo.

La dimensión de adaptación de la información se sustenta en varias teorías de sistemas: entre las más importantes la autopoiesis, la cibernética de segundo orden y la auto-organización.

Autopoiesis significa literalmente "auto-creación" [16]. El término describe la naturaleza de los sistemas vivientes. Los sistemas autopoiéticos son una red de procesos que producen los componentes que regeneran continuamente la red de procesos que los producen. Los sistemas autopoiéticos incluyen colmenas, seres humanos y sociedades. La autopoiesis ve a la adaptación del sistema como una regeneración, o perpetua auto recreación. La información para un sistema autopoiético no es cómo el sistema percibe el mundo, y no lo que el sistema sabe de sí mismo, sino más bien cómo el sistema se recrea y se adapta sobre la base de las diferencias que percibe.

La cibernética es el estudio de la autorregulación de los sistemas. Cada sistema cibernético se esfuerza por seguir siendo viable (sostener la existencia) mediante el control de los parámetros del sistema dentro de los límites de tolerancia frente a las perturbaciones del medio que inciden en la sistema. Por ejemplo, el sistema termostático en un edificio se esfuerza para mantener la temperatura interior a un nivel constante por medio de iniciar calentamiento o enfriamiento, en base a la temperatura detectada en el medio ambiente. Los seres humanos y las organizaciones se auto regulan, adaptan sus variables al medio ambiente para seguir siendo viables.

Por último la teoría de la auto-organización de las organizaciones expresa que las organizaciones del futuro serán auto-organizadas más bien que diseñadas. Algunas de las discusiones más estimulantes acerca de la forma de la organización futura son aquellas centradas en la teoría del caos, ecología y biología. El libro de gran influencia de Wheatley [23], Liderazgo y la Ciencia Nueva, es un ejemplo. Los otros dos de Kevin Kelly que abren la mente son Fuera de Control [12] y las Nuevas Reglas [13] para la Nueva Economía. El principio básico

de esos trabajos es que la naturaleza provee un buen modelo para las organizaciones futuras; esto es, las organizaciones que deben tratar con complejidad, compartir información y conocimiento y enfrentar los cambios continuos y discontinuos. De acuerdo a ambos, en un caos aparente, podemos conseguir orden para libertad. Como con un fenómeno natural, las empresas lo harán mucho mejor si son autoorganizadas o emergentes, más bien que diseñadas. Los libros dan muchos ejemplos de estos principios.

Una primer ventaja de la dimensión de adaptación es el de generalizar, ya que la mayoría de los fenómenos más interesantes se pueden interpretar como la adaptación de sistemas. Además, la adaptación es el único medio de hacer frente a los ambientes ambiguos e impredecibles. Si los problemas no pueden ser claramente especificados, y el cambio es rápido e impredecible, las estrategias de modelamiento ortodoxo (representación) se ven abrumadas ya que el problema cambia mientras se hace la representación.

El apoyo en SI para los usuarios que se enfrentan a esta ambigüedad y la imprevisibilidad ya ha empezado a cambiar los SI. Por ejemplo, el software de código abierto y las redes sociales están diseñados para ser flexibles y adaptativos, cambian a medida que cambian las necesidades del usuario. Esto contrasta con el énfasis histórico de SI en las representaciones estáticas, tales como formularios, informes y gráficos. Tales representaciones estáticas tienden a bloquear las organizaciones en estados particulares, con sistemas de TI que son lentos para adaptarse a los cambios del usuario [20].

Una segunda ventaja de la dimensión de adaptación es que expande el dominio de los SI, porque se aplica a todos los sistemas, no sólo al sistema humano-observador de la dimensión de la representación. Los SI como adaptación generaliza a sistemas dentro y fuera del contexto tradicional de negocio y organizacional. Todo sistema crea y utiliza la información para la adaptación. Mientras que el negocio y las organizaciones son sistemas en permanente adaptación, también lo son las bases de datos, los individuos, los equipos, las industrias y las sociedades. Cualquier sistema se puede entender con la dimensión de la adaptación. La dimensión de adaptación no solo permite la investigación sobre muchos más tópicos que los de negocio y organización, sino que también apoya la investigación en SI en sistemas colectivos emergentes con tecnología única tales como Facebook, Wikipedia, y otros sitios Web 2.0.

Una tercera ventaja de la dimensión de adaptación es el conocimiento. Las suposiciones subjetivas que sustentan la dimensión de la adaptación han producido un considerable conocimiento de la filosofía, sobre todo la filosofía de la ciencia física. El conocimiento que resulta de aplicar esta dimensión a los temas de SI puede ser importante, como sucede en el siguiente ejemplo que analiza la adopción del usuario de nuevos SI desde la dimensión de la adaptación.

La vista tradicional de la adopción de la tecnología sugiere que la aceptación del usuario se debe a que ellos encuentran que un nuevo SI tiene información más útil que el sistema anterior. Un desarrollador crea un nuevo sistema y cree que tiene más información que el antiguo sistema, debido a que el nuevo sistema incluye más símbolos, mejor sintaxis, o puede representar más eventos. Sin embargo, los usuarios tienen dificultades para encontrar la información en el nuevo sistema de TI. De acuerdo con la dimensión de adaptación, la lucha es inevitable ya que los usuarios están buscando

información en el sistema TI en vez de darse cuenta de que debe ser creada por ellos mismos. Los usuarios tampoco se dan cuenta de que el sistema TI viejo nunca ha proporcionado información, más bien, han aprendido a crearla a partir de los datos obtenidos del sistema de TI. Entonces, los usuarios se sienten decepcionados porque el nuevo sistema de TI no tiene su información, o que parece complicarla innecesariamente. Los desarrolladores no pueden ofrecer a los usuarios lo que quieren, lo que los usuarios erróneamente piensan que es posible.

Una segunda fuente de resistencia a la adopción es la percepción por los usuarios de que un nuevo sistema TI carece de flexibilidad. Cuando un nuevo sistema TI está instalado, los usuarios comienzan de inmediato a crear nueva información de los datos de ese sistema. Los usuarios pueden estar confundidos y decepcionados con el nuevo sistema TI y sus desarrolladores, que no facilitaron la nueva información (como ellos creían) en un formato más conveniente, fácil de usar. Los desarrolladores frustrados llaman a esto la crisis del alcance, pero esto en realidad es una señal saludable y predecible de la adaptación de un sistema. Los usuarios quieren un sistema flexible que los mantendrá al día con sus necesidades de adaptación; los desarrolladores, contando con más visiones tradicionales, creen que se les paga para construir sistemas estáticos que satisfacen un conjunto fijo de requisitos de representación.

Otra ventaja importante de la dimensión de la adaptación es que proporciona un nuevo fundamento teórico para combatir la percepción de que los SI se están convirtiendo en una mercancía (commodities) [5]. Mientras los símbolos, los datos y la tecnología pueden ser commodities, la dimensión de adaptación sugiere que la información es una creación única de cada sujeto en función de sus necesidades adaptativas. La dimensión de adaptación ve la información no como un commodity omnipresente con una interpretación común, si no, como una clara creación por un usuario en particular basada en experiencia, la interpretación y el conocimiento [20]. La información es la fuente única de cambio para cada uno y todos los sistemas, no una mercancía universal indiferenciada. Como tal, la información juega un papel clave en la explicación de cómo los usuarios toman sentido, interpretan y crean información en sus entornos, tres temas con posibilidades de investigación significativas para los SI.

3 Resultados

El principal resultado de este trabajo es una taxonomía de SI. La taxonomía consiste de tres dimensiones en SI: el procesamiento de datos, la representación y la adaptación.

Esta taxonomía aporta un marco de referencia importante a la identidad de la disciplina ya que ayuda a clasificar los escasos fundamentos conceptuales de la disciplina [4] y [18]. También permite colaborar con la formulación de curriculas [8] [19]. Los SI como procesamiento de datos están muy vinculados con la ciencia de la computación, como representación están muy vinculados a la ciencia de la administración y como adaptación a la cibernética y la biología.

4 Discusión

Es sorprendente que exista tan poca investigación en aspectos teóricos de SI, y que la mayoría del trabajo en la disciplina y las curriculas están orientados hacia aspectos tecnológicos y metodológicos [6], [15] y [21]. Numerosos autores han señalado esta situación y en la necesidad de teorizar [3] y [11].

Los artículos teóricos más relevantes tenidos en cuenta en este trabajo son: [3], [4], [17], [18] y [22].

El objetivo del trabajo es convocar a futuras investigaciones sobre una serie de temas relacionados con la disciplina. Entre ellos tenemos: ¿Cuál es la identidad de la disciplina [2]?, ¿Es el campo de SI una ciencia como lo plantea Kuhn [14]?, ¿Cuáles son las teorías fundamentales [11]?

La acelerada evolución de los SI explica mucho de la crisis. Las dimensiones presentadas están relacionadas con la evolución y por lo tanto disminuye la crisis. Además de la evolución en el tiempo, existe una relación de inclusión: la dimensión de representación incluye a la dimensión de datos y la de adaptación incluye a la de representación.

Por último, la dimensión de adaptación ofrece una ventaja significativa. En el procesamiento de datos y la representación la información es lineal y finita, pero en la dimensión de adaptación, el número de creaciones es exponencial e infinito. El potencial de la información de la adaptación, que permiten a los sistemas un éxito (ser viable) en ambientes ambiguos e impredecibles, hunde a las otras dos dimensiones.

Así, en un mundo caracterizado por la imprevisibilidad y ambigüedad, la dimensión de adaptación permite más sistemas para sobrevivir y prosperar. El punto de vista de adaptación admite el mejor de los mundos de los SI, por ser capaz de utilizar las otras dimensiones y generalizando a muchos más eventos.

5 Conclusión

Existe un debate actual, que tuvo su auge en el periodo 2003-2006, acerca de una crisis de identidad en la disciplina de SI [2], [3], [9] y [22]. Existen tres aspectos en los que se manifiesta la crisis: el académico, el organizacional y el de los desarrolladores de SI. Se han realizados trabajos de investigación en concordancia con dicha crisis [10], [15] y [21].

Se expone una taxonomía de SI que ayuda a entender dicha disciplina y colabora para la formulación curricular. La taxonomía es de tres dimensiones en SI que son: el procesamiento de datos, la representación y la adaptación (Tabla 1).

Las dimensiones están relacionadas e incluyen aspectos evolutivos y conceptuales de los SI. La palabra dimensión no implica ortogonalidad, más bien existe una inclusión y una creciente complejidad en los SI según la jerarquía de sistemas establecida por Kenneth Boulding [26].

Se remarca la necesidad de teorizar en SI y se presentan varios temas de investigación significativos. Dicho trabajo futuro de investigación estará guiado principalmente por la dimensión de adaptación.

Referencias

1. ACM, AIS, 2010, IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems.

2. Bakshi, Sumit, 2006, “Crisis in the Information Systems discipline: A reflection”. SSRN Social Science Research Network, June 2006. Available at SSRN: http://ssrn.com/abstract=921787

3. Banville, Claude, and Landry, Maurice, 1989, Can the Field of MIS be Disciplined? Communication of the ACM, (32:1), pp. 48 – 60.

4. Benbasat, Izak, and Zmud, Robert W., 2003, The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline’s Core Properties, MIS Quarterly, (27:2), pp. 183 -194

5. Carr, Nicholas George, 2003, IT Doesn´t Matter, Harvard Business Review. May 2003.

6. Ciborra, Claudio U., 1998, "Crisis and foundations: an inquiry into the nature and limits of models and methods in the information systems discipline", Journal of Strategic Information Systems, Volume 7, pp. 5-16, March 1998.
Available at:
http://www.sciencedirect.com/science/article/pii/S0963868798000201

7. García, Sebastián José, 2011, Restricciones para el Crecimiento de los SI, SABTIC 2011, Rosario, Argentina.

8. García, Sebastián José, 2012a, Crisis Curricular en los SI, TE&ET 2012, Argentina.

9. García, Sebastián José y otros, 2012b, Crisis en la Disciplina SI, SABTIC 2012, Brasil.

10. García, Sebastián José y otros, 2013, La Confusión entre los SI y las TI, Editorial Giménez.

11. Gregor, Shirley, 2006, "The Nature of Theory in IS", Australian National University, Canberra, MIS Quarterly Vol. 30 No. 3, pp. 611-642/ September 2006.

12. Kelly, Kevin, 1994, Out of Control: The New Biology of Machines, Social Systems, and the Economic World, Addison-Wesley Publishing, Reading, MA.

13. Kelly, Kevin, 1998, New Rules for the New Economy: Ten Radical Strategies for a Connected World, Viking, New York.

14. Kuhn, Thomas S., 1962, The Structure of Scientific Revolutions, first edition, University of Chicago Press, Chicago.

15. Martínez, María Fernanda, 2012, Análisis de la Confusión de los SI con la TI en la Organización, Tesis de grado para obtener el Título de Licenciado en Sistemas de Información en la Universidad Nacional de San Juan, Biblioteca de la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de San Juan.

16. Maturana, Humberto, Varela, Francisco, 1980, Autopoiesis and Cognition: The Realization of the Living, Boston: Reidel.

17. McKinney Jr, Earl H., and Yoos II, C. J., 2010, Information about Information: A Taxonomy of Views, MIS Quarterly (34:2), pp 329-344.

18. Olave Cáceres, Yesid A. y Gómez Flórez Luis C., 2007, Una Reflexión Sistémica Sobre los Fundamentos Conceptuales para los Sistemas de Información. Revista Colombiana de Computación, vol. 8 Nro. 1, Colombia.

19. Resolución 786/2009 del Ministerio de Educación publicada en el Boletín Oficial el día 4 de junio del 2009.

20. Tuomi, I. 1999. "Data is More than Knowledge: Implications of the Reversed Knowledge Hierarchy for Knowledge Management and Organizational Memory," Journal of Management Information Systems (16:3), pp. 103-117.

21. Villavicencio, María V., 2013, Los Fundamentos Conceptuales de la Licenciatura en Sistemas de Información de la Universidad Nacional de San Juan, Tesis de grado para obtener el Título de Licenciado en Sistemas de Información en la Universidad Nacional de San Juan, Biblioteca de la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de San Juan.

22. Weber, Ron, 2003, Editor's Comments: still desperately seeking the IT artifact, MIS Quarterly, (27:2), pp. iii – xi.

23. Wheatley, Margaret, 1994, Leadership and the New Science: Learning About Organizations from an Orderly Universe, Berrett-Koehler Publishers, San Francisco.

24. Smith, Barry, 2004, Beyond Concepts: Ontology as Reality Representation, Proceedings of FOIS 2004, Turin- 4-6 November 2004.

25. Smith, Barry, Kusnierczyk, Waclaw, Schober, Daniel, Ceusters, Werner, 2006, Towards a Reference Terminology for Ontology Research Development in the Biomedical Domain, KR-MED 2006 "Biomedical Ontology in Action", November 8, 2006, Baltimore, MD, CEUR, vol. 222, 57-65.

26. Boulding, Kenneth E., 1956, General Systems Theory: The Skeleton of Science, Management Science, Vol. 2 no. 3 April 1956.

Desarrollo de un Agente Inteligente Basado en el Estándar ANSI/ISA-95

Melina C. Vidoni, Aldo R. Vecchietti
Instituto de Desarrollo y Diseño, Ingar UTN-CONICET
Santa Fe, Argentina
melinavidoni@santafe-conicet.gov.ar, aldovec@santafe-conicet.gov.ar

Abstract

Los cambios en las organizaciones y la búsqueda de la integración, ha generado una necesidad de estandarizar las estructuras de datos empleadas para compartir información, con el objetivo de aumentar la eficiencia del flujo de información. El estándar ANSI/ISA-95 ha cobrado gran relevancia como un medio para la estandarización y automatización de sistemas empresariales, así como también debido a la estructura de información de manufactura que propone y define en la primera y tercera parte del mismo. Este trabajo propone a GrACED, un agente inteligente basado en conocimiento, que procesa lenguaje natural mediante bolsas de palabras, para analizar y clasificar la estructura de las tablas de las base de datos de los ERP, en las categorías propuestas por el estándar ANSI/ISA-95. El objetivo de GrACED y la propuesta de este trabajo es promover un medio, adaptable, portable y con bases estandarizadas, para analizar de forma automatizada la información que puede contener cada tabla en la base de datos, como así también estudiar la adecuación de dicho ERP al estándar, para lograr el objetivo último de facilitar el estudio del sistema empresarial, favoreciendo su integración con otros sistemas.

1. Introducción

Un importante desafío para las organizaciones es el cambio de sus entornos, lo que implica una alta necesidad de flexibilidad, agilidad, eficiencia y calidad en sus procesos. Debido a esto, la Comisión Europea [1] recomendó la mejora de los procesos de integración a través de su estandarización y posterior automatización. La toma de decisiones integradas y la optimización colaborativa dentro de las empresas, pasó a tener un rol crucial en la interrelación de organizaciones. Con este objetivo en mente, se han desarrollado sistemas tipo MES (*Manufacturing Execution Systems*) o CPM (*Collaborative Production Management*) con un único objetivo en mente: anular la brecha entre los procesos, las comunicaciones y los sistemas ERP (*Enterprise Resource Planning*) [2].

Para alcanzar esta integración, es imperativo definir estructuras de información y herramientas sofisticadas que permitan explotar dichas configuraciones, con el objetivo de mejorar la disponibilidad y comunicación de los datos, más específicamente de manufactura, si lo que se desea es integrar cadenas de suministro. Siguiendo esta línea, se han propuesto muchos estándares para mejorar la eficiencia y el flujo de la información de manufactura, entre ellos el ANSI/ISA-95 [3].

ANSI/ISA-95 es un estándar internacional para desarrollar interfaces automatizadas entre empresas y sistemas de control, que propone un conjunto de modelos y definiciones fundadas en una terminología consistente, para describir las tareas e información de manufactura y producción que deben ser intercambiadas en sistemas que se interrelacionan [4]. En los últimos años, este estándar ha sido ampliamente aceptado, debido a que especifica un modelo funcional completo [5].

Se han realizado varios trabajos académicos para favorecer el intercambio de información estandarizada según los modelos del ANSI/ISA-95, así como diferentes formas de implementación. En 2009 [2] varios autores propusieron una plataforma para el intercambio de información utilizando diagramas BPMN (*Business Process Model and Notation*) basados en los modelos del ANSI/ISA-95. También se han efectuado avances en el área de simulación, con el objetivo de generar especificaciones para desarrollar sistemas uniformes [6]. Otros autores [3] han empleado ontologías para generar un *framework* que integra la toma de decisiones, utilizando las estructuras del ANSI/ISA-88. Finalmente, He y otros (2012) [7] realizaron una herramienta para el modelado de empresas, con fundamentos en el ISA-95 y en el IEC 62246.

Sin embargo, si bien muchos estudios se enfocan en diseñar nuevos sistemas y herramientas basados en el ANSI/ISA-95 [8], muy pocos intentan analizar los ya existentes y proveer un informe sobre su adecuación al estándar, o estudiar qué tipo de información de manufactura contienen, a la luz de la clasificación propuesta en el ANSI/ISA-95. La realización de esto favorecería la integración entre sistemas sin obligar a las empresas a cambiar radicalmente su forma de trabajar, como así también proveer un marco para el análisis de los mismos y posibles formas de modificarlos -con el objetivo de adecuarse al ANSI/ISA-95.

A su vez, en un desarrollo previo [9], los autores generaron un prototipo de un sistema tipo APS (*Advanced Planning and Scheduling*) y encontraron la necesidad de presentar al usuario alguna indicación sobre la ubicación de la información de manufactura en la base de datos del ERP, con el objetivo de enlazar los modelos matemáticos del APS, con el ERP empleado.

Utilizando esta idea como disparador inicial, la propuesta detallada en este artículo es utilizar un agente inteligente, cuya base de conocimiento esté dada por el ANSI/ISA-95, y que pueda clasificar el contenido de una base de datos de un ERP o sistema empresarial, en cada una de las categorías que el estándar propone, empleando el enfoque *bag of words* (o bolsas de palabras). Una de las fortalezas de utilizar un agente inteligente, es la capacidad inherente del mismo de procesar el lenguaje natural.

En esta línea, también se han realizado proyectos sobre categorización de textos o estructuras, utilizando agentes inteligentes. En uno de ellos, se ha propuesto una clasificación sobre fuentes de generación de gas, utilizando algoritmos genéticos y redes neuronales para posteriormente compararlos [10]. Una investigación relevante generó un método de clasificar documentos de texto de forma automatizada, usando un conglomerado de múltiples agentes que procesaban lenguaje natural; en este enfoque, cada agente sólo catalogaba en una sola categoría [11]. Finalmente, la idea de bolsas de palabras también ha sido empleada, a través de un modelo bayesiano, para la generación de documentos de texto usando agentes inteligentes [12].

Cabe destacar que hasta el momento no se han encontrado trabajos que empleen agentes inteligentes para analizar sistemas existentes a la luz de los conceptos propuestos por el estándar ANSI/ISA-95.

2. ANSI/ISA-95: Base de Conocimiento

Siguiendo la definición de Russel y Norvig [13], un agente inteligente es una entidad autónoma inserta en un ambiente, que percibe lo que sucede en él a través de percepciones (realizadas mediante sensores), y responde a ellas actuando de forma racional, a través de acciones ejecutadas por actuadores. En este punto es importante mencionar que no todos los agentes inteligentes aprenden y actualizan su conocimiento de forma automática; ellos son una especificación de la definición presentada, y se conocen como *learning agents*.

Hay muchos tipos de agentes, entre ellos los *knowledge based*. Estos especializan la definición presentada, ya que poseen una representación del conocimiento y un proceso de razonamiento que lo ejecuta y puede combinarlo con las percepciones del estado actual, antes de seleccionar acciones. Este tipo tiene muchas variantes, ya que puede denotarse como basado en metas, puede o no interactuar con otros agentes, y puede desarrollarse con o sin aprendizaje.

Estos agentes son muy utilizados en el procesamiento de lenguaje natural, dado que su comprensión radica en inferir los estados ocultos, es decir, la semántica detrás de las palabras.

Respecto a los ERP, la persistencia de la información de estos sistemas se realiza en sus bases de datos, las cuales son en su mayoría de tipo relacional. De esta forma, si se quiere analizar cómo se organiza la información en un ERP, es necesario analizar y clasificar la estructuración de los datos en las tablas de su base de datos.

Para poder categorizar, es imperativo tener categorías definidas y estandarizadas, que tengan una aceptación moderada a amplia, y que formen parte de la base de conocimiento (BC) del agente inteligente.

Esto mismo fue lo que llevó a la utilización y aplicación del ANSI/ISA-95, el cual propone de modelos y definiciones consistentes para generalizar la estructura y nombramiento de las tareas e información de manufactura y producción [4]. Más específicamente, en la Parte 3 [14] propone clasificar la información de manufactura en cuatro categorías que definen la información de productos y de producción, las cuales pueden observarse en la Figura 1[1].

De estas categorías, se decidió trabajar sólo con *Product Definition*, *Production Capability* y *Production Schedule*, dado que su composición es más fácil para detectar con el acercamiento propuesto en el trabajo. Esto es consecuencia de que la información de la categoría *Production Response* es a menudo representada sólo en atributos de las tablas y no en tablas completas, lo que

[1] *Es importante mencionar que las categorías se han trabajado en inglés, con el objetivo de mantener los nombres originales del estándar, mientras que las descripciones y otros conceptos serán presentados en español, para una mayor claridad.*

aumentaría la complejidad de la clasificación.

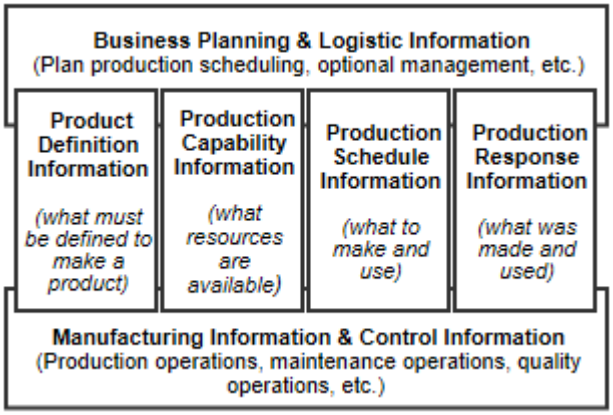


Figura 1.*Categorías de información de ANSI/ISA-95.*

Por otro lado, la Parte I del estándar [4] es la que genera todas las definiciones y conceptos que son posteriormente utilizados para generar la estructura y definir las categorías en las que se va a clasificar. Como parte de estas definiciones, el estándar propone gráficos que serán denominados *de superposición*, que no sólo explicitan subcategorías de información para cada categoría de la Fig. 1, sino que también definen las subcategorías, y cómo se superponen entre ellas.

En la Figura 2 se observa el gráfico de superposición para la categoría *Product Definition*, que es la que posteriormente se ha utilizado en los casos de estudio, con el objetivo de acotar las primeras evaluaciones del agente.

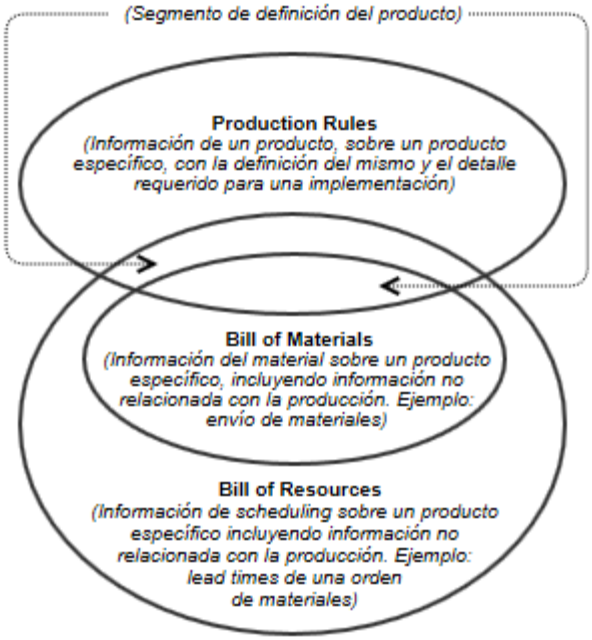


Figura 2.*Superposición de información en la Definición del Producto, para ISA-95 Parte I.*

2.1. Generación de Categorías

De esta forma, para poder generar la BC, el primer paso fue estructurar las categorías y subcategorías de información de manufactura presentadas en el ANSI/ISA-95 en un grafo, el cual puede observarse en la

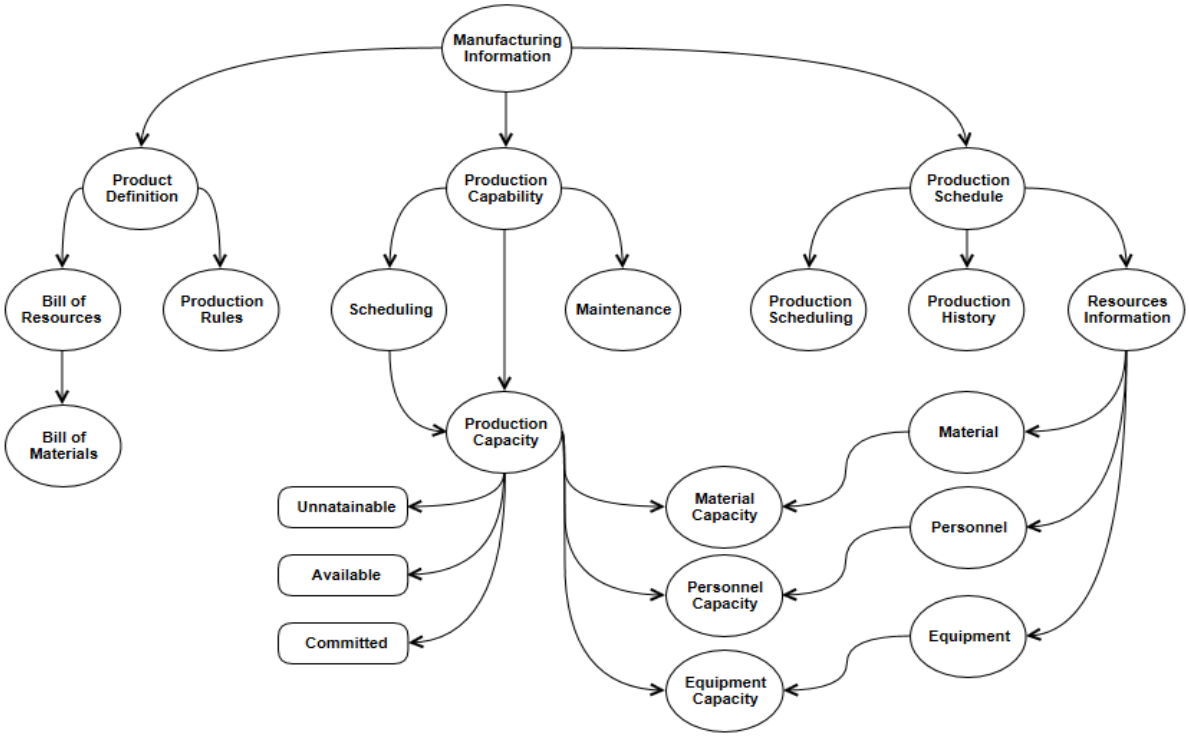


Figura 3.*Grafo de Categorías derivado del ANSI/ISA-95 Parte I.*

Figura 3. Los nodos ovalados representan las categorías (con sus nombres originales en inglés) que pueden utilizarse para clasificar, mientras que los rectangulares de bordes redondeados son presentados en el estándar, pero no se van a emplear en la clasificación.

El nodo raíz representa a la totalidad de información de manufactura, mientras que los nodos de nivel 1 son las grandes categorías de la Fig. 1 (sin usar a *Production Response*, como se mencionó previamente). Por otro lado, los nodos de niveles sucesivos e inferiores fueron obtenidos de los gráficos de superposición (por ejemplo, las categorías visualizadas en la Fig. 2, son representadas como los hijos del nodo *Product Definition* en el grafo de la Fig. 3 y de las descripciones de cada categoría).

2.2. Bolsas de Palabras

El siguiente paso para generar la BC, fue obtener un enfoque que permitiera emparejar las tablas de la base de datos del ERP, con una o más categorías de las presentadas en el grafo. Más allá del Sistema de Gestión de Base de Datos (SGBD) que se emplee, tanto las tablas como las columnas tienen nombres que las identifican, los cuales -generalmente- se eligen para darle significado semántico al contenido que almacenan.

Dado que los ERP son generalmente implementados con BD relacionales, la extracción de las palabras a partir de la definición de sus tablas resulta un enfoque apropiado, dado que dichas estructuras no son propensas a cambiar y fluctuar en el tiempo.

Por esto mismo, se decidió trabajar con lenguaje natural, clasificando las tablas por las palabras que la definen.

El enfoque utilizado es *bag of words (BoW)* o bolsas de palabras. En este enfoque, las bolsas son una representación simplificada utilizada en el procesamiento de lenguaje natural, donde cada clase o documento se representa en un *multi-set* (o bolsa) de palabras, sin considerar la gramática (formación de sentencias) ni el orden de las palabras [12].

En este momento es importante mencionar que el estándar sólo define las categorías, indicando qué tipo de información se incluye, pero sin proporcionar las palabras para formar las BoW. La generación de las BoW fue parte del desarrollo de este proyecto, y será precisado en la siguiente sección.

Sin embargo, no todas las palabras tienen la misma relevancia, la cual incluso puede variar de categoría en categoría; por esto mismo, se decidió asignar pesos a las palabras. Cada bolsa tiene un peso total de 100, el cual fue dividido internamente entre las palabras, dando un peso mayor a las que son más representativas. Además, estos pesos dependen de la categoría que representan.

Un punto importante para mencionar es que, muy a menudo, las palabras que se utilizan para definir nombres de tablas no suelen ser las mismas que se emplean en los nombres de las columnas, aun cuando pertenezcan a la misma categoría. Debido a esto, se decidió asociar dos bolsas de palabras por categoría (o nodo ovalado en el grafo de la Fig. 3): una para las palabras en los nombres de las tablas, y otra para las columnas.

Finalmente, para completar la BC era necesario considerar el uso de sinónimos (palabras escritas de diferente forma pero que significan lo mismo) o abreviaturas (convenciones ortográficas que acortan la escritura de cierto término o expresión) al momento de nombrar las tablas o columnas. Agregar cada combinación para cada palabra a las BoW no es conveniente, ya que no sólo introduce redundancia y aumenta el tiempo de procesamiento, sino que también reduciría los pesos de las palabras dentro de las bolsas; esta redundancia también impactar en el porcentaje de pertenencia final de una tabla a una categoría.

Como consecuencia de la riqueza de los lenguajes naturales, las palabras pueden tener distintos significados, sinónimos y abreviaturas. Estas variaciones dependen de la categoría en la que está siendo clasificada la palabra (por ejemplo, tanto `product` como `production` puede ser abreviado como `prod`). Por esto, se agregaron archivos exclusivos de sinónimos y abreviaturas que fueron relacionados directamente a cada palabra en cada BoW. Estos archivos serán directamente nombrados como *Synonyms Files*, incluso si contienen abreviaturas.

Cabe destacar que, a pesar de la mencionada riqueza de los lenguajes, la cantidad de sinónimos y abreviaturas permanece dentro de un rango manejable para el agente, por varios motivos. Uno de ellos, es que la mayoría de los desarrollos de gran envergadura, tales como los ERP, son codificados en inglés (nombres de variables, métodos, tablas, clases, etc.), el cual es justamente el idioma principal del agente. El otro motivo es que al programar sólo se utilizan sinónimos o abreviaturas convencionales *de facto*, lo que disminuye la cantidad de sinónimos existentes.

2.3. Implementación de la Base de Conocimiento

Como se explicó anteriormente, se asociaron dos BoW por cada nodo del grafo de la Fig. 3. Por esto mismo, se decidió almacenar dicho grafo como el índice de la BC del agente, que contiene las referencias a las categorías y a las bolsas de palabras, pero manteniendo las relaciones de niveles. Esta implementación se realizó utilizando archivos XML [15], y siguiendo la estructura presentada en la Figura 4:

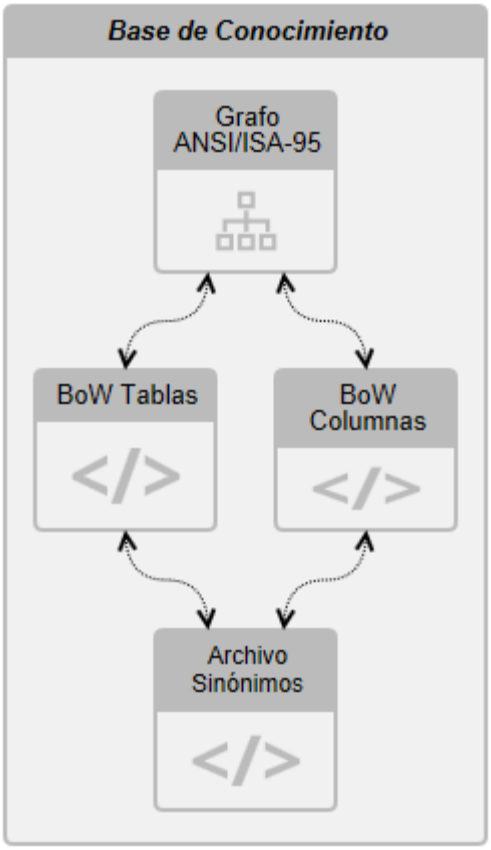


Figura 4.*Estructura de la base de conocimiento propuesta, basada en el ANSI/ISA-95.*

De este modo, debido a que el grafo se guarda como un archivo XML, cada nodo es un elemento dentro del mismo, el cual tiene atributos para el nombre, y para el nombre de archivo de cada bolsa de palabra, diferenciando el uso de cada una. En la Fig. 5 puede observarse un extracto del código XML que representa al nodo *Bill Of Materials*, donde los atributos `columnameBow` y `tablenameBow` son los que guardan el nombre de archivo de las respectivas BoW.

El procedimiento empleado para generar las bolsas de palabras y los archivos de sinónimos fue manual, y se describe a continuación:

1. Se listaron los nombres de tablas y las columnas de cada ERP.
2. Para cada ERP:
    a. Se clasificó manualmente cada tabla, considerando las descripciones de contenido del estándar ANSI/ISA-95.
    b. Se separaron las palabras que conformaban cada nombre de tabla y las de los nombres de columnas. Por ejemplo, el nombre de tabla `stock_inventory_move`, se transformó en tres palabras: `stock`, `inventory` y `move`.
3. Manteniendo la distinción del origen de las palabras (es decir, si eran de los nombres de tablas o de los nombres de columnas) se agruparon todas las palabras de cada categoría (todas las pertenecientes a *Bill of Materials*, las pertenecientes a *Production Rules*, etc.).
4. Para cada grupo de palabras:
    a. Se contó la cantidad de veces que aparecía cada palabra, para obtener la "relevancia" o "nivel de descripción" que aporta la misma para una categoría.
    b. Separadamente, se anotaron cada palabra y los sinónimos de la misma.
    c. Se sumó la cantidad de apariciones de la palabra y sus sinónimos.
    d. Dándole un peso total de 100 a cada BoW, se otorgó un peso a cada palabra, considerando la cantidad de apariciones encontrada en el punto anterior.

3. GrACED: Agente Inteligente

Basándose en la definición de Russel y Norvig presentada al inicio de la sección 2, se propuso un agente inteligente denominado GrACED (por las siglas en inglés de *Grammar Agent for Classifying ERP Databases*). Siguiendo los componentes mencionados endicha definición, en la Figura 6 puede observarse la estructura básica de la propuesta.

De este modo, GrACED representa al agente inteligente propuesto que se encuentra inserto en un ambiente, representado por el ERP que se desea analizar. Este ambiente posee un estado compuesto por los datos de conexión a la base de datos del ERP y la lista de los nombres de las tablas existentes en dicha base, que son las que se van a analizar y clasificar.

Por otro lado, GrACED tiene dos percepciones relacionadas entre sí: obtener el siguiente nombre de tabla a analizar, y luego los nombres de las columnas de dicha tabla. Estas percepciones son almacenadas en el estado del agente, mientras se está ejecutando la única acción que posee: Clasificar. Los otros componentes del estado del agente lo enlazan a la base de conocimiento (la

```
<!--BILL OF MATERIALS NODE-->
<tns:node tns:nodeName="Bill Of Materials" tns:columnameBow="bom_col.xml"
tns:tablenameBow="bom_tab.xml" tns:usable="true">
    <tns:relation tns:relationName="partOf"/>
</tns:node>
```

Figura 5.*Extracto de código XML para el grafo ANSI/ISA-95.*

cual será desarrollado en la siguiente sección) y a una lista temporal para las preclasificaciones obtenidas para la tabla que está analizando mientras la acción Clasificar está siendo ejecutada.

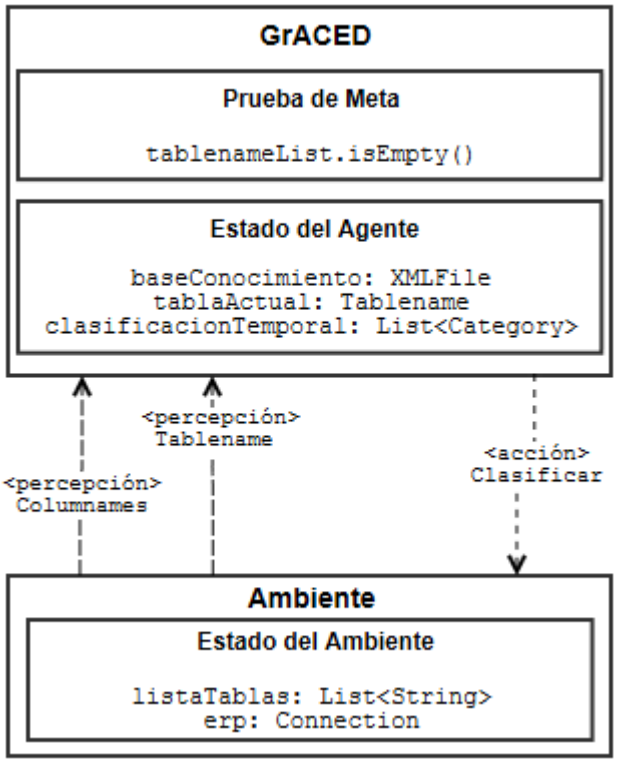


Figura 6. *Estructura básica del agente inteligente.*

Finalmente, el agente también tiene una prueba de meta, la cual le permite evaluar si ha llegado a su objetivo, o si aún necesita continuar trabajando.

3.1. Algoritmo de Razonamiento

El algoritmo de razonamiento es ejecutado durante la acción de clasificar, con el objetivo de emparejar cada tabla con una o más categorías que representen la información que contiene. Para esto, emplea los nodos "habilitados" del grafo, y que sirve para encontrar qué tipo de información almacena cada tabla.

Dado que hay dos BoW por categoría, este algoritmo también tiene dos pasos: el primero, es tomar el nombre de la tabla y clasificarlo utilizando las bolsas de palabras que tiene para ese efecto. Este proceso puede verse en la Figura 7.

Puede notarse que hay dos "filtros" para determinar si una clasificación es adecuada o no. El primero de ellos se realiza comparando la cantidad total de palabras del nombre de una tabla, contra la cantidad de ellas que fueron encontradas en la BoW; para poder pasar a la siguiente etapa, al menos la mitad de las palabras del nombre debe estar en la bolsa. El segundo filtro implica calcular el porcentaje de pertenencia (sumar los pesos de las palabras encontradas), y si dicho valor es menor a un 10%, la clasificación se descarta.

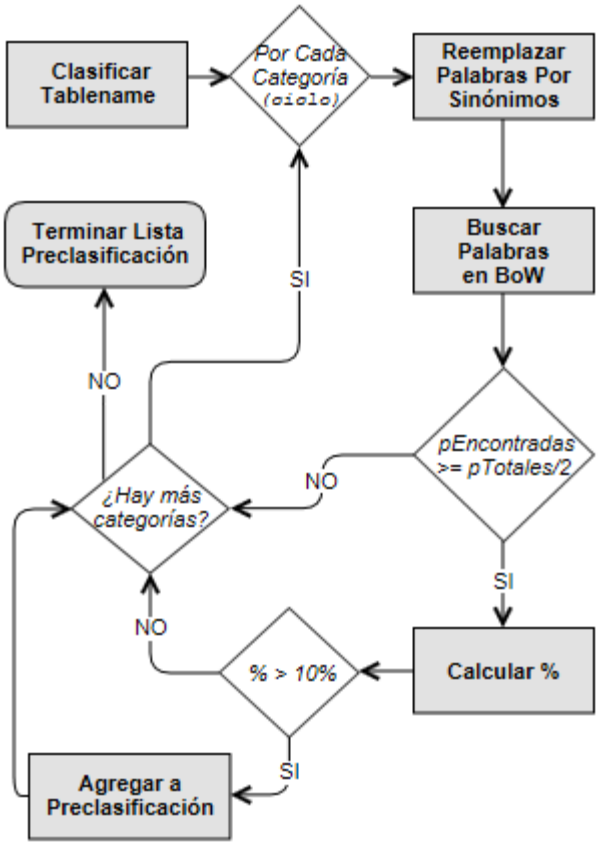


Figura 7.*Preclasificación de tablas – Parte 1 del Algoritmo de Razonamiento.*

El porcentaje del segundo filtro fue seleccionado siguiendo las premisas detalladas a continuación. En promedio, el nombre de una tabla no suele tener más de cuatro palabras, mientras que cada bolsa de palabra para los nombres de tablas debe guardar entre 25 y 30 registros, ya que hay una amplia variedad que puede usarse para cada categoría. Esto hace que encontrar un peso mayor al 10% en el nombre de la tabla, ya genere una clasificación de relativa importancia. Por esto mismo, más adelante se detallarán las tipificaciones que GrACED realiza con las clasificaciones obtenidas.

El segundo paso para la clasificación, es evaluar las palabras en los nombres de las columnas, sólo en las categorías que sobrepasaron la clasificación con el nombre de la tabla. Los pasos seguidos son muy similares a los de la Figura 7, pero con un solo filtro: evaluar que el porcentaje de pertenencia obtenido es mayor que un 15%. Si el filtro no es superado, la categoría es removida de la preclasificación de la tabla. Dado que todas las bolsas pesan 100, y las BoW para las columnas tienen mayor cantidad de palabras (por ende,

menos pesadas), tras evaluar distintas posibilidades, se llegaron a mejores resultados con este valor de filtro.

3.2. Persistencia de los Resultados

La acción del agente inteligente finaliza al persistir los datos de las clasificaciones en dos archivos XML:

- *Tablas Clasificadas:* contiene las tablas que han sido clasificadas en al menos una categoría, el nombre de la categoría, y los porcentajes de pertenencia.
- *Tablas No Clasificadas:* contiene las tablas que no pertenecen a una clasificación. Este segundo archivo existe debido a que no todas las tablas de una base de datos de un ERP contienen información de manufactura.

Por otro lado, GrACED también ofrece gráficos en su interfaz de usuario al finalizar la clasificación, lo que le permite al usuario poder realizar un estudio más complejo de los resultados obtenidos. Dichos gráficos son:

a. Un gráfico de torta con la proporción de tablas que contienen información de manufactura, y las que no. Esto es especialmente útil para analizar la distribución de los datos y la relevancia que cada empresa le da a los mismos.

b. Un gráfico de barras para cada tabla, mostrando las categorías en las que fue clasificada, y el porcentaje que obtuvo al ser analizada por nombre de tabla, por nombre de columna y el promedio. Es decir, que cada tabla puede tener pertenencia a más de una categoría, debido a la ambigüedad del lenguaje natural y a la superposición de los datos.

c. El último gráfico es un diagrama de torta que asigna un tipo a cada uno de los promedios de clasificación, con el objetivo de obtener mayor información respecto a las clasificaciones. Este tema será desarrollado a continuación, con mayor profundidad.

Dado que por cada categoría hay dos bolsas de palabras disyuntas (una para tablas, y otra para columnas), cada una genera un porcentaje de pertenencia distinto. El motivo para esto es que se puede generar un análisis mucho más valioso al evaluar dónde se presenta una pertenencia más fuerte, ya sea en el nombre de la tabla o en el de las columnas. Sin embargo, sí se muestra al usuario el promedio de ambas pertenencias.

Estos dos valores de pertenencia fueron tipificados en 3 tipos. Estos tipos no determinan la cuán correcta es una categorización, sino que estudian la predominancia de palabras genéricas o representativas (o una mezcla de ambas) en la definición de las tablas de la BD. Estos tipos pueden ser:

♦**Falsos Positivos (Tricky):** son clasificaciones en las que la pertenencia del nombre de la tabla es mucho mayor que la obtenida con los nombres de las columnas. Esto sucede en casos donde el nombre de la tabla tiene palabras muy específicas para una categoría, mientras que las columnas tienen palabras genéricas cuyos pesos son medios o bajos. Sin embargo, no se descartan porque más allá de la combinación de pertenencias obtenidas, la tabla puede contener información relevante.

♦**Positivos Totales (True):** representan a aquellas clasificaciones donde ambas pertenencias tienen porcentajes altos, ya que fueron encontradas muchas palabras clave de peso elevado (muy representativas para una categoría). Por lo general, muy pocas tablas por categoría pertenecen a este tipo.

♦**Neutrales (Neutral):** son clasificaciones no englobadas en las anteriores, donde generalmente ambas pertenencias son de nivel medio, y sólo contienen palabras de relevancia intermedia, no muy importantes pero tampoco genéricas. Suelen ser tablas que almacenan información complementaria a las Positivas Totales, tales como tablas que derivan de relaciones del diagrama Entidad-Relación de la BD. La existencia de categorizaciones de este tipo no habla de una mala categorización, sino de que la base de datos emplea muchas palabras genéricas y pocas palabras realmente significativas.

3.3. Separación de Palabras

Como puede notarse, la implementación de un agente que procese lenguaje natural, siempre va a depender de dos situaciones ajenas al mismo: la sintaxis y la semántica de las palabras. Si las palabras son escritas en idiomas que el agente no comprende, o con errores ortográficos, éste no podrá procesarlas. Lo mismo sucede si la semántica de las palabras no es utilizada adecuadamente; por ejemplo, si una columna se llama `cellphone_number` pero en realidad contiene un nombre de persona física, el agente generará una clasificación basada en el nombre de la columna, y no en el contenido de la misma, ya que la semántica de la etiqueta ha sido usada erróneamente.

Un punto importante relacionado con las palabras, es la separación de las mismas. Generalmente, en los lenguajes de programación se utilizan convenciones de nombres (o *naming conventions* por el nombre en inglés), que establecen métodos para separar las palabras. Las bases de datos actuales no tienen ninguna convención preestablecida -y aunque la tuvieran, no hay forma de asegurar que los desarrolladores las utilizarían- por lo que el problema de la separación no resulta trivial.

Para solucionar esto, antes de comenzar la

clasificación, el agente solicita que se le instruya qué método de separación va a emplearse, lo cual puede observarse en la Figura 8.

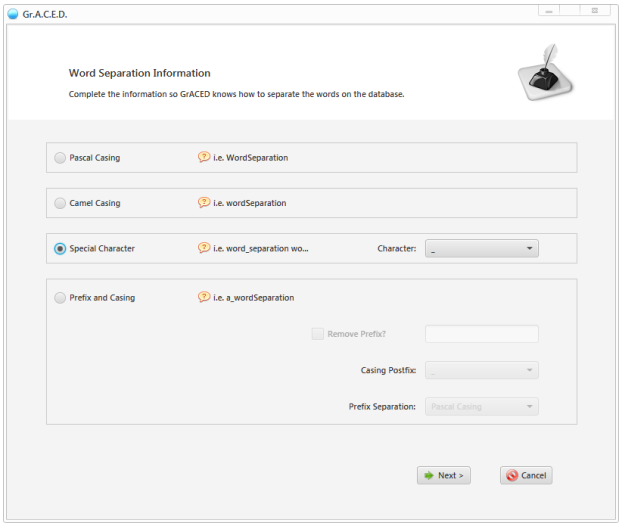


Fig. 8. *Interface gráfica que solicita información sobre la separación de palabras.*

Los tipos de separación comprendidos, por el momento, son:

- ***Pascal Casing*:** las palabras se escriben juntas, y cada una empieza con letra capitalizada. Ejemplo: `UnEjemploDePalabras`.
- ***CamelCasing*:** similar al anterior, la primera palabra lleva letra minúscula. Ejemplo: `unEjemploDePalabras`.
- **Separación por Caracteres:** las palabras son escritas en minúsculas, y cada una se separa usando un carácter especial (punto, espacio, guion medio, guion bajo). Ejemplo: `un_ejemplo_de_palabras`.
- **Separación Mixta:** es una separación más compleja y personalizable, y permite seleccionar un prefijo que será eliminado y no analizado, una separación para el prefijo del resto del nombre, y una para el nombre restante.

Cabe mencionar que si una base de datos no mantiene una semántica adecuada, ni consistencia en el método de separación de palabras, GrACED no clasificará a su máxima capacidad. Esto será demostrado en uno de los casos de estudio analizados en la siguiente sección.

4. Implementación y Casos de Estudio

La implementación de un agente inteligente es una tarea compleja, por lo que se decidió utilizar FAIA [16]: un *framework* generado en Java, que ofrece una estructura de clases abstractas que generan varios tipos de agentes inteligentes (reactivos, basados en metas, basados en conocimiento, etc.), y que sirven de marco para implementar la funcionalidad básica de todo agente (la entidad, el ambiente, estado del ambiente, estado del agente, percepciones y acciones).

A su vez, la base de conocimiento ha sido implementada en XML, como se mencionó previamente, debido a la portabilidad, flexibilidad y universalidad que este lenguaje ofrece, además de permitir una fácil modificación y agregado de palabras. Otra ventaja es que a partir de la versión Java 8, las librerías para la lectura/escritura de este tipo de archivos ya se encuentran incorporadas en el lenguaje, quitando la necesidad de utilizar archivos JAR externos.

Con el objetivo de realizar una primera implementación y evaluar el comportamiento y la arquitectura propuesta para GrACED, se decidió trabajar inicialmente sólo con una rama del grafo de categorías (Fig. 3). De este modo, en la Figura 9 puede observarse el grafo sobre el que se trabajó en la implementación.

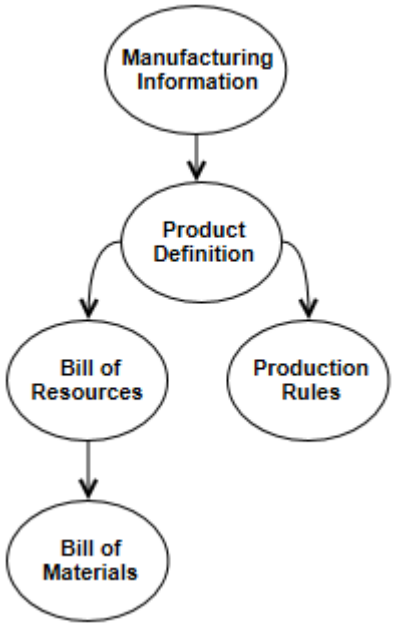


Figura 9.*Implementación inicial del grafo derivado del estándar ANSI/ISA-95.*

Para generar las BoW, los archivos de sinónimos y posteriormente estudiar el comportamiento de GrACED, se emplearon ERP de código abierto pero de amplia utilización en el mercado.

De esta forma, de todos los ERP *open-source* seleccionados para la utilización, sólo se emplearon cuatro para generar la BC del agente (esto incluye BoW de tablas/columnas, y archivos de sinónimos): Compiere [17], OpenERP [18], ERPNext [19] y JFire [20]. Se reservaron dos ERP completamente distintos como conjunto de evaluación: Dolibarr [21] y Libertya [22] (este último no llegó a ser evaluado para este artículo), pero también se agregaron dos casos de estudio

"especiales": OpenERP (ya utilizado en el subconjunto de entrenamiento) y Adempiere [23] (un *fork* de Compiere, pero con bastantes diferencias).

A su vez, el lenguaje natural que se consideró para realizar la BC, fue el idioma inglés. Cualquier palabra en otro idioma, fue tratada como sinónimo de su correspondiente palabra en inglés.

Del grafo de la Figura 8, sólo los nodos *Bill of Materials*, *Bill of Resources* y *Production Rules* fueron empleados para clasificar. A su vez, en la Tabla 1 pueden observarse algunas estadísticas de la base de conocimiento empleada, para poder considerar su tamaño:

Tabla 1. *Datos de la composición de la BC inicial de GrACED.*

Cant. de BoW para Tablenames	3
Cantidad de BoW para Columnames	3
Archivos de Sinónimos	189
Palabras en BoW de Tablenames	70
Palabras en BoW de Columnames	423
Proporción Columnames por Tablenames	6,043
Palabras Totales	493

4.1. OpenERP

OpenERP [18] es una suite ERP de código abierto, publicado con una licencia AGPL2 [24] e implementado como una aplicación web. Su funcionamiento se centra en la lógica de negocios y en el módulo MRP. Esta suite también fue utilizada por los autores en el caso de estudio de una investigación previa [9].

La base de datos de OpenERP fue implementada en PostgreSQL, sí mantiene consistencia en la convención de nombres, usando siempre las letras en minúsculas, separadas con guiones bajos. De esta forma, `m_production_id` fue considerado como un nombre adecuadamente separado (o preciso), mientras que `movementdate` se consideró incorrecto por la falta de guion bajo entre ambas palabras.

Las estadísticas de precisión de la separación de palabras de la BD de OpenERP pueden observarse en la Tabla 2:

Tabla 2. *Datos de Precisión de BD de OpenERP.*

Total de Nombres de Tablas	450
Nombres de Tablas Correctos	416
Total de Nombres de Columnas	4753
Nombres de Columnas Correctos	4652

De este modo, el 92.44% de los nombres de las tablas de la BD se encuentran adecuadamente separados, mientras que el 97.87% de los nombres de columnas tiene una separación correcta. Esto da un total de 97.405% de precisión en la base de datos.

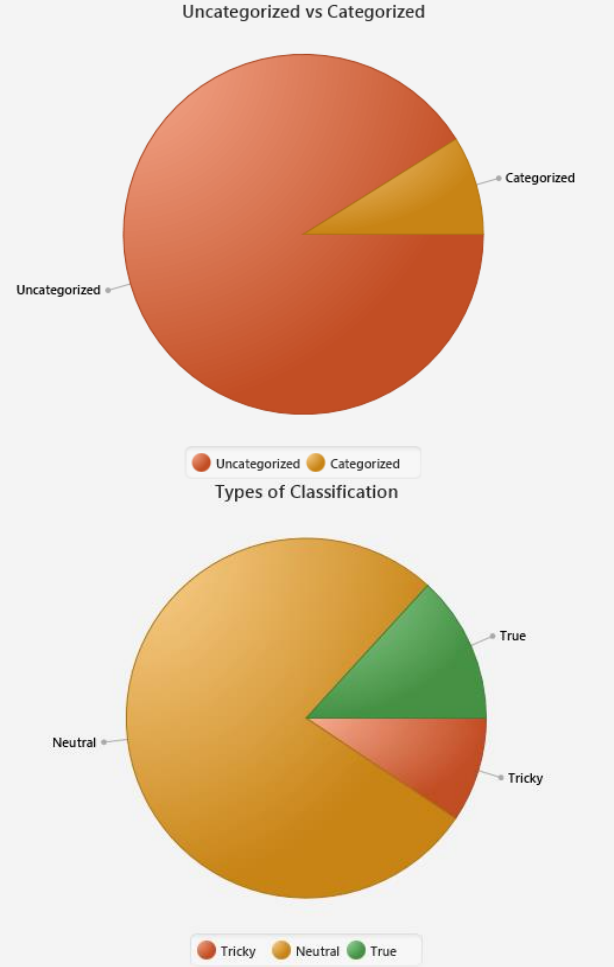


Fig. 10. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).OpenERP.*

Este ejemplo fue analizado con GrACED y algunos de los resultados pueden verse a continuación en la Fig. 10 (arriba), donde se puede observar una de las pestañas con resultados generados por el agente, donde del total de tablas, el 8.89% contiene información sobre la categoría *Product Definition* y el 91.11% no. Por otro lado, en la Fig. 10 (abajo) se observa la separación en tipos de clasificación, para todas las tablas que han sido categorizadas. Aquí se cuenta el total de clasificaciones, ya que una tabla puede pertenecer a más de una categoría; de esta forma, hay una mayoría de categorizaciones de tipo neutral (77.36% del total) dado que en la mayoría de las etiquetas o nombres de tablas no se emplean palabras realmente representativas. Del total de categorizaciones, un 13.21% fue tipificada como Positivo Total (pertenencia de nombre de tabla y de nombre de columnas mayor al 50%), y el restante 9.43% se consideró Falso Positivo.

Para evaluar el comportamiento obtenido con GrACED se compararon los resultados automatizados del

agente, contra una clasificación manual realizada por expertos. De esta forma, los expertos realizaron 44 categorizaciones, y GrACED coincidió con 40, lo que representa un 90.91% de certeza. A su vez, el agente agregó 13 categorizaciones, de las cuales 9 fueron posteriormente consideradas correctas por los expertos, tras estudiar el contenido de información y palabras de las mismas.

4.2. Dolibarr

Dolibarr [21] es un ERP de código abierto, publicado bajo una licencia GNU General Public License 3.0 [25], orientado a empresas y compañías de tamaño medio. De origen francés, Dolibarr tiene más de 26 módulos, considerando entre ellos un catálogo de productos y servicios, administración de órdenes de venta y producción, envíos, entre otros.

Para este estudio, se empleó la versión estable 3.5.2 liberada en Abril de 2014, y la base de datos fue implementada en MySQL. Cabe destacar que este ERP no fue utilizado para generar la BC de GrACED.

Similarmente a OpenERP, el método principalmente empleado para la separación de las palabras en dicha BD, es el caracter especial guion bajo. Sin embargo, este ERP no posee la misma precisión que en el caso de estudio anterior, y esto puede verse en los datos de la Tabla 3:

Tabla 3. *Datos de Precisión de BD de Dolibarr.*

Total de Nombres de Tablas	176
Nombres de Tablas Correctos	130
Total de Nombres de Columnas	1967
Nombres de Columnas Correctos	1855

De este modo, Dolibarr posee un 73.86% de precisión en la separación de palabras de los nombres de tablas, y un 94.31% de precisión en los nombres de columnas. Esto da una precisión promedio de 92.63%.

Sin embargo, este ERP tiene un detalle que es importante mencionar: muchas palabras en las etiquetas de columnas y tablas fueron escritas en francés, en lugar de inglés. Para analizar la incidencia del idioma francés en la BD, se separaron las palabras manualmente (corrigiendo aquellas separaciones incorrectas) y se obtuvieron los datos de la Tabla 4:

Tabla 4. *Estadísticas del idioma en Dolibarr.*

Total de Palabras en Tablenames	569
Palabras en Francés en Tablenames	126
Total de Palabras en Columnames	3235
Palabras en Francés en Columnames	307

Calculando los porcentajes, hay un 22.14% de las palabras en los nombres de tablas (o *tablenames*) escritas en francés, y un 9.49% en los nombres de columnas (o *columnames*). Analizando el ERP completo, se obtiene que el 11.38% de las palabras empleadas en la BD fueron escritas en francés, en lugar de inglés.

Para trabajar con estas palabras, el procedimiento fue distinguir las palabras en francés y armar una lista con los significados en inglés de cada una, y posteriormente agregar las palabras francesas al archivo de sinónimos de la palabra en inglés. De este modo, no se modificó la base de conocimiento ni las bolsas de palabras, pero se le dio a GrACED la capacidad de comprender (limitadamente) el francés.

Una vez completados los archivos de sinónimos, Dolibarr fue analizado con GrACED y se obtuvieron los resultados que pueden verse en la Figura 11.

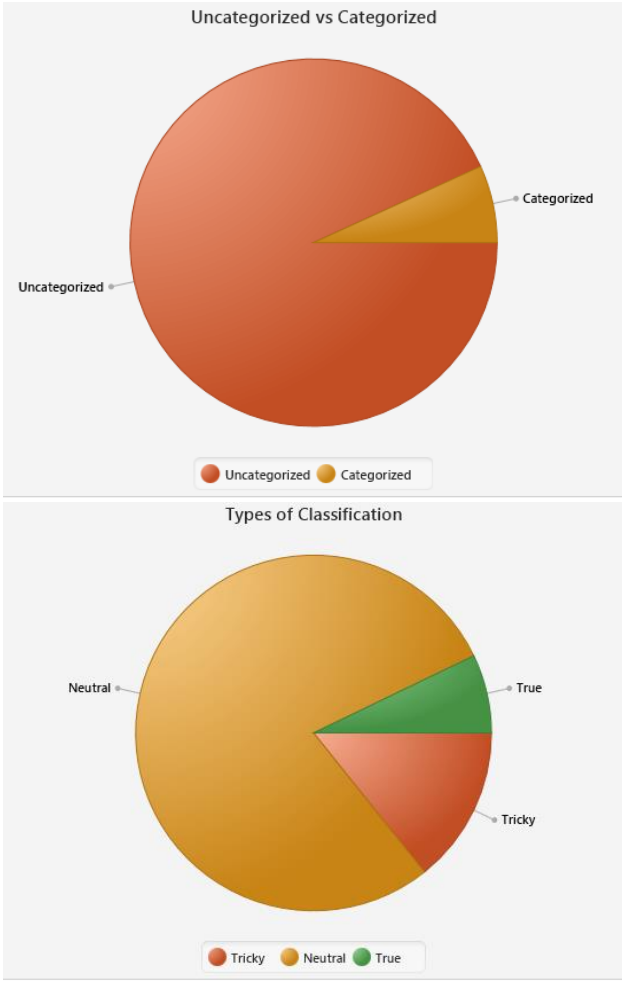


Fig. 11. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Dolibarr.*

De este análisis surge que el 6.82% de las tablas de Dolibarr contienen información de la categoría *Product Definition*¸ que es lo que puede observarse como la porción denominada *Categorized* en la Fig. 11 arriba. Considerando las categorizaciones realizadas en los tipos

definidos previamente, se concluyó que el 78.57% de las son de tipo Neutral, el 7.15% son Positivos Totales (*True*, en el gráfico Fig. 11 abajo) y el 14.28% restante fueron consideradas Falsos Positivos (*Tricky*, en la Fig. 11, abajo).

Nuevamente, y con el objetivo de evaluar el comportamiento de GrACED en el proceso, se compararon los resultados automatizados contra una categorización manual realizada por expertos. Así, los expertos realizaron 16 categorizaciones y GrACED coincidió con 13, lo que representa un 81.25% de certeza. A su vez, el agente agregó sólo 1 categorización, la cual fue posteriormente aceptada como correcta por los expertos.

Este caso de estudio fue considerado como exitoso, si bien es importante destacar la diferencia de tamaño de las bases de datos de OpenERP y Dolibarr, dado que este último tiene una BD de un tamaño 60% menor, aproximadamente. Esto deriva en menor cantidad de tablas con información de Definición del Producto y, por ende, menos clasificaciones.

4.3. Adempiere

Adempiere [23] es otro ERP de código abierto, desarrollo como un *fork*[2] de Compiere, y publicado bajo una licencia GNU General Public License [25]. Este sistema tiene una base de datos de gran tamaño, implementada en Oracle 10g XE.

Este ERP tiene una gran base de datos (con 726 tablas y más de 14000 columnas) pero no posee una buena precisión en la separación de palabras, ya que no ha empleado una convención en particular.

Con los datos de la Tabla 5, se obtiene que sólo 43.52% de los nombres de tablas tienen una separación correcta, mientras que en las columnas el porcentaje es aún menor, apenas alcanzando el 35,36%. El porcentaje total de precisión es35,76%.

Tabla 5. *Datos de Precisión de BD de Adempiere.*

Total de Nombres de Tablas	726
Nombres de Tablas Correctos	316
Total de Nombres de Columnas	14033
Nombres de Columnas Correctos	4958

Esto sucede debido a que no hay una estandarización en la utilización de un *naming conventions*: la única separación que puede considerarse como tal es la utilización del guion bajo, debido a que es la única notable. Oracle es una base de datos sensible a las mayúsculas, y las etiquetas de tablas/columnas han sido escritas totalmente en mayúsculas, aun cuando se podía llegar a emplear *Pascal* o *Camel casing*.

Otro problema en esta base de datos, es el uso de palabras genéricas como "`bname`" o "`description`", sin emplear otros modificadores que agreguen mayor valor semántico, lo que disminuye considerablemente el porcentaje de pertenencia obtenido al intentar clasificar las tablas de esta base de datos.

Finalmente, otro problema de Adempiere es la redundancia de información: se encuentran repetidas gran cantidad de tablas que guardan la misma o similar información, y que sólo agregan datos duplicados, dificultad de mantenimiento y de integración con otros sistemas.

Por lo problemas mencionados es que se decidió analizar Adempiere con GrACED, con el objetivo de poder estudiar el comportamiento del agente en ambientes que no son óptimos. Los resultados de este estudio se encuentran en la Fig. 12.

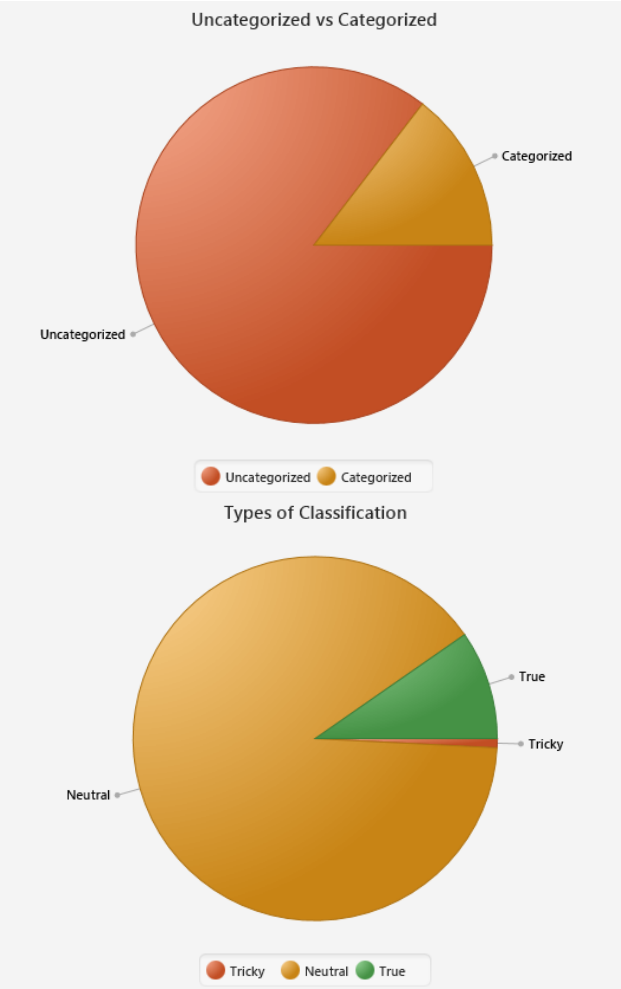


Fig. 12. *Resultado comparando las tablas categorizadas contra las no categorizadas (arriba). Resultado de las tablas categorizadas, separadas por tipo (abajo).Adempiere.*

[2]Un *fork* sucede cuando los desarrolladores copian el código fuente de un paquete de software y comienzan un desarrollo independiente sobre éste, creando un software distinto. Común en desarrollos de código abierto.

De los resultados obtenidos se puede ver que el 14.56% de las tablas de la BD de Adempiere contienen información de *Product Definition*, generando 93 clasificaciones. A su vez, tipifica las clasificaciones de la siguiente forma: el 89.52% son Neutrales, el 98.68% son Positivos Totales y sólo el 0.81% son Falsos Positivos.

Sin embargo, el impacto de la redundancia y la incoherente separación de palabras se ve reflejado en las coincidencias: los expertos sólo clasificaron 30 tablas, de las cuales GrACED coincidió con 19 (un 63.33%). No obstante, el agente agregó 73 clasificaciones y sólo 13 fueron consideradas como adecuadas por los expertos; esto significa que el 65% de las categorizaciones fueron agregados inconsistentes del agente.

Se podría decir que en este caso de estudio, y ante la presencia de problemas de redundancia, carencia de *naming conventions* y utilización excesiva de palabras no representativas, GrACED comienza a "sobre-categorizar" las tablas, en lugar de clasificar de menos. Esto se debe a lo siguiente: un nombre de columna como `ismanufacturingresource` debería contar como 3 palabras pero en realidad, el agente la distingue como una sola palabra porque no la puede separar. Esto hace que la cantidad de palabras totales sea menor que las reales, y se pasen los filtros de cantidad de palabras encontradas.

Podemos concluir que, de cierta manera, es un caso exitoso porque podemos sostener las premisas que se plantearon al inicio: al trabajar en lenguaje natural, existe una fuerte dependencia entre la separación de las palabras y los resultados de la clasificación, así como también entre la semántica de las palabras empleadas y las BoWs generadas.

5. Conclusiones

El presente trabajo propone la estructura básica para un agente inteligente basado en conocimiento y denominado GrACED, el cual trabaja con lenguaje natural (idioma inglés) y que utiliza una base de conocimiento estructurada en bolsas de palabras y generada a partir de la estructura de datos, modelos y definiciones de categorías de información de manufactura propuestos en el estándar ANSI/ISA-95.

El objetivo de GrACED es enlazarse con un sistema ERP (reconocido como su ambiente) y analizar el contenido de su base de datos, para estudiar no solo cómo se estructuran los datos, sino también para encontrar la información necesaria para la integración entre sistemas. Esto resulta especialmente útil al momento de la integración de sistemas de los miembros de una cadena de suministro, o al intentar lograr la colaboración entre el sistema empresarial y un sistema tipo APS (*Advanced Planning and Scheduling*).

Su funcionalidad fue evaluada a través de tres casos de estudio, empleando sistemas ERP de código abierto: OpenERP, Dolibarr y Adempiere, logrando comportamientos favorables, con una precisión total mayor al 80% en los casos exitosos, y un comportamiento esperado en el caso negativo.

Como prototipo del proyecto, la implementación actual de GrACED ha logrado buenos resultados por lo que surgen varios trabajos futuros, entre ellos, lograr la utilización completa del grafo de clasificación, y evaluar un caso más de estudio: Libertya, un ERP de código abierto, origen argentino y base de datos completamente en español.

Otro punto importante, es lograr la propagación de pertenencia a las distintas categorías. Observando el grafo de la Figura 3, la propuesta es que, utilizando las pertenencias obtenidas en la clasificación básica desarrollada en este trabajo, se pueda propagar el porcentaje hacia arriba en el grafo con el objeto de encontrar el impacto que cada tabla tiene en el total de la información de manufactura contenida en la base de datos. A su vez, dado que una tabla puede pertenecer a más de una categoría, esto serviría para dar mayor información sobre a qué categoría de nivel uno pertenece con mayor intensidad.

Al lograr una pertenencia total, también puede estudiarse la adecuación de la base de datos al ANSI/ISA-95, lo cual es el objetivo último de este proyecto, ya que permitiría aplicar a GrACED para estudiar los ERPs que, por ejemplo, desearían aplicarse a una empresa, u obtener indicaciones sobre dónde se encuentra la información necesaria para un intercambio de datos estandarizado.

6. Trabajos citados

[1] EU-Commission, MANUFACTURE - A vision for 2020. Assuring the future of manufacturing in Europe., Office for Official Publications of the European Communities, 2004.

[2] I. Harjunkoski, R. Nyström y A. Horch, «Integration of scheduling and control - Theory or practice?,» *Computers and Chemical Engineering,* vol. 33, pp. 1909-1918, 2009.

[3] E. Muñoz, E. Capón-García, A. Espuña y L. Puigjaner, «Ontological framework for entreprise-wide integrated decision-making at operational level,» *Computers and Chemical Engineering,* vol. 42, pp. 217-234, 2012.

[4] ISA, ANSI/ISA-95.00.01-2000. Enterprise-Control System Integration. Part 1: Models and terminology, ISBN: 1-55617-727-5, 2000.

[5] L. Prades, F. Romero, A. Estruch, A. García-

Dominguez y J. Serrano, «Defining a Methodology to Design and Implement Business Process Models in BPMN according to the Standard ANSI/ISA-95 in a Manufacturing Enterprise,» *The Manufacturing Engineering Society International Conference, MESIC 2013*, vol. 63, pp. 115-122, 2013.

[6] C. Kardos, G. Popovics, B. Kádár y L. Monostori, «Methodology and data-structure for a uniform system's specification in simulation projects,» de *Forty Six CIRP Conference on Manufacturing Systems 2013*, 2013.

[7] D. He, A. Lobov y J. L. Matinez-Lastra, «ISA-95 Tool for Enterprise Modeling,» de *ICONS 2012: The Seventh International Conference on Systems*, 2012.

[8] D. Brandl, «Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95,» *Revue de l' electricite et de l' electronique*, nº 8, pp. 46-52, 2002.

[9] M. Vidoni y A. Vecchietti, «E2OL: Sistema de Planeamiento y Scheduling Personalizable e Integrable con ERPs,» de *1º Congreso Nacional de Ingeniería Informática y Sistemas de Información*, Córdoba, 2013.

[10] O. P. Quiñonez-Gámez y R. G. Camacho-Velázquez, «Validation of production data by using an AI-based classification methodology; a case in the Gulf of Mexico,» *Journal of Natural Gas Science and Engineering*, vol. 3, pp. 729-734, 2011.

[11] Y. Fu, W. Ke y J. Mostafa, «Automated Text Classification Using a Multi-Agent Framework,» de *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, Denver, Colorado, 2005.

[12] H. M. Wallach, «Topic Modeling: Beyond Bag-of-Words,» de *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburg, PA, 2006.

[13] P. Norvig y S. Russel, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.

[14] ISA, ANSI/ISA-95.00.03-2005. Enterprise-Control System Integration. Part 3: nActivity models of manufacturing operations management, 1-55617-955-3 ed., ISA, 2005.

[15] W3C Recommendation, «Extensible Markup Language (XML) 1.1 (Second Edition),» 2006. [En línea]. Available: http://www.w3.org/TR/xml11/#sec-xml11. [Último acceso: 01 04 2014].

[16] J. Roa, M. Gutierrez, M. Pividori y G. Stegmayer, «How to develop intelligent agents in an easy way with FAIA,» de *Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design*, IGI global, ed. Francisco V. Cipolla Ficarra, 2010, pp. 120-140.

[17] A. L. Pretorious, Compiere 3, Birmingham: Packt Publishing Ltd., 2010.

[18] OpenERP S.A., «OpenERP,» 2012. [En línea]. Available: https://www.openerp.com/. [Último acceso: 20 April 2014].

[19] Panorama Consulting Solutions, «ERPNext,» 19 Noviembre 2010. [En línea]. Available: http://panorama-consulting.com/erp-vendors/erpnext/. [Último acceso: 2014].

[20] NightLabs Consulting GmbH, «JFire,» 2011. [En línea]. Available: http://www.jfire.net/. [Último acceso: 2014].

[21] L. Destailleur, «Dollibar ERP/CRM,» 2014. [En línea]. Available: http://www.dolibarr.org/. [Último acceso: 2014].

[22] F. Cristina, M. Mauprivez, M. Nerón Cap, J. M. Castro y F. Bonafine, «Libertya ERP,» 2011. [En línea]. Available: http://www.libertya.org/producto/preguntas-frecuentes. [Último acceso: 2014].

[23] B. C. Pamungkas, «ADempiere 3.4 ERP Solutions,» Birmingham, UK, Packt Publishing, 2009.

[24] GNU Affero, «Affero General Public Licence,» 2007. [En línea]. Available: http://www.gnu.org/licenses/agpl-3.0.html. [Último acceso: 20 April 2014].

[25] Free Software Foundation Inc., «GNU General Public Licence,» 29 June 2007. [En línea]. Available: https://gnu.org/licenses/gpl.html. [Último acceso: 2014 April 20].

QRipt

Herramienta para automatizar la carga de documentos usando códigos QR que contienen datos en formato JSON

Armando, Silvana; Bianciotti, Andrés; Bonino, María Belén; Calloni, Juan Carlos; Gioino, Mauro; Knüssel, Federico; Mulassano, Micaela; Ponce, Martín; Rossi, Martín

Universidad Tecnológica Nacional, Facultad Regional San Francisco

Abstract
El proyecto consiste en una librería que permite leer y grabar información masiva resultante de transacciones comerciales (como pueden ser facturas, cheques, remitos, entre otros documentos) de forma correcta, a través de códigos QR (Quick Response) y utilizando JSON (JavaScript Object Notation) como formato estructurado para el intercambio de datos. Su objetivo es brindar soporte a los desarrolladores de sistemas de gestión, e intenta resolver los problemas originados por la carga de grandes volúmenes de datos en sistemas de información. En la actualidad este proceso se realiza en forma manual, provocando errores de tipeo y grandes demoras de tiempo.

Palabras Clave
QR (Quick Response), compresión, JSON (JavaScript Object Notation), JavaScript, Data Entry, comprobantes, documentos, eliminación de errores.

Introducción

Este proyecto intenta ser una herramienta para desarrolladores que permita reemplazar la carga manual y masiva de documentos con un gran volumen de información en sistemas de gestión en general (por ejemplo sistemas administrativos, contables, financieros, entre otros). El objetivo del proyecto es, entonces, permitir la realización de transacciones correctas (esto es, sin errores resultantes de la carga manual de datos) empleando códigos QR y datos estructurados comprimidos.

Elementos del Trabajo y metodología

En una primera etapa definimos el marco teórico del proyecto analizando la presencia de antecedentes. Se definen los conceptos y proposiciones que permiten abordar la problemática planteada, ubicándola en un contexto unificado, integrándola con conocimientos previos y ordenando los mismos para que resulten útiles al proyecto.

Luego de esto se realiza una recolección de datos en forma empírica con el fin de obtener los conocimientos necesarios para la construcción de la herramienta y el uso de códigos QR con JSON. El equipo se divide en subgrupos, donde cada uno realiza una de las siguientes actividades de investigación:

- ***QR (Quick Response)***

Se estudia en profundidad el estándar ISO/IEC 18004:2000 que define la tecnología QR principalmente para determinar dos puntos importantes. Uno de ellos es el tipo de código QR que se va a utilizar, en base a los objetivos y lo que se pretende realizar, y por otro lado se analiza la cantidad de información que pueden almacenar teniendo en cuenta el tipo de dato que se intenta incluir en los mismos.



Figura 1. Estructura de un código QR

- ***Estructura de Documentos Comerciales***

Se investiga la estructura genérica de las diferentes categorías de comprobantes existentes en el mercado (facturas, recibos, cheques, entre otros), a los fines de determinar la cantidad de caracteres a representar.

En base a esto, se busca minimizar la cantidad de códigos QR que se utilizan para representar un comprobante.

El resultado de este paso se refleja en el siguiente modelo de dominio para facturas siguiendo el estándar consignado por AFIP en su sitio web oficial, con el que se trabajará en el desarrollo de la herramienta:

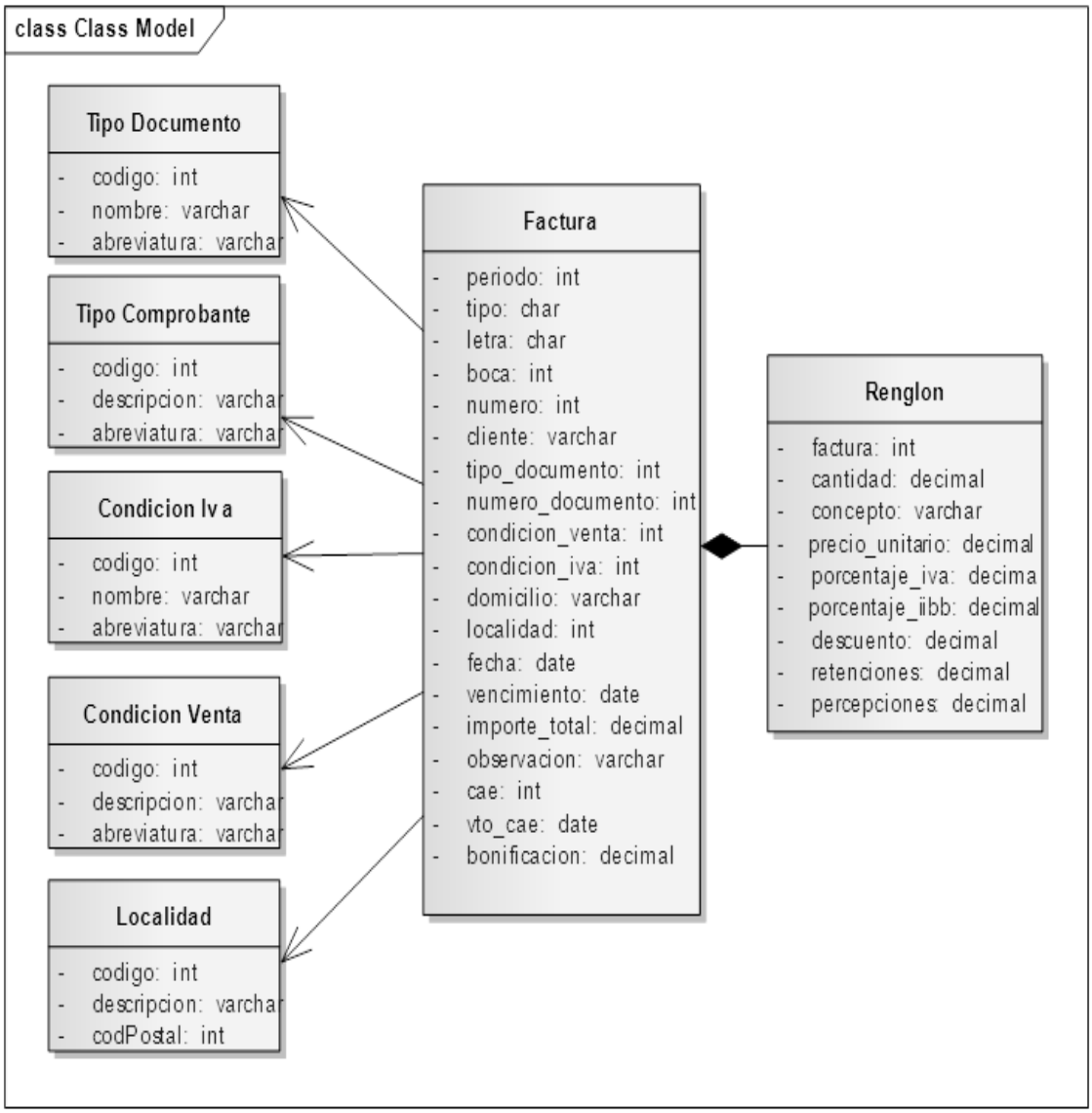


Figura 2. Modelo de dominio de facturas siguiendo el estándar de AFIP

- ***JSON***

Se indaga acerca de JSON (JavaScript Object Notation) y sus distintas características como estándar de representación en intercambio de datos. Cada sección de un comprobante se representa mediante un código QR que contiene datos en formato JSON, dado que es un lenguaje de intercambio de información estructurada utilizado para almacenar datos de forma legible y flexible a esquemas diversos.

- ***Compresión***

Se evalúan las diferentes alternativas existentes en materia de compresión de archivos JSON, debido a la necesidad de minimizar el tamaño del código QR que se adjuntará en el comprobante. Se tienen en cuenta las herramientas de compresión genéricas (ZIP, GZIP, RAR, entre otras) y los compresores específicos de JSON (HPack, CJSON, JSONH). Estos compresores específicos están diseñados para aprovechar la estructura de los documentos JSON y así obtener un mejor nivel de compresión. JSON puede ser muy compacto y eficiente si se usa de manera efectiva.

Tabla 1. HERRAMIENTAS DE COMPRESIÓN

GENÉRICAS	*NATIVAS*
RAR	*HPack*
ZIP	*CJSON*
gzip	*JSONH*
kgb	
7z	

- ***Librería***

Se investiga cómo construir una herramienta que sea extensible a la mayor cantidad posible de plataformas de desarrollo y lenguajes de programación (Java, .NET, PHP, Python, Ruby, etc.). Por este motivo se evalúa el lenguaje JavaScript a los fines de determinar si es el más adecuado para llevar a cabo este desarrollo y cumplir este objetivo primordial que es la portabilidad, siendo el resultado favorable. En este sentido, una librería constituye una forma de implementación ideal para este proyecto, ya que potencia la flexibilidad de la herramienta al mismo tiempo que le aporta un grado importante de facilidad de uso para que sea adoptada en la mayor cantidad posible de sistemas de gestión existentes y futuros.

Una vez finalizada la etapa de investigación, se seleccionan las tecnologías, algoritmos y técnicas a utilizar.

La metodología de trabajo para gestionar el proyecto se basa en Scrum, en la cual se aplica de manera regular un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado de un proyecto. No está basada en el seguimiento de un plan sino en la adaptación continua a las circunstancias de la evolución del proyecto y está orientada a las personas más que a los procesos. Esta metodología se basa en el desarrollo incremental, donde al final de cada iteración se dispone de cierto producto que se puede evaluar. Estas iteraciones se conocen como *sprints* y tienen una duración promedio de 30 días. Por cada sprint el equipo define objetivos, tareas a realizar y responsabilidades de cada integrante a lo largo de dicha iteración. La evolución del proyecto se gestiona a través de reuniones breves periódicas en las que todo el equipo revisa el trabajo realizado hasta el momento y lo previsto. De esta manera el grupo de trabajo se auto organiza, tomando las decisiones que se crean necesarias en cuanto al desarrollo y avance del proyecto. Al finalizar el sprint se realiza el análisis y revisión del incremento generado y se toman las decisiones oportunas para pasar a la siguiente fase de desarrollo.

Posteriormente, se realizará una simulación de situaciones reales en entorno de laboratorio para probar las tecnologías seleccionadas (QR, JSON, plataformas, sistemas de gestión que lo puedan embeber, entre otras).

Se analizarán casos prácticos en empresas locales que posean sistemas de gestión, con el objetivo de validar requerimientos y necesidades reales. Los parámetros a medir y comparar serán el tiempo total de carga y promedio de errores ocurridos.
Se realizarán las mismas pruebas en un entorno de laboratorio utilizando un prototipo de la solución propuesta para obtener estadísticos que permitan confrontar con los resultados obtenidos anteriormente.

Además, se pretende evaluar el grado de impacto que puede provocar el proyecto en los usuarios finales, a los fines de obtener información que permita cuantificar la mejora en los procesos actuales y la utilidad del desarrollo.

Luego de obtener como resultado las necesidades técnicas para el desarrollo del proyecto basadas en los conocimientos empíricos adquiridos, se deberá contemplar que pueden sufrir modificaciones en el tiempo en base a avances tecnológicos.

A continuación se presentan diagramas BPMN y escenarios que describen el funcionamiento de la librería a desarrollar.

PROCESO DE GENERACIÓN

La figura 3 detalla las actividades y la figura 4 muestra un escenario simbólico del proceso de generación de los códigos QR en los comprobantes físicos. Los pasos del proceso son los siguientes:

1. El sistema de gestión proporciona los datos de facturación.

2. Estos datos son captados por la librería, la cual realiza las siguientes actividades sobre los mismos:
   a. Los estructura en formato JSON.
   b. Reduce su tamaño mediante un algoritmo de compresión.
   c. Genera la cantidad necesaria de códigos QR para representar los datos en el comprobante físico.

3. Luego se imprime el comprobante con los códigos QR.

4. El comprobante puede ser transferido empleando la dinámica habitual, pero con la ventaja que ahora tiene un código QR anexo que facilita la posterior lectura y carga de datos.

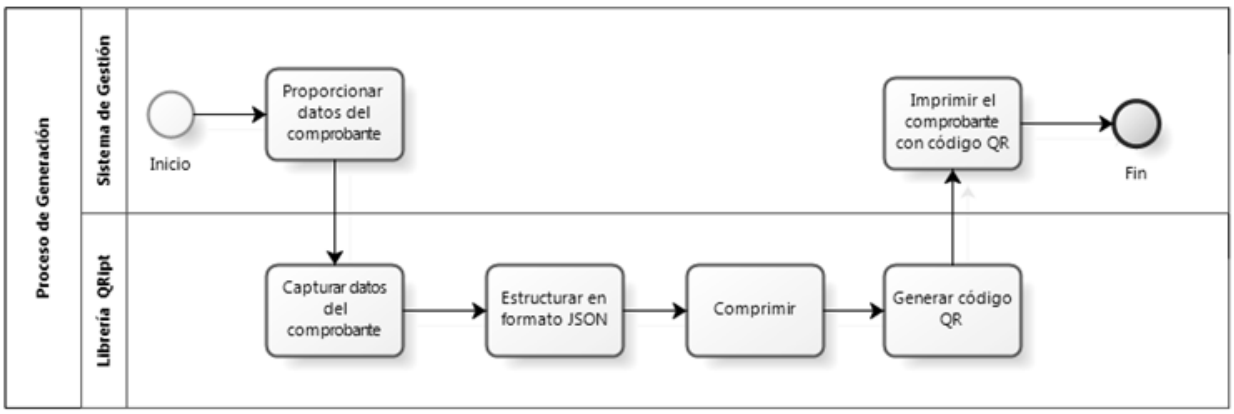


Figura 3. Diagrama BPMN del proceso de generación de los códigos QR en los comprobantes físicos

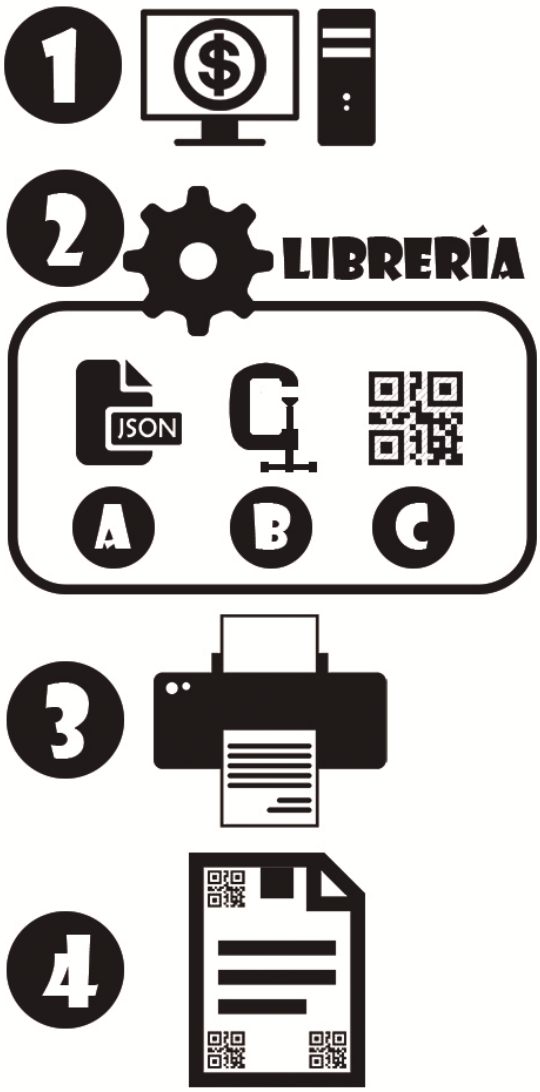


Figura 4. Escenario de generación de los códigos QR en los comprobantes físicos

PROCESO DE LECTURA E INTERPRETACIÓN

La figura 5 detalla las actividades y la figura 6 muestra un escenario simbólico del proceso de lectura e interpretación de los códigos QR en los comprobantes físicos. Los pasos del proceso son los siguientes:

1. El comprobante es recibido de la manera habitual.

2. Mediante un lector de código QR se captura cada uno de los códigos impresos en el comprobante. Esto ahorra tiempo de carga y elimina posibles errores de tipeo.

3. Los datos son enviados a la librería, la cual realiza las siguientes actividades sobre los mismos:
   a. Interpreta el contenido de los códigos QR.
   b. Lo descomprime utilizando el mismo algoritmo de compresión empleado en la etapa de generación.
   c. Identifica y estructura los datos en formato JSON.

4. Por último, los datos son enviados al sistema de gestión de manera automática.

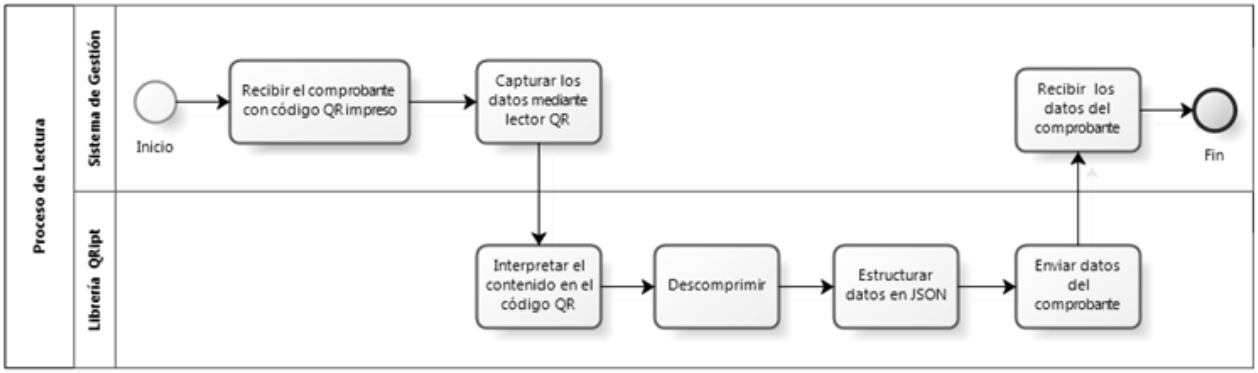


Figura 5. Diagrama BPMN del proceso de lectura e interpretación de los códigos QR en los comprobantes



Figura 6. Escenario de lectura e interpretación de los códigos QR en los comprobantes físicos



Figura 7. Logotipo de identificación del proyecto

Resultados

Se espera desarrollar una metodología automatizada, simple, rápida y libre de errores para el procesamiento de transacciones en áreas de Data Entry, al mismo tiempo que se realiza un aporte a las teorías de Compresión de Información y Gestión de Datos Estructurados.

Discusión

Durante la realización del presente proyecto se propuso agregar la función de cifrado de los datos en formato JSON dentro de la librería. Esto aumenta el tamaño de la información que se debe representar en el código QR, por lo que en primera instancia no se ha incluido. Se considera que en un futuro puede ser necesario contemplar esta funcionalidad en el caso de usar la librería para representar documentos que contengan información de carácter sensible, tales como comprobantes financieros, historias clínicas, expedientes judiciales, entre otros.

Si bien se plantea un nuevo paradigma de presentación y almacenamiento de información, esto está ligado a la utilización de documentos físicos en donde pueda ser impreso y desde donde pueda ser leído el código QR. La tendencia actual hacia la emisión de comprobantes electrónicos significa un riesgo para el presente proyecto, debido a que el desarrollo obtenido no va a ser aplicable si los datos se transfieren en formato digital entre los diferentes sistemas.

La investigación y los desarrollos efectuados dentro del proyecto en materia de compresión de datos y transferencia de información utilizando JSON son independientes de la representación mediante códigos QR. Esto permite reutilizarlos en nuevos métodos de transferencia de información en caso de que los documentos físicos dejaran de ser viables. Sin embargo, luego de analizar este

impacto, se infiere que los documentos en papel como medio de transacción no serán reemplazados en forma masiva en un futuro cercano.

Conclusión

Se plantea un nuevo paradigma de presentación y transferencia de información mediante la implementación de códigos QR que almacenan datos estructurados en formato JSON comprimido. Este paradigma es aplicado en primera instancia al ámbito de comprobantes financieros, documentos comerciales, documentos médicos, documentos legales y luego expandible hacia nuevas áreas de interés donde pueda resultar útil. También este proyecto abre nuevas líneas de investigación destinadas a combinar otras tecnologías de estructuración y codificación de datos que se adapten mejor a dominios específicos.

Referencias

1. MODELOS DE COMPROBANTES FISCALES. BIBLIOTECA AFIP [en línea]. Disponible en: http://biblioteca.afip.gob.ar/afipres/RG_AFIP_0163_A001_V000.htm Fecha de consulta: martes 17 de junio de 2014.

2. ANEXO II TIPOS DE COMPROBANTES FISCALES. BIBLIOTECA AFIP [en línea]. Disponible en: http://www.afip.gov.ar/afip/259IIcap1a.htm http://www.afip.gov.ar/afip/259IIcap1b.htm http://www.afip.gov.ar/afip/259IIcap1c.htm http://www.afip.gov.ar/afip/259IIcap1d.htm Fecha de consulta: martes 17 de junio de 2014.

3. BORDESE, Matías. Análisis y Alternativas para la Compresión de XML. Trabajo Integrador (Especialista en Sistemas y Servicios Distribuidos). Córdoba, Argentina, UNIVERSIDAD NACIONAL DE CÓRDOBA, Facultad de Matemática, Astronomía y Física (FaMAF), Agosto, 2009.

4. JACOBSON, Daniel; BRAIL, Greg & WOODS, Dan. APIs A Strategy Guide. O'Reilly Media. (2011).

5. INTERNATIONAL STANDARD ISO/IEC 18004:2000, Information technology – Automatic identification and data capture techniques - Bar code symbology – QRCode, First Edition (2000).

6. FAWCETT, Joe; AYERS, Danny; QUIN, Liam R. E.; Beginning XML, Wrox, 2012.

7. ST. LAURENT, Simon; FITZGERALD, Michael; XML Pocket Reference, O'Reilly Media, 2005.

8. SRIPARASA, Sai Srinivas; JavaScript and JSON Essentials, Packt Publishing, 2013.

9. ALLAMARAJU, Subbu; RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity, Yahoo Press, 2010.

10. MCFARLAND, David Sawyer; JavaScript & jQuery: The Missing Manual, O'Reilly Media, 2011.

11. STEFANOV, Stoyan; JavaScript Patterns, O'Reilly Media, 2010.

Tablas

Tabla 1. Herramientas de compresión

Figuras

Figura 1. Estructura de un código QR
Figura 2. Modelo de dominio de facturas siguiendo el estándar de AFIP
Figura 3. Diagrama BPMN del proceso de generación de los códigos QR en los comprobantes físicos
Figura 4. Escenario de generación de los códigos QR en los comprobantes físicos
Figura 5. Diagrama BPMN del proceso de lectura e interpretación de los códigos QR en los comprobantes físicos
Figura 6. Escenario de lectura e interpretación de los códigos QR en los comprobantes físicos
Figura 7. Logotipo de identificación del proyecto

Estrategia para la implementación de aplicaciones móviles basadas en servicios de geolocalización y crowdsourcing

María Roxana Martínez, Rocío Rodríguez, Pablo Vera

CAETI, Centro de Altos Estudios en Tecnología Informática
Facultad de Tecnología Informática, Universidad Abierta Interamericana (UAI)
Av. Montes de Oca 745 - (C1270AAH) Ciudad Autónoma de Buenos Aires, República Argentina
{Maria.Martinez; RocioAndrea.Rodriguez; PabloMartin.Vera}@uai.edu.ar

Resumen

En el presente artículo se plantea la importancia de los servicios de geolocalización y la información provista en forma colaborativa por los propios usuarios de las aplicaciones. Con el objetivo de diseñar una estrategia para la implementación de una aplicación en este dominio, se analizan distintas aplicaciones existentes para luego considerar los aspectos más relevantes de las mismas. Finalmente se plantea una estrategia y se ofrece un prototipo de la aplicación a implementar con el objeto de acompañar a los turistas ofreciendo recorridos y puntos de interés de una forma amigable.

1. Introducción

Existe un gran aumento en el mercado de las aplicaciones móviles que se orientan a brindar servicios basados en la localización, LBS (*Location Based Services*). Es posible encontrar software de búsquedas de mapas, por ejemplo: para determinar zonas urbanas específicas; búsquedas georeferenciadas por cercanía; creación de rutas turísticas en base a determinadas funciones que se necesiten por parte de las personas; etc.

La geolocalización, además, puede ser empleada para identificar grupos de personas, las cuales mediante una metodología como ser crowdsourcing, pueden obtener datos en forma colaborativa a distancia entre varios usuarios conectados. Los proyectos que emplean crowdsourcing, cuentan con las técnicas necesarias para "realizar una llamada" en forma solidaria a diferentes usuarios de la red con el fin de resolver un determinado problema, como ser a nivel turístico, sociológico, educativo, etc.

Esté artículo abordará en la sección 2 la temática de crowdsorcing. En la sección 3 se presenta un relevamiento de aplicaciones basadas en crowdsourcing y geolocalización, de las cuales en la sección 4 se destacan las características más sobresalientes. En la sección 5 se resumen las buenas prácticas, en la sección 6 se plantea una estrategia que permitirá construir el prototipado de una aplicación el cual es plasmado en la sección 7. Finalmente se plantean los trabajos futuros y las conclusiones alcanzadas.

2. Crowdsourcing

Diversos autores han definido el concepto de Crowdsourcing en esta sección se han elegido las definiciones más representativas. Crowdsourcing:

- Es "un modelo estratégico para atraer una multitud interesada y motivada de individuos capaz de proporcionar soluciones superiores en calidad y cantidad a aquellas que pueden proporcionar formas de negocio tradicionales" [1].
- Consiste en "una convocatoria abierta para recibir contribuciones de miembros de la multitud para resolver un problema o llevar a cabo tareas inteligentes, normalmente a cambio de micro-pagos, reconocimiento social (micromecenazgo) o entretenimiento" [2].
- "es un tipo de actividad en línea participativa en la que una persona, institución, organización sin ánimo de lucro, o empresa, propone a un grupo de individuos, mediante una convocatoria abierta y flexible, la realización libre y voluntaria de una tarea. La realización de la tarea, de complejidad y modularidad variable, y en la que la multitud debe participar aportando su trabajo, dinero, conocimiento, y/o experiencia, siempre implica un beneficio mutuo. El usuario recibirá la satisfacción de una necesidad concreta, ya sea esa económica, de reconocimiento social, de auto-estima, o de desarrollo de aptitudes personales, mientras que el crowdsourcer obtendrá y

utilizará en su beneficio la aportación del usuario, cuya forma dependerá del tipo de actividad realizada" [3].

Existen diversos ejemplos de aplicaciones basadas en crowdsourcing (los más relevantes serán explicados en el relevamiento presentado en la sección 4 del presente artículo). Un sencillo ejemplo son los etiquetados ó marcadores en Internet, los marcadores sociales, también llamados etiquetados colaborativos. En los sistemas de marcadores sociales, los usuarios asignan etiquetas a los recursos compartidos con otros usuarios, lo que da lugar a un tipo de organización de la información que surge de este proceso de "crowdsourcing" " [4].

Cuando la información se brinda mediante crowdsourcing es en parte compartida y provista por los propios usuarios. En estos casos es necesario identificar métodos que permitan corroborarla con el fin de evitar información errónea. En esta área también hay trabajos relacionados que son de gran interés enfocados al diseño de incentivos para la recolección y verificación de información. Esta propuesta abarca proyectos como la Wikipedia pero también en situaciones contrarreloj, como en las cartografías de urgencia. Por ello algunos autores han puesto el foco en "combinar los incentivos para reclutar participantes que verifiquen la información. Cuando un participante presenta un informe, su reclutador se convierte en responsable de verificar su exactitud. Las compensaciones al reclutador y al informador por presentar un informe correcto, así como las sanciones para los informes incorrectos, aseguran que el reclutador realizará la verificación" [5].

3. Relevamiento de Aplicaciones Móviles

Existen diversas aplicaciones móviles que utilizan crowdsourcing y se nutren de la información que en forma colaborativa brindan sus propios usuarios. En esta sección se presentan algunas aplicaciones que se han escogido por su relevancia, las cuales tienen distintas finalidades.

3.1. Aplicaciones sobre Crowdsourcing

Existen diversas aplicaciones basadas en Crowsorcing con diversos objetivos generales, entre ellas se destacan:

- BOINC [6] en donde los distintos usuarios colaboran con proyectos de investigación desarrollados en la Universidad de Berkeley, aportando la capacidad de proceso de los dispositivos móviles. Es un proyecto de investigación científica que se apoya en la computación distribuida para construir una "gran computadora" en base a los ciclos de CPU que muchos usuarios ceden de sus computadoras ó dispositivos móviles (cuando estos estaban ociosos). BOINC (Berkeley Open Infrastructure for Network Computing) permite concentrar todos estos "ciclos de CPU" que se aportan voluntariamente y distribuirlos entre múltiples proyectos de investigación que requerían la realización de cálculos de alta complejidad computacional. Tras iniciar la aplicación en Android, se debe seleccionar el proyecto con el que se desee colaborar. Esta aplicación solamente se podrá utilizar si la batería del dispositivo está por encima del 90% y si se cuenta con conexión Wi-Fi, con el fin de no generar costos al usuario que lo utiliza.
- YEEPLY [7] las características que presenta el crowdsourcing son fundamentales para lograr el desarrollo de una aplicación móvil cuando realmente no se sabe dónde encontrar al profesional adecuado. La colaboración que presenta el crowdsourcing permite dirigirse a una amplia comunidad de expertos que resuelvan su necesidad en el entorno móvil. En este ámbito Yeeply se presenta como la primera plataforma online en externalización de servicios móviles. Su objetivo es el de poner en contacto a empresas y particulares interesados en el desarrollo de un proyecto de aplicación móvil con aquellos desarrolladores freelance capacitados para llevarlo a cabo. Yeeply ofrece una solución a los proyectos de aplicaciones móviles que los clientes presentan. Es necesario que las empresas suban su proyecto a la página web y esperar a que los expertos realicen las ofertas. De entre todas las ofertas, los clientes eligen la que más se adapte a las condiciones de su proyecto y negocian con el desarrollador los puntos que falten por concretar.
- DUOLINGO [8] Esta aplicación es un buen ejemplo de crowdsourcing en el que la colaboración se ve recompensada con el aprendizaje y perfeccionamiento de otros idiomas. Duolingo está disponible en web, iOS y Android. Duolingo propone textos para ser traducidos por sus usuarios al alemán, español, francés, inglés, italiano ó portugués. Estos textos que no son casuales ya que las frases a traducir proceden de páginas webs que requieren ser traducidas para que los contenidos estén accesibles a muchas más personas.
- WIKIPEDIA [9] se sustenta, precisamente, en el crowdsourcing; es decir, en la colaboración voluntaria de millones de personas de todo el mundo. Los textos que forman parte de Wikipedia, las imágenes son también contenidos que tienen gran peso y un doble papel: ilustrar los artículos de Wikipedia y servir de banco de imágenes para muchas webs que usan este material respetando la licencia Creative Commons bajo la que se publican estos contenidos. Wikimedia Commons está disponible para Android y iOS, su uso es extremadamente sencillo: tomar fotos mientras un

usuario está en la calle. Luego, asignarles un título, una descripción y una categoría y "colaborar o cederlas" a Wikipedia.

3.2. Aplicaciones Geolocalización

- ACROSSAIR [10]: Es una aplicación guía-GPS de realidad aumentada, la cual es un buscador de diferentes puntos de referencia para las personas que estén haciendo un recorrido, como ser: colectivos, bancos, restaurantes o bien lugares turísticos, cada uno de estos dependiendo la posición actual, ya que este punto es utilizado como referencia de geolocalización. Esta aplicación posee conexión con Google, Bing, Yelp o Qype. También posee relación con las redes sociales Facebook y Twitter con el fin de compartir la ubicación actual. Si se enfoca con la cámara del dispositivo móvil hacia el suelo, puede verse un mapa con los puntos de referencia como se muestra en la figura 1.c), una vez que se coloque el dispositivo apuntando hacia delante se mostraran indicaciones para alcanzar el objetivo de interés del usuario como se muestra en las figuras 1.a) y 1.b).

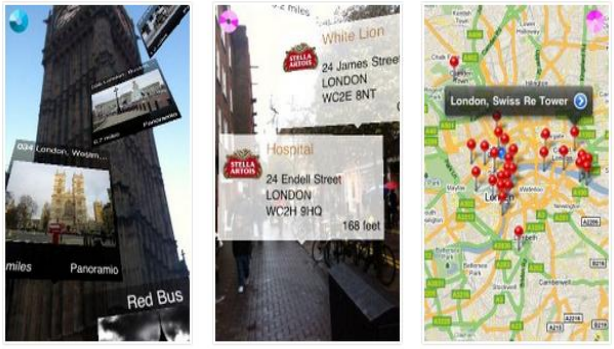


Figura 1. Aplicación Acrossair: Figura 1.a) y 1.b) Indicadores para alcanzar el punto de interés. Figura 1.c) Mapa con puntos de referencia.

- LAYAR [11]: Esta aplicación también está orientada a realidad aumentada, puntualmente a objetos y lugares. Con la cámara de foto se puede apuntar al objeto y la aplicación permite geolocalizarlo, por ejemplo: al estacionar un auto el usuario puede tener registrada la ubicación donde quedó el mismo.
- MINUBE [12]: Mediante esta aplicación se pueden localizar distintos lugares turísticos, de los cuales se puede obtener la información necesaria para conocer los sitios donde comer, o bien donde dormir. También, para cada referencia de punto turístico, se visualizan las zonas más populares que se presentan alrededor y/o más cercanas si se selecciona esa opción. Para visualizar el detalle de la opción seleccionada, se debe realizar un toque (touch), a continuación se muestra un resumen del lugar con una imagen y algunas opciones como ser: llamar al lugar, o bien compartir la localización "Estuvimos aquí".
- TRIPADVISOR [13]: Esta aplicación permite obtener los lugares de importancia turística desde la posición actual en la que se encuentra el usuario y ofrece diferentes opciones, con el fin de obtener una guía de referencia. Para cada uno de estos puntos, es posible saber la distancia y comentarios de otras personas. Un punto importante es la búsqueda de otros lugares que pudieran no figuran en el mapa mostrado para los puntos cercanos a nuestra posición actual. Mediante los comentarios y participación de otros usuarios, se conforma una red social de información de lugares turísticos. las opciones que ofrece la aplicación son: hoteles, restaurantes, atracciones, compras y actividades.
- TRIPIT [14]: Esta aplicación permite planificar un viaje desde cero. Posee un calendario de eventos los cuales podrán ser personalizados por el usuario, incorporando descripciones, observaciones, planes, nombre o lugar requerido. El funcionamiento de la aplicación es mediante la obtención del correo de confirmación de un determinado lugar en el que se realiza una compra o reserva (hotel o vuelo), la aplicación detecta cada uno de los datos y realiza una planificación de pasos a seguir identificado con mapas, e incluyendo en algunos casos, horarios de estos pasos. El objetivo de esta aplicación esta en identificar los horarios que ya tiene preestablecido de los hoteles o bien de los vuelos, ya que muestra el horario de check-in, el mapa con la localización del mismo y algunos puntos de interés para visitar. La ventaja principal es que cuenta con una sincronización de los datos para una cuenta Google. La parte de red social involucrada es en el tratamiento de contactos que posee la aplicación instalada en los dispositivos, con el fin de identificar si los contactos se encuentran cerca de los puntos de interés.
- TRIPJOURNAL [15]: Provee al usuario información tal como: Donde comer, lugares turísticos para visitar, datos de los puntos de referencia (dirección, teléfono, etc.), pero su punto fuerte es el gran poder de actualización en tiempo real que posee de los datos de los destinos. También calcula, distancias recorridas, tiempos, etc. Una característica importante de la aplicación es que se integra con Google Earth, además que tiene vinculación diversas redes sociales: Facebook, Twitter, Flickr, Picasa y Youtube.
- TOURISTEYE [16]: Esta aplicación es más sencilla a comparación de la anterior descripta, apunta a los

usuarios que les gusta más el contacto con las redes sociales, ya que la idea básica de la aplicación es registrar todas las actividades en forma de libro diario compartido y visualizado por los contactos que posean esta aplicación. Por lo que la opinión de otros viajeros es almacenada para un determinado punto turístico. Un ventaja que posee es la opción de descarga off-line de la información seleccionada, incluye, para algunos casos, la red de transporte público del lugar elegido. Esta aplicación se muestra en la figura 2.



Figura 2. Aplicación TouristEye

- FOURSQUARE [17]: Esta aplicación posee las funciones de localización de los puntos turísticos más conocidos dentro de un radio identificado por la posición actual en que se encuentra el usuario. Pero además es posible ingresar un lugar que no se encuentre identificado en el mapa que se visualiza en la aplicación, para ello se muestra un formulario para incorporar los datos más destacados de este. Una ventaja de Foursquare es que posee opción de filtros de búsqueda para refinar la información obtenida para el usuario. También, presenta conexión con las redes sociales más destacadas. Si bien al principio, esta aplicación era conocida como un juego, hoy por hoy se destaca por sus puntos de referencias turísticas.
- LABTRIP [18]: Una aplicación sencilla la cual presenta puntos turísticos de algunos países, también trabaja con realidad aumentada mediante la Geolocalización.

3.3. Aplicaciones sobre Geolocalización y Crowdsourcing

- OPENSTREETMAP [19]: es un proyecto colaborativo que tiene como objetivo desarrollar un sistema de mapas totalmente libre y abierto. Esta alternativa libre a Google Maps, que por ejemplo han adoptado servicios como Foursquare, se construye gracias a la colaboración de miles de personas de todo el mundo, empresas y entidades públicas que aportan cartografías, mapas y correcciones para hacer que este sistema sea cada vez más completo y aporte información de calidad a los usuarios. Con un dispositivo Android, es posible utilizar OSMTracker for Android para realizar anotaciones o aportes de correcciones a OpenStreetMap. Con un dispositivo con GPS el usuario puede localizar las coordenadas de la ruta que realizará y efectuar anotaciones de puntos de interés, dirección de la circulación de las calles por las que transita, adjuntar fotos o notas escritas o de voz. Una vez finalizada la ruta, el archivo GPX que se genere puede llevarse a OpenStreetMap con las herramientas que esta comunidad pone a disposición de los usuarios y así realizar un aporte al proyecto.
- OPENSIGNAL [20]: Esta aplicación permite que muchos usuarios colaboren con datos para generar mapas. Con la idea de crear un gran mapa de cobertura móvil y también de cobertura Wi-Fi nació el proyecto OpenSignal. Este proyecto se fundamenta en el crowdsourcing y convierte a los dispositivos móviles en sondas de medida que sirven para realizar mediciones de cobertura y de la calidad de servicio que se percibe, por ejemplo, al realizar una pequeña descarga de prueba. Todos estos datos, sumados a los de redes Wi-Fi abiertas, forman una gran base de datos mundial en la es posible encontrar información de 824.297 estaciones base, 825 operadores de todo el mundo, más de 5 millones de medidas realizadas por los usuarios y más de 1.200 millones de redes Wi-Fi catalogadas que servirán para ofrecer una idea de la cobertura o el servicio disponible en casi cualquier rincón del planeta.
- WEDDAR [21]: Mediante el crowdsourcing, Weddar plantea a los usuarios una serie de "descripciones sencillas" que definen el estado climático de su localidad; un dato que es posible subir al servicio y así aportar cómo el estado climático en una zona de manera sencilla y, sobre todo, muy visual. Weddar, una aplicación para iOS gratuita, las preguntas del tipo: ¿Hace buen tiempo? ¿Hace demasiado calor? ¿Hace mucho frío?, son respondidas gracias a las aportaciones de otros

usuarios que aportan sus respuestas al servicio. Esta aplicación se muestra en la figura 3.



Figura 3. Aplicación WEDDAR.

- USHAHIDI [22] es una aplicación en Internet que permite mapear información vital en zonas de catástrofe o de conflicto. Combina información sobre activismo social, periodismo ciudadano y nuevas tecnologías de información geoespacial.
- HOLLABACK [23]: Esta aplicación fue desarrollada para una organización sin fines de lucro presente en 34 ciudades, de 14 países, en 8 lenguas diferentes. Hollaback es un movimiento para acabar con el acoso en las calles alimentado por una red de activistas locales en todo el mundo. Mediante el crowdsourcing que ofrece la aplicación, se utiliza para promover conversaciones públicas, y desarrollar estrategias innovadoras para garantizar la igualdad de acceso a los espacios públicos. Permite geolocalizar la información enviada a través de mensajes de texto SMS para incluir en un mapa en donde recopilar todos estos datos. Esta aplicación, también ofrece capacitación gratuita para quienes quieran crear su propio sitio en su barrio o ciudad. Con esta finalidad se brindan "talleres educativos en las escuelas, universidades y grupos de la comunidad, para comprometer a los ciudadanos a través de medios tradicionales y sociales" [23].

4. Análisis de las Aplicaciones relevadas

En esta sección se listan las características destacables de las aplicaciones relevadas en la sección 3. Por cada característica se consignan las referencias de las aplicaciones que la implementa:

- Poseen la opción de posición actual de la ubicación del usuario con los puntos referentes más destacados a nivel importancia en el mapa visualizado, es decir, donde los usuarios móviles se encuentran en el momento en que ejecutan la aplicación [10], [11], [12], [13] y [17].
- Para cada lugar referenciado en estas aplicaciones, se obtienen servicios en base a una determinada ubicación, siendo: restaurantes cercanos [10], [12], [15] y [17], información de lugares turísticos [12], [13], [15] y [17], opiniones o comentarios sobre un local comercial [13], estado del clima en un lugar [21], localización de amigos por zonas [14], etc.
- La mayoría de estas aplicaciones se manejan en tiempo real, para una mejor interacción con los usuarios que las emplean.
- Las aplicaciones relevadas en las secciones 3.1 y 3.2 de crowdsourcing se alimentan de las opiniones y/o datos cargados por otros usuarios, es decir, cada una de estas aplicaciones proporciona una base de datos constantemente actualizada por ellos mismos.
- La colaboración de todos los usuarios forma parte de la utilización de estas aplicaciones, creando un efecto de valor agregado en la persona que realiza un aporte [6], [22] y [23]. Un ejemplo de esto es la contribución en los proyectos de software [7], donde las personas no sólo pueden participar a nivel colaborativo aportando ideas básicas para soluciones de desarrollo de aplicaciones móviles, sino que además, pueden obtener un empleo por parte de las empresas que proponen estos proyectos.
- La mayoría de estas aplicaciones permiten utilizar la totalidad de la pantalla, para visualizar los mapas, sin visualizar demasiadas opciones que distraen y ocupan espacio en la misma [10], [11], [17], [18], [19] y [20].
- Fácil acceso desde los dispositivos móviles.
- Mediante estas aplicaciones se puede acceder a varios servicios on-line gratuitos [8], [9], [14], [21] y [22].
- Opciones o funciones bien definidas.
- Compartir información entre numerosas personas.
- Servicios Off-line. Un ejemplo de esto son las descargas de los archivos en los dispositivos móviles, como ser: download de mapas geográficos de determinados recorridos turísticos [16].
- Las aplicaciones de las secciones 3.2 se encuentran orientadas a ofrecer mayormente, servicios en los sectores de turismo, trasporte o bien comercialización.
- Las aplicaciones de las secciones 3.3 se enfocan en cubrir necesidades puntuales de los usuarios que la utilizan, por ejemplo: localizar zonas de catástrofe o críticas frente a un determinado problema [22] o bien para garantizar la igualdad de acceso en los espacios públicos para todas las personas [23]. Esta clasificación intenta promover diferentes estrategias de comunicación y acción, generando una especie de

alerta entre los ciudadanos de las áreas geográficas afectadas por determinados motivos.

5. Buenas Prácticas de Diseño y Desarrollo

En esta sección se exponen las pautas a considerar al momento de diseñar y desarrollar una aplicación móvil. Las cuales se fundamentan en las guías ofrecidas por la W3C (World Wide Web Consortium) [26], [27], [29], [33], [34]; comunidad internacional que desarrolla normas en forma abierta, que están orientadas a crear estándares que brindan una mejor calidad en el software que es creado bajo estas. A esto se le suma también pautas basadas en el análisis de aplicaciones del dominio de estudio (basadas en crowdsourcing y geolocalización).

Las pautas resultantes son:

- Realizar un diseño de aplicación que esté orientado para varios dispositivos móviles. De esta forma, se logra la interoperabilidad.
- Debe ser sencillo, ya que es más conveniente para una mejor usabilidad del software realizado. El concepto de usabilidad implica la capacidad de adquirir fácilmente experiencia al interactuar con la aplicación móvil [28].
- Seleccionar una tecnología para desarrollar que permita visualizar funciones estándares o compartidas entre varios dispositivos móviles.
- Facilidad en el acceso y navegación de la aplicación.
- Verificación de una correcta visualización de todos los objetos contenidos en la aplicación.
- Tener en cuenta el tiempo de acceso en cada una de las páginas o secciones a navegar.
- Utilizar los mejores recursos posibles a nivel red para facilitar una adecuada usabilidad.
- Facilitar la correcta entrada de datos.
- Utilización de funciones para reducir el tiempo de acceso a la aplicación [33].
- Utilización de contextos amigables a nivel interfaz [33].
- Tener en cuenta los principios definidos en la etapa de diseño móvil, para desarrollar aplicaciones de calidad a nivel programación [33].
- Mantener un diseño flexible de la aplicación [33].
- Aprovechar las funcionalidades del dispositivo móvil [33].
- Los tiempos de los usuarios móviles son demasiado cortos, es necesario tener este punto en cuenta para que los datos más relevantes sean los más visuales en la aplicación.
- Considerar que el tamaño de pantalla de los dispositivos móviles es reducida. Con lo cual se debe prever un diseño de botones y otros elementos que sean cómodos en este tipo de pantalla.
- Permitir fácil acceso a las opciones de las principales funciones de la aplicación. La aplicación debería ser "amigable".
- Tener en cuenta las imágenes que serán incorporadas, ya que esto enlentecerá a la aplicación. Sólo se incorporarán imágenes que sean necesarias para la aplicación, reduciendo la cantidad de elementos decorativos y cuidando el tamaño de los mismos.
- Incorporar mapas que puedan ser visualizados lo más rápido posible, ya que debe tenerse en cuenta que los usuarios móviles no poseen mucha paciencia a la hora de esperar el tiempo de respuesta.
- Facilitar el uso de las búsquedas en mapas, "la paciencia de los usuarios de dispositivos móviles es nula, ridícula y casi inexistente. […] hay que compensar esta falta de interés con algo que se conoce como "búsqueda predictiva". Un ejemplo de búsqueda predictiva es GOOGLE INSTANT [30], con el que Google completa tus búsquedas mientras estás escribiendo y te ofrece sugerencias en función de tu historial, tu ubicación y la tendencia del momento." [31]
- No colocar imágenes de referencia en determinados puntos del mapa, demasiado grandes, ya que pierde el sentido de visualización de la información a nivel general en la cartografía.
- Evitar la sobrecarga de información en la visualización de los mapas. "Contar con demasiada información para tomar una decisión o permanecer informado sobre un determinado tema, puede generar contradicciones en la información disponible, una relación señal-ruido, dificultando la información que es realmente relevante" [32]. Es necesario excluir la publicidad, ya que la búsqueda de información se hace dificultosa y la clasificación de lo que se necesita en forma puntual lleva mucho trabajo.
- Desarrollar aplicaciones lo más estándares posibles, es decir, debe ser accesible a la mayor cantidad de plataformas de dispositivos (Android, iOS, Windows Phone, etc.).
- No utilizar elementos de flash ya que no son compatibles con la mayor parte de los dispositivos móviles.
- Utilizar la funcionalidad de posición actual, ya que es muy probable que los usuarios móviles quieran localizar servicios de su alrededor geolocal actual.
- Para compartir o colaborar información en la aplicación, deben ser usuarios registrados, con el fin de tener los datos de las personas que realizan aportes.
- Limitar la cantidad de información colaborativa a cargar en la aplicación.

- Editar información compartida por los usuarios registrados.

6. Estrategias para el diseño y desarrollo de las Aplicaciones

La sección 5 propone un listado sintetizado de buenas prácticas tanto destinadas a las aplicaciones móviles y además consideraciones para este dominio en particular. No obstante al momento de construir la aplicación será importante planificar una metodología de trabajo adecuada a las necesidades que se proyectan y a los recursos actuales disponibles en el equipo de trabajo.

"Las metodologías ágiles son una excelente alternativa para guiar proyectos de desarrollo de software de tamaño reducido, como es el caso de las aplicaciones para dispositivos móviles." [25]

Como metodología ágil, Mobile-D [24] resulta ser una alternativa relevante para el desarrollo de software de aplicaciones móviles. Mobile-D se basa en cinco fases: exploración, iniciación, producción, estabilización y prueba. Las especificaciones de esta metodología están disponibles en línea [24].

A nivel desarrollo de aplicaciones de crowdsourcing es necesario realizar un repositorio de datos, con el fin de proporcionar un "*home*" compartido para los usuarios voluntarios que contribuyen con estos en la aplicación [35]. Para este tipo de aplicaciones es necesario incorporar un modelo de diseño con procesos de evaluación y validaciones especiales del contenido brindado por el usuario. Esto se debe a que dicho contenido, es reutilizado por otros usuarios para continuar interactuando con la aplicación móvil a nivel colaborativo o bien utilizarla para la toma de decisiones en base a esta. Con el fin de evitar información errónea, ya que esta información aportada es la base del crowdsourcing en el software, deben existir estructuras de participación que controlen la calidad del contenido presentado.

Otra decisión importante es, en el caso de aplicaciones nativas, la elección del sistema operativo a considerar. En este trabajo se eligió Android por ser uno de los sistemas operativos más masivos que existe en la actualidad. En base a los datos provistos por Kantar World Panel [36], se puede observar que el liderazgo de Android en smartphones, pasó de tener una participación del 50,3% en septiembre de 2012 a un 73,4% en el 2013. En la figura 4 se presentan los porcentajes de inserción en América Latina de los sistemas operativos móviles [37].

El desarrollo de los prototipos que serán expuesto en la siguiente sección de este artículo, fueron enfocados en Android, ya que brinda soporte directo con la plataforma de Google (appengine), la cual se utilizará para la implementación a nivel Geolocalización como trabajo futuro propuesto.

Otro punto técnico a tener en cuenta es el método de localización en exteriores para los dispositivos móviles, el mismo podrá realizarse con tecnología Wi-Fi y receptores GPS (Global Positioning System). En caso que el dispositivo móvil no cuente con GPS, se podrá aprovechar la multilateralización (triangulación) de las señales de radio entre (varias) torres de radio de la red y el dispositivo móvil propiamente dicho.

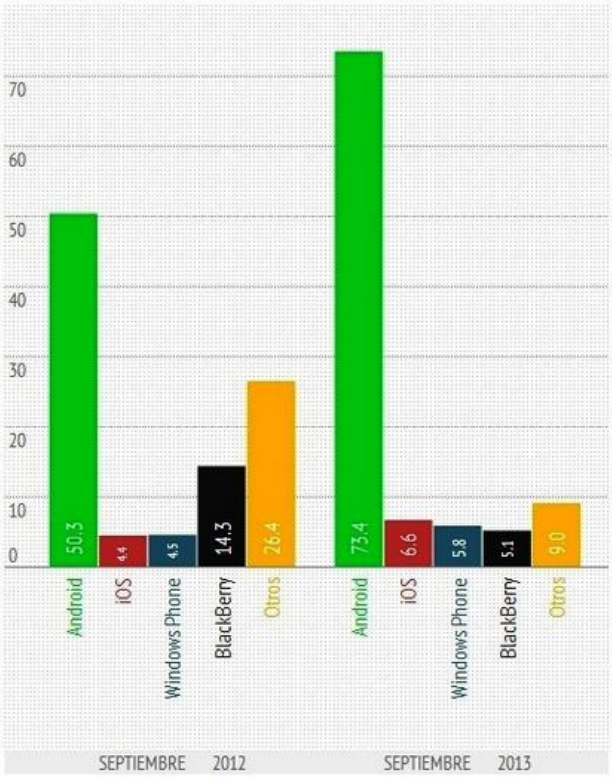


Figura 4. Estadísticas del crecimiento de los Sistemas operativos móviles en América Latina.

7. Prototipado

Teniendo en cuenta las estrategias planteadas y características más sobresalientes de las aplicaciones móviles investigadas para geolocalización y crowdsourcing, se desarrolló un prototipo de aplicación llamada *Hourus*, su lema *Horus Traveling Companion*, es decir, compañero de viaje. Hourus se basa en la localización de puntos turísticos más destacados de la zona identificada, ya sea desde la posición actual en la que se encuentra el usuario con su dispositivo móvil, o bien mediante la búsqueda realizada de un determinado lugar puntual.

El objetivo de este prototipo es enfocar las funcionalidades que se caracterizan en las aplicaciones

que utilizan crowdsourcing, puntualmente abarcar la colaboración de los distintos usuarios que interactúen con esta.

Opciones principales del prototipo de aplicación:

- *Opción Dónde estoy?*: me permite acceder a mi ubicación actual desde el dispositivo móvil.
- *Opción Qué visitar?*: esta opción pretende mostrar los distintos lugares turísticos más relevantes de una determinada zona, que bien el usuario podrá seleccionar. El usuario podrá optar si desea que sea la zona de la posición actual en la que se encuentra, o bien de un determinado lugar geográfico a localizar por medio de un buscador.
- *Opción Recorridos:* esta opción muestra un buscador con identificación de los datos desde la posición actual en donde se encuentra el usuario hasta un determinado destino que ingrese este. La aplicación mostrará las opciones brindadas por los usuarios que realizaron la colaboración de aportes de los recorridos desde la zona cercana del usuario hasta determinados puntos turísticos que se encuentren cerca del destino que indicó el usuario que realizó la búsqueda.
- *Opción Mi Perfil:* se podrá realizar el logueo de un usuario ya registrado o bien la registración de un usuario inexistente.
- *Opción Mis Lugares:* en esta opción se muestran los lugares favoritos de un usuario que previamente se encuentra registrado y logueado.
- *Opción Servicios Urgencias:* se visualizan los datos básicos para las urgencias básicas: llamado a 911, bomberos o bien policía del área cerca a la posición actual del usuario.

La aplicación prototipo tendrá la posibilidad de logueo, es decir, al acceder a la misma, se podrá optar por ingresar o navegar como un usuario visita, o bien se podrá acceder iniciando sesión como un usuario registrado. Dependiendo del tipo de usuario que interactúe con la aplicación, las opciones a visualizar serán diferentes.

Los tipos de usuarios que podrán acceder a la aplicación serán: (1) El tipo de usuario administrador, este usuario podrá acceder a opciones más técnicas como ser visualización y edición de la información colaborativa, brindada por parte de la comunidad de usuarios que la utilicen. Además, tendrá permisos especiales de ediciones de los comentarios y rutas turísticas identificadas como sugerencias que realizaron otras personas. El administrador podrá modificar los comentarios realizados por los usuarios, ya que pueden estar erróneos o bien recibir una baja puntuación a nivel recomendación por parte de los usuarios logueados. El prototipo contará con un sistema de conteo de puntuación por cada lugar turístico agregado y/o existente, es decir, cada usuario registrado en la aplicación, podrá puntuar el recorrido y comentarios agregados por otro usuario. De esta forma, se valorizará más el aporte brindado por una persona, cobrando una mayor fuerza con respecto a la sugerencia realizada. Al obtener mayor puntaje frente a un determinado punto turístico o recomendación, más credibilidad tendrá la información brindada por la aplicación, es decir, más usuarios confiarán en lo indicado y/o calificado. (2) Usuario registrado, este usuario podrá acceder a la mayoría de las opciones brindadas por el software, con la posibilidad de editar comentarios anteriores que éste haya realizado, es decir, que sean propios, no así comentarios de otros usuarios o bien incorporaciones de recorridos turísticos en la aplicación. (3) Usuarios invitados, estos sólo accederán a las opciones estándares, como ser: Opción Dónde estoy?, Opción que Visitar?, sin posibilidad de agregar comentarios al respecto. Opción Recorridos, sin posibilidad de agregar, modificar o bien descargar en forma off-line un determinado mapa. Opción Mi Perfil, con el fin de loguearse o bien crear una cuenta en caso que no posea. Y finalmente, acceder a la opción Servicios Urgencias.

Si ingresamos a la aplicación como usuario registrado, se visualizarán los lugares favoritos indicados por este, en el mismo se podrá acceder desde la opción Mis Lugares (ver figura 5.a), con el fin de que pueda acceder a modo de acceso directo a la información más relevante para este.

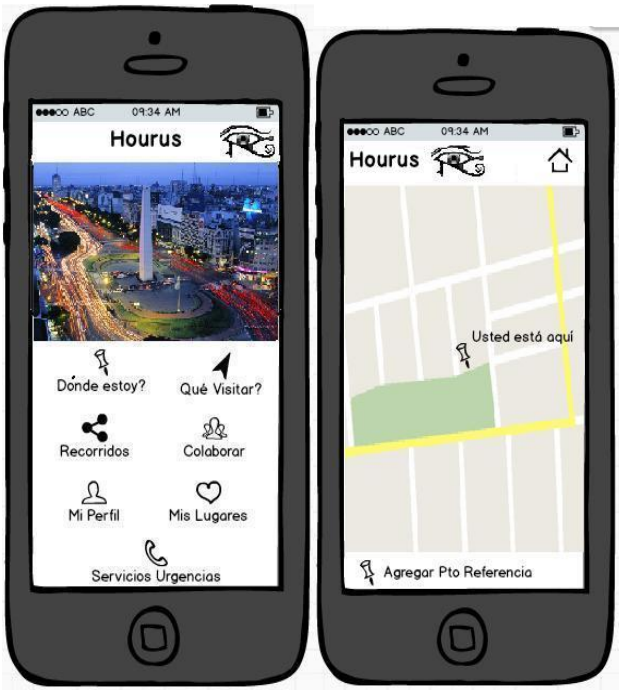


Figura 5. Prototipo de Aplicación Hourus: Figura 5.a) Pantalla inicial de la aplicación. Figura 5.b) Mapa de posicionamiento.

Para la registración de los usuarios (ver figura 5.a), cada uno deberá indicar los siguientes datos:

- Nombre de usuario.
- Contraseña de usuario.
- Foto de perfil en la sesión de la aplicación.
- Dirección de mail para verificar la cuenta de registración, se realiza validación de la misma.
- Identificación de los temas de interés (opcional).

En la figura 5 se muestran dos capturas de pantallas de este prototipo. En la Figura 5.a se representa la pantalla inicial de la aplicación. En la figura 5.b se muestra la opción "Dónde estoy?".

Algunas de las búsquedas que se podrán realizar en la aplicación serán: museos, restaurantes, zonas destacadas como turísticas, monumentos referentes en las áreas georeferenciadas, etc. Esto es, se podrá localizar información, no sólo de algún tipo de estos filtros que se podrán aplicar en el mapa que se visualizará, sino que además, se permitirá a los usuarios colaborar a nivel comentarios para cada uno de estos, según las búsquedas realizadas.

En la Figura 6.a se representa una búsqueda sobre Museos de Buenos Aires. Se muestran los resultados encontrados con las opciones de localizar en el mapa un determinado museo que le interese al usuario. Por otra parte, se muestran las imágenes del museo. Además, se brinda la opción de colaborar con este museo, es decir, se podrán realizar aportes de información al respecto, indicando comentarios sobre la experiencia obtenida con este, o bien cargar una foto del mismo, como se indicó anteriormente, el usuario debe estar logueado.

La figura 6.b) muestra la opción Localización de los Museos de Buenos Aires (se accede desde icono de la izquierda inferior de la pantalla anterior). En esta pantalla se ven los puntos de referencia de cada uno de los museos para la zona que buscó el usuario. También, se podrá aplicar colaboración, ya que se muestra la opción Agregar Pto Referencia, esta opción permite abrir una nueva ventana donde se indicarán los datos de un nuevo museo (para este caso, ya que se podrán localizar otros puntos de interés turísticos, como ser Iglesias, Monumentos, etc.), se solicitará datos identificados con el punto de referencia a agregar, como ser: nombre, dirección, teléfono, latitud, longitud (o bien marcado directo desde el mapa), carga de imagen y brindar algún comentario al respecto. Finalmente se podrá indicar un puntaje a dicho lugar (ver figura 7.b). Esta forma de colaboración, es decir, de aportar información desde la aplicación, también será posible desde la opción Colaborar de la figura 5.a.

En la figura 7.a se representa la pantalla "Qué visitar?", la cual muestra opciones para tildar y visualizar en el mapa. En la figura 7.b se muestra la opción "Colaborar", en esta pantalla se ve como el usuario puede aportar información sobre un nuevo Cajero automático cargando determinados datos del mismo, con el fin de brindar su aporte al resto de los usuarios de la aplicación Hourus.

El aporte que se muestra en la figura 7, es otro aspecto con respecto a la recomendación de puntos o lugares de interés en la ciudad de Buenos Aires cargados o colaborados por los usuarios, los cuales pueden ser compartidos entre estos mediante la aplicación. Algunos de los puntos que se muestren en la figura 7.a son filtrados de los puntos de referencias por: Restaurantes; Bares; Cines; Galerías de Arte; Museos; Bancos; Cajeros Automáticos.

Para cada uno de estos filtros se podrá tildar uno o más de un ítem de búsqueda. Dependiendo de los datos tildados los mismos se mostrarán en el mapa a nivel servicio georeferencial. Por ejemplo: si tildamos la opción Bancos y Restaurantes, se visualizará en el mapa todos los Bancos y Restaurantes de la zona de la posición actual en la que se encuentre el usuario móvil, en el momento que esté accediendo desde el dispositivo móvil utilizado.

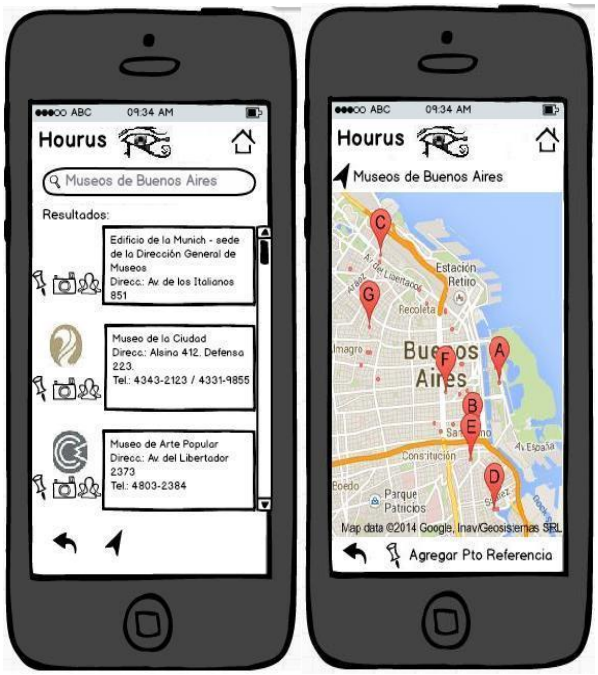


Figura 6. Prototipo de Aplicación Hourus: Figura 6.a) Pantalla de búsqueda. Figura 6.b) Localización de resultados.

Otros requerimientos funcionales que tendrá la aplicación es que brindará el acceso a la aplicación Google Goggles [38], en la cual se podrá acceder a distinta información apuntando con la cámara del dispositivo móvil a un determinado elemento (un cuadro en un museo con el fin de facilitar información al

respecto, un edificio histórico para conocer sobre su arquitectura o bien su historia, etc.), si Goggles encuentra dicho objeto en su base de datos, mostrará toda la información que posea, siendo texto y por lo general imágenes. En caso que no lo encuentre en su base de datos registrado, se visualiza un mensaje de objeto no encontrado. Esta parte del prototipo no pretende ser colaborativa, ya que es una funcionalidad con relación de acceso directo a la aplicación de Goggles.



Figura 7. Prototipo de Aplicación Hourus: Figura 7.a) Pantalla de Qué Visitar?. Figura 7.b) Opción de Colaboración.

A modo de resumen, este prototipo pretende brindar rutas turísticas con la señalización de los puntos de interés. Como se muestra en la figura 6.b), se pueden agregar nuevos puntos de referencias, realizando un aporte de colaboración de información para todos los usuarios que utilizan o bien utilizarán la aplicación. Para cada uno de estos puntos se aportará: información breve, imagen, experiencias o comentarios de los usuarios, e identificación de los tiempos de trayecto. Mediante éste último dato, se puede acceder al recorrido más corto de un lugar indicado o bien ver más opciones para llegar al destino (opción Ver más recorridos).

8. Trabajo a futuro

Continuar con el desarrollo del prototipo descripto en la sección 7, y ofrecer nuevas opciones, entre ellas la comunicación entre otras aplicaciones. Se procederá a incorporar tanto en el diseño, como en el desarrollo, los siguientes enfoques:

- Posibilitar a los usuarios ver las experiencias de otras personas hacia la ruta turística o bien cargar su propia experiencia vivida, por ejemplo, hoteles donde se alojaron por esa ruta indicada, ingresar comentarios, fotos, o bien puntaje del mismo. (Lograr una interacción con Redes Sociales: Google Local y/o bien TripAdvisor).
- Brindar la opción de realidad aumentada en folletos entregados a las personas, con el fin de conocer la futura aplicación. Mediante un patrón en papel, enfocar con la cámara de celular y mostraría realidad aumentada, la cual visualizaría imágenes de diferentes zonas urbanas de la ciudad de Buenos Aires (en principio), mostraría comidas típicas, lugares históricos, puntos referentes, etc.
- Interactuar con más redes sociales como ser: Facebook, YouTube, Twitter y Picasa.

Por otra parte se prevee analizar la información que puede ser suministrada por los usuarios, como validarla y ofrecer un sistema de recompensas que permita estimular a los usuarios que provean al sistema información valiosa.

9. Conclusiones

La geolocalización permite identificar puntos de interés cercanos al usuario, guardar registro de los lugares transitados, de los objetos de interés, etc. Al combinarlo con realidad aumentada se suma poderío a las aplicaciones, donde el usuario puede con la cámara de un celular obtener información sobre un objeto, edificio, etc. Por otra parte es importante mantener actualizados los datos que utilizan estas aplicaciones, aquí la colaboración de los propios usuarios es fundamental por ello el crowdsourcing cobra un gran interés.

Es importante destacar que los dispositivos móviles cuentan actualmente con un gran poderío en cuanto a hardware, los Smartphone cuentan con Cámara, GPS, Brújula (lo que ayuda a contribuir en determinar el sentido al que apunta el celular y de este modo por ejemplo, conocer si el usuario está de espalda a un determinado objeto). Las posibilidades de conectividad mediante redes Wi-Fi o planes de datos son cada vez mayores esto permite que los usuarios móviles estén conectados con gran frecuencia y puedan compartir sus opiniones en redes sociales ó en aplicaciones instaladas. La combinación de Geolocalización, Crowdsourcing y Realidad Aumentada permitirá en el campo del turismo planificar una aplicación que permita acompañar al turista en su recorrido brindando información de interés. Esto trae aparejado diversos retos: que sea usable en dispositivos que tienen pantalla reducida; planificar el acceso a las opciones más importantes de forma ágil y facilitar la visualización. También proveer estrategias de validación de la información provista por los usuarios es

fundamental para que la aplicación sea útil. Si algo es peor a no contar con información que un usuario necesita, es brindar datos falsos que causen que el usuario tome una mala decisión. Esto conllevaría a que la aplicación tenga un corto tiempo de vida útil a pesar de que su interfaz tenga un diseño excelente. Es decir esta temática requiere considerar diversos aspectos y plantea diversos desafíos. Es por ello, la necesidad de aplicar una correcta metodología de trabajo para este tipo de aplicaciones en dispositivos móviles y lograr seguir las buenas prácticas de la W3C para el desarrollo de software móvil, ya que facilita el diseño y desarrollo eficiente con el adecuado aprovechamiento de los recursos disponibles, tanto a nivel hardware como software.

10. Referencias

[1] Brabham D. C., *Moving the crowd at iStockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application*, First Monday, 2008, vol. 13, no 6.

[2] Kazai, Gabriella., "In search of quality in crowdsourcing for search engine evaluation", *Advances in information retrieval*, Springer Berlin Heidelberg, 2011, p. 165-176.

[3] Estellés Arolas, E. González Ladrón-De-Guevara, "Towards an integrated crowdsourcing definition", *Journal of Information science*, 2012, vol. 38, no 2, p. 189-200.

[4] Jan Marco Leimeister Michael Huber Ulrich, Bretschneider, Helmut Krcmar, "Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competitions", *Journal of Management Information Systems*, 2009.

[5] Carlos Gómez Abajo, "Una nueva fórmula aumenta la fiabilidad del crowdsourcing", *Revista Electrónica de Ciencia, Tecnología, Sociedad y Cultura*, ISSN 2174-6850, 2014, http://www.tendencias21.net/Una-nueva-formula-aumenta-la-fiabilidad-del-crowdsourcing_a13738.html.

[6] Berckeley, Universidad de California, "*BOINC, Programas de código abierto para computación voluntaria y computación en red (grid)"*, 2014, http://boinc.berkeley.edu/.

[7] YEEPLY, "*Contrata a los mejores expertos para desarrollar tu aplicación móvil*", 2014, https://www.yeeply.com.

[8] DUOLINGO, "*Aprende un idioma gratis y diviértete*", 2014, https://es.duolingo.com/.

[9] WIKIPEDIA, "*La enciclopedia libre*", 2014, http://es.wikipedia.org/wiki/Wikipedia:Portada.

[10] ACROSSAIR, "*About*", 2014, http://www.acrossair.com/about.htm.

[11] LAYAR, "*The World's #1 Augmented Reality Plataform*", 2014, https://www.layar.com.

[12] MINUBE, "*Tu próximo viaje comienza aquí hoteles, restaurantes y sitios que ver recomendados por millones de viajeros*", 2014, http://www.minube.com.ar/.

[13] TRIPADVISOR, "*Planificá tu viaje perfecto*", 2014, http://www.tripadvisor.com.ar/.

[14] TRIPIT, *"The all-in-one, travel organizer"*, 2014, https://www.tripit.com/.

[15] TRIPJOURNAL, "*About trip journal*, 2014, http://www.trip-journal.com/.

[16] TOURISTEYE, "*Plan your trip in seconds. Use our mobile apps to discover the best places and tips*", 2014, http://www.touristeye.com/.

[17] FOURSQUARE, "*Presentamos el nuevo foursquare, que aprende lo que te gusta y te lleva a lugares que te encantarán"*, 2014, https://es.foursquare.com/.

[18] LABTRIP, "*¿Qué es labtrip?"*, 2014, http://blog.labtrip.com/que-es-labtrip/.

[19] OPENSTREETMAP, "*¡Bienvenido a Openstreetmap!"*, 2014, http://www.openstreetmap.org/.

[20] OPENSIGNAL, "*Find the best network in your area*", 2014, http://opensignal.com/.

[21] WEDDAR, "*How does it feel?*", 2014, http://www.weddar.com/.

[22] USHAHIDI, "*Ushahidi services*", 2014, http://www.ushahidi.com/services/

[23] HOLLABACK, "*You have the power to end street harassment*", 2014, http://www.ihollaback.org/.

[24] MOBILE-D, "*Mobile-D™ is VTT's methodology for agile software development.*", 2014, http://virtual.vtt.fi/virtual/agile/mobiled.html .

[25] Balaguera, Yohn Daniel Amaya, "Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual", *Journal Technology*, 2013, vol. 12, no 2, p. 111-124.

[26] W3C (World Wide Web Consortium), "*Sobre el W3C.*", 2014, http://www.w3c.es/Consorcio/.

[27] W3C (World Wide Web Consortium), "*Buenas Prácticas en web Móvil*", 2014, http://www.w3c.es/Divulgacion/Tarjetas/MWBP/ .

[28] Budiu, R., & Nielsen, J., "*Usability of Mobile Websites and Applications: Design Guidelines for Improving the User Experience of Mobile Sites and Apps*", Nielsen Norman Group, 2011.

[29] W3C (World Wide Web Consortium), "Mobile Web Best *Practices 1.0*", W3C Recommendation 29 July 2008, http://www.w3.org/TR/mobile-bp/.

[30] GOOGLE INSTANT, "*Acerca de Google Instant*", 2014, http://www.google.es/instant/.

[31] MARÍN, JUAN RAMÓN. "*SEO móvil: por qué adaptar tu estrategia de Posicionamiento a los Dispositivos Móviles*",

2012, http://www.eduardomartinezblog.com/seo-movil-las-reglas-han-cambiado/.

[32] TOFFLER, ALVIN. ALEU, J. FERRER. "*El "shock" del futuro*", Plaza & Janés, 1972.

[33] W3C (World Wide Web Consortium), "*Tarjetas de Buenas Prácticas en Aplicaciones para la Web Móvil*", 2014, http://www.w3c.es/Divulgacion/Tarjetas/MWABP/ .

[34] W3C (World Wide Web Consortium), "*Mobile Web Application Best Practices*", W3C Recommendation 14 December 2010, http://www.w3.org/TR/mwabp/.

[35] W3C (World Wide Web Consortium), "*Crowd-sourcing de datos?*", 2014, http://www.w3.org/People/shadi/Talks/2012/0301/CSUN/slide7-0.html .

[36] Kantar World Panel, "*High Definition Inspiration: Kantar Worldpanel es el líder mundial en conocimientos sobre el consumidor[…]*", 2014, http://www.kantarworldpanel.com/ar/Sobre-nosotros.

[37] Movilion, "*Estadísticas: sistemas operativos móviles en América Latina 2003*", 11 de noviembre 2013, http://www.movilion.com/estadisticas-sistemas-operativos-moviles-america-latina-2013/.

[38] GOOGLE GOGGLES, "*Información general y requisitos de Goggles*", 2014, https://support.google.com/websearch/answer/166331?hl=es.

Modelo de Vigilancia Tecnológica e Inteligencia Competitiva (VTeIC) Basado en Minería de Datos

Marcelo Karanik, Leonardo Wanderer, Fabián Levin
Grupo de Investigación Sobre Inteligencia Artificial
Universidad Tecnológica Nacional – Facultad Regional Resistencia
French 414 - (H3500CHJ) Resistencia - Chaco.
marcelo@frre.utn.edu.ar, {leonardo.wanderer; fabianlevin60} @gmail.com

Palabras clave: Vigilancia Tecnológica, Inteligencia Competitiva, Minería de Textos, Minería de Web.

Resumen

El formidable avance de las Tecnologías de la Información y las Comunicaciones ha generado que hoy en día las organizaciones tengan que analizar enormes cantidades de información para definir las estrategias y políticas que le permitan competir adecuadamente. El problema es que la información está diseminada en muchas fuentes y no siempre es de la calidad y utilidad requerida. La Vigilancia Tecnológica (VT) y la Inteligencia Competitiva (IC) proveen métodos y técnicas que permiten filtrar y convertir la información para que sea útil. Ambas, involucran procesos inteligentes e intentan detectar y predecir los cambios del entorno a fin de que la organización pueda articular mecanismos de anticipación. Este artículo describe un modelo de análisis de información que integra técnicas de Minería de Datos (específicamente Minería de Textos y Minería de Web) para examinar documentos de dominio público en la Web y obtener, a partir de ellos, información de valor para la toma de decisiones dentro de la organización.

1. Introducción

El éxito de una empresa depende de que sus clientes adquieran los productos y/o servicios a un precio tal que permita cubrir los costos y además genere el mayor margen posible de ganancia. Este concepto, a priori muy sencillo, tiene asociado múltiples variables inciertas (internas y externas) que condicionan el funcionamiento exitoso de la empresa. Las variables internas son aquellas sobre las cuales los integrantes de la organización tienen control (por ejemplo, la cantidad y calidad de los productos y/o servicios, la innovación de los procesos productivos, la estrategia de venta a utilizar, la imagen de la empresa, etc.). En cambio, las externas son las que están fuera del control de la organización (por ejemplo, la estrategia de la competencia, la innovación tecnológica, los proveedores, el contexto social, condiciones de mercado, etc.). Claramente, las variables externas afectan el funcionamiento de la empresa, condicionando el control sobre las variables internas. Por ejemplo, la innovación tecnológica afecta la manera de fabricar los productos y, por ende, no innovar en los procesos puede generar costos de producción elevados comparados con los de la competencia. Como consecuencia, las empresas deben estar atentas constantemente a la evolución de los factores externos para generar estrategias que la vuelvan competitiva o, en el peor de los casos, generar estrategias correctivas que le permitan seguir funcionando.

Actualmente hay dos actividades muy utilizadas que las empresas usan para analizar las variaciones de su entorno: la Vigilancia Tecnológica (VT) y la Inteligencia Competitiva (IC). Aunque que con distintos enfoques, ambos procesos producen información de alto valor para las decisiones estratégicas y proporcionan un mecanismo apropiado de alerta sobre las variaciones relevantes del entorno. La VT incluye aquellos métodos y técnicas que identifican y recopilan información de carácter científico-tecnológico potencialmente útil para la empresa [1]. Por otro lado, la IC contiene a aquellos métodos y técnicas que obtienen, dentro de un marco legal y ético, información pública sobre los competidores y otros agentes externos a la empresa, de modo que ésta pueda anticiparse a los cambios en su entorno [2]. Por tanto, la información suministrada por la VT y la IC se utiliza en el proceso de toma de decisiones directivas de la organización. Esta información combinada en efectivos Sistemas de Soporte de Decisiones (SSD) se transforma en ventajas competitivas que la empresa puede aprovechar en la interacción con su entorno. Estos sistemas concentran y procesan información ayudando a los decisores correspondientes a abordar problemas de Toma de Decisiones (TD), comúnmente estratégicos y tácticos del entorno empresarial [3-5]. Queda claro que la utilización eficaz de los métodos de VT e IC proporciona información de calidad que puede aprovecharse en los SSD para seleccionar las estrategias convenientes para la empresa; pero lograr reunir información útil para estos procesos no es una tarea sencilla debido a la gran cantidad de información disponible.

Existen técnicas de Inteligencia Artificial (IA) que pueden ser utilizadas para filtrar información disponible

en la web. Las técnicas de Minería de Datos (MD), concretamente Minería de la Web (MW) y Minería de Textos (MT) proveen herramientas para extraer conocimiento de grandes volúmenes de información [6], que son adecuadas para ser utilizadas en VT e IC.

Este artículo describe un modelo para realizar VT e IC basado en técnicas de MW y MT. En la Sección 2 se hace una descripción de las actividades relacionadas con la VT e IC. Luego, en la Sección 3 se mencionan algunas técnicas de MW, MT y SSD. El Framework de implementación y el modelo de capas se describen en la Sección 4. Finalmente, en la Sección 5, se discuten algunas características del modelo propuesto y se mencionan algunos trabajos actuales y futuros.

2. Vigilancia Tecnológica e Inteligencia Competitiva (VT e IC)

2.1. Vigilancia Tecnológica

En general la Vigilancia Tecnológica incluye todos aquellos métodos que intentan anticipar y entender la dirección potencial, las características y efectos de los cambios tecnológicos, especialmente en lo referido a invención, innovación y uso [1]. La VT es un proceso continuo de captación selectiva de información del exterior y de la propia organización referida a ciencia y tecnología, que se transforma luego en conocimiento para minimizar los riesgos en el proceso de toma de decisiones y anticipar los cambios del entorno. Es un proceso sistemático de búsqueda, recolección y análisis de información de acceso público tanto en el entorno de la organización como en el entorno de la competencia.

Lo que se intenta con la VT es detectar fuentes de información relevantes sobre tendencias tecnológicas, potenciales ventajas tecnológicas, tecnologías emergentes, aspectos regulatorios, restricciones de mercado, investigaciones, patentes, etc. Toda esta información es luego analizada y transferida a los decisores para que puedan delinear planes estratégicos de actualización y adecuación tecnológica de los procesos de la empresa. Del punto de vista práctico, la VT no es una actividad sencilla de realizar debido al dinamismo actual de la evolución de la ciencia y la tecnología. Ese dinamismo exige que la organización desarrolle y utilice mecanismos de actualización tecnológica constantemente. Los procesos de VT han sido utilizados durante mucho tiempo por las empresas pero tal vez no de manera formal y ordenada. Hoy en día el exceso de información y la facilidad de acceso a la misma a través de las TICs, exigen organizar y sistematizar los procesos de recolección, análisis y difusión del conocimiento dentro de la empresa.

Existen muchos métodos de VT los cuales pueden agruparse en las siguientes clases [7, 8]:

- *Métodos de monitoreo e inteligencia*: El objetivo del monitoreo es simplemente encontrar indicaciones de posibles desarrollos tecnológicos a futuro, a fin de anticiparse lo antes posible a los cambios [9]. Luego, estas indicaciones son analizadas por expertos en el área de desarrollo y determinan la importancia de las indicaciones detectadas.
- *Análisis de tendencias*: caracterizado por la predicción de situaciones futuras en base a los datos cuantitativos históricos. Para el análisis se utilizan técnicas de regresión, ajuste de curvas, etc.
- *Opinión de los expertos*: Implica la comprensión del desarrollo tecnológico a través de consultas intensivas con expertos en el área que se investiga. Uno de los métodos más utilizados es Delphi [10, 11].
- *Métodos estadísticos*: los métodos más populares son el análisis de correlación y el análisis bibliométrico. La utilización de estos métodos supone que los datos considerados como patrones de tecnologías existentes están disponibles para el análisis. Básicamente se intenta correlacionar los datos disponibles con ciertos patrones previamente definidos. La minería de textos [12] es una herramienta muy utilizada para este fin.
- *Métodos de modelado y simulación*: los modelos pueden utilizarse para simular el comportamiento futuro de los sistemas complejos. Claramente, la utilización de este tipo de métodos requiere un elevado esfuerzo para simular las interacciones de todas las variables del sistema, por lo que generalmente se intenta simplificar la visión del problema. Entre los métodos más utilizados se encuentra el modelado de Agentes que tiene como objetivos desarrollar aplicaciones que representen al usuario en la búsqueda de información según ciertas restricciones impuestas. El funcionamiento y la interacción con otros agentes (e incluso otros sistemas) se basa en la utilización de reglas de comportamiento [13].
- *Modelado de escenarios*: la descripción de diferentes escenarios que proponen concepciones del futuro de la tecnología es una herramienta muy útil si se basa en un conjunto sólido de presunciones. Luego es el experto quien analiza las características más representativas de cada uno y establece cuál de ellos tiene mayor probabilidad de ocurrencia. De la misma manera que en el modelado, los escenarios deben restringir la cantidad de variables a analizar ya que la sobresaturación de información puede generar distorsiones en el proceso de selección.

Todos estos métodos brindan herramientas que facilitan la captación, procesamiento, análisis, codificación y difusión de la información relevante para la organización. Pero el éxito de la anticipación dependerá de, además de la calidad de la información obtenida, el posterior procesamiento que realicen los SSD.

2.2. Inteligencia Competitiva

Como se ha mencionado anteriormente las empresas deben conocer como varía su entorno para poder adaptar sus políticas y estrategias. Dentro de ese entorno una variable que condiciona el accionar de la organización es la competencia. Debido al gran avance de las TICs las empresas tienen a su disposición grandes cantidades de datos de sus competidores que le permiten utilizar técnicas de IC para anticipar eventuales modificaciones o generar patrones de comportamiento para ajustar sus políticas.

La IC es un conjunto de métodos éticos y legales que una organización puede utilizar a fin de poder obtener información valiosa sobre sus competidores. Básicamente la IC tiene como objetivo conseguir información sobre las actividades de su entorno y anticipar el comportamiento futuro de los competidores, proveedores, clientes, mercados, productos y servicios [2].

Por lo general se suele confundir la IC con el espionaje. Si bien esta confusión está ligada con la recolección de información sobre la competencia, la diferencia es clara: no se intenta obtener información confidencial de los competidores sino simplemente captar, analizar y procesar información de dominio público que está disponible y al alcance de cualquier persona.

Muchas de las herramientas descriptas en la sección 2.1. se pueden utilizar para hacer IC. Existen tres etapas bien definidas en la IC [14]:

- *Recolección de información*: implica la utilización de métodos de detección de información relevante.
- *Extracción de información*: involucra la utilización de métodos de análisis de información para establecer la importancia de la información obtenida.
- *Contextualización de la información*: establece el contexto de utilización de la información (comercialización, producción, administración, etc.).

Según [15] los objetivos del uso de un sistema de información de IC pueden resumirse en tres aspectos:

- Mejorar la competitividad de la empresa.
- Predecir, con un alto nivel de confianza, la evolución del entorno (acciones de los competidores, requerimientos de los clientes e incluso la influencia de cambios políticos).
- Proveer un buen soporte para el proceso de decisión estratégica.

Evidentemente los procesos de VT e IC tienen grandes similitudes e intentan comprender y anticipar los cambios en el entorno con el objetivo de hacer que la organización no pierda competitividad.

3. Técnicas de Minería y Sistemas de Soporte de Decisiones

3.1. Minería de Textos y de la Web

Como se ha mencionado en las secciones anteriores, la gran evolución de las TICs ha provocado la generación y el fácil acceso a una extraordinaria cantidad de información. Este exceso de información afecta a los procesos de toma de decisiones ya que no existe manera de procesar todas las fuentes de manera eficiente. Por lo tanto se torna realmente necesario contar con herramientas que ayuden a discernir la información relevante de la que no lo es.

Una de las áreas en las que se estudian y desarrollan métodos para buscar información de valor en grandes volúmenes de datos se conoce como Minería de Datos (MD) [6]. Esta área es una rama de la Inteligencia Artificial (IA) que se enfoca en el desarrollo de algoritmos que permitan extraer conocimiento desde los datos. Una lista de métodos y técnicas de MD más utilizados pueden encontrarse en [16].

Entre esos métodos se destaca la utilización de redes neuronales, algoritmos genéticos, predicción dinámica, agentes inteligentes, clustering, reglas de asociación, árboles de decisión, análisis de correlación, análisis semántico, análisis de regresión, entre otros.

Desde el punto de vista de la VT y la IC, existen dos actividades que son de gran utilidad: la exploración y análisis de textos (MT) y la exploración y análisis de la información de la Web (MW).

MT es un campo interdisciplinar que involucra recuperación de información, Data Mining, aprendizaje de máquina, estadísticas y computación lingüística [17]. Los tipos de algoritmos que pertenecen al área son:

- Algoritmos para descubrir asociaciones de conceptos.
- Algoritmos para realizar clasificaciones.
- Algoritmos para descubrir clusters.

Estos algoritmos se utilizan básicamente para obtener información relevante de grandes volúmenes de texto. Existen muchos métodos de IA para clasificar, buscar asociaciones y descubrir clusters, entre ellos, los algoritmos bioinspirados y el análisis estadístico son de suma importancia y han brindado resultados muy satisfactorios. A pesar de esto, la exploración de nuevas técnicas y la propuesta de modificaciones de los algoritmos existentes es una actividad de relevancia.

La minería de Web, por otro lado, intenta determinar cuáles son, donde se encuentran y como acceder a los recursos disponibles en Internet. Básicamente MW es el uso de técnicas de MD para descubrir y extraer información de los documentos y recursos disponibles de la Web [18]. Esta actividad puede clasificarse en tres categorías [19]:

- Minería del contenido Web.
- Minería de la estructura de la Web.
- Minería del uso de la Web.

De estas tres categorías, la más relevante para la VT y la IC es la minería del contenido Web, la cual abarca el

descubrimiento de recursos [20], categorización y clustering de documentos [21] y extracción de información de páginas Web [22].

Desde el punto de vista del análisis y procesamiento de la información pública disponible en la Web, los algoritmos de MW proveen las herramientas adecuadas para monitorear los cambios del entorno de la organización. Debido a que el dinamismo constante de la Web aporta complejidad al problema de la búsqueda de información, se necesitan técnicas que puedan adaptarse a la variabilidad de las condiciones de la Web. Esto provoca que los métodos para MW requieran de un alto grado de inteligencia para poder retornar información de utilidad.

Evidentemente, contar con algoritmos confiables y eficientes para la exploración de textos y de la Web permite determinar el valor correcto que tiene la información analizada para una organización. Por lo tanto, se justifica el desarrollo modelos de análisis de información que utilicen MT y MW para VT e IC.

3.2. Sistemas de Soporte de Decisiones

Más allá de los procesos mencionados en las secciones anteriores, es claro que se necesitan sistemas que den apoyo a la toma de decisiones que analicen la información derivada de la VT y de la IC. Esas aplicaciones se conocen como Sistemas de Soporte de Decisiones y proveen una serie de técnicas y herramientas que sirven para asistir al decisor durante el proceso de TD [3-5]. Los SSD pertenecen al área de la ingeniería de sistemas que tiene como objetivos dar soporte y mejorar las decisiones gerenciales, ocupándose del desarrollo y la implementación de estos sistemas. Si bien es un área de tecnología de la información relativamente nueva, se han desarrollado múltiples herramientas de uso específico.

Arnott y Pervan [23] clasifican a los SSD en: Sistemas de Soporte de Decisiones Personales, Sistemas de Soporte de Grupos, Sistemas de Soporte de Negociaciones, Sistemas de Soporte de Decisiones Inteligentes, Sistemas de Soporte de Decisiones basados en Conocimiento Gerencial, Data Warehousing y Sistemas de Reporte y Análisis Empresariales. En cada una de esas sub-categorías se desarrollan e implementan las herramientas y técnicas particulares para cubrir las demandas de cada tipo de organización. Los Sistemas de Soporte de Decisiones Personales son aplicaciones pequeñas destinadas a dar soporte a los decisores en tareas de TD cotidianas [24]. Los Sistemas de Soporte de Grupos [25-27] y los Sistemas de Soporte de Negociaciones [28] son sistemas que implementan técnicas de manejo de decisiones grupales, ya sean en ambientes colaborativos o competitivos. Los Sistemas de Soporte de Decisiones Inteligentes [29-32] y los Sistemas de Soporte de Decisiones basados en Conocimiento Gerencial son sistemas que implementan modelos matemáticos y las técnicas de Inteligencia Artificial para mejorar la calidad de análisis de los entornos. Los sistemas de Data Warehousing y los Sistemas de Reporte y Análisis Empresariales [33, 34] son los más utilizados en la práctica, ya que incluyen aplicaciones para el tratamiento de grandes volúmenes de datos y herramientas específicas como los Sistemas de Información Ejecutivos, Sistemas de Inteligencia de Negocios y Sistemas de Administración de Prestaciones Corporativas.

Aunque los SSD requieren hoy día un fuerte componente computacional, el desarrollo y la implementación de los modelos matemáticos siguen siendo la clave del éxito de este tipo de sistemas. Esto significa que es de primordial importancia que los SSD se fundamenten en sólidos modelos que permitan manejar y tratar las variables relevantes del proceso de TD. En este sentido, los modelos cuantitativos han tenido gran relevancia en el desarrollo de los SSD ya que suministran resultados que el decisor puede interpretar fácilmente. Entre ellos se destacan los modelos basados en ecuaciones algebraicas y diferenciales, las herramientas de análisis como las matrices y árboles de decisión, los métodos de análisis de decisión multi-criterio y los modelos de predicción.

Es evidente que las necesidades del decisor determinan los tipos de herramientas que se deben implementar en un SSD. Dado que la estructura y el grado de incertidumbre del problema, mencionadas en la Sección 1.1, son factores propios del entorno, gran parte del esfuerzo en el diseño de los SSD se ha orientado a desarrollar métodos y técnicas relacionadas con el tratamiento de las limitaciones cognitivas del decisor. En este sentido, los métodos análisis de decisión multi-criterio han sido muy útiles para evaluar y valorar las alternativas disponibles, ya que consideran varios criterios simultáneamente durante el proceso de TD. En general, estos métodos entregan como resultado un ranking de preferencias de las alternativas evaluadas, tomando como base a dos o más criterios. Debido a que la valoración de los criterios es propia del decisor, el resultado obtenido es acorde a los juicios subjetivos expresados. Por ende, la utilización de este tipo de métodos es una alternativa eficiente para diversos contextos.

4. Modelo Propuesto

4.1. Framework

La propuesta de este trabajo busca definir framework que permita a las organizaciones captar la información que le sea relevante, analizarla y presentarla, de manera tal que las decisiones estratégicas puedan ser tomadas con un grado menor de incertidumbre y de riesgo. Los objetivos de este framework son identificar las necesidades organizacionales, relevar información y presentarla a quién la necesita, considerando la calidad y la certeza de

la información. En la Figura 1 se esquematiza la estructura general del framework.

La organización es la encargada de definir y establecer cuáles son los requerimientos de información, los cuales son provistos al sistema (Figura 1, Flujo A). Éstos son utilizados para obtener las claves de búsqueda y también permiten definir la calidad de la información que es obtenida de Internet. El Flujo B de la Figura 1 muestra que, una vez definidos dichos requerimientos, los mismos son utilizados por el sistema para buscar información en base a los criterios establecidos, aplicando técnicas de MT y MW. Además, estos requerimientos también alimentan al SSD para la evaluación de la información obtenida y la generación de información para la toma de decisiones. Cuando el sistema de MD se conecta a Internet accede a información de distinto contenido, como pueden ser texto e imágenes. Esta información es almacenada en una estructura lógica de archivo e ingresa al sistema como puede observarse en el Flujo C de la figura. Posteriormente, esta información es sometida a las técnicas de minería y recibe un valor de utilidad frente a los requisitos antes mencionados. Esta utilidad sirve de criterio para determinar qué información puede o no serle útil a la organización y se realiza una etapa de filtrado. Los archivos filtrados son descartados y aquellos que tienen información relevante se almacenan (Figura 1 Flujo D) para que el SSD los recupere luego (Figura 1, Flujo E). El SSD analiza el contenido de toda la información obtenida en base a los requerimientos para encontrar relaciones, patrones y tendencias útiles para la toma de decisiones. Finalmente, esta información se retorna a la organización que la solicitó (Figura 1, Flujo F).

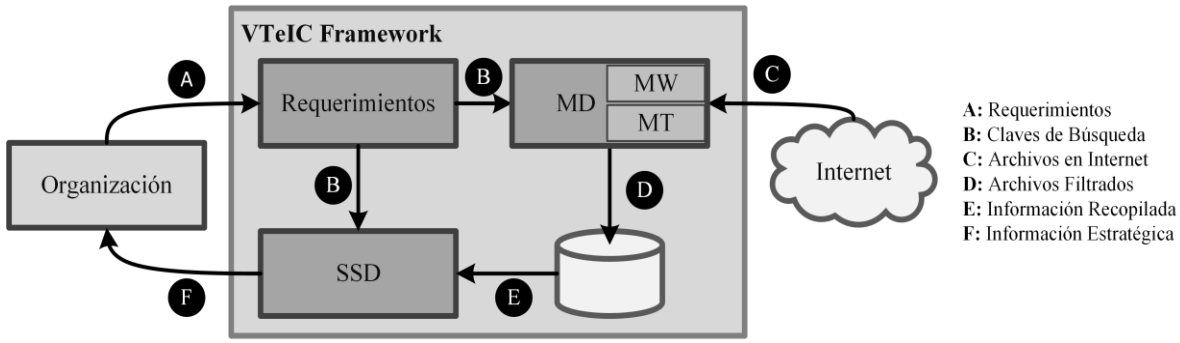


Figura 1. Framework para VT e IC.

4.2. Modelo de Capas

Dado el esquema de trabajo explicado anteriormente, se propone un sistema cuya arquitectura en la que se pueden identificar los siguientes sub-sistemas: el Sistema de Recopilación de Requerimientos (SRR), el Sistema de Minería de Datos (SMD) y el Sistema de Soporte de Decisiones (SSD).

La característica fundamental del modelo es que estos sub-sistemas no dependen el uno del otro en forma directa sino que comparten algunas estructuras de datos. De esta manera, cada uno de ellos funciona como un sistema autónomo que procesa datos y genera tanta información como pueda a partir de los mencionados datos (entrada al sub-sistema). Este esquema de funcionamiento permite obtener información continuamente, en el contexto de cada sub-sistema.

Las estructuras de datos con las que trabaja esta arquitectura son: Requerimientos, Resultados de Búsqueda e Información Estratégica. El SRR utiliza mecanismos para encontrar, a través de un diálogo con el usuario, estas necesidades. Estos mecanismos son, por ejemplo, un conjunto de preguntas cerradas y predefinidas estructuradas jerárquicamente. Su objetivo es asistir al usuario a encontrar aquellas palabras que sirvan como claves de búsqueda para el SMD. Estas claves de búsqueda también tienen que ser validadas por el usuario antes de ser almacenadas y utilizadas por los demás componentes del sistema. Una vez definidas las claves de búsqueda, el SMD accede a Internet para recopilar información contenida en distintos formatos, unificarla y valorar su utilidad en relación a los requerimientos. Posteriormente, el SSD toma la información clasificada y obtiene Información Estratégica, como por ejemplo, tendencias, estimaciones y proyecciones. En la Figura 2 se observa el modelo de capas y las relaciones entre sus componentes y los resultados que se obtienen de cada uno.

La primera capa, Presentación, es la interfaz con el usuario y es donde se implementan las funciones de recopilación de información y la de exposición de resultados. La capa siguiente, denominada Procesamiento, contiene el núcleo de la aplicación y es donde se encuentran los sub-sistemas nombrados. La capa de Acceso a Datos provee repositorios comunes de información para que el SRR, el SMD y el SSD puedan tomar y almacenar la in-

formación. La capa de Datos brinda el soporte necesario para almacenar físicamente la información para que pueda ser posteriormente compartida por cada uno de los sub-sistemas. Además, se puede observar que es el SMD el componente que establece el enlace con Internet para recopilar información.

También en la Figura 2 se enumera, a modo de descripción general, el flujo de sucesos que se llevan a cabo.

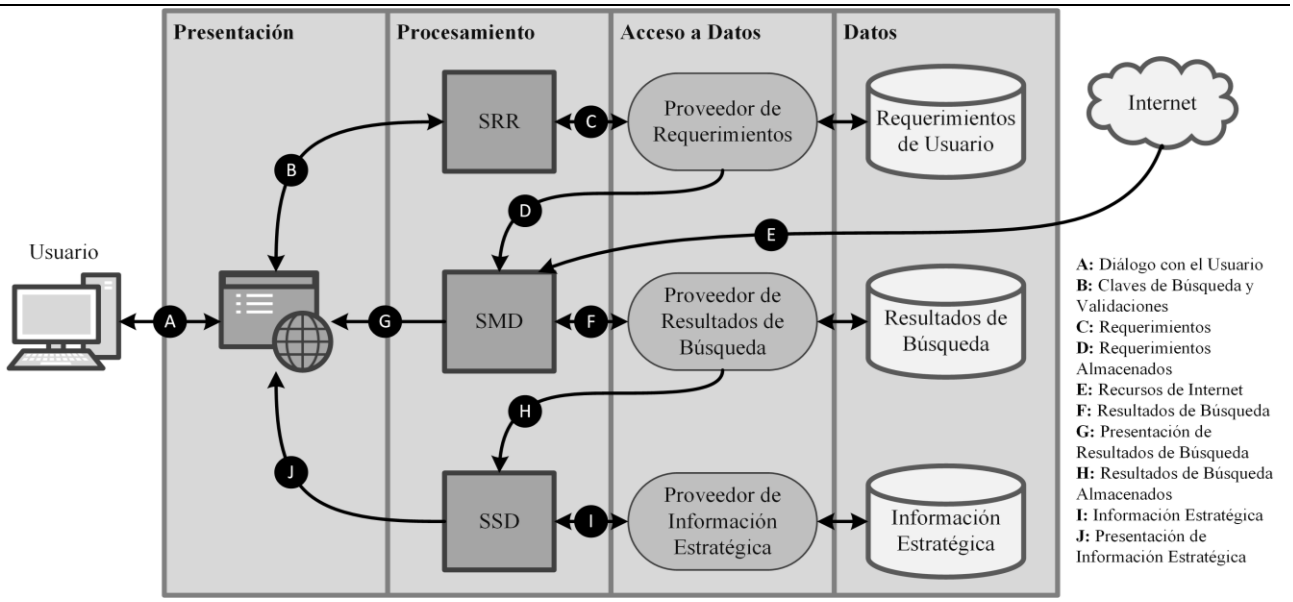


Figura 2. Proceso de obtención de Requerimientos.

El sistema genera, en primer lugar, un diálogo en el cual el SRR debe identificar concretamente qué necesita el usuario (Figura 2, Flujos A y B). Para ello se utilizan distintos mecanismos de consulta que guían al usuario a caracterizar sus requerimientos y poder determinar un conjunto de claves de búsqueda. El sistema actualiza los requerimientos del usuario si estos fueron anteriormente relevados (Figura 2, Flujo C).

El SMD toma de los requerimientos almacenados las claves de búsqueda (Figura 2, Flujo D) y accede a Internet para obtener distintos tipo de recursos (Figura 2, Flujo E). Durante esta exploración se generan estructuras homogéneas de documentos junto a medidas de la utilidad de cada recurso, agrupadas como un resultado de búsqueda, actualizando la información ya recopilada (Figura 2, Flujo F). En este punto, el usuario ya recibe la primera salida del sistema, que son los resultados obtenidos en el proceso de búsqueda (Figura 2, Flujo G). Estos resultados contienen la información recopilada de Internet y son ordenados en base al nivel de utilidad que recibieron en el proceso, lo que le permite al usuario ver la información en un estado "crudo", dado a que es propiamente el recurso de Internet. Este resultado tiene una medida de utilidad propia a la información que contiene, sin considerar el resto de los resultados en el contexto de la búsqueda.

Posteriormente, los resultados de búsqueda son utilizados por el SSD para generar información estratégica (Figura 2, Flujo H) mediante un conjunto de técnicas de regresión y tendencias, entre otras. Recién en este momento los resultados de búsqueda son relacionados entre sí y recibirán un grado mayor o menor de utilidad, dependiendo de las relaciones que se encuentren entre ellos. Esta información también se almacena para ser utilizada en otras instancias del procesamiento del SSD (Figura 2, Flujo I). Finalmente se la presenta al usuario (Figura 2, Flujo J).

4.3. Sistema de Recopilación de Requerimientos

Por lo general, existe una diferencia entre lo que los usuarios necesitan y lo que expresan como necesidad. Reducir ese gap es el objetivo del SRR, es decir, transformar la información que brinda el usuario en claves de búsqueda útiles. De esta manera, se logra enfocar el proceso de análisis de información sobre los temas que el usuario necesita.

Normalmente en un proceso de estas características un grupo de especialistas entrevista al usuario y realizan varias actividades de adquisición de información. Para el modelo propuesto se propone un mecanismo de preguntas para guiar al usuario en la determinación de sus requerimientos reales. Este mecanismo comienza con la determinación del tema principal que le interesa al usuario y continua con un proceso de refinamiento mediante preguntas estándares de respuestas cerradas. Además se

capturan las características principales del objeto de búsqueda mediante comentarios o particularidades que pueda expresar el usuario sobre lo que le interesa.

Luego esta información obtenida se concatena en expresiones lógicas que serán utilizadas como claves de búsqueda para el inicio del proceso de Minería. En la Figura 3 puede observarse un ejemplo del proceso de obtención de requerimientos. Básicamente se muestra el proceso simplificado de obtención de las características básicas, la finalidad (adquisición o venta) y finalmente si hay interés por la patente del elemento buscado.

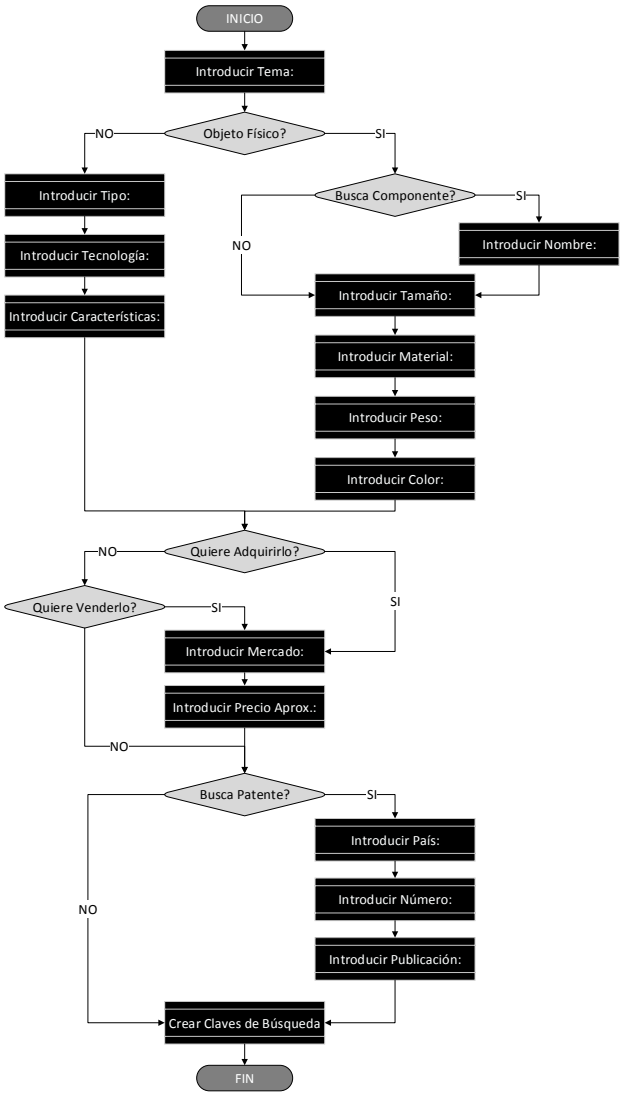


Figura 3. Proceso de obtención de Requerimientos.

Si bien este proceso puede ser refinado con mayores detalles, la idea es acotar el ámbito de alcance del inicio de la búsqueda de información. Luego, con la utilización de las técnicas de MD (y la valoración del usuario de la información adquirida mediante dichas técnicas de MD) se actualizarán las claves de búsqueda.

4.4. Sistema de Minería de Datos

El SMD (Figura 4) es el encargado de llevar a cabo el proceso de búsqueda, en el cual releva información accediendo a Internet y le asigna una medida de utilidad. Esta medida de utilidad es utilizada para filtrar los documentos en base a su contenido, y se obtiene analizando la correspondencia que existe con las claves de búsqueda. El acceso a un recurso de información en Internet se realiza mediante un Identificador de Recursos Uniforme o IRU (del inglés, Uniform Resource Indentifier o URI). Éstos son los enlaces a distintos recursos alojados en servidores en Internet, que se almacenan como archivos de distintos formatos (pdf, txt, rft, png, por ejemplo). Cada archivo es el soporte lógico de un documento de información, contenga texto, imágenes, videos u otro tipo de material. Los documentos son las estructuras que el SMD analiza y asigna un grado de utilidad, o nivel de confianza, de acuerdo al conjunto de claves de búsqueda y a las características del documento.

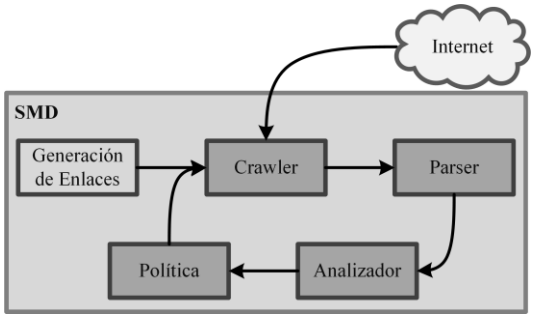


Figura 4. Interacción de los Componentes del Proceso de obtención de Resultados de Búsqueda.

La información contenida en un documento, además, puede contener enlaces a otros recursos en Internet. Por lo que el proceso que recopila y analiza esta información es cíclico. Las etapas de dicho ciclo van desde la recopilación de los recursos a partir de un conjunto de enlaces, la homogenización en una estructura de documentos, la evaluación del nivel de confianza y el valor informacional, aplicando distintas técnicas de MT y MW, y el análisis de los distintos enlaces contenidos dentro de él. El proceso para lograr esto cuenta de cinco etapas desempeñadas por los componentes detallados a continuación.

Generación de Enlaces

Para iniciar el proceso de búsqueda es necesaria una etapa previa de generación de enlaces. Esta etapa alimenta al proceso nombrado por primera vez, proveyendo enlaces que son generados en base a las reglas de distintos buscadores web (Google, Yahoo! y Bing) y a las claves de búsquedas relevadas por el SRR. Se obtienen con

ello los IRU de partida. Numerosos sitios web exponen públicamente su interfaz de programación de aplicaciones o IPA (o en inglés, Application Programming Interface o API), a través de la cual ofrecen distintos tipos de servicios. Los buscadores nombrados permiten recuperar los resultados de búsqueda que se generan al enviar un conjunto de claves. Para ello, se conforma una IRU que se adapta a la IPA expuesta por el buscador, contiene las claves de búsqueda que el SRR almacenó y, además, agrega un conjunto de parámetros adicionales como, por ejemplo, la cantidad de resultados a recuperar. Cuando el servidor remoto responde al IRU que se consulta, devuelve el conjunto de resultados del buscador junto a los IRUs a esos recursos. Estos IRUs reciben un valor de confianza dependiendo al número de apariciones y a la posición dentro de la lista de resultados de todos los buscadores. En este momento se tienen los enlaces para que el Crawler empiece a consultarlos.



Figura 5. Definición de la Interfaz del Componente de Generación de Enlaces

Crawler

Este componente se encarga de obtener los archivos a partir de los enlaces. Es una etapa puramente de recopilación y que accede a Internet a partir de un IRU. Para ello, selecciona el IRU, no explorado aún, que tiene mayor nivel de confianza y establece una conexión. El Crawler debe trabajar con características particulares de servidores y establecimiento de conexiones con ellos a modo de obtener el recurso remoto.

El Crawler, también, maneja estructuras de archivos, que son el soporte común que genera este componente y que se envía a la etapa siguiente. Cuando accede al recurso, éste es descargado y almacenado como un archivo, dado a que, generalmente, los recursos en Internet son archivos y no requieren de más procesamiento. En otros casos, los recursos pueden ser recibidos como una cadena de texto, la cual debe ser almacenada en un archivo. Este archivo tiene la información cruda como fue recibida del servidor y el enlace que dio origen a ella. Posteriormente es enviado el archivo al Parser.

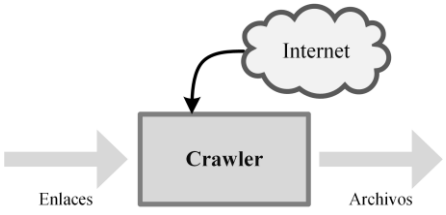


Figura 6. Definición de la Interfaz del Crawler

Parser

Es el componente encargado de la homogeneización de la información que se recibió de la etapa anterior y que se encuentra dentro del archivo. Se encarga de realizar una etapa de transformación. Durante su operación, selecciona entre los archivos descargados por el Crawler, aquél que esté relacionado al enlace con mayor nivel de confianza. Luego identifica características del archivo y del enlace que le permiten seleccionar una estrategia de cómo tratarlo para poder generar un documento. Este documento es la estructura de información uniforme que se analiza, y brinda la flexibilidad de almacenar información de manera plana como jerárquicamente, a modo que la siguiente etapa no tenga que lidiar con detalles relativos a la estructura y características del archivo. Asimismo, el Parser identifica en un archivo la existencia de enlaces que parten hacia otros recursos, como ser otros sitios web o archivos, denominados "enlaces siguientes". Estos nuevos enlaces son necesarios para realimentar el proceso de búsqueda, que los expande si se determina que es necesario continuar (en base al nivel de confianza del enlace). A la salida de esta etapa se tiene un documento, que contiene una referencia al archivo que lo generó, el conjunto de enlaces siguientes y la información del contenido del archivo homogenizada.



Figura 7. Definición de la Interfaz del Parser

Analizador

Este componente se encarga de procesar toda la información del documento. En primer lugar, determina un nivel de confianza del documento, analizando el contenido en contraste con las claves de búsqueda que se definieron. Concretamente, aplica las técnicas de MT y MW, nombrados en la Sección 3.1 y asigna una medida de utilidad basada a la información que contiene. Luego evalúa los enlaces siguientes del documento, asignando un nivel de confianza a cada uno de ellos en base a su IRU. Los enlaces tienen una medida de utilidad dado a que son utilizados para continuar con el proceso de búsqueda, realimentando al proceso, y se determina analizando el IRU.

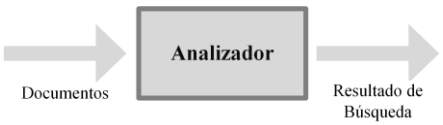


Figura 8. Definición de la Interfaz del Analizador

Esta información del documento y los enlaces es denominada resultado de búsqueda, ya que contiene la información relativa al análisis de un recurso obtenido en el

proceso de búsqueda. Estos resultados son los que permiten continuar el proceso de MD, presentar la información encontrada al usuario del sistema y alimentar al SSD.

Política

Este componente se encarga de realizar la retroalimentación al proceso cíclico. Para ello determina cuáles serán los enlaces a ser explorados. Toma el conjunto de documentos y los enlaces, ambos junto a su respectivo nivel de confianza, y mediante estrategias de ponderación y evaluación decide con cuáles de ellos continuar, comunicándolos al Crawler. La ponderación debe considerar el nivel de confianza del enlace y el nivel de confianza del archivo en el cual se encontraba el enlace. Debe reflejar la conveniencia de continuar expandiéndolo, y es fundamental para que la retroalimentación del proceso de búsqueda sea eficiente, de modo que no se pierdan recursos ni tiempo analizando información irrelevante.



Figura 9. Definición de la Interfaz de la Política

4.5. Sistema de Soporte de Decisiones

Este subsistema tiene como objeto brindar información adicional sobre los recursos obtenidos por el proceso de MD. Básicamente se trata de un módulo Análisis de Decisión Multi Criterio (ADMC) que implementa técnicas de categorización de los mencionados recursos.

El SMD devuelve documentos (obtenidos a través de los procesos de MW y MT) que tienen asociado un nivel de confianza relacionado con la calidad del recurso. Esto permite construir un ranking de los documentos para que el usuario pueda analizarlos. Si bien, este nivel de confianza es derivado de factores como la calidad del recurso, el origen, la cantidad y la calidad de referencias externas, etc., es necesario poder relacionar los documentos entre sí. Para ello se definen criterios de evaluación de los documentos que, luego de ser analizados, permiten su clasificación.

El proceso de ADMC es altamente deseable en este tipo de sistemas ya que permite caracterizar los recursos en base a criterios previamente definidos. En este sentido, el modelo propuesto utiliza métodos que toman como base la Teoría de Utilidad con Multi-atributos, asignan un valor de utilidad a cada documento. Ese valor es un número real que representa el grado preferencia del documento en cuestión. La utilidad se obtiene como resultado de la sumatoria de las utilidades marginales que cada criterio asigna al documento. Este tipo de mecanismo generalmente se utiliza en la vida real cuando para elegir una alternativa se estima el peso relativo de cada aspecto uniéndolos en un solo grado de preferencia.

Finalmente, luego del proceso de ADMC, se le brinda al usuario información de calidad y adecuadamente clasificada, relacionada con su área de interés.

5. Discusión y Trabajos Futuros

El modelo presentado en este trabajo tiene como objeto el análisis y filtrado de información para realizar Vigilancia Tecnológica e Inteligencia Competitiva. La arquitectura propuesta posee tres componentes básicos:

- *Sistema de Recopilación de Requerimientos*, que se encarga de realizar el filtrado inicial de las necesidades del usuario automáticamente. La idea principal de este refinamiento es acotar el ámbito inicial de búsqueda.
- *Sistema de Minería de Datos*, que se encarga del análisis de la información obtenida, asignándole niveles de confianza a los recursos recopilados.
- *Sistema de Soporte de Decisiones*, que se encarga de la clasificación y presentación de la información al usuario facilitando su interpretación.

En esta propuesta, el usuario define sus claves de búsqueda orientado por el sistema que se encarga de utilizarlas para la obtención de información. Los recursos encontrados son valorados y se extrae de ellos la información relacionada con sus contenidos que se utiliza para realimentar el proceso de búsqueda. Este procedimiento es constante y el sistema se encarga de presentar la información clasificada y ordenada (que ha sido recopilada hasta el momento) cada vez que el usuario lo solicita.

El modelo propuesto tiene como objetivo dar soporte a las decisiones estratégicas de una organización, brindando información adecuada en tiempo y forma sobre avances tecnológicos y actividades del mercado.

Actualmente, se encuentran en desarrollo los Sistemas de Recopilación de Requerimientos y de Soporte de Decisiones y en programación el Sistema de Minería de Datos (específicamente, el generador de enlaces, el crawler y el parser). A futuro se prevé avanzar con los analizadores de texto utilizando técnicas bayesianas y algoritmos de clasificación.

Agradecimientos

Este trabajo corresponde al proyecto "**Modelos de Análisis de Información para Vigilancia Tecnológica e Inteligencia Competitiva (VTeIC)**" EIUTNRE0002106 (Universidad Tecnológica Nacional, Facultad Regional Resistencia).

Referencias

[1] Firat, A., Woon, W., and Madnick, S., *Technological Forecasting – A Review*, Working Paper CISL# 2008-15, Composite Information Systems Laboratory (CISL), Massachusetts Institute of Technology. Cambridge, MA 02142, 2008.

[2] Vedder, R. G., Vanecek, M. T., Guynes, C., and Cappel, J., *CEO and CIO Perspectives on Competitive Intelligence*, Communications of the ACM, Vol 42. No 8, 1999.

[3] Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J. Sharda, R., and Carlsson, C., "Past, Present, and Future of Decision Support Technology", *Decision Support Systems*, 33, 2, 2002, pp. 111-126.

[4] Holsapple, C. W., "Decision Support Systems", *Encyclopedia of Information Systems*, 1, 2003, pp. 551-565.

[5] Power, D. J., and Sharda, R., "Decision support systems", *Springer Handbook of Automation*, 2009, pp. 1539-1548.

[6] Weiss, S. H., and Indurkhya, N., *Predictive Data Mining: A Practical Guide*, San Francisco, CA: Morgan Kaufmann Publishers, 1998.

[7] Coates, V., Faroque M., Klavins R., Lapid K., Linstone H. A., Pistorius C., and Porter A. L., "On the future of technological forecasting", *Technol. Forecast. Soc. Change*, 67, 1, 2001, pp. 1-17.

[8] Gordon, T. J., and Glenn, J. C., (*Eds.*), *Futures research methodology*, Version 2.0 Millennium Project of the American Council for the United Nations University, 2003.

[9] Madnick, S., and Woon, W. L., *Technology Forecasting Using Data Mining and Semantics*, MIT/MIST Collaborative Research, 2009.

[10] Parente, F., Anderson, J., Myers, P., O'Brien, T., "An examination of factors contributing to Delphi accuracy", *J. Forecast*, 3, 1984, pp. 173-182.

[11] Levary, R., and Han, D., "Choosing a technological forecasting method", *Industrial Management*, 37, 1, 1995, p. 14.

[12] de Miranda Santo, M., Massari Coelhno, G., dos Santos D. M., and Fellows Filho, L., "Text mining as a valuable tool in foresight exercises: a study on nanotechnology", *Technol. Forecast. Soc. Change*, 73, 2006, pp. 1013-1027.

[13] Gordon, T. J., "A simple agent model of an epidemic" *Technological Forecasting & Social Change*, vol. 70, 2003, pp. 397-417.

[14] Hiltbrand, T., *Learning Competitive Intelligence From a Bunch of Screwballs*, Business Intelligence Journal, Vol: 15, No: 4, 2010.

[15] Popa, I., and Anica, Cucui G., "A Framework for Enhancing Competitive Intelligence Capabilities using Decision Support System based on Web Mining Techniques", *Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. IV, No. 4*, 2009, pp. 326-334.

[16] Shu-Hsien, L., Pei-Hui, C., and Pei-Yuan, H., "Data mining techniques and applications – A decade review from 2000 to 2011", *Expert Systems with Applications 39*, 2012, pp. 11303-11311.

[17] Ghosh, S., Roy, S., and Bandyopadhyay, S., *A tutorial review on Text Mining Algorithms*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, 2012.

[18] Etzioni, O., "The World Wide Web: Quagmire or gold mine", *Communications of the ACM*, 39, 11, 1996, pp. 65-68.

[19] Kosala, R., and Blokeel, H., "Web mining research: A survey", *ACM SIGKDD Explorations*, 2, 1, 2000, pp. 1-15.

[20] Chakrabarti, S., Van Der Berg, M., and Dom, B., "Focused crawling: A new approach to topic-specific Web resource discovery", *In Proceedings of the 8th International Conference on World Wide Web WWW8*, 1999, pp. 545-562.

[21] Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Paatero, V., and Saarela, A., "Self-organization of a massive document collection", *IEEE Transactions on Neural Networks*, 11, 3, 2000, pp. 574-585.

[22] Chang, C. H., Hsu, C. N., and Lui, S. C., "Automatic information extraction from semi-structured Web pages by pattern discovery", *Decision Support Systems*, 35, 2003, pp. 129-147.

[23] Arnott, D., and Pervan, G., "Eight Key Issues for the Decision Support Systems Discipline", *Decision Support Systems*, 44, 3, 2008, pp. 657-672.

[24] Gao, S., Wang, H., Xu, D., and Wang, Y., "An Intelligent Agent-assisted Decision Support System for Family Financial Planning", *Decision Support Systems*, 44, 1, 2007, pp. 60-78.

[25] Gray, P., "Group Decision Support Systems", *Decision Support Systems (North-Holland)*, 1987, pp. 233-242.

[26] Nour, M. A., and Yen, D. C. C., "Group Decision Support Systems: Towards a Conceptual Foundation", *Information & Management*, 23, 2, 1992, pp. 55-64.

[27] Pervan, G. P., "A Review of Research in Group Support Systems: Leaders, Approaches and Directions", *Decision Support Systems*, 23, 2, 1998, pp. 149-159.

[28] Tawfik Jelassi, M., and Foroughi, A., "Negotiation Support Systems: An Overview of Design Issues and Existing Software", *Decision Support Systems*, 5, 2, 1989, pp. 167-181.

[29] Gottinger, H. W., and Weimann, P., "Intelligent Decision Support Systems", *Decision Support Systems*, 8, 4, 1992, pp. 317-332.

[30] Dutta, A., "Integrating AI and Optimization for Decision Support: A Survey", *Decision Support Systems*, 18, 3, 1996, pp. 217-226.

[31] Yen, J., "Special Issue of Decision Supports Systems - Intelligent Agents and Digital Community", *Decision Support Systems*, 28, 3, 2000, pp. 217-218.

[32] Robbins, R.W., and Wallace, W. A., "Decision Support for Ethical Problem Solving: A Multi-agent Approach", *Decision Support Systems*, 43, 4, 2007, pp. 1571-1587.

[33] Sadagopan, S., "Enterprise Resource Planning", *Encyclopedia of Information Systems*, 2, 2003, pp. 169-184.

[34] Holsapple, C. W., and Sena, M. P., "ERP Plans and Decision-support Benefits", *Decision Support Systems*, 38, 4, 2005, pp. 575-590.

Una Estrategia de Aprendizaje Autónomo para Sistemas Expertos, con Aplicación Concreta al Proyecto – Generador Automático de Modelos de Datos Normalizados en Bases de Datos Relacionales

Luis Esteban Damiano
UTN-FRC
luis.damiano@gmail.com

María Alejandra Paz Menvielle
UTN-FRC
pazmalejandra@gmail.com

Roberto Miguel Muñoz
UTN-FRC
robertmunioz@gmail.com

Juan Carlos Cuevas
UTN-FRC
Juancarloscue@gmail.com

María Soledad Romero
UTN-FRC
romeroma.soledad@gmail.com

Carlos Bartó
UNC
cbarto@gmail.com

Abstract

Presentamos las ideas surgidas en relación a la adquisición de conocimiento por parte de un sistema experto (SE), dentro del avance del desarrollo de mismo. Este SE está orientado a la validación de estructuras de datos, dentro de un modelo de datos solución, para Base de Datos Relacionales. El proyecto surge en el seno de un grupo de docentes de la cátedra Gestión de Datos de la carrera Ingeniería en Sistema de Información, como producto para la mejora de los recursos de práctica de los estudiantes. El desarrollo no se encuentra culminado, pero si ya están establecidos los lineamientos y características principales. Se trabaja en el diseño de un sistema visual, de características intuitivas, que lleva embebido un SE desarrollado en el lenguaje Clips. Ambos sistemas se desarrollan en paralelo. Desarrollamos conocimientos en: Ingeniería del Conocimiento, programación de Sistemas Expertos, Lenguaje de programación Clips, Metodología CommondKads, Ontologías.

1. Introducción

1.1. Contexto Institucional

En la Facultad Regional Córdoba de la Universidad Tecnológica Nacional (UTN-FRC), se encuentra el Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información (GIDTSI), en este ámbito se ubica el proyecto de investigación que da origen a este artículo. El mismo se encuentra homologado por la Universidad con el código UTN1702, titulado *Generador Automático de Modelos de Datos, Normalizados en Bases de Datos Relacionales*.

1.2. Antecedente

Algunos investigadores de este proyecto, ya han participado en otras investigaciones, igualmente homologadas por UTN, también referidas a la temática de bases de datos, pero con otras orientaciones, estas son: PROMETEO - Desarrollo de un método y una herramienta para el aprovechamiento de Metadatos de Base de Datos Relacionales (2010), Análisis y aplicación de metodologías para la generación de consultas complejas utilizando esquemas OLAP (2010), y administrador de Base de Datos Relacional TecnoDB (2007).

El Proyecto de Investigación y de Desarrollo (PID), está constituido por un equipo de docentes de diversas especialidades: Programación, Inteligencia Artificial, Base de Datos, Computación y Sistemas de Información, también participan estudiantes avanzados de la carrera Ingeniería en Sistemas de Información.

Este proyecto ya ha sido presentado en otros encuentros Científicos y de Educación en las Ingenierías, siendo estos: WICC 2013 [16], ConaIISI 2013 [1,17], JEIN 2014 [9], donde en cada uno de ellos se ha tratado diversos temas inherentes a este proyecto.

1.3. Origen de Proyecto

En el plan de estudios de la Cátedra Gestión de Datos de la carrera Ingeniería en Sistemas de Información, de la UTN-FRC, uno de los temas a tratar fuertemente en el dictado de clases teórico y práctico, es la Normalización de Modelos de Datos para BD relacionales.

En relación a este tema y con el interés de algunos profesores, se empezó a hablar sobre la posibilidad de

tener un herramienta para mejorar las prácticas de los estudiantes, que permitiera la mejora de una serie de aspectos, tales como: a) realizar prácticas áulicas y extra áulicas asistidas por un software que los oriente en el diseño de un modelo de datos que respete las formas normales. b) que este software sea capaz de emitir opinión y consejo (sugerencias) sobre errores estructurales, c) mejorar la visualización de un modelo de datos resultado, al permitir cambiar la presentación desde una hoja de papel escrita a mano, a presentación en pantalla realizada por un software de fácil utilización que presente las estructuras del modelo con mayor claridad, d) permitir la fácil portabilidad de los resultado de una computadora a otra, e) que el software pueda recolectar información sobre los problemas de construcción de modelos (por parte de los estudiantes), para poder establecer cuáles son las dificultades comunes, f) que la herramienta (soft) se vuelva un instrumento de autoevaluación para los estudiantes, permitiéndole mejorar su desempeño y aprendizaje.

En torno de estos conceptos, es que se formó el equipo docente que presentó el proyecto de investigación y desarrollo, en principio con un conjunto de ideas que fue mutando con el paso del tiempo. Ya desde hace un tiempo se recogen distintas propuestas que ofrecen una visión innovadora del trabajo en el aula en el ámbito del área de la Tecnología; se identifican problemas y se plantean alternativas y soluciones. Actualmente se está dando un fenómeno de co-evolución entre lo tecnológico y lo social. Esta nueva evolución en conjunto establece las pautas hacia nuevas formas de conceptualizar y de llevar a la práctica el modelo educativo.

1.4. Las Ideas y nuestros Problemas

La idea original del proyecto era la construcción de una herramienta que permitiese la validación estructuras de modelos de datos relacionales, y la construcción de estructuras o modelos de datos a partir de un paquete de datos ingresado. Ambos productos concebidos o pensados en un único proyecto de solución. Esta idea fue cambiando con el paso del tiempo y el conocimiento adquirido, pues el dominio del problema era demasiado grande para ser tratado como una único proyecto de solución, considerando las premisas del diseño de SE y Clips (herramienta para la implementación de estos sistemas)[2,3,13-15,19] .

Esta idea original cambió en su construcción conceptual a dos proyectos en paralelo, uno dedicado a la validación de modelos, y el otro a la construcción de modelos, de esta forma el dominio de cada uno, se restringió a una sola tarea, y pasamos a tener claramente acotado cada dominio, premisa fundamental para el desarrollo de un Sistema Basado en Conocimiento (SBC)[3].

Luego de rediseñar el proyecto de desarrollo del SE, empezamos a ver las dificultades reales de la interfaz con el usuario, pues Clips [15] está pensado (de acuerdo a nuestra percepción) como una herramienta *banco de pruebas* para el núcleo de un SE, más que una herramienta para desarrollos finales. Claramente no podríamos a través de Clips lograr una interfaz con el usuario de tipo amigable, de diseño visual y de ágil utilización. Esto nos llevó a realizar investigación sobre posibles derivaciones de la herramienta, experiencias de otros investigadores o desarrolladores, que nos permitieran encontrar un camino alternativo. Es decir, que sin dejar de lado el producto que ya habíamos empezado a estudiar fuertemente, nos allane el camino para lograr un ensamble entre Clips con otras herramientas, con capacidad de construir un entorno visual de alta productividad. Recorrimos diversos camino investigativos, pero el que nos dio verdadera satisfacción fue el sitio oficial (http://clipsrules.sourceforge.net) del producto Clips. Nuestro especial interés era la utilización de VB.Net (Visual basic punto net, herramienta integrada en el visual estudio de Microsoft.)[7] como herramienta de desarrollo visual, pues ya estábamos en contacto en Vb.Net y C# [6] de la herramienta Visual Studio de Microsoft, y Clips como herramienta de razonamiento (motor de inferencia), estudiábamos este producto previo al inicio del grupo de investigación. Si bien en el sitio oficial hay muestra de la herramienta compilada desde .Net, ésta, está desarrollada en C++ [23] , ahí también teníamos un limitante fuerte, pues en el equipo tenemos conocimiento de VB y C#, pero no C++, y no encontrábamos viable tener que estudiar C++ también, para el desarrollo del producto nuestro, pues ya, tendríamos que ampliar el conocimiento con Clips, y el proyecto tiene un límite temporal. También hallamos productos para trabajar con java, JESS, y un producto desarrollado para entornos Windows llamado wxClipis, para aclarar por qué un producto y no otro, sencillamente nuestro interés primero es Clips, luego compatibilizar este producto con .Net, sin desmerecer las bondades y virtudes de las demás herramientas encontradas, especialmente considerando que el equipo docente del proyecto, está fuertemente ligado a materias que utilizan como recurso de enseñanza productos de Microsoft. Luego de seguir estudiando la información del sitio de Clips y sus derivaciones recomendadas, llegamos a un equipo de investigaciones que ofrecía una DLL compilada para montar en un proyecto .Net, este sitio se presenta como CLIPS.NET (http://clips.codeplex.com/ Project Hosting for Open Source Software - Clips.Net Disponible: Agosto 2014.), las pruebas que realizamos fueron totalmente exitosas, los prototipos que teníamos ya corriendo en Clips, los montamos en una aplicativo VB.Net junto a la dll y corrieron satisfactoriamente arrojando los mismos resultados. Dentro de este sitio hay

diversas ejemplificaciones de utilización, todos los productos vistos en distintos sitios, respetan en un porcentaje alto, los procedimientos definidos en el mismo Clips para ser embebido.

También se trabajó investigando sobre el diseño de los formularios que permitieran captar el modelo de bases de datos relacional pensado por el usuario del producto. Se investigó si los productos de bases de datos relacionales basados en tecnologías FLOSS (siglas de free/libre and open source software, en inglés http://www.flossproject.org/ Free/Libre and Open Source Software: Survey and Study) como MySql (MySql: Sistema de gestión de base de datos relacional de código abierto. http://www.mysql.com/ - Disponible Agosto 2014) contaban con alguna manera de exportar los modelos relacionales diseñados a través de la interfaz gráfica. También se evaluó alternativas con componentes de productos privados. Dado que no se hallaron posibilidades se decidió aprovechar el análisis introspectivo realizado en distintos sistemas de gestión de bases de datos a fin de conformar el requerimiento con el diseño de interfaz necesario para captar el esquema pensado por el usuario final del producto. Este documento de requerimiento sirvió de base para definir los lineamientos generales del desarrollo del producto con herramientas de tipo visual.

Llegar a este punto tuvo un desarrollo en el tiempo que consumió el 50% del tiempo previsto para el PID (Proyecto de Investigación y Desarrollo), ahora que teníamos todos los implementos para el desarrollo que queríamos alcanzar, el tiempo no era suficiente, por ende solicitamos una necesaria ampliación de tiempo para el desarrollos del PID, para no ver frustrado nuestro trabajo. El equipo cuenta con integrantes con diversas especializaciones, pero ninguno había explorado este ámbito, el de desarrollar un SE con estas características.

1.5. Sistema Validador

Un SE que permita capturar un modelo de datos en formato visual, donde el usuario podrá describir relaciones (entidad, tabla con mayor nivel de restricciones) con sus atributos, dependencias funcionales, clave primaria y foráneas, como si lo estuviera esquematizando en un hoja de papel, luego de terminado el modelo, podrá someterlo a evaluación, tarea que realizará la programación Clips embebida en el aplicativo. También le permitirá el almacenamiento del modelo con un nombre específico en una base de datos. Esta persistencia del modelo, le permitirá discontinuar la tarea en cualquier momento, pudiendo retomarla después. También el sistema tendrá un mecanismo de almacenamiento progresivo, un histórico de la evolución de un mismo modelo, para que el usuario pueda ver cómo fue cambiando su conceptualización de la solución desarrollada de un problema determinado. Se podrá almacenar n soluciones. Está prevista la portabilidad de una solución determinada a través de XML hacia otro computador con el mismo Soft. Dentro del ámbito de la Universidad en los laboratorios de Informática de la carrera Ingeniería en Sistemas de Información, donde los estudiantes harán sus prácticas, el sistema tendrá la propiedad de almacenar las dificultades que detectó en las validaciones, esta información luego podrá ser extraída y entregada al Director de Cátedra, para que puedan evaluarse diferentes estrategias para mejorar el aprendizaje de los estudiantes.

Como se comentó anteriormente, originalmente estaba previsto que el SE desarrollara también la solución de un modelo de datos, a partir de un conjunto de datos (dominios) suministrados. Se evaluó que si bien la idea es interesante, los tiempos del proyecto eran insuficientes como para llevar adelante ambos prototipos. También se observó que el nivel de aplicación y productividad académica, estarían fuertemente satisfechos por la variante de validación y no por la de generación de un modelo. Quedando sí la posibilidad de retomar su análisis en un proyecto posterior.

2. Elementos del Trabajo y Metodología

2.1. Normalización de Modelos de Datos

La tarea del Diseño Lógico de un modelo de datos, para bases de datos relacionales está fundamentada en el cumplimiento de las Formas Normales, éstas son un conjunto de reglas que tienen como propósito la configuración de la representación, de una solución a un problema de almacenamiento de datos, que satisface características estructurales que se deben cumplir, para que ésta pueda ser montado en un sistema de gestión de base de datos relacionales. Esta tarea se denomina habitualmente *Normalización de una base de datos*, y como se mencionó, está condicionada por el cumplimiento de las Formas Normales, que son cinco (5), numeradas en forma ascendente desde uno (1), más una especialización de la tercera forma normal, denominada: forma normal Boyce-Codd. La nomenclatura habitual es 1FN para primera forma normal, las demás respetan esta característica, sólo cambiando la numeración, salvo la forma normal Boyce-Codd cuya sigla es FNBC [10, 11, 20,21].

El informático experto en normalización tiene que realizar la tarea de comprensión de un problema real de almacenamiento de datos, logrando un correcto entendimiento de esta problemática, éste, a través de diferentes estructuras (entidades o conjunto de éstas) combinadas entre sí, realizar una representación del problema, que entendemos es la solución de

almacenamiento del mismo. Está acción de normalización en la mayoría de los casos, para los expertos, no implica un análisis de cada estructura (entidad) que propone como parte de la solución, para establecer si la misma se encuentra cumpliendo las formas normales. El experto humano construye un modelo automáticamente, que siempre respeta las formas normales. Esta forma de hacer su trabajo, sin necesidad de validar cada paso, implica un conocimiento de cómo hacer, que está profundamente internalizado y pareciera no formar parte de acciones conscientes o reflexivas. Este cómo hacer, que desde fuera (observador externo) pareciera una automatización en el pensamiento, es el principal interés del Ingeniero del Conocimiento, dentro del PID.

También es destacable que si observamos a dos expertos en normalización de modelos de datos (para base de datos relacionales), que desarrollan una solución de un mismo problema cada uno, con la misma fuente de conocimiento, claramente los dos llegan a la misma representación de solución, donde pueden diferir los nombre de los componentes, pero las estructuras deben ser las mismas. En este tipo de representación existe un conjunto de estructuras para ser utilizadas en diversas combinaciones, estas son una cantidad limitada, pero permite resolver toda clase de problemas. Existen muy pocos casos en donde hay más de una estructura (muy parecidas entre sí) para representar la solución a un mismo problema, pero siempre una de ellas más deseable que la otra [20].

2.2. Ingeniería del Conocimiento

Es la especialidad dentro de la Informática que permite el análisis y extracción del conocimiento desde los saberes y prácticas de un experto en una tarea específica. El Ingeniero del Conocimiento, es una Informático que se dedica al desarrollo de SBC, cuya tarea se puede sintetizar en: análisis, documentación, extracción e interpretación del conocimiento de un experto en el problema tratado, que luego transformará en estructuras típicas de los sistemas basados en el conocimiento [3, 13,19].

Al contrario de lo expresado en el concepto de Normalización, donde dijimos que el experto en esta tarea debe entender un problema, el Ingeniero del Conocimiento debe producir una interpretación del conocimiento extraído del experto en normalización, pues debe transformar ese conocimiento a un conjunto de patrones y reglas [2, 3] que lo represente lo más claramente posible, de hecho no hay un único camino de representación, pues se podría construir diversas representaciones que conduzcan en el razonamiento al mismo resultado.

El Ingeniero del Conocimiento debe construir la Base de Conocimiento (BC) del sistema experto, que está constituida por un conjunto de reglas que analizan patrones (hechos). Por propia definición y aprendizaje, hemos llegado a interpretar que la base de conocimiento puede estar desdoblada en dos componentes: a) un conjunto de reglas que analizan los hechos (patrones) que surgen de los datos provenientes del modelo de datos, estas reglas producen inferencias, éstas surgen del saber hacer del experto humano en el problema. b) un conjunto de hechos que conforman grupos de patrones de conocimiento, que representan el conocimiento de estructuras típicas que posee el experto, de las que se vale para realizar comparaciones y en función de ellas, y construir soluciones reutilizando estas estructuras, o tomándolas como modelos para producir una variación ajustada, dentro de la problemática que está desarrollando.

2.3. Clips - Sistemas Expertos

Clips (C Language Integrated Production System)[15] es una herramienta de diseño de Sistemas Lógicos basado en los Sistemas de Producción y también en Objetos o Marcos, que produce razonamientos a través de un motor de inferencia. Es un producto que permite desarrollar sistemas expertos, siendo Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center, originalmente los productores iníciales de la herramienta, que ahora ha quedado a cargo de sus autores.

Hacia 1984 comienzan a desarrollar este producto, y desde entonces se ha ido mejorando hasta hoy, encontrándose actualmente en la versión 6.3 beta. El sistema ha logrado integrarse con los principales producto de este momento como .Net, Java, C++ borland, en su desarrollo original se planteaba la integración con un conjunto de productos característicos de los años 80/90.

La construcción de un SE desde su origen no tiene las mismas características de diseño que la de un sistema convencional orientado a objeto, que podríamos documentar con métodos de ingeniería de software, donde suele utilizarse UML. Los sistemas expertos tienen una mecánica interna distinta que amerita una forma de diseño particular, donde la ejecución está conducida por los datos, y no por un flujo de trabajo que es desviado por estructuras de control.

El ciclo de vida de un SE está supeditado a que exista un hecho o un paquete de éstos que satisfaga una o varias reglas, entrando éstas en agenda para su ejecución. Conforme a esta ejecución de reglas, se dispara la programación del lado derecho de las mismas, donde se pueden producir o no nuevos hechos, que nuevamente pondrán en agenda un conjunto de reglas, si los patrones de análisis de las mismas son satisfechos.

Esta metodología de ejecución, que resulta de una programación típica de los sistemas basados en el conocimiento, requiere de una forma de documentación específica y particular para realizar la tarea previa al desarrollo de SE. Algunos conceptos importantes para desarrollo de un SE son: a) que el problema a solucionar no encuentre fácil o ningún tipo de solución a través la programación algorítmica procedural, b) que el dominio del problema sea restringido y totalmente delimitado; para el caso particular bajo análisis, la validación de 1FN, dentro de un modelo de datos que representa la solución a un problema determinado, es un dominio acotado y totalmente delimitado, c) si el SE tiene que validar o analizar varias restricción en paralelo que implican dominios particulares que conviven dentro del dominio de problema, podríamos decir que para cada uno de estos dominios particulares, debe haber un SE que realice su trabajo de validación. Como consecuencia a lo antes descrito, se podría realizar una SE modular que separe en distintos procesos, las particularidades del dominio del problema.

2.4. CommonKads

No es nuestra intención ahondar en los conceptos de CommondKADS, pues estos ya lo expusimos en el artículo presentado en ConIISI 2013 [17], referido a la problemática aquí expuesta. Nuestra elección de producto está relacionada a los aportes de conocimientos vertidos por Ing. Carlos Bartó, en el momento que se incorporó al grupo de investigación, pues desconocíamos hasta ese momento la metodología.

La metodología CommonKADS [19,22] es un mecanismo para generar documentación para SE, para el manejo de las siguientes tareas: a) organizativas, b) de seguimiento, c) documentación, d) control de proceso, y diseño. Esta metodología que se construyó específicamente para sistemas basados en conocimiento (SBC), similar a la metodología de diseños de sistemas tradicional denominada Ingeniería del Software (IS). Surgió dentro del ámbito de la comunidad de investigadores de diversos países de la comunidad Europea (August Schreiber [8]), alrededor de 1983.

Este método trata tres conceptos fundamentales: modelado, reutilización y gestión del riesgo. Es claro que estos tres conceptos, no son específicos de la Ingeniería del Conocimiento (IC), también se incluyen en el ámbito de la Ingeniería del Software en general.

La tecnología y métodos de CommonKADS cubren todos los espacios del desarrollo de software de un SE, a través de un conjunto de modelos de documentación tipificados interrelacionados, que permiten plasmar los rasgos más destacados del sistema y su medio.

En el desarrollo de SBC utilizando CommonKADS se debe completar un conjunto de planillas estandarizadas, donde se conceptualizan los modelos. En función de ellas CommonKADS establece estados de los modelos, que definen puntos destacados o críticos en el desarrollo.
Estos estados establecen un camino dentro de la gestión del proyecto, donde el desarrollo se produce en forma cíclica (reiterada) dirigida por los riesgos.

3. Resultados

Dentro de este proyecto, todas las instancias fueron un desafío para el equipo investigador, pues la gran mayoría de sus integrantes no proviene del mundo de la inteligencia artificial en su especialidad de sistemas expertos. Cuando se llegó a la etapa de construcción de los primeros prototipos de prueba en programación Clips, se encontró que la base de conocimiento no podía ser estática, pues siempre surgían nuevos componentes para agregar, referidos a conocimiento. Se pudo observar con claridad la diferenciación entre el conocimiento que provenía de la acumulación de información del experto, del conocimiento de cómo hacer o conocimiento de las tareas.

El conocimiento referido a la información acumulada, era susceptible de crecimiento permanente y cotidiano para el experto humano, pues con la construcción de cada nuevo modelo, existe un fuerte potencial de desarrollar una nueva estructura (entidad, tabla) que nunca antes había desarrollado, que se podría encuadrar dentro de alguna estandarización, ya aprendida. Por ejemplo el experto en su conocimiento puede tener claro la clasificación de tipos (entidad que sirven para clasificar tipos) que responde a una entidad estandarizada, cuya estructura tiene los atributos “id del tipo” y “descripción del tipo”, que puede tener sutiles variaciones en la estructura, también conocer un conjunto de todas las tipificaciones que ha realizado (ej. tipo de documento, tipo de estado civil, etc.), pero es altamente probable que deba incorporar nueva información al encontrar un tipo que nunca antes haya trabajado. Esto dio origen a la división conceptual del diseño, donde el conocimiento informativo debía ser separado del conocimiento de cómo hacer, dando lugar a una estrategia de adquisición del conocimientos, de información no registrada en la BC (datos).

3.1. Estrategia de Adquisición de Conocimiento

Se requerirá una base de conocimiento dinámica, que pueda ser cambiada, tanto en su contenido (crecimiento), como en la cantidad de componentes (atributos) que son relevantes en el entendimiento de una estructura (entidad) [12, 14,19].

Se contará con una base de conocimiento inicial, que tendrá un crecimiento sostenido (importante) en un

tiempo inicial, y luego se espera continúe creciendo a una tasa menor.

La utilización de una base de datos, como ya se mencionó, para el almacenamiento de los modelos desarrollados por los usuarios, también deberá ser soporte de la base de conocimiento de SE.

3.2. Estandarización de Estructuras - Ontología de Dominio

El Ontólogo (Bunge 1977): *"debe reconocer, analizar e interrelacionar aquellos conceptos que le permitan producir una representación unificada de la realidad"* [24]

Ontología (Gruber en 1993): *"Una ontología es una especificación explícita de una conceptualización"*, la conceptualización, que para nuestro caso es una estructura de representación del conocimiento, en sí no es la ontología, sino los objetos que esta estructura representa.

Surge la necesidad de tener un único formato de compresión del dominio inherente al conocimiento (base de conocimiento), que será tratado por tres sistemas diferentes (la base de datos, la interfaz con el usuario y el sistema experto), y que deben poder comunicarse entre sí, en forma fluida, fuera de cualquier nivel de dificultad.

Esta ontología de dominio del conocimiento, representadas por una plantilla en el SE (ver Figura 1), no será la única que se deba pensar, existen otros caminos a esclarecer, como, la forma de comunicación de opinión y diagnóstico que el SE emitirá luego de un análisis. Dicha información será transferida desde el SE a la Interfaz del usuario, aquí tendremos que también estandarizar un flujo de información (mensajes) de otro tipo, que representa la transformación del conocimiento del SE aplicado a la valoración de un modelo (opinión, diagnóstico), fundado en una ontología de dominio. Más allá de estas dos anteriores, existirá otra ontología referida a la acumulación de conocimiento informativo para la Cátedra. En el SE las ontologías están reunidas, cada una en torno de una plantilla (ver Figura 2), que a su vez, y estas tendrán una imagen en una clases del Sistema Visual, y otra imagen en una tabla del Sistema de Base de Datos.

3.3. Estandarización de Vocabulario

Si se le permite al usuario expresar libremente el modelo de dato solución a un problema, seguramente el SE encontrará errores desde la primera letra a la última, o bien su conocimiento nunca será suficiente para efectuar la tarea esperada. Se deberá establecer un vocabulario estricto para mencionar las estructuras (entidades) y sus componentes, respetando rigurosamente una semántica que sea congruente con la ontología de dominio del conocimiento, se tendrá una ontología. Esta semántica, de cómo mencionar o expresar, dará sustento a una posibilidad real, de analizar los componentes de la solución, con el conocimiento ya existente dentro de la base de conocimiento.

```
(deftemplate formato_tipo_tabla "conocimiento -
plantilla para definir el formato de cada tipo_tabla"
        (slot tipo_tabla
                (type SYMBOL)
                (allowed-symbols
                        sin_estructura
                        padre_hijo
                        padre_multihijos
                        autoreferencia
                        referencia_tipo
                        referencia_maestro
                        muchos_a_muchos))
        (multislot atributo
                        (type SYMBOL))
        (slot pk
                        (type SYMBOL)
                        (allowed-symbols
                                si
                                no)
                        (default no) )
        (slot fk
                        (type SYMBOL)
                        (default nil))
        (slot tabla_referenciada_fk
                        (type SYMBOL)
                        (default nil))
        )
```

Figura 1. Ejemplo de plantilla para definir estructura de una tipo de tabla

3.4. Aprendizaje Orientado al Conocimiento

Establecida la ontología del dominio del conocimiento, se deberá diseñar una estrategia de cómo adquirirlo. Para determinar qué debe aprender el SE: a) Se debe descartar cualquier error por parte del usuario, para ello se deberá cotejar que la estructura (entidad) no conocida, respeta los estándares de vocabulario y expresión, b) Superada esta etapa, se le concederá el control al módulo de adquisición de conocimiento, que deberá en relación a las palabras utilizadas en la semántica de la entidad, y el formato de su estructura, decidir qué estructura del conocimiento existente en la BC es la más representativa. Las mismas están tipificadas dentro del dominio del conocimiento, c) Si el sistema no puede resolver eficientemente esta etapa de aprendizaje, iniciará un diálogo dirigido con el usuario, con el fin de esclarecer conceptos en duda, este diálogo poseerá una secuencia de preguntas dirigidas, para

esclarecer dentro de que tipificación estándar se incluye a la entidad [12].

```
(deffacts df2 "conocimiento -
 definición del FORMATO referencia_tipo para todas sus
instancias "
        (formato_tipo_tabla
                (tipo_tabla referencia_tipo)
                (atributo
                        identificador
                        id
                        pk
                        id_nombre_tabla
                )
                (pk si)
        )
        (formato_tipo_tabla
                (tipo_tabla referencia_tipo)
                (atributo
                        descriptor
                        descripcion
                        nombre
                        nombre_nombre_tabla
                )
        )
)
```

Figura 2. Ejemplo de estructura de tabla

3.5. Aprendizaje por Analogía

De todas las formas de aprendizaje que se pueden establecer, el experto humano, la que más utiliza, es la asociación de entidades por analogía. Esto lo hace tanto por la forma de mencionar (su nombre) una entidad, como también por cómo está construida (su estructura interna). Deduce la representación de una entidad: tanto por los atributos que se reúnen bajo una misma semántica que la determinan, como también por el nombre de la entidad, y de ello determina su estructura. Esto lo logra pues ya tiene un conocimiento previo estandarizado, y lo reutiliza para construir nuevas entidades (aprender) que nunca había tratado.

Teniendo: a) Un vocabulario para expresiones permitidas restringido, b) Con estructuras estandarizadas que representa el conocimiento, c) Junto a una biblioteca de nombres de entidades que se ajustan a cada tipo de estructura, d) Una interfaz visual que delimita fuertemente la posibilidad de utilización del vocabulario, a sólo el permitido, por parte del usuario. Resta elaborar una estrategia de reglas que puedan realizar las tareas de validación, y por otro lado un conjunto de reglas (ver Figura 3) que realice las tareas inherentes a la adquisición del conocimiento, para los casos en que la entidad analizada no se encuentre en la base de conocimiento [12,19].

3.6. Aprendizaje con Autonomía en la Inferencia

De la misma forma que el experto humano aprende, se establecerán un conjunto de reglas que exploren las estructuras estandarizadas de entidades que forman parte de la base de conocimiento. También se realizará un estudio del vocabulario del nombre de la entidad analizada y, en función de los resultados obtenidos, se inferirá una estructura que representa a la analizada. Se prevé establecer niveles de ponderación, que deberán ser superados para tomar la inferencia como aceptable [19].

```
(defrule r6-1  "analizar la estructura de
atributos de la tabla ingresada cuando no
se pudo lograr encontrar el nombre de tabla"
(resultado
        (regla r1-1)
        (tabla ?tabla_ingresada))
        (tabla_ing ( nombre_tabla ?tabla_ingresada)
        (atributo ?atributo)(pk  ?pki))
(formato_tipo_tabla
        (tipo_tabla ?tipo_tabla)
        (atributo $?antes ?atributo $?despues)
        (pk ?pk_formato))
( tabla
        (nombre_tabla $?ant ?tabla_posible $?desp)
        (tipo_tabla ?tipo_tabla))
=>
(assert  (analisis_2
        (tipo_tabla ?tipo_tabla)
        (atributo ?atributo)
        (pk ?pk_formato)
        (nombre_tablas  ?tabla_posible)))
(assert     (resultado
        (regla r6-1)
        (tabla ?tabla_ingresada)
        (tipo_tabla ?tipo_tabla)
        (tabla_posible ?tabla_posible)
        (atributo ?atributo)
        ( mensaje
                (str-cat
                "Conclución de búsqueda: La tabla "
                (implode$ (create$ ?tabla_ingresada) )
                "PODRÍA ser tratada como la tabla  "
                (implode$(create$ tabla_posible))
                " que responde al tipo "
                (implode$ (create$ ?tipo_tabla))))))
)
```

Figura 3. Ejemplo de regla

3.7. Aprendizaje por Interacción con el Usuario

Si los niveles de ponderación no ha sido superados, se deberá disparar una batería de preguntas al usuario, para esclarecer y tomar conocimiento adicional proveniente de

él, para llegar a un nivel de certidumbre y que el SE pueda tomar como valido el conocimiento que va adquirir.

Respecto del aprendizaje: Los tipos de estructuras para representar una solución no son infinitas, son un número reducido y cuantificable. Esto hace que el dominio sea claramente delimitado. Puede suceder que para una misma entidad, en un caso se requiera un atributo que para otro caso no, o también puede ser que se requiera un atributo desconocido. Por ello se ha estandarizado los tipos de entidades, en un conjunto limitado de caracterizaciones que tienden a cubrir el espectro posible de éstas. Esto no quiere decir que estén todas las entidades descritas; esto afirma que una entidad desconocida podrá ser encuadrada dentro de algún estándar [19].

4. Discusión

El paradigma de programación de SE está lejos del de la programación usual (POO, procedural), pero tiene singulares propiedades que permiten pensar soluciones con mayor riqueza de resultado.

En todo el proceso de investigación surgió la pregunta: ¿Por qué no está más difundido?

La unión de los dos paradigmas informáticos, sin dudas podría otorgar mejores y más productivas soluciones. No estamos diciendo que haya un desconocimiento pleno, pero sí, que su difusión no es importante en el ámbito general de la informática. Ya hay muestra tangible que esto es posible, en emprendimientos de Google, y algunos otros importantes medios universitarios.

Cuando dentro de un equipo de producción de desarrollos informáticos, se plantea la solución de un problema, en términos generales, sólo se apela a la Ingeniería del Software, nunca se menciona la Ingeniería del Conocimiento, por más que sea evidente que esta forma de ver y entender el problema, sea la más conveniente para determinados casos. ¿Será que hay un masivo desconocimiento de estas herramientas y metodologías?

La masividad de la bibliografía está orientada a la programación de recursos utilizando Ingeniería del Software, en las librerías especializadas se obtiene un número muy limitado de libros que hablan sobre Ingeniería del Conocimiento. Esto también es un indicativo del nivel de difusión que tiene este tipo de metodologías y tecnologías.

5. Conclusión

En este momento los integrantes del grupo de investigación se encuentran en pleno desarrollo de las ideas que dan fundamento al proyecto en su aspecto material. Se ha alcanzado suficiente conocimiento y claridad para retomar el desarrollo de la meta propuesta. Una parte del equipo está trabajando sobre los aspectos visuales de la interfaz gráfica, que todavía no implican el acuerdo ontológico que se requiere para la correcta transmisión de información (los distintos conocimientos: base de conocimiento, modelo de usuario, retroalimentación de aprendizaje). Otro grupo trabaja sobre la definición de la ontología dominio del conocimiento, tratando de que ésta tenga amplitud para abarcar todos los subconceptos que implican este dominio de conocimiento. Para esto último el equipo se encuentra corriendo lotes de pruebas con conjuntos de reglas que realizan tareas modulares, en prototipos ideales, con el fin de generar conocimiento de mecánicas de funcionamiento.

6. Agradecimientos

Al profesor Ingeniero Carlos Bartó que trajo luz a este equipo de investigación, cuando se iniciaba en el camino de la Ingeniería del Conocimiento y el diseño de Sistemas Expertos.

7. Referencias

[1] Agustín Carasco, Lucas Giacomazi,Ana Mercado - "Inferencia de modelos de base de datos normalizados" - 1º Congreso Nacional de Ingeniería Informática / Sistemas de Información - UTN-FRC - Simposio de Estudiantes. (Noviembre 21-22 2013 http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/PostersAlumnos/InfModelosNormalizadosBD.pdf (Septiembre 2013)

[2] Alanis Barrera, Ricardo- Sistemas Expertos e Inteligencia Artificial CLIPS - Burgos CyL., Febrero de 2005 - Universidad de Burgos: http://es.scribd.com/doc/6732509/Clips. Accedido en junio de 2013.

[3] Alonso Betanzos, Amparo et. al. - Ingeniería del Conocimiento. Aspectos Metodológicos - Editorial Pearson S.A. - Madrid - 2004 - ISBN 84-205-4192-3.

[4] Blanco Roberto; Casado, María Luisa; Cervera, David; Días de Prado, Fausto; Gómez Arias, José Javier; Martín, Francisco José; Martinez, José Miguel; Ramos, María José - Tecnología Investigación, innovación y buenas prácticas - Aprendizaje Automático - Editorial Graó - España - 2010 - ISBN 978-84-7827-995-1

[5] Calvo Cuenca, Antonio - Programación en lenguaje Clips - Editorial Universitaria Ramón Areces, 2008 - ISBN 9788480048798.

[6] Ceballos, Francisco Javier - Enciclopedia de Microsoft - Visual C# - Editorial Alfaomega - México - 2007 - ISBN: 9789701514252

[7] Ceballos, Francisco Javier - Visual Basic .NET - Curso de Programación - Editorial Alfaomega -Rama - México - 2008 - ISBN: 9701513703

[7] Ceballos Sierra, Francisco Javier - Enciclopedia de

Lenguaje C - Editorial Alfaomega - México - 1997 - ISBN: 9786077686439
[9] Damiano, Luis Esteban; Muñoz, Roberto Miguel; Paz Menvielle, María Alejandra; Romero María, Soledad; Cuevas, Juan Carlos; Bartó, Carlos - Aplicación en el ámbito académico del proyecto "Generador automático de modelos de datos normalizados en bases de datos relacionales" - IV Jornadas de Enseñanza de la Ingeniería - UTN-FRA - Septiembre 4-5 2014
[10] Date, Christopher - Introducción a los Sistemas de Bases de Datos - Volumen 1 - Quinta Edición - Edit. Addison Wesley Iberoamericana – Estados Unidos - 1993 - ISBN: 0-201-51859-7.
[11] Elmasri, R. y Navathe, S. - Fundamentos de Sistemas de Bases de Datos-5ta Edición – Edit. Pearson- Estados Unidos - 2007. ISBN: 978-84-7829-085-7.
[12] Garcia Martinez, Ramon - Sistemas Autónomos - Aprendizaje Automático - Editorial Nueva Librería - Argentina - 1997 - ISBN 950-9088-84-6
[13] García Martínez - Britos - Ingeniería de Sistemas Expertos - Editorial Nueva Librería - Bs.As. Argentina - 2004 - ISBN 9789871104154
[14] García Serrano, Alberto - Inteligencia Artificial - Fundamentos, práctica y aplicaciones - Editorial Alfaomega - México - 2012 - ISBN 9788493945022
[15] Giarratano, Joseph - Riley, Gary - Sistemas Expertos Principios y Programación - Editorial Cengage Learning / Thomson Internacional, 2005 - ISBN 9789706860590
[16] María A. Paz Menvielle, Juan C. Cuevas, Luis E. Damiano, Roberto M. Muñoz, Sergio Quinteros, - Generador Automático de Modelos de Datos Normalizados en Bases de Datos Relacionales - XV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN. Abr. 18-19 2013
http://sedici.unlp.edu.ar/bitstream/handle/10915/27120/Generador+Autom%E1tico+de+Modelos+de+Datos.pdf?sequence=1 (Septiembre 2014)
[17] Paz Menvielle, María Alejandra; Bartó, Carlos Alberto; Cuevas, Juan Carlos; Muñoz, Roberto; Damiano, Luis; Romero, María Soledad - Seguimiento y Control de Diseño en Proyecto de Ingeniería del Conocimiento - 1º Congreso Nacional de Ingeniería Informática / Sistemas de Información - UTN-FRC - Noviembre 21-22 2013
http://conaiisi.unsl.edu.ar/2013/237-648-1-DR.pdf (Septiembre 2014)
[18] Mario Bunge - Tratado de Filosofía – Ontología II – Editorial Gedisa – ISBN: 978-84-9784-197-9
[19] Pajares Martinsanz, Gonzalo y Santos Peñas, Matilde - Inteligencia Artificial e Ingeniería del Conocimiento - Editorial RA-MA - Madrid - 2007 - ISBN 84-7897-676-0.
[20] Reinosa, E.; Maldonado, C.; Muñoz, R.; Damiano, L.; Abrutsky, M. - Bases de Datos - Edit. AlfaOmega Editores - Argentina - 2012- ISBN: 978-987-1609-31-4.
[21] Silberschatz y otros- Fundamentos de Bases de Datos - Quinta Edición- Edit. Mc Graw Hill- Estados Unidos - 2006 - ISBN: 84-481-4644-1.
[22] Schreiber, August et. al. - Knowledge Engineering and Management. The CommonKADS Methodology - Editorial Massachusetts Institute of Technology - Estados Unidos - 2000 - ISBN 0-262-19300-0.
[23] Schildt, Herbert - C Manual de Referencia - Editorial Mc. Graw Hill - Madrid - 2004 - ISBN: 9789701047705
[24] Mario Bunge – Tratado de Filosofía – Ontología I – Editorial Gedisa – Barcelona 2011 – ISBN: 978-84-9784-196-2
[25] Wroblewski, Luke - Usabilidad y diseño de Formularios - 2011 -
http://www.slideshare.net/interactionpatterns.org/best-practices-for-form-design-presentation: (Ago 2014)

Desarrollo de una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica

Garbarini, Ramiro; Gaspes Ezequiel; Macri Matías; Osuna Gustavo; Larran Germán

Grupo de Estudio en Metodologías de Ingeniería de Software.
Laboratorio de Sistemas de Información. Departamento de Ingeniería en Sistemas de Información. Facultad Regional Buenos Aires. Universidad Tecnológica Nacional.
Facultad de Ciencias Naturales y Museo. Universidad Nacional de La Plata.

Abstract

La Argentina perdió en el siglo pasado casi el 70% de su bosque nativo. Como respuesta a esto, en 2007 se sanciono la Ley Nacional 26.331 de protección de Boques Nativos. Esta ley establece que cada provincia debe realizar el Ordenamiento Territorial de Bosques nativos localizados en su territorio, basados en diversos criterios de sustentabilidad, y definiendo tres categorías de conservación (I –rojo, II- amarillo, III-verde) basadas en el valor ambiental de las distintas unidades de bosque nativo y de los servicios ambientales que éstos presten.

En este contexto, el trabajo presenta la experiencia realizada para definir una metodología para el ordenamiento territorial de bosques nativos mediante la utilización de un producto software, desarrollad para tal fin, basado en tecnología de Sistemas de Información Geográfica (SIG).

El trabajo realizado contempló 5 etapas: 1- estudio del marco normativo de bosques nativos; 2- análisis de la propuesta de ordenamiento de Bosques Nativos de la Provincia de Buenos Aires; 3- elaboración de la metodología de apoyo para el ordenamiento territorial de bosques nativos (caracterización de los criterios de sustentabilidad aplicables a bosque nativos; establecimiento de una estrategia de priorización de los criterios de sustentabilidad aplicable a cada unidad de bosque nativo); 4- diseño y desarrollo del software que implementa la metodología elaborada (construido sobre plataforma ArcGis, utilizando Visual Basic .Net y ArcGIS Runtime SDK); 5- validación de la metodología elaborada con la Dirección Provincial de Recursos Naturales de la provincia.

Los resultados obtenidos se ajustan al trabajo con que se había construido el Ordenamiento Territorial de Bosques Nativos en la provincia. Los tiempos de respuesta del software fueron muy superiores, y la calidad de la información generada fue muy superior a aquella obtenida hasta el momento por el Organismo Ambiental provincial.

Palabras Clave

Sistemas de Información Geográfica, bosques nativos, ordenamiento territorial, evaluación áreas geográficas.

Introducción

La Argentina perdió en el siglo pasado casi el 70% de su bosque nativo. La extensión de la frontera agrícola aceleró en los últimos años esa tendencia destruyendo en promedio 200.000 hectáreas por año. Según el Inventario Nacional de Bosques Nativos del año 2005, todavía la Argentina posee 31,4 millones de hectáreas de bosques nativos, un patrimonio de alto valor ambiental. Según el Ordenamiento Territorial de Bosques Nativos de la Provincia de Buenos Aires, en su territorio existen 1.025.000 hectáreas de bosques nativos.

En noviembre de 2007 fue sancionada la Ley Nacional 26.331 [1], estableciendo los presupuestos mínimos de protección ambiental para el enriquecimiento, la restauración, conservación, aprovechamiento y manejo sostenible de los bosques nativos.

La Ley establece que cada provincia realice el Ordenamiento de los Bosques Nativos existentes en su territorio, de acuerdo a los criterios de sustentabilidad que la misma establece, según tres categorías de conservación basadas en el valor ambiental de las distintas unidades de bosque nativo y de los servicios ambientales que éstos presten.

Las tres categorías de conservación son: Categoría I (rojo): sectores de muy alto

valor de conservación que no deben transformarse. Incluirá áreas que por sus ubicaciones relativas a reservas, su valor de conectividad, la presencia de valores biológicos sobresalientes y/o la protección de cuencas que ejercen, ameritan su persistencia como bosque a perpetuidad, aunque estos sectores puedan ser hábitat de comunidades indígenas y ser objeto de investigación científica.; Categoría II (amarillo): sectores de mediano valor de conservación, que pueden estar degradados pero que a juicio de la autoridad de aplicación jurisdiccional con la implementación de actividades de restauración pueden tener un valor alto de conservación y que podrán ser sometidos a los siguientes usos: aprovechamiento sostenible, turismo, recolección e investigación científica; y Categoría III (verde): sectores de bajo valor de conservación que pueden transformarse parcialmente o en su totalidad aunque dentro de los criterios fijados por la ley.

La Categoría I no puede estar sujeta a aprovechamiento forestal, si bien podrán realizarse actividades de protección, mantenimiento, recolección y otras que no alteren los atributos intrínsecos, incluyendo la apreciación turística respetuosa, las cuales deberán desarrollarse a través de Planes de Conservación. También podrá ser objeto de programas de restauración ecológica ante alteraciones y/o disturbios antrópicos o naturales.

Las actividades a desarrollarse en la Categoría II, deberán efectuarse a través de Planes de Conservación o Manejo Sostenible, según corresponda.

Las actividades a desarrollarse en la Categoría III, deben efectuarse a través de Planes de Conservación, Manejo Sostenible o de Aprovechamiento del Cambio de Uso del Suelo según corresponda.

La Autoridad nacional de aplicación debe mantener actualizada la información sobre la superficie cubierta por bosques nativos y su estado de conservación. Por su parte, al igual que el Inventario Nacional, cada provincia debe actualizar el ordenamiento de Bosques Nativos en su territorio cada 5 años.

Elementos del Trabajo y metodología

El objetivo del trabajo fue desarrollar y validar una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica.

El trabajo realizado contempló las siguientes etapas: 1- estudio del marco normativo de bosques nativos Ley Nacional 26331/2007 y Decreto reglamentario 91/09 [2]; 2- análisis de la propuesta de ordenamiento de Bosques Nativos de la Provincia de Buenos Aires [3]; 3- elaboración de la metodología de apoyo para el ordenamiento territorial de bosques nativos (caracterización de los criterios de sustentabilidad aplicables a bosque nativos; establecimiento de una estrategia de priorización de los criterios de sustentabilidad aplicable a cada unidad de bosque nativo); 4- diseño y desarrollo del software que implementa la metodología elaborada; 5- validación de la metodología elaborada con la Dirección Provincial de Recursos Naturales de la provincia.

Etapas de trabajo

1- estudio del marco normativo de bosques nativos. La ley nacional 26331/2007 establece que cada provincia debe realizar el Ordenamiento de los Bosques Nativos existentes en su territorio, de acuerdo a criterios de sustentabilidad, definiendo tres categorías de conservación (I – rojo; II – amarillo; III - verde) basadas en el valor ambiental de las distintas unidades de bosque nativo y de los servicios ambientales que éstos presten. Incluye los parámetros que definen un bosque nativo, principalmente cobertura y especie vegetal predominante.

2- análisis de la propuesta de ordenamiento de Bosques Nativos de la Provincia de Buenos Aires. Se analizó la delimitación

geográfica de los bosques nativos realizada por la Dirección Provincial de Recursos Naturales del Organismo Provincial para el Desarrollo Sostenible de la Provincia de Buenos Aires (OPDS) y se indagó sobre el procedimiento utilizado por dicha dependencia para su elaboración.

3- elaboración de la metodología de apoyo para el ordenamiento territorial de bosques nativos.
3.a- Elementos para determinar un Bosque Nativo.
Cada unidad de bosque nativo propuesta se divide en una matriz de celdas rectangulares (por defecto media hectárea) que permite evaluar para cada celda si cumple con los requisitos para considerar un área bosque nativo:

- 0,5 hectárea (unidad de análisis) de ocupación continua.
- 20 % de cobertura de copas mínima.

3.b- Se identificaron y caracterizaron los criterios de sustentabilidad aplicables a bosques nativos contemplando:

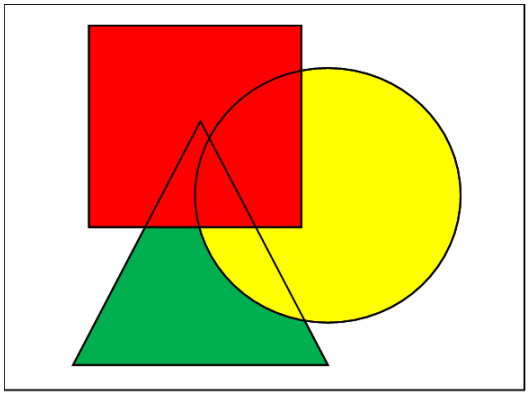
- Superficie
- Vinculación con otras comunidades naturales.
- Vinculación con áreas protegidas existentes e integración regional
- Existencia de valores biológicos sobresalientes
- Conectividad entre eco regiones (corredores boscosos y riparios)
- Estado de conservación
- Potencial forestal
- Potencial de sustentabilidad agrícola (aptitud del suelo)
- Potencial de conservación de cuencas (nacientes, márgenes de cursos de agua, sitios Ramsar, etc.)
- Valor que las Comunidades Indígenas y Campesinas dan a las áreas

Se estableció un dominio de valores (rojo, amarillo y verde) que cada criterio puede tener, en función a su criticidad respecto de su valor de conservación.

Se definió que para una unidad de bosque nativo cada criterio de sustentabilidad puede tomar dos valores: presente / ausente. Para cada unidad de bosque nativo a evaluar se definirán los criterios presentes, y para cada uno de estos se establecerá su criticidad respecto de su valor de conservación.

3.c- Respecto de la estrategia de priorización de los criterios de sustentabilidad aplicable a cada unidad de bosque nativo, se definieron dos métodos: i) superposición de criterios; ii) ponderación de valores.

i) Superposición de criterios. Debido a que en el análisis de una unidad de bosque nativo se pueden superponer espacialmente distintos criterios de sustentabilidad, puede ocurrir que, en un mismo punto o zona, se presenten más de un valor de conservación (rojo, amarillo, verde). En este caso, el criterio de priorización definido, establece que la prioridad será el valor rojo, luego el valor amarillo y finalmente el valor verde (ver figura 1).



[1] Superposición de criterios con valores diferentes. Figura 1.

i) Ponderación de valores. A cada ítem u opción de los criterios se le asignará, además del valor de conservación, un valor arbitrario (a determinar por el organismo) de impacto. La sumatoria de todos los impactos de los criterios presentes en una unidad de bosque nativo determinada será

el 100%, y con ese porcentaje se determinará que:

- Ante el caso de superposición de áreas, tiene prioridad el que tenga, en proporción del total del área, mayor peso a nivel de color.
- En caso de igualdad de porcentajes, tiene prioridad el mayor valor de conservación.

La tabla 1, presenta un ejemplo para una unidad de bosque nativo determinada:

Criterio	Opción	Valor de Conservación	Impacto	PESO
1	1	Rojo	5	25
	2	Amarillo	3	15
	3	Verde	2	10
2	1	Verde	2	10
	2	Amarillo	2	10
	3	Amarillo	3	15
	4	Rojo	3	15
RESULTADO			20	100

[1] Ejemplo de la definición de valores de los criterios de sustentabilidad para una unidad de bosque. Tabla 1.

Sumatoria de criterios que se superponen en el área geográfica evaluada suponiendo que se intersectan los criterios: 1.1 (rojo 25), 1.2 (Amarillo 15), 2.3 (Amarillo 15), en la Tabla 2 se ejemplifica el resultado obtenido:

Ítem	Suma	Proporción (%)
Rojo	5	25
Amarillo	6	30
Verde	0	0
Resultado	**AMARILLO**	

[2] Ejemplo del resultado de la ponderación y evaluación de los criterios de sustentabilidad para una unidad de bosque. Tabla 2.

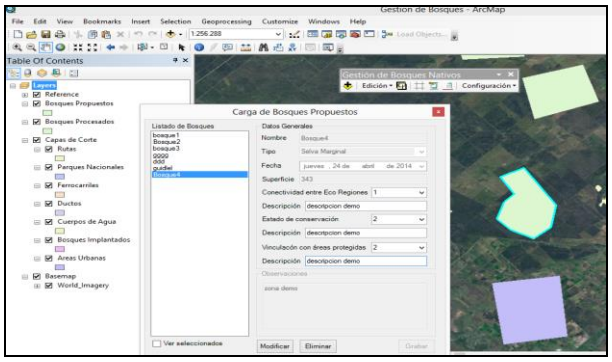
Si las proporciones de impactos resultasen iguales para amarillo y rojo, sin embargo, por ser categoría mayor, el área será categorizada roja.

4- diseño y desarrollo del software que implementa la metodología elaborada;

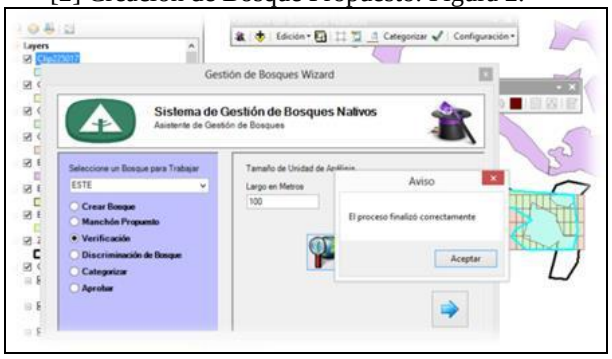
El producto software fue construido sobre plataforma ArcGis [4], utilizando Visual Basic .Net [5] y ArcGIS Runtime SDK [6] para el desarrollo. A partir del análisis de la normativa y de la metodología elaborada, se llevó adelante el diseño técnico de la funcionalidad a desarrollar y del modelo de datos, tanto geográfico como tabular. A continuación se listan los principales objetos ArcGIS que se utilizaron en el desarrollo, las funcionalidades construidas y los elementos más representativos del modelo de datos (Tabla 3):

- Objetos ArcGIS utilizados en el desarrollo:
  - Gridindex, genera una grilla sobre un polígono seleccionado.
  - Spatial Queries, permite realizar consulta geo-espaciales.
  - Intersect, permite evaluar la superposición entre distintos elementos geométricos.
  - Clip, elimina las geometrías que no estas superpuestas.
  - Field Calculator, permite realizar cálculos sobre atributos (campos) de elementos (tabla y/o featuresclass) de la base de datos.
  - Copy Features, almacena las geometrías generadas.
  Erase, elimina áreas seleccionadas.
  - Union, permite une geometrías y obtener un solo elemento geométrico.
  - Symetric Difference, realiza un Union y Erase juntos con geometrías superpuestas
  - SelectFeatures, selecciona geometrías.

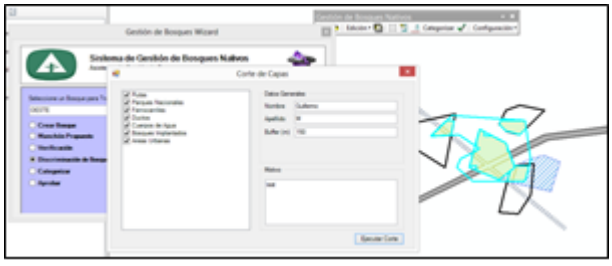
- Funcionalidades construidas:
  - Crear Bosque (ver figura 2)
  - Verificar (ver figura 3)
  - Discriminar (ver figura 4)
  - Categorizar (ver figuras 5 y 6)
  - Aprobar



[2] Creación de Bosque Propuesto. Figura 2.



[3] Verificación de Bosque Propuesto. Figura 3.



[4] Discriminar. Figura 4.



[5] Definición de Criterios de Sustentabilidad Ambiental para Categorizar una unidad de bosque. Figura 5.



[6] Unidad de bosque categorizada. Figura 6.

• Elementos más representativos del modelo de datos:

Nombre	Tipo	Geo-metría
Capas Geográficas		
fd_bosques	**Feature DataSet**	
fc_bosques_categorizados	FeatureClass	Polígono
fc_bosques_propuestos	FeatureClass	Polígono
fc_grilla	FeatureClass	Polígono
fc_suelo	FeatureClass	Polígono
fc_unidad_bosques	FeatureClass	Polígono
fd_criterios_sust	**Feature DataSet**	
fc_areas_protegidas	FeatureClass	Polígono
fc_areas_urbanas	FeatureClass	Polígono
fc_areas_valoracion	FeatureClass	Polígono
fc_bosques_otros	FeatureClass	Polígono
fc_cuerpos_agua	FeatureClass	Polígono
fc_departamentos	FeatureClass	Polígono
fc_ductos	FeatureClass	Línea
fc_ductos_p	FeatureClass	Polígono
fc_fauna	FeatureClass	Polígono
fc_ferrocarriles	FeatureClass	Línea
fc_ferrocarriles_p	FeatureClass	Polígono
fc_hidrografia	FeatureClass	Línea
fc_hidrografia_p	FeatureClass	Polígono
fc_localidades	FeatureClass	Punto
fc_rutas	FeatureClass	Línea
fc_rutas_p	FeatureClass	Polígono
fc_suelo	FeatureClass	Polígono
fc_valores_biologicos	FeatureClass	Polígono
Tablas		
tbl_bosques_nativos	Tabla	-
tbl_criterios_sust	Tabla	-
tbl_especies	Tabla	-
tbl_especies_bosque	Tabla	-
tbl_ley_decreto	Tabla	-
tbl_parametros_general	Tabla	-
tbl_recortes_capas	Tabla	-
tbl_suelo_bosque	Tabla	-

[3] principales elementos del modelo de datos. Tabla 3.

Los datos geográficos de la propuesta de Bosques Nativos fueron obtenidos de la Infraestructura de Datos Espaciales de la

Provincia de Buenos Aires [7]. Las estructuras de datos espaciales correspondiente a los criterios de sustentabilidad fueron diseñadas a partir de la información geográfica que está referenciada en coordenadas geográficas en el Sistema WGS 84, Marco de Referencia POSGAR 94 y su escala de captura es 1:250.000.
5- validación de la metodología elaborada con la Dirección Provincial de Recursos Naturales de la provincia de Buenos Aires.
Se tomaron diversas unidades de bosques nativos determinadas por la Dirección Provincial de Recursos Naturales, a las cuales se aplicó la metodología desarrollada y se realizó la operación con el software desarrollado, lográndose resultados que se ajustan al trabajo con que se había construido el Ordenamiento Territorial de Bosques Nativos en la provincia. Los tiempos de respuesta del software fueron muy superiores, y la calidad de la información generada fue muy superior a aquella obtenida hasta el momento por el Organismo Ambiental provincial, y la posibilidad de contar para cada resultado obtenido con el respaldo documentado de el / los criterios de sustentabilidad aplicados a cada caso, con su ponderación.
La metodología es adecuada. El software realiza el análisis correctamente, y representa geográficamente las unidades de bosque nativo, y logra presentar información clara como respaldo a los resultados alcanzados.

Resultados

Se ha desarrollado una metodología y un producto software adecuado para realizar en forma automatizada el ordenamiento territorial de los bosques nativos, aplicando los criterios de sustentabilidad establecidos por la Ley de Bosques Nativos. Esta metodología y producto software facilitan a los responsables de la gestión de Recursos Naturales de los distintos organismos ambientales de las provincias llevar adelante el proceso de creación de nuevas unidades de bosques nativos, o de actualización de los límites y categorías de las unidades de bosques nativos existentes.

La metodología y el producto software desarrollados aceleran el proceso de validación del Ordenamiento Territorial de Bosques Nativos que cada provincia realiza, agilizando el trabajo de campo que cada Organismo Ambiental debe llevar a cabo para finalmente terminar de ajustar la categorización de cada Unidad de Bosque Nativo. De esta manera, los organismos ambientales tienen a disposición una metodología de trabajo que plantea un análisis con mayor objetividad y precisión del utilizado hasta el momento en el proceso de ordenamiento territorial de bosques nativos. Como resultado de la utilización de esta metodología y software, los organismos ambientales tendrán mayor precisión y velocidad en el análisis y definición del ordenamiento territorial de bosques nativos.

Discusión

En primer lugar, los resultados del presente trabajo comprenden una metodología y un producto software basada en tecnología SIG para el ordenamiento territorial de los bosques nativos, que va más allá de un mero mapeo geográfico pues comprende el desarrollo de una herramienta de análisis que facilita la gestión de los bosques nativos en Argentina. El éxito de este proceso de construcción multidisciplinario permite anticipar la posibilidad de llevar este tipo de estrategias y proyectar el desarrollo de metodologías analíticas soportadas en sistemas de información geográfico, a otras áreas de gestión y conservación de los recursos naturales a cargo de entes gubernamentales.

Por su parte, cabe destacar que llevar a cabo este trabajo requirió superar diversos desafíos: hallar un lenguaje común entre profesionales con diferente formación (ingeniería en sistemas de información,

biología, recursos naturales); lograr estandarizar la visión de los gestores de recursos naturales y traducirla a un producto software, y que este producto pueda ser utilizado por éstos; disponer de datos actualizados y en formato de trabajo geo-espacial adecuado. Sin embargo, los resultados alcanzados muestran que superarlos es posible, y que la proyección de desarrollo de sistemas software en el campo de los recursos naturales es amplia, y se proyecta con gran potencial.

Conclusión

En función que la Ley Nacional 26.331 de bosques nativos establece que cada provincia debe realizar el Ordenamiento de los Bosques Nativos existentes en su territorio, de acuerdo a los criterios de sustentabilidad definidos en ella, según tres categorías de conservación basadas en el valor ambiental de las distintas unidades de bosque nativo y de los servicios ambientales que éstos presten, se propuso desarrollar y validar una metodología de apoyo para el ordenamiento territorial de los bosques nativos de la provincia de Buenos Aires mediante la utilización de sistemas de información geográfica.

El trabajo realizado contempló 5 etapas: 1- estudio del marco normativo de bosques nativos; 2- análisis de la propuesta de ordenamiento de Bosques Nativos de la Provincia de Buenos Aires; 3- elaboración de la metodología de apoyo para el ordenamiento territorial de bosques nativos (caracterización de los criterios de sustentabilidad aplicables a bosque nativos; establecimiento de una estrategia de priorización de los criterios de sustentabilidad aplicable a cada unidad de bosque nativo); 4- diseño y desarrollo del software que implementa la metodología elaborada (construido sobre plataforma ArcGis, utilizando Visual Basic .Net y ArcGIS Runtime SDK para el desarrollo. El diseño técnico de la funcionalidad a desarrollar y del modelo de datos, tanto geográfico como tabular se llevó adelante a partir del análisis de la normativa y de la metodología elaborada en la etapa 3); 5- validación de la metodología elaborada con la Dirección Provincial de Recursos Naturales de la provincia (los resultados obtenidos se ajustan al trabajo con que se había construido el Ordenamiento Territorial de Bosques Nativos en la provincia, sumado que los tiempos de respuesta del software fueron muy superiores, y la calidad de la información generada fue muy superior a aquella obtenida hasta el momento por el Organismo Ambiental provincial).

En síntesis, se ha desarrollado una metodología y un producto software adecuado para realizar en forma automatizada el ordenamiento territorial de los bosques nativos, aplicando los criterios de sustentabilidad establecidos por la Ley de Bosques Nativos. Esta metodología y producto software facilitan a los responsables de la gestión de Recursos Naturales de los distintos organismos ambientales de las provincias llevar adelante el proceso de creación de nuevas unidades de bosques nativos, o de actualización de los límites y categorías de las unidades de bosques nativos existentes.

Como conclusión podemos afirmar que la metodología y el producto software obtenidos son propicios para llevar adelante el ordenamiento territorial de los bosques nativos, aplicando los criterios de sustentabilidad establecidos por la Ley de Bosques Nativos, basado en el uso de sistemas de información geográfica.

Referencias

[1] Honorable Congreso de la Nación Argentina, 2007. http://www.infoleg.gob.ar/infolegInternet/anexos/135000-139999/136125/norma.htm Página vigente al 02/08/2014.

[2] Poder Ejecutivo Nacional, 2009. http://www.infoleg.gob.ar/infolegInternet/anexos/15

0000-154999/150399/norma.htm Página vigente al 02/08/2014.

[3] Organismo Provincial para el Desarrollo Sostenible, Dirección Provincial de Recursos Naturales de la Provincia de Buenos Aires, 2011 http://www.opds.gba.gov.ar/BNSite/ Página vigente al 02/08/2014.

[4] Esri, ArcGis http://www.esri.com/software/arcgis/arcgis-for-desktop. Página vigente al 02/08/2014.

[5] Microsoft, Visual Basic .Net http://msdn.microsoft.com/en-us/vstudio/hh388573.aspx Página vigente al 02/08/2014.

[6] Esri, ArcGIS Runtime SDK http://resources.arcgis.com/en/help/runtime-wpf/concepts/index.html#/Getting_started_with_Visual_Studio/01700000002q000000/ Página vigente al 02/08/2014.

[7] Infraestructura de Datos Especiales de la Provincia de Buenos Aires http://sig.gobierno.gba.gov.ar/sig/ideba/ Página vigente al 02/08/2014.

Datos de Contacto:

Ramiro Garbarini, Matías Macri, Universidad Tecnológica Nacional Facultad Regional Buenos Aires. Medrano 951, Ciudad Autónoma de Buenos Aires. Ramiro.garbarini@gmail.com; mmmatias@gmail.com
Ezequiel Gaspes, Germán Larrán. Facultad de Ciencias Naturales y Museo. Universidad Nacional de La Plata. Avenida 122 y 60 - La Plata ezequiel@iadsargentina.org; germanlarran@gmail.com

Rediseño E Implementación De Herramientas Informáticas Para La Auditoria De Sistemas En Red

Ciolli María Elena, Porchietto Claudio, Rossi Roberto
Grupo de Investigación Instituto Universitario Aeronáutico
Córdoba, Argentina

Resumen. *Esta ponencia presenta el resultado del análisis e implementación de herramientas para el control remoto del hardware y software de una red informática. Tras un análisis comparativo basado en la conceptualización GLPI (gestión libre del parque informático) y en la norma ISO 27002 dominio 7 (gestión de activos) sección 7.1 (inventario de activos) entre dos plataformas: OCS Inventory NG [1] y Open Audit [2], se concluyó que la segunda plataforma es la que más se adecua a los objetivos planteados [3]. Sin embargo también reveló que no satisface dichos objetivos en su totalidad, por lo que fue necesario reprogramar su código fuente corrigiendo errores y agregando nuevas funciones tales como detección de intrusos con NMAP, centralizado de configuración, Autenticación de scripts de auditoria, formularios en HTML5, etc.*

A través del presente trabajo se pretende mostrar la experiencia, los resultados y los motivos por los que se sigue trabajando en la herramienta Open Audit luego de haberla desplegado a la red interna del Instituto Universitario Aeronáutico. El mismo cuenta con un plantel de más de 1000 máquinas, repartidas entre las distintas dependencias del Instituto Universitario Aeronáutico (IUA), en su sede central y centros de apoyo de Rosario y Buenos Aires. El despliegue se realizó a través de los dominios internos, fuera de áreas críticas, sobre un total de 70 hosts auditados. Este desarrollo fue hecho sobre máquinas basadas sobre entorno Windows, por lo que se continúa la experimentación de auditorías sobre host fuera de dominio o intrusos.

Palabras clave: Open Audit, GLPI, Auditoria, Monitoreo, NMAP.

INTRODUCCIÓN

Los distintos entes que intervienen en una auditoría son:

- Estaciones de trabajo.
- Auditor.
- Informe o reporte.
- Base de datos.
- Estación para el análisis de datos.
- Lista de procedimientos.

Este modelo pese a ser muy rudimentario y simplista, nos ayuda a contextualizar el origen de nuestro trabajo.

El rol del auditor lo encarna una persona física. El informe o reporte obtenido, es transportado en un medio físico y la base de datos la constituye una PC donde se guardan los informes [4] [5]. Todo esto se ejecuta en base a unos procedimientos internos estandarizados y normalizados.

Como resulta evidente, es muy ardua la tarea de tener actualizada dicha base de datos, por lo que resulta imprescindible la investigación, el desarrollo y la implementación de un software que permita el control automático y la generación y actualización de reportes [6] del parque informático, mediante supervisión de la base de datos.

El esquema concebido está centrado en la auditoría de las máquinas que pertenecen a una red determinada. Todas las máquinas pueden contactarse con un servidor. A priori esta red se encuentra segmentada, contando con diferentes dominios, sistemas operativos y usuarios. De más está decir que una herramienta de auditoría es netamente un sistema distribuido en toda la red.

¿Cómo funciona Open Audit para auditar un dominio determinado?

La figura 1 muestra un esquema general de auditoría de dominio con la herramienta Open Audit.

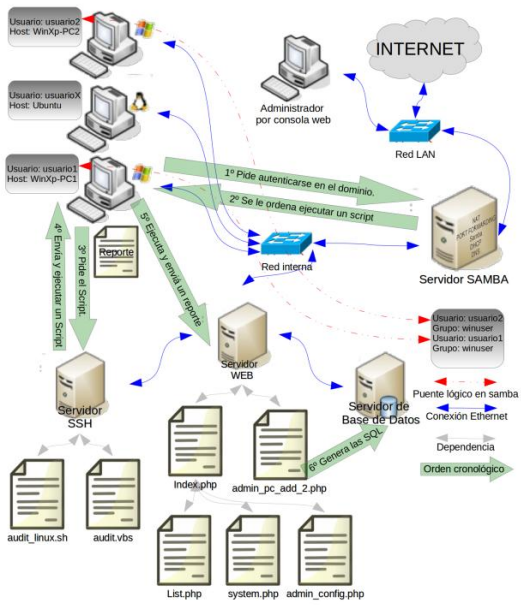


Figura 1 - Auditoria de dominio

Al iniciar sesión los usuarios del dominio se registran ante el PDC (controlador primario de dominio). En este caso particular es implementado por un servidor Samba, que instruye a ejecutar el Script de auditoría a las estaciones de trabajo. Dependiendo del sistema operativo el Script es diferente [7] [8] pero su reporte está estandarizado.

El Script para Linux está compuesto de una serie de sentencias de consola cuya salida es analizada, clasificada y segmentada por herramientas para procesar texto como awk.

En Windows se utiliza un Script semejante al de Linux que está codificado en Visual Basic Script y basado en instrucciones de WMI Service (Windows Management Instrumentation).

¿Cómo funciona Open Audit para auditar máquinas fuera del dominio?

Un servidor con un método de Polling [9] es el encargado de enviar el auditor. En la figura 2 se observa que aunque el segundo proceso de distribución parece más simple, no lo es, ya que es necesario cargar máquina por máquina en una lista con sus correspondientes nombres de usuario con privilegios de administrador.

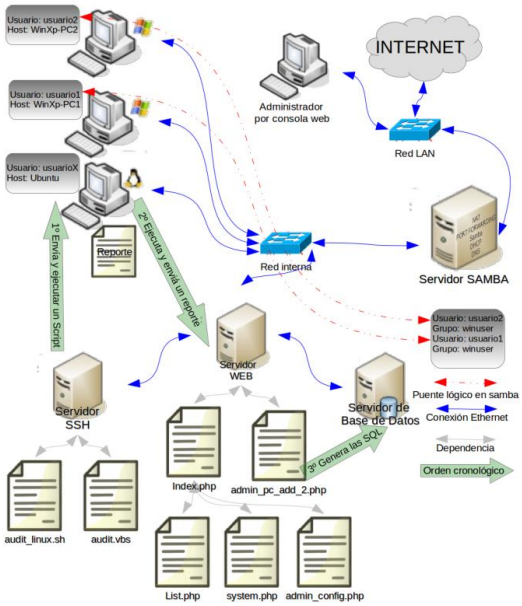


Figura 2 - Auditoria fuera del dominio

Todo esto conlleva a la necesidad de tener una gestión distribuida en la red y no concentrada.

¿Cómo almacenan la información?

Ambos Scripts guardan en cadenas de texto la información extraída que contiene un identificador de cabecera y caracteres especiales como separador de campos. Estas cadenas son almacenadas temporalmente en un archivo de texto (reporte) cuyo nombre es el de la máquina auditada. Una vez realizadas todas las consultas, el archivo de reporte es enviado vía html, previa autenticación, al servidor del Open Audit que se encarga de cargar los datos en el servidor de base de datos.

¿Cómo se genera un reporte del estado general de la red?

En la figura 3 se aprecia cómo es el flujo de información a la hora de realizar una consulta. No es necesario que las máquinas auditadas estén en línea al momento de realizar la consulta. Esto cumple con la disponibilidad de datos pedida por la normativa [10] [11] [6].

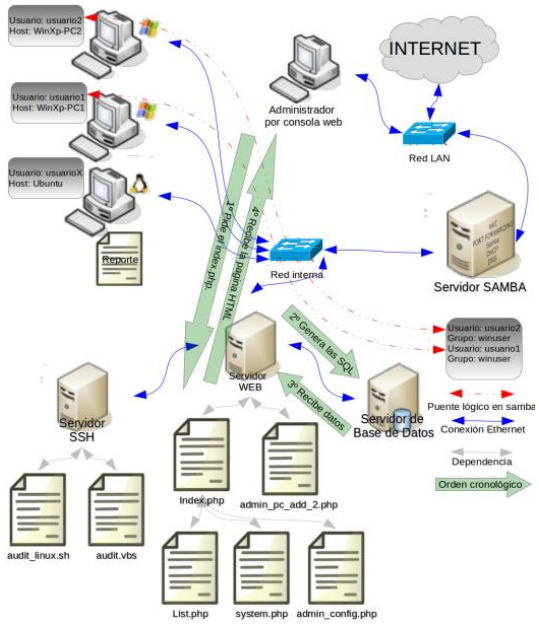


Figura 3 - Consulta generica

En concordancia con el modelo pre sistema, el rol del auditor es cambiado de una persona física a un Script, el reporte que se trasladaba y almacenaba manualmente ahora lo hace a través de la red LAN interna cuyo único requerimiento es que brinde conectividad entre las estaciones de trabajo y el servidor del Open Audit.

Los procedimientos estandarizados están almacenados en el controlador de dominio que es quien va a decidir qué máquinas son auditadas y cuándo. Para el caso de máquinas que no pertenecen a ningún dominio pero sí son parte de la zona auditada, el servidor de Open Audit efectúa una conexión mediante SSH con cada uno de los hosts que estén registrados.

¿Qué sucede con los Host intrusos?

Al no estar blindada la red, los usuarios pueden conectarse a la misma con sus ordenadores personales. Esto sucede por un tema de flexibilidad, ya que muchos usuarios quieren compartir archivos entre sus ordenadores y los ordenadores que se encuentran en la zona auditada.

Todos los hosts que no están en la lista de auditoría o se autentican contra un controlador de dominios o son considerados intrusos [12].

Los mismos si no son auditados presentan un riesgo para la seguridad de la red.

Una solución a esta situación fue integrar la herramienta Nmap a la suite de Open Audit. Con ella se puede detectar los hosts que están conectados a la red, todos ellos. A los mismos se les extrae información como la dirección MAC, la dirección IP, y el nombre de host. Estos datos son comparados en el servidor de Open Audit con la lista de estaciones de trabajo auditadas y de no corresponder a ninguna se los almacena como "host no auditado" representando una alerta de seguridad.

En la figura 4 se puede apreciar cómo es insertado el servidor Nmap al esquema de funcionamiento de la red.

Como en los casos anteriores en la implementación realizada en la zona auditada, el servidor de Nmap está implementado sobre el mismo servidor físico donde se encuentra instalado el Open Audit así como el servidor MySQL.

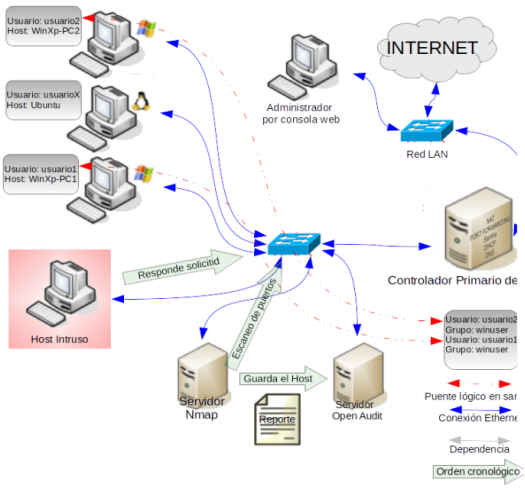


Figura 4 - Nmap

Gracias a la adición de Nmap al esquema de auditoría se puede tener un control más estricto del total de máquinas reales que contiene la red momento a momento. A pesar de ello no es todavía posible auditar los host intrusos.

METODOLOGÍA

En base a los resultados de las pruebas efectuadas, la herramienta Open Audit resultó seleccionada como plataforma base para continuar con el desarrollo en curso, tras la comparación cuantitativa de las auditorías realizadas.

Para cuantificar las plataformas se tomó como referencia la fidelidad de los datos, el volumen de tráfico de red y las facilidades que brinda la herramienta para poder ser desplegada. Todo esto fue introducido en la tabla 1 acompañado por un cuantificador de importancia dando como resultado una puntuación fácilmente comparable.

Cada elemento a auditar en el entorno de evaluación de la herramienta, posee características documentadas que sirven como base de comparación con las extraídas por el auditor. En base a cuan fiel es cada auditoria se le otorga un valor entre 1 y 5.

No todos los elementos auditables tienen el mismo grado de importancia. Conocer el número de serie de un ratón no es tan relevante como el de un disco duro o el de un software con licencia. Por ende es necesaria la introducción de una columna que pondere los valores de fidelidad. Los valores de dicha columna son arbitrarios y elegidos por un administrador de la red, ya que su experiencia valida su magnitud.

Finalmente la calificación está conformada como la suma del todos los valores de fidelidad ponderados. El rango se sitúa entre 88 y 17. Open Audit obtuvo 71.2, 70.2 y 63.8 en Windows XP, Windows 7 y Ubuntu respectivamente contra 68.2, 65 y 49.8 de OCS Inventory en el mismo orden.

El mismo método fue utilizado para controlar el avance de la plataforma Open Audit. Los nuevos resultados obtenidos se situaron en 78.8, 77.8, 66.0 y 69 para Windows XP, Windows 7, Ubuntu 14.04 y Windows 8.

Este método de cuantificación no contempla mejoras introducidas a la herramienta como la detección de intrusos. No obstante los cambios en el indicador garantizan que la herramienta está en proceso de evolución.

RESULTADOS

De la información recogida en la prueba y muestra efectuada, el total de estaciones de trabajo fueron de 91 y el total de servidores de dominio 3.

En la figura 5 se aprecia la estructura de red.

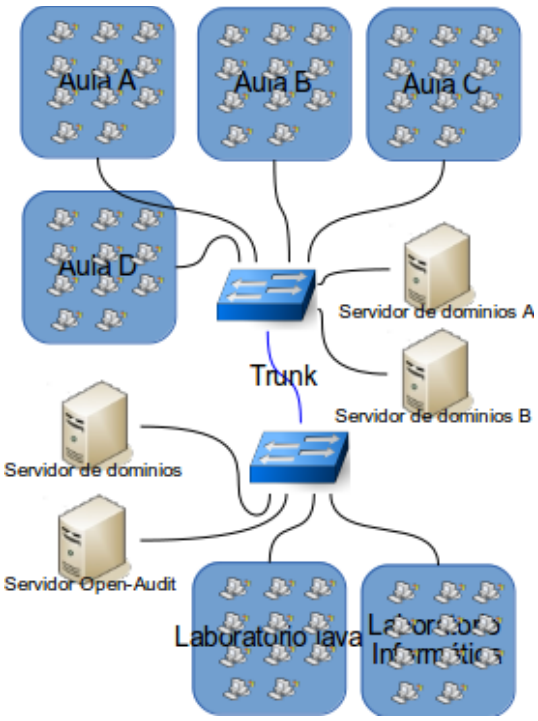


Figura 5 - Dominios donde se desplegó Open Audit

En el tiempo de prueba el sistema logró auditar 86 estaciones de trabajo y los tres servidores. Las restantes PC no se auditaron por no haber iniciado sesión en el dominio en el tiempo que lleva en funcionamiento el servidor.

Además se localizaron un total de 21 Host intrusos gracias a Nmap. Es de aclarar que todo dispositivo de red es detectado por un escaneo de puertos, tanto routers, como switchs administrables, etc. No obstante estos dispositivos de red no representan un problema para el sistema, dado que se le reporta la detección al administrador una vez y luego se procede a ignorarlos.

En cuanto a la performance del servidor, en la figura 6 se presenta un histograma del consumo de memoria y espacio libre en disco.

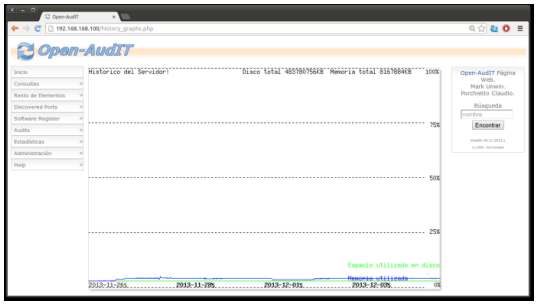


Figura 6 - Histograma

El impacto en el hardware es prácticamente inapreciable para el total de 91 estaciones de trabajo, lo que nos alienta a desplegarlo en un dominio de mayor dimensión. Los impactos en el hardware del servidor y en la red cableada son ínfimos comparados con el tráfico que genera el inicio de sesión en dominio de Windows.

El Sistema funciona sobre redes LAN, siendo posible su aplicación en redes WAN, pero en este caso es necesario modificar su estructura para tal integración.

Las figuras 7, 8, 9 y 10 presentan las distintas interfaces desplegadas por la herramienta.

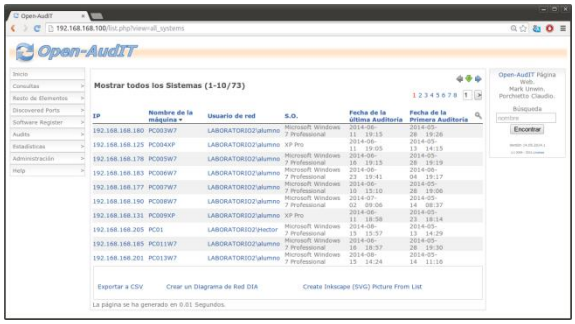


Figura 7 - Todos los Sistemas Auditados.

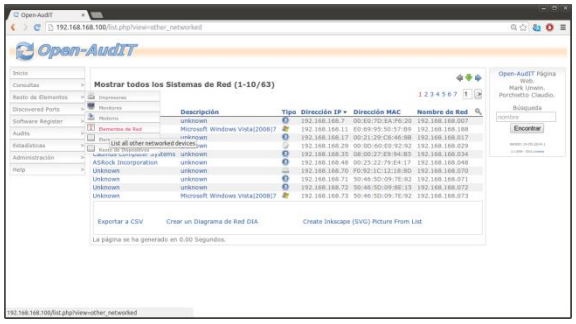


Figura 8 - Hosts descubiertos por Nmap.

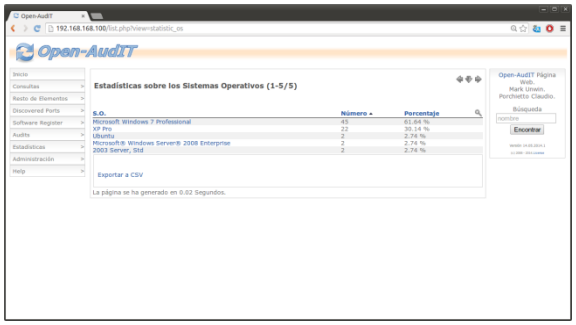


Figura 9 - Estadísticas sistema Operativo.

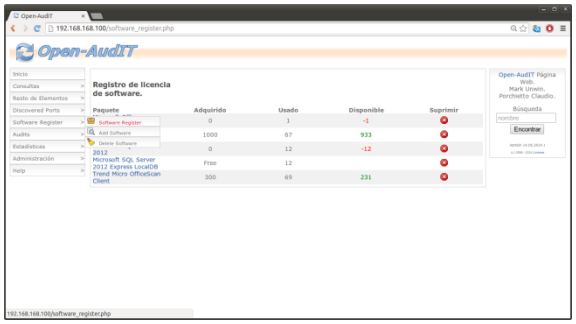


Figura 10 - Seguimiento de Software.

CONCLUSIONES

Con las mejoras realizadas a la suite Open Audit tanto en la depuración de los agentes auditores como en la integración con Nmap, se puede garantizar que el plantel de estaciones de trabajo se encuentra monitoreado de forma permanente, alertando ante la posible remoción de algún hardware en los equipos que se encuentren en la zona auditada como así también ante la instalación de nuevo software no autorizado. Además cualquier host conectado a la red es detectado procesado y notificado si no está auditado.

TRABAJOS FUTUROS

Actualmente se está trabajando en forzar la auditoría de las estaciones de trabajo que no están en un dominio ni en una lista, o sea los "hosts intrusos". Esta nueva etapa es presentada como "Sistema de control de acceso discriminado para estaciones móviles".

Es habitual que los usuarios de la red tengan sus propias estaciones móviles. Además estos quieren compartir archivos entre sus estaciones móviles y las estaciones de la institución. Es por ello que dichas estaciones ponen en riesgo la red si no son auditadas periódicamente.

Se está implementado un prototipo de Access Point (AP) WI-FI en el que se discrimine dispositivos de usuarios públicos y usuarios privados. De este modo el equipamiento móvil de los usuarios perteneciente a los usuarios con privilegios en la red puede acceder a la red cableada privada y los usuarios sin privilegios o invitados pueden tener acceso a una red pública. El usuario que desee conectarse a la red interna se deberá dar de alta ante el AP. En ese momento el AP comprobará que se ejecute el agente de auditoría.

Referencias

[1] MODx. (2013, Aug.) www.ocsinventory-ng.org. [Online]. http://www.ocsinventory-ng.org/en/

[2] Mark Unwin. (2013, May) open-audit.org. [Online]. http://www.open-audit.org/

[3] Porchietto Claudio. (2013, Oct.) www.unpl.edu.ar. [Online]. http://sedici.unlp.edu.ar/bitstream/handle/10915/31337/Documento_completo.pdf?sequence=1

[4] Jose Antonio Echenique Garcia, *Auditoria en Informática*.: Méjico - McGraw Hill, 2001.

[5] Chris Jackson, *Network Security Auditing*, Primera ed.: Cisco Press, 2010.

[6] Agustín López Neira. (2013, Oct.) www.iso27000.es. [Online]. http://iso27000.es/

[7] Mario Piattini Velthuis, *Auditoria de Tecnología y Sistemas de Información*, Primera ed.: Alfaomega, 2008.

[8] Abe Fettig, *Twisted Network Programming Essentials*, Segunda ed.: O'Reilly, 2013.

[9] Chris McNab, *Network Security Assessment: Know Your Network*, Segunda ed.: O'Reilly Media, 2007.

[10] Sarahol. (2013, Nov.) /www.17799.com. [Online]. http://www.17799.com/

[11] 123 Innovation Group, S.L. Auditoria Sistemas. [Online]. http://auditoriasistemas.com/estandares-ti/

[12] Jean-Philippe Martin-Flatin, *Web Based Management of IP Networks & Systems*, Primera ed.: Wiley, 2002.

[13] 123 Innovation Group S.L.. (2014, Mar.) auditoriasistemas. [Online]. http://auditoriasistemas.com/auditoria-de-sistemas-informaticos/

[14] Barzan "Tony" Antal, *It Inventory and Resource Management with Ocs Inventory Ng 1.02*, 102nd ed., Barzan "Tony" Antal, Ed.: Packt Publishing , 2010.

ANEXO

Herramienta / *Atributo*	*Valoración de Importancia*	*OCS inventory Windows xp*		*OCS inventory Windows 7*		*OCS inventory Ubuntu*		*Open Audit Windows xp*		*Open Audit Windows 7*		*Open Audit Ubuntu*	
Software de base con licencia - Sistema Operativo	5	5	5	4	4		0	5	5	4	4		0
Actualizaciones de Sistema Operativo	3	5	3	5	3	5	3	5	3	5	3	5	3
Software de aplicaciones con licencia	5	4	4	4	4		0	5	5	5	5		0
Antivirus	4	4	3,2	4	3,2		0	5	4	5	4		0
Software gratuito	4		0		0	5	4		0		0	5	4
Inventario de Hardware			0		0		0		0		0		0
Motherboard	5	2	2	2	2	2	2	4	4	4	4	4	4
Procesadores	5	4	4	4	4	4	4	5	5	5	5	5	5
Memoria	5	4	4	4	4	4	4	4	4	4	4	4	4
Almacenamiento físico HDD	5	5	5	4	4	5	5	4	4	4	4	5	5
Almacenamiento físico (CD, pen, etc)	5	4	4	4	4	4	4	4	4	4	4	3	3
Almacenamiento lógico	5	5	5	5	5	5	5	5	5	5	5	5	5
Video	5	3	3	3	3	3	3	3	3	3	3	3	3
Sonido	3	3	1,8	3	1,8	3	1,8	3	1,8	3	1,8	3	1,8
Red	5	5	5	5	5	5	5	5	5	5	5	5	5
BIOS	5	4	4	4	4	4	4	4	4	4	4	5	5
Monitor	4	5	4	5	4	1	0,8	5	4	5	4	5	4
Dispositivos de entrada.	3	4	2,4	4	2,4	2	1,2	4	2,4	4	2,4	4	2,4
Impresoras	4	4	3,2	4	3,2		0	2	1,6	2	1,6	4	3,2
Impacto en red			0		0		0		0		0		0
Volumen de tráfico en la auditoría	4	3	2,4	3	2,4		0	3	2,4	3	2,4	3	2,4
Volumen de tráfico en el despliegue	1	1	0,2	1	0,2		0	5	1	5	1	5	1
Facilidades			0		0		0		0		0		0
Desligue	3	5	3	3	1,8	5	3	5	3	5	3	5	3
TOTAL			68,2		65		49,8		71,2		70,2		63,8

Tabla 1 - Tabla de valoraciones

Optimización de Constantes Numéricas en Regresión Simbólica utilizando un Framework de Tree-Based Genetic Programming

Elli, Soledad[1], Jimenez, Victor A.[2], Will, Adrian[1,2], Rodríguez, Sebastián[2]
[1] Universidad Nacional de Tucumán, Facultad de Ciencias Exactas y Tecnología
[2] Grupo de Investigación en Tecnologías Avanzadas, U.T.N. – F.R.T.

Abstract

La programación genética (PG) es un conjunto de técnicas de computación evolutiva que permiten resolver problemas automáticamente y que están basadas en Algoritmos Genéticos (AG). La PG ha sido un método muy utilizado para encontrar y evolucionar nuevas e inesperadas maneras de resolver problemas, en particular problemas de Regresión Simbólica (RS). A pesar de esto, uno de los problemas de la PG es la falta de exploración y optimización de las constantes numéricas (o parámetros) dentro de la estructura de árbol con la que se representa un programa. Este trabajo se enfoca en investigar e implementar distintos métodos para la optimización de estas constantes que se encuentran en las terminales de los árboles, de manera de comparar las ventajas relativas de cada método. Los métodos implementados fueron validados con 7 problemas de benchmark.

Palabras Clave

Programación Genética, Tree-Based Genetic Programming, Optimización de constantes numéricas, Regresión Simbólica.

1. Introducción

La programación genética es una metodología basada en algoritmos evolutivos, los cuales están inspirados en la evolución biológica. Es una especialización de los algoritmos genéticos que intenta encontrar soluciones a problemas a partir de la inducción de programas. Existen numerosas variantes de estos algoritmos, que se diferencian principalmente por la forma de representar estos programas. En particular, Tree-Based Genetic Programming (Tree-GP) utiliza la codificación basada en estructura de árboles [1].

De manera similar a los AG, Tree-GP inicia con una población cuyos individuos son generados aleatoriamente. A esta población se le aplica operadores de selección, cruzamiento y mutación, lo cual deriva en una nueva población de individuos en cada generación. Este proceso se repite hasta que se verifica una condición de terminación, que por lo general consiste en alcanzar una cierta cantidad de generaciones. Al igual que en la naturaleza, este es un proceso con muchas componentes aleatorias, y no hay hipótesis sobre el problema que permitan garantizar que se encontró el óptimo (como derivadas, convexidad, etc.). En la resolución de problemas con estos métodos se considera suficiente encontrar una buena solución, aunque no sea necesariamente el óptimo global [2], [3], [4].

Una de las aplicaciones de la PG y un área activa de investigación es Regresión Simbólica (RS). Esta técnica consiste en encontrar, a partir de un conjunto de datos experimentales, la expresión algebraica que identifica su comportamiento. Dado que es necesario encontrar una respuesta de manera tangible a datos obtenidos, surge la pregunta (y la propuesta): ¿Cómo evidenciar las relaciones que existen entre una variable dependiente y, y una o más variables independientes $(x_1, x_2, \ldots, x_n)$ para poder predecir, pronosticar, conocer o controlar algo? Es ahí cuando entra en juego la PG ya que, debido a la naturaleza de cada problema, no todos se pueden resolver de manera determinística usando métodos numéricos que respondan a los datos de entrada. GP nace para dar respuesta a estos problemas.

En el caso de RS aplicada a GP, la función de Fitness debe estar diseñada para dar mejores valores a soluciones que ajusten mejor los valores de salida sobre los datos de entrada dados. En el presente trabajo la función fitness se define como el Error Cuadrático Medio (ECM):

$$ECM_i = \sqrt{\frac{\sum|valorOriginal_i - valorCalculado_i|^2}{n}}$$

La particularidad que tiene la PG sobre los problemas de RS es que los nodos de los árboles que forman los individuos pueden ser de dos tipos: operador $(+, -, *, /)$ y nodo terminal que, a su vez, puede ser una variable o una constante $(a, y, a, 1.2, 5)$. Si consideramos sólo operadores de aridad dos, podemos representar las expresiones con árboles binarios. Esta simplificación permite evaluar la expresión recorriendo el árbol recursivamente en preorden calculando el valor en cada nodo. En la figura 1 se observa un ejemplo de la representación de una ecuación particular mediante un árbol binario.

En general en PG y como método base (ver [1]), como primera aproximación se fija un conjunto de constantes numéricas que se juzgan oportunas o importantes para el problema (por ejemplo $1, 2, 3, \pi, e$). Sin embargo es muy probable que este conjunto de valores prefijados no sea suficiente para obtener buenas soluciones, por lo que es necesario el uso de algoritmos que exploten y optimicen estas características de las expresiones obtenidas por el AG.

$$f(x) = \frac{sen\big(x(-6.39589 + 3.27152)\big) - \text{sen}(-3.39689)}{1.47056^x}$$

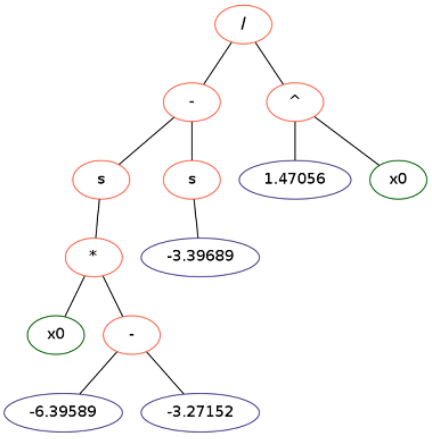


Figura 1: Ejemplo de un árbol de regresión simbólica.

El objetivo del trabajo es implementar los métodos más conocidos para manejo de constantes en Tree-GP, y analizar sus ventajas y desventajas en una serie de problemas conocidos, de manera de poder utilizarlos más eficientemente en problemas reales. Para lograr este objetivo se utilizó y adaptó un framework de Programación Genética desarrollado en trabajos anteriores [5], [11]. Este framework desarrollado en lenguaje C++ cuenta con la estructura básica de los algoritmos genéticos para su aplicación en programación genética basada en Tree-GP. Su diseño permite utilizar diferentes operadores genéticos tradicionales, pero a su vez permite el uso de operadores particulares, que el usuario final puede implementar para problemas específicos.

Este trabajo está estructurado de la siguiente manera: En las siguientes secciones se describen las características generales del problema en particular de la optimización de las constantes en PG para RS. Luego se describen los métodos implementados y utilizados en los casos de estudio y, por último, los resultados obtenidos y conclusiones.

2. Problema de las constantes numéricas

La problemática de las constantes tiene su origen en una de las debilidades que presenta la PG, el descubrimiento de los valores numéricos en los nodos terminales. Como se dijo anteriormente, Tree-GP utiliza un conjunto de funciones y un conjunto de terminales en el momento de armar los árboles que representarán a los individuos. En ambos conjuntos se pueden encontrar una variedad muy amplia de elementos en función del dominio del problema que se intenta resolver.

En este punto podríamos formularnos una pregunta muy importante: ¿Por qué la PG es tan vulnerable a problemas de elección de constantes, si los AG de los cuales deriva son especialmente buenos para resolver justamente este problema? Y la

respuesta a esa pregunta tiene que ver con la codificación utilizada en la PG [1]. El problema surge porque los operadores genéticos para Tree-GP operan sobre la estructura de un árbol, y no sobre los valores que están contenidos en los nodos de éste, no permitiendo que estos evolucionen hacia sus valores óptimos.
La forma de afrontar el problema de las constantes numéricas es uno de los factores más importantes para lograr el éxito del algoritmo y mejorar los tiempos de ejecución del mismo. Si la expresión obtenida tiene las constantes apropiadas es más probable que el algoritmo obtenga a una mejor solución que si no las tiene. Por otro lado, mientras más constantes estén involucradas en el problema, mayor será el tiempo necesario para encontrar buenos valores numéricos para las mismas.

3. Métodos para optimizar constantes

Existe una gran variedad de métodos para optimizar constantes en RS. De los métodos más simples podemos nombrar:

- Constante aleatoria efímera
- Constantes pre-especificadas
- Funciones sin argumentos
- Combinación Aritmética
- Génesis Aritmético

Además de estos métodos existen otros más complejos que intentan solucionar problemas específicos relacionados a la búsqueda de constantes numéricas. Por lo general, los métodos más sofisticados dependen del fitness y/o de algún parámetro adicional que, combinados de cierta forma permiten obtener un mejor valor para la constante numérica a reemplazar. Algunos métodos definen un rango para limitar los valores numéricos que se obtienen de las modificaciones que sufren las constantes a través de las generaciones o de acuerdo al valor de fitness. A continuación se describen los métodos que se implementaron para el desarrollo de este trabajo.

3.1 Numeric Mutation

Numeric Mutation (NM) [6] propone reemplazar cada constante por un valor generado aleatoriamente dentro de un rango determinado. El rango se determina tomando el valor actual de la constante y sumándole o restándole el valor del factor de temperatura actual. El factor de temperatura (TF, *Temperature Factor*) está determinado por el producto entre el valor del fitness del mejor individuo de la generación actual ($bestFitness_i$) y una constante especificada por el usuario (TC, *Temperature Constant*).

$$TF = bestFitness_i * TC$$

La efectividad del método puede explicarse de la siguiente forma. Cuando el mejor individuo de la población tenga un bajo fitness, el rango de selección será más grande, por lo que el cambio en los valores de las constantes numéricas será mayor. Por el contrario, cuando existan buenos individuos en la población, el valor del fitness se aproximará a cero, lo que provocará que el rango de variación se reduzca progresivamente favoreciendo la búsqueda local alrededor de la mejor solución. Es importante destacar que el funcionamiento se basa en que el Fitness **se aproxime a cero** ($bestfitness_i$ tiende a cero cuando la solución se aproxima a la solución óptima objetivo de la RS).

3.2 Hill Climbing

Hill Climbing (HC) es una técnica tradicional de optimización heurística que pertenece a la familia de los métodos búsqueda locales sin uso de derivadas. La relativa simplicidad del algoritmo hace que sea popular para la elección entre los algoritmos de optimización. Es por esto que, a pesar de ser un método de búsqueda local y optimización heurística, se lo aplica con éxito para encontrar y evolucionar constates numéricas en la PG.

3.3 Simulated Annealing

Simulated Annealing (SA) es una evolución del método anterior. A diferencia de éste, SA es una heurística de búsqueda global ya que posee mecanismos que le permiten realizar la búsqueda más allá del óptimo local más cercano.

La aceptación de malas soluciones es una propiedad fundamental de este tipo de heurísticas ya que permite salir de la zona de óptimos locales. Es decir, en el caso particular de SA para manejo de constantes en Tree-GP, se adapta el concepto de temperatura y enfriamiento usual de SA, para relacionarlo a la cantidad de generaciones faltantes hasta el final de la corrida y la calidad del individuo encontrado. Así, si el fitness de un individuo es bajo, entonces el factor de temperatura (T_o) que controla esta propiedad se incrementa, de forma de abarcar un rango más amplio. Por el contrario, si el fitness del mejor individuo de la generación es mayor, entonces el factor de enfriamiento se reduce, provocando cambios más pequeños. Es así como se toma este concepto para aplicarlo a la generación y aceptación de nuevas constantes en un individuo.

3.4 Multi-Dimensional Hill Climbing

Este método recibe su nombre porque utiliza Hill Climbing en cada una de las constantes, en cada dimensión [7]. Multi-Dimensional Hill Climbing o (MDHC) consiste en elegir un número Δ al azar entre 0 y el factor de temperatura actual, para cada una de las constantes numéricas que se desean modificar. Luego de obtenido Δ, se consideran k nuevos números por cada una de las constantes que se tienen: $c - \Delta$ y $c + \Delta$. Posteriormente se evalúa el árbol correspondiente al individuo en cuestión reemplazando las constantes por los nuevos valores. Se deben probar todas las combinaciones posibles, de las cuales la combinación que produjo un individuo con mejor fitness será la que se conserve.

La principal ventaja de este método es que asegura que la solución obtenida será igual o mejor que la actual en cuanto a Fitness, ya que se evalúan todas las posibles alternativas conservando la mejor.

Sin embargo la implementación de este método requiere mucho procesamiento ya que la cantidad de evaluaciones de la función fitness se incrementa exponencialmente con la cantidad de constantes involucradas. La cantidad de evaluaciones adicionales requerida en cada generación es:

$$n = 3^{Cant_Constantes} * Cant_Individuos$$

Se debe considerar que la evaluación suele ser la operación que consume más procesamiento en un algoritmo genético, por lo tanto los tiempos de ejecución pueden incrementarse considerablemente.

3.5 Differential Evolution

Differential Evolution (DE) es un método que optimiza un problema iterativamente tratando de mejorar una solución candidata con respecto a un determinado fitness [8]. Una de las variantes más sencillas del algoritmo DE trabaja en base a una población de soluciones candidatas, llamados agentes. Estos agentes se mueven alrededor en el espacio de búsqueda mediante la elección arbitraria de tres individuos distintos al azar de la población de constantes X_1, X_2, X_3 y un lugar de asignación i. Entonces:

$$X_i = X_1 + F * (X_2 - X_3)$$

dónde F es básicamente un factor de escala.

Algunas variantes de DE se diferencian en la cantidad de puntos considerados, otras toman X_1 (el vector de constantes numéricas) del mejor o los mejores individuos de la población.

DE es un método sencillo y rápido, pero presenta una pobre estabilidad a repeticiones debido a que tiene una componente aleatoria muy fuerte.

4. Pruebas

En esta sección se presentan las pruebas realizadas con los algoritmos implementados.

4.1 Testeo de métodos

La función de *Rastriguin* es una función diferenciable y no convexa, multimodal, comúnmente utilizada como benchmark para probar la eficiencia y rendimiento de los algoritmos de optimización. Como el problema planteado radica en la obtención de constantes dentro de un problema de RS y no en la optimización global de un problema, se usó esta función con sus constantes modificadas y definida para una sola variable independiente de manera que permita probar la eficacia y performance del sistema en casos difíciles. Teniendo en cuenta esto, la ecuación final para ser evaluada es:

$$f(x) = 0.95x^2 - 5.5\cos(\pi x)$$

Los parámetros utilizados para las pruebas, dependiendo de cada método, fueron:

- Generaciones: 500
- Tamaño Población: 500
- Cruzamiento: 80%
- Mutación: 10%
- Elite: 5%
- Set de Funciones: +, -, *, /, *sen*, *cos*
- Porcentaje de población afectada a la optimización: 80%
- TC: 0.002
- Iteraciones (HC, SA): 15
- Tamaño del paso Máximo (HC, SA): 1
- Temperatura inicial (SA): 2
- Constante de enfriamiento (SA): 0.05
- Rango de definición: [-5, 5]
- Generaciones Máximas: 10
- Altura máxima del árbol: 6

4.1.1 Numeric Mutation

Utilizando los parámetros de entrada descritos anteriormente, la mejor solución obtenida con este método se muestra en la figura 2. Luego de un post-procesamiento, la ecuación final corresponde a:

$$f(x) = \cos(3.13881x) * 0.950526x^2 - 0.422558\cos(3.13993\,x)$$

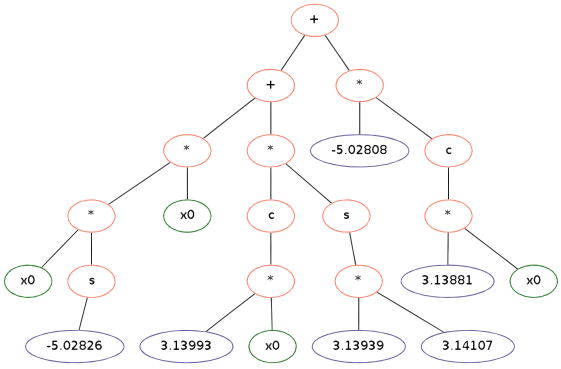


Figura 2: Mejor individuo obtenido con NM.

El fitness que obtuvo el mejor individuo de esta prueba fue igual a -0.0649662 en un tiempo total de $146{,}21$ segundos. En la figura 3 se puede ver la curva correspondiente al mejor individuo de salida comparado con la función original.

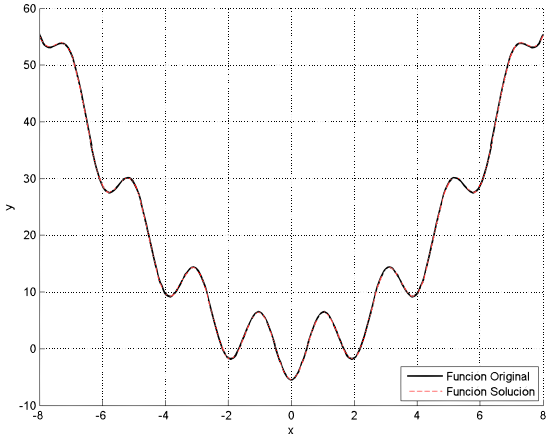


Figura 3: Función de salida vs función original para NM.

4.1.2 Hill Climbing

Con este método se obtuvo un individuo final correspondiente al de la figura 4, y una ecuación simplificada equivalente a:

$$f(x) = -0.3432x^2(\cos(3.2277\,x) - 3.4554)\cos(0.8047\cos(1.4139x))$$

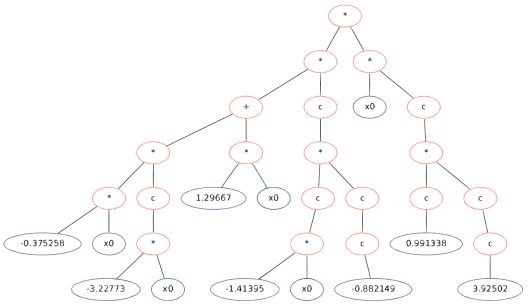


Figura 4: Mejor individuo obtenido con HC.

La ejecución se llevó a cabo en 8337.71 segundos y se obtuvo un fitness igual a -2.28326. La solución obtenida y el desempeño de éste método para el ejemplo planteado se puede ver en la figura 5. Si bien se observa que el algoritmo sigue la tendencia general de la función, los resultados no son tan buenos como en los otros casos.

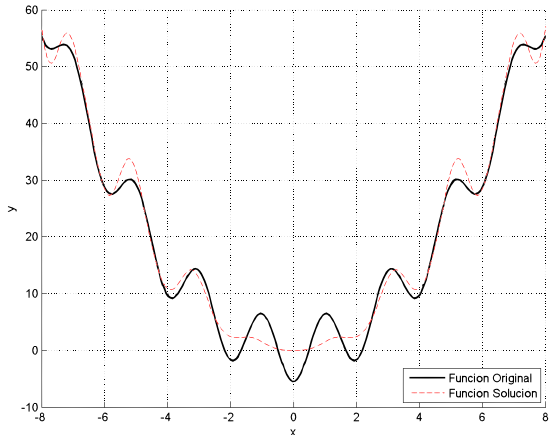


Figura 5: Función de salida vs función original para HC.

4.1.3 Simulated Annealing

Con este método se obtuvo el individuo representado en la figura 6 y como salida la ecuación simplificada:

$$f(x) = x^2 - sen(-x + sen(x) + 0.5309) - 5.0079\cos(3.1382x) - 0.9909$$

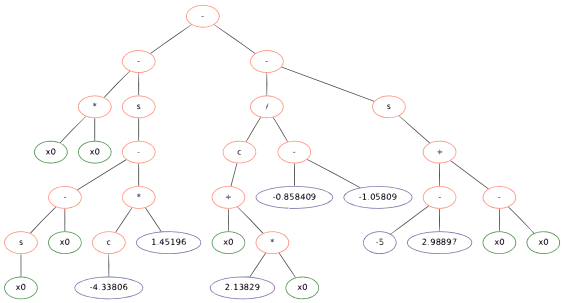


Figura 6: Mejor individuo obtenido con SA.

Diferente al caso anterior, este método pudo aproximar en su mayoría los datos de entrada, logrando un fitness de -1.09076 en 11150.8 segundos, como se puede ver en la figura 7.

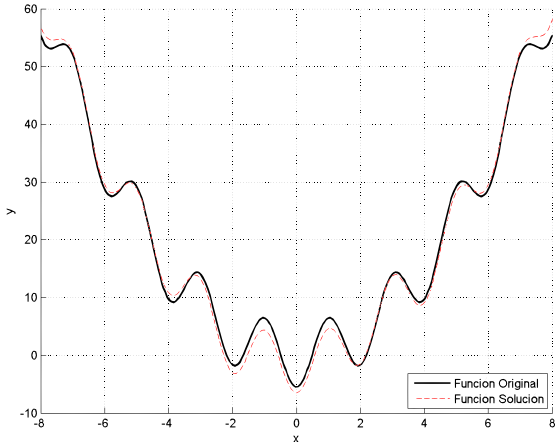


Figura 7: Función de salida vs función original para SA.

4.1.4 Differential Evolution

Con este método se obtuvo el individuo de la figura 8 y como salida la ecuación simplificada:

$$f(x) = -sen(1.5803 - 3x) + x^2 - sen(3x + 8.2301)$$

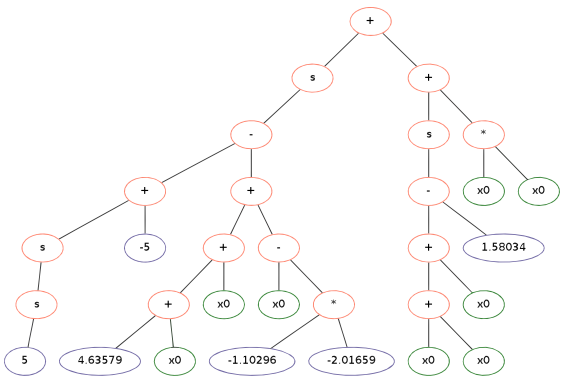


Figura 8: Mejor individuo obtenido con DE.

A partir de la figura 9 y el valor de fitness obtenido se concluye que es necesaria una segunda corrida para obtener mejores resultados y reutilizar el resultado previo encontrado por el sistema, al igual que en el caso de HC.

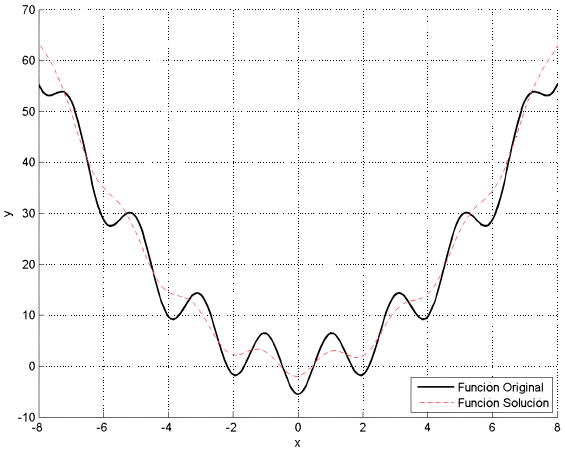


Figura 9: Función de salida vs función original para DE.

4.1.5 Multi Dimensional Hill Climbing

Con este método se obtuvo el individuo de la figura 10 y como salida la ecuación simplificada:

$$f(x) = 0.9519x^2 - 5.5048\cos(3.1391x)$$

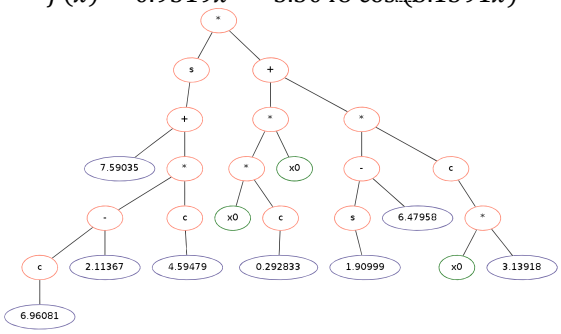


Figura 10: Mejor individuo obtenido con MDHC.

El individuo consiguió un valor de fitness de -0.0778178 en 1506.56 segundos como se puede observar en la figura 11.

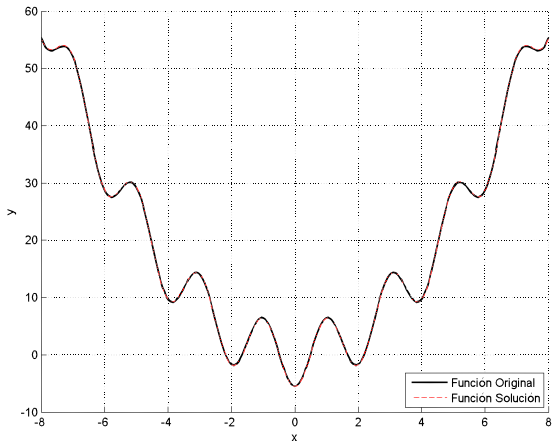


Figura 11: Función de salida vs función original para MDHC.

4.2 McCormick Function

McCormick es otra función utilizada como benchmark para problemas de optimización [9]. La misma está definida por la siguiente ecuación y representada por la gráfica de la figura 12.

$$f(x,y) = sen\,(x+y) + (x-y)^2 - 1.5x + 2.5y + 1$$

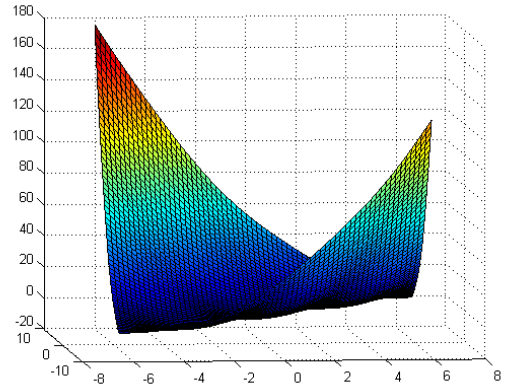


Figura 12: Gráfica de McCormick Function

Esta función de benchmarking fue probada con el algoritmo MDHC, el cual demostró tener una performance superior en comparación con los otros métodos ante problemas más avanzados. Los parámetros de ejecución del sistema fueron:

- Generaciones: 200
- Tamaño de la población: 150
- Cruzamiento: 80%
- Mutación: 30%
- Elite: 5%
- TC: 1.6
- Set de Funciones: +, -, *, /, *sen*, *cos*

El individuo de salida corresponde al árbol de la figura 13 y su ecuación simplificada es:

$$f(x) = x^2 - 2xy - 1.3323x + y^2 + 2.6209y + 1.0887$$

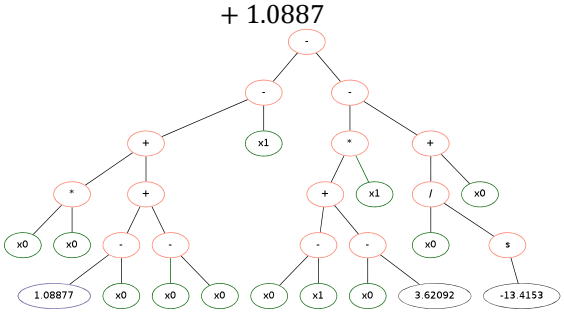


Figura 13: Mejor individuo obtenido.

Este individuo consiguió un fitness igual a -1.40393 en 7980.15 segundos de ejecución. El algoritmo encontró fácilmente las componentes principales debido a que sus valores numéricos son muy significativos comparado con los otros términos. Por este motivo no fueron encontradas las componentes más pequeñas. Para solucionar esto se obtuvo un residuo, calculando la diferencia entre los resultados obtenidos por el algoritmo en la primera corrida con los datos de prueba. Luego se hizo una segunda corrida con el residuo calculado, utilizando los mismos parámetros y llegando a los siguientes resultados:

$$f(x) = 0.3999sen(x+y) - 0.1999x - 0.1999y - 0.1998$$

Lo que nos lleva a la ecuación final del sistema:

$$f(x) = x^2 - 2xy + 0.3399sen(x+y) - 1.5322x + y^2 + 2.4209y + 0.8889$$

Esta ecuación posee todos los términos de la función original. La diferencia entre las constantes encontradas es menor a 0.15, excepto por el coeficiente del la función seno que está un poco más alejado (0.66). Esto se debe a que en el intervalo considerado, la función seno tiene un valor máximo en valor absoluto de 1, que es 2 órdenes de magnitud menor que los de mayor grado. Para mejorar este resultado se requieren métodos de ajuste local como regresión lineal, o métodos con derivada si se sospecha que hay constantes dentro de la función seno (en lugar de sólo $x + y$). Es importante destacar que incluso en la primera fase descripta, se obtuvo un fitness de -1.4, lo cual indica que la función está próxima en valor absoluto a la original. El error obtenido en el individuo final se puede ver en la figura 14.

Para este caso de prueba se requirió una segunda corrida del sistema teniendo en cuenta las componentes no encontradas en la primera solución para poder llegar a un resultado razonablemente bueno.

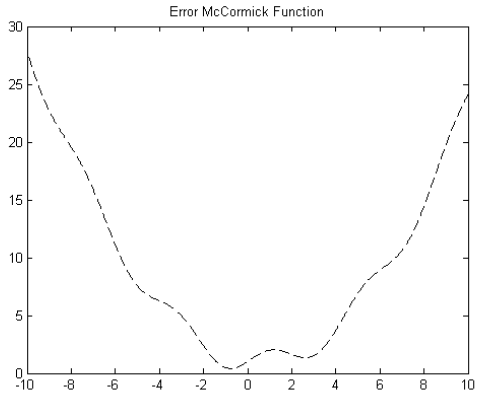


Figura 14: Gráfica del error obtenido en la solución final.

4.3 Pasajeros de Aerolínea Internacional

Esta es una prueba con datos reales publicado para su reproducción y análisis [10]. El caso consiste en una serie de tiempo, en donde se involucran la cantidad de pasajeros de una aerolínea internacional (medidos en cientos de pasajeros) y la fecha de viaje, tomando como referencia el periodo de Enero de 1949 a Diciembre de 1960.

La prueba se realizó con Multi-Dimensional Hill Climbing utilizando los siguientes parámetros:

- Población: 180
- Generaciones: 450
- Cruzamiento: 65%
- Mutación: 10%
- Elite: 5%
- TC: 1.
- Set de Funciones: +, -, *, /, *sen*, *cos*, *pow*, *sqrt.*

En la figura 15 se pueden observar los resultados obtenidos y comparar la semejanza entre la serie de tiempo original (línea llena) y la obtenida con el framework (línea punteada).

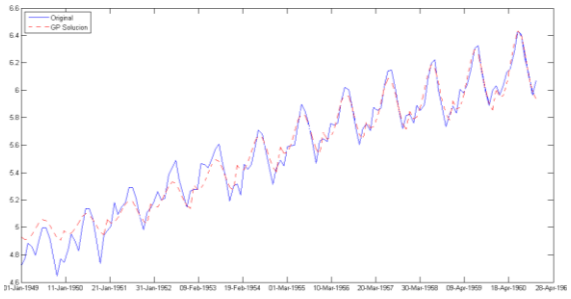


Figura 15: Función de salida vs función original.

A sí mismo, en la figura 16 se puede observar el árbol de salida con sus constantes correspondientes. Al recorrer el árbol en preorden se puede obtener la función que define a esta serie de tiempo.
Esta reproducción obtuvo un fitness igual a -19.9834 en 250.48 segundos, resultados en los cuales se ve reflejada la complejidad del problema. El análisis realizado en el trabajo original muestra que la serie tiene componentes autorregresivos (se debe incorporar datos de fechas anteriores como variables de entrada), lo cual mejoraría los resultados. Sin embargo se nota el potencial de la herramienta consiguiendo el resultado indicado.

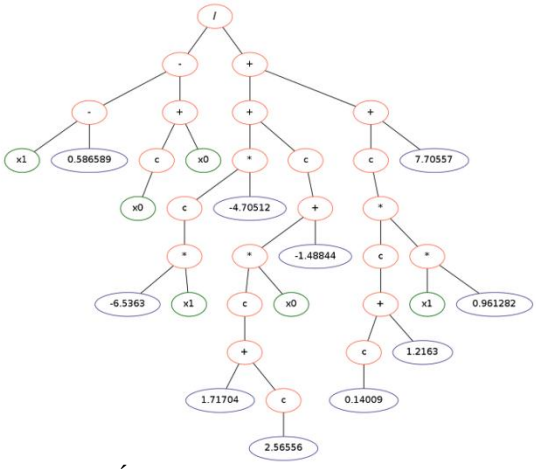


Figura 16: Árbol del mejor individuo obtenido.

5. Resultados y Discusión

En las pruebas realizadas a funciones de benchmarking el sistema responde correctamente a los datos de entrada. En la mayoría de los casos se obtuvieron las componentes de mayor variación o de mayor impacto sin mayores inconvenientes, sin embargo las componentes pequeñas (residuo) no siempre fueron encontradas.
Como se puede ver en los resultados obtenidos en el testeo de los métodos, con NM, MDHC y SA las soluciones encontradas son bastante buenas y aproximadas a los datos de entrada, pero existe una diferencia significativa con los tiempos de ejecución y los requerimientos de procesador y memoria. Al tener que realizar tantas evaluaciones en los métodos generacionales (SA, HC y MDHC) el tiempo necesario para llegar a una buena solución se eleva demasiado y resulta poco eficiente y práctico la utilización de estos métodos para este tipo de problemas. Cabe destacar que NM es mucho más simple y eficiente que los otros cuatro métodos en la mayoría de los casos simples, pero no así para los casos más difíciles.
El Framework utilizado demostró tener la robustez necesaria para la aplicación de problemas de Regresión Simbólica de distintas características obteniendo soluciones de buena calidad, rápida convergencia y tiempos acotados.
Esta implementación abre las puertas a un gran conjunto de problemas que en la actualidad no tienen solución determinística, y en donde dicha solución puede representar una gran mejora en campos de aplicación de cualquier área cuyo funcionamiento sea en base a conjuntos de datos.
Resulta importante destacar que existen otras heurísticas como Redes Neuronales que son capaces de reproducir eficazmente la forma de una función, pero no son susceptibles de análisis posteriores. Las soluciones proporcionadas por PG son fórmulas matemáticas y como tales susceptibles de un análisis teórico posterior, como análisis de sensibilidad y de propagación de errores, e incluso dar lugar a la generación de mejores modelos teóricos sobre la base de las fórmulas encontradas.

6. Conclusiones

De lo analizado y probado se concluye que la herramienta sólo está destinada a ser una ayuda y una aproximación a una solución analítica, siendo necesario en todos los casos un procesamiento matemático posterior y un análisis por parte de expertos para conseguir resultados utilizables.

Dicho esto, resulta importante destacar que el problema demostró ser muy complejo ya que existen distintos métodos de resolución para los distintos problemas que se pueden presentar.

Las aplicaciones para el sistema desarrollado superaron las expectativas tanto en tiempo como en performance. En las pruebas efectuadas probó ser estable a repeticiones en la mayoría de los casos, encontrando una solución muy aproximada en cada problema planteado. Esta es una de las características más difíciles de lograr debido a la aleatoriedad intrínseca de los algoritmos genéticos de los que se basa la RS dentro de un sistema de PG. Específicamente se puede resaltar los siguientes puntos:

- Se lograron excelentes resultados en los casos de benchmark y en los problemas comparados de la bibliografía. Los tiempos de ejecución son estables y proporcionales a la complejidad del problema.
- En los casos más complejos, combinando más de 6 términos, con potencias y otros tipos de funciones combinadas, se observa una disminución en la tasa de éxitos.
- El programa encuentra la respuesta correcta en 1 de cada 5 veces aproximadamente, y en varios casos es necesario realizar el proceso en varias etapas, haciendo un análisis por residuos para encontrar un resultado correcto.
- Los métodos anteriormente mencionados en al comienzo de la sección 2 son sumamente básicos para un problema tan complejo, y fueron descartados en etapas preliminares del sistema.
- Se pudo concluir que el método de NM es sumamente rápido y eficiente para casos relativamente simples, pero no asegura buenos resultados en todos los casos.
- DE es eficiente en procesamiento pero tampoco asegura resultados, siendo sobrepasado en eficiencia y resultados por NM.
- Los métodos de SA y HC proporcionan buenos resultados pero requieren mucho más procesamiento y ajustes experimentales que el resto de los métodos.
- El método MDHC es sumamente costoso para grandes cantidades de constantes, pero aplicarlo con un valor de n relativamente bajo (como n = 3 o 4) asegura que esas constantes tomarán el mejor valor posible, lo que mejora los resultados sin aumentar exageradamente la cantidad de procesamiento.
- En la codificación actual del sistema se aplica un método por vez. A futuro claramente es deseable aplicar más de un método en forma simultánea de forma de aprovechar las ventajas de cada uno.
- Para los casos de prueba se tuvieron en cuenta un grupo mínimo de constantes numéricas de entrada, por lo general en un rango [-5,5], sólo para que el sistema sea capaz de introducir en las primeras generaciones estas constantes para que luego sean optimizadas por los algoritmos. Todas las pruebas se corrieron en una máquina virtual bajo un entorno de Linux con un solo procesador Core i5 de 2.67 Mhz y 3GB de memoria RAM.
- La herramienta desarrollada y la PG a nuestro juicio tiene como objetivo principal proponer un modelo que debe ser analizado y mejorado con herramientas de búsqueda local para mejorar las constantes conociendo la estructura de la solución, desde regresión lineal, métodos con derivadas, hasta métodos estadísticos de selección de parámetros o específicos para series de Fourier, que permitan obtener los valores correctos dado que se conoce la estructura de la solución.

7. Trabajos Futuros

Como parte del trabajo futuro correspondiente a la investigación presentada, se tienen en cuenta las siguientes observaciones:

- Resulta imprescindible mejorar la definición de convergencia correcta. En algunos casos se obtienen fitness excelentes de 0.01 o menores, pero sin embargo en el gráfico correspondiente se observa que el ajuste no es perfecto. Sin embargo, esto no se considera un problema en ciertos casos en donde el objetivo es solamente detectar tendencias generales en los datos (como datos económicos o bursátiles).
- Mejoras en la metodología para analizar los residuos de las soluciones obtenidas en los casos en que el fitness no se acerca a los valores deseados, siendo necesario correr nuevamente el algoritmo sobre la diferencia de los datos con los primeros resultados (residuo).
- Implementar mejoras para encontrar soluciones más simples a un problema. Esto puede lograrse con un sistema para eliminación de intrones o simplificación de fórmulas.

Referencias

[1] Poli, R., Langdon, W. B., McPhee, N. F., y Koza, J. R., "A field guide to genetic programming", *Lulu .com*, 2008. URL: http://books.google.com.ar/books?id=3PBrqNK5fFQC, accedido el 24/08/2014

[2] Goldberg, D. y Richardson, J., "Genetic algorithms with sharing for multi-modal function optimization", en *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 44 - 49. ICGA, 1987.

[3] Borenstein, Y. y Poli, R., "Empirical approach for theory of randomized search heuristics", 2006.

[4] Xu, X. y Gen, M., "Introduction to Evolutionary Algorithms. Decision Engineering", *Springer*, ISBN 9781849961288, 2010. URL: http://books.google.com.ar/books?id=rHQf_2Dx2ucC.

[5] Santochi, D., Lenis J., "Plataforma de Programación Genética en C++". *Tesis de grado de la carrera de Ingeniería en Computación de la Universidad Nacional de Tucumán*, 2011.[1]

[6] Evett, M. y Fernandez, T., "Numeric mutation: Improved search in genetic programming", en *Diane J. Cook, ed., FLAIRS Conference*, pp. 106-109, AAAI Press, ISBN 1-57735-051-0, 1998. URL: http://dblp.uni-trier.de/db/conf/flairs/flairs1998.html#EvettF98

[7] Fernandez, T., "Evolution of numeric constants in genetic programming", 1997. URL: http://www.cse.fau.edu/~thomas/AI/GP_and_Numeric_Constants.PDF, accedido el 24/08/2014.

[8] Rainer, S. y Kenneth, P., "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces". *Journal of Global Optimization 11*: pp. 341–359, Kluwer Academic Publishers, 1997.

[9] Hyndman, R. J., "Time series data library", 2005. URL: http://data.is/TSDLdemo, accedido el 24/08/2014.

[10] Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M., y Yang, Z., "Benchmark functions for the cec2008 special session and competition on large scale global optimization", 2008. URL: http://nical.ustc.edu.cn/cec08ss.php

[11] Navarro, F., Remis, L., Santochi, D., Lenis, J., Will, A., y Rodriguez, S., "Programación Genética y aplicaciones a Robótica: Cortadora de Pasto y Resolución de Laberintos," en *Actas de las VII Jornadas Argentinas de Robótica, Olavarria*, 2012.

[1] **El informe y el código están disponibles sobre pedido en la biblioteca de la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán.**

Diseño eficiente de redes de radio frecuencia con algoritmos CHC en comunicaciones inalámbricas

D. Molina [1], *D. Pandolfi* [1], *A. Villagra* [1], *G. Leguizamón* [2]

[1] *Universidad Nacional de la Patagonia Austral, Argentina*
[2] *Universidad Nacional de San Luis, Argentina*

Resumen

La popularización de los servicios inalámbricos de comunicación (telefonía, internet, etc.) es cada vez mayor en la actualidad. La demanda hace que los usarios exijan la disponibilidad de estos servicios en todo momento y en cualquier lugar. El costo de los equipamientos para brindar el servicio con la calidad adecuada es elevado. En consecuencia, es de vital importancia seleccionar los emplazamientos geográficos que permitan lograr una cobertura óptima de señal minimizando el uso de equipamientos. Esta tarea se la denomina diseño de la red de radio (RND) y es un problema NP-duro de optimización, por lo tanto, es factible de ser tratado con metaheurísticas. Las metaheurísticas son métodos que integran procedimientos de mejora local y estrategias de alto nivel para realizar una búsqueda robusta en el espacio del problema. En el presente trabajo se proponen diferentes versiones de CHC (Crossover elitism population, Half uniform crossover combination, Cataclysm mutation), diferenciadas a través del mecanismo de selección poblacional y una función objetivo basada en la eficiencia del uso de recursos. Se comparan los algoritmos propuestos con algoritmos genéticos (AGs). Los resultados obtenidos son muy satisfactorios en cuanto a eficiencia y calidad.

Palabras clave servicios inalámbricos, red de radio, metaheurísticas, algoritmo CHC

1. Introducción

El uso de medios inalámbricos basados en la propagación de las ondas de radio en los sistemas de comunicaciones es una forma de realizar una difusión masiva de cualquier servicio que pueda ser transmitido por este medio. Lugares como las montañas, las rutas, sitios rurales o las ciudades no podrían disfrutar de la magia de la telefonía móvil o sería casi imposible para nuestro proveedor de internet, brindarnos el servicio sin la existencia de este medio de comunicación. Según [3] el problema del posicionamiento de antenas puede ser descripto como: dado un conjunto de sitios candidatos, un área geográfica discretizada y un conjunto de puntos que deben intercomunicarse entre si, se debe seleccionar un subconjunto de sitios, del conjunto de sitios candidatos, que maximicen la cobertura empleando el mínimo de recursos, cumpliendo con la estimación de tráfico y el umbral de recepción de señal entre los puntos. Las ondas electromagnéticas trasladan el mensaje entre el emisor y el receptor generando un intercambio de mensajes. Las ondas al desplazarse por el espacio libre están sujetas a ruidos (atenuación, reflexión, refracción y difracción) provocados por fenómenos atmosféricos y obstáculos [13]. Como consecuencia existen pérdidas en el nivel de la señal y el receptor podría no ser capaz de decodificar el mensaje. Por lo tanto, lograr el posicionamiento de las antenas de manera tal que la mayoría de los receptores puedan tener una línea de vista sin obstáculos, asegurando un nivel de señal la mayor parte del tiempo es un gran desafío. La tarea responsable de resolver este problema según la forma tradicional se la denomina RND [7]. La complejidad del problema motivó el uso de las metaheurísticas, podemos ver a [1] como uno de los primeros antecedentes en el tema con el uso de Recocido Simulado (*Simulated Annealing*, SA). Las metaheurísticas son un enfoque viable para manejar de manera práctica el problema de seleccionar sitios para instalar radio bases en un entorno urbano. También podemos ver que [5] presenta un algoritmo denominado algoritmo adaptativo para el posicionamiento de radio bases (ABPA) que también utiliza SA para el análisis de las soluciones y un modelo de línea de vista (LOS) para determinar el nivel de cobertura de la señal. En [14] se realiza una revisión general de diferentes metaheurísticas resolviendo el problema de RND. El estudio pretende ofrecer una base de referencia confiable sobre un amplio espectro de algoritmos y medidas precisas de comparación de la eficiencia, confiabilidad y rapidez de las diferentes técnicas aplicadas a la resolución del RND. Numerosos AGs han sido utilizados con implementaciones paralelas o multiobjetivo. Un ejemplo es [16] donde se modela el problema en su máxima complejidad práctica,

basado en una optimización combinatoria con restricciones multiobjetivo con un enfoque Pareto para encontrar el conjunto de redes no dominantes y diversificadas. En [3] se plantea el uso de una heurística para resolver el problema de RND, formada por tres fases secuenciales. La primera fase ejecuta el preprocesamiento de las restricciones del problema, eliminando de esta forma un gran número de "malas" combinaciones. La segunda ejecuta una optimización utilizando Búsqueda Tabú (*Taboo Search*, TS) sobre un espacio de búsqueda reducido. La tercera es una post optimización ejecutando la sintonía fina de los parámetros de las antenas. Basados en el enfoque tradicional de diseño de una red de radio [7] se propone analizar y evaluar metaheurísticas para encontrar una solución aceptable al problema de RND. Se implementan AGs con operadores de recombinación uniforme, de uno y dos puntos corte. Tomando como base el algoritmo CHC canónico [4] se generan distintas versiones del algoritmo CHC y se realiza una comparación con los AGs. Cada versión utiliza un mecanismo de selección poblacional distinto y la variabilidad poblacional como un mecanismo para la prevención del incesto y la detección de la convergencia prematura. Además una variante aportada en este trabajo es el uso de una función objetivo distinta a la presentada por [12], basada en la proporcionalidad de la cobertura de la superficie con respecto a la señal irradiada que minimiza las interferencias por superposición de lóbulos de radiación y el uso de radio bases como recursos de elevado costo. El presente trabajo se organiza de la siguiente manera: en la Sección 2 se presentan los elementos básicos y la metodología utilizada, dando una explicación sobre los AGs, el algoritmo CHC canónico, los mecanismos de selección poblacional para cada una de las variantes propuestas, la función objetivo y la forma de modelizar el problena con un terreno discretizado. En la Sección 3 se presentan los resultados obtenidos en el experimento con las distintas instancias propuestas. En la Sección 4 se realiza una discusión sobre los resultados obtenidos y se proponen posibles aplicaciones en la vida real. En la Sección 5 se emiten las conclusiones y los trabajos futuros.

2. Elementos del trabajo y metodología

En la presente sección se desarrollan los conceptos relacionados con los AGs implementados, los métodos de selección poblacional para cada una de las versiones de CHC propuestas , la forma de modelizar el terreno y los diferentes escenarios de prueba generados junto con la función objetivo propuesta.

2.1. Algoritmos genéticos

Los AGs son algoritmos de búsqueda que imitan el comportamiento de la selección natural. Buscan la mejor solución de un determinado problema (e.g. maximizar una función) mediante la generación de un conjunto (población) de soluciones potenciales (individuos) para el problema y a través de operaciones de mutación y recombinación se espera que surjan mejores soluciones a partir del conjunto inicial. Este proceso continua hasta que se encuentra una solución lo suficientemente aceptable para el problema [6]. Los AGs codifican las variables de decisión de un problema de búsqueda dentro de un alfabeto finito de una cadena de caracteres con una cierta cardinalidad. La cadena que representa una solución candidata para el espacio de búsqueda del problema es el cromosoma, el alfabeto son los genes y el valor de los genes son los alelos. Para evolucionar las buenas soluciones e implementar la selección natural, se necesita una medida para distinguir "buenas" soluciones de las "malas" soluciones. La medida es la función objetivo o fitness, que puede ser representada mediante un modelo matemático, una simulación de computadora o podría ser una función que permita seleccionar las mejores soluciones de otras peores. En esencia, la función objetivo establece una medida que debe determinar la calidad de la solución candidata, que luego será utilizada por el AG como guía en su evolución a mejores soluciones.

Algoritmo 1 Pseudocódigo de un AG

```
1:  /* Parámetros de un AG */
2:  t = 0 //número de generación
3:  P(0)
4:  evaluar P(0)
5:  while not (Condición de terminación) do
6:     P'(t)=seleccionar(P(t))
7:     P'(t) =aplicar operadores de cruce(P(t))
8:     P(t+1) = reemplazar (P(t),P'(t))
9:     evaluar P(t+1)
10:    t=t+1
11:    return Mejor Solución Encontrada
12: end while
```

En el Algoritmo 1 se presenta el pseudocódigo de un AG canónico. Los operadores de cruce utilizados en este trabajo son:

- Cruzamientos de un punto: el operador de cruzamiento de un punto selecciona un punto de cruzamiento de manera aleatoria sobre la longitud total de la cadena y los alelos uno a uno son intercambiados entre los individuo (AG-OPX).

- Cruzamiento de dos puntos: selecciona dos puntos de cruce en forma aleatoria. Los alelos entre los dos puntos de cruzamiento son intercambiados entre los dos individuos apareados (AG-TPX).

- Cruzamiento Uniforme: en este operador de cruzamiento cada alelo es intercambiado de a pares mediante una selección basada en probabilidad *pe*, conocida como probabilidad de intercambio (AG-UX).

2.2. Algoritmos CHC

El algoritmo CHC combina una estrategia de selección conservativa que preserva los mejores individuos encontrados [4], [14]. El operador de recombinación genera descendientes que maximizan sus diferencias genéticas con respecto a sus padres. La reproducción solo se produce si la distancia de Hamming (diferencia genética entre los padres) es mayor al umbral de incesto ($1/4$ del tamaño del cromosoma). El operador de cruzamiento HUX (*half uniform crossover*) es usado para maximizar la distancia genética entre individuos. La nueva población se genera con una selección de los mejores individuos entre padres e hijos. Por cada generación sin mejoras se disminuye la distancia, cuando la cantidad de bits distintos entre los padres seleccionados es menor al umbral, significa que los individuos son demasiado similares. Esto determina que la población ha convergido, indicando un estancamiento del algoritmo, por lo que se produce un reinicio de la población mediante un cataclismo [2]. El Algoritmo 2 muestra el pseudocódigo de CHC canónico.

Para el problema de RND con cadenas de bits extensas (ej. 200 bits) el umbral de incesto para el reinicio de la población demanda un gran esfuerzo computacional en iteraciones y tiempo. El método propuesto utiliza la variabilidad genética de la población para medir el nivel de diversidad de los individuos. Toma como base la probabilidad de aceptación del SA [8], pero de forma inversa, es decir, la temperatura aumenta con el paso del tiempo, incrementado la probabilidad de un cataclismo. Por cada iteración sin mejoras se compara la variabilidad de la población generada con respecto a la variabilidad patrón. La diferencia entre estas dos debe ser siempre menor al entorno con centro en la variabilidad patrón y de radio igual a la probabilidad de aceptación. A medida que las iteraciones pasan el entorno se reduce y si la variabilidad de la población generada queda fuera del entorno, se produce el cataclismo. La Ecuación 1 define la variabilidad como:

Algoritmo 2 Pseudocódigo CHC

```
1:  t = 0 //generación de tiempo
2:  Inicializar (Pa, umbralIncesto)
3:  while not (Condición de terminación(t,Pa)) do
4:     padres=seleccionarPadres(Pa,umbralIncesto)
5:     hijos=HUX(P(t))
6:     evaluar(padres,hijos)
7:     Pn = SelecciónPoblación(padres,hijos)
8:     if not (modificación (Pa,Pn))
9:        umbralIncesto=umbralIncesto-1
10:        if umbralIncesto <= k
11:           Pn = reinicio(Pa)
12:        end if
13:    endif
14:    t = t+1
15:    Pa = Pn
16: end while
17:  return Mejor Solución Encontrada
```

$$v = \frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{c} x_{ij}\right)}{n \times c} \qquad (1)$$

$$x \in (0, 1)$$

donde v es la variabilidad de la población, c el tamaño del cromosoma y n la cantidad de individuos de la población. La Ecuación 2 define el entorno de aceptación (ea) como:

$$ea = 1 - e^{\left(\frac{(f(x')-1)}{T}\right)} \qquad (2)$$

donde $f(x')$ es el valor del óptimo local y T la cantidad de iteraciones sin mejoras.

El algoritmo CHC propuesto queda definido según el Algoritmo 3 en el que se modifican las líneas 9 a 13 del Algoritmo 2 incorporando la prevención del estancamiento y la forma de detección del incesto. Por ser solamente estas las diferencias con el CHC canónico el algoritmo propuesto se lo denominó QCHC (Quasi CHC). Las nuevas versiones utilizan éste algoritmo y modifican el método de selección poblacional.

Los operadores de selección poblacional propuestos para este trabajo son: (1) Reemplazo Elitista (RE), todas las parejas seleccionadas generan hijos y la nueva población se compone de los mejores individuos entre padres e hijos [4]; (2) Torneo Elitista (TE), todas las parejas seleccionadas generan descendientes y todos los individuos compiten en un torneo sin reposición (de cada pareja seleccionada el individuo con mejor valor objetivo pasa a la nueva población [15]); (3) μ + λ (ML, por Mu y Lambda), el mejor individuo generado reemplaza al peor individuo de la población actual, conformando de esa manera la nueva población.

Para identificar el método de selección poblacional en cada versión propuesta se invoca el método correspondiente en el parámetro "método" de la línea 7 en el Algoritmo 3. Cada nueva versión se denomina de la siguiente manera: QCHC-RE (implementa al método 1), QCHC-TE (implementa al método 2) y QCHC-ML (implementa al método 3).

Algoritmo 3 Pseudocódigo QCHC

```
1:  t = 0 //generación de tiempo
2:  Inicializar (Pa, umbralIncesto)
3:  while not (Condición de terminación(t,Pa)) do
4:     padres=seleccionarPadres(Pa,umbralIncesto)
5:     hijos=HUX(P(t))
6:     evaluar(padres,hijos)
7:     Pn = SelecciónPoblación(padres, hijos,método)
8:     if not (modificación (Pa,Pn))
9:     varPatron =0.5 // variabilidad patrón
10:       VariabilidadPoblaciónEstancada(Pa)
11:       RadioAceptacion(T,valorObjetivoSinMejora)
12:         if variabilidad está fuera de rango
13:           Cataclismo(Pa)
14:         else
15:           T = T+1 // se incrementa T disminuyendo el rango de aceptación
16:       end if
17:    t = t+1
18:    Pa = Pn
19: end while
20:  return Mejor Solución Encontrada
```

A continuación se describe el mecanismo utilizado para discretizar el terreno.

2.3. Discretización del terreno

Para discretizar la superficie del terreno se utiliza una matriz M de $f \times c$, relacionando la latitud y la longitud con los subíndices i, j de las celdas de la matriz. Luego la altura de cada punto a nivel del mar se almacena en cada celda de la matriz. Esta forma de discretización responde al modelo *raster* utilizado en los sistemas de información geográfica (SIG) y ha sido usado en [9], [12] para resolver el problema de RND. Los sitios candidatos para la instalación de las antenas se representan como un subconjunto de celdas pertenecientes a la matriz M. Para este problema se utilizan antenas omnidireccionales y se considera el terreno como plano. La Figura 1 muestra la cobertura lograda por una posible red de radio. Si la celda es cubierta por una sola antena se coloca un 1, si es cubierta por más de una celda se produce una interferencia y se coloca un -1, y si no tiene cobertura se coloca un 0. Cada posible red de radio se modela mediante la activación o no de cada uno de los emplazamientos candidatos. El espacio de búsqueda del problema queda determinado por 2^n siendo n la cantidad de sitios candidatos.

Seguidamente se presenta el mecanismo utilizado para calcular la calidad de las soluciones.

1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figura 1. Terreno discretizado con una matriz con valores de cobertura de una red de radio candidata

2.4. La función objetivo

La función objetivo propuesta utiliza los valores de la matriz de terreno M y asigna un único valor numérico a cada solución como medición de la calidad de la red de radio evaluada. Debe guiar a los diferentes algoritmos en la búsqueda de la red de radio que maximice la cobertura, minimice las interferencias y optimice el uso de recursos. La suma de $1s$ indica la cantidad total de puntos cubiertos, la suma de $0s$ determina la cantidad de puntos no cubiertos y la suma de $-1s$ establece la cantidad de sitios iluminados por más de una antena. En la Ecuación 3 se muestra la función objetivo utilizando los valores obtenidos.

$$f(x) = (1 - \frac{pnc}{tpt}) \times (1 - \frac{pi}{tpc}) \quad (3)$$

donde *pnc* es la cantidad depuntos no cubiertos, *tpt* es la cantidad total de puntos del terreno, *pi* es la cantidad de puntos interferidos y *tpc* la cantidad *total de puntos cubiertos*.

El primer factor determina el grado de cobertura del terreno ya que si todos los sitios tienen señal, el numerador del cociente es 0. Con esto se satisface la primera parte del objetivo que es maximizar el área cubierta. El segundo factor de la Ecuación 3 analiza la calidad con la cual se realiza la cobertura del terreno, cuanto menor es la cantidad de sitios interferidos es mejor la calidad de la cobertura lograda de los puntos irradiados. Finalmente, la aptitud de la solución es mejor cuando el valor objetivo es más cercano a 1.

En la siguiente sección se describen los experimentos realizados.

2.5. Descripción del experimento

Para analizar el desempeño de los algoritmos CHC propuestos (QCHC-RE, QCHC-TE y QCHC-ML) se los compara con los AGs con operadores de cruce de un punto (AG-

OPX), dos puntos (AG-TPX) y uniforme (AG-UX). Por cada matriz se varió la cantidad de sitios candidatos. La configuración de las siete instancias utilizadas se muestran en la Tabla 1. La columna *Instancia* presenta el nombre de cada una de las instancias, la columna *Matriz* muestra los valores que indican la cantidad de filas y columnas de cada una de las matrices utilizadas para discretizar el terreno. La columna *Tamaño* indica la cantidad de celdas de cada una de las matrices como resultado del producto de las filas por las columnas. La columna *Sitios* indica la cantidad de sitios candidatos para cada instancia. El tamaño inicial de la matriz es de 12x12 elementos, en incremento de 3 celdas por fila y columna hasta llegar a 30x30. Se analiza el comportamiento de los algoritmos con diferentes grados de complejidad, tanto en las dimensiones del terreno como en la cantidad de sitios que componen la solución óptima.

Las antenas utilizadas por cada sitio son del tipo omnidirec-

Tabla 1. Configuraciones de los escenarios de cobertura

Instancia	Matriz	Tamaño	Sitios
Inst_01	12x12	144	32
Inst_02	15x15	225	50
Inst_03	18x18	324	72
Inst_04	21x21	441	98
Inst_05	24x24	576	128
Inst_06	27x27	729	162
Inst_07	30x30	900	200

cional con centro c_{ij} y radio de un elemento alrededor del centro cubriendo una cantidad de nueve celdas de la matriz del terreno por antena: ocho r_{ij} + el centro $c_{i,j}$. La Figura 2 muestra un ejemplo de cobertura de una antena omnidireccional sobre la matriz M. Los AGs utilizan una población

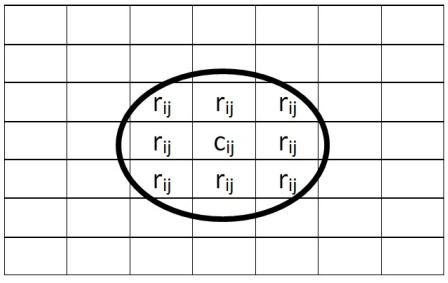


Figura 2. Modelado de una antena omnidireccional

de 100 individuos y 5.000 iteraciones con una probabilidad de cruce de $0,9$ y de mutación de $0,001$. Para el algoritmo QCHC-RE se utilizó una probabilidad de mutación de $0,05$, para QCHC-TE se definió como plantilla patrón el $95\,\%$, variando solamente el $5\,\%$ de los bits de la mejor solución. Todos los algoritmos QCHC- utilizan una población de 500 individuos y 2.500 iteraciones. En todos los casos el mejor individuo siempre fue conservado en la nueva población. Para cada una de las instancias del experimento se realizaron 30 corridas independientes para asegurar la significación estadística. Se emplea Windows 8 como plataforma corriendo sobre un procesador Intel Core@ i7-3630QM con 12 Gb de memoria RAM y 1 Tb de disco rígido. Para analizar el rendimiento de los algoritmos propuestos se comparan los algoritmos que lograron el 100 % de las veces encontrar la cobertura ideal. Únicamente a los algoritmos que cumplen esta condición, se aplican los test estadísticos a las variables de performace tiempo y número de evaluaciones. Estas métricas permiten establecer si existen diferencias estadísticamente significativas entre ellos y cual es el algoritmo de mejor rendimiento.

A continuación se explican las tablas con los resultados obtenidos del experimento.

3. Resultados

Los resultados obtenidos por el experimento se organizan en diferentes tablas para poder evaluar el rendimiento de los algoritmos presentados. Se toma como parámetro las variables de performance tiempo y evaluaciones para las instancias en las que al menos dos algoritmos hayan alcanzado el $100\,\%$ de la veces el mejor valor conocido en las 30 corridas independientes. En las instancias en las que ningún algoritmo haya alcanzado la condición anterior se toma la variable valor objetivo como métrica de rendimiento. También se presenta una tabla con los resultados de los tests estadísticos donde se indica si existen diferencias estadísticamente significativas entre los algoritmos. Las Tablas 2 y 3 muestran el porcentaje de éxitos obtenidos por AG-UX, AG-OPX, AG-TPX y los algoritmos QCHC-RE, QCHC-TE y QCHC-ML respectivamente para las siete instancias. Para ambas tablas la columna *Instancia* contiene la referencia de las diferentes configuraciones descriptas en la Tabla 1. Las columnas restantes tienen el nombre de cada algoritmo analizado y el porcentaje de éxitos obtenidos en cada instancia. La fila *Promedio* muestra una primera aproximación sobre la eficiencia de cada algoritmo conforme a la complejidad de cada instancia. Las Tablas 4 y 5 muestran los valores promedio obtenidos por cada uno de los algoritmos evaluados en las instancias que lograron el 100 % de éxitos.

Tabla 2. Porcentaje de éxitos obtenidos por AG-UX, AG-OPX y AG-TPX

Instancia	AG-UX	AG-OPX	AG-TPX
Inst_01	**100 %**	**100 %**	**100 %**
Inst_02	93,33 %	70 %	**100 %**
Inst_03	66,7 %	66,7 %	**100 %**
Inst_04	**100 %**	**100 %**	86,67 %
Inst_05	90,00 %	80,00 %	70,00 %
Inst_06	63,33 %	66,67 %	86,67 %
Inst_07	26,67 %	33,33 %	30 %
Promedio	77,41 %	73,81 %	**81,90 %**

Tabla 3. Porcentaje de éxitos obtenidos por los algoritmos QCHC-RE, QCHC-TE y QCHC-ML

Instancia	QCHC-RE	QCHC-TE	QCHC-ML
Inst_01	**100 %**	**100 %**	**100 %**
Inst_02	**100 %**	**100 %**	93 %
Inst_03	**100 %**	**100 %**	93 %
Inst_04	**100 %**	**100 %**	80 %
Inst_05	**100 %**	97 %	77 %
Inst_06	**90 %**	80 %	30 %
Inst_07	**43 %**	13 %	0 %
Promedio	**90 %**	84 %	68 %

La Tabla 4 corresponde a los resultados obtenidos por los AGs y la Tabla 5 los valores promedio observados para los CHCs para la variable de performance tiempo.

Tabla 4. Valores promedio para la variable de performance tiempo por AG-UX, AG-OPX y AG-TPX

Instancia	AG-UX	AG-OPX	AG-TPX
Inst_01	**1,71**	2,81	1,95
Inst_02	N/A	N/A	3,32
Inst_03	N/A	N/A	5,67
Inst_04	**9,88**	17,53	N/A

Los resultados obtenidos para las variables de performance número de evaluaciones se muestran en Tablas 6 y 7 para los AGs y CHCs respectivamente. Ambas tablas muestran los valores promedio obtenidos por cada uno de los algoritmos evaluados en las instancias que lograron el 100 % de éxitos. Se utiliza la sigla N/A a los valores que no cumplen esta condición. Las Tablas 8 y 9 muestran el valor objetivo promedio de las instancias en que los algoritmos

Tabla 5. Valores promedio obtenidos para la variable de performance tiempo por los algoritmos QCHC-RE, QCHC-TE y QCHC-ML.

Instancia	QCHC-RE	QCHC-TE	QCHC-ML
Inst_01	**0,53**	1,04	2,20
Inst_02	2,29	**1,85**	N/A
Inst_03	**4,06**	9,15	N/A
Inst_04	**7,41**	20,44	N/A

que no alcanzaron el 100 % de éxito para los AGs y CHCs respectivamente. Se utiliza la sigla N/A a los valores que no cumplen alguna condición .

Tabla 6. Valores promedio obtenidos la variable de performance número de evaluaciones por AG-UX, AG-OPX, AG-TPX

Instancia	AG-UX	AG-OPX	AG-TPX
Inst_01	**17023**	28217	19663
Inst_02	N/A	N/A	**32307**
Inst_03	N/A	N/A	**52793**
Inst_04	**86247**	153673	N/A

Tabla 7. Valores promedio obtenidos para la variable de performance número de evaluaciones por los algoritmos QCHC-RE, QCHC-TE y QCHC-ML

Instancia	QCHC-RE	QCHC-TE	QCHC-ML
Inst_01	**6550**	8100	33567
Inst_02	26733	**11800**	N/A
Inst_03	**40650**	54400	N/A
Inst_04	**67000**	73933	N/A

Por la naturaleza del experimento los valores obtenidos cumplen la condición de independencia, se aplica el test de Kolgomorov-Smirnov para determinar si los datos se ajustan a una distribución normal y el test de Levene para verificar si los datos tienen homocedasticidad. Los resultados obtenidos por los algoritmos no cumplen con las condiciones de normalidad y homocedasticidad, por lo tanto se utiliza el test no paramétrico de Kruskal-Wallis para detectar si existen diferencias estadísticamente significativas.

Tabla 8. Valores promedio obtenidos para la variable de performance valor objetivo por los algoritmos AG-UX, AG-OPX, AG-TPX

Instancia	AG-UX	AG-OPX	AG-TPX
Inst_06	0,9968	0,9979	**0,9992**
Inst_07	0,9926	0,9952	**0,9936**

Tabla 9. Valores promedio obtenidos para la variable de performance valor objetivo por los algoritmos QCHC-RE, QCHC-TE y QCHC-ML

Instancia	QCHC-RE	QCHC-TE	QCHC-ML
Inst_06	**0,9993**	0,9975	0,9878
Inst_07	**0,9961**	0,9873	0,9633

En la Tabla 10 la columna *Instancia* tiene el nombre de las instancias evaluadas, las columnas *Evaluaciones*, *Tiempo* y *Valor Objetivo* contienen los resultados del test de *Kruskal-Wallis*. Se utiliza el símbolo $(+)$ si existen diferencias y el símbolo $(-)$ en caso contrario. En todos los casos donde se detectan diferencias estadísticamente significativas se aplica el test de Tukey para identificar el algoritmo de mejor rendimiento. Para todas las pruebas realizadas se usó un nivel de significancia $\alpha = 0{,}05$.

Tabla 10. Diferencias estadísticamente significativas obtenidas en cada test

Instancia	Evaluaciones	Tiempo	Valor objetivo
Inst_01	(+)	(+)	N/A
Inst_02	(-)	(-)	N/A
Inst_03	(-)	(-)	N/A
Inst_04	(+)	(+)	N/A
Inst_05	N/A	N/A	N/A
Inst_06	N/A	N/A	(+)
Inst_07	N/A	N/A	(+)

4. Discusión

Observando los resultados de la Tabla 2 con respecto a los resultados de la Tabla 3 podemos ver que el algoritmo propuesto QCHC-RE logra el 100 % de los éxitos en 5 de las 7 instancias propuestas. También obtiene el mayor porcentaje promedio de todos los algoritmos propuestos (90 %) seguido por el QCHC-TE que logró resolver 4 de las 7 instancias (84 %). En cuanto a los AGs, podemos observar que AG-TPX tuvo el mayor porcentaje (81,90 %) de éxito (Inst_01, Inst_02 e Inst_03) con respecto a AG-UX y AG-OPX. Estos últimos tuvieron rendimientos similares, con solo 2 instancias con un 100 % de éxito y un promedio de 77,41 % para AG-UX y un 73,81 % de éxito para AG-OPX. Cabe destacar que el QCHC-RE es el único algoritmo que resuelve la instancia 5 con el 100 % de éxito y es el que ha obtenido un mejor desempeño en las instancias Inst_06 (90 %) e Inst_07 (43 %). Podemos comparar los resultados obtenidos en la Tablas 4 y 5 para la variable de performance tiempo y el Algoritmo QCHC-RE resulta el de mejor rendimiento para las instancias Inst_01, Inst_03 e Inst_04. Para la instancia Ins_2 el Algoritmo QCHC-TE obtiene el mejor tiempo promedio. Las Tablas 6 y 7 muestran los valores obtenidos para la variable de performance número de evaluaciones requeridas para alcanzar el óptimo. Nuevamente el algoritmo QCHC-RE obtiene el menor número en las instancias Inst_01, Inst_03 e Inst_04 por sobre los AGs y las versiones de QCHC. Solamente en la Inst_02, QCHC-TE obtiene el mejor valor de rendimiento. Para todos los casos las variantes de QCHC-RE y QCHC-ME tienen un mejor redimiento que los AGs.

Observando los resultados de las Tablas 8 y 9 para la variable de performance valor objetivo QCHC-RE es apenas superior que AG-TPX pero tiene el mejor valor de rendimiento con respecto a AG-UX, AG-OPX, AG-TPX y QCHC-ML. Con respecto a la Inst_07 el QCHC-RE obtiene el mejor valor objetivo promedio con respecto al resto de sus competidores. La Tabla 10 muestra los resultados del test de Kruskal-Wallis para determinar si existen diferencias estadísticamente significativas. En caso de existir estas diferencias se aplicó el test de Tukey para identificar entre qué algoritmos. Se observa que solamente en la Inst_01 y la Inst_04 aparecen diferencias estadísticamente significativas para las variables de performance número de evaluaciones y tiempo. En las instancias Inst_06 e Inst_07 aparecen diferencias estadísticamente significativas para la variable de performace valor objetivo. Podemos afirmar que la propuesta QCHC-RE tiene una mejor performance con un 95 % de confianza para los test estadísticos aplicados. Esto significa que esta versión del algoritmo CHC es una mejora efectiva con respecto a los AGs competidores y las otras versiones del QCHC para las instancias Inst_01 e Inst_04. Con respecto a las instancias Inst_02 e Inst_03 no se puede determinar cuál es el mejor algoritmo dado que el test de Kruskal-Wallis no establece diferencias estadísticamente significativas. La gráfica que se muestra en la Figura 3 representa la cantidad de instancias con el 100 % de los éxitos obtenido por cada algoritmo. Podemos observar que el Algoritmo propuesto QCHC-RE es el que más instancias resuelve, seguido por el QCHC-TE. Con respecto a las instancias Inst_06 e Inst_07 podemos afirmar que el QCHC-RE tuvo el mejor rendimiento dentro del grupo con mejor valor objetivo promedio con un 95 % de confianza para los test estadísticos aplicados.

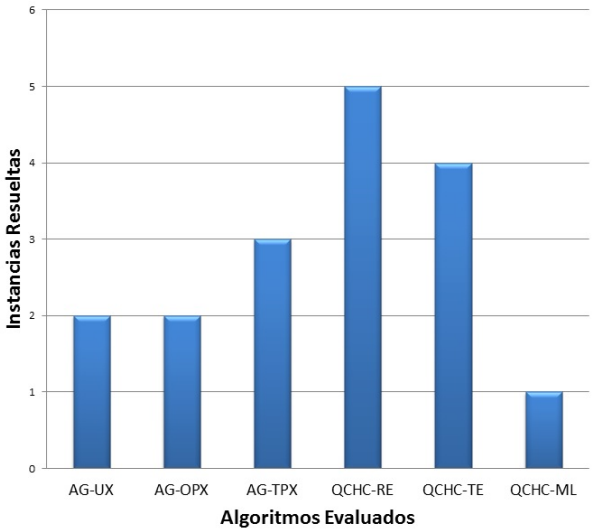


Figura 3. Cantidad de instancias resueltas por los algoritmos evaluados

En resumen, la instancia Inst_01 fue resuelta por todos los competidores y hubo diferencias estadísticamente significativas tanto en tiempo como en iteraciones a favor del QCHC-ME. Las instancias Ins_02 e Inst_03 fueron resueltas por QCHC-RE, QCHC-TE y AG-TPX. El Algoritmo QCHC-ME tuvo mejores resultados pero sin establecer diferencias estadísticamente significativas ente ellos sobre las varibales de performance tiempo e iteraciones. La instancia Inst_05 fue resuelta solamente por QCHC-ME y las instancias Inst_06 e Inst_07 no fueron resueltas con el 100 % de éxitos por ninguno de los algoritmos propuestos. No obstante el Algoritmo QCHC-RE fue el que mejor rendimiento tuvo en cuanto a la cantidad de instancias resueltas (5). Teniendo en cuenta los escenarios generados para este problema podemos decir que el uso de este tipo de escenarios tiene como finalidad analizar el comportamiento de los algoritmos con diferentes grados de complejidad. Esto permite extender la aplicabilidad de estos algoritmos a casos reales donde la matriz del terreno puede contener información geográfica a escala real. En particular, a la zona de explotación petrolera entre las provincias de Chubut y Santa Cruz, esto permitiría tener una idea a priori de los mejores lugares para lograr una cobertura de señal en una geografía extensa y compleja. Las necesidades de telesupervisar los pozos, las baterías de concentración de petróleo, los oleoductos, gasoductos, etc. así como los estudios para la extensión de los sistemas de comunicaciones para el personal que ejecuta las tareas de campo, pueden verse altamente satisfechas con el uso de metaherísticas. Además se pueden incorporar otros modelos de iluminacón de antenas (paneles de 60^o, 90^o y 120^o) y la validación de los puntos cubiertos, no cubiertos e interferidos puede ser determinada por un modelo matemático de cálculo de enlace [10]. Todos estos elementos forman parte del fenotipo del problema, la forma de representación en el espacio del genotipo es la misma. La función objetivo presentada busca la eficiencia en la iluminación de los puntos del terreno y la optimización de los recursos es una consecuencia de la eficiencia. Esta función difiere respecto al trabajo realizado por [11] donde se toman estos objetivos como separados y se los analiza con un enfoque multi objetivo. El control de la variabilidad genética propuesto permite tener una idea del comportamiento general de la población. Permitiendo distinguir si la población tiende a converger en un óptimo local y si dos individuos son lo suficientemente distintos para prevenir una situación de incesto. A diferencia del mecanismo original que espera a que dos individuos se parezcan bit a bit, esto permite tener una observación a priori, ahorrando el costo computacional de comparar los individuos. Esto se vuelve más notable en cadenas de gran extensión. Se deben seguir explorando soluciones algorítmicas que permitan llegar al 100 % de éxitos en todas las instancias, una alternativa es realizar variaciones en los métodos de sacudida. Posiblememte incorporando mayor divesidad en la nueva población permita al algoritmo explorar otras zonas. Esto puede deducirse de los resultados obtenidos por el QCHC-ML, que es el de peor rendimiento general de las variantes presentadas, en particular en las instancias más complejas. Este algoritmo utiliza estrategia de reemplazo que le quita diversidad a cada nueva población.

5. Conclusión

En este trabajo se presentan tres AGs denominados AG-UX, AG-OPX, AG-TPX y tres versiones CHC denominadas QCHC-RE, QCHC-TE y QCHC-ML. Se utilizó un nuevo mecanismo de prevención de incesto basado en la variabilidad de la población y una función objetivo orientada a la eficiencia de la cobertura de señal en el terreno. Para el uso de cadenas de bits extensas la forma tradicional del algoritmo CHC para la prevención del incesto demanda un gran esfuerzo computacional. El mecanismo propuesto puede detectar la convergencia prematura y dar una idea general del comportamiento de toda la población con un menor costo computacional. En todo momento se mantuvo el operador HUX y los mecanismos de selección de la población mantuvieron el criterio elitista como lo indica el algoritmo CHC canónico. El algoritmo QCHC-RE es el más similar al CHC canónico donde el criterio de selección poblacional es completamente elitista, el QCHC-TE utiliza la selección por toneos sin reemplazo entre dos individuos seleccionando el de mejor valor objetivo y el QCHC-ML solamente reemplaza el peor individuo de la antigua población por el mejor en-

contrado. Los dos primeros algorimtos QCHC tuvieron en general mejores resultados que el resto de los algoritmos, el QCHC-ML fue el de peor desempeño y el que más tendía al estancamiento debido al mecanismo de selección poblacional. Con respecto a las variables de performance establecidas QCHC-RE se encuentra siempre en el grupo de los mejores algoritmos en las instancias donde existen diferencias estadísticamente significativas con los mejores valores en rendimiento. En las instancias donde no hay diferencias estadísticamente significativas el QCHC-TE aparece como un posible competidor superando en una de las instancias en rendimiento al QCHC-RE. En las instancias donde ninguno de los algoritmos pudo resolverlas con el 100 % de éxitos el QCHC-RE nuevamente obtiene los mejores valores de rendimiento. Estos resultados permiten seguir buscando metaheurísticas capaces de resolver las siete instancias propuestas con una complejidad creciente de 2^n. En trabajos futuros se podrán abordar tamaños de problemas con un espacio de búsqueda superior y se analizarán otros aspectos de mejoras del CHC como por ejemplo el mecanismo de selección y la paralelización de los algoritmos propuestos.

Agradecimientos

Se agradece la cooperación del equipo de proyecto del LabTEm y a la Universidad Nacional de la Patagonia Austral, por el continuo apoyo y los valiosos aportes para el desarrollo de este trabajo. Además el último autor agradece el constante apoyo de la Universidad Nacional de San Luis y a ANPCYT que financia su investigación.

Referencias

[1] H. Anderson and J. McGeehan. Optimizing Microcell Base Station Locations Using Simulated Annealing Techniques. In *Proceedings 44th IEEE Conference on Vehicular Technology*, pages 858–862. Morgan Kaufmann, 1994.

[2] P. Chawdhry, R. Roy, and R. Pant. *Soft Computing in Engineering Design and Manufacturing*. Springer London, 1998.

[3] D. W. Corne, G. Smith, and M. Oates. *Telecommunications Optimization: Heuristic and Adaptive Computation Techniques*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2000.

[4] L. Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In *Foundations of Genetic Algorithms*, pages 265–283. Morgan Kaufmann, 1991.

[5] T. Fritsch, K. Tutschku, and K. Leibnitz. Field strength prediction by ray-tracing for adaptive base station positioning in mobile communication networks, 1995.

[6] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Artificial Intelligence. Addison-Wesley, 1989.

[7] N. G. K. Tutschku and P. Tran-Gia. An integrated Approach to Cellular Network Planning. In *Institute of Computer Science*. University of Wurzburg, 1995.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680, 1983.

[9] F. Luna, J. J. Durillo, A. J. Nebro, and E. Alba. Evolutionary algorithms for solving the automatic cell planning problem: a survey. *Engineering Optimization*, 42(7):671–690, Dec. 2010.

[10] P. Muñoz. *Estudio del modelo de radio propagación Longley-Rice en entornos rurales para la banda de 2.4 GHz*. 2011.

[11] A. J. Nebro, G. Alba, Enrique y Molina, F. Chicano, and J. J. Luna, Francisco y Durillo. Optimal antenna placement using a new multi-objective chc algorithm. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 876–883. ACM, 2007.

[12] P. K. D. K. P. Callegari, F. Guidec. Parallel island-based genetic algorithm for radio network design. In *Journal of Parallel and Distributed Computing (47)*, pages 86–90, 1997.

[13] J. Parsons. *The Mobile Radio Propagation Channel*. Wiley, 2000.

[14] S. Priem-Mendes, G. Molina, M. A. Vega-Rodríguez, J. A. G. Pulido, Y. Sáez, G. Miranda, C. Segura, E. Alba, P. Isasi, C. León, and J. M. Sánchez-Pérez. Benchmarking a wide spectrum of metaheuristic techniques for the radio network design problem. *IEEE Trans. Evolutionary Computation*, 13(5):1133–1150, 2009.

[15] E. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.

[16] E. G. Talbi and N. Cahon, S. y Melab. Designing cellular networks using a parallel hybrid metaheuristic on the computational grid. *Comput. Commun.*, 30(4):698–713, Feb. 2007.

Generador de Modelos de Programación Lineal para la optimización del cursado de la carrera de Ingeniería de Sistemas de Información dirigida a estudiantes con disposiciones horarias condicionadas por trabajo

Rojas, Cristina
Lescano, Gonzalo Emmanuel
Khouri, Ramiro Federico
Ibarra, Gerardo Daniel

Universidad Tecnológica Nacional – Facultad Regional Tucumán

Abstract
Este documento surge de la necesidad de contribuir a la disminución del promedio de años de cursado de los estudiantes de la carrera Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional de la Facultad Regional de Tucumán. El generador de modelos de programación lineal se formulará para permitir que aquellos estudiantes que no disponen de la cantidad de horas necesarias para cumplir en tiempo y forma con las exigencias de la carrera, puedan ocupar sus tiempos de la manera más beneficiosa posible. Se ha optado por un modelo multicriterio por ponderación para la resolución del problema, teniendo en cuenta criterios de restricción de materias correlativas, franjas horarias disponibles, beneficio por cursar la materia (inmediato o a largo plazo) y restricciones de horarios. La ponderación será utilizada para poder llevar a cabo la elección de una materia que sea más conveniente respecto a otra en su cursado. Mediante la teoría de grafos y empleo de programación lineal por objetivos llegamos a la formulación de un modelo que se ajusta a las situaciones particulares de un estudiante que requiera por diferentes motivos la optimización del cursado de materias, en virtud de su tiempo disponible, dando como resultado la mejor combinación de materias para concluir la carrera en el mejor tiempo disponible.

Palabras Clave
Estado académico, cursado, plan de estudio, materia regular, materia finalizada, correlativas.

Introducción

Este documento surge de la necesidad de contribuir a la disminución del promedio de años de cursado de los estudiantes de la Universidad Tecnológica Nacional de la Facultad Regional de Tucumán en la carrera Ingeniería en Sistemas de Información.
Hay una gran cantidad de estudiantes que estudian y trabajan, lo que los lleva a disponer de menos tiempo para cursar, y no poder realizar la carrera académica estipulada. Además, existe un importante desconocimiento en el plan de estudios, lo que lleva a cursar materias sin considerar sus correlativas futuras. Se intentará resolver dicha problemática con un generador de modelos de programación lineal.
El modelo de programación lineal se formulará para permitir que aquellos estudiantes que no disponen de la cantidad de horas necesarias para cumplir en tiempo y forma las exigencias de la carrera, puedan ocupar sus tiempos de la manera más beneficiosa posible.

La carrera Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional Facultad Regional Tucumán, en los últimos 10 años cuenta con un bajo porcentaje de alumnos que finalizan el cursado de la carrera en el tiempo especificado de 5 años.
Presenta, además, un considerable número de abandonos de alumnos durante su cursado, debido en parte, a que los mismos no realizaron una buena planificación en las materias involucradas en su cursado anual.
Se observa además que una importante cantidad de estudiantes cursan pocas

materias durante el año lectivo, por razones tales como no cumplir con las materias finalizadas necesarias para cursar las correlativas, debido al desconocimiento del plan de estudios.
Además, se considera el fenómeno de deserción tardía; ya que hay un conjunto de alumnos avanzados de cohortes anteriores que no se han graduado y que se vuelven crónicos o que aumentan la duración real de sus carreras dado que a partir del tercer año son rápidamente incorporados por el mercado laboral. Este fenómeno debe entenderse como parte de la situación socio económica imperante en el alumnado tucumano. La UTN y la Facultad a través de las becas de servicios y el Ministerio de la Nación a través de las becas Tics y Bicentenario intentan paliar las situaciones de equidad que llevan al alumno a buscar ingresos y atrasarse en sus estudios[1].

Objetivo

El objetivo de nuestro trabajo se basa en realizar un documento que contendrá un generador de modelos de programación lineal, que permita brindar una recomendación a los estudiantes sobre la elección de materias a cursar en el próximo período de inscripciones de acuerdo a su disposición horaria.
Se desea hacer viable la posibilidad de trabajar y estudiar a la vez, obteniendo el mayor beneficio en base a las horas disponibles para el cursado.
El resultado del modelo generado permitirá que el alumno pueda optimizar el cursado, de acuerdo a su situación académica actual y al tiempo que tenga disponible, determinado por las materias que representan un mayor beneficio a la hora de cursarlas.
Además, se contemplará la minimización de los espacios de tiempo libre entre el cursado de materias.
Se ha optado por la generación de un modelo multicriterio por ponderación para la resolución del problema, teniendo en cuenta criterios de restricción de materias correlativas, franjas horarias por trabajo, beneficio por cursar la materia (inmediato o a largo plazo) y restricciones de horarios. La ponderación será utilizada para poder llevar a cabo la elección de una materia que sea más conveniente respecto a otra en su cursado, de acuerdo a los beneficios mencionados anteriormente.

Elementos del Trabajo y metodología

Para desarrollar la propuesta se recurrió a Teoría de Grafos y Programación por Metas.
Una red es un grafo con etiquetas en los arcos (i,j) que representan costos y con flujos (xij) asociados a tales arcos. En nuestro modelo, los arcos representan la carga horaria, mientras que los flujos simulan las relaciones de correlatividad entre las asignaturas.
En ocasiones es necesario mostrar la equivalencia entre diferentes problemas de flujo en redes o formular el problema de tal forma que se lo pueda representar en algún código de computadora. Las transformaciones para problemas de flujo de costo mínimo, también se pueden aplicar a casos especiales tales como problemas del camino más corto, flujo máximo y problemas de asignación [2].
En caso de ser preciso atribuir a cada arco un número específico, llamado valuación, ponderación o coste según el contexto, y se obtiene así un grafo valuado o ponderado.
La Programación Lineal es un procedimiento o algoritmo matemático mediante el cual se resuelve un problema indeterminado, formulado a través de un sistema de inecuaciones lineales, optimizando la función objetivo, también lineal.
Consiste en optimizar (minimizar o maximizar) una función lineal, denominada función objetivo, de tal forma que las variables se emplean para resolver situaciones del tipo "inclusión" o "exclusión"[3].
La Programación por Metas es un caso de Programación Lineal. Se trata de una técnica aplicada a problemas con múltiples

criterios de optimización, es decir, con metas que desean lograrse, y que sobre una estructura de programación lineal permite seleccionar las variables que ofrecen la mejor solución al problema abordado, con la posibilidad de alcanzar solo algunos de los objetivos incluidos en el modelo. Se ha recurrido a modelar con Programación por Metas, usando variables enteras binarias, para obtener el conjunto de materias recomendadas para la optimización de los objetivos de tiempo y beneficio.

Modelo Resultante

El plan de estudios se modela previamente en dos grafos dirigidos, que representan las materias correlativas del Plan de Estudio. El primero se denomina GRPC y refleja la relación de correlatividad de materias regulares para cursar; el segundo se denomina GAPC, y refleja la relación de correlatividad de materias aprobadas para cursar.

Entrada

$$\mathrm{M} = \{\text{Materias del Plan}\}$$
$$\mathrm{A} = \{\text{Materias Aprobadas}\}$$
$$\mathrm{R} = \{\text{Materias Regularizadas}\}$$
FH: Franja Horaria Disponible

Generación de Grafos

Relación de adyacencia de materias regularizadas para cursar

$$RPC = \left\{ \begin{matrix} (a,b) \in MxM: \\ b\ requiere\ la\ regularidad\ de\ a \\ para\ ser\ cursada \end{matrix} \right\}$$

Relación de adyacencia de materias aprobadas para cursar

$$APC = \left\{ \begin{matrix} (a,b) \in MxM: \\ b\ requiere\ la\ aprobación\ de\ a \\ para\ ser\ cursada \end{matrix} \right\}$$

Grafo de correlativas de Materias Regularizadas para Cursar

$$GRPC = G(M, RPC)$$

Grafo de correlativas de Materias Aprobadas para Cursar

$$GAPC = G(M, APC)$$

Subgrafos

Utilizamos el grafo GRPC y analizando un estado académico en particular se eliminan los nodos de las materias regularizadas.
Del mismo modo se toma el grafo GAPC (correspondiente a las materias aprobadas para cursar) y analizando un determinado estado académico se eliminan los nodos de las materias aprobadas.
De esta manera obtenemos un subgrafo a través de la eliminación de nodos, es decir, un subgrafo a partir de RPC y otro a partir de APC; lo que nos indica, que las materias habilitadas para cursar son aquellas que poseen grado interno 0 en ambos grafos.

PA: $Materias\ Pendientes\ de\ Aprobar$
$PA = M - A$
Todas las materias del plan exceptuando las materias ya aprobadas.

PR: $Materias\ Pendientes\ de\ Regularizar$
$PR = M - (R \cup A)$
Todas las materias del plan exceptuando, las materias regularizadas y las aprobadas.

Relación de adyacencia de materias regulares para cursar, pendientes

$$PRPC = \left\{ \begin{matrix} (a,b) \in PR\ x\ PR: \\ b\ requiere\ la \\ regularidad\ de\ a \\ para\ ser\ cursada \end{matrix} \right\}$$

Relación de adyacencia de materias aprobadas para cursar, pendientes

$$PAPC = \left\{ \begin{matrix} (a,b) \in PA\ x\ PA: \\ b\ requiere\ la\ aprobación \\ de\ a\ para\ ser\ cursada \end{matrix} \right\}$$

Grafo de correlativas de Materias Regularizadas para Cursar pendientes

$$GPRPC = G(PR, PRPC)$$

Grafo de correlativas de Materias Aprobadas para Cursar pendientes

$$GPAPC = G(PA, PAPC)$$

Materias Posibles a Cursar

$$MP = \{m \in GPAPC: \\ GPRPC.Grado_{int}(m) \\ + GPAPC.Grado_{int}(m) = 0 \}$$

Cálculo de beneficios de las materias

$Sea\ m\ \in MP$
$Y\ sean\ Grado_{int}\ y\ Grado_{ext}\ funciones\ correspondientes\ a\ GPRPC:$

Beneficio de Regularizar una materia:

$$BR: Beneficio\ de\ Regularizar = 1$$

Beneficio inmediato de una materia es igual al grado externo de la misma en GPRPC:

$$BI(m): Beneficio\ Inmediato \\ = Grado_{ext}(m)$$

El beneficio inmediato fraccionario de una materia es igual a la sumatoria de la inversa del grado interno de cada sucesor de la misma:

$$BIF(m): Beneficio\ inmediato\ fraccionario \\ = \sum_{c \in sucesores(m)} \frac{1}{Grado_{int}(c)}$$

El beneficio inmediato truncado de una materia es igual a la sumatoria de la parte entera de la inversa del grado interno de cada sucesor de la misma.

$$BIT(m): Beneficio\ inmediato\ truncado \\ = \sum_{c \in sucesores(m)} INT\left(\frac{1}{Grado_{int}(c)}\right)$$

El beneficio a largo plazo de una materia es igual a la sumatoria de los beneficios inmediatos de cada sucesor de la misma:

$$BLP(m): Beneficio\ largo\ plazo \\ = \sum_{c \in sucesores(m)} BI(c)$$

Ponderación de los beneficios:

Ponderación

$$P(m): Ponderación\ de\ la\ materia \\ = BC(m) + 4 * BIT(m) \\ + 2 * BIF(m) + 0.5 * BLP(m) \\ + 0.25 * BI(m)$$

Funcional:

$$Z[MAX] = \sum_{m \in MP} Ponderación(m) * M_m$$

El modelo determinará las materias que representan un mayor beneficio a la hora de cursarlas, a partir de la situación académica del alumno, teniendo en cuenta sus restricciones horarias. Ver Figura 1.

Para la realización de este modelo se llevará a cabo un estudio de la carga horaria de las materias del correspondiente plan de estudios, descriptas por los módulos u horas cátedra de cada materia, presentes en la carrera considerando la comisión en las cuales se las puede cursar, el día y turno (mañana, tarde, noche) en el que se dictan las mismas.
A partir del relevamiento sobre los datos de las correspondientes materias se procederá a su representación en el modelo mediante una variable binaria la cual servirá en la realización de las restricciones necesarias para la resolución del problema planteado:

($M_mC_cD_dT_tO_o$) => Variable binaria que se puede leer como "Materia m, cursada en la comisión c el día d, durante el turno t en el horario o.

Subíndices de las Variables
m: Materia
c: comisión

d: día (1 Lunes | 2 Martes | 3 Miércoles | 4 Jueves | 5 viernes)
t: turno (0 mañana | 1 tarde | 2 noche)
o: módulo

Durante el estudio del problema planteado se ha optado por la construcción de un modelo que divida las materias por módulos, permitiendo la formulación matemática de restricciones que intenten resolver las diferentes dificultades con las que nos encontramos en la hipótesis de dicho problema, estas son:

Día y Módulo: para verificar que se ocupe un módulo de un día con sólo una materia

$$\sum_{1}^{n}(M_m C_c D_d T_t O_o)_i \leq 1$$

Para las n variables que posean igual $\boldsymbol{d}, \boldsymbol{t}$ *y* $\boldsymbol{o}$

Cursado Completo: para verificar que se curse la *carga horaria* total de una *materia* en sólo una *comisión*

$$\sum_{1}^{n}(M_m C_c D_d T_t O_o)_i = CMH * \boldsymbol{M_m C_c}$$

Para las n variables que posean igual $\boldsymbol{m}$ *y* $\boldsymbol{c}$

CMH: *cantidad de módulos de la carga horaria de la materia*

Comisión: para verificar que se curse una *materia* en sólo una *comisión*

$$\sum_{1}^{n}(M_m C_c D_d T_t O_o)_i = CMH * \boldsymbol{M_m}$$

Para las n variables que posean igual $\boldsymbol{m}$

Colisiones de horarios: para que no coincidan los horarios de materias en

- Turno 1, Modulo 5 y Turno 2, Módulo -1
- Turno 1, Modulo 6 y Turno 2, Módulo -1
- Turno 1, Modulo 6 y Turno 2, Módulo 0
- Turno 1, Modulo 7 y Turno 2, Módulo 0
- Turno 1, Modulo 7 y Turno 2, Módulo 1

$$\sum_{1}^{n}(M_m C_c D_d T_1 O_5)_i + \sum_{1}^{m}(M_m C_c D_d T_2 O_{-1})_j \leq 1$$

Para las n y m variables que posean igual $\boldsymbol{d}$

$$\sum_{1}^{n}(M_m C_c D_d T_1 O_6)_i + \sum_{1}^{m}(M_m C_c D_d T_2 O_{-1})_j \leq 1$$

Para las n y m variables que posean igual $\boldsymbol{d}$

$$\sum_{1}^{n}(M_m C_c D_d T_1 O_6)_i + \sum_{1}^{m}(M_m C_c D_d T_2 O_0)_j \leq 1$$

Para las n y m variables que posean igual $\boldsymbol{d}$

$$\sum_{1}^{n}(M_m C_c D_d T_1 O_7)_i + \sum_{1}^{m}(M_m C_c D_d T_2 O_0)_j \leq 1$$

Para las n y m variables que posean igual $\boldsymbol{d}$

$$\sum_{1}^{n}(M_m C_c D_d T_1 O_7)_i + \sum_{1}^{m}(M_m C_c D_d T_2 O_1)_j \leq 1$$

Para las n y m variables que posean igual $\boldsymbol{d}$

Franja Horaria:

$$\sum_{1}^{n}(M_m C_c D_d T_t O_o)_i = 0$$

Para las n variables que estén fuera de la franja horaria.

Resultados

Como resultado del documento planteado, se alcanzó la representación formal de un generador de modelos de programación lineal por metas con ponderaciones que intenta recomendar sobre el empleo óptimo de cursado dada una situación particular.
Se logró modelizar el plan de estudios, representar con relaciones binarias las exigencias que este propone, representar con modelos matemáticos los impedimentos horarios y las condiciones necesarias para que se llegue a un modelo eficiente y confiable.

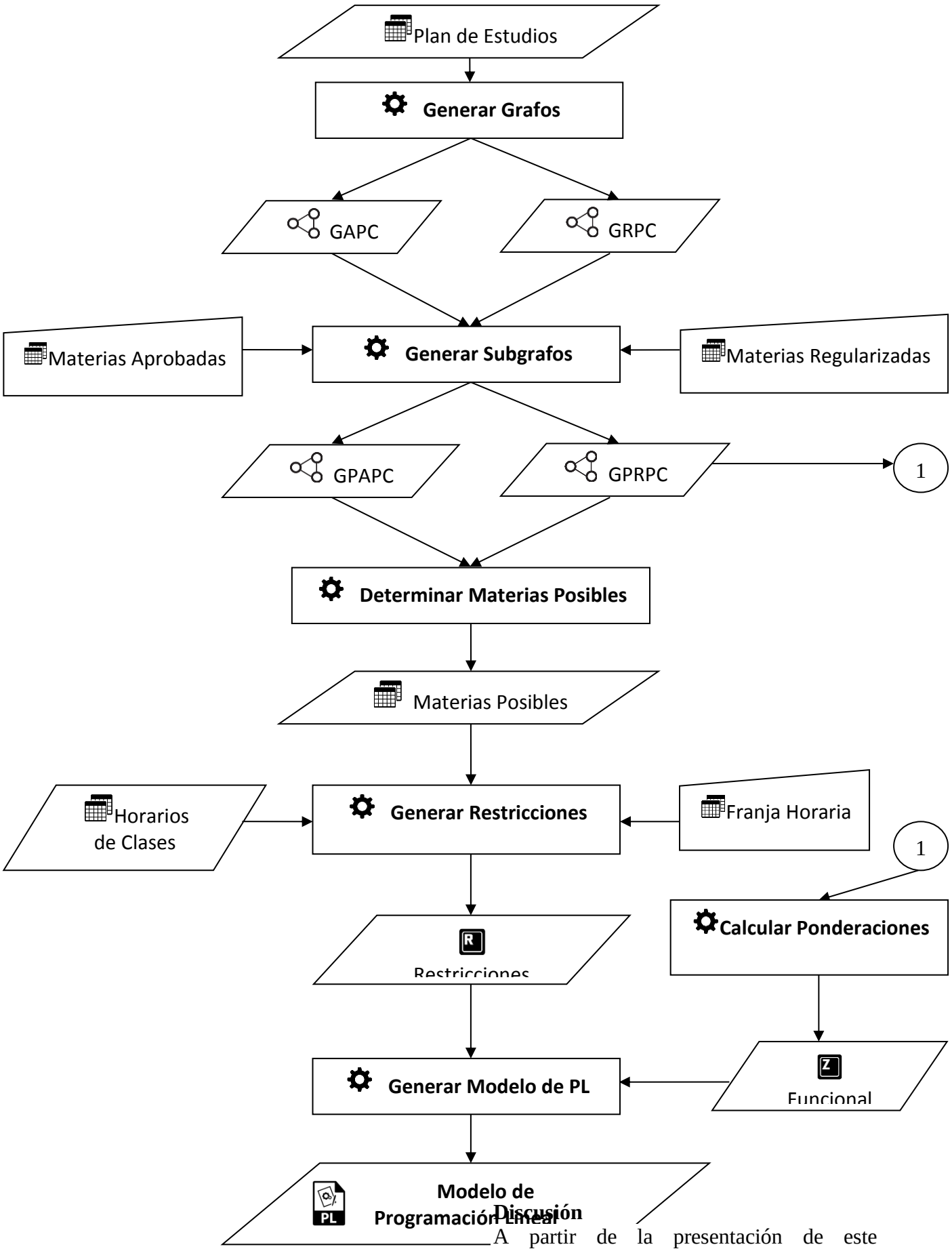


Fig 1: Representación de alto nivel del Generador de Modelos de Programación Lineal

Discusión

A partir de la presentación de este anteproyecto se abre camino al planteo de

mejoras, tanto como de diferentes aplicaciones. Se continua trabajando sobre el modelo con la incorporación de factores probabilísticos (tales como tiempos medios requeridos para el cursado, la regularización de cada asignatura, percepción de grado de dificultad de la asignatura) y la planificación de los exámenes finales, hasta alcanzar un generador de modelos genérico que funcione para cualquier plan de estudios.
Se ha analizado su aplicación y/o adaptación del modelo fuera del ámbito académico a diferentes escenarios en los que se requiera cumplir condiciones previas para la realización de actividades, además de realizar la totalidad de las mismas, es decir, recorrer la totalidad de los nodos.
Se llegará a obtener una aplicación que permita modelar escenarios que se modelan como problemas de flujo máximo o mínimo, tanto como Camino del Viajante. Además de gestión de proyectos y procesos de producción industriales.

Conclusión

Se formuló un algoritmo, el cual, teniendo un modelo del plan de estudios en donde estén representadas exigencias de cursado, los horarios de clase de la universidad, la situación académica del alumno, y la franja horaria disponible, genera un modelo de programación lineal por metas con ponderaciones, restricciones, funcional, capaz de optimizar el tiempo y beneficio de cursado.

Agradecimientos

Gracias a los integrantes de la cátedra de Investigación Operativa de la Universidad Tecnológica Nacional - Facultad Regional Tucumán por su dedicación a la enseñanza de los contenidos de la materia.
También al personal de la Secretaría Académica que posibilitó el acceso a la información de seguimiento de cohortes de carrers de ingeniería de la FRT – UTN.

Referencias

[1] “Informe de autoevaluación Informe de Autoevaluación UTN-FRT Octubre del 2010”. Universidad Tecnológica Nacional - Facultad Nacional Tucumán.
[2] “Diseño e Implementación de un Algoritmo para Optimizar el Flujo en una Red” – Tesis de Grado - Carlos Anibal Suárez Hernández.
[3] Programación Lineal – Wikipedia.
[4] Anderson D.R. Sweeney

Datos de Contacto:

Rojas, Cristina. Universidad Tecnológica Nacional – Facultad Regional Tucumán. Rivadavia 1050 - T4001JJD - San Miguel de Tucumán. *cristinarojas3@gmail.com*

Gonzalo Emmanuel Lescano. Universidad Tecnológica Nacional – Facultad Regional Tucumán. Rivadavia 1050 - T4001JJD - San Miguel de Tucumán. *onzalo@gmail.com*

Ramiro Federico Khouri. Universidad Tecnológica Nacional – Facultad Regional Tucumán. Rivadavia 1050 - T4001JJD - San Miguel de Tucumán. *ramirokhouri@gmail.com*

Gerardo Daniel Ibarra. Universidad Tecnológica Nacional – Facultad Regional Tucumán. Rivadavia 1050 - T4001JJD - San Miguel de Tucumán. *gdibarra89@gmail.com*

Robot Educativo con Visión Desarrollado como Sistema Embebido de Arquitectura Abierta

Rafael Ignacio Zurita, José Riquelme, Rodolfo del Castillo
Departamento de Ingeniería de Computadoras
Facultad de Informática
Universidad Nacional del Comahue
{rafa,jose.riquelme,rdc}@fi.uncoma.edu.ar
http://www.fi.uncoma.edu.ar

Resumen

Actualmente se están utilizando técnicas poco convencionales aplicadas a la enseñanza. En este sentido muchos estudios han sido realizados y probados sobre cómo la utilización de robots educativos incentiva el aprendizaje de la informática.

En este trabajo presentamos la arquitectura de un sistema robótico con visión, de bajo costo, construido para ser utilizado como plataforma educativa.

La adaptación de un sistema Linux embebido y la incorporación de visión permitieron ampliar el uso general de esta plataforma.

Se ha utilizado una metodología de arquitectura abierta para poder compartir el desarrollo con otras instituciones y organismos, al mismo tiempo que se tuvo en cuenta el costo asociado y la complejidad de crear una plataforma de este tipo.

En la Facultad de Informática se lo utiliza para estudiar Sistemas Embebidos, verificar técnicas de Inteligencia Artificial, incentivar el estudio de la programación, y divulgar computación a través de proyectos de extensión universitaria.

1. Introducción

Desde el año 2006, con la aparición del Instituto IPRE (Institute for Personal Robots in Education) y esfuerzos similares, se ha comenzado a trabajar con plataformas robóticas educativas en la escuela media y materias introductorias universitarias a las ciencias de la computación [1].

La utilización de estos dispositivos a través de lenguajes de programación de alto nivel motiva al estudiante, ya que los robots exponen aspectos de la computación que generalmente no son tangibles en los primeros años de las carreras universitarias. El objetivo inicial es bajar la tasa de deserción que se había observado en la carrera de grado de Ciencias de la Computación [2].

Muchos tipos de robots educativos y herramientas de programación para estos objetivos fueron desarrollados. En particular, Myro (My Robot), desarrollado en conjunto por varios organismos a través de IPRE, es uno de los frameworks más ampliamente adoptado [3]. Myro implementa un protocolo de mensajes para ser enviados (y respondidos) por un robot. Su objetivo inicial es permitir un fácil acceso a todas las capacidades del robot, y que permita su uso a través de varios lenguajes de programación [4].

Myro ha sido integrado también en Calico, un entorno de desarrollo más actual, que permite realizar muchas tareas de programación inicial en un entorno sencillo de utilizar. Aunque Calico integre a Myro, el objetivo principal de esta herramienta es el foco en el entorno de programación para estudiantes iniciales de Ciencias de la Computación [5].

Las instituciones que adopten este tipo de herramientas en educación cuentan con algunas alternativas al momento de selección del hardware, ya que Calico-Myro soporta varios tipos de robots educativos comerciales [6].

A pesar de estas alternativas, el unico tipo de reúso fácil de adoptar es el software de control, el cual es open source. Pero, las plataformas de hardware soportadas por Calico (robots) son comerciales [7-10].

En la Argentina, se han realizado algunos esfuerzos para desarrollar dispositivos robóticos para educación, ya que en nuestra región es costosa la adquisición de robots que se deben importar. Ejemplo de estos esfuerzos es el robot educativo N6, desarrollado por RobotGroup Argentina, y que es utilizado con fines educativos en la escuela media [11, 12]. N6 incorpora una placa de hardware libre llamada Arduino UNO, utilizada para el

desarrollo de prototipos de electrónica. Aun así, el resto de la arquitectura del robot no puede ser reproducida fácilmente, ya que no se ha liberado el resto de la documentación necesaria para la fabricación de los mismos.

Con estas premisas, en nuestra Facultad se diseñó un robot de bajo costo, compatible con Myro, el cual se desarrolló y liberó bajo licencias de hardware y software libre, lo cual permite su uso de manera local y su promoción a otras instituciones interesadas en fabricar y mejorar una plataforma de estas características.

Además, se decidió diseñar la plataforma no sólo para su uso en cursos iniciales de ciencias de la computación, sino que también para ser utilizada en trabajos de investigación. Por tal motivo, se optó por el agregado de visión, lo cual aportaría un elemento muy utilizado en áreas como Inteligencia Artificial y Visión por Computador.

Una limitante conocida, para este último objetivo, es que las plataformas robóticas educativas generalmente utilizan sensores básicos, que permiten ser controlados por un microcontrolador de bajos recursos. Sin embargo, para lograr la característica de visión se necesita una cámara integrada, por lo cual fue necesario agregar recursos computacionales que posibiliten la transmisión de las secuencias de video.

El propósito de este artículo es describir la arquitectura del sistema. Luego, se exponen los resultados obtenidos tanto en las experiencias reales usando esta plataforma en ciclos iniciales de las carreras de Informática, como así también de su uso como herramienta en investigación.

2. Arquitectura del Robot Educativo con Visión

2.1. Sistema de arquitectura abierta

La arquitectura abierta es un tipo de hardware o software que permite a los usuarios el acceso a todas sus partes sin restricciones propietarias [13]. Permite añadir, actualizar, modificar e intercambiar componentes. Normalmente, una arquitectura abierta publica la totalidad o parte de las especificaciones y diseños que el desarrollador o integrador quiere compartir. Los sistemas de arquitectura abierta son sistemas que proporcionan una combinación válida de interoperabilidad, portabilidad y estándares abiertos de hardware y software. También puede significar sistemas configurados que permiten el acceso sin restricciones de personas u otros equipos.

Ventajas de la arquitectura abierta [14]:

- No está sujeta a un solo tipo de hardware propietario.
- Cumplen o generan estándares y sus especificaciones son generales.
- Múltiples proveedores de hardware y software.
- Múltiples soluciones disponibles de acuerdo a necesidades específicas.
- Amplio rango de herramientas de desarrollo.

El sistema de hardware y software propuesto para el robot comparte muchas de estas características. El mismo ha sido diseñado en base a las necesidades requeridas del proyecto y para ello se han realizado diversos experimentos y pruebas tanto del software como del hardware involucrado. En el caso del hardware se fueron probando distintos módulos que hacían a la parte móvil del robot, como lo son el control de los motores, el control de la cámara y la conexión inalámbrica. En el caso del software se fueron adaptando paquetes de programas embebidos que hacían el nexo entre el usuario y el robot. Como cada parte del sistema se fue probando por separado y luego en combinación, de acuerdo a los resultados se fue creando una plataforma del tipo abierta que permitió este tipo de operaciones. Las distintas partes del sistema serán descriptas en las siguientes secciones pero cabe destacar que las mismas pueden adaptarse fácilmente o reemplazarse según las necesidades o complejidad del proyecto en base a su evolución en el futuro.

2.2. Arquitectura del Hardware

En la “Figura 1” se puede observar el esquema básico de la arquitectura del hardware del robot.

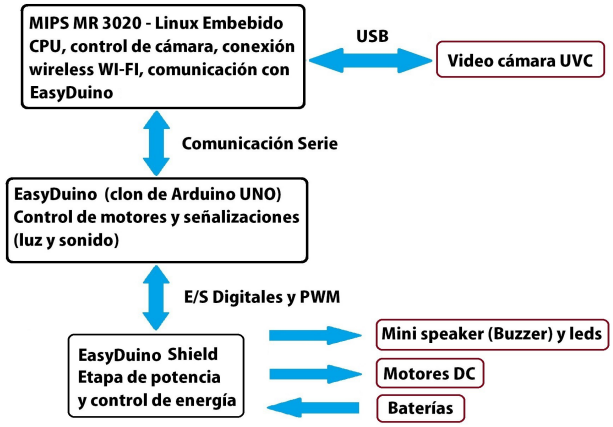


Figura 1. Arquitectura del Hardware

El sistema se compone de una placa controladora MIPS MR3020 [15], cuyas principales características se presentan en la “Figura 2”. Dicha placa actúa como interfaz entre el usuario (PC) y el robot, y provee:

Comunicación inalámbrica mediante un canal en 2,4 Ghz (Wi-Fi); control de una video cámara UVC tipo webcam; y comunicación con la placa EasyDuino.

En el proceso de selección del hardware para el sistema Linux embebido se estudiaron placas similares, realizando ingeniería inversa con algunas de ellas, lo cual permitió experimentar con Linux y realizar pruebas de laboratorio para analizar y comparar sus características [16].

* CPU MIPS 32 Bits.
* 32 Mb RAM.
* 4 Mb Flash.
* USB Host (Video Cámara).
* Serial.
* Wireless (Wi-Fi).
* Tamaño Reducido: 5,7 cm x 5,7 cm - PCB.



Figura 2. Placa MIPS MR3020

En la "Figura 3" se presentan las principales características de la placa EasyDuino. Esta placa es un clon de Arduino UNO, que cuenta con las mismas prestaciones que el original, a diferencia que la programación del mismo se realiza sólo mediante una conexión serie, que a su vez, se utiliza para la comunicación con el MIPS MR3020.

* Microcontrolador AVR ATMEGA328.
* 13 E/S Digitales.
* 6 Entradas Analógicas de 10 bits.
* Clock 20 Mhz.
* PWM - Serial - SPI.
* 32 Kb Flash - 2 Kb RAM.
* Compilador GNU GCC.

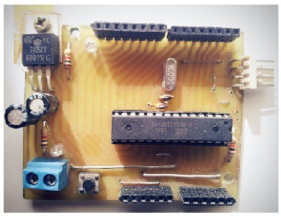


Figura 3. Placa EasyDuino

Para la etapa de potencia, es decir el control de los motores DC, se utiliza un Shield para el EasyDuino que posee un chip LD293 (puente H). Además en el Shield, se colocaron tres leds y un mini speaker (Buzzer) que permiten realizar distintas señalizaciones y, sobre la misma placa, se encuentra la conexión del pack de baterías que alimenta todo el sistema. Las características de esta interfaz se presentan en la "Figura 4".

* LD293 (Puente H).
* 2 Motores Independientes.
* Regulador y Puerto USB 5v.
* 3 Leds.
* Mini Speaker (Buzzer)

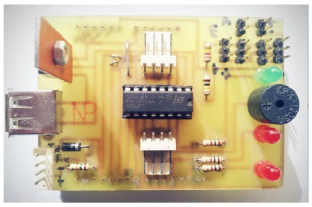


Figura 4. Placa EasyDuino Shield

En la "Figura 5" se puede observar el robot construido y la disposición de los distintos componentes de hardware mencionados.

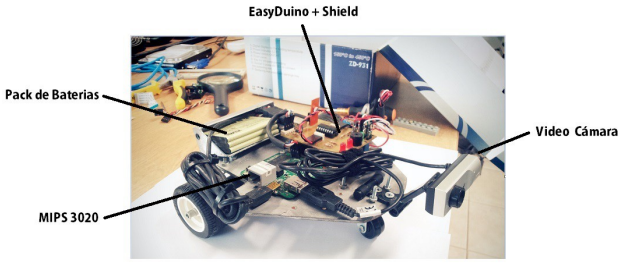


Figura 5. Robot Ensamblado

2.3. Arquitectura del Software

La arquitectura del software, como se observa en la "Figura 6", está compuesta por cinco capas claramente definidas, distribuidos en los dos componentes necesarios para la utilización de la plataforma robótica educativa. Los módulos en recuadros de línea cortada fueron desarrollados específicamente para el proyecto. Los demás componentes son piezas de software open source que fueron adaptadas para los objetivos de la plataforma robótica.

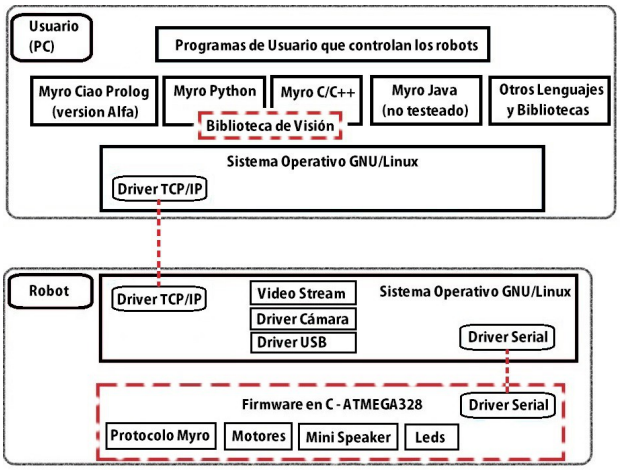


Figura 6. Arquitectura del Software

Software de Usuario

En el componente remoto (PC, portátil o dispositivo móvil) se observan los siguientes niveles de software:

- Programas de usuario.
- Biblioteca de Myro y de Visión.
- Sistema Operativo con capacidad de comunicación TCP/IP.

Los programas de usuario son aquellos que se ejecutan en una PC o portátil remota para controlar los robots (mediante la invocación a funciones de la

biblioteca de Myro). Debido a que la biblioteca de Myro está implementada en diferentes lenguajes de programación, el usuario tiene una variedad de opciones al momento de seleccionar un lenguaje para el desarrollo de programas en esta plataforma educativa.

En nuestra implementación se ha decidido utilizar comunicaciones inalámbricas (Wi-Fi), que posibilitan un mayor ancho de banda y mayor alcance; características necesarias para posibilitar la transmisión del streaming de visión y de los paquetes del protocolo Myro. Se reemplazaron los módulos de comunicaciones originales de Myro con las modificaciones necesarias para soportar conexiones TCP/IP, que permiten la comunicación por red inalámbrica con los sistemas operativos Linux embebidos en los robots.

El componente de visión se implementó como una biblioteca compartida, para ser utilizada por los programas de usuario con cualquiera de las implementaciones de Myro existentes.

Software del robot

En el robot se distinguen los siguientes niveles de software:

- Sistema Operativo Linux embebido con drivers TCP/IP, wireless, USB, UVC y serial.
- Firmware para el Atmega8 de la placa Easy-Duino con drivers serial, de motores, leds y parlante.

Además, en el firmware, se implementó la capa de software que interpreta el protocolo de Myro, para procesar los paquetes provenientes de la biblioteca de Myro del computador remoto.

El sistema Linux fue desarrollado utilizando Openwrt, el cual se compiló y configuró para que pueda ser utilizado como interfaz de comunicaciones, y para el control de la visión a través de una cámara UVC (USB Video Class). Los paquetes del protocolo Myro remotos recibidos por el robot de manera inalámbrica, son encaminados por el sistema Linux al firmware del EasyDuino, a través de una comunicación serial. De esta manera, varios clientes TCP/IP pueden hacer uso del robot de manera independiente. La comunicación inalámbrica también es utilizada para enviar las imágenes de la cámara del robot, y para la administración del sistema.

El software desarrollado para la placa EasyDuino (firmware para el microcontrolador AVR Atmega328) presenta cuatro módulos principales: el controlador de leds y parlante, el controlador de motores, el controlador serial, y el módulo que interpreta el protocolo Myro.

Cuando un paquete es recibido por la interfaz serial desde el sistema embebido Linux, se lo deriva e interpreta por el módulo que decodifica paquetes Myro. Una vez decodificado se activan los actuadores correspondientes a la orden Myro recibida (motores, leds, parlante, gestión del nombre, etc) y este mismo módulo envía por la interfaz serial un paquete de respuesta definido por el protocolo.

Utilizar un microcontrolador AVR Atmega como arquitectura para la placa EasyDuino permitió desarrollar todo el software del robot con herramientas open source, principalmente GNU GCC (The GNU Compiler Collection).

2.4. Visión

La visión es controlada por el sistema Linux en el robot, a través del driver UVC y la aplicación mjpg_streamer [17] embebida. Además, fué desarrollada una biblioteca de visión en lenguaje C, que permite solicitar y procesar imágenes de la cámara. Se integró esta biblioteca a Myro, para que los programas de usuario puedan utilizar la visión del robot para el sensado del ambiente.

Cuando un programa de usuario utiliza la API (Application Programming Interface) de visión, la biblioteca de visión solicita imágenes al robot para ser procesadas. La aplicación mjpg_streamer del robot envía entonces un streaming de video de cuatro FPS (frames per second - cuadros por segundo), con una resolución de 160x120 pixels para cada imagen.

La biblioteca de visión, recibe estas imágenes, y detecta objetos por color generando una estadística de los elementos identificados. Luego, el programa de usuario puede llamar a funciones de consulta para identificar los tamaños de los objetos encontrados y su posición frente al robot. De esta manera, y con precisión en pixels, un programa de usuario puede reconocer a "cuantos" pixels a la izquierda o derecha del centro del robot se encuentra un objeto de interés.

Utilizando estas primitivas de reconocimiento en base a tamaño del objeto (el cual da una referencia de distancia), y posición (la cual da una referencia sobre la orientación horizontal con respecto a la visión del robot) se pueden implementar programas sencillos de "búsqueda" de objetos, interactuando con el mundo real.

2.5. Utilizando Robots con Vision

El software Myro es utilizado desde la PC o netbook de los estudiantes y profesores, controlando el robot de forma remota a través de programas escritos en Python, C, C++, Java o CiaoProlog.

En la “Figura 7” se puede observar un programa de pocas sentencias en lenguaje Python. En este ejemplo, el robot girará sobre su eje hasta que, utilizando su visión, encuentre un objeto de color blanco. Como se observa, la API es fácil de utilizar, lo que permite acceder a las distintas capacidades de la plataforma (movimientos,

visión, dispositivos de salida led y parlante) de manera sencilla.

```
# python
>> from robofai import *
>> init()
>> detectar_objetos()
>> while porcentaje_color(BLANCO) < 90:
>>           izquierda(.3, 1)
>>           detectar_objetos()
>>
>> print "Encontramos el objeto color blanco. Festejamos!"
>> encender_led_izq()
>> beep()
>> apagar_led_izq()
```

Figura 7. Programa para Controlar el Robot

3. Resultados

Un primer prototipo fue utilizado en el segundo cuatrimestre de 2012, en dos charlas de divulgación de las carreras informáticas de nuestra Facultad, y en dos clases prácticas de la materia *"Introducción a la programación"* de nuestras Tecnicaturas Universitarias. Estas primeras pruebas reales tuvieron algunos inconvenientes esperados en cuanto a la electrónica, software y partes mecánicas. Pero, los docentes a cargo de esas actividades y proyecto, incentivados por la innovación en las clases, realizaron varios aportes para mejorar la robustez del software, lo que permitió que algunos problemas electrónicos fueran reconocidos por el software y se tomarán acciones automatizadas para mejorar el uso. Todos estos cambios pueden visualizarse en el histórico de versiones del repositorio del proyecto [18].

Sustanciales mejoras similares se realizaron durante 2013, con aportes en las distintas capas de software y de hardware. En ese mismo año, se los utilizó como plataforma principal en el proyecto de extensión anual *"Divulgando Computación con Robots en la Escuela Media"* que realizó encuentros con alumnos de escuela secundaria, en dónde se les enseñó a programar utilizando robots. En cada encuentro se observaron pocas fallas (de una a tres) relacionadas con la plataforma robótica, y todas estas pudieron ser solucionadas en el momento [19].

También en el 2013, y en el marco de las X Jornadas de Informática de la Universidad de la Patagonia Austral, Río Gallegos, estos robots educativos fueron utilizados como herramienta de pruebas en cuatro actividades universitarias [20]:

1. Curso de extensión "Robótica Cognitiva".
2. Curso de postgrado de Inteligencia Artificial "Robótica Cognitiva".
3. Taller de programación para alumnos universitarios.
4. Charla de divulgación de las carreras informáticas que se realizó durante las Jornadas.

Durante el 2014 se los está utilizando como plataforma principal en el proyecto de extensión anual *"Con la Escuela Media Programamos robots y Conocemos más de Computación"*, en las clases iniciales de Inteligencia Artificial y como herramienta de laboratorio en el desarrollo de varias tesis de grado relacionadas con visión e inteligencia artificial [21].



Figura 8. Proyecto de Extensión Universitaria

4. Conclusiones

En este artículo se presentó la arquitectura de un robot educativo con visión. El diseño actual ha permitido fabricar varios ejemplares de estos robots de bajo costo, lo que facilita el uso de esta plataforma en distintos lugares simultáneamente, y con distintos objetivos. El uso real también nos permite mejorar ambas arquitecturas, y se espera continuar mejorando el hardware y software a través de la retroalimentación recibida de los docentes y estudiantes, los cuales reportan problemas o solicitan características nuevas.

Nuestras experiencias han despertado también interés en los medios locales [22] y [23], los cuales ayudan a promover las carreras de informática, a través de la publicación de las experiencias y proyectos que se llevan adelante con la extensión universitaria.

A futuro, se espera finalizar el prototipo del framework Myro para el lenguaje de programación Ciao Prolog, el cual se encuentra en etapas iniciales de desarrollo. Este lenguaje es utilizado en materias dentro del área de la Inteligencia Artificial, y en las tesis de grado afines.

Además, es necesario aún realizar algunas tareas para facilitar el reúso de todo el diseño. Por ejemplo, los esquemáticos y los archivos de las placas de circuito impreso deberían ser portadas a formatos soportados por

software libre. De momento, se encuentran en formato Eagle, que aunque es un software gratis no es libre. Formatos más adecuados son los generados por Kicad o Geda.

La documentación existente, esquemáticos, diseños de las placas de circuito impreso, software de los robots, bibliotecas asociadas, ejemplos de programas de usuario, tutoriales de instalación, y toda la información relacionada al proyecto, se puede obtener desde nuestro sitio web y desde el repositorio de versiones git, lo que cumple en gran parte con los requisitos necesarios para ser definido como una arquitectura abierta [24,25].

5. Referencias

[1] Briand, L. C., Daly, J., and Wüst, J., "A unified framework for coupling measurement in objectoriented systems", *IEEE Transactions on Software Engineering*, 25, 1, January 1999, pp. 91-121.

[1] Tucker Balch, Jay Summet and Doug Blank,"Designing Personal Robots for Education: Hardware, Software, and Curriculum", *IEEE Pervasive Computing*, Vol. 7, pp. 5-9, 2008.

[2] Mikko Apiola, Matti Lattu and Tomi A.,"PasanenCreativity and intrinsic motivation in computer science education: experimenting with robots", *ITiCSE '10 Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* Pages 199-203 - ACM New York, NY, USA ©2010 - ISBN: 978-1-60558-820-9.

[3] Wiki del Institute for Personal Robots in Education: http://wiki.roboeducation.org

[4] Stefanie A. Markham and K. N. King,"Using personal robots in CS1: experiences, outcomes, and attitudinal influences", *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, pp. 204-208, (2010).

[5] Sitio Web del proyecto Calico: http://calicoproject.org/ - último acceso: 07/08/2014.

[6] Douglas Blank, Jennifer S. Kay, James B. Marshall, Keith O'Hara and Mark Russo, "Calico: a multi-programming-language, multi-context framework designed for computer science education", *Proceeding SIGCSE '12 Proceedings of the 43rd ACM technical symposium on Computer Science Education* Pages 63-68 ACM New York, NY, USA ©2012 - ISBN: 978-1-4503-1098-7.

[7] Sitio Web de Calico-Myro Robots: http://calicoproject.org/Calico_Myro#Robots - último acceso: 07/08/2014.

[8] Sitio Web del Robot Scribbler: http://wiki.roboteducation.org/Myro_Hardware - último acceso: 07/08/2014.

[9] Sitio Web del Robot The Finch: http://www.finchrobot.com/ - último acceso: 07/08/2014.

[10] Sitio Web del Robot Hummingbird: http://www.hummingbirdkit.com/ - último acceso: 07/08/2014.

[11] Sitio Web del Grupo RobotGroup: http://www.robotgroup.com.ar/ - último acceso: 07/08/2014.

[12] Díaz Javier F., Banchoff Tzancoff Claudia M., Martin Sofía y López Fernando, "Aprendiendo a Programar con Juegos y Robots", *VII Congreso de Tecnología en Educación y Educación en Tecnología. Universidad Nacional del Noroeste (Buenos Aires, Argentina)*, Junio 2012 - http://hdl.handle.net/10915/19307 - último acceso: 07/08/2014.

[13] Clifton A. Ericson, "Concise Encyclopedia of System Safety: Definition of Terms and Concepts", *Willey*, August 2011.

[14] Lic. Wilfrido Caonabo, "Arquitectura del computador", *Instituto Tecnológico del Cibao Oriental (ITECO)*, 2014

[15] Wiki de Openwrt para el hardware MR3020: http://wiki.openwrt.org/toh/tp-link/tl-mr3020

[16] Rafael Zurita, Rodolfo del Castillo, Miriam Lechner y Eduardo Grosclaude, "Reverse-Engineering a Closed-Box Hardware and Software Linux Embedded System", *Libro de Trabajos de Foro Tecnológico y Pósters - Congreso Argentino de Sistema Embebidos*, 14-16 de Agosto 2013, Buenos Aires - ISBN 978-987-9374-88-7.

[17] Sitio Web de mjpg_streamer http://sf.net/projects/mjpg-streamer/ - último acceso: 07/08/2014.

[18] Repositorio del Proyecto del Robot Educativo Frankestito: http://github.com/se_uncoma/ - último acceso: 05/08/2014.

[19] Sitio Web del proyecto de extensión año 2013 "`Divulgando Computación con Robots en la Escuela Media". Universidad Nacional del Comahue: http://divulgando.fi.uncoma.edu.ar - último acceso 05/08/2014.

[20] Boletín Oficial de Noviembre de 2013 - Gobierno de Santa Cruz: www.santacruz.gov.ar/boletin/13/noviembre13/12noviembre2013.pdf - último acceso: 05/08/2014.

[21] Sitio Web del proyecto de extensión año 2014 "`Con la Escuela Media Programamos robots y Conocemos más de Computación". Universidad Nacional del Comahue: http://robots.fi.uncoma.edu.ar - último acceso 05/08/2014.

[22] Diario La Mañana del Neuquén - Edición del 04/10/2013: http://www.lmneuquen.com.ar/noticias/2013/10/4/aprendieron-a-programar-y-crearon-a-frankestito_202090 - último acceso: 07/08/2014.

[23] Programa de Televisión "`Código Rupestre", capitulo "`El lenguaje de los Robots": http://www.codigorupestre.com.ar/?p=443 - último acceso: 07/08/2014.

[24] Wiki del Robot Educativo Fai UNCOMA: http://labti.fi.uncoma.edu.ar/trac/wiki/RoboTitos

[25] Repositorio Git del Robot Educativo Fai UNCOMA: https://github.com/zrafa/se_uncoma

Análisis y procesamiento de imágenes para la detección del contorno labial en pacientes de odontología

César Osimani
Centro de Investigación Aplicada y Desarrollo en Informática y Telecomunicaciones(CIADE-IT)
Universidad Blas Pascal
Av. Donato Álvarez 380 - Córdoba - Argentina
cosimani@ubp.edu.ar

Abstract

En este trabajo se presenta una combinación de algoritmos de análisis y procesamiento de imágenes para la detección de rasgos faciales en tiempo real que permiten obtener el contorno labial de las personas. Se propone una combinación del método Haar Cascades, con un tratamiento de imágenes en el espacio de color Lab y el algoritmo de Canny conjuntamente con un análisis de bordes en la región de los labios. Los resultados se presentan en un software desarrollado en C++ con bibliotecas de programación tales como OpenCV, OpenGL y Qt.

Con estas técnicas es posible realizar la segmentación del rostro de una persona para extraer la región interna o externa al contorno labial y con ello tener la posibilidad de desarrollar aplicaciones utilizando técnicas de Realidad Aumentada que permitan una visualización previa en tiempo real de los resultados esperados con un tratamiento odontológico.

1. Introducción

Las técnicas y algoritmos utilizados para la detección de rasgos faciales en imágenes se encuentran en pleno desarrollo y especialización, tal como lo demuestra la cantidad creciente de trabajos de investigación presentados en congresos alrededor del mundo. Esto es así porque se requieren cubrir diversas necesidades, entre ellas, la precisión en la detección y la reducción en los tiempos de procesamiento. Existen gran cantidad de algoritmos que ayudan a identificar este tipo de patrones en las imágenes y es necesario integrar varios de ellos en nuevos algoritmos si se desea identificar rasgos más particulares.

Una de las herramientas más utilizadas para el desarrollo de software para el procesamiento de imágenes es la biblioteca OpenCV. Es una biblioteca de programación que dispone de un amplio abanico de algoritmos que solucionan gran parte de las necesidades en el procesamiento de las imágenes. Sin embargo, mientras más se avanza en esto, más necesidades surgen cuando necesitamos identificar patrones más particulares. Y aquí es donde se debe experimentar con nuevas técnicas como así también realizar una combinación de otras.

En este trabajo se realiza una combinación de técnicas para la detección del contorno labial con el objetivo de ser aprovechado para el desarrollo de aplicaciones en el ámbito odontológico. Una interesante opción es el desarrollo de aplicaciones que permitan detectar el contorno labial de un paciente para luego poder superponer, con realidad aumentada, una nueva dentadura del paciente generada con modelado 3D y permitir así visualizar los resultados de un tratamiento odontológico en momentos previos al inicio de dicho tratamiento.

El lenguaje de programación utilizado para este trabajo es C++ y las bibliotecas de programación elegidas son: OpenCV (para captura de video y procesamiento de las imágenes), Qt (para la interfaz gráfica de usuario y manejo de eventos), OpenGL (para presentación de las imágenes procesadas) y la biblioteca estándar de C++.

En el presente trabajo se amplían las capacidades de algunos algoritmos para lograr detectar los rasgos faciales que son de interés aquí. En las secciones 2 y 3 se presenta la manera en que se puede reconocer un rostro y cómo se puede delimitar la región de la boca. En la sección 4 se explica la detección de los labios y cómo obtener el contorno interno del contorno labial. En las restantes secciones se presentan los resultados, referencias al código fuente, conclusiones y trabajos futuros.

2. Detección de rostros en imágenes

OpenCV agrupa interesantes soluciones para el análisis y procesamiento de imágenes. Las técnicas de reconocimiento facial se pueden agrupar como sigue:

- Métodos basados en reglas: Relaciona distintos rasgos faciales, por ejemplo, calculando las distancias entre los ojos, distancia entre la comisura derecha de los labios y la nariz, etc. [1][2].

- Métodos estadísticos: No toman información previa de las características del rostro, este método se basa en muestras de entrenamiento (imágenes con rostros e imágenes sin rostros) de las cuales se extrae información que distingue un rostro de aquello que no es un rostro [3][4]. Aquí se incluye uno de los métodos más utilizados: Cascada de clasificadores basados en características Haar [5] ('Cascada de clasificadores' de ahora en adelante).

2.1. Detección de rostros usando Cascada de clasificadores

Cascada de clasificadores es un método desarrollado por Viola & Jones [5] que realiza un entrenamiento determinando las características de un objeto basado en sumas y restas de los niveles de intensidad en una gran cantidad de imágenes positivas y negativas. De esta forma se obtiene un clasificador del objeto. Este método debe su nombre en honor al matemático húngaro Alfréd Haar y porque el resultado es consecuencia de varias etapas de clasificación en cascada.

Si bien es posible utilizar Cascada de clasificadores para detectar cualquier objeto, aquí ponemos como ejemplo la detección de rostros. El algoritmo necesita una gran cantidad de imágenes positivas (imágenes con rostro) e imágenes negativas (imágenes sin rostro) para entrenar al clasificador. Para extraer las características se utilizan Características Haar (Figura 1). Cada característica es un único valor que se obtiene restando la suma de la intensidad de píxeles en el rectángulo blanco con la suma de la intensidad de píxeles en el rectángulo negro.

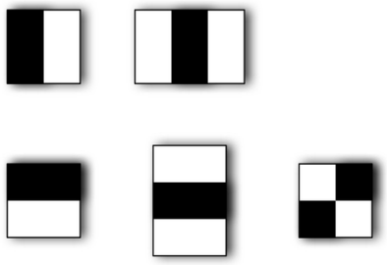


Figura 1. Características Haar.

Las ubicaciones de cada Característica Haar se utiliza para obtener las peculiaridades del objeto en cuestión. Supongamos el rostro de la Figura 2 donde la primer Característica de Haar se centra en la región de los ojos que, generalmente, son más oscuras que la zona de la nariz y las mejillas. También se observa que la segunda Característica de Haar también se centra en los ojos que son más oscuros que el puente de la nariz.



Figura 2. Clasificadores débiles.

Con esto se obtienen clasificadores débiles, a los cuales se los llama así ya que por sí solos no pueden clasificar a la imagen. Con la suma ponderada de estos clasificadores débiles se obtiene un clasificador fuerte (técnica que se conoce como AdaBoost).

La propuesta de Viola & Jones sugiere una utilización en cascada de estos clasificadores fuertes para ir descartando regiones de la imagen donde no se encuentre el objeto buscado (el rostro en este caso) que, finalmente, será detectado si supera la totalidad de las etapas.

OpenCV brinda diferentes clasificadores ya entrenados, entre ellos clasificador de rostros, clasificador de ojos, de boca, entre otros. Estos clasificadores ya entrenados están en formato de archivo XML y trabajan con imágenes en escala de grises. En este trabajo se utiliza para la detección de rostros la clase C++ CascadeClassifier que se encuentra dentro de la biblioteca OpenCV y particularmente se utiliza el método

detectMultiScale. Su utilización puede ser consultada en [6] donde está disponible el código fuente de este trabajo.

3. Detección de la región de la boca

Luego de detectar el rostro, la siguiente región de interés es la boca. Utilizar Cascada de clasificadores para detectar la boca trae aparejado problemas con falsos positivos, por tal motivo se utiliza una metodología propuesta en [7] donde se logra obtener la región de la boca tomando como referencia la posición de los ojos.

A continuación se mencionan los pasos para lograrlo:

- Detección del rostro con Cascada de clasificadores. A esta región se la denomina ROI-face. Siendo ROI la región de interés (region of interest).
- Por experimentación con rostros en posición frontal se determina que la región aproximada para los ojos comienza al 10% de los laterales de ROI-face, poseen un ancho del 40% del ancho de ROI-face, comienza al 10% de la parte superior de ROI-face y posee un alto del 50% del alto de ROI-face. Se tiene así: ROI-right-eye-approx y ROI-left-eye-approx.
- Dentro de las regiones ROI-right-eye-approx y ROI-left-eye-approx detectamos con Cascada de clasificadores la ubicación precisa del ojo derecho (ROI-right-eye) e izquierdo (ROI-left-eye) respectivamente.
- Para permitir que el rostro pueda inclinarse se calcula desde el punto central de ROI-right-eye y ROI-left-eye la inclinación del rostro (se la denomina face-rotation).
- Trazando líneas imaginarias verticales partiendo desde ROI-right-eye y ROI-left-eye, la boca se encontrará dentro de esa región. La inclinación de estas líneas verticales imaginarias son corregidas en base al valor de face-rotation.
- La región de la boca se encuentra entre las líneas verticales imaginarias y por debajo de una línea horizontal imaginaria trazada al 50% de ROI-face.
- La región de la boca es denominada ROI-mouth (Figura 3).

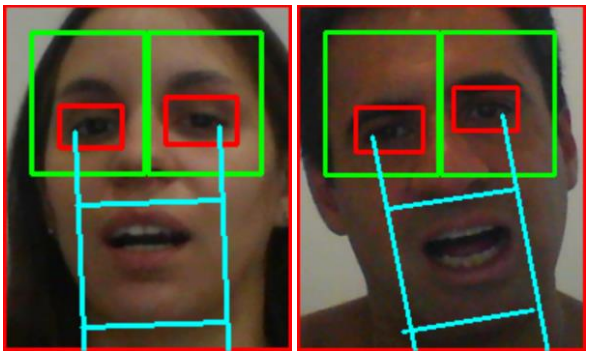


Figura 3: Detección de la región de la boca

4. Detección de labios

Luego de ubicar la región de la boca, debemos hacer un seguimiento de los labios. Primero se utiliza una detección de los labios basado en color y luego se detecta el contorno labial haciendo un análisis basado en contornos. Si bien existen diferentes métodos de segmentación basados en color (Red Exclusion, Mouth-Map, Pseudo hue), el método que mejores resultados ofrece en la detección del contorno labial es el propuesto en [6] que utiliza el componente a del espacio de color Lab.

El espacio de color Lab traduce un cambio en el valor de un color en un cambio con una muy similar importancia visual. Los tres parámetros representan la luminosidad del color (L=0 negro y L=100 indica blanca), su posición entre rojo y verde (a, valores negativos para verde y valores positivos para rojo) y su posición entre amarillo y azul (b, valores negativos para azul y valores positivos para amarillo).

Luego de transformar ROI-mouth al espacio de color Lab se puede discriminar con el componente a las regiones de color de labios con la elección de un valor umbral apropiado [6]. Este umbral será el valor medio del componente a de todos los píxeles de ROI-mouth más la varianza del mismo componente. A partir de esto generamos una imagen binaria de la siguiente manera:

- Valor medio del componente $a \rightarrow a_{means}$
- Varianza del componente $a \rightarrow a_{std}$
- El umbral es: $\theta = a_{means} + a_{std}$
- Siendo $a(i,j)$ el valor del componente a de cada píxel de ROI-mouth, el valor de la imagen binaria resultante será:

$$a(i,j) > \theta \rightarrow \text{Color blanco}$$

y

$$a(i,j) \leq \theta \rightarrow \text{Color negro}$$



Figura 4: Imagen binaria obtenida a partir del análisis en el espacio de color Lab.

4.1. Detección de bordes

Una vez obtenida la imagen binaria (Figura 4) que representa la ubicación de los labios, aplicamos el algoritmo de Canny [8] para detectar los bordes en esta imagen binaria. Este algoritmo está considerado como uno de los mejores métodos para la detección de contornos mediante el empleo de máscaras de convolución y basado en el gradiente. Los puntos de contorno son zonas de píxeles en las que existe un cambio brusco de nivel de intensidad. El gradiente toma el valor cero en todas las regiones sin variación de intensidad y tiene un valor constante donde existe transición de intensidad.

OpenCV trae incorporada una función para detección de bordes utilizando el algoritmo de Canny. En el código fuente [6] de este trabajo se puede ver su utilización. Primero se realiza la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente.

Para obtener los bordes (Figura 5) aplicamos un filtro para calcular el gradiente a través del operador Sobel con las siguientes máscaras de convolución:

$$G_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$G_y = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Como en general la matriz de coeficientes es mucho menor que las dimensiones de la imagen, por ejemplo, una matriz de 3x3 frente a 640x480. El proceso de convolución aplica una ventana a la imagen de las dimensiones de la matriz de coeficientes del filtro, la cual es centrada en cada uno de los píxeles teniendo en cuenta también los píxeles que lo rodean.

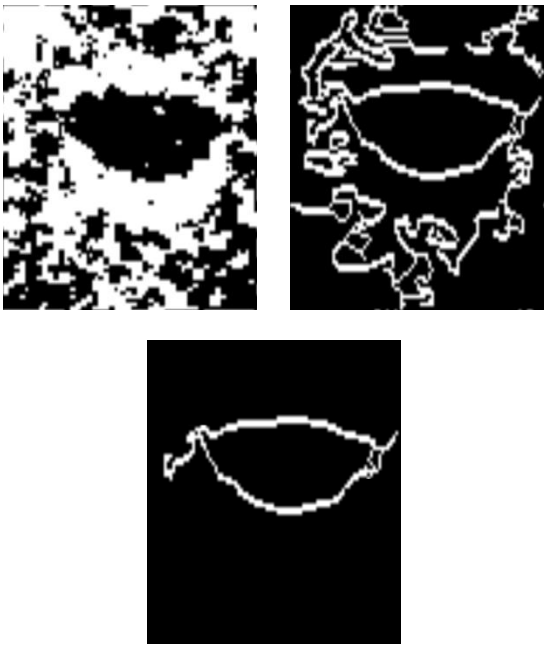


Figura 5: Bordes detectados con algoritmo de Canny y procesamientos de contornos.

Para aplicar las máscaras de convolución a la imagen original y así obtener dos imágenes que representan para cada punto las aproximaciones horizontal y vertical de las derivadas de intensidades, realizamos lo siguiente:

$$G(x,y) = \sum_{i=0}^{2}\sum_{j=0}^{2} f(x+i, y+j)\, g(i,j)$$

donde:

- Imagen original = $f(x,y)$
- Máscara de convolución 3x3 = $g(i,j)$
- Imagen filtrada = $G(x,y)$

En cada punto (x,y) de la imagen, los resultados de las aproximaciones de los gradientes horizontal y vertical pueden ser combinados para obtener la magnitud del gradiente, mediante:

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Ahora se hace uso de un umbral superior y otro inferior para determinar si un píxel es borde o no, de la siguiente forma:

- Si el gradiente del pixel es mayor que el umbral superior, el pixel se acepta como un borde.
- Si el gradiente es menor que el umbral inferior, el píxel se rechaza como borde.
- Si el gradiente del píxel se encuentra entre los dos umbrales, entonces será aceptada como borde sólo si está conectado a un píxel que es mayor que el umbral superior.

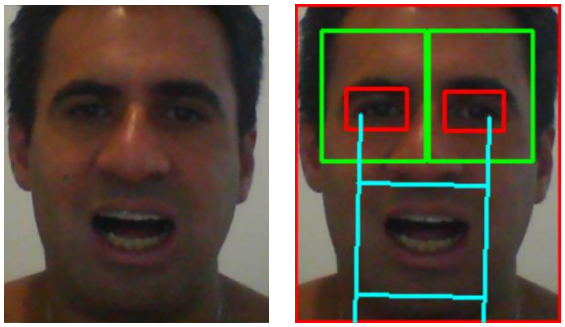
5. Experimentos con el software desarrollado

El software[1] desarrollado para realizar los experimentos obtienen las imágenes de una cámara de video y realizan su procesamiento en tiempo real. De las observaciones es conveniente destacar las siguientes particularidades:

- La elección aproximada de ROI-right-eye-approx y ROI-left-eye-approx basada en reglas disminuye el tiempo de procesamiento y aumenta la exactitud en la detección.

- La elección del espacio de color Lab y el uso de la fórmula $\theta = a_{means} + a_{std}$ para la detección de los labios no sólo es apropiado para realizar la detección de los labios sino también para que el procesamiento sea independiente de la iluminación ambiente y del color de piel que pueden tener distintas personas.

A continuación una muestra de cada etapa del análisis y procesamiento de las imágenes:



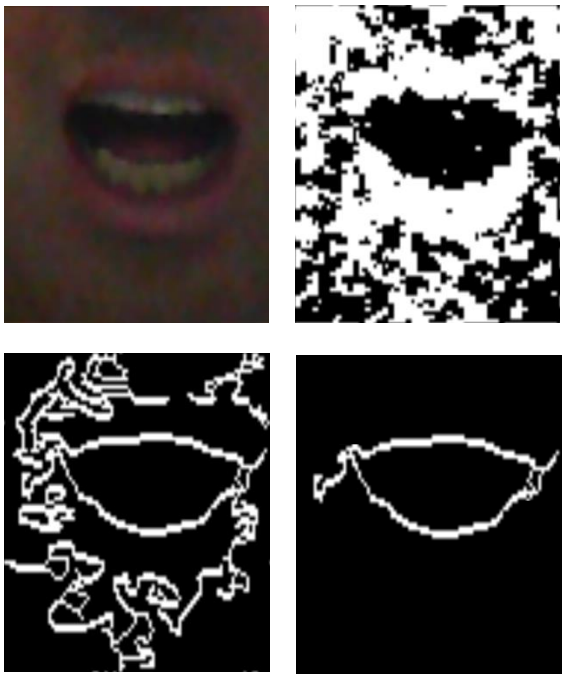


Figura 6: Todas las etapas del procesamiento.

6. Conclusiones y futuros trabajos

En este trabajo se hacen muchos esfuerzos para lograr un sistema en tiempo real y, en su búsqueda, es muy evidente la reducción del tiempo de procesamiento de las imágenes cuando se utiliza una selección basada en reglas para ignorar lo más pronto posible aquellas regiones de las imágenes donde se sabe no se encontrarán las características buscadas.

Interesantes ventajas, a la hora de detectar rasgos faciales, se logran trabajando en el espacio de color Lab ya que con los valores del componente a se pueden distinguir cómodamente las diferencias de matices en el rostro. Esto es así porque el componente a contiene información sobre los matices del color de la piel, como así también los cambios de valores de este componente a se traducen en un cambio proporcional de la percepción del color por el ojo humano.

El análisis de bordes es un trabajo arduo y brinda resultados notables luego de realizar una buena cantidad de experimentos para seleccionar adecuadamente los umbrales para el algoritmo de Canny, medir la longitud de estos bordes y definir nuevos umbrales para excluir algunas longitudes, detectar regiones cerradas, entre otras características.

Con las técnicas utilizadas en este trabajo se logra obtener el contorno labial y con ello se invita a continuar

[1] El código fuente se encuentra disponible en [6].

investigaciones y desarrollos dentro de esta temática. Se prevén trabajos futuros que utilicen esta detección del contorno labial para extraer la región interna y poder reemplazarla, con técnicas de Realidad Aumentada, por una dentadura modelada en 3D con la idea que el paciente pueda observar por adelantado los resultados del tratamiento odontológico al cual se va a someter. Para lograr este objetivo final también es necesario realizar experiencias en la detección de la alineación en la que se encuentra un rostro con respecto a la cámara de video. Para detectar esta alineación se pueden utilizar algoritmos tales como SIFT (Scale-invariant feature transform) que detectan ciertos puntos en una imagen que tienen una importante cantidad de información a su alrededor y los hace aptos para poder realizarles un seguimiento cuadro por cuadro. Debido a que estos puntos no se pierden de vista cuando se producen cambios de escala o rotaciones de los objetos visibles, lo hace un algoritmo adecuado para detectar la rotación de un rostro.

7. Agradecimientos

A los alumnos de la carrera de Ingeniería Informática de la UBP Ramiro Laso, Emiliano Kohmann e Ignacio Rocca por colaborar en la programación y a Néstor Pisciotta, quien está a cargo de la Secretaría de Investigación de la UBP, por suministrar el equipamiento necesario para los experimentos.

8. Referencias

[1] Yang, G., Huang, T. S., *Human face detection in complex background*, Pattern recognition, 1994.

[2] Yang, M., Kriegman, D., Ahuja, N, "Detecting faces in images: a Survey", in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002.

[3] Rowley, H.A., Baluja, S., Kanade, T., "Neural network-based face detection", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1998.

[4] Turk, M.A., Pentland, A.P., "Face recognition using eigenfaces", in *Computer Vision and Pattern Recognition,IEEE Computer Society Conference on*, 1991.

[5] Viola, P., Jones, M., "Rapid object detection using a boosted cascade of simple features, computer vision and patterns recognition", *Computer Vision and Pattern Recognition*, 2001.

[6] *Source Code: [Online]*, Available: https://github.com/cosimani/conaiisi-2014

[7] WenJuan, Y., YaLing, L., "A real-time lip localization and tracking for lip reading", in *Advanced Computer Theory and Engineering (ICACTE) 3rd International Conference on*, 2010.

[8] Canny, J., A "Computational Approach to Edge Detection", in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1986.

Experiencia sobre la Incorporación de Herramientas de Software Libre a la Extracción de Conocimiento en Datos Astronómicos

Raúl Oscar Klenzi[1], Alejandra Malberti[1]Georgina Codwell[1-2]
1 Departamento de Informática Facultad de Ciencias Exactas Físicas y Naturales
Universidad Nacional de San Juan.
2 *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*
Ignacio de la Roza y Meglioli – Complejo Universitario Islas Malvinas (CUIM)
rauloscarklenzi,amalberti,georginacoldwell@gmail.com

Resumen

En el marco del proyecto "Extracción de Conocimiento de Datos Masivos" Res. 018-14 CS CICITCA- Universidad Nacional de San Juan CICITCA-UNSJ se propone la aplicación de herramientas de software libre del área de minería de datos y aprendizaje de máquina a datos masivos extraídos del área de conocimiento Astronomía. Las herramientas a utilizar son las últimas versiones de RapidMiner(RM) y Orange que poseen capacidades de ejecutar algoritmos de aprendizaje de máquina. En particular para el presente trabajo, la extracción de conocimiento se realiza mediante la aplicación de RM en tareas de preprocesamiento y de Orange en tareas de segmentación y clasificación. Tras las tareas de preprocesamiento que permitan clarificar, comprender e interpretar los datos, se analiza el desempeño de algoritmos de segmentación y de clasificación, con el propósito de representar adecuadamente la información y conocimiento contenido en los mismos, desde la perspectiva del usuario experto.

1. Introducción

Actualmente se dice que se vive "la era de la información", sin embargo se puede afirmar que en realidad se está viviendo "la era de los datos". Terabytes o petabytes de datos se vierten, cada día, en redes de computadoras domésticas (Intranets), la World Wide Web (WWW-Internet), y en diferentes dispositivos de almacenamiento de datos masivos provenientes, entre otras, de aplicaciones de negocio, de redes sociales, y de experiencias científicas de las más diversas áreas de conocimiento. [1]

Este fenómeno de grandes cantidades de datos (Datos Masivos) ha aparecido en áreas como la meteorología, la investigación genómica, datos biológicos, **astronomía**, búsqueda en Internet, finanzas, y muchas más. El término grandes volúmenes de datos se utiliza para describir conjuntos de datos a partir de unos cientos de gigabytes a decenas de terabytes y más [2]. La Astronomía se ha beneficiado del explosivo avance de la tecnología computacional y de la información. La era de la astronomía digital es un ambiente tan extremadamente rico en datos que desafía los enfoques y metodologías tradicionales para la extracción de nuevos conocimientos a partir de los datos. El proceso de descubrimiento científico es cada vez más dependiente de la capacidad de analizar grandes cantidades de datos complejos generados por instrumentos científicos y simulaciones. Este análisis, que rápidamente se convierte en un cuello de botella, motiva necesidades de innovación, fomento y colaboración entre la comunidad científica y la tecnología informática [3].

En este sentido el campo de la minería de datos, que consiste en la extracción de conocimiento no trivial en bases de datos que favorece y apoya la toma de decisiones, permite importantes avances. De igual manera, la potenciación del procesamiento paralelo asociado a las unidades de procesamiento central (Central Process Unit - CPU) multi-núcleos con las que cuentan las unidades de computo actuales, es una alternativa de solución que la tecnología actual presenta.

Durante la próxima década, los volúmenes de datos entrarán al dominio petabyte, y es de destacar que los estudiantes de Astronomía y Física no están lo suficientemente formados y entrenados para manejar conjuntos de datos tan voluminosos y complejos [4]

En el campo de la investigación y enseñanza de la Astronomía, en diciembre de 2013 Chile ha puesto a

consideración la [1] Plataforma Astro-Informática Chilena unida al Observatorio Virtual internacional. Este proyecto busca administrar y analizar los cerca de 250 terabytes de datos anuales que generará el radiotelescopio Atacama Large Millimeter/submillimeter Array (ALMA), que se incorporó a la Alianza Internacional de Observatorios Virtuales, consolidando así una iniciativa clave en el aporte de Chile al contexto internacional de la astro-informática.

La Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de San Juan (FCEFN-UNSJ) cuenta con las carreras de Licenciatura en Astronomía, Licenciatura en Ciencias de la Computación y Licenciatura en Sistemas de Información. Los autores de este trabajo, docentes e investigadores de los Departamentos de Informática y de Astronomía de la mencionada facultad, han iniciado un proceso de interacción a partir de la conformación de un grupo de investigación, con el propósito de incrementar la experiencia en áreas de conocimiento como es la del aprendizaje de máquina y la minería de datos paralela y distribuida, aplicadas a datos astronómicos.

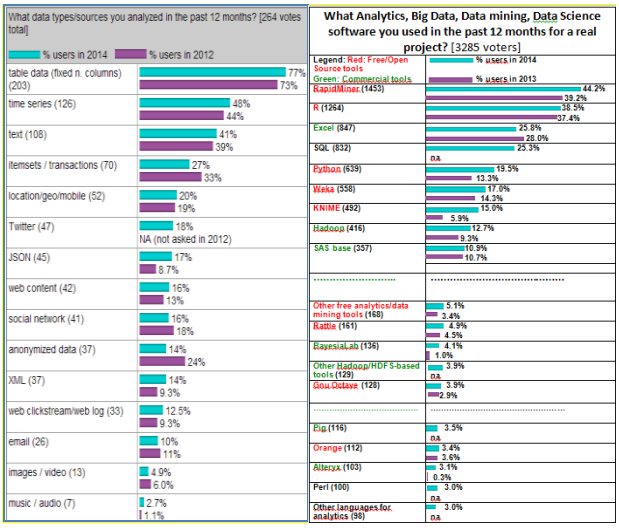
Un grupo de investigadores de CONICET, desde las tareas enmarcadas en los proyecto PICT 2010 Bicentenario N°0680 y PIP 2012-2014 GI (Código: 11220110100298) se encuentran trabajando principalmente en la realización de análisis estadísticos utilizando catálogos públicos de galaxias, cúmulos de galaxias, quasares, galaxias con núcleos activos, etc. Para ello recopilan permanentemente grandes cantidades de datos de objetos extragalácticos con el propósito de obtener resultados claros y confiables en el proceso de investigación. Un ejemplo de ello es el uso del séptimo relevamiento fotométrico del catálogo Sloan Digital SkySurvey (SDSS-DR7). Este catálogo consta de 357 millones de objetos para los cuales una gran cantidad de parámetros fueron medidos conformando un total de 18 TB.

Actualmente la aplicación en estudios astronómicos de aprendizaje de máquina, o de metodologias relacionadas, es bastante escaza. Las redes neuronales se han utilizado con éxito en la discriminación estrella / galaxia, clasificación de morfologías de galaxias, espectros de galaxias, y espectros estelares. Clasificadores bayesianos se han utilizado para la clasificación de poblaciones estelares cinemáticas, quasars y espectros estelares y morfologías de galaxias. En la selección fotométrica de quásares se ha empleado el algoritmo k-means. Aprendizaje de máquina también se utiliza, a menudo, en física solar y del espacio [4].

Sin embargo la gran mayoría de los astrónomos son sólo vagamente conscientes de los avances recientes en técnicas de aprendizaje automático, y no las incorpora en sus programas de investigación.

En este sentido existen numerosas herramientas de software que permiten construir modularmente las diferentes etapas que constituyen la extracción del conocimiento subyacente en los datos. Según los resultados de la encuesta presentados en Tabla1[2], los reservorios de datos más usados son las tablas en tanto también, desde la misma encuesta, surgen las herramientas de software mayormente utilizadas en los últimos dos años en aplicaciones reales. Se puede observar que aparecen muchas herramientas de software libre del tipo RapidMiner, Weka, R, Orange, Knime entre otras.

Tabla 1



Así, entre las diez herramientas más utilizadas se encuentran:

1. RapidMiner, 44.2% share (39.2% in 2013)
2. R, 38.5% (37.4% in 2013)
3. Excel, 25.8% (28.0% in 2013)
4. SQL, 25.3% (na in 2013)
5. Python, 19.5% (13.3% in 2013)
6. Weka, 17.0% (14.3% in 2013)
7. KNIME, 15.0% (5.9% in 2013)
8. Hadoop, 12.7% (9.3% in 2013)
9. SAS base, 10.9% (10.7% in 2013)
10. Microsoft SQL Server, 10.5% (7.0% in 2013)

Para la presente propuesta se trabaja con RapidMiner 5.3.15 y Orange Canvas. La primera es una herramienta que dispone de diferentes módulos que permiten realizar

[1] http://www.almaobservatory.org/es/sala-de-prensa/anuncios-eventos/658-chilean-astroinformatics-platform-joins-international-virtual-observatory

[2] www.Kdnuggets.com/polls/2014/data_types_sources_analyzed.html (31/05/2014)

todas las etapas de extracción de conocimiento en datos con una interfaz visual de calidad pero que muchas veces torna lentas las aplicaciones sobre problemas reales de magnitud. Orange también con una adecuada interfaz visual, mejora la performance sobre problemas reales y está desarrollado en el lenguaje de programación Python versión 2.7 siendo este muy utilizado en aplicaciones de minería de datos en astronomía.[4]

2. Minería de datos y Astronomía

Desde siempre la curiosidad ha llevado a buscar nuevo conocimiento en los cielos. En particular la Astronomía ha combinado idealización matemática, ingenio tecnológico, y recolección de datos, con procedimientos de búsqueda en datos reunidos a lo largo del tiempo. Habitualmente los astrónomos realizan inferencias acerca de hipótesis científicas basadas en datos históricos recolectados con diferentes propósitos, o inferencias a nuevas hipótesis motivadas por aspectos de los mismos datos bajo investigación. En algunos casos se han aplicado procedimientos algorítmicos para identificar grupos físicamente ligados de galaxias a partir de su distribución bidimensional, sin información previa acerca de las ubicaciones de grupos o poblaciones. Turner y Gott señalaron círculos, con un radio elegido, alrededor de cada galaxia. Cuando los círculos de galaxias próximas se superponen, se consideran miembros de un solo grupo. Este enfoque algorítmico, que ha sido utilizado en cientos de documentos durante los años 1990 y 2000, permite que los datos sean usados para determinar si un grupo es pequeño, con pocos miembros, o grande y populoso. [3]

En los últimos tiempos, los desarrollos tecnológicos han propiciado el aumento en la cantidad de datos recolectados. Las bases de datos crecen en el orden de los tera o petabytes por año, por día, y así sucesivamente, lo que ha obligado a tener en cuenta su complejidad y estrategias para su análisis. En lo que refiere a la complejidad de los datos, que se nota especialmente en la era de la Astronomía digital, el progreso se manifiesta, entre otros, en la aplicación de sofisticados procesos de minería de datos.

Los astrónomos son, y siempre han sido, mineros de datos, cuya tarea no es sólo recopilar datos, sino también encontrar una explicación de un conjunto de fenómenos, una explicación que esté en consonancia con los conocimientos físicos y en el mejor de los casos, al menos aproximadamente verdadera. Un ejemplo de minería de datos, sin que fuese llamada así, fue la explotación que realizó Kepler de los datos dejados por Tycho Brahe; Kepler construyó sus leyes de trabajo sobre los datos de Brahe. [3]

La exploración de los datos es un proceso que consta de varias etapas en las que se trabaja con una metodología estructurada para descubrir y evaluar los problemas apropiados, definir soluciones y estrategias de implementación, y producir resultados medibles.

La Minería de Datos es el análisis de datos observacionales, a menudo grandes cantidades, con el propósito de encontrar relaciones insospechadas y de resumir los datos en nuevas formas que sean comprensibles y útiles para el titular de los datos. [5]

La definición menciona grandes conjuntos de datos, lo que conlleva nuevos aspectos a considerar. Por ejemplo resumir o comprimir un conjunto de datos muy grande de tal manera que el resultado sea más comprensible, sin intención de generalización o, por otro lado, procesar el conjunto completo de datos recolectados, lo que requiere de herramientas adecuadas para ese fin.

La minería de datos está situada en un contexto más amplio, que es el de descubrimiento de conocimiento en bases de datos (Knowledge Discovery Databases- KDD), término originado en el campo de la Inteligencia Artificial (IA). El proceso KDD consta de varias etapas: pre-procesamiento de datos transformándolos en caso necesario, aplicación de minería de datos para extraer patrones y relaciones, y luego interpretación y evaluación de las estructuras descubiertas. Sin embargo los límites precisos de la parte del proceso correspondiente a minería de datos no son estrictos ya que, por ejemplo, la transformación de datos suele ser considerada una parte intrínseca de la minería de datos. [5]

Mientras que en Estadística la recolección de datos generalmente se realiza con el propósito de responder preguntas previamente formuladas, en minería de datos se usan datos disponibles ya recolectados. Por esta razón en minería de datos se refiere a menudo al análisis de datos "secundarios". Esto lleva a que los proyectos de análisis de datos necesiten una gran cantidad de esfuerzo para el preprocesamiento y limpieza de datos. Se suele decir que el 80% de la obra consiste en preprocesamiento y sólo el 20% es el modelado y evaluación.

Los datos pueden ser de diferente naturaleza, por ejemplo valores continuos o valores categóricos. También puede ocurrir que haya valores faltantes o aun valores fuera de rangos válidos, situaciones que deben ser contempladas y resueltas en la etapa de preprocesamiento, siempre con la intervención del experto en el dominio de aplicación. El valor de la minería de datos depende de la calidad de los datos tratados y de los métodos aplicados a ellos.

Entre los problemas que presenta el análisis de datos está el encontrar una forma correcta de calcular una relación entre las variables de una muestra, y encontrar métodos para probar hipótesis con los datos de la misma.

Es cierto también que las grandes cantidades de datos observacionales, recolectadas en Astronomía, requieren frecuentemente de la aplicación de estrategias de procesamiento paralelo a conjuntos de datos masivos.

En el desarrollo de la ciencia se encuentra inmerso el proceso de clasificar, esto es distribuir elementos conformando conjuntos, dando origen a diversas taxonomías. Si bien este proceso ocurre en diferentes áreas de conocimiento, ha desempeñado un papel de enorme importancia en el desarrollo de la Astronomía. Una gran cantidad de taxonomías, o clasificaciones, fueron desarrolladas utilizando métodos heurísticos, basados en evaluaciones cualitativas y subjetivas de propiedades espaciales, temporales o espectrales. La Clasificación es un tipo de tarea que se realiza en minería de datos en la que cada instancia, o registro de la base de datos, pertenece a una clase, lo cual se indica mediante el valor de un atributo llamado atributo clase. Este atributo particular puede tomar diferentes valores discretos. El objetivo de la clasificación en minería de datos es hallar un modelo para el atributo clase, como una función de los valores de otros atributos.

La Segmentación, Agrupamiento, o Clustering, es otra tarea que se realiza en minería de datos. Esta tarea descriptiva consiste en encontrar grupos a partir de los datos; los datos son reunidos basándose en la similitud entre los elementos de un grupo, minimizando la similitud entre los distintos grupos. Los objetos de un mismo grupo son similares entre si y a la vez diferentes de los objetos de otros grupos. [6]

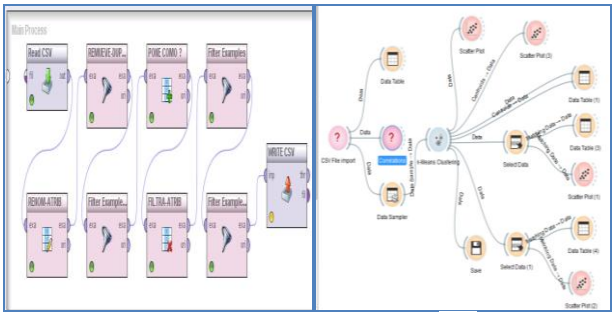
En general se espera encontrar en los datos relaciones y estructuras novedosas y comprensibles respecto al conocimiento previo del usuario experto. En particular los astrónomos suelen estudiar los problemas disponiendo de considerable conocimiento previo. Se han estudiado intensamente muestras limitadas de objetos brillantes, cercanos, que pueden servir entonces como "conjuntos de entrenamiento" para la clasificación de muestras más grandes y menos caracterizadas.

Existen variadas herramientas de software libre que poseen algoritmos que realizan tareas de clasificación y de segmentación tales como RM y, Orange. Asimismo estas herramientas soportan el procesamiento de datos masivos, incorporando, entre otros aspectos, el procesamiento paralelo de los datos tratados.

3. Aplicación

Los datos considerados para la presente propuesta consisten en una tabla en la que se representan más de 300.000 registros compuestos por 12 atributos. Los miemos, en su mayoría de tipo numérico, explicitan la siguiente información: Coordenadas ecuatoriales que indican la posición de los objetos extragalácticos en la esfera celeste, redshift como estimador de la distancia, magnitudes absolutas Petrosian (Petrosian 1976) en los filtros u, g, y r, contenido de masa estelar de las galaxias medida en unidades de masa solar –variable: **Masterisco-**, tasa específica de formación estelar anual de las galaxias -variable: **SFRsobreMasterisco-**, índice de sersic (Sersic 1963 -variable: **n_sersic-**), el indicador de edad de la galaxia, -variable: **Dn4000-** y colores dados por la diferencia de magnitudes, Mu-Mr -variable: **ColorUmenosR-** y Mg-Mr. -Variable: **ColorGmenosR-**.

La aplicación se ha llevado adelante mediante la utilización de RM 5.3.15 GNU-GPL dic. 2013 y Orange Canvas basado en Python 2.7. Por medio de la primer herramienta se realizó un preprocesamiento inicial que involucra tareas de renombrar, ajustar y filtrar tanto registros con valores fuera de rango como atributos que no habrán de considerarse. El archivo con 374669 registros se almacena, en formato .csv, para su posterior tratamiento en Orange. En esta herramienta se realizan tareas de segmentación y clasificación con árboles de decisión.

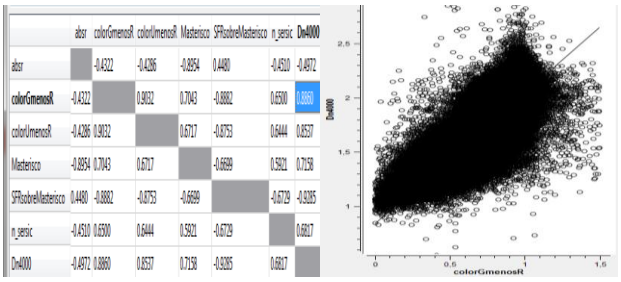


(a) (b)

Figura 1

En el entorno RM se aprecia la configuración de módulos que realizan el preprocesamiento -Figura 1a-. En este caso: se eliminaron atributos considerados en esta oportunidad no necesarios por el astrónomo experto, se establecieron rangos de valores permitidos para algunos atributos, se reemplazaron por "?" valores faltantes con el objetivo de compatibilizar formatos de datos ausentes con otras herramientas y finalmente, los registros resultantes se guardaron en formato CSV.

Con la aplicación ORANGE CANVAS se realizaron los restantes pasos de extracción de información y conocimiento. Se puede apreciar en la Figura 1(b) el esquema de módulos de las distintas tareas llevadas a cabo en este entorno de software.



(a) (b)

Figura 2

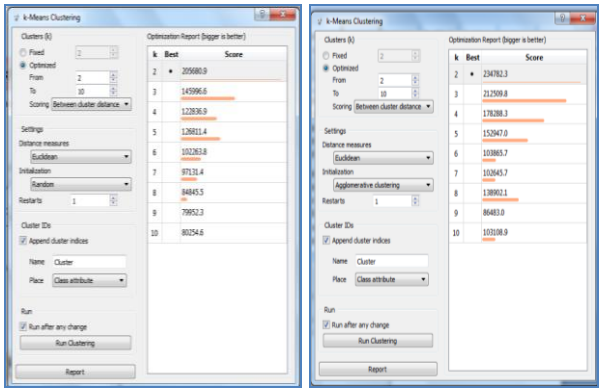
Tras el cálculo de la matriz de correlación - Figura 2(a) - se puede observar la relación directa (0,8860) que se presenta entre las variables colorGmenosR (colores dados por la diferencia de magnitudes Mg-Mr) y Dn4000 (indicador de edad de la galaxia) evidenciada en la gráfica de la Figura 2(b), incluso con una aproximación de los puntos, por la recta con menor error cuadrático medio.

Esta correlación observada indica una importante y conocida propiedad evolutiva de las galaxias. Las galaxias, a medida que van evolucionando, sufren diferentes procesos que modifican su color y por lo tanto el de las estrellas que las componen. El parámetro Dn4000, indicador de edad de las galaxias, toma valores mayores hacia colores más rojos, correspondiente con mayores valores del colorGmenosR, es decir que las galaxias más viejas son también las más enrojecidas.

3.1. Tarea de segmentación

Se utilizó el algoritmo de segmentación Kmeans, que es un algoritmo de clustering partitional muy difundido por su simplicidad y eficiencia. Dado un conjunto de puntos de datos y el número requerido de grupos k (k es especificado por el usuario), este algoritmo iterativamente divide los datos en k grupos basados en una función de distancia. La complejidad temporal de Kmeans es O(tkn) donde n es el número de puntos de datos (registros), k es el número de segmentos y t es el número de iteraciones [7]. Posteriormente y basado también en su simpleza y didáctica se aplicó un algoritmo de clasificación para reconocer qué atributos permiten la caracterización de los segmentos encontrados.

A los efectos de determinar cuál es la cantidad de segmentos que permite una mejor partición según la distribución de los datos, se utiliza un módulo provisto por Orange. En este caso se consideró la tarea de optimizar la cantidad de segmentos desde k= 2 a k=10 evaluándose en cada caso el valor de la distancia entre cluster. El mayor valor de este parámetro permite alcanzar una mayor y mejor separación entre datos. Se puede apreciar que la herramienta, con dos formas de inicialización distintas, propone k=2 como la cantidad de segmentos más adecuada -Figura 3 (a) y (b)-.



(a) (b)

Figura 3

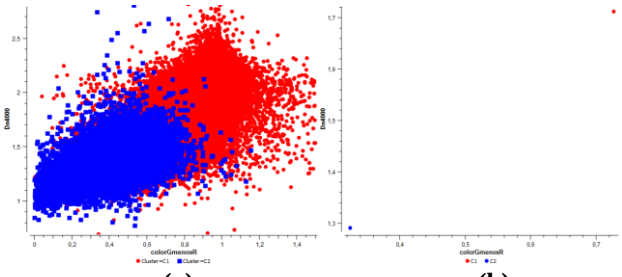
De las dos alternativas de inicialización utilizadas, la basada en un cluster aglomerativo, es la que mejor resultados brinda atento a la separación de los datos asociados a cada segmento.

Tabla 2

table (Data) table (Centroids)

	absr	colorGmenosR	colorUmenosR	Masterisco	FRsobreMasterisc	n_sersic	Dn4000	Cluster
1	-20.42373	0.726565	2.181351423263...	10.3364	-10.75562	2.6968	1.7127	C1
2	-18.43500	0.324155	1.261958003044...	8.9716	-9.48181	1.1039	1.2914	C2

La -Tabla 2- muestra los valores de los centroides representativos de cada segmento o cluster



(a) **(b)**

Figura 4

En la Figura 4 se presenta el resultado de la segmentación (Segmentos encontrados (4a) y Centroides asociados (4b)) entre los parámetros Dn4000 y colorGmenosR. Allí se observa una clara división de poblaciones, conocida como bimodalidad. Esta es una característica notoria de la población general de galaxias que ha sido adecuadamente segmentada, como puede observarse de los valores de los centroides.

3.2. Tarea de Clasificación

Tras la segmentación, y siempre utilizando Orange, se planteó realizar una tarea de clasificación. Se utilizó el algoritmo de clasificación en árboles que, basado en la ganancia de información relativa, permite clasificar el atributo clase, en este caso el cluster encontrado en el paso anterior, según los restantes atributos.

El diagrama en bloques del proceso realizado se presenta en la Figura 5.

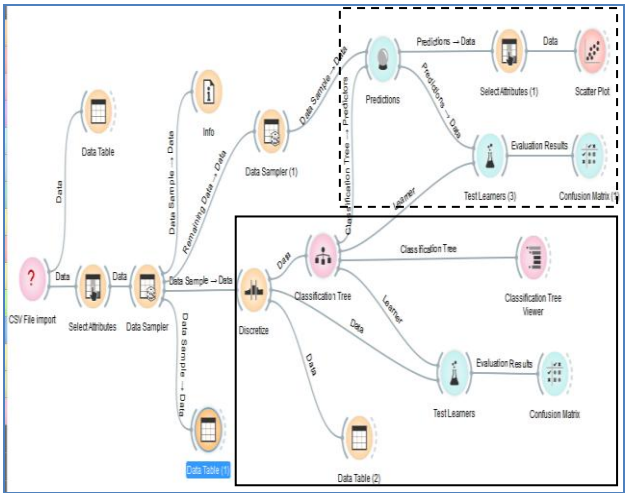


Figura 5

Las capacidades de las herramientas se ven limitadas al momento de clasificar la totalidad de los registros asociados a datos residentes en memoria principal. Por ello se modelaron los datos desde una muestra extraída de aquellos 374669 registros -Figura 5: rectángulo con línea llena-. En este caso particular se utilizaron tan sólo 15000 registros del total, evaluada la estrategia de clasificación mediante técnicas de crossvalidación que muestran la bondad del algoritmo en su matriz de confusión. -Figura 6-.

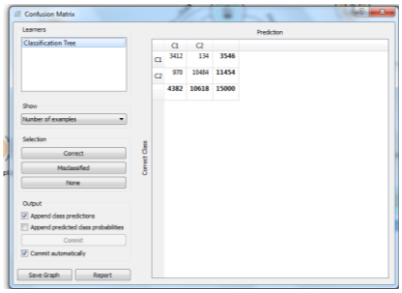


Figura 6

Posteriormente, y para otros 10000 registros tomados al azar, se predijeron los valores de los segmentos con el modelo clasificador y se contrastaron con los valores reales --Figura 5: rectángulo con línea punteada. Los niveles de exactitud (Clasification accuracy) sensibilidad (sensitivity), especificidad (specificity) y Área bajo la curva ROC (Area under ROC Curve) estuvieron por encima del 90%. –Figura 7-.

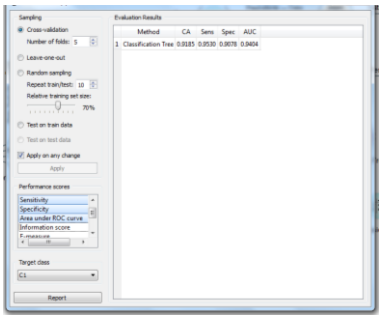


Figura 7

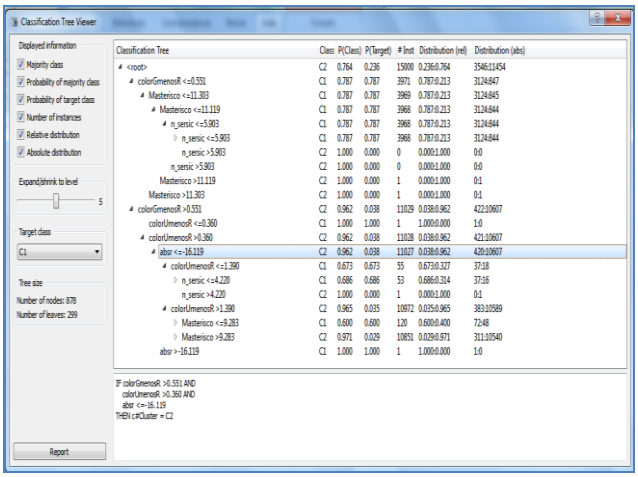


Figura 8

En la Figura 8 se observa como intervinieron los diferentes atributos para la obtención de los segmentos. El principal atributo para la segmentación viene dado por el colorGmenosR para un valor de colorGmenosR<=0.551. Dado que todos los atributos representativos de las características de las galaxias están relacionados entre sí, vemos que dentro del Segmento C1, establecido para las galaxias más azules, se incluyen aquellos objetos con Masterisco <= 11.303, que corresponde a galaxias con bajo contenido de masa estelar, y n_sersic <=5.903, para galaxias con morfología tipo disco. Además esta segmentación se ve reflejada en

los atributos restantes, absr y colorUmenosR, perteneciendo al segmento C1 aquellas galaxias con absr <=-16.119 que tienen colores más azules que colorUmenosR <=1.390 y n_sersic <= 4.220.
Esto representa una adecuada selección de galaxias pertenecientes a poblaciones jóvenes con morfología, edad y colores típicos. Por otro lado, los objetos del segmento C2 poseen atributos indicativos de poblaciones evolucionadas, con colores enrojecidos, mayor tamaño, contenido estelar y luminosidad que las galaxias del segmento C1.

4. Conclusiones

En general se ha logrado interactuar adecuadamente con expertos de un área de conocimiento de gran capacidad de adquisición de datos. Los resultados tanto de segmentación como clasificación han sido válidos desde la perspectiva del experto. Se abre a partir de esta experiencia un camino tendiente a incorporar la minería de datos y el aprendizaje automático en actividades desarrolladas en el Departamento de Astronomía.
Es notoria la necesidad de trabajar con algoritmos y herramientas de software que permitan la manipulación de datos distribuidos. En particular herramientas como RapidMiner, y Orange, poseen algoritmos paralelos que aprovechan los beneficios del hardware multinúcleo. Este aspecto constituye una ampliación necesaria al camino iniciado.

5. Referencias

[1] Krishnan, Krish. Data Warehousing in the age of Big Data. Newnes, 2013.

[2] North Matthew. Datamining for the masses A Global Text Project Book 2012

[3] Way, Michael J., et al., eds. Advances in machine learning and data mining for astronomy. CRC Press, 2012.

[4] Ivezić, Željko, et al. Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data. Princeton University Press, 2014.

[5] Hand, David J., Heikki Mannila, and Padhraic Smyth. Principles of data mining. MIT press, 2001.

[6] Hernández Orallo, José, María José Ramírez Quintana, and César Ferri Ramírez. "Introducción a la Minería de Datos." Editorial Pearson Educación SA, Madrid (2004).

[7] Liu B. Web DataMining Exploring Hyperlinks, Contents, and Usage Data. 2007

Sistemas de Información Geográfica en la gestión de Instituciones Universitarias

M. Soledad Retamar[1], Juan Pablo Nuñez[1], Anabella De Battista[1], Andrés Pascal[1], Francisco Savoy[2], Norma Edith Herrera[3]

1 Dpto. Ing. en Sist. de Información, Univ. Tecnológica Nacional, FRCU, E. Ríos, Argentina
{retamars, jpnunez, debattistaa, pascala}@frcu.utn.edu.ar
2 CONICET. Fac. de Cs. De la Salud, Universidad Nacional de Entre Ríos, Argentina
fransavoy@hotmail.com
3 Dpto. de Informática, Universidad Nacional de San Luis, Argentina
nherrera@unsl.edu.ar

Abstract

Un sistema de información geográfica es una herramienta que permite analizar, visualizar e interpretar hechos relativos a la superficie terrestre, siendo de gran utilidad para la resolución de problemas complejos de planificación y gestión. Existen múltiples aplicaciones donde los sistemas de información geográfica son de suma utilidad, una de ellas es la gestión de instituciones. En este trabajo presentamos un aplicación de los sistemas de información geográfica en la gestión de una institución universitaria, con el objetivo de generar una herramienta que brinde soporte a la toma de decisiones en la implementación de estrategias de promoción y difusión de carreras dictadas en la institución.

1. Introducción

Hace algunas décadas la única forma de sintetizar y representar información geográfica era a través de mapas en papel y la gestión de la información asociada a los mismos se veía reducida a procesos manuales limitados. Actualmente, el desarrollo de nuevas tecnologías ha permitido recolectar y digitalizar datos geográficos y ha generado la necesidad de analizar y manipular de forma interactiva este tipo de información. Todo esto ha generado la necesidad de contar con software específico, y esto a su vez ha derivado en la generación de Sistemas de Información Geográfica (SIG) [1].

Un SIG ofrece funcionalidades para capturar, almacenar, visualizar y gestionar tanto información alfanumérica como información que requiere una representación en el plano de los objetos que se manejan en el dominio de un problema. La posibilidad de analizar la distribución de datos espaciales en un momento de tiempo determinado representa una ventaja para la resolución de problemáticas en diversas áreas de conocimiento como salud, medioambiente, prevención y atención de desastres, seguridad, marketing, planificación urbana y comercial, entre otras [2].

En la actualidad, el incremento en la potencia del equipamiento informático ha hecho posible trabajar con grandes volúmenes de datos, tanto convencionales como aquellos a los que puede asociarse información geográfica. Es interesante destacar que la mayor parte de la información que se maneja en cualquier disciplina está georreferenciada, es decir, que se gestiona información a la que puede asociarse una posición geográfica además de otros datos descriptivos relativos al fenómeno que ocurre en dicha localización.

En este contexto, el presente trabajo presenta una aplicación de SIG para el análisis de información relacionada con los alumnos ingresantes a carreras de grado de una institución universitaria nacional. El objetivo principal es obtener una herramienta para la toma de decisiones al momento de delinear estrategias de promoción y difusión de dichas carreras universitarias en el ámbito local, regional, provincial y nacional.

El resto del artículo está organizado de la siguiente manera. En la Sección 2 presentamos una breve reseña de los SIG. En la Sección 3 se presenta el problema y los objetivos planteados. En la Sección 4 se presenta la metodología utilizada para desarrollar el trabajo. En la Sección 5 se plantea la discusión y finalmente, en la Sección 6, las conclusiones.

2. Sistemas de Información Geográfica

Como ya lo mencionáramos, un SIG es una herramienta que permite analizar, visualizar e interpretar hechos relativos a la superficie terrestre. Un SIG provee funcionalidades para capturar, almacenar, visualizar y gestionar tanto información alfanumérica como información que requiere una representación en el plano de los objetos que se encuentran en el dominio de un problema.

Los SIG han logrado una gran evolución en sus interfaces gráficas, permitiendo generar capas de información que posibilitan la representación tanto de cuestiones básicas, como sería por ejemplo la ubicación de centros educativos de una región o la extensión de las áreas ocupadas por localidades, así como también de algunas un poco más complejas, como por ejemplo la distribución de cultivos o la sectorización por grupos etarios de la población de todo un territorio.

Una capa se define como una relación que contiene en su esquema un atributo cuyo tipo de dato es uno de los tipos de datos geográficos soportados [3]. Una capa soporta tanto las operaciones usuales del álgebra relacional sobre sus atributos no geográficos, como operaciones específicas entre capas referentes a sus atributos geográficos.

Las funcionalidades que implementan los SIG permiten explotar eficientemente la información que almacenan, incluyendo la capacidad de realizar operaciones espaciales sobre los datos geográficos y la de consultar y analizar gráficamente dicha información. Por lo antes expuesto es sumamente importante la componente visual, ya que esta funcionalidad permite al usuario evidenciar más naturalmente las relaciones entre datos geográficos y datos alfanuméricos [3].

En una sociedad donde la información y la tecnología son dos de los pilares fundamentales los SIG constituyen, sin lugar a dudas, la tecnología apropiada para el manejo de información geográfica y los elementos básicos que permiten la gestión de todo aquello que, de un modo u otro, presente una componente geográfica susceptible de ser aprovechada.

Resulta necesario comprender algunos conceptos básicos relacionados con las características de la información que un SIG puede manejar. Los SIG trabajan sobre un espacio geográfico, que es un espacio de coordenadas en el que se representan los datos geográficos. La información de tipo geográfico puede clasificarse en dos grupos dependiendo de si representa características del espacio geográfico (atributos del espacio geográfico) o por el contrario representa propiedades de los objetos gestionados (atributos referentes a los objetos).

En relación al espacio geográfico, se almacenan atributos alfanuméricos que pueden ser continuos o discretos, dependiendo del tipo de valores de su dominio que pueden almacenar, también llamados atributos del espacio geográfico [3].

Sobre dicho espacio se representan objetos geográficos, que son manipulados y representados gráficamente por los SIG. Estos pueden clasificarse en:

- *Objetos geográficos*: de los cuales se guarda tanto información alfanumérica como geográfica, lo que permite representarlos sobre un mapa. Por ejemplo, de una ciudad se pueden almacenar tanto su nombre y su población como su ubicación en base a coordenadas representada mediante un punto y el área que ocupa mediante un polígono.
- *Atributos geográficos*: representan información de características geográficas del objeto al que está asociado. Se representan mediante figuras geográficas.
- *Figuras geográficas*: representan gráficamente sobre el plano atributos geográficos de un objeto. Las más comunes son punto, línea, región.

3. Presentación del Problema de Aplicación

Para una institución de educación superior la posibilidad de contar con información georreferenciada sobre la procedencia de sus alumnos representa una oportunidad para fortalecer y focalizar las actividades de difusión de las carreras universitarias de grado y posgrado que dicta. Por tal motivo, en este trabajo se propone el uso de un SIG para la confección de capas temáticas con información de los alumnos ingresantes, utilizando dos niveles de análisis:

- Una capa de puntos que georreferencia a los aspirantes provenientes de la misma localidad en la que se sitúa la institución educativa.
- Otra capa de puntos que identifican a aquellos alumnos provenientes de otras localidades de cualquier punto del país.

Este trabajo se realizó en el marco de un equipo multidisciplinario del que participaron especialistas en Bases de Datos, Teledetección y Sistemas de Información Geográfica.

Se planteó como objetivo principal del trabajo obtener una herramienta que contribuya al análisis de la información relacionada con la procedencia de los alumnos ingresantes. Esto resultará de suma relevancia para las autoridades universitarias en los siguientes aspectos:

- Para tareas de planificación y desarrollo de políticas de promoción y difusión de las carreras de grado y posgrado que brinda la institución.
- Para establecer contacto con instituciones de distintos puntos del país para la realización de prácticas profesionales por parte de los alumnos.
- Para detectar relaciones que difícilmente puedan detectarse si no es a través de la visualización en un mapa, como por ejemplo la relación entre la localización de instituciones de salud y la procedencia de los alumnos.
- Para identificar instituciones de salud de todo el país con las cuales establecer contacto con el objetivo de favorecer la inserción laboral de los graduados una vez que han obtenido su titulación de grado.

4. Metodología de Trabajo

Este trabajo constó básicamente de tres etapas: la captura de datos, la geocodificación de los mismos y la obtención de resultados. Detallamos a continuación cada una de ellas.

4.1. Captura de Datos

La información relacionada con los alumnos y su historia académica en la institución universitaria es administrada a través del sistema de gestión académica SIU-Guaraní, que se encarga de almacenar los datos desde que una persona ingresa a una carrera hasta su egreso. Este sistema implementa su base de datos en un motor Informix 9.21 y almacena los datos censales de las personas a través de un modelo relacional como se muestra en la *Figura 1*.

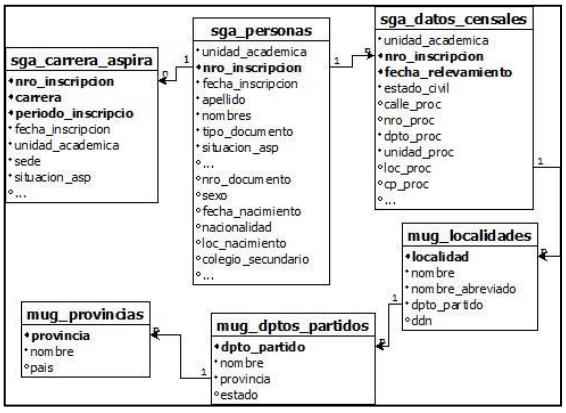


Figura 1: relaciones que permiten almacenar los datos censales de alumnos

En el conjunto de estos datos descriptivos se encuentran atributos candidatos a ser referenciados geográficamente. Los datos geográficos representan hechos que ocurren en la superficie terrestre y constituyen el antecedente necesario para el conocimiento de un fenómeno espacial. Los datos georreferenciados tienen una posición o localización sobre la superficie de la tierra mediante coordenadas establecidas con respecto a un sistema de referencia. Un caso particular de la creación de puntos con coordenadas es la asignación de direcciones dentro de núcleos urbanos, tales como direcciones postales. Estas direcciones son de especial importancia en el desarrollo de actividades dentro del entorno urbano, ya que es más habitual referirse al emplazamiento de un determinado elemento (por ejemplo, un domicilio particular) en términos de su dirección postal que en coordenadas espaciales tales como las que se manejan en un SIG [2].

Hay dos tipos básicos de modelos de datos planimétricos, es decir, dos conjuntos de directrices para la representación lógica de la realidad: Vector y Ráster. Estos dan lugar a los dos grandes tipos de capas de información espacial. Para este trabajo se eligió el modelo vectorial, en donde las entidades se representan en el espacio mediante puntos, líneas o polígonos. Se representaron las provincias a través de polígonos, las localidades mediante puntos y las calles de la ciudad donde se ubica la institución a través de líneas o poligonales.

El municipio de la ciudad donde se ubica la institución educativa en estudio, cuenta con una capa de todas las calles de la localidad y su correspondiente numeración georreferenciadas. Tal capa fue provista al equipo de trabajo para la realización de este análisis. La misma fue ampliada y corregida a través de un relevamiento mediante la captura de coordenadas desde un GPS permitiendo obtener una capa de calles y alturas de la ciudad más ajustada a la realidad. Estos datos fueron tomaron como referencia para poder localizar las direcciones postales de procedencia de los alumnos de la ciudad.

En el caso de alumnos procedentes de otras localidades, se decidió ubicar el punto que lo represente sobre el punto que representa a esa localidad en el mapa provincial o nacional, según corresponda. La capa de los centros poblados del país se obtuvo de la base de datos geográfica del Instituto Geográfico Nacional (IGN) [6]. Para los aspirantes provenientes de la ciudad de la institución en estudio se tuvieron en cuenta los atributos *calle_proc* y *nro_proc* de la relación *sga_datos_censales* de la base de datos del sistema académico SIU-Guaraní, que representan la calle y altura de su domicilio respectivamente, y para

los alumnos oriundos de otras localidades se consideraron los atributos *loc_proc* de la relación *sga_datos_censales* y *provincia* de la relación *mug_provincias*.

La institución en estudio cuenta actualmente con sedes o dependencias en distintas localidades del país en las que se dictan diferentes carreras que, si bien no son de dictado permanente, convocan a gran cantidad de estudiantes. Las cantidades de alumnos ingresantes, según su año de ingreso y su procedencia, se representan en la *Tabla 1*. Según su procedencia se diferenciaron dos grupos: aquellos oriundos de la misma localidad en la que se emplaza la sede central (columna Ciudad) de aquellos provenientes de otras localidades del país (columna Otras localidades).

Se tomó como período de análisis el comprendido entre los años 2008 y 2014, ya que a partir de 2008 se comenzó a utilizar el sistema de gestión de alumnos SIU-Guaraní, por lo que se cuenta con la información completa en el sistema para todos los ingresantes desde ese año. Se presenta en la *Tabla 2* otra clasificación de los ingresantes en la que se presenta por un lado la procedencia de los alumnos que cursan en la sede central y por otro lado, la procedencia de los alumnos que cursan en otras sedes del país.

Tabla 1: Distribución de alumnos ingresantes según año de ingreso y procedencia

	Procedencia	
Año	Ciudad	Otras localidades
2008	441	464
2009	276	252
2010	560	454
2011	349	457
2012	299	347
2013	457	361
2014	326	207

Como se puede observar en el modelo de datos de alumnos (*Figura 1*), los atributos a georreferenciar no se encuentran normalizados a nivel de datos, motivo por el cual se realizó un proceso de depuración de los mismos.

Tabla 2: distribución de alumnos ingresantes según la sede donde se dicta la carrera

	Procedencia del alumno	
Sede de la carrera	Ciudad	Otras localidades
Ciudad	50,4 %	49,6 %
Otras sedes	3,4 %	96,6 %

Con el objetivo de realizar una primera comparación entre los datos existentes y los de referencia se insertaron en la base de datos de Informix dos tablas auxiliares conteniendo la información normalizada correspondiente a las calles de la ciudad y a las demás localidades del país.

El primer control realizado fue la cantidad de aspirantes de la ciudad que no poseían calles de procedencia pertenecientes a la tabla auxiliar de calles normalizada, lo que representaba un 60% del total. Respecto a las localidades un 38% de los registros no se correspondían con la tabla normalizada. Para corregir esta información se exportaron los datos a una planilla de cálculo que permitiera manipularlos sin necesidad de modificar los registros reales.

Entre los registros que no se correspondían ya sea con la tabla de calles como con la de localidades se pueden mencionar los siguientes escenarios:

- *A nivel localidad*: como se explicó anteriormente los nombres de localidades o centros poblados, obtenidos del IGN, fueron incorporados en una tabla auxiliar en la base de los aspirantes y así se pudieron detectar los casos en los que la localidad de procedencia del aspirante no coincidía con ninguna de las localidades de referencia. Los tipos de inconsistencias encontradas fueron:
  - Los registros de aspirantes que poseían el atributo localidad de procedencia vacía fueron desestimados ya que no era posible determinar su localidad de origen, los mismos representan un 0,5% del total.
  - Los registros vinculados a localidades de procedencia denominadas como *Indeterminada* fueron excluidos por la imposibilidad de igualarlos con los datos de referencia. Estos casos representan un 3% del total.
  - Se modificó el nombre de la localidad de los registros que poseían errores ortográficos en su denominación, por ejemplo: “Tucuman” por “Tucumán”, “Saenz Peɒa” por “Saenz Peña”.
  - Se reemplazaron los nombres de las localidades en el reporte de aspirantes que en la tabla de referencia poseían una denominación diferente, por ejemplo: “Ciudad

Autonoma de Buenos Aires" por "Capital Federal".

- *A nivel domicilio*: se identificaron todos los registros de aspirantes que inicialmente no tenían correspondencia con los nombres de calles de referencia.
  - Se modificaron los registros de aspirantes en los que el campo calle poseía errores ortográficos, por ejemplo: la calle "Miillan" se reemplazó por "Millán".
  - Se detectaron y modificaron los registros que poseían nombres de calles desactualizados, esto sucede generalmente porque luego del registro del alumno en el sistema SIU-Guaraní la calle cambió su denominación, por ejemplo la calle "32 del Oeste" en la actualidad es "Del Ciervo", otro caso es el de la calle conocida popularmente como "9 del Norte" que en realidad se denomina "Rubinsky".
  - Se descartaron los registros que no poseían la altura de la calle o la misma estaba mal expresada, por ejemplo: "Metz ***", "Artusi s/n", "Mitre. Estos registros representaban un 0.01% del total.
  - Los registros con nombres de calles inexistentes y que no presentaban ningún dato relevante para poder ser localizados fueron descartados. Este tipo de casos sumaban un total de 11 registros, es decir el 0,009% del total. Ejemplo: "Ssss"; "nn" o vacías.
  - Respecto a las direcciones que estaban registradas como la intersección de dos calles, se controló que incluyeran entre ellas el conector "Y" para asegurar la localización correcta. Ejemplo: "Galarzay Torra" se reemplazó por "Galarza y Torrá".
  - Si la dirección indicaba el nombre de un barrio, que en algunos casos no coincidía con los definidos por el municipio ya que pertenecen a complejos habitacionales, se localizó manualmente un punto que represente la dirección indicada y se reemplazó esa dirección en el registro por la calle y altura a la que pertenecía con un rango de error de una cuadra. Ejemplo: "Barrio del Turf 14 A" se reemplazó por "Pública 0412 725".

Luego de realizar las correcciones mencionadas se generó el listado de los aspirantes provenientes de la misma ciudad donde funciona esta institución universitaria. El resultado de esta consulta se exportó a una planilla de cálculo que permita manipular los datos sin alterar los originales. Se agregó una columna concatenando los atributos número y calle para los registros de la ciudad, y los atributos ciudad y provincia para los pertenecientes a otras ciudades.

4.2. Geocodificación de los datos

El proceso de asignar una referencia geográfica a un registro con la finalidad de poder localizarlo en un espacio geográfico se denomina geocodificación. En este caso no existe un mapa o documento cartográfico, sino una serie de datos espaciales expresados de forma alfanumérica que son susceptibles de convertirse en una capa y emplearse así dentro de un SIG.

Existen diferentes formas de geocodificar información, una de las más generalizadas es utilizar los ejes de las calles con su respectiva numeración -callejero-. En esta modalidad la información de origen debe contener direcciones compuestas por calles y números, es posible utilizar también códigos postales u otros datos de las direcciones, como por ejemplo distintas jerarquías dentro de la red vial (ej.: avenidas).

El proceso de geocodificación se llevó a cabo con la herramienta *ArcCatalog* de *ArcGIS 10.1*.

Para geocodificar un dato en forma automática se debe disponer de un localizador que se encargue de leer cada uno de los registros a localizar, buscar la coincidencia de los mismos en los datos de referencia y finalmente crear el atributo de tipo geográfico por cada registro geocodificado conservando la tabla de datos asociada.

En *ArcGIS* el localizador de direcciones es creado en base a un estilo específico que dependerá de la información que se desee geocodificar y de los datos de referencia existentes, es decir, de la forma en que se encuentre representada la información en los atributos y la topología de los mismos.

Para este trabajo se decidió crear dos localizadores diferentes, uno para el caso de los ingresantes de la ciudad que los ubique en base a su dirección postal (calle y número), y otro para el resto de los ingresantes provenientes de otras localidades del país, que tenga en cuenta la localidad y provincia de procedencia.

Para los puntos en la ciudad se construyó un localizador que permita ubicar puntos para cada lado de la calle y luego fueron compuestos ambos localizadores. El estilo utilizado para la creación de los mismos fue el denominado *One Range* en *ArcGIS*, debido a que en los datos de referencia existe un rango para cada segmento de la calle.

Para el localizador de las ciudades se escogió el estilo *US Address-City State* y se escogieron los atributos *nombre de ciudad* y *provincia* de la tabla primaria para ser comparados.

En ambos localizadores se configuraron los siguientes valores de parámetros para no disminuir la performance del proceso:

- *Rigor ortográfico de 80%*. Este parámetro permite controlar las variaciones de ortografía que admite el localizador cuando busca probables candidatos en los datos de referencia. Cuanto menor sea este valor más candidatos puntuará para la concordancia y el tiempo que tome el proceso de geocodificación se incrementará.
- *Puntuación mínima de candidato en 10*. Este valor es un umbral utilizado por el localizador en la búsqueda de posibles candidatos entre los datos de referencia. Las ubicaciones que obtengan una puntuación menor a este valor no serán mostradas.
- *Puntuación mínima de concordancia en 85*. Este valor permite controlar el nivel de concordancia de las direcciones con los posibles candidatos para que sean considerados válidos. Cuanto mayor sea este valor mayor será el nivel de confianza en el que se ubiquen las direcciones.
- *Conector de intersección Y*. Como se explicó anteriormente para considerar las direcciones expresadas por la intersección de dos calles es necesario configurar en el localizador todas las cadenas que debe reconocer como conectores.
- *Sistema de Referencia Espacial (SRID) de la salida*: Un sistema de referencia espacial es un conjunto de parámetros que se utiliza para representar una geometría. Estos parámetros son: el nombre del sistema de coordenadas del cual se obtienen las coordenadas; el identificador numérico que identifica de forma exclusiva al sistema de referencia espacial; coordenadas que definen la máxima extensión de espacio posible a la que se hace referencia mediante un rango determinado de coordenadas; números que, cuando se utilizan en ciertas operaciones matemáticas, convierten las coordenadas recibidas como datos de entrada en valores que se pueden procesar con máxima eficacia. Al definir el SRID será posible determinar la posición del objeto en el espacio. En este trabajo se georreferenciaron los puntos que representan a los alumnos ingresantes sobre la proyección 22175, que es la de uso común por los organismos oficiales de la provincia de Entre Ríos.

Una vez construidos los localizadores de puntos se importó en *ArcCatalog* el archivo con los registros de ingresantes y se procedió a la geocodificación de los mismos. Durante este proceso se analiza la dirección asociada a cada registro, se crean varias representaciones de la misma, se busca en el localizador ingresado, se establecen puntuaciones de concordancia, se filtra una lista de candidatos y, cuando la puntuación se encuentra dentro del intervalo establecido, se indica la entidad geocodificada.

4.3. Resultados obtenidos

Para el caso de los registros de la ciudad, para los cuales se utilizó la localización a nivel de domicilio, el proceso de geocodificación tuvo un 87% de éxito sobre el total de registros evaluados (*Figura 2*). Con los registros geocodificados se generó una capa de puntos que representa los alumnos ingresantes procedentes de la ciudad para el período 2008-2014.



Figura 2. Geolocalización a nivel localidad utilizando domicilios postales

En la ejecución de la geocodificación de registros de alumnos procedentes de otras ciudades del país se obtuvo un 85% de éxito sobre el total de evaluaciones, a partir de las cuales se generaron varias capas:

- Una capa de la provincia de Entre Ríos para el período 2008-2014 (*Figura 3*).
- Una capa de Entre Ríos con los ingresantes de cada año contemplado en el análisis.
- Capas que representan la distribución de aspirantes de todo el país según su año de ingreso para el período considerado (Figuras 4, 5, 6 y 7).

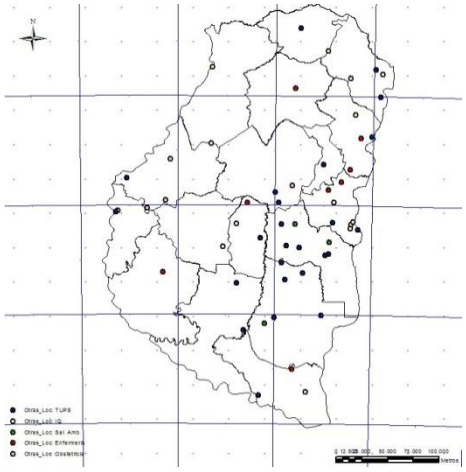


Figura 3. Geolocalización a nivel provincial utilizando localidad de procedencia

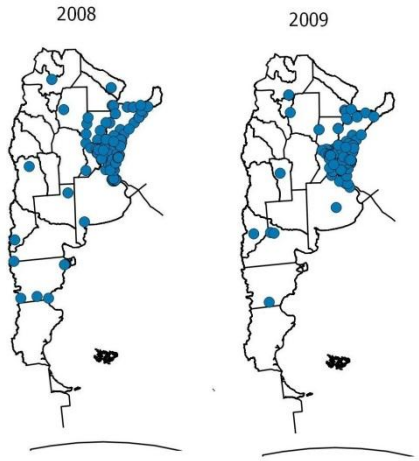


Figura 4: distribución de ingresantes de todo el país para el período 2008-2009

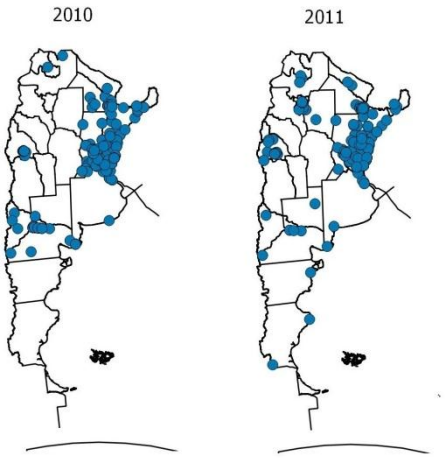


Figura 5: distribución de ingresantes de todo el país para el período 2010-2011

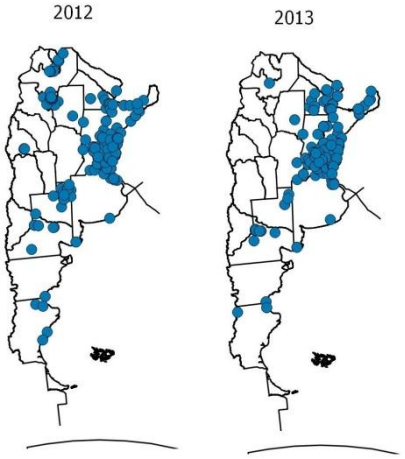


Figura 6: distribución de ingresantes de todo el país para el período 2012-2013



Figura 7: distribución de ingresantes de todo el país para el año 2014

Para completar el análisis se prevé la superposición de estas capas de alumnos ingresantes con la capa de centros de salud de todo el país. Esto permitirá analizar e identificar relaciones entre la distribución de los Centros de Atención Primaria de la Salud y Hospitales y los lugares de procedencia de los alumnos ingresantes de la institución en estudio como así también establecer contacto con las instituciones de salud para que los futuros egresados puedan realizar prácticas profesionales en las mismas.

5. Discusión

La ejecución de este trabajo fue posible gracias a la conjunción de una serie de factores fundamentales en el desarrollo de una actividad de este tipo, como

son: la existencia de un equipo de trabajo multidisciplinario del que participaron especialistas en Bases de Datos, Teledetección y Sistemas de Información Geográfica; la disponibilidad de la información de los alumnos ingresantes de la institución universitaria, de capas de información geográfica que pudieron utilizarse directamente o que pudieron ser actualizadas para realizar el trabajo, del apoyo de las autoridades universitarias en la realización de la tarea comprendiendo los beneficios que representarían los resultados; y la disponibilidad de las herramientas SIG y de bases de datos. Una ventaja de este trabajo es que la información puede actualizarse muy fácilmente, permitiendo también la actualización de las capas de información en los próximos años, con lo que podría también evaluarse la efectividad de las nuevas decisiones que se tomen al planificar y/o modificar las estrategias de difusión y promoción de las carreras de grado y posgrado que ofrece esta institución.

6. Conclusiones

Mediante la utilización de Sistemas de Información Geográfica ha sido posible la generación de una herramienta que facilite a las autoridades de una institución universitaria el análisis de la información relacionada con la procedencia de los alumnos que asisten a la misma para obtener una titulación de grado o posgrado, permitiendo esto determinar si las actividades de difusión realizadas en distintas zonas de la región y del país han sido productivas y además, poder determinar la existencia de otros factores que hacen que un alumno opte por realizar su formación universitaria en determinada área de estudio.

El proceso de análisis de las capas de datos generados está comenzando, por lo que la determinación de factores determinantes en la elección de carreras universitarias de alumnos de la ciudad y la zona estará disponible en los próximos meses.

7. Referencias

[1] Rigaux, P., Scholl, M., & Voisard, A. (2001). Spatial databases: with application to GIS. Morgan Kaufmann.

[2] Sistemas de Información Geográfica. Víctor Olaya. http://wiki.osgeo.org/wiki/Libro_SIG. Versión 1.0. Rev. 25 de noviembre de 2011

[3] Brisaboa, N. R., Lema, J. A. C., Fariña, A., Luaces, M. R., & Viqueira, J. R. Sistemas de Información Geográfica: Revisión de su Estado Actual.

[4] Câmara, G., Monteiro, A. M., Fucks, S. D., & Carvalho, M. S. (2009). Spatial analysis and GIS: a primer. National Institute for Space Research. Brazil.

[5] Manolopoulos, Y., Papadopoulos, A. N., & Vassilakopoulos, M. G. (Eds.). (2005). Spatial databases: technologies, techniques and trends. Igi Global.

[6] Sitio web del Instituto Geográfico Nacional. http://sig.ign.gob.ar/

[7] Sitio web ArcGIS http://www.esri.com/software/arcgis

Detección de Patrones Circulares en Imágenes

Facundo Solano, José María Maffei, Francisco G. Gutiérrez,
Agustín Martino, Juan Eduardo Picco
Centro Universitario de Desarrollo en Automación y Robótica
Universidad Tecnológica Nacional – Facultad Regional Córdoba
Maestro López Esq. Cruz Roja Argentina. CP X5016ZAA. Córdoba, República Argentina.
facundosolano@gmail.com, josemmaffei@gmail.com,fggutierrez@gmail.com,
agustinmartino@hotmail.com.ar, primerjepicco@gmail.com

Abstract

This work explains a method to be able to measure hair of camelids through the vision of a camera. In order to do this, the libraries of OpenCV are used, since they give all the tools needed to process images of every kind and post analyze them in any way. It is because of this, in this paper is given a full explanation of the functions and algorithms used, in order this work to be easy of understanding.

1. Introducción

Hoy en día la necesidad de poder interpretar información en forma de imágenes es muy grande. Esto es particularmente aplicable a muchos campos de la ciencia, ya sea en aplicaciones relacionadas con la medicina, la biología, el agro o bien con la industria, para mencionar algunos. Para dar solución a algunos de los problemas que se presentan, es que se ha generado una gran diversidad de algoritmos y formas de analizar una imagen particular, buscando determinar patrones o características de interés. En este caso, el presente trabajo surge debido a la inquietud de tener un sistema (ver Figura 1) capaz de medir el diámetro de fibras animales, tales como pelos de camélidos (Alpacas, Llamas y Vicuñas, solo para mencionar algunos), fibras que se utilizan en la industria textil. El diámetro de la fibra, es uno de los parámetros que se utilizan para determinar la calidad de la misma. Con el objeto de obtener un procedimiento sencillo, simple y con un estándar de precisión adecuado, se trabaja en lograr un sistema de medición que además sea de bajo costo, para que lo puedan disponer los pequeños productores.

El objetivo de este trabajo es realizar un software de aplicación que sea capaz de reconocer a partir de una fotografía o imagen, patrones circulares y que luego los identifique dibujando un círculo por encima de los mismos. Se ha optado por la utilización de bibliotecas de

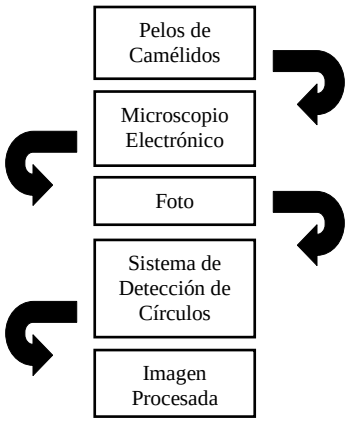


Figura 1: Diagrama en Bloques del Sistema

código abierto escritas en lenguaje de programación C++, pertenecientes a OpenCV [1,2,3], que provee todas las herramientas para poder realizar el estudio que se busca.

Este artículo presenta una descripción general del paquete de bibliotecas OpenCV seguido de una descripción más detallada de las bibliotecas utilizadas en el software desarrollado. Posteriormente se presenta una descripción del software, explicando el porqué de cada sección, agregando un diagrama de flujo de datos para mayor comprensión. Y finalmente, se mostrará una serie de resultados, haciendo comparaciones entre diferentes imágenes para mostrar la eficiencia del código, como así también sus falencias.

2. OpenCV

OpenCV es un paquete de bibliotecas orientado al procesamiento de imágenes, contiene otras bibliotecas escritas para diferentes lenguajes de programación haciendo su uso más versátil. De la página web de OpenCV [3], se puede inferir que estas bibliotecas fueron

construidas para proveer una infraestructura común para aplicaciones visuales en computadoras y para acelerar el uso de la percepción de las máquinas en aplicaciones industriales y comerciales.

Esta biblioteca tiene cerca de 2500 algoritmos optimizados, que pueden ser usados para reconocimiento de rostros, identificación de objetos, clasificación de acciones humanas en videos, seguimiento de movimientos de cámaras, seguimiento de objetos en movimiento, extracción de modelos 3D de objetos, seguimiento ocular, entre otros.

OpenCV tiene bibliotecas codificadas en C, C++, Java, Python y MatLAB, y además es compatible con los sistemas operativos Windows, Android, Linux y MacOS.

En la aplicación desarrollada para el reconocimiento de patrones circulares se utilizaron dos bibliotecas de OpenCV, la biblioteca Improc y la biblioteca HighGUI. Estas bibliotecas se describen a continuación.

3. Biblioteca Improc

Filtrado de imágenes lineales y no lineales. Las funciones y clases utilizadas para las operaciones de filtrado de imágenes en 2D, implican que la ubicación de un píxel (X ; Y), y la vecindad del mismo de la imagen fuente, se utiliza para calcular la respuesta. En el caso de un filtro lineal es la suma ponderada de los valores de los pixeles. En el caso de las operaciones morfológicas, puede ser el valor máximo o el mínimo, u otro valor dependiendo de la operación que se aplique.

Las respuestas calculadas se almacenan en una imagen destino con la misma ubicación del píxel (X ; Y). La imagen tiene el mismo tamaño que la imagen original, además soportan imágenes con multi-canal, donde cada canal se procesa independientemente y por lo tanto la imagen de salida tiene la misma cantidad de canales.

Estas funciones si bien son funciones aritméticas simples, necesitan extrapolar valores no existentes, por ejemplo es el caso de los bordes de una imagen algunos pixeles de la vecindad no existen. OpenCV permite especificar el método de extrapolación.

Transformaciones geométricas: Redimensionamiento, afinado y deformación de perspectiva, las transformaciones geométricas de imágenes 2D no cambian el contenido de la imagen pero deforman la cuadricula y mapa de los pixeles, la salida es almacenada en una imagen destino.

Conversión de espacios de colores: Permiten el paso de un sistema de colores a otro. El espacio de colores definido por defecto es el RGB para OpenCV.

Histogramas: El tratamiento del histograma de una imagen es una de las operaciones más elementales e importantes. Se basa en la distribución de los colores alrededor de la imagen, a partir de las funciones incluidas en la biblioteca se puede tanto obtener como mejorar dicha distribución para obtener un mejor contraste, definición, entre otras características de la imagen a procesar.

Análisis estructural y Descripción de Formas: se incluyen funciones para la descripción y determinación de contornos en una imagen tales como *moments*(…), *findContours*(…), *drawContours*(…), entre otras. Estas funciones son algoritmos optimizados para la detección de características incluidas en una imagen, las cuales son de interés en la visión computacional.

Detección de Características: estas funciones son específicas para hallar bordes, esquinas, círculos, líneas, segmentos, etcétera; en una imagen. Aplican algoritmos estadísticos de análisis y reconocimiento de patrones

4. Biblioteca Highgui

HighGUI es una interfaz de usuario gráfica de alto nivel, que a través de las funciones incluidas en la biblioteca permite la interacción con el sistema operativo, con los archivos del sistema, y con el hardware como cámaras de video, etc.

Se pueden crear ventanas para mostrar imágenes, leer y/o escribir archivos gráficos relacionados (imágenes y videos).

La librería HighGUI está dividida en tres partes: hardware, archivos de sistema y sistema de ventanas (o GUI).

La primer parte se concentra principalmente en la operación de las cámaras, permite de manera rápida obtener imágenes desde una cámara.

La parte de archivos de sistema se dedica a la carga y almacenamiento de imágenes. Una de las características de esta biblioteca es que permite leer video usando el mismo método con el cual se lee una imagen, además de abstraerse de cualquier dispositivo particular. Estas funciones se basan únicamente en la extensión del fichero manejando automáticamente la codificación y/o decodificación necesaria.

La tercer parte del GUI (el sistema de ventanas) provee funciones simples para crear una ventana e introducir imágenes en la misma. También permite registrar y responder ante eventos del mouse y teclado en una ventana. Además en las ventanas se pueden crear *trackbars* y otros componentes gráficos útiles para aplicaciones de interfaz con el usuario.

En esta biblioteca no se tratan independientemente estas partes, ya que están conjugadas en las funciones universales descriptas en las mismas.

5. Desarrollo del Software de aplicación

El software desarrollado puede ser dividido en 3 etapas, las cuales son: 1) carga y acondicionamiento de la imagen, 2) detección de círculos y 3) procesamiento de



Figura 2: Diagrama de Flujo Carga y Acondicionamiento de la Imagen

los datos obtenidos. Dentro de cada una de estas etapas se pueden encontrar diferentes funciones [2], algunas de ellas desarrolladas para este sistema y otras pertenecientes a las bibliotecas de OpenCV.

5.1. Carga y Acondicionamiento de la Imagen

En esta etapa se carga la imagen al programa desde un archivo, y se realiza el acondicionamiento de la misma para su posterior procesamiento. El diagrama de flujo de esta etapa se puede ver en la Figura 2.

Cuando la imagen se carga en el programa lo hace en el tipo Mat, la cual es una clase común para tratar imágenes en OpenCV basada en arreglos numéricos multi-canal o de mono canal de n dimensiones que puede almacenar matrices, números complejos, etc [2]. A continuación la imagen es transformada a escala de grises, para trabajar con un solo canal, ya que de esta forma es más sencillo tratarla, a diferencia de lo que sería utilizar 3 canales RGB, y contar con una imagen que tiene 3 matrices asociadas, debido a que es una matriz por color [1].

Continuando con el acondicionamiento de la imagen, se realiza un *blur* [1], se hace pasar una matriz con un cierto tamaño por la imagen que hace que la misma se vuelva un poco más "borrosa", esto se realiza para eliminar cualquier imperfección que haya sido provocada por la cámara o ruido indeseado en la imagen. Seguidamente se realiza una ecualización de histograma [1], ya que la curva de contraste en las imágenes tomadas por una cámara no es muy uniforme. Esto es debido a que el obturador de las cámaras tiene que hacer un compromiso en el tiempo de apertura, para poder tomar mejor áreas oscuras o áreas con mayor luminosidad. Por ejemplo si se toma una fotografía a un objeto que está muy iluminado, pero detrás de éste se encuentran otros objetos que no lo están, entonces para captar con más detalle los objetos del fondo, el obturador debería de permanecer más tiempo abierto, lo cual provocaría que el primer objeto salga muy iluminado. Sin embargo el caso contrario sería un tiempo corto con el obturador abierto, provocando que los objetos del fondo salgan muy oscuros. Por lo tanto para realizar esto se aplica la función *equalizeHist(…)* [2]. Una vez aplicada dicha función, la curva de contraste de la imagen se vuelve más uniforme y se realzan más los contornos.

Luego de aplicarse estas funciones, se hace uso de una función de Umbral Adaptivo [1, 2, 4] o bien la función *adaptiveThreshold(…)*. El objetivo del uso de ésta es poder hacer un filtrado más a la imagen, que sirva para desechar pixeles que no son de interés, por ejemplo aquellos que se consideran como ruido en la imagen. Esta función permite generar un umbral que se adapte según la imagen a evaluar, para procesarla de la mejor forma. Generalmente es usada para imágenes que tienen fuertes puntos de iluminación. En su argumento se encuentran una serie de parámetros entre los cuáles, los más importantes a destacar son el método adaptativo y el tipo de umbral. En este trabajo se usó el *ADAPTIVE_THRESH_MEAN_C* como método adaptativo, ya que se tuvo un buen resultado y utilizando otro método no se vieron diferencias notables. Este método se establece sobre una base de píxel por píxel mediante el cálculo de un promedio de los pesos de cada pixel de la región *bxb* alrededor de cada posición del píxel central, menos una constante, donde *b* está dada por el parámetro *block_size* y la constante está dada por *param1*. Si el método que se elige es el CV_ADAPTIVE_THRESH_MEAN_C, entonces, todos los píxeles de la zona tienen el mismo peso.

Además se elige el parámetro *THRESH_BINARY* como tipo de umbral, porque se busca tener un buen contraste entre los "objetos" de la imagen y el fondo de la misma, esto binariza a la imagen ya que la pasa a una matriz que está compuesta solo de unos y ceros.

Finalmente, antes de que se realice la detección de círculos se aplica una última función, *morphologyEx(…)*[2], la cual permite realizar transformaciones morfológicas en la imagen, tales como

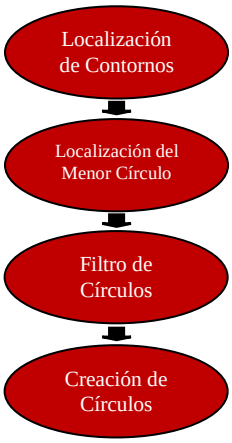


Figura 3: Diagrama de Flujo Detección de Círculos

erosiones y dilataciones. Dichas acciones se utilizan para unir objetos en la imagen analizada, o bien para separarlos, así como también otras funciones como eliminar ruido, encontrar agujeros, etc. En este caso se utiliza la transformación *Opening* [1] la cual consiste en realizar primero una erosión y luego una dilatación. El objetivo de ejercer esta acción es tratar de separar los objetos circulares que pudiesen estar muy juntos. La forma en la que funcionan ambas funciones son similares, las dos utilizan un *Kernel*, lo cual se puede interpretar como una máscara que puede tener cualquier forma, generalmente un cuadrado, esta máscara es una matriz de unos y ceros que equivalen a los pixeles de la imagen. Dicho *Kernel* tiene en su centro un pixel, denominado *pixel ancla*, el cual en el caso de la dilatación toma el valor del pixel con el mayor valor en su vecindad, de manera contraria sucede con la erosión, que toma el pixel con menor valor.

5.2. Detección de Círculos

A partir de la imagen binarizada se procede a la determinación de contornos circulares que se hallan en la imagen, mediante la función *cvFindContours(…)*, esta es la encargada de encontrar secuencias que describen un contorno en la imagen. Un contorno se define como una lista de puntos que a su vez puede ser interpretado como una curva. El diagrama de flujo de esta etapa se puede ver en la Figura 3.

Se define una variable la cual contendrá los contornos hallados esta es del tipo "*vector< vector< Point >>*", es decir es un vector de vectores de puntos, correspondientes a los contornos definidos por dichos puntos. Se pasa a la función *cvFindContours(…)* la imagen binaria y esta almacena en la variable "contornos" las curvas encontradas, además se especifican otros parámetros propios de dicha función como: el modo de recuperación de contornos *CV_RETR_EXTERNAL* que indica al algoritmo que se debe devolver solamente un contorno exterior al contorno analizado, es decir en un área se puede detectar contornos internos y externos y con este parámetro solo se devuelve el contorno más externo del área. Otro parámetro es el método de aproximación de contornos *CV_CHAIN_APPROX_SIMPLE* que comprime segmentos verticales, horizontales y diagonales que definen el contorno y devuelve solo los puntos finales de los mismos. El parámetro *Offset* hace que la función devuelva los contornos corridos a partir de desviar la posición de los puntos encontrados, este corrimiento se indica como un punto *cvPoint(0,0)*.

Luego se generan dos variables para almacenar los radios y centros de los círculos que se crearan a partir de los contornos encontrados por *cvFindContours(…)*. De acuerdo con la cantidad de contornos hallados se procede al análisis de los mismos. La función *minEnclosingCircle(…)* devuelve el centro y radio del mínimo circulo que encierra un área definida por el contorno, para ello se pasan como parámetros el contorno y las variables en donde se guardan los valores calculados.

A continuación se calcula el área del contorno, mediante *contourArea(…)*, y el área del círculo, aplicando la formula geométrica de cálculo de área para círculos. Se compara el área del contorno con un porcentaje del área del círculo, ya que este contorno puede no ser regular pero lo suficientemente aproximado a un círculo, con lo cual se eliminan los contornos que no son de interés. Además se hace una selección de círculos que estén dentro de un rango de valores de radios limitados.

Por último, con la función *circle*(…) se dibujan los círculos de los contornos en la imagen, para ello se pasan como parámetros a esta función la imagen procesada, los radios y centros de los círculos definidos anteriormente

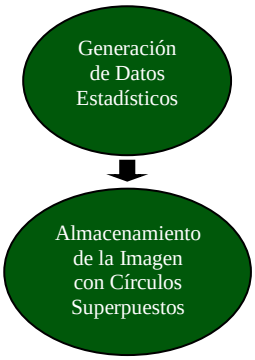


Figura 4: Diagrama de Flujo Procesamiento de los Datos Obtenidos

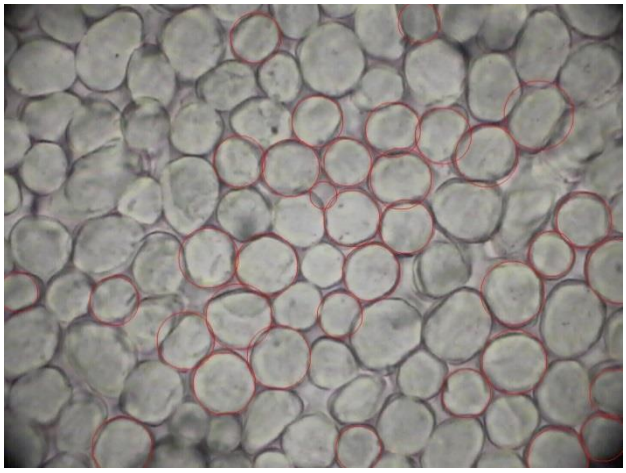


Figura 5: Imagen Original 1 sin procesar con círculos encontrados.

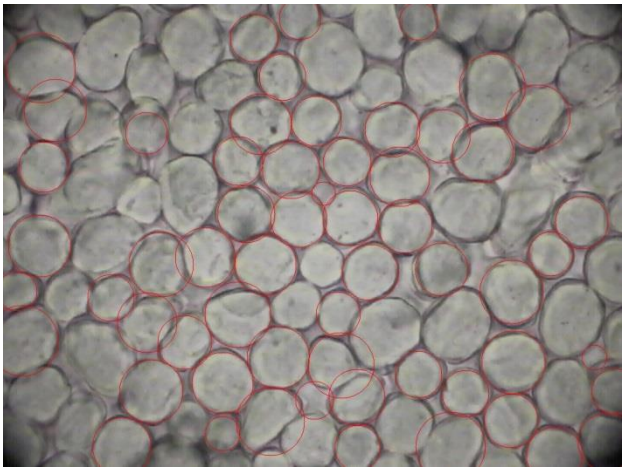


Figura 6: Imagen Procesada 1 con círculos encontrados.

5.3. Procesamiento de los Datos Obtenidos

Uno de los objetivos principales en este desarrollo, además de la detección de círculos, es poder realizar un sistema de medición del diámetro de los círculos detectados. Es por ello, que en la sección anterior, a medida que se detecta un círculo se van acumulando sus radios en una variable, y de la misma forma se van acumulando el total de círculos encontrados en otra variable. El diagrama de flujo de esta etapa se puede ver en la Figura 4.

De esta forma, con estos datos, se puede sacar una media del tamaño de los círculos encontrados. Y luego calibrando el tamaño de las fotos, y conociendo a cuántos micrómetros equivale un pixel, se puede obtener una medida normalizada del promedio del radio de círculos encontrados.

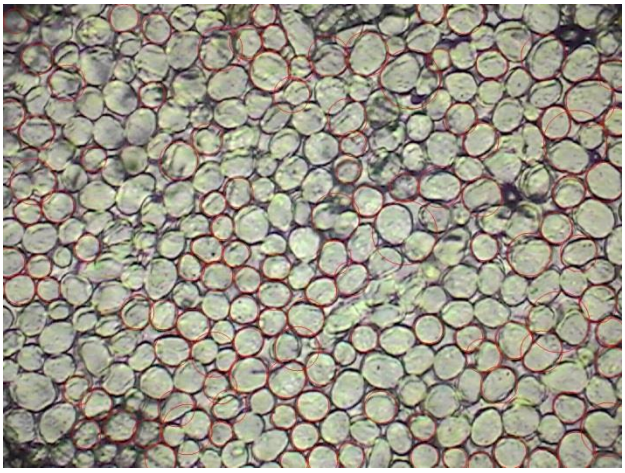


Figura 7: Imagen Original 2 sin procesar con círculos encontrados.

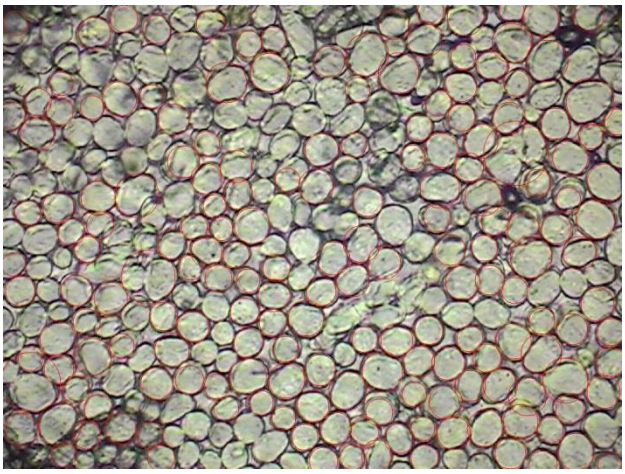


Figura 8: Imagen Procesada 2 con círculos encontrados.

Como último paso se genera una imagen, la cual es la original, pero con los círculos encontrados superpuestos a la misma. Además se genera un archivo en el que se guardan todos los radios de los círculos encontrados, así como también el promedio y otros datos útiles para el análisis.

6. Resultados y Comparaciones

Luego de varias pruebas y modificaciones al programa principal, se decidió que el algoritmo que se presenta es el que mejor se adapta para el tipo de imágenes con las que se trabajó (ver Figura 5, 6, 7, 8). Sin embargo no es perfecto, ya que no todos los pelos de camélidos son detectados, y pese a que con las fotografías con las que se trabajó se obtuvieron buenos resultados, en el momento que se cambió de microscopio para obtener una imagen,

el procesamiento no tuvo el mismo resultado que con las imágenes anteriores.

Se realizaron comparaciones entre las diferentes imágenes obtenidas, por un lado se puede ver la detección de círculos en la imagen original y por el otro lado se observa el incremento de círculos detectados gracias al procesamiento de la misma. Estas comparaciones se pueden ver en las figuras 5, 6, 7 y 8.

Si bien el algoritmo presenta una notable mejora en la detección en todos los casos tiene la falencia de que se encuentran contornos erróneos.

El problema se lo atribuye a que a pesar de que el método ecualiza la imagen y le realiza todo un tratamiento para acondicionarla a la detección de círculos, no funciona para imágenes que entran con cualquier característica visual al sistema. Es decir luminosidad, contraste, separación entre los objetos contenidos en la imagen, entre otros.

7. Referencias

[1] Gary Bradski and Adrian Kaehler,*Learning OpenCV*,O'Reilly Media,United States of America, 2008.

[2] http://opencv.org/documentation.html,*The OpenCV*Reference Manual Release 2.4.6.0, 5, July, 2013.

[3]OpenCV, *About*, http://opencv.org/, 2014.

[4] E. Davies, *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990.

Análisis Comparativo en Múltiples Plataformas de los Principales Métodos Holístico Usados para la Identificación de Rostros

Pablo Santiago Pugliani
F.C.E.F.yN. Postgrado Telecomunicaciones
Universidad Nacional de Córdoba
pugliani@gmail.com

Elizabeth Vera
F.C.E.F.yN.
Universidad Nacional de Córdoba
elizabethvera99@gmail.com

Resumen

El trabajo presenta un análisis comparativo de los principales métodos heurísticos de reconocimiento de rostro: Eigen Face, Fisher Face, Local Histogram Binary Patterns y Compressing Sensing. A todos ellos se los codifica para testearlos en Matlab y Octave, con el sistema operativo Windows 7 y Linux Ubuntu 12.04. Se los compara en base al tiempo de procesamiento y a la tasa de reconocimiento de rostros correctamente e incorrectamente identificados.

1. Introducción

El procesamiento de imágenes ha tenido en las últimas décadas un vertiginoso desarrollo debido a varios factores entre los que podemos destacar: la potencia de cálculo que ofrecen las computadoras y los dispositivos móviles (notebook, computadoras de escritorio, tabletas, teléfonos inteligentes, etc.), la masivización por sus bajos costos y la demanda de los usuarios por aplicaciones que les permita interaccionar con imágenes y/o gráficos.

En particular, presentan un interés especial por múltiples razones: desde problemas de seguridad a comunicaciones en redes sociales, el reconocimiento y la detección de rostros. Ambos están relacionados pero tienen misiones distintas. El reconocimiento de rostros es comparar una imagen que contiene el rostro de una persona contra un grupo de imágenes de rostros previamente identificados y decidir si hay o no coincidencia con alguno de ellos, mientras que la detección de rostros consiste en ubicar un rostro en una imagen. Si se plantea un sistema automático de reconocimiento de rostros, seguramente la primera etapa sería la detección y la siguiente la identificación del rostro detectado. Cada una tiene su forma de trabajo, si bien comparten algunos algoritmos, son dos campos autónomos. En lo que sigue se trabaja solo con la identificación de rostros.

Estos utilizan imágenes en donde únicamente se encuentra un rostro, el cómo se lo obtuvo no es parte de la problemática. Los métodos encargados de la identificación de rostros los podemos clasificar en dos grupos: geométricos y holísticos. Los primeros se caracterizan por obtener puntos característicos en un rostro (nariz, ojos, etc.), a partir de ellos calcular un vector con las características propias del rostro y posteriormente compararlo con los vectores de características de los rostros ya identificados [1,2,3,4]. Los métodos holísticos trabajan la imagen como un todo y así obtienen el vector de características del rostro a identificar. Los principales métodos holísticos son: Eigen Face [5], Fisher Face [6], Local Histogram Binary Patterns (LHBP) [7] y Compressed Sensing (CS) [8].

A las imágenes se las representa como matrices (usualmente se trabaja con escala de grises debido que se reduce en menos de un tercio la carga computacional) y en la ingeniería se utilizan programas de cálculo basado en ellas. Son dos los programas más usados: MalLab (programa con licencia paga) y Octave (con licencia GNU). Ambos son interpretados, aceptan archivos con instrucciones por ejecución por lote (stripts), son multiplataformas (Windows y Linux) y tiene una compatibilidad en el código próxima al 100%, permitiendo usar el mismo script en cualquiera de los dos programas y en cualquiera de las plataformas en el que esté instalado. Otra ventaja de utilizar estas herramientas es la facilidad de la codificación dado que las instrucciones tienden a guardar similitud con la notación matemática, simplificando la lectura de un script.

El trabajo está organizado de la siguiente manera: la sección 2 aborda los principales métodos holísticos desde el punto de vista de la implementación, la sección 3 presenta las bases de datos de las imágenes usadas y el protocolo para comparar los distintos métodos, en la

sección 4 los resultados y se finaliza con las conclusiones.

2. Métodos Holísticos

Los métodos holísticos toman a la imagen como un todo y trabajan con toda la información de ella, en escala de grises para alivianar la carga de procesamiento. Todos los métodos que se tratan, se los aborda desde el punto de vista de la implementación, ejemplificando con fracciones de códigos de los scripts.

Todos trabajan con las imágenes como vectores columnas, la transformación se logra de dos formas: tomando todas las columnas y concatenándolas una debajo de otra o tomando las filas, transponerlas y concatenarlas. Cualquiera de las dos opciones es válida, solo que se deben usar de la misma forma en todas las implementaciones.

2.1. Eigen Face

Lo proponen Turk y Pentland [5]. Está basado en el análisis de componentes principales (PCA). El método tiene dos instancias: el entrenamiento y reconocimiento.

2.1.1. Entrenamiento. El objetivo es encontrar un conjunto de vectores que sea representativo del grupo de rostros reconocidos y se la conoce como base EigenFace del espacio de rostros.

La implementación se realiza de la siguiente forma:
a) Cada imagen de entrenamiento se la representa $\vec{\Gamma}_i \in R^{Nx1}$ donde N es la cantidad de filas por columnas. Al conjunto de todas las imágenes de entrenamiento M se las representa por la matriz $\Gamma = \left[\vec{\Gamma}_1 \vec{\Gamma}_2 .....\vec{\Gamma}_M\right] \in R^{NxM}$

b) Se calcula el promedio $\vec{\psi} = \frac{1}{M}\sum_{i=1}^{M}\vec{\Gamma}_i \in R^{Nx1}$ y a continuación las diferencias $\vec{\phi}_i = \vec{\Gamma}_i - \vec{\psi}$ con las que se construye la matriz $\Phi = \left[\vec{\phi}_1 \vec{\phi}_2 .....\vec{\phi}_M\right] \in R^{NxM}$ y con ella la matriz de covarianza $C = \Phi\Phi^T \in R^{NxN}$

c) Se calculan los autovectores y autovalores de la matriz C. Sin embargo esta es una tarea de muy alta carga computacional debido a las dimensiones de la matriz C lo que lo hace casi impracticable. Turk y Pentland [5] proponen trabajar usando la propiedad que dice que $\Phi\Phi^T \in R^{NxN}$ y $\Phi^T\Phi \in R^{MxM}$ tienen los mismos valores propios no nulos y son ellos los útiles

d) La base EigenFace del espacio de rostros $u = \left[\vec{u}_1 \vec{u}_2 ... \vec{u}_M\right] \in R^{NxM}$ se obtiene a partir del cálculo de los autovalores λ y los autovectores $v = \left[\vec{v}_1 \vec{v}_2 ... \vec{v}_M\right] \in R^{MxM}$ resolviendo $\Phi^T\Phi v = \lambda v$. A los autovectores se los ordena de mayor a menor según los valores de λ. Los M vectores $\vec{u}_i$ se obtienen haciendo $\vec{u}_i = \frac{\Phi v_i}{\sqrt{\lambda_i}}$

e) Para concluir se calculan las proyecciones de las M imágenes usadas en el entrenamiento sobre los eigenvectores: $\Omega = \left[\vec{\omega}_1 \vec{\omega}_2 .....\vec{\omega}_M\right] \in R^{NxM}$ donde $\vec{\omega}_i = u^T(\vec{\Gamma}_i - \vec{\psi})$

2.1.2. Reconocimiento. Esta etapa es la encargada de determinar si la imagen de un rostro es o no uno de los rostros reconocidos. Para lograrlo se proyecta la imagen a reconocer $\vec{\Gamma}_?$ al espacio EigenFace: $\vec{\omega}_? = u^T(\vec{\Gamma}_? - \vec{\psi})$, se calcula la distancia euclidiana contra todas las proyecciones Ω, $e = [e_1 e_1 .... e_M]^T \in R^{Mx1}$ donde $e_i = \left\|\vec{\omega}_? - \vec{\omega}_i\right\|$, se busca la menor distancia $e_{\min}$ y se la compara contra dos umbrales:

Si $e_{\min} \leq umbralMin$ es posible reconocer el rostro y la posición de la menor distancia da la identificación.

Si $e_{\min} \geq umbralMax$ el rostro no se puede reconocer

Cuando $umbralMin < e_{\min} < umbralMax$ hay incertidumbre. En este caso se puede tomar como que no se logró reconocer y deberá aplicarse otro método.

2.2. Fisher Face

El método fue propuesto por Belhumeur, y otros [6]. La idea es implementar un procedimiento más robusto que el Eigenface para lo cual trabaja con clases que corresponden a una colección de imágenes del rostro de una persona en distintas condiciones de iluminación y expresiones faciales. Toma la idea del discriminante lineal de Fisher (FLD) [9]. Tiene dos instancias, el entrenamiento y el reconocimiento.

2.2.1. Entrenamiento. En esta etapa se busca la matriz que reduzca la dimensionalidad y logre proyecciones de los rostros pertenecientes a las distintas clases lo más alejadas posible.

Cada imagen con N píxeles, se transforma en un vector $\vec{\Gamma}_i \in R^{Nx1}$. En total hay c clases, cada clase *i* se representa: $\Gamma_i = \left[\vec{\Gamma}_1 \vec{\Gamma}_1 .... \vec{\Gamma}_{ni}\right]$ donde ni es la cantidad de imágenes de la clase i (del mismo sujeto en distintas

condiciones de iluminación y expresión). La cantidad total de imágenes que se usa es $M = \sum_{i=1}^{c} ni$ Se calcula el valor medio de las imágenes $\vec{\Psi} = \frac{1}{M}\sum_{i=1}^{M} \vec{\Gamma}_i \in R^{Nx1}$ y el valor medio de cada clase $\vec{\psi}_i = \frac{1}{ni}\sum_{i=1}^{ni} \vec{\Gamma}_i \in R^{Nx1}$ .La forma de relacionar las clases es con el cálculo de la dispersión entre clases S_B y la dispersión de la clase S_W :

$$S_B = \sum_{i=1}^{c} \left(\vec{\psi}_i - \vec{\Psi}\right)\left(\vec{\psi}_i - \vec{\Psi}\right)^T \in R^{NxN}$$

$$S_W = \sum_{i=1}^{c} \left[\sum_{j=1}^{ni} \left(\vec{\Gamma}_j - \vec{\psi}_i\right)\left(\vec{\Gamma}_j - \vec{\psi}_i\right)^T\right] \in R^{NxN}$$

valores con los que se procede a calcular la matriz de transformación al espacio FisherFace W_{op} resolviendo:

$$W_{op} = arg_máx \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \in R^{NxM}$$

$W \in R^{NxN}$ son los autovectores de $S_W{}^{-1} S_B$. La inversa $S_W{}^{-1}$ es difícil de obtener debido a su gran tamaño. La solución que proponen los autores es: $W_{op}{}^T = W_{FLD}{}^T W_{PCA}{}^T$ donde $W_{PCA} \in R^{Nx(M-c)}$ es la matriz base EigenFace usando únicamente los principales $M - c$ vectores columnas y $W_{FLD} \in R^{(M-c)x(M-c)}$ son los autovectores de $\left|\left(W_{PCA}{}^T S_W W_{PCA}\right)^{-1}\left(W_{PCA}{}^T S_B W_{PCA}\right)\right|$

Hay dos formas se resolver esta última ecuación: calcular S_B, S_w, luego proyectarlas usando W_{PCA}, calcular W_{FLD} y W_{op} con el inconveniente que puede desbordarse la memoria al trabajar con matrices de NxN . La otra alternativa es aplicar W_{PCA} a todas las imágenes de entrenamiento y sobre las proyecciones aplicar el método. Esta es la forma que se usa en la parte experimental del trabajo.

Por último se proyectan las imágenes de entrenamiento al espacio FisherFace:

$\Omega = \left[\vec{\omega}_1 \vec{\omega}_2 .....\vec{\omega}_M\right] \in R^{(M-c)xM}$ donde

$\vec{\omega}_i = W_{op}{}^T (\vec{\Gamma}_i - \vec{\Psi})$

2.2.2. Reconocimiento. Se aplica de igual manera que en Eigen Face, ver 2.1.2

2.3. LBPH

El método fue propuesto por Ahonen y otros[7]. Considera la información de textura y los bordes de los rostros. Tiene como ventaja sobre el resto, que se puede agregar nuevas imágenes de rostros identificados a la base de imágenes sin necesitar de recalcular. Se basa en el operador LBP de Ojala [10]

El método tiene dos instancias, el entrenamiento y el reconocimiento. Ambos quieren el cálculo del operador LBP

2.3.1. Operador LBP. El operador resalta la información de textura, trabaja sobre un píxel central y sus vecinos, en la Figura 1 se utiliza una cantidad de pixeles $P = 8$, con un radio $R = 1$, esta es la configuración usada por defecto porque de otra forma se debe interpolar los píxeles.

g_7	g_6	g_5
g_0	g_c	g_4
g_1	g_2	g_3

i=0,1 ,2 ,3 ,4 ,5 ,6 ,7
g_i= píxel vecino
g_c= píxel central

Figura 1 Recinto al que se le aplica el LBP

El operador en la Figura 1 se aplica de la siguiente manera:

$$LBP_{P,R} = nuevo_valor_central$$

$$LBP_{8,1} = \sum_{i=0}^{P-1} s(g_i - g_c)^{2^i}$$

$$s(g_i - g_c) = \begin{cases} 1 si(g_i - g_c) \geq 0 \\ 0 si(g_i - g_c) < 0 \end{cases}$$

El resultado es un valor comprendido entre 0 y 255 (8bits), todas son posibles formas, pero Ojala llega a la conclusión que el 90% de ellas se pueden agrupar en 10 patrones que se denominan uniformes (Figura 2). Cada resultado del operador $LBP_{8,1}$ se lo compara con los patrones uniformes (pueden tener los bits rotados pero siempre deben estar todos los bits presentes en el orden del patrón), en el caso que no esté en ninguno de los patrones del 0 al 8 se lo etiqueta como 9.

El etiquetar el resultado de $LBP_{8,1}$ como uno de los posibles patrones uniformes, se cambia la notación por $LBP_{8,1}^{u2} = etiquetado(LBP_{8,1})$

El operador se aplica a todos los píxeles de una imagen. En los bordes se puede optar por replicar o empezar a aplicarlo un R más adentro.

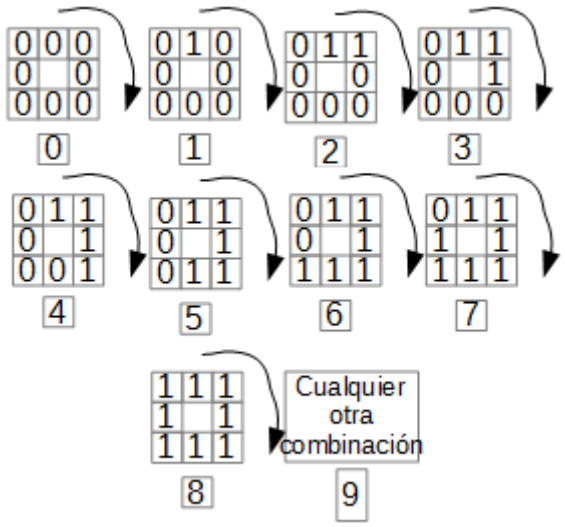


Figura 2 Patrones Uniformes y sus etiquetas

2.3.2. Entrenamiento. Se basa en calcular el histograma a una imagen a la cual previamente se le aplico el operador $LBP_{8,1}^{u2}$ a todos los píxeles. Hacer un único histograma a toda la imagen genera 10 valores, cifra que es muy escueta para usar a la hora de realizar comparaciones para identificar un rostro. Por esta razón se segmenta la imagen y en cada región se aplica el histograma, la cantidad de divisiones en el eje x se la conoce como $grillaX$ y en el eje y $grillaY$ todos los histogramas se concatenan uno a continuación del otro para formar un vector columna. La forma de concatenación tiene que ser la misma cuando se realice el reconocimiento.

En resumen el entrenamiento consta de las siguientes etapas: 1) se obtienen las imágenes de los rostros previamente identificados en escala de grises (se trabaja como matrices y no vectores). 2) a cada imagen se la segmenta (teniendo en cuenta $grillaX$ y $grillaY$), en cada subregión se aplica el operador $LBP_{8,1}^{u2}$, se calcula el histograma y se normaliza el resultado. Se arma el vector con todos los histogramas. Este paso se lo realiza a todas las imágenes. 3) se crea una matriz en donde cada columna es el resultado de todas las operaciones previas a cada imagen identificada. En la Figura 3 se observa un rostro original y uno segmentado con los siguientes valores: $grillaX = 2$ y $grillaY = 2$ y en la Figura 4 el resultado de aplicarle el histograma y el operador $LBP_{8,1}^{u2}$ a las 4 regiones.

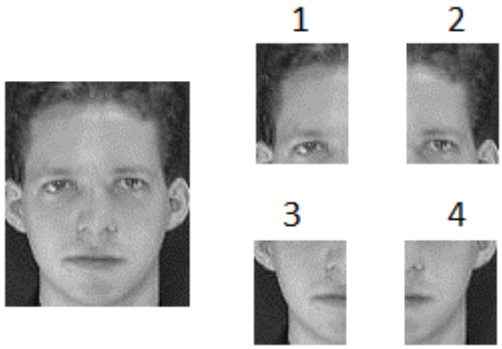


Figura 3 Rostro Original (iz) y segmentado (de)

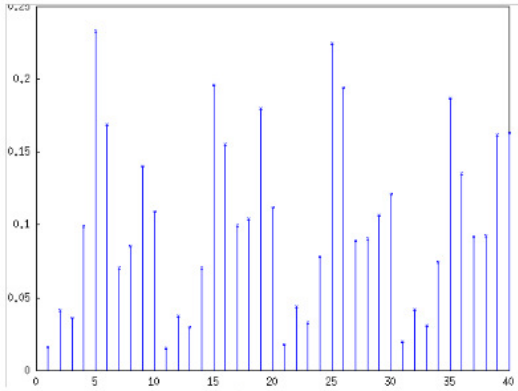


Figura 4 resultado de aplicarle el histograma y el operador LBP

2.3.3. Reconocimiento. A la imagen del rostro a identificar se la segmenta, se aplica el operador $LBP_{8,1}^{u2}$, se calcula el histograma, se lo normaliza y se arma el vector con todos ellos concatenados. Posteriormente se calcula la distancia euclidiana con cada vector de los histogramas calculados en el entrenamiento, si la menor de las distancias es menor o igual a un umbral hay reconocimiento.

2.4. CS

Compressed Sensing (CS) es una técnica de procesamiento de señales diseñado para la eficiente adquisición y reconstrucción de la señal. Explota el hecho que muchas señales de la naturaleza son ralas o compresibles, en el sentido que ellas tienen una representación con pocos términos no nulos cuando se expresan en una base adecuada.

Los fundamentos desde el punto de vista matemático se debe a los trabajos de Candès[11] y Donoho [12]. Candès y Romberg crean las librerías *l1-magic* [13] que están codificadas como un script compatible con MatLab y Octave. Los trabajos de Baraniuk [14], Yang [8] y

Davenport [15][16] lo llevaron al mundo de la ingeniería y al reconocimiento de rostros.

Matemáticamente, se plantea una matriz $A\ (DxM)$ con $D < M$. Definido el sistema $Ax = b$ este tiene más incógnitas que ecuaciones, presentándose dos alternativas: no tiene solución si b no pertenece al subespacio generado por las columnas de A, o tiene infinitas soluciones. De entre todas ellas se desea encontrar la solución rala, esto es la que presente el menor número de valores no nulos.

2.4.1. Entrenamiento. El método trabaja con clases y las imágenes se las transforma a escala de grises y como vectores columna. La cantidad total de imágenes es M , N es la cantidad de píxeles de cada una de ellas y se las representa como $\vec{\Gamma}_i \in R^{Nx1}$, la cantidad total de clases es ct y la cantidad de imágenes de una clase (o de una persona) es c_i . Luego $M = \sum_{i=1}^{Ct} c_i$

El paso siguiente es armar la matriz base del CS o diccionario: $\Psi \in R^{NxM}$ donde $\vec{\psi}_i = \dfrac{\vec{\Gamma}_i}{\left\|\vec{\Gamma}_i\right\|_2} \in R^{Nx1}$

Ya que Ψ posee una elevada dimensión complicando o hasta imposibilitando trabajar con esa cantidad de elementos, se crea una matriz de medición que se caracteriza por tener una distribución Gaussiana o pseudo-Gaussiana (para que cumpla las condiciones la propiedad de isometría restricta (PIR) requerida por el método, los vectores columnas están normalizados y la cantidad de filas tiene que ser $D < M$ quedando: $\Phi = \left[\vec{\phi}_1 \vec{\phi}_2 ... \vec{\phi}_N\right] \in R^{DxN}$ con

$\vec{\phi}_i = norm(valoresAleatoriosGassianos)$

Con todo lo calculado se obtiene la matriz $A = \Phi\Psi \in R^{DxM}$, la que se usa para aplicar CS, arrojando más incógnitas que ecuaciones y cuya solución rala se encuentra usando la minimización de la norma l1 en la etapa de reconocimiento.

Reconocimiento. Lo primero es transformar la imagen del rostro a reconocer como una vector columna, quedando $\vec{\Gamma}_? \in R^{Nx1}$ Posteriormente se le practica una medición no lineal usando Φ y se la normaliza: $\vec{b} = \dfrac{\Phi\vec{\Gamma}_?}{\left\|\Phi\vec{\Gamma}_?\right\|_2} \in R^{Dx1}$ que permite plantear la ecuación: $\vec{b} = A\vec{s}$ donde $\vec{s} \in R^{Mx1}$. El problema es resolver el sistema de ecuaciones, esto es encontrar el valor de $\vec{s}$ que haga mínimo $\left\|\vec{s}\right\|_1$. Candès desarrolló la librería *l1-magic*, pero requiere una condición inicial $\vec{s}_0$ que puede ser la obtenida con la seudo-inversa de A. En la Figura 5 se muestra la forma de llamar a la librería desde un script de MatLab u Octave.

```
s=l1eq_pd(s0,A,A',b);
```

Figura 5 Llamado de la función l1-Magic

$\vec{s}$ es un vector que contiene los coeficientes para la representación de la imagen en el dominio CS. Si la imagen a reconocer coincide con algún rostro usado en el entrenamiento, solamente tendremos coeficientes en las posiciones correspondientes a la clase a la que pertenece. En el caso extremo (el rostro no está en ninguna de las clases) $\vec{s}$ tendrá todos los coeficientes no nulos, si el rostro a reconocer debe ser identificado y no se usó en el entrenamiento, puede haber coeficientes en alguna posición ajena a su clase. Para levantar ese inconveniente se calcula el índice de concentración de esparcidad (SCI):

$$SCI(\vec{s}) = \frac{\dfrac{ctMáx\left\|\vec{\delta}(\vec{s})_i\right\|}{\left\|\vec{s}\right\|_1} - 1}{ct - 1} \qquad 0 \le SCI(\vec{s}) \le 1$$

$\vec{\delta}(\vec{s})_i$ es un vector que contiene únicamente los elementos de $\vec{s}$ que pertenecen a la clase i , en total hay ct vectores $\vec{\delta}(\vec{s})_i$ El escalar $Max\left\|\vec{\delta}(\vec{s})_i\right\|$ se obtiene de calcular las ct normas l_1 y la mayor es la que se utiliza. $SCI(\vec{s}) = 1$ significa que la imagen a reconocer se puede representar únicamente por los elementos de su clase (reconocimiento perfecto), $SCI(\vec{s}) = 0$ no puede ser reconocida. Usando un umbral podemos realizar el reconocimiento: $SCI(\vec{s}) > umbralMáx$ hay reconocimiento y si $SCI(\vec{s}) < umbralMín$ la imagen no puede ser reconocida, el caso intermedio es incertidumbre que se la puede tomar como no reconocido u aplicarle otro método.

Confirmada la inecuación del reconocimiento, se identifica la imagen con el valor del índice i del $Max\left\|\vec{\delta}(\vec{s})_i\right\|$

2.4.2. Reconocimiento ante la presencia de la oclusión. Si la imagen del rostro a reconocer presenta una oclusión,

no está bien alineado y presenta alguna clase de ruido, se trabaja de la siguiente manera:

$\vec{b} = A\vec{s} + \vec{e}$ donde $\vec{e} \in R^{Dx1}$. Matricialmente podemos representar $\vec{b} = [AI]\begin{pmatrix}\vec{s}\\ \vec{e}\end{pmatrix} = \Omega\ \vec{\mu}$ donde $\Omega \in R^{Dx(M+D)}$ y $\vec{\mu} \in R^{(M+D)x1}$.Al realizar el cambio de variables se trabaja buscando la solución de $\vec{b} = \Omega\vec{\mu}$ como en el apartado anterior. El resultado es el vector $\vec{\mu}$ en donde lo primeros M elementos correspondes a $\vec{s}$ y los restantes a $\vec{e}$. Al usar el error, Ganesh [16] llega a la conclusión que por más que se viole el PIR la solución genera un vector ralo y los resultados son prometedores porque se asemejan más a la realidad.

El programa elaborado para testear el método trabaja con oclusión para que sea más resistente dado que las imágenes que se usa para comparar su funcionamiento están en distintas condiciones de iluminación, pose, gestos, etc.

Se puede además usar el $\|\vec{e}\|_2$ para reforzar o para determinar si se puede reconocer o no una imagen. Se lo usa de forma similar al umbral en el método EigenFace.

3. Protocolo de Comparación

Se comparan los distintos métodos con las mismas bases de imágenes de rostros y en las mismas condiciones.

Se usó una computadora personal tipo portátil, marca BHG Positivo, modelo A497, con 4GB de memoria RAM y un procesador I7 multicore 2630QM. Con dos sistemas operativos, Windows 7 de 64 bits y Linux Ubuntu 12.04 LTS de 32bits (no se usó el de 64 bits por presentar problemas con los drivers). En cada sistema se dispone de Octave 3.8 y Matlab 12 (en el caso de Windows) y Octave 3.6 y Matlab 10 en Linux. Al utilizar las funciones básicas no hay inconvenientes al usar distintas versiones. La ventaja de trabajar con sistemas de 64 bits es la posibilidad de utilizar más de 4Gb de memoria. Al momento de realizar los testeos, la computadora no tenía otra aplicación corriendo en simultáneo.

3.1. Base de Imágenes

Se utilizan 5 bases de imágenes para testear los métodos codificados como algoritmos. Ellas son: Olivetti Research Laboratory (ORL) 400 fotos, Yale primera generación 165 fotos, Labeled Faces in the Wild (LFW) 13233 fotos, Flia y Flia Solo Recorte (FSR). Las 3 primeras están disponibles en internet y son usadas en la mayoría de los trabajos académicos. Además se generó una colección de imágenes propias, con fotos obtenidas de familiares, de ahí el nombre. Posee 630 imágenes de 5 personas, las fotografías fueron tomadas a color con un fondo uniforme, son 126 imágenes por personas, de las cuales hay con 2 fuentes de iluminación, 3 poses (frontal, ½ perfil y perfil), 7 distintas expresiones (normal, sorpresa, miedo, disgusto, cólera, felicidad y tristeza) y oclusión (lentes comunes y contra el sol). En al Figura 6 se observa algunas de ellas.

A excepción de ORL, a todas las imágenes se las recortó de manera manual manteniendo la relación de filas y columnas de ORL y posteriormente se las transformo a 112 x 92 en escala de grises. En FSR las imágenes solo fueron recortadas (de 800 x 657).

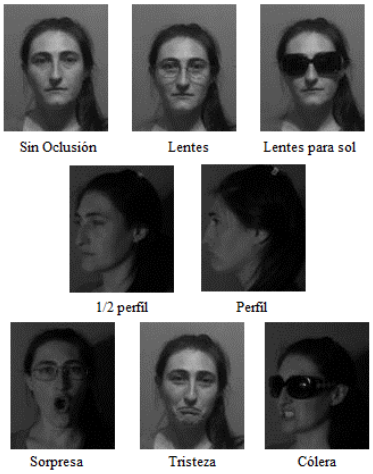


Figura 6 Ejemplos de la Base Flia

3.2. Protocolo de Análisis

Integran el protocolo de análisis: la estrategia del testeo, las imágenes de los rostros en el entrenamiento, obtención de umbral y reconocimiento, los distintos parámetros y los resultados buscados. Este se debe respetar en todos los métodos, para poder lograr resultados que permitan hacer una comparación lo más objetiva posible.

Cada parte del protocolo se corre en los programas Octave y MatLab y en los sistemas operativos en Windows 7 y Linux Ubuntu 12.04

Todos los métodos fueron codificados en archivos ".m" o scripts (compatibles para MatLab y Octave) siguiendo las ecuaciones presentadas en la sección anterior.

3.2.4 Elección de las Imágenes de Entrenamiento. Entrenar en el presente protocolo significa tomar un grupo de fotografías previamente identificadas de una base de imágenes y correr sobre ellas el algoritmo de

entrenamiento. Se hace notar que de todas las fotografías de una base de imágenes, algunas se usan para el entrenamiento, otras en la obtención de los umbrales y todas en el análisis del reconocimiento y tiempos de procesamiento. En la siguiente figura se observa gráficamente lo planteado.

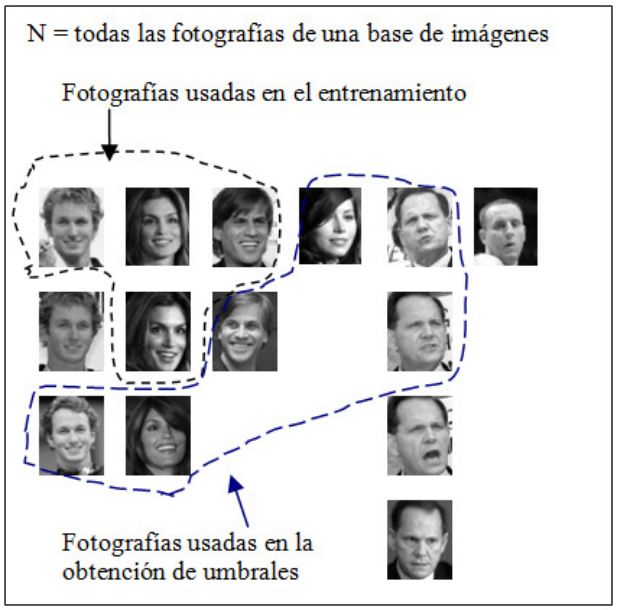


Figura 7 Ejemplo gráfico de las imágenes usadas en el entrenamiento, obtención de los umbrales y análisis del reconocimiento y tiempos de procesamientos

Se realizan 16 combinaciones, cada una de ellas se escribe en un archivo de texto en donde cada línea contiene la ubicación de una imagen y su código de identificación (formato CVS).

La cantidad de clases o personas (*c*) y la cantidad de imágenes de ellas (*M*) que se usan en el entrenamiento, se obtuvo de un script realizado a tal fin basado en la función *'rand'*. En la Tabla 1 se exhiben las 16 combinaciones. En total hay *16x5=80* archivos de imágenes de entrenamiento. Las imágenes de entrenamiento de Flia y FSR son las mismas (solo cambian el tamaño de las imágenes)

3.2.4 Obtención de los Umbrales. La bibliografía consultada no plantea una estrategia para obtenerlos. Para establecer los valores se utiliza un conjunto de imágenes la cuales están todas identificadas y en donde algunas corresponden a las personas usadas en el entrenamiento y otras no (ver Fig. 7). Para cada entrenamiento corresponde obtener los umbrales.

Al conjunto de imágenes se divide las imágenes en dos grupos: personas reconocidas y no reconocidas, a cada grupo se le aplica el algoritmo de reconocimiento y se calcula y almacena la distancia euclidiana o el SCI según corresponda. Posteriormente se obtienen los mínimos y máximos y se busca la región en donde no se presente un solapamiento o éste sea lo menor posible. Los valores de la región o regiones son los umbrales, con esta salvedad el algoritmo de reconocimiento trabaja con zonas de umbrales.

Tabla 1 Cantidad de imágenes y clases en cada archivo usado en el entrenamiento

Nro. de archivo	ORL		Yale		Flia y FSR		LFW	
	M	*c*	*M*	*c*	*M*	*c*	*M*	*c*
1	22	8	11	3	26	1	1373	1150
2	23	8	8	3	7	1	1373	1150
3	19	8	11	3	35	1	1348	1150
4	23	8	6	3	36	1	1379	1150
5	29	8	13	3	10	1	1788	1150
6	26	8	8	3	29	1	1602	1150
7	26	8	10	3	62	1	1563	1150
8	31	8	5	3	14	1	1740	1150
9	38	16	14	6	31	2	2757	2300
10	43	16	18	6	51	2	2799	2300
11	41	16	18	6	59	2	2747	2300
12	43	16	12	6	61	2	2905	2300
13	48	16	20	6	80	2	3095	2300
14	68	16	25	6	46	2	3378	2300
15	46	16	23	6	61	2	3056	2300
16	58	16	18	6	96	2	3214	2300

3.2.4 Análisis del Reconocimiento. Se utilizan todas las fotografías de cada una de las bases de imágenes en los 16 posibles casos de entrenamiento y su umbral o umbrales asociados. En cada caso se determina la cantidad teórica o ideal de reconocimientos (CRT): es cantidad desea que todo método debería alcanzar, la cantidad de reconocimientos usando el umbral (CRU): son los rostros que el método reconoce y la cantidad de reconocimientos correctos (CRC): es el subgrupo de CRU que no posee falsos positivos.

Se aclara que todas las imágenes de las bases están previamente identificadas. En CS se analiza el reconocimiento basado en el SCI y en el error por oclusión y se amplía el análisis usando las imágenes de entrenamiento para la obtención de los umbrales. En LBPH se toman dos casos de valores de grilla.

Se corren los algoritmos en MatLab y Octave y en los sistemas operativos Windows y Linux.

Tiempo de Procesamiento. Se toman los tiempos de realizar el entrenamiento y el reconocimiento sin el cálculo de los parámetros del apartado anterior. Al haber 16 casos en cada base de imágenes, 2 sistemas operativos y 2 programas, solamente se usan los tiempos promedios de cada caso. Se corren los algoritmos en MatLab y Octave y en los sistemas operativos Windows y Linux.

4. Resultados

Primero se presentan los resultados obtenidos en el reconocimiento y posteriormente los tiempos de procesamientos.

Se recuerda que en por cada caso se corrieron 16 pruebas y los valores establecidos en las tablas son los promedios.

Tabla 2 Análisis de Reconocimiento

ORL			
promedio	CRT	CRU	CRC
EigenFace	120,00	82,94	80,06
FisherFace	120,00	71,81	67,50
CS Error Umbral	120,00	36,50	36,50
CS Error Entre.	120,00	109,25	31,25
CS SCI Entre.	120,00	39,19	37,81
CS SCI Umbral	120,00	60,75	18,31
LBPH 2x2	120,00	21,438	15,25
LBPH 4x4	120,00	63,21	57,20
Yale			
promedio	CRT	CRU	CRC
EigenFace	49,50	33,50	29,69
FisherFace	49,50	31,25	28,06
CS Error Umbral	49,50	14,31	13,81
CS Error Entre.	49,50	36,50	11,50
CS SCI Entre.	49,50	37,69	19,63
CS SCI Umbral	49,50	60,00	24,88
LBPH 2x2	49,50	16,75	10,75
LBPH 4x4	49,50	28,40	27,20
Flia			
promedio	CRT	CRU	CRC
EigenFace	189,00	116,56	113,38
FisherFace	189,00	97,19	93,06
CS Error Umbral	189,00	44,31	44,00
CS Error Entre.	189,00	70,50	65,81
CS SCI Entre.	189,00	204,13	85,19
CS SCI Umbral	189,00	313,38	108,13
LBPH 2x2	189,00	94,44	88,50
LBPH 4x4	189,00	103,94	98,75
FSR			
promedio	CRT	CRU	CRC
EigenFace	189,00	115,00	110,81
FisherFace	189,00	96,81	92,13
CS Error Umbral	189,00	75,50	57,06
CS Error Entre.	189,00	77,69	43,88
CS SCI Entre.	189,00	184,25	71,13
CS SCI Umbral	189,00	441,00	107,75
LBPH 2x2	189,00	74,61	68,15
LBPH 4x4	189,00	82,11	76,04
LFW			
promedio	CRT	CRU	CRC
EigenFace	3920,25	2262,06	2258,94
FisherFace	3920,25	2264,31	2257,38
CS Error Umbral	No se calcula por el excesivo tiempo que demanda		
CS Error Entre.			
CS SCI Entre.			
CS SCI Umbral			
LBPH 2x2	3920,25	1712,22	1562,20
LBPH 4x4	3920,25	1921,31	1892,12

Tabla 3 Tiempos de Entrenamiento

ORL Tiempo de reconocimiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	0,57	3,85	4,27	4,70
FisherFace	0,79	2,39	4,29	4,90
CS Error Umbral	2,42	3,89	7,78	43,55
CS Error Entre.	2,72	3,86	8,41	43,55
CS SCI Entre.	2,42	3,47	7,81	43,03
CS SCI Umbral	2,71	4,10	7,81	43,02
LBPH 2x2	129,20	122,33	1883,43	1287,77
LBPH 4x4	117,13	119,19	1958,211253	1321,67
Yale Tiempo de reconocimiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	0,23	0,85	1,36	1,92
FisherFace	0,23	0,85	1,25	1,99
CS Error Umbral	0,67	1,41	2,49	6,94
CS Error Entre.	0,53	1,41	2,27	7,54
CS SCI Entre.	0,67	1,39	2,27	6,86
CS SCI Umbral	0,60	1,42	2,96	6,98
LBPH 2x2	50,59	49,34	792,68	572,12
LBPH 4x4	51,12	49,758	822,6741793	582,23

Flia Tiempo de reconocimiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	1,26	3,85	7,33	4,73
FisherFace	1,50	4,16	5,09	4,85
CS Error Umbral	5,45	8,91	15,68	92,92
CS Error Entre.	5,32	9,40	15,93	93,44
CS SCI Entre.	5,26	8,94	15,64	109,21
CS SCI Umbral	5,31	9,09	16,63	93,24
LBPH 2x2	193,73	196,23	3321,23	2028,24
LBPH 4x4	185,736	195,54	3298,172957	2081,63
FSR Tiempo de reconocimiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	22,55	27,52	66,23	57,78
FisherFace	33,02	38,45	64,76	58,76
CS Error Umbral	29,19	32,42	87,57	110,52
CS Error Entre.	29,35	31,95	88,06	110,01
CS SCI Entre.	29,23	32,62	87,68	94,25
CS SCI Umbral	29,35	32,39	88,06	109,21
LBPH 2x2	10698,9	10231,21	mayor a las 40Hs	
LBPH 4x4	9782,34	10388		
LFW Tiempo de reconocimiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	1212,18	561,91	1267,88	1167,67
FisherFace	582,31	813,29	2101,68	1849,48
CS Error Umbral	No puede calcular por excesivo tiempo. Se estima que cada caso demanda más de 200hs			
CS Error Entre.				
CS SCI Entre.				
CS SCI Umbral				
LBPH 2x2	3956,73	4134,88	mayor a las 15Hs	
LBPH 4x4	3987,03	4243,1		

Tabla 4 Tiempos de Reconocimiento

ORL Tiempo de Entrenamiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	0,14	0,62	0,86	0,67
FisherFace	0,86	0,76	3,34	0,70
CS Error Umbral	0,10	0,23	0,79	0,61
CS Error Entre.	0,11	0,22	0,85	0,61
CS SCI Entre.	0,10	0,21	0,79	0,62
CS SCI Umbral	0,12	0,25	0,79	0,62
LBPH 2x2	11,39	11,02	172,56	107,67
LBPH 4x4	11,21	11,00	178,31	112,56

Yale Tiempo de Entrenamiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	0,12	0,13	0,29	0,24
FisherFace	0,15	0,23	0,50	0,24
CS Error Umbral	0,05	0,10	0,48	0,36
CS Error Entre.	0,04	0,09	0,44	0,38
CS SCI Entre.	0,05	0,10	0,45	0,36
CS SCI Umbral	0,04	0,04	0,58	0,37
LBPH 2x2	4,21	4,07	65,19	43,21
LBPH 4x4	4,29	4,11	67,72	48,22
Flia Tiempo de Entrenamiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	0,91	0,62	4,96	0,98
FisherFace	0,75	1,11	5,91	0,98
CS Error Umbral	0,14	0,34	0,93	0,72
CS Error Entre.	0,13	0,36	0,93	0,74
CS SCI Entre.	0,13	0,34	0,94	0,85
CS SCI Umbral	0,13	0,37	0,99	0,72
LBPH 2x2	13,15	13,78	212,12	169,58
LBPH 4x4	13,08	13,694	228,162	177,28
FSR Tiempo de Entrenamiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	12,83	20,43	35,58	32,13
FisherFace	22,23	33,67	107,71	87,53
CS Error Umbral	2,30	6,39	22,16	17,73
CS Error Entre.	2,25	6,34	20,54	17,89
CS SCI Entre.	2,27	6,37	21,87	17,33
CS SCI Umbral	2,25	6,40	22,45	17,33
LBPH 2x2	671,81	691,43	mayor a 2Hs	
LBPH 4x4	698,89	715,72		
LFW Tiempo de Entrenamiento (seg)				
promedio	MatLab Win	MatLab Linux	Octave Win	Octave Linux
EigenFace	273,60	725,53	803,13	698,72
FisherFace	341,68	1059,11	1512,69	1311,23
CS Error Umbral	No se calcula por excesivo tiempo que requiere el reconocimiento.			
CS Error Entre.				
CS SCI Entre.				
CS SCI Umbral				
LBPH 2x2	678,90	701,78	mayor a 1Hs	
LBPH 4x4	684,78	702,785		

5. Conclusiones

Con respecto a los sistemas operativos, aquellos basados en 64bits permiten manejar memorias RAM superiores a 4GB, característica que es vital si se trabaja

con base de imágenes con una elevada cantidad de imágenes o ellas en alta resolución. Todo indica que los sistemas de 32Bits tienden a desaparecer, es más hoy Ubuntu 14.01 LTS recomienda descargar la versión de 64bits (al contrario de la versión 12.04). En ambos sistemas, los programas al igual que los sistemas operativos en todo momento respondieron correctamente (no se produjo el efecto conocido como "tildado de la PC"). Se detecta que los tiempos obtenidos con el sistema operativo Linux tienen menos dispersión comparado con Windows, de la lectura de la documentación técnica se debe a la forma de manejar la memoria virtual y Peque [17] hacen la comparativa.

En el entrenamiento cuando la cantidad de imágenes es pequeña y cada una de ellas con un número reducido de pixeles, Matlab sobre Windows y sobre Linux presentan tiempos similares. Se evidencia una mejora en el tiempo a favor del primero cuando el escenario es el opuesto. Octave presentó similitud en todos los casos, pero siempre demandando más tiempo que la competencia comercial.

Los métodos que permiten incorporar nuevos rostros al conjunto usado en el entrenamiento sin necesidad de recálculo (LBPH y CS) demandan entre 40 y 300 veces más tiempo que los otros.

En el reconocimiento está la mayoría de la carga computacional, cuando el algoritmo no se puede resolver usando los operadores matriciales y se debe usar bucles "for" o bloques iterativos, los tiempos aumentan de manera exponencial por la falta de optimización. Hay que diferenciar el comportamiento de Matlab con el de Octave, el primero está más preparado para trabajar en estos escenarios que la alternativa GNU.

EigenFace presenta la mejor tasa de reconocimiento, se esperaba que FisherFace la superara, esto no ocurre debido a que no se puede implementar con la formulación teórica (ver **2.2.1**) y se debe usar las proyecciones de las imágenes en PCA. Al realizarlo se pierde algo de información y no se logra la maximización basada en clases.

CS es un método que tiene como base la compresión, usar imágenes de mayor tamaño no representa un beneficio, Tiene dos inconvenientes: la recursividad para resolver el sistema de ecuaciones basado en la minimización de la norma l1 y la baja tasa de reconocimientos. La primera se puede mejorar si se resuelve con programación en paralelo y la segunda se la puede mejorar condicionando la matriz de medición y tomar muestras aleatorias en zonas de interés (Ej. zonas de los ojos). Hay correspondencia en los resultados al usar el SCI y el error basado en la oclusión. En ambos se concluye que la elección de las imágenes para la obtención del o de los umbrales no puede ser completamente aleatoria y se recomienda las del entrenamiento agregando algunas otras.

LBPH mejora la tasa de reconocimiento si se aumentan las grillas, usar los patrones universales en los histogramas tiene el beneficio de representar la información de interés con la menor cantidad información posible, otra ventaja es la simplicidad de las operaciones (suma, restas y comparación) y como contra la gran cantidad de ellas (por Ej. al calcular el operador LBP) lo que conlleva a tiempos de entrenamiento y reconocimiento elevados, estos tiempos se pueden reducir si se lo implementa en computación paralela. Con un valor de grillas 4x4 (usando 40 coeficientes por imagen) se logra una tasa próxima a EigenFace.

Usar Octave y/o Matlab tienen la ventaja de lograr algoritmos fiables con reducidos tiempos de desarrollo, ambos son interpretados lo que significa que si al mismo algoritmo se lo codifica en C/C++ los tiempos se reducen como mínimo en un orden de magnitud. Octave demanda en promedio unas 5 veces más tiempo que Matlab. En el caso de desarrollar una aplicación con fines comerciales, usar Octave en al etapa de experimentación y luego codificar en algún lenguaje de programación (que sea compilado) es una alternativa de bajo costo y bajos tiempos de desarrollo.

Del análisis de tiempos y la elección del sistema operativo-programa se desprende que no todos los métodos son aptos para trabajar en tiempo real (como el caso de vídeo identificación).

6. Referencias

[1]Toshiyuki Sakai, Makoto Nagao y Takeo Kanade, "Computer Analysis And Classification Of Photographs Of Human Face", *First USA-JAPAN Computer Conference,*1973 pp, 55-62

[2]R. Brunelli y T. Poggio, "Face Recognition through Geometrical Feactures", *IN EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV)*, pp. 792-800,1992

[3]Wiskott, L. and Fellous, J.-M. and Kuiger, N. y Von der Malsburg, C., "Face recognition by elastic bunch graph matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 775-779, 1997

[4]T.F. Cootes, G.J. Edwards y C.J. Taylor, "Active Appearance Model", *ECCV,* Vol. 2 ,pp.484-498,1998

[5]M. Turk and A. Pentland, "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, pp. 71–86, 1991

[6]Peter N. Belhumeurm, Joao P. Hespanha, David J. Kriegman, Eigenface vs. Fisherface: Recognition Using Class Specific Linear Projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 19, pp. 711-720, 1997

[7]Timo Ahonen , Abdenour Hadid , Matti Pietikäinen, "Face recognition with local binary patterns", *ECCV*, pp. 469-489, 2004

[8]Yang, Allan Y. and Wright, John and Ma, Yi and Sastry, S. Shankar, "Feature Selection in Face Recognition: A Sparse

Representation Perspective", *EECS Department, University of California, Berkeley* Vol. 19, pp. 99, 1997

[9]R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Annals of Eugenics 7*, pp. 179–188, 1936

[10]Timo Ojala, Matti Pietikäinen, David Harwood, "A comparative study of texture measures with classification based on feature distributions*", Pergamon: Pattern Recognitio*n, Vol. 29, pp. 51-59, 1996

[11]Emmanuel J Candès, "Compressive sampling", *Proceedings on the International Congress of Mathematicians: Madrid,* pp. 1433-1452, 2006

[12]David L. Donoho, "Compressed Sensing", *Department of Statistics of Stanford University*, September 2004

[13]Emmanuel Candès y Justin Romberg, "11-magic : Recovery of Sparse via Convex Programming", 2005

[14]Richard Baraniuk, "Compressive sensing", *IEEE Signal Processing Magazine*, Vol. 24, pp. 118-121, 2007

[15]Mark A. Davenport, Marco F. Duarte, Yonina C. Eldar, and Gitta Kutyniok, Compressed Sensing Theory and Applications, Cap 1,pp. 1-64, 2012

[16] Arvind Ganesh, Andrew Wagner, Zihan Zhou, Allen Y. Yang,Yi Ma, and John Wright, Compressed SensingTheory and Applications, Cap 12, pp. 515-539, 2012

[17] Miguel Sánchez de León Peque y Antón Kalinov Iliev, "Análisis Comparativo del Uso de la Memoria en Linux y Windows", *Estadística, ETSI Industriales, Universidad Politécnica de Madrid*, Enero de 2009

Análisis de Decisiones Multi Criterio Utilizando Agregación de Expertos basada en Calificaciones

Jorge Roa, Rubén Bernal, Marcelo Karanik
Grupo de Investigación Sobre Inteligencia Artificial
Universidad Tecnológica Nacional – Facultad Regional Resistencia
French 414 - (H3500CHJ) Resistencia - Chaco.
jorge@internea.com.ar,{rbernal,marcelo}@frre.utn.edu.ar

Palabras clave: Análisis de Decisión Multi Criterio, Operadores de Agregación, AHP, OWA.

Resumen

Por lo general, tomar decisiones en ámbitos financieros, como el otorgamiento de créditos, implica tratar con incertidumbre. La multiplicidad de factores a evaluar exige contar con métodos adecuados de Análisis de Decisión Multi-criterio. Si a esta situación se le añade la necesidad de contar con opiniones de varios expertos, el problema se vuelve extremadamente complejo para lograr un consenso apropiado. En el presente trabajo se estudia la influencia de la experiencia y la subjetividad de los evaluadores en un problema de Análisis de Decisión Multi-criterio. Para la evaluación, cada experto utiliza el Proceso Analítico Jerárquico para estructurar el problema y valorar las alternativas. Posteriormente se califica a los evaluadores según su experiencia y formación, agregando los resultados individuales con operadores OWA y medias ponderadas.

1. Introducción

La toma de decisiones está en todos los ámbitos de nuestra vida diaria. Seleccionar una de varias alternativas, es algo que se realiza cotidianamente, por ejemplo, qué transporte tomar para llegar al trabajo, qué colegio elegimos para nuestros hijos, qué almuerzo tomaremos hoy, cómo nos vestimos para una fiesta. Pero qué hay de las decisiones que se toman y que pueden influir en la vida de los demás ¿Cómo determinar si son acertadas dichas decisiones? Por ejemplo, cuando una entidad financiera tiene que decidir sobre el otorgamiento de créditos a familias en situación desfavorable. Este es el caso de la entidad llamada Banco Grameen [1], un banco social sin fines de lucro que tiene sede en Bangladesh, pero opera a nivel mundial con diferentes sucursales en todo el mundo. El Banco Grameen ha sido galardonado, junto con su creador el Prof. Muhammad Yunus, con el premio Nobel de la Paz en 2006. El Banco Grameen es una entidad sin fines de lucro que otorga créditos, sin garantía material, a familias pobres con el objetivo de incentivarlas a iniciar una actividad que les generen ganancias suficientes como para mejorar su calidad de vida mediante su propio esfuerzo. Dichos créditos deben ser devueltos luego para que ese dinero sea destinado a otras familias, generando de esta manera un círculo virtuoso. Tanto los voluntarios del Grameen, como los integrantes de las familias que desean acceder al crédito tienen a su disposición diferentes cursos de capacitación para llevar adelante sus emprendimientos, los cursos son sobre los sistemas de créditos, reglas de participación, agrupaciones y trabajo en grupos, etc. Claramente la selección correcta de los beneficiarios es un factor crítico y el proceso de selección debe lidiar con niveles de incertidumbre e intentar minimizar la cantidad de malas decisiones para que el sistema pueda seguir funcionando adecuadamente. Por ello, uno de los problemas de otorgar créditos es la evaluación y selección de los beneficiarios, es decir, cómo seleccionar a las familias beneficiarias que, luego de haber iniciado su emprendimiento, reintegrarán ese dinero para que pueda ser otorgado a otros beneficiarios.

La situación descripta anteriormente es un problema de Análisis de Decisión Multi-Criterio (del inglés Multi Criteria Decision Analysis – MCDA) y existen muchos métodos para poder tratarlo. Uno de ellos es Proceso Analítico Jerárquico (del inglés Analytic Hierarchical Process – AHP) [2] desarrollado por Thomas Saaty. Este método considera la descomposición jerárquica del problema en niveles de criterios y alternativas que, luego de un proceso de comparación entre pares, resume en un ranking el orden de mérito de las opciones disponibles. Las aplicaciones de AHP son múltiples y en muy diversas áreas. Algunos trabajos relacionados con la propuesta de este artículo, incluyen la utilización de AHP, por ejemplo, en manufactura integrada [3], evaluación de inversiones de tecnología [4], planificación estratégica

[5], logística [6], selección de rutas de transporte [7], selección de proveedores [8], selección de tecnologías de acceso a internet [9] y selección de proyectos de Tecnologías de la Información [10, 11], entre muchas otras. La gran variedad de trabajos que utilizan AHP evidencia la versatilidad del método para dar soporte a las decisiones.

Si bien la utilización de AHP es de gran utilidad, por lo general devuelve como resultado la opinión (expresada como ranking) de un solo experto. Para poder tener la visión más amplia de varias opiniones se puede presentar el problema a varios expertos, que utilicen AHP y luego resumir los resultados utilizando métodos de agregación [12]. Este trabajo propone la utilización de AHP por parte de varios evaluadores para seleccionar entre distintas opciones de otorgamiento de microcréditos y agregar los resultados obtenidos. En la Sección 2, se hace una breve descripción de AHP. Luego, en la Sección 3, se mencionan los operadores de agregación más utilizados. La descripción del escenario y los resultados de las simulaciones se muestran en la Sección 4. Finalmente, en la Sección 5 se enuncian las conclusiones obtenidas del trabajo realizado y las líneas futuras de investigación.

2. Proceso Analítico Jerárquico (AHP)

Como se ha mencionado en la sección anterior, tomar decisiones complejas requiere mucha atención sobre varios aspectos simultáneamente. Esto significa que se deben identificar cuáles son las variables relevantes que intervienen en un problema dado y cómo se relacionan entre sí. Una vez que se tiene un panorama general de la estructura del problema, hay que establecer la importancia relativa entre los criterios que se seleccionaron y cómo influyen sobre las alternativas disponibles. Este análisis de decisión multi-criterio proporciona como resultado un orden de prioridades o preferencias sobre las alternativas que luego se utiliza para seleccionar el curso de acción adecuado.

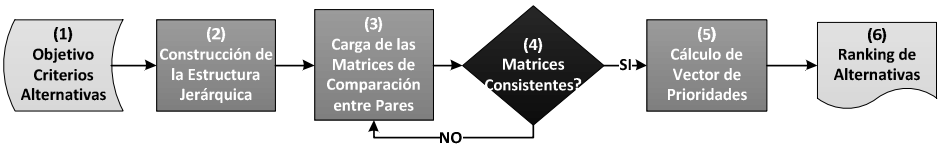


Figura 1. Proceso de Toma de Decisiones con AHP.

A medida que aumentan los criterios de evaluación, el análisis se torna más complicado y si a esta situación se le suman las limitaciones cognitivas propias de las personas, es probable que el decisor necesite herramientas que le ayuden a elegir la mejor alternativa disponible. Una de esas técnicas es AHP [2] y se utiliza para establecer un orden de prioridades de alternativas sobre la evaluación de múltiples criterios.

En la Fig. 1 se muestra el proceso completo desde la definición de los objetivos, criterios y alternativas para un problema particular hasta la obtención del ranking de alternativas.

En la Fig. 2 se muestra una jerarquía AHP como una abstracción de la estructura de un problema que se quiere representar. Para su construcción (Fig. 1 (2)), el objetivo se coloca en la raíz de un árbol de jerarquías y los criterios y las alternativas que serán analizadas derivan de ella [13]. Además de las jerarquías, se debe tener en cuenta que los elementos de un nivel deben poder ser comparados según criterios de nivel superior.

El proceso de comparación comienza desde los niveles de criterios hacia los niveles de alternativas, utilizando matrices de comparación de pares, en las cuales el decisor completa cada elemento de la porción triangular superior de la matriz (los elementos sobre la diagonal principal) con un valor obtenido de una escala verbal de juicios [2]: igual importancia, moderadamente más importante, fuertemente más importante, muy fuertemente más importante y extremadamente más importante (Fig. 1 (3)). A esta escala se asocian los juicios representados con valores numéricos correspondiente a cada uno de los mencionados anteriormente: (1, 3, 5, 7, 9) y valores de compromiso (2, 4, 6, 8). El resto de los valores de la matriz se completa con los valores recíprocos de los juicios. De esta manera las matrices de comparaciones entre pares son matrices recíprocas (condición de reciprocidad).

En [14] se define para AHP, además de la condición de reciprocidad, tres condiciones más: condición de Homogeneidad (permite colocar en cada nivel las características que puedan ser comparadas); condición de Dependencia (permite establecer las conexiones entre los niveles creando una jerarquía de dependencias externas entre niveles) y condición de Expectativas (permite tener una visión completa del problema incluyendo todos los aspectos relevantes).

En los casos en que el número de elementos de las matrices de comparación entre pares es bajo, el proceso

es simple pero, si este número crece, completar las matrices no es una tarea sencilla porque los juicios podrían volverse inconsistentes. Hay varias maneras de medir la consistencia de una matriz recíproca de comparación entre pares [2, 15, 16]. En este trabajo se utiliza la medida de consistencia basada en la relación multiplicativa propuesta por Saaty debido a que es frecuentemente usada para testear la consistencia de las matrices resultantes de algún proceso de reconstrucción [17, 18].

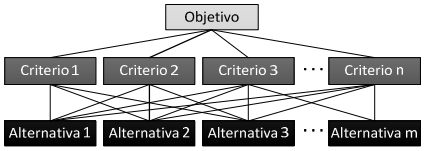


Figura 2. Esquema Niveles Jerárquicos

Saaty ha demostrado que la matriz recíproca A es perfectamente consistente si el autovalor máximo de A es igual al orden de la matriz [2]. El índice de consistencia (Consistency Index – CI) está dado por:

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{1}$$

donde λ_{max} es el autovalor máximo de la matriz A y n es su orden.

Además, se define la Medida de Consistencia (CR) como:

$$CR = \frac{\text{CI}}{RI} \leq 0{,}1 \tag{2}$$

donde RI es el índice aleatorio (Random Index – RI) obtenido de 500 matrices recíprocas positivas generadas de manera aleatoria que contienen valores de la escala propuesta por Saaty (Tabla 1).

Tabla 1. Valores de RI.

n	3	4	5	6	7	8
RI	0.58	0.9	1.12	1.24	1.32	1.41

Para determinar si una matriz de comparación entre pares es consistente (Fig. 1 (4)), su CR debe estar por debajo de 0.1 [18], si esta condición no se cumple el decisor debe modificar los valores de comparación en la matriz correspondiente.

Cuando las matrices de comparación entre pares están completas se realiza el proceso de cálculo de prioridades (Fig. 1 (5)). Para ello se calcula el autovector asociado al autovalor máximo de cada matriz, utilizando por ejemplo el conocido método de las potencias. El autovector debe ser normalizado antes de ser utilizado en el nivel inferior. Este procedimiento se inicia en el nivel de criterios y se obtiene el ranking relativo de los mismos.

Una vez calculados los valores de las prioridades de los criterios se construyen las matrices de comparación entre pares para las alternativas por cada criterio y nuevamente se calculan y normalizan los autovectores asociados a los autovalores máximos de cada una de las matrices implicadas. Cada autovector de las alternativas con respecto a los criterios representa una columna en una nueva matriz que será utilizada para determinar los valores finales de prioridades asociados a cada alternativa.

Para obtener el ranking final (Fig. 1 (6)) de alternativas, se multiplica la matriz formada por todos los autovectores normalizados obtenidos para cada alternativa respecto a cada criterio por el autovector asociado a la matriz de comparación de pares de los criterios. El vector obtenido indica el valor de prioridades para cada alternativa, recomendándose utilizar la de mayor valor.

3. Operadores de Agregación

Los operadores de agregación son funciones especiales con entradas pertenecientes a un subdominio $\mathbb{I}$ de la recta real. La característica principal de todas las funciones de agregación es que deben ser acotadas monótonas crecientes. Esto significa que ante valores de entrada crecientes no pueden obtenerse respuestas decrecientes y que los valores de agregación obtenidos deben pertenecer a la misma escala de los valores de entrada respectivamente.

Formalmente una función de agregación en un intervalo $\mathbb{I}^n$ se define como [12]:
$A^{(n)}: \mathbb{I}^n \to \mathbb{I}$ tal que:

i. es no decreciente (en cada variable);

ii. cumple las siguientes condiciones de

$$\inf_{x \in \mathbb{I}^n} A^{(n)}(\boldsymbol{x}) = \inf \mathbb{I} \quad \text{y} \quad \sup_{x \in \mathbb{I}^n} A^{(n)}(\boldsymbol{x}) = \sup \mathbb{I} \tag{3}$$

$$\text{si } \mathbb{I} = [a, b], \quad A^{(n)}(\mathbf{a}) = a \quad \text{y} \quad A^{(n)}(\mathbf{b}) = b \tag{4}$$

$$\text{donde} \quad \mathbf{a} = (a, \dots, a) \quad \text{y} \quad \mathbf{b} = (b, \dots, b)$$

iii. $A^{(1)}(x) = x$ para todo $x \in \mathbb{I}$

El entero n es el número de variables de entrada y representa la cardinalidad de la función de agregación. Como ejemplos de estas funciones podemos considerar la media aritmética, media geométrica, la función Mínimo, la función Máximo, OWA [19], etc.

3.1. Media aritmética ponderada

La media aritmética ponderada WAM (del inglés Weighted Arithmetic Mean) de dimisión n, es una medida de tendencia central y puede definirse como un operador de agregación $M: \mathbb{R}^n \to \mathbb{R}$ con un vector de pesos asociado $w = [w_1, w_2, \dots, w_n]$ tal que $\sum_{i=1}^{n} w_i = 1$, $0 \leq w_i \leq 1$ y donde:

$$M(a_1, a_2, \dots, a_n) = \sum_{i=1}^{n} a_i w_i \quad (5)$$

Claramente el orden de los elementos no interesa, sino que a cada característica se le asocia un peso relativo. Esto permite otorgar distintos niveles de importancia a las características. Una alternativa para este tipo de agregación son los operadores OWA que tienen en cuenta el orden de los elementos a agregar.

3.2. Operador OWA

El operador OWA (del inglés Ordered Weighted Averaging), es una clase de función de agregación parametrizada que permite generar el operador OR (*Max*), el operador AND (*Min*) y cualquier otro operador entre ellos como la media. El operador OWA de dimisión n es un mapeo $A: \mathbb{R}^n \to \mathbb{R}$ con un vector de pesos asociado $w = [w_1, w_2, \dots, w_n]$ tal que $\sum_{i=1}^{n} w_i = 1,\ 0 \leq w_i \leq 1$ y donde:

$$A(a_1, a_2, \dots, a_n) = \sum_{i=1}^{n} b_i w_i \quad (6)$$

donde b_i es el i-esimo elemento más grande de la colección $(a_1, a_2, \dots, a_n)$.

Este operador tiene diferentes comportamientos basados en el vector de pesos asociado. Por ejemplo si se agregan n valores con los vectores de pesos $[1, 0, 0, \dots, 0]$, $[0, 0, 0, \dots, 1]$ y $[1/n, 1/n, 1/n, \dots, 1/n]$, produce los operadores *max*, *min* y media aritmética respectivamente.

Yager introdujo el concepto de *orness* [19] que caracteriza a este operador y se define como:

$$orness(w_1, w_2, \dots, w_n) = \sum_{i=1}^{n} \frac{n-i}{n-1} w_i \quad (7)$$

Agregaciones OWA con *orness* mayor o igual a $1/2$, se consideran del tipo OR y menor o igual que $1/2$, del tipo AND. Las primeras corresponden a preferencias más bien optimistas mientras que las segundas representan preferencias pesimistas.

La agregación con un vector de pesos decreciente $w_1 \geq w_2 \geq \cdots \geq w_n$ determina un vector optimista (del tipo OR), mientras que si los pesos son crecientes $w_1 \leq w_2 \leq \cdots \leq w_n$, define un vector pesimista (del tipo AND).

La idea fundamental de los operadores de agregación es obtener un valor representativo de varios valores. Por esto para el problema planteado en este trabajo se pretende utilizarlos a fin de orientar al decisor sobre qué alternativa es la adecuada.

4. Simulaciones y Resultados

4.1 Escenario

El problema tratado en este trabajo, es un caso de estudio hipotético, que consiste en la evaluación y selección de la/las familias a las cuales se le otorgará un crédito, bajo las condiciones indicadas en la Sección 1. Se trata de tomar la decisión que beneficie a una familia dado que el presupuesto no es suficiente para todas las postulantes.

La primera alternativa es entregar el beneficio a la familia más necesitada económicamente (la más vulnerable), pero lo que puede ocurrir es que esta no sea la que realmente lo merezca o que aproveche mejor la asistencia recibida. Incluso puede ser que esta familia no se preocupe en devolver el crédito y lo utilice para consumo al no tener en cuenta la finalidad del beneficio.

Ahora bien, si realmente se quiere tener en cuenta la equidad, deben considerarse diferentes criterios de selección que definan un perfil para las diferentes alternativas, tal como lo indica la Tabla 2.

Tabla 2: Criterios de selección de alternativas.

Criterio	Nombre	Descripción
C1	Ingresos Monetarios	Ingresos monetarios en la familia.
C2	Asistencia a Capacitaciones	Asistencia a las capacitaciones sobre micro-emprendimientos.
C3	Emprendimientos Actuales	Emprendimientos realizados actuales o anteriores.
C4	Cantidad de Integrantes c/estudio	Cantidad de integrantes con estudios primarios, secundarios o terciarios.
C5	Logros Académicos Hijos	Logros académicos de los hijos.
C6	Logros no Académicos Hijos	Logros no académicos de los hijos.

Familias postulantes a recibir el beneficio:

Para poder establecer las características de las Familias postulantes se hace una breve descripción de la situación general de cada una. Las siguientes tres familias son postulantes para acceder a microcréditos:

Familia 1(F1)

Esta familia consta de 3 integrantes: el padre, la madre y una hija de 10 años. El padre recibe un jornal no formal y eventual, en trabajos de construcción. Por su parte la madre recibe ingresos eventuales como modista.

El padre y la madre han manifestado que están dispuestos a asistir a las capacitaciones sobre micro-emprendimientos.

El padre de familia tiene la primaria completa pero no ha cursado la escuela secundaria. La madre, en cambio, tiene la primaria completa y concurre a un instituto de nivel adulto para terminar el nivel secundario. La hija está en 5to grado de primaria y figura en la segunda posición en el cuadro de honor, también ha sido designada para participar en los juegos deportivos inter-escolares, en representación de su escuela.

Familia 2 (F2)

Familia que consta de 4 integrantes, el padre, la madre y dos hijos varones de 15 y 18 años.

En este caso el padre, cuenta con instrucción primaria pero no secundaria. Recibe remuneraciones informales y ocasionales por los trabajos de jardinería que ofrece en diferentes lugares de la ciudad, en este trabajo lo ayuda su hijo mayor quien no ha terminado la secundaria.

La madre, quien también tiene instrucción primaria pero no secundaria, hace comidas y postres y los ofrece en su barrio y zonas aledañas.

El hijo menor asiste a la escuela secundaria por la tarde y por la mañana sale a ofrecer los productos de su madre. No tiene logros académicos o no académicos hasta el momento, pero nunca ha repetido de grado. La madre, por su parte, está dispuesta a asistir a los cursos de micro-emprendimientos, pero el padre alega no tener tiempo para asistir a dicha actividad.

Familia 3 (F3)

La tercera familia consta de 4 integrantes, la madre, y 3 hijos, dos hijos varones y una hija, de 15, 17 y 20 años respectivamente.

La madre, que es responsable del hogar, trabaja en la cocina de un restaurante por lo cual recibe una remuneración mensual, tiene secundaria completa y está interesada en emprender un pequeño negocio de comidas buscando un mejor pasar económico.

Los dos hijos mayores no han terminado la secundaria, el mayor trabaja como cobrador de cuotas en un club, el otro no trabaja ni estudia hasta el momento. La hija estudia una tecnicatura en programación y cuenta con una beca para terminar sus estudios, su promedio de calificaciones es 7.3, según el certificado analítico de materias aprobadas.

Evaluadores

A continuación se describe el perfil de los evaluadores involucrados en el proceso de decisión.

Evaluador 1 (E1)

Es una directora de escuela primaria de 55 años de edad que trabaja en un colegio religioso y desempeña funciones en una fundación que otorga créditos, se decidió a participar en estas actividades por su afán de ayudar a los más necesitados.

Evaluador 2 (E2)

Es un ingeniero civil y magister en administración de empresas de 42 años, es docente universitario y ejerce su profesión en el ámbito particular. En su tiempo libre participa en una fundación que otorga créditos. Su decisión de participar en esta fundación, nace por el interés de ayudar a los demás y por su particular disconformidad con las obras de caridad, que según su punto de vista, generan mayores índices de pobreza y no estimulan a la población más necesitada a salir de su estado actual.

Evaluador 3 (E3)

Es un abogado, de 38 años, con un posgrado en Ciencias Políticas, ejerce sus funciones en la administración pública, específicamente en una secretaría de Desarrollo Social. Tiene como objetivo desarrollar una carrera política y postularse en futuras elecciones. Participa también en una fundación que otorga créditos para conocer mejor y ayudar a la población más humilde de su ciudad.

Ponderación de los evaluadores

Con el objetivo de lograr que las decisiones obtenidas por parte de los evaluadores sean lo más objetivas posibles, se realizará una ponderación de los 3 evaluadores en base a tres perfiles, Formación Académica (FA), Experiencia en Gestión (EG) y en Asistencialismo Social (EAS). Cada criterio será ponderado favoreciendo la Experiencia en Gestión, luego Formación Académica y finalmente el Asistencialismo Social, debido a la necesidad de optimizar el retorno de los préstamos otorgados.

De esta forma se obtienen las valoraciones de la Tabla 3.

Tabla 3. Ponderación de los Evaluadores

Evaluador	FA – 0.3	EG – 0.5	EAS – 0.2
E1	30	60	35
E2	50	90	30
E3	60	30	65

4.1 Evaluación de las Alternativas

Cada evaluador utiliza AHP para obtener el ranking de las familias postulantes. De esta manera expresan sus valoraciones en dos niveles de comparación entre pares: comparación entre los criterios mostrados en la Tabla 2 y comparación entre las Familias. Es decir, mediante valoraciones, indican cuáles son sus preferencias respecto de los elementos a evaluar.

En la Tabla 4 se muestran los valores de comparación entre criterios del Evaluador 1 (E1). Puede observarse que la tasa de consistencia (CR) se encuentra por debajo de 0.1, lo que indica que sus valoraciones tienen un grado

de consistencia aceptable. Además, también en la Tabla 4, puede observarse el orden de prioridades de criterios obtenido para E1. En este caso el criterio más importante es C1 (Ingresos Monetarios) y el segundo es C2 (Asistencia a Capacitaciones).

Tabla 4. Evaluación de Criterios del Evaluador 1.

	C1	C2	C3	C4	C5	C6	Vector de Prioridades
C1	1	3	9	5	6	7	0.4436
C2	1/3	1	7	5	6	7	0.2981
C3	1/9	1/7	1	1/6	1/4	1/2	0.0273
C4	1/5	1/5	6	1	3	5	0.1286
C5	1/6	1/6	4	1/3	1	1	0.0581
C6	1/7	1/7	2	1/5	1	1	0.0444
λ_{max} = 6.5050, CI = 0.1010, CR = 0.0814							

Tabla 5. Evaluación de las Familias según cada Criterio para Evaluador 1.

Ingresos Monetarios	F1	F2	F3	Vect. De Prioridades
F1	1	3	1/3	0.2583
F2	1/3	1	1/5	0.1047
F3	3	5	1	0.6370
λ_{max} = 3.0385, CI =0.0193, CR=0.0332				

Asistencia a Capacitaciones	F1	F2	F3	Vect. De Prioridades
F1	1	3	1	0.4286
F2	1/3	1	1/3	0.1429
F3	1	3	1	0.4286
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Emprendimientos Actuales	A	B	C	Vect. De Prioridades
F1	1	1	1	0.3333
F2	1	1	1	0.3333
F3	1	1	1	0.3333
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Cant. Integrantes c/estudio	A	B	C	Vect. De Prioridades
F1	1	1	3	0.4286
F2	1	1	3	0.4286
F3	1/3	1/3	1	0.1429
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Logros Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	3	1	0.4286
F2	1/3	1	1/3	0.1429
F3	1	3	1	0.4286
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Logros no Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	3	3	0.6000
F2	1/3	1	1	0.2000
F3	1/3	1	1	0.2000
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Tabla 6. Resultados Finales del Evaluador 1.

	C1	C2	C3	C4	C5	C6	Priorid. Finales
	0.4436	0.2981	0.0273	0.1286	0.0581	0.0444	
F1	0.2583	0.4286	0.3333	0.4286	0.4286	0.6000	0.3580
F2	0.1047	0.1429	0.3333	0.4286	0.1429	0.2000	0.1704
F3	0.6370	0.4286	0.3333	0.1429	0.4286	0.2000	0.4716

En la Tabla 5 se muestran las valoraciones de comparación entre las Familias. Como en el caso de la comparación de criterios todas las matrices tienen una tasa de consistencia menor a 0.1, lo que indica que dichas valoraciones tienen un nivel de inconsistencia aceptable. Puede observarse además que el orden de prioridad varía por cada criterio. Finalmente en la Tabla 6 se muestra el ranking final de la evaluación. Este ranking se obtiene, como se mencionó en la Sección 2, mediante la multiplicación de la matriz cuyas columnas son los rankings individuales de las Familias (según cada Criterio) por el vector de criterios. De esta manera, para E1, la familia que debería ser beneficiaria del crédito es F3. Luego le siguen F1 y F2.

De manera similar, los resultados obtenidos para E2 se muestran en las Tablas 7, 8 y 9. Para E2 los criterios más importantes, a diferencia de E1, son C4 (Cantidad de Integrantes c/Estudios) y C5 (Logros Académicos de los Hijos). Luego de la valoración de E2, los resultados obtenidos el orden de mérito es: F1, F3, F2.

Finalmente, la evaluación de E3, se muestra en las Tablas 10, 11 y 12. También, como en las evaluaciones

anteriores, la consistencia de las valoraciones es adecuada para todas las matrices. Los criterios más importantes para E3 son C1 (Ingresos Monetarios) y C4 (Cantidad de Integrantes c/Estudios) y el orden final de mérito es F3, F1, F2.

Puede observase que los órdenes de mérito coinciden para los evaluadores E1 y E3. Esto indica que si bien los criterios utilizados para valorar son distintos, ambos evaluadores tienen similitudes en sus consideraciones. En cambio las apreciaciones de E2, difieren sustancialmente del resto.

Tabla 7. Evaluación de Criterios del Evaluador 2.

	C1	C2	C3	C4	C5	C6	Vector de Prioridades
C1	1	1/5	1/6	1/9	1/8	1/3	0.0246
C2	5	1	1/5	1/6	1/5	3	0.0745
C3	6	5	1	1/4	1/4	4	0.1566
C4	9	6	4	1	2	7	0.3961
C5	8	6	4	1/2	1	5	0.3027
C6	3	1/3	1/4	1/7	1/5	1	0.0455

λ_{max} = 6.6029, CI = 0.1206, CR = 0.0972

Tabla 8. Evaluación de las Familias según cada Criterio para Evaluador 2.

Ingresos Monetarios	F1	F2	F3	Vect. De Prioridades
F1	1	3	1	0.4434
F2	1/3	1	1/2	0.1692
F3	1	2	1	0.3874

λ_{max} =3.0183 , CI =0.0091, CR=0.0158

Asistencia a Capacitaciones	F1	F2	F3	Vect. De Prioridades
F1	1	3	2	0.5396
F2	1/3	1	1/2	0.1634
F3	1/2	2	1	0.2970

λ_{max} = 3.0092, CI =0.0046, CR=0.0079

Emprendimientos Actuales	A	B	C	Vect. De Prioridades
F1	1	5	7	0.7306
F2	1/5	1	3	0.1884
F3	1/7	1/3	1	0.0810

λ_{max} = 3.0649, CI =0.0324, CR=0.0559

Cant. Integrantes c/estudio	A	B	C	Vect. De Prioridades
F1	1	7	2	0.5821
F2	1/7	1	1/3	0.0695
F3	1/2	3	1	0.3484

λ_{max} = 3.0324, CI =0.0162, CR=0.0279

Logros Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	7	2	0.5821
F2	1/7	1	1/3	0.0695
F3	1/2	3	1	0.3484

λ_{max} = 3.0324, CI =0.0162, CR=0.0279

Logros no Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	3	3	0.6000
F2	1/3	1	1	0.2000
F3	1/3	1	1	0.2000

λ_{max} = 3.0000, CI =0.0000, CR=0.0000

Tabla 9. Resultados Finales del Evaluador 2.

	C1	C2	C3	C4	C5	C6	Priorid. Finales
	0.0246	0.0745	0.1566	0.3961	0.3027	0.0455	
F1	0.4434	0.5396	0.7306	0.5821	0.5821	0.6000	0.5996
F2	0.1692	0.1634	0.1884	0.0695	0.0695	0.2000	0.1035
F3	0.3874	0.2970	0.0810	0.3484	0.3484	0.2000	0.2969

Tabla 10. Evaluación de Criterios del Evaluador 3.

	C1	C2	C3	C4	C5	C6	Vector de Prioridades
C1	1	6	8	2	3	4	0.3980
C2	1/6	1	2	1/3	1/3	1/2	0.0638
C3	1/8	1/2	1	1/5	1/4	1/3	0.0394
C4	½	3	5	1	2	4	0.2490
C5	1/3	3	4	1/2	1	2	0.1552
C6	¼	2	3	1/4	1/2	1	0.0946

λ_{max} = 6.1256, CI = 0.0251, CR = 0.0203

Tabla 11. Evaluación de las Familias según cada Criterio para Evaluador 3.

Ingresos Monetarios	F1	F2	F3	Vect. De Prioridades
F1	1	3	1/3	0.2583
F2	1/3	1	1/5	0.1047
F3	3	5	1	0.6370
λ_{max} = 3.0385, CI =0.0193, CR=0.0332				

Asistencia a Capacitaciones	F1	F2	F3	Vect. De Prioridades
F1	1	3	4	0.6337
F2	1/3	1	1	0.1919
F3	1/4	1	1	0.1744
λ_{max} = 3.0092, CI =0.0046, CR=0.0079				

Emprendimientos Actuales	A	B	C	Vect. De Prioridades
F1	1	1	1	0.3333
F2	1	1	1	0.3333
F3	1	1	1	0.3333
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Cant. Integrantes c/estudio	A	B	C	Vect. De Prioridades
F1	1	2	1/2	0.2970
F2	1/2	1	1/3	0.1634
F3	2	3	1	0.5396
λ_{max} = 3.0092, CI =0.0046, CR=0.0079				

Logros Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	2	1/3	0.2297
F2	1/2	1	1/5	0.1220
F3	3	5	1	0.6483
λ_{max} = 3.0037, CI =0.0018, CR=0.0032				

Logros no Académicos Hijos	A	B	C	Vect. De Prioridades
F1	1	5	5	0.7143
F2	1/5	1	1	0.1429
F3	1/5	1	1	0.1429
λ_{max} = 3.0000, CI =0.0000, CR=0.0000				

Tabla 12. Resultados Finales del Evaluador 3.

	C1	C2	C3	C4	C5	C6	Priorid. Finales
	0.398	0.0638	0.0394	0.249	0.1552	0.0946	
F1	0.2583	0.6337	0.3333	0.2970	0.2297	0.7143	0.3335
F2	0.1047	0.1919	0.3333	0.1634	0.1220	0.1429	0.1402
F3	0.6370	0.1744	0.3333	0.5396	0.6483	0.1429	0.5263

Agregación de los resultados

Con la finalidad de obtener los resultados finales se utiliza una agregación a partir de los datos consignados en la Tabla 3. Para ello se calculan los valores especificados mediante la Media Aritmética (MA), Promedio Ponderado (PP) y operadores OWA optimista y pesimista según la ordenación del vector de pesos (Tabla 13).

Tabla 13. Agregación de Ponderación de Evaluadores

	FA	EG	EAS	MA	PP	OWA Opt	OWA Pes
Imp	0,3	0,5	0,2	(calif)	(calif)	(calif)	(calif)
E1	30	60	35	33.33	46.00	46.50	37.50
E2	50	90	30	33.33	66.00	66.00	48.00
E3	60	30	65	33.33	46.00	56.50	46.00

Con los valores obtenidos se agregan los resultados parciales de cada evaluador. La Tabla 14 muestra la agregación utilizando media aritmética (igual peso para todos los evaluadores). Se puede observar que mediante este proceso de agregación, la Familia beneficiaria debe ser F3 (valor 0.4316 del Ranking Final). Esto tiene sentido ya que dos de los tres evaluadores coincidieron que F3 era la adecuada para recibir el beneficio según los cálculos realizados con AHP.

Tabla 14. Agregación con Vector de Pesos Agregado Mediante Media Aritmética.

	E1	E2	E3	
(calif)	33.33	33.33	33.33	Ranking
w	0.33	0.33	0.33	Final
F1	0.3580	0.5996	0.3335	0.4304
F2	0.1704	0.1035	0.1402	0.1380
F3	0.4716	0.2969	0.5263	**0.4316**

Al utilizar el operador PP, la situación es distinta, se puede observar en Tabla 15 que la Familia ganadora es F1 (valor 0.4518 del Ranking Final).

Tabla 15. Agregación con Vector de Pesos Agregado Mediante Promedio Ponderado.

	E1	E2	E3	
(calif)	46.00	66.00	46.00	Ranking
w	0.29	0.42	0.29	Final
F1	0.3580	0.5996	0.3335	**0.4518**
F2	0.1704	0.1035	0.1402	0.1337
F3	0.4716	0.2969	0.5263	0.4145

Esta situación ocurre debido a que la calificación de E2 es claramente más elevada que las calificaciones de E1 y E3. Debido a ello, su opinión se vuelve más importante y por lo tanto ante la paridad de los valores, toma mayor importancia F1.

Al utilizar OWA (Tablas 16 y 17) se obtiene también a F1 como ganadora (valores 0.4442 y 0.4376 del Ranking Final respectivamente). Se puede observar que inclusive con un criterio pesimista las calificaciones de E2, prevalecen sobre las de E1 y E3.

Tabla 16. Agregación con Vector de Pesos Agregado Mediante OWA optimista.

	Ev1	Ev2	Ev3	
(calif)	**46.50**	**66.00**	**56.50**	**Ranking**
w	**0.28**	**0.39**	**0.33**	**Final**
F1	0.3580	0.5996	0.3335	**0.4442**
F2	0.1704	0.1035	0.1402	0.1342
F3	0.4716	0.2969	0.5263	0.4217

Tabla 17. Agregación con Vector de Pesos Agregado Mediante OWA pesimista.

	Ev1	Ev2	Ev3	
(calif)	**37.50**	**48.00**	**46.00**	**Ranking**
w	**0.29**	**0.37**	**0.35**	**Final**
F1	0.3580	0.5996	0.3335	**0.4376**
F2	0.1704	0.1035	0.1402	0.1354
F3	0.4716	0.2969	0.5263	0.4270

Los resultados obtenidos muestran que la decisión debe ser otorgar el crédito a F1 a pesar que solo uno de los evaluadores da como ganadora a esa familia.

5. Conclusiones

En este artículo se ha presentado una propuesta de Análisis de Decisión Multi-criterio para Múltiples Expertos que integra AHP y Operadores de Agregación para situaciones de incertidumbre.

Los resultados obtenidos muestran que el procedimiento propuesto trata adecuadamente las situaciones sesgadas por la subjetividad de los decisores y no solamente por el conocimiento o experiencia sobre la temática a evaluar.

Si se pondera a los decisores en base a sus capacidades de evaluación objetiva, se obtienen resultados que hacen que la alternativa seleccionada por el evaluador más objetivo sea favorecida por sobre las demás, algo que como se ve, no ocurre en caso contrario.

Claramente la utilización de los operadores de agregación utilizados en este artículo brinda un marco adecuado para resumir los resultados individuales obtenidos mediante el uso de AHP.

Actualmente se está trabajando sobre toma de decisiones con múltiples criterios aplicados en el ámbito de los negocios. Dicho estudio consiste en el análisis de decisiones de inversión en proyectos de negocios con el apoyo de expertos en el ámbito empresarial. La idea principal es analizar los criterios de evaluación aisladamente, y también tener en cuenta las sinergias entre ellos para ver como se potencian o degradan cuando aparecen juntos en una misma alternativa.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del proyecto "**Modelos de Análisis de Información para Vigilancia Tecnológica e Inteligencia Competitiva (VTeIC)**" EIUTNRE0002106 (Universidad Tecnológica Nacional – Facultad Regional Resistencia).

Referencias

[1] http://grameen-info.org, *Apartado About us*, Sitio Web Oficial de Grameen Bank.

[2] Saaty, T.L., *The Analytic Hierarchy Process*. McGraw-Hill, 1980.

[3] Putrus, P., "Accounting for Intangibles in Integrated Manufacturing (nonfinancial justification based on the Analytical Hierarchy Process)," *Information Strategy*, vol. 6, 1990, pp. 25–30.

[4] Boucher, T.O., MacStravic, E.L., "Multiattribute evaluation within a present value framework and its relation to the analytic hierarchy process," *The Engineering Economist*, vol. 37, no. 1, 1991, pp. 1–32.

[5]Yang, J., Shi, P., "Applying Analytic Hierarchy Process in Firm's Overall Performance Evaluation: A Case Study in China," *International Journal of Business*, vol. 7, no. 1, 2002, pp. 31–46.

[6] Ghodsypour, S.H., O'Brien, C., "A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming," *International journal of production economics*, vol. 56, 1998, pp. 199–212.

[7] Korpela, J., Kyläheiko, K., Lehmusvaara, A., Tuominen, M., "An analytic approach to production capacity allocation and supply chain design," *International Journal of Production Economics*, vol. 78, no. 2, 2002, pp. 187–195.

[8] Wang, G., Huang, S.H., Dismukes, J.P., "Manufacturing supply chain design and evaluation," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 1, 2005, pp. 93–100.

[9] Malladi, S., Min, K.J., "Decision support models for the selection of internet access technologies in rural communities," *Telematics and Informatics*, vol. 22, no. 3, 2005, pp. 201–219.

[10] Kwak, N.K., Lee, C.W., "Business process reengineering

for health-care system using multicriteria mathematical programming," *European Journal of Operational Research*, vol. 140, no. 2, 2002, pp. 447–458.

[11] Kearns, G.S., "A Multi-Objective, Multi-Criteria Approach for Evaluating IT Investments: Results from Two Case Studies," *IRMJ*, vol. 17, no. 1, 2004, pp. 37–62.

[12]Grabisch, M., Marichal, J.-L., Mesiar, R. and Pap, E., "Aggregation functions: Means", *Information Sciences,* 2011, pp. 1–22.

[13] Saaty, T.L., "How to make a decision: the analytic hierarchy process," *European journal of operational research*, vol. 48, no. 1, 1990, pp. 9–26.

[14] Saaty, T.L., "Axiomatic foundation of AHP," *Management science*, vol. 32, no. 7, 1986, pp. 841–855.

[15] Peláez, J., Lamata, M., "A new measure of consistency for positive reciprocal matrices," *Computers & Mathematics with Applications*, vol. 46, no. 12, 2003, pp. 1839–1845.

[16] Shiraishi, S., Obata, T., Daigo, M., "Properties of a positive reciprocal matrix and their application to AHP," *Journal of the Operations Research Society of Japan*, vol. 41, no. 3, 1998, pp. 404–414.

[17] Hu, Y.-C., Tsai, J.-F., "Backpropagation multi-layer perceptron for incomplete pairwise comparison matrices in analytic hierarchy process," *Applied Mathematics and Computation*, vol. 180, no. 1, Sep. 2006, pp. 53–62.

[18] Chen, Q., Triantaphyllou, E., "Estimating data for multicriteria decision making problems: optimization techniques," *Encyclopedia of Optimization*, vol. 2, 2001, pp. 27–36.

[19] Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems,* Man and Cyber. 18, 1988, pp. 183–190.

Modelo Semántico Basado en una Meta-Ontología de Información Geográfica

Tolaba, Ana Carolina[1]; Caliusco, Ma. Laura[2]; Galli, Ma. Rosa[3]
[1,2]CONICET – CIDISI UTN, Facultad Regional Santa Fe
[3]INGAR – CONICET – UTN, Facultad Regional Santa Fe
Santa Fe, 3000
[1]atolaba@frsf.utn.edu.ar;[2]mcaliusc@frsf.utn.edu.ar;[3]mrgalli@santafe-conicet.gov.ar

Abstract

La descripción de la información geográfica implica una serie de manipulaciones que deben definirse a través de parámetros específicos que contemplen su semántica, es decir que capturen las particularidades de la información espacial. En este trabajo se discute el proceso de construcción de un modelo semántico basado en una meta-ontología para representar los conceptos, relaciones y restricciones que tienen los datos geográficos, así como un caso de estudio del modelo generado.

Palabras Clave
Información Geográfica, Modelo Semántico, Ontologías, Bases de Datos Espaciales.

1. Introducción

Los avances de las Tecnologías de la Información Geográfica han incrementado la producción, recopilación y difusión de datos geográficos, lo que favorece el diseño y desarrollo de Sistemas de Información Geográfica (GIS - Geographical Information Systems). Estos avances, han dado lugar a que las técnicas de representación, organización y búsqueda de información geográfica sean importantes debido a la gran cantidad de datos que se generan y que deben ser procesados de la mejor manera para poder extraer toda la información útil que sea posible [1].

En un GIS la información se encuentra distribuida en múltiples fuentes como bases de datos geográficas, metadatos, catálogos de fenómenos, vocabularios, y modelos de datos [2]. Además, se debe tener en cuenta que la información empleada por los GIS puede haber sido generada por diferentes productores, con diferentes puntos de vista y vocabularios. Esto refleja la coexistencia de una gran variedad de fuentes con diferente información, estructura y semántica, lo que ocasionan problemas de heterogeneidad [3][4].

Todo esto denota la necesidad de contar con una solución que permita la integración de los datos y el conocimiento embebido en los mismos, de manera eficiente. Es decir, se debe permitir a los usuarios acceder a los datos almacenados en fuentes de datos heterogéneas, presentando una única vista unificada de esos datos, de forma que el usuario no llegue a percibir esta heterogeneidad [5]. La integración de las fuentes de información implica resolver el problema de heterogeneidad.

Se identifican tres tipos de heterogeneidad: sintáctica, esquemática y semántica. La heterogeneidad sintáctica se refiere a la diferencia en el formato de datos, la heterogeneidad esquemática se refiere a las diferencias en el modelo de datos, en los esquemas. La heterogeneidad semántica es referida a las diferencias en la definición, en el significado que se pretende dar a los términos en contextos específicos [6].

Aproximadamente desde hace dos décadas se ha introducido el concepto de ontología como herramienta para solucionar los problemas principales de la heterogeneidad semántica [7]. Las ontologías fueron introducidas por Grüber [6] como una "especificación formal y explıcita de una conceptualización compartida". El termino conceptualización se refiere a un modelo abstracto de como el ser humano piensa comúnmente cosas del mundo real, como por ejemplo, un río. El termino especificación explícita se refiere a que se ha dado un nombre y una definición a los conceptos y relaciones creados en el modelo abstracto. Formal se refiere al hecho de que la ontología debe ser entendida por una computadora. Finalmente, compartida refleja la noción de que una ontología captura conocimiento consensuado, es decir, es aceptada por un grupo o comunidad. Así, surgió una gran cantidad de propuestas diferentes con mayor o menor nivel de complejidad. En [3] se analizan diferentes propuestas que definen posibles enfoques de integración de GIS basados en ontologías. En general, los problemas de integración que surgen en los GIS son los mismos que en sistemas convencionales

además de aquellos que sólo se aplican a este tipo de sistemas debido a la naturaleza de los mismos. Es decir, debido a la naturaleza de la información geográfica. Por lo tanto, surge la necesidad de un nuevo enfoque que considere las características particulares de la información geográfica.

En [8] se presentó una meta-ontología que modela las características principales del dominio geográfico. El propósito del presente trabajo es presentar un conjunto de reglas que a partir de una Base de Datos Geográfica genere una ontología que se adecúe más al dominio geográfico. Para ello, el resto del trabajo se estructura como sigue. La sección 2 presenta las características de modelado de la información geográfica. La sección 3 y 4 describen las diferentes formas consideradas para la construcción del modelado semántico de la información geográfica. Un caso de estudio es presentado en la sección 5. Finalmente, en la discusión se revisan las ideas presentadas y se plantean los siguientes trabajos a realizar.

2. Modelado de la Información Geográfica

La información que emplean los GIS puede proceder de diferentes fuentes y productores de información. Entre los elementos de información que se pueden encontrar están: bases de datos geográficas (Spatial Database - SDB), metadatos, catálogos de objetos geográficos entre otros.

Una SDB es una colección de datos organizados de tal manera que da soporte al almacenamiento de objetos espaciales y sirva efectivamente para una o varias aplicaciones GIS. Esta base de datos comprende la asociación entre sus dos principales componentes: datos espaciales y atributos o datos no espaciales [9]. Los datos espaciales describen la ubicación y topología de puntos, líneas, polígonos y rasgos de la superficie, en tanto los no espaciales describen las características de estos rasgos.

Generalmente, los modelos de datos empleados para la representación de datos geográficos son una extensión de los modelos de datos relacionales, con la particularidad que estos definen un tipo de dato que permite especificar los atributos espaciales o geométricos de los datos representados. Los motores de bases de datos espaciales adoptan el estándar OpenGIS [10] que define datos de tipo geométricos, además, incluye el identificador del sistema de referencia espacial (SRID), el cual es requerido cada vez que se inserta una instancia en un objeto espacial. El modelo conceptual de los objetos espaciales en el estándar OGC, Figura 1, está formado por tres tablas. En una se define el objeto espacial en sí, TABLE_NAME, y en otras dos tablas se define el sistema de referencia espacial llamada SPATIAL_REF_SYS y la columna o atributo geométrico, llamada GEOMETRY_COLUMNS [11].

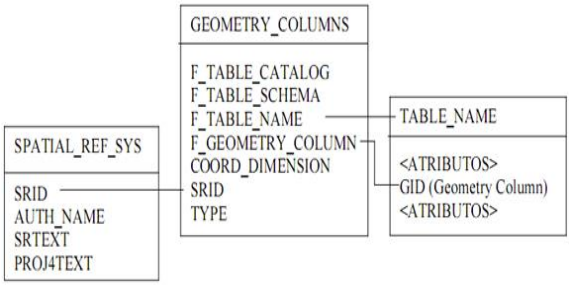


Figura 1. Definición de un objeto espacial en el estándar OGC

Un modelo de datos que represente la información geográfica debe proveer de un nivel de abstracción de modelo de modo que permita representar y diferenciar entre los diferentes tipos de datos involucrados en aplicaciones geográficas, representar diferentes tipos de relaciones, ser capaz de especificar restricciones de integridad así como las relaciones entre los objetos. Es decir, debe ser capaz de proporcionar una visualización clara y de fácil comprensión de la estructura de los datos [12].

El proceso de modelado conceptual debe incluir, por lo tanto, la descripción y definición de posibles contenidos de los datos, así como las estructuras y reglas que se le aplican. Para el modelado de la información geográfica es necesario considerar los siguientes requisitos [13]:

- *Propiedades espacio-temporales*, incluyen los requisitos espaciales (coordenadas en un sistema de referencia, representación geométrica, puntos, líneas y polígonos), el tiempo (necesidad de registrar el tiempo de existencia y los cambios sufridos por un objeto) y atributos necesarios para la representación de los objetos.
- *Roles*, un mismo objeto geográfico puede definirse de diferentes maneras dependiendo del universo de discurso.
- *Asociaciones*, incluye las relaciones entre los objetos (pueden indicar que un objeto compone otro objeto), relaciones topológicas (meet, equal, overlap).
- *Restricciones*, debe ser posible fijar limitaciones a los objetos (por ejemplo, limitando valores de algún atributo). Las restricciones se asocian con la calidad de los datos, y se ve afectada negativamente cuando no se cumplen.
- *Calidad de los datos,* esta información es importante con el fin de conocer la credibilidad de la fuente y los datos.

3. Modelo Semántico de una SDB

Un modelo semántico proporciona una abstracción del mundo real y permite representar el conocimiento tácito de un área de conocimiento. Un modelo semántico se constituye habitualmente por una Ontología. Es decir, permite representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos de manera exhaustiva y rigurosa, con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas.

El conocimiento representado en una ontología es formalizado a través de los siguientes componentes [14]:

- Conceptos (Class): son las ideas a formalizar, son todas las ideas relevantes para un determinado dominio de aplicación y que pueden estar organizados en taxonomías.
- Funciones (Object Properties): son casos especiales de relaciones, donde se generan elementos mediante el cálculo de una función.
- Propiedades (Datatype properties): Los objetos se describen por medio de un conjunto de características o atributos. Estos almacenan diferentes clases de valores.
- Instancias (Individuals): se usan para representar elementos o individuos en una ontología.
- Axiomas (Axioms): Sirven para modelar sentencias que siempre van a ser ciertas. Se usan para representar conocimiento.

La obtención de una ontología puede llevarse a cabo mediante el empleo de técnicas de aprendizaje de ontologías a partir de fuentes de datos, estas técnicas comúnmente son denominadas como técnicas de Ontology Learning (OL) [15]. Estas técnicas extraen los conceptos y las relaciones en función del análisis del esquema de las fuentes de datos.

Sin embargo, la semántica obtenida mediante el empleo de técnicas de OL no puede cumplir con el requisito de la construcción de una ontología que contemple los requerimientos definidos para el modelado de la información geográfica. Esto se debe a que no hay un estándar establecido en los métodos de OL y muchos de estos métodos no están asociados a modelos de datos espaciales sino que estos han sido desarrollados para trabajar con modelos de datos relacionales tradicionales.

4. Meta-ontología Geoespacial

Para describir el conocimiento sobre las características de los datos espaciales necesarios para expresar la comprensión de los objetos espaciales, en [8] se propuso una versión inicial de una meta-ontología geoespacial que cuenta con diferentes elementos que permiten expresar la comprensión de los objetos espaciales por parte de los usuarios/modeladores.

Los elementos que componen la meta-ontología geoespacial son:

- Conceptos:
  - Objetos Geográficos. Representan al conjunto de objetos que tienen asociado una representación espacial y que pueden ser localizados empleando un sistema de coordenadas
  - Objetos No Geográficos. Representan a entidades del mundo real asociadas con un objeto geográfico o al dominio.
  - Geometría. Representan el conjunto de objetos relacionados a la representación espacial de un objeto geográfico.
  - Sistema de Referencia. Describe donde un objeto geográfico puede ser localizado en el mundo real.
  - Temporalidad. Representan el conjunto de objetos al que se les asigna un tiempo de validez, de transacción y de existencia.
- Relaciones:
  - Relaciones explicativas. Relación semántica y referencial que existe entre dos conceptos geográficos o no.
  - Relaciones topológicas (Equal, onTheLine, Overlap, Cross, Meet). Se establecen entre las instancias de los objetos geográficos.
- Axiomas
  - Axiomas de Integridad. Tiene como propósito restringir la definición de conceptos y sus relaciones, de manera que las definiciones de los términos de dominio se vuelvan más precisas.
  - Axiomas de Derivación. Consisten en una o más condiciones y conclusiones.

La meta-ontología puede ser utilizada como marco para la construcción de ontologías de dominio a partir de una SDB en contraposición a las técnicas de OL. Esto facilita la integración de ontologías que comparten la visión de un dominio, Figura 2.

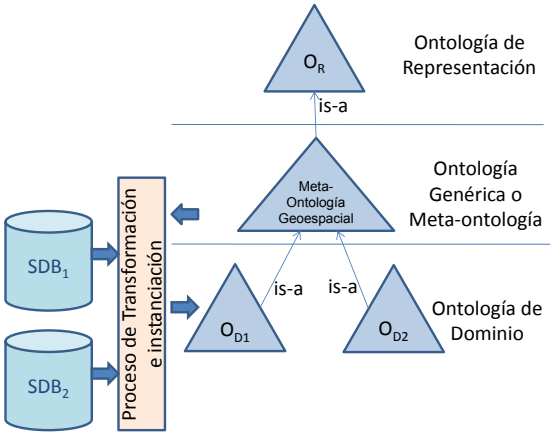


Figura 2. Funcionalidad de la Meta-ontología Geoespacial

Utilizando esta meta-ontología geoespacial se puede generar un modelo semántico de una SDB aplicando las siguientes reglas de transformación:

4.1.1. Reglas para transformar tablas

En la SDB diferentes tipos de tablas pueden ser identificadas. Cada una de estas tablas, pueden ser transformadas en diferentes elementos de la meta-ontología.

La SDB basadas en el Modelo de Datos Espacial del estándar OpenGIS define un tipo de datos *geometry* que contiene la geometría del objeto y sus coordenadas. Las tablas que cuenten con el atributo *geometry* son tablas que describen objetos espaciales por lo tanto se convertirán en conceptos geográficos. Además, de estas tablas se pueden encontrar otras, las cuales son empleadas para describir el dominio de información, estas tablas conformarán los conceptos no geográficos.

Por lo tanto, las tablas que representan objetos espaciales en la SDB se transforman en conceptos geográficos en la ontología con el mismo nombre. Del mismo modo aquellas tablas que representan objetos no espaciales, que no tienen el atributo *geometry*, serán transformadas en conceptos no geográficos considerando aquellas tablas donde la clave primaria de la tabla no está formada únicamente por claves foráneas.

4.1.2. Reglas para transformar columnas

Los atributos de la información son almacenados en las tablas como columnas, estos deben ser transformados como propiedades de los conceptos. Aquellos atributos que no forman parte de la clave foránea se corresponderán a los *datatype properties* de la ontología con el mismo nombre que corresponde con la columna. El dominio es el concepto creado por la tabla y el rango el *datatype* de la columna de la tabla.

En el caso de los atributos espaciales, *geometry*, de un objeto espacial debido a que no se corresponde con ningún tipo de dato estándar, como entero o cadena se establece su representación mediante los conceptos de Geometría, Sistemas de Referencia.

Mientras que los que pertenecen a la *Foreing key* se transformarán en *Object Properties* de la ontología. Existe otro tipo de *Object Properties* que permiten definir las relaciones espaciales entre conceptos. Esta relación establece la relación entre un objeto que no tenga el atributo *geometry* en la tabla pero si este relacionado con un objeto espacial.

4.1.3. Reglas para transformar registros

Estas permitirán realizar un proceso de instanciación de la ontología consistente en el mapeo de tuplas almacenadas en la SDB como instancias de la ontología. El mapeo puede definirse como un conjunto de correspondencias orientadas en la que los elementos de las tuplas de la SDB aparecen en más de una oportunidad. La relación entre una tupla de la SDB y un conjunto de elementos de la ontología es la equivalencia, expresada como *<id, sdbe, oe>*. Donde *id* es el identificador de la equivalencia, *sdbe* es el elemento de la SDB (tabla, columna, clave) identificada de acuerdo a su tipo y nombre y por último *oe* es el conjunto de elementos de la ontología que deben ser instanciados considerando las reglas definidas para la meta-ontología.

5. Caso de Estudio

A continuación se presenta un ejemplo de modelado de una aplicación GIS Web mediante la técnica de Ontology Learning y luego mediante el empleo de una meta-ontología. La aplicación GIS Web constituye una herramienta que permite la publicación y consulta de información geográfica de la ciudad de San Salvador de Jujuy [16], ésta aplicación fue generada a partir de información catastral de la ciudad. La aplicación permite la visualización de mapas temáticos de acuerdo a puntos de interés, como así también la identificación de distintos lugares que prestan diferentes servicios. A partir de la aplicación GIS Web se pudo identificar distintos elementos involucrados (Figura 3), lo que permitió definir los dominios comprendidos por la aplicación en este caso el dominio geográfico referido a la información catastral y el dominio del turismo.

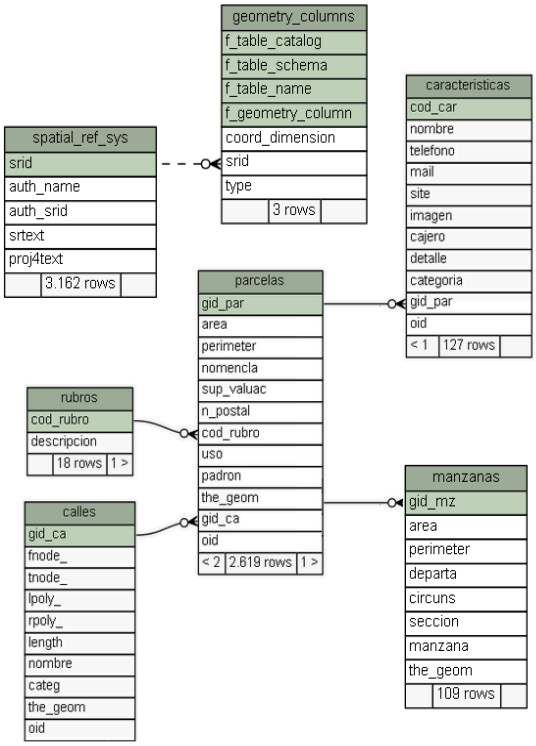


Figura 3. Esquema de datos de la SDB

5.1. Modelado mediante proceso de Ontology Learning

Para la obtención de la ontología a partir de la SDB de la aplicación GIS se empleó el editor de ontologías Protégé[1] que es la herramienta Open Source más ampliamente utilizada con el plug-in DataMaster. El empleo de este plug-in permite obtener una ontología que representa el esquema de la estructura y los datos de una base de datos.

En el caso de estudio, la ontología generada a partir de la SDB de la aplicación GIS, se identificaron conceptos que se corresponde con los elementos involucrados en el dominio de la aplicación como Manzanas, Parcelas, Rubros, Característica, Calles, a estos se suman los términos *spatial_ref_sys* y *geometry_columns* que especifican la parte geográfica de las entidades espaciales, sistema de referencia espacial y geometría vinculada a los objetos de la base de datos.

De acuerdo a la representación de las entidades se verificó que hay algunas relaciones que no están representadas en la ontología. En el caso de las entidades espaciales manzanas, parcelas y calles las mismas no tienen relación con la entidad *geometry_columns* la cual establece la geometría del objeto, ni con la entidad *spatial_ref_sys* que establece el sistema de referencia. Además, se identificó la ausencia de relación entre las entidades manzana y parcela. En la ontología obtenida, Figura 4, se observan las siguientes relaciones entre las principales entidades:

- *gid_par_INSTANCE* que representa la relación entre parcela y características.
- *gid_ca_INSTANCE* que representa la relación entre parcela y calle.
- *cod_rubro_INSTANCE* que representa la relación entre parcela y rubro.

5.2. Modelado mediante el empleo de la Meta-ontología

Para la obtención de los elementos de la meta-ontología se aplicaron reglas de transformación que permiten realizar la transformación del esquema de datos de la SDB basada en los estándares de la Open GIS.

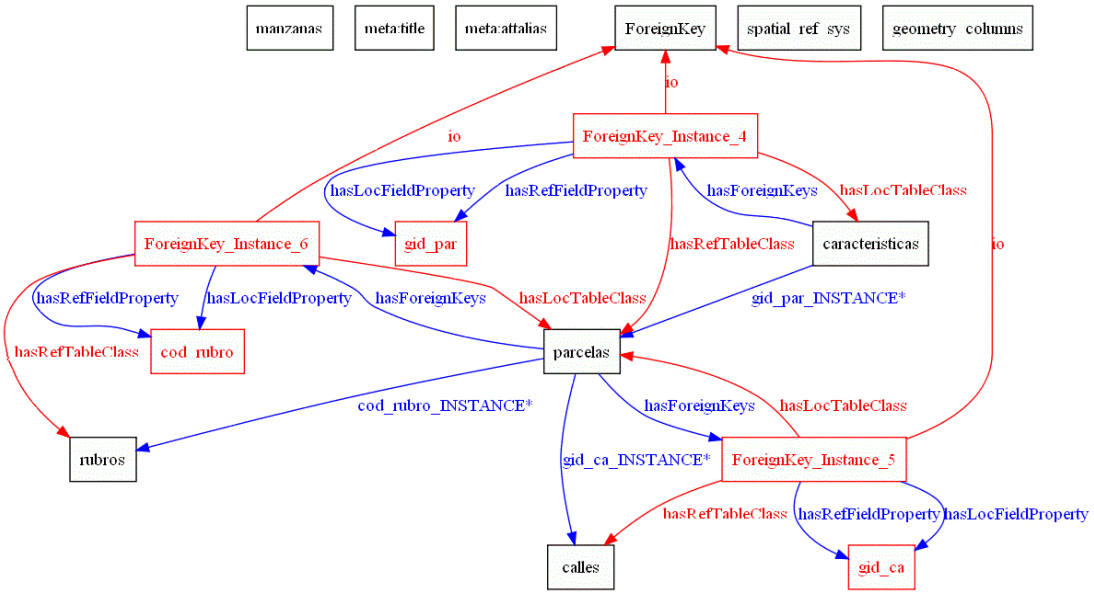


Figura 4. Ontología obtenida mediante Ontology Learning.

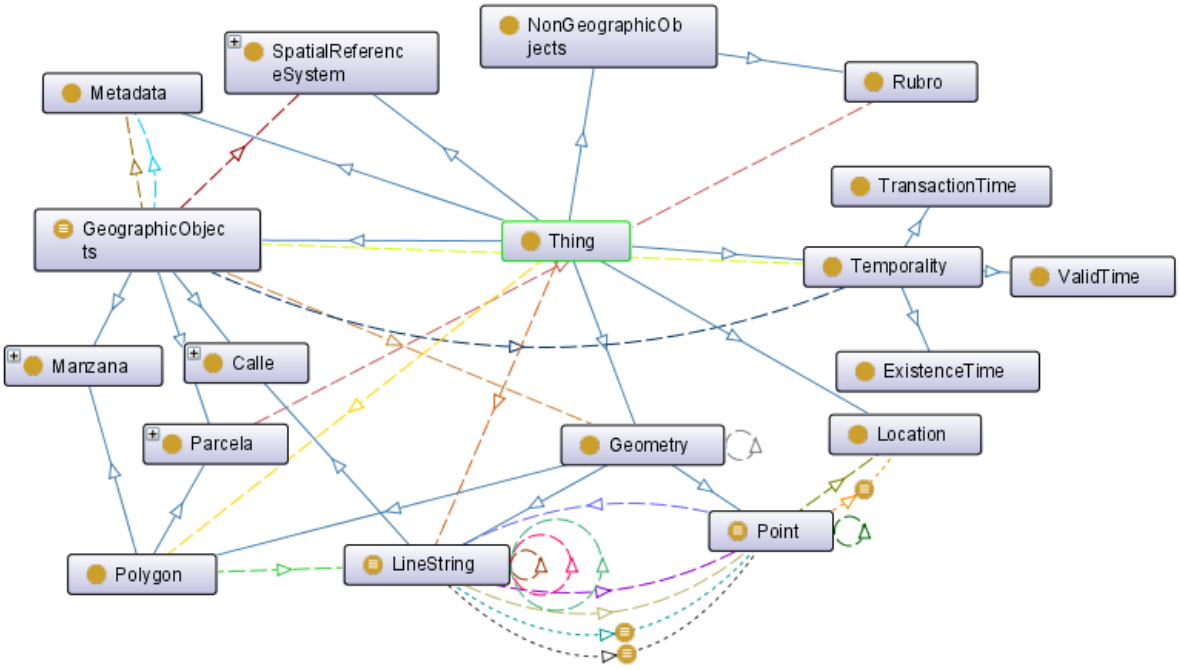


Figura 5. Ontología obtenida como extensión de la meta-ontología.

Mediante el empleo de estas reglas se procede a la extensión de la meta-ontología. En la Tabla 1 se muestra la estructura de las tablas de la SDB de la aplicación de turismo implementada en PostGIS[2].

Tabla 1. Esquema de la SDB de la aplicación GIS.

Nº	Tabla	Type
1	Parcelas(gid_par, nomecla, padron, n_postal, gid_mz, gid_rubro, gid_ca)	T_{GO}
2	Manzanas(gid_mz, area, perimeter, departa, seccion)	T_{GO}
3	Calles(gid_ca, nombre, categ, length, fnode, tnode)	T_{GO}
4	Rubros(gid_rubro, descripcion)	T_{NGO}
5	Caracteristicas(cod_car, nombre, telefono, mail, site, detalle, categoría, gid_par)	T_{NGO}

De acuerdo con las reglas de transformación del esquema de la SDB se establecieron correspondencias entre los elementos de la meta-ontología, Tabla 2.

Tabla 2. Ejemplo de mapeo entre elementos de la SDB y la meta-ontología.

Id	sdbe	Oe
1	SDB Table (parcelas)	Geographic Object (Parcelas)
2	SDB Table (Manzanas)	Geographic Object (Manzanas)
3	SDB Table (Calles)	Geographic Object (Calles)
4	SDB Table (Rubros)	NonGeographic Object (Rubros)
5	SDB Column (Parcelas.Padron)	Datatype Properties (Padron, domain: Parcela, dange:xsd:string)
6	SDB Column (Calles.geom)	Geometry: LineString Object Propertie Has_Geometry(Calles. Linestring)

[2] PostGIS modulo que da soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL.

En la Figura 5 se muestra la ontología con las correspondencias obtenidas del proceso de transformación.

5.3. Evaluación de los modelos

La evaluación del modelo semántico debe realizarse como cualquier componente de software, el proceso consiste en la emisión de un juicio técnico del contenido con respecto a un marco de referencia, requerimientos, entre otros [2]. El objetivo de la evaluación es identificar errores en el modelado de los objetos geográficos, verificando que no se definió o que se hizo de forma incorrecta.

Para garantizar la consistencia de la ontología se validó la definición de los conceptos y su jerarquía de modo de identificar posibles errores de integridad. Para realizar estas tarea se empleó el razonador Pellet [17]. Pellet puede ser instalado como un plugin de Protégé, es compatible con el lenguaje OWL2 y con el lenguaje de definición de reglas SWRL (Semantic Web Rule Language), Figura 6.

Para realizar un análisis de los modelos ontológicos obtenidos ya sea mediante técnicas de Ontology Learning o el empleo de la meta-ontología se definieron las siguientes métricas:

- Tamaño:
  - Número Total de Clases
  - Número Total de Object Properties
  - Número Total de Datatype Properties
- Completitud: se refiere a si el modelo contiene toda la información requerida para apoyar la funcionalidad del sistema.
  - RequirementsCoverage=X/Y, donde X es el número de requerimientos cubiertos por los elementos del modelo. Y, es el número total de requerimientos.
- Interpretabilidad: se refiere a si es fácil de interpretar por los usuarios.
  - I=SW/WCP, donde SW es el número de palabras usado para definir clases y propiedades que tienen sentido en al menos un marco de referencia. WCP número total de diferentes palabras usadas para definir clases y propiedades en la ontología.
- Claridad: se refiere a la facilidad con la que los conceptos y estructura en el modelo pueden ser entendidas.
  - Cl=ΣSi/TN, donde Ni es el número de clases o propiedades en la ontología. Sea Si el número de palabras en el marco de referencia y TN el número total de clases o propiedades en la ontología.

Los modelos obtenidos deben satisfacer estas métricas. El rango de valores oscila entre 0 y 1, donde 0 indica que el modelo no cuenta con esta característica y 1 que sí. En la Tabla 3 se muestra una comparación de los valores obtenidos.

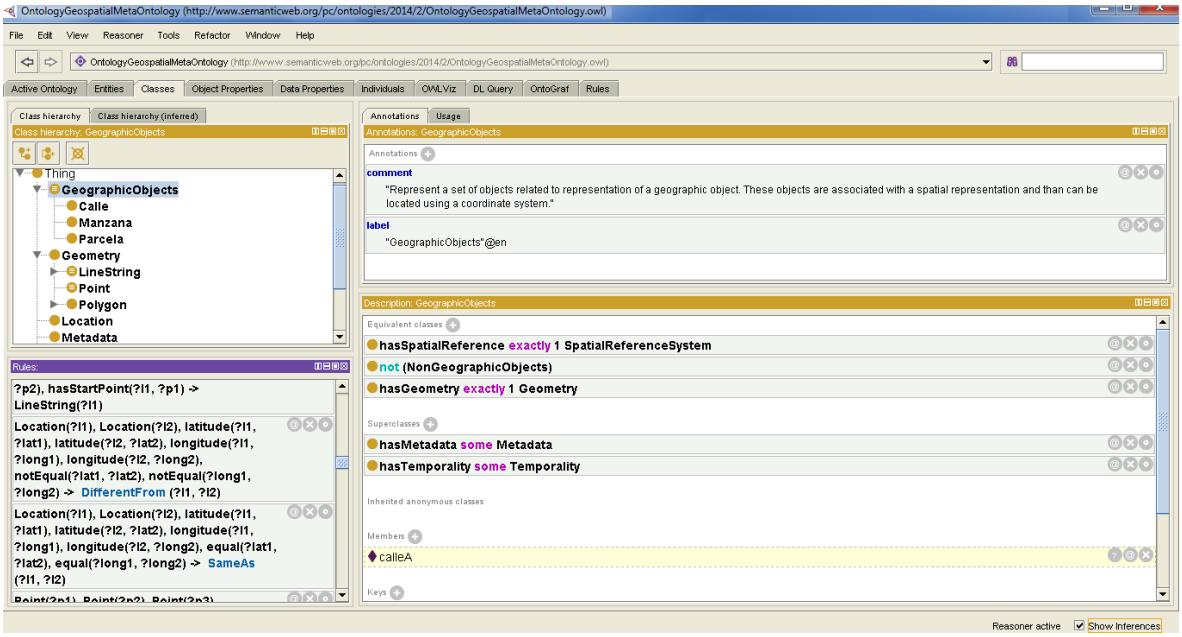


Figura 6. Ontología, razonador Pellet y Reglas implementadas en SWRL.

Tabla 3. Métricas de evaluación de los modelos ontológicos.

Métrica	Modelo A (Ontology Learning)	Modelo B (Meta-ontología Geoespacial)
Tamaño	Clases: 10 Object Properties:10 Datatype Properties: 10	Clases: 17 Object Properties: 21 Datatype Properties: 5
Completitud	0,4	0,8
Interpretabilidad	0,33	0,872
Claridad	0,55	0,93

6. Discusión y Trabajo Futuro

En este trabajo se ha presentado un caso de estudio que permite el análisis de métodos para la obtención de una ontología que permite el modelado de información geográfica de modo de proporcionar al usuario un conjunto de características que deben expresar los conceptos relacionados con el dominio geoespacial.

Si bien existen herramientas que pueden facilitar la tarea de extracción de forma automática de los datos existentes en la representación de la ontología, Ontology Learning, los resultados obtenidos pueden no satisfacer plenamente los requerimientos establecidos, es por ello que es necesario realizar las modificaciones que se consideren necesarias para mejorar su representación.

Por lo tanto, al obtener ontologías mediante las técnicas de Ontology Learning debe considerarse optar por el enriquecimiento de la ontología obtenida mediante el empleo de otros recursos ontológicos llevando a cabo un proceso de reingeniería.

El empleo de la meta-ontología geoespacial ofrece un conjunto de conceptos para extender que sirven de apoyo para la representación de los datos geográficos que describen objetos del mundo real y consideran las características de los diferentes objetos. De los resultados presentados en la Tabla 3, se observa que la creación de una ontología a partir de la meta-ontología ofrece mejores resultados ya que las métricas indican que la ontología obtenida será más clara, interpretable y más completa, esto evidencia que debido a las particularidades de las SDB el empleo de técnicas de ontology learning requieren necesariamente de un proceso de reingeniería.

Como trabajo futuro, se plantea la definición de un marco de trabajo que permita obtener una ontología enriquecida de dominio a partir de las fuentes de datos, lo que permitirá agregar nuevos conceptos, nuevas relaciones y restricciones.

7. Referencias

[1] R. Larin Fonseca and E. Garea Llano, “Integración Semántica de Datos Espaciales con Sistemas de Información Geográfica,” in *XIII Convención Internacional Informática 2009*, 2009.

[2] M. C. Suárez-Figueroa and A. Goméz-Pérez, “First attempt towards a standard glossary of ontology engineering terminology,” in *8th International conference on Terminology and Knowledge Engineering*, 2008, pp. 1–15.

[3] A. Buccella and A. Cechich, “Towards Integration of Geographic Information Systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 168, pp. 45–49, 2007.

[4] C. Wey, “Using Ontology to Achieve the Semantic Integration and Interoperation of the Geography Information System,” Wuhan University, 2004.

[5] T. Berners-Lee, “Design Issues: Linked Data,” 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[6] H. Stuckenschmidt, “Ontology-Based Information Sharing in Weakly Structured Environments,” Vrije Universiteit Amsterdam (NL), 2003.

[7] A. Goméz-Pérez, M. Fernández-López, and Ó. Corcho, *Ontological engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. 2004, p. 420.

[8] A. C. Tolaba, M. L. Caliusco, and M. R. Galli, “Meta-ontología Geoespacial: Ontología para Representar la Semántica del Dominio Geoespacial,” in *1er Congreso Nacional de Ingeniería Informática / Sistemas de Información*, 2013.

[9] I. E. E. Services, “Environmental Systems Research Institute,” 1998.

[10] OGC, “OpenGIS Consortium Standards,” 2006. [Online]. Available: http://www.opengis.org.

[11] I. S. O. ISO, “ISO/TC 211 19125-2. Simple Feature Access - Part 2:SQL Option.” 2001.

[12] K. A. V. Borges, C. A. Davis, and A. H. F. Laender, “OMT-G: An Object-Oriented Data Model for Geographic Applications,” *Geoinformatica*, vol. 5, no. 3, pp. 221–260, 2001.

[13] A. Friis-Christensen, N. Tryfona, and C. S. Jensen, “Requirements and research issues in geographic data modeling,” in *Proceedings of the ninth ACM international symposium on Advances in geographic information systems - GIS ’01*, 2001, p. 2.

[14] S. Staab and R. Studer, Eds., *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[15] L. Drumond and R. Gerardi, “A Survey of Ontology Learning Procedures,” in *3rd Workshop on Ontologies and their Application*, 2008.

[16] G. Revollo Sarmiento, A. C. Tolaba, and N. Revollo Sarmiento, “SIG Web en Entornos Open Source Aplicados a la Reconstrucción de Información Geográfica de la Ciudad de San Salvador de Jujuy,” in

1ras Jornadas de las Nuevas Tecnológias de la Información Geográfica del Sur Argentino, 2010.

[17] K. Clark and B. Parsia, "Pellet: Owl 2 reasoner for java." [Online]. Available: http://clarkparsia.com/pellet/. [Accessed: 01-Apr-2014].

Desarrollo de una Infraestructura de Datos Espaciales Académica

[1]Martín, Horacio - [2]Linares, Santiago - [1]Suarez, Marcos - [1]Magliocchetti, Agustín - [1]Sajewicz, Brian - [1]Gil, Guillermo - [2]Ortmann, Mauro
[1]Universidad Tecnológica Nacional, Facultad Regional Trenque Lauquen
[2]Universidad Nacional del Centro de la Provincia de Buenos Aires, Facultad de Ciencias Humanas

Abstract

Una Infraestructura de Datos Espaciales (IDE) es un Sistema de Información Geográfica (SIG) distribuido. Está integrada por datos y atributos geográficos (metadatos), tecnologías de red y servicios.
Partimos de la convicción de que una IDE académica institucional brindará una mayor disponibilidad de información espacial, un mejor conocimiento de los trabajos de investigación ya realizados y un mayor aprovechamiento de los esfuerzos puestos en la creación de geodatos, como fruto de la cooperación y compartición de recursos.
En este trabajo se presentan los avances del proyecto de investigación y desarrollo en curso en la Facultad Regional Trenque Lauquen de la Universidad Tecnológica Nacional (UTN), que se propone la construcción de una IDE académica para la Facultad de Ciencias Humanas de la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), aportando una primera solución, totalmente funcional, a las necesidades de la institución.

Palabras Clave
Infraestructura de datos espaciales, IDE, e-Ciencia.

Introducción

Los Sistemas de Información Geográfica (SIG) [1] se han constituido en una de las más importantes herramientas de trabajo en todas aquellas actividades que tienen como insumo el manejo de la información relacionada con diversos niveles de agregación espacial o territorial.

Sencillamente podríamos definir a los SIG como un conjunto de métodos, herramientas y datos que fueron diseñados para capturar, almacenar, analizar, transformar y presentar la información geográfica y de sus atributos de forma coordinada y lógica.

Por otro lado, una Infraestructura de Datos Espaciales (IDE) [2] [3] es un sistema informático que combina las potencialidades de los SIG y la accesibilidad y disponibilidad de la web; suele ser definida como un SIG distribuido. Está integrada por datos y atributos geográficos (metadatos), tecnologías de red y servicios.

Los servicios de una IDE son las funcionalidades accesibles mediante un navegador de Internet que se ofrecen al usuario para aplicarlos sobre los datos geográficos.

A fin de garantizar la interoperabilidad del sistema, los recursos deben cumplir una serie de condiciones (normas, especificaciones, protocolos, interfaces) que permitan que un usuario, utilizando un simple navegador, pueda utilizarlos y combinarlos según sus necesidades.

Las Universidades, como cada una de las facultades y núcleos de actividades científico-tecnológicas que las conforman, en tanto instituciones públicas creadoras de conocimientos, poseen la obligación no sólo de responder a las demandas sociales emergentes a partir de las temáticas que abordan, sino también de poner al servicio de la comunidad los resultados de tales abordajes. En tal sentido, deben interesarse por la forma en la que se difunde hacia afuera de las instituciones universitarias la información que generan.

En nuestro medio, se percibe una demanda creciente de información geográfica por parte de diferentes docentes-investigadores,

pertenecientes y ajenos a la Facultad de Ciencias Humanas de la UNCPBA, y se considera importante establecer algún tipo de mecanismo que registre, regule y facilite la distribución y el acceso a este tipo de datos. La creación de información espacial es inherente en alguna dimensión a cada una de las disciplinas existentes [4], ya sea como insumo o como materialización del trabajo de investigación. Se trata de recursos de costosa producción y difícil acceso, debido a formatos incompatibles, modelos divergentes, políticas de distribución restringidas o falta de conocimiento.

Partimos de la convicción de que una IDE Académica, con el respaldo institucional continuado y decidido que merece, redundará sin lugar a dudas en importantes beneficios relacionados con la obtención de una mayor disponibilidad de información espacial, un mejor conocimiento de los trabajos ya realizados y un mayor aprovechamiento de los esfuerzos puestos en la creación de geodatos, como fruto de la cooperación y compartición de recursos.

Las IDEs dedicadas al soporte de la actividad científica no son un concepto realmente nuevo. Ya en 1998 Al Gore, entonces vicepresidente de EEUU, presentó la visión de Digital Earth: un modelo tridimensional de la Tierra que fuera capaz de representar información geográfica de carácter físico y humano para mejorar la comprensión de los procesos globales, tanto naturales cómo sociales [5].

El concepto de e-Ciencia aparece en el marco de Information Society Technologies (IST) del programa europeo y se define como “ciencia desarrollada a través de colaboraciones globales posibilitadas de forma distribuida en Internet” [6]. El desarrollo de la e-Ciencia está ligado a los avances tecnológicos y las IDE juegan en ello un papel muy importante.

Actualmente existen diversas IDEs dedicadas a las actividades científicas en Europa y Norte América [5].

En el mismo sentido se están orientando en Latinoamérica, proyectos de construcción de IDEs dedicadas al sector académico, por ejemplo, la propuesta para el ámbito de las Estaciones Experimentales de investigación agrícola de Venezuela [7].

En una escala mayor, existe un proyecto de gran envergadura, encarado por LATIN IDE (Comunidad Latinoamericana de Infraestructura de Datos Espaciales), comunidad virtual sin restricciones de acceso, que agrupa a investigadores de 20 instituciones de 8 países (Argentina, Bolivia, Chile, Colombia, Costa Rica, Ecuador, Venezuela y Perú), en torno al trabajo investigativo y académico acerca de IDEs.

En nuestro país, hay desarrollos como la IDE LabSIG de la UNGS que plantea la necesidad de implementar una IDE sobre el Conurbano Bonaerense (área comprendida por la Ciudad de Buenos Aires y los 24 partidos que la rodean), procurando facilitar el acceso y la integración de la información espacial entre múltiples usuarios, especialmente vinculados a la actividad académica [8].

Nuestra propuesta parte de un proyecto de investigación y desarrollo de SIGs, que involucra a dos universidades nacionales y una extranjera, a través de sendas Facultades: la Facultad de Ciencias Humanas de la UNCPBA, la Facultad Regional Trenque Lauquen de la UTN y la Facultad de Ciencias Económicas y Empresariales de la Universidad de Castilla La Mancha. En el marco del mismo se ha construido un prototipo de SIG mediante herramientas de software libre.

Así, la disponibilidad del know-how tecnológico de una y la amplia experiencia en el manejo y procesamiento de geodatos

de otras, ha llevado a la concreción de un acuerdo específico entre las Facultades para satisfacer la necesitad de implementación de una IDE Académica.

En este trabajo se presentan los avances del proyecto de investigación y desarrollo en curso en la Facultad Regional Trenque Lauquen de la UTN, que se propone la aplicación del prototipo SIG en la construcción de tal IDE Académica para la Facultad de Ciencias Humanas de la UNCPBA, aportando una primera aproximación, totalmente funcional, a la solución del problema.

Una IDE corporativa, como la aquí desarrollada, permitirá armonizar las múltiples bases de datos espaciales ya generadas y ponerlas al servicio de nuevos proyectos, reutilizándolas, aprovechando esfuerzos y disminuyendo costos hacia el interior de la comunidad universitaria. Además, la utilización de una IDE por parte de la ciudadanía en su conjunto, permitirá extender el conocimiento generado en instituciones y organismos públicos y el uso de la información geográfica en gran variedad de situaciones.

Elementos del Trabajo y metodología

Los sistemas que estudiamos son aplicaciones atípicas que involucran un ajuste complejo de diversos frameworks y programación. Se han propuesto metodologías complejas para su diseño como la empleada por Abarca & Bernabé Poveda [7], basada en el modelo de de Hjelmager et al. [9].

Estamos experimentando con modelos de procesos de desarrollo a fin de seleccionar el más adecuado. Empleamos un modelo evolutivo basado en prototipos. Para ello, se pactaron etapas de presentación del sistema con funcionalidades definidas. El Sistema fue alojado en el servidor de la Facultad de Ciencias Humanas de la UNCPBA y las updates son realizadas vía Internet por el equipo de desarrollo. El mismo ha distribuido sus tareas de acuerdo a las funcionalidades requeridas y emplea para la gestión de versiones un framework open source.

Para el diseño y documentación del modelo de clases, nos hemos propuesto utilizar la extensión del Lenguaje Unificado de Modelado (UML) [10] para aplicaciones web denominada WAE (Web Application Extensions) [11]. UML tiene definido un mecanismo de extensión que permite incluir nuevos atributos, diferente semántica y restricciones adicionales. WAE es el conjunto de extensión de UML formado por valores etiquetados, estereotipos y restricciones que posibilitan modelar las peculiaridades de una aplicación web.

La arquitectura de software sigue el patrón MVC (modelo – vista – controlador) [12] que separa la información con la que trabaja la aplicación, es decir, su lógica de negocio (el modelo), de la presentación al usuario (la vista), siendo el controlador el responsable de procesar las interacciones entre ambos. Esto facilita el desarrollo y mantenimiento de las aplicaciones.

Para alcanzar los requerimientos demandados, se debieron seleccionar las tecnologías y herramientas de software que se emplearon en el proyecto. Los criterios que se tomaron con este fin fundamentalmente fueron su robustez, estabilidad, seguridad y alto rendimiento en el manejo de información geoespacial. Se optó por utilizar exclusivamente software libre.

Se escogieron las siguientes tecnologías:
- software de base: sistema operativo Linux, servidor HTTP Apache;
- base de datos: PostgreSQL con extensión PostGIS, administrador de bases de datos pgAdmin; lenguaje de consulta SQL.
- programación en servidor: lenguaje PHP, con el framework Mapscript, entorno de

desarrollo NetBeans, servidor de mapas MapServer;
- programación en cliente y diseño web: lenguaje JavaScript, con tecnología Ajax, frameworks OpenLayers, JQuery y ExtJS;
- procesamiento de datos geoespaciales: aplicación Quantum GIS.

Para alojar el sistema se configuró un servidor web. Como sistema operativo se eligió la distribución de Linux OsGeo que tiene incorporadas la mayoría de las herramientas y utilitarios a emplear.

Resultados

La gestión de usuarios exige tres niveles de autenticación: usuario administrador, usuarios registrados y usuarios no registrados.

La Fig. 1 muestra la asignación de funcionalidades a los actores. Los usuarios se registran accediendo, mediante el ícono de login, al formulario correspondiente. El sistema responde enviándole un email con una clave de acceso.

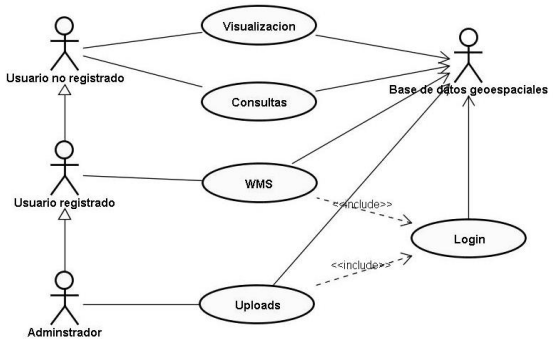


Fig. 1. Diagrama de Casos de Uso

La IDE desarrollada comprende funcionalidades que permiten la visualización de información geoespacial y consultas sobre la misma, tanto por selección gráfica sobre las capas, como por la estipulación de filtros por valores alfanuméricos según criterios preestablecidos.

Además provee la visualización de mapas online y descarga de archivos en formato shp o mediante un Servicio de Mapas Web (WMS), de nuestro servidor y de otros servidores. WMS es un estándar del OGC (Open Geospatial Consortium) que permite acceder a mapas generados por un servidor de mapas, accediendo a datos georreferenciados. WMS es accesible vía HTTP a través de una URL que devolverá el mapa solicitado, permitiendo crear mapas compuestos combinando coberturas de diferentes peticiones.

El OGC define otros estándares para las IDE cuya implementación será encarada en etapas posteriores de refinamiento del sistema.

La IDE desarrollada cuenta con políticas de edición, uso y metadatos conforme a las normas y estándares Internacionales definidas por el OGC. El OGC, en su carácter de consorcio industrial, está presentando sus especificaciones a la ISO para su normalización a través del ISO/TC 211 [13]. La IDE utiliza un paquete de descriptores compuesto por 28 elementos, utilizado por el Perfil de Metadatos de la Infraestructura de Datos Espaciales de la República Argentina (IDERA), basados en el núcleo de la norma ISO 19115:2003.

La IDE permite la incorporación (uploads) en su base de datos de información geográfica y alfanumérica a partir de los datos obtenidos de diversas fuentes. La información geoespacial debe ser procesada para lograr compatibilizar los distintos formatos y sistemas de proyección cartográfica. Esta es una tarea crítica en la implementación de la IDE [14].

La incorporación de nuevas capas estará a cargo del administrador del sistema por solicitud de los usuarios registrados, de acuerdo a flujos de trabajo previamente definidos y políticas de edición estipuladas.

Al ingresar a la página del sistema nos encontramos con la interfaz y sus secciones (Fig. 2). En la misma se puede apreciar la

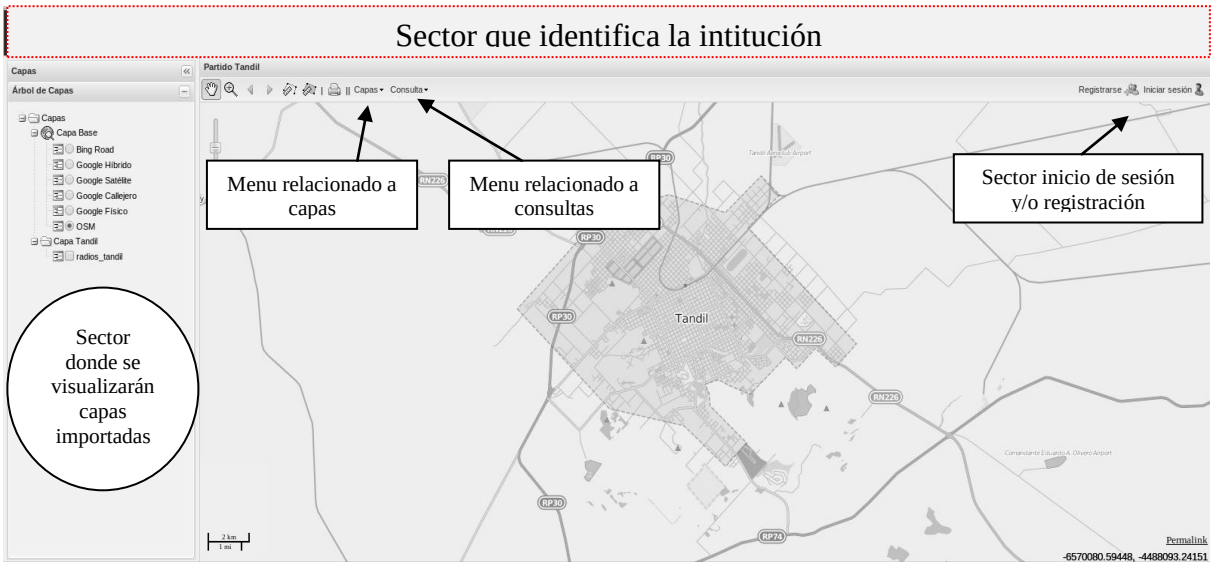


Fig. 2. Pantalla inicial de la IDE

zona de la barra de título que presenta un texto con el nombre de la aplicación y el logo de la institución para la cual se ha desarrollado.

Por debajo y sobre la derecha se ve el panel rebatible que permite seleccionar las capas cargadas, agrupadas en un árbol según una categorización consensuada entre los investigadores proveedores de información, derivada de la propuesta adoptada por el IGN en su perfil de metadatos PMIDERA a partir de la norma ISO 19110 e ISO 19126, organizada en las siguientes categorías:

- Economía
  - Agricultura y ganadería
  - Minería
  - Pesca
  - Industria
  - Comercio y Servicios
  - Sistema bancario, financiero y aseguradoras
  - Turismo
- Población y Viviendas
  - Estructura demográfica
  - Salud
  - Educación
  - Vivienda y Hogares
  - Asentamientos humanos
- Ordenamiento territorial, infraestructura y comunicación
  - Zonificación y usos del suelo, planeamiento catastral
  - Equipamientos colectivos
  - Redes de comunicación, transporte y accesibilidad
  - Infraestructura de servicios
  - Divisiones administrativas
- Marco ambiental
  - Climatología y Meteorología
  - Biogeografía
  - Geomorfología y Edafología
  - Elevación
  - Hidrología

Ocupando la mayor parte de la interfaz se encuentra el sector del mapa, principal área de visualización de información del sistema desde donde además se accede a los menús para gestión de usuarios, de capas y de consultas. En él se alojan los controles (Fig. 3) para un manejo más dinámico: barra de zoom (1); barra de herramientas (2) (con las opciones de movimiento, zoom, avanzar o retroceder según los movimientos en el mapa, medir, área e imprimir); escala (3) y posición del cursor (4).

Discusión

La evaluación del producto obtenido debe hacerse desde dos perspectivas distintas. Por un lado, su eficacia, que estará dada por

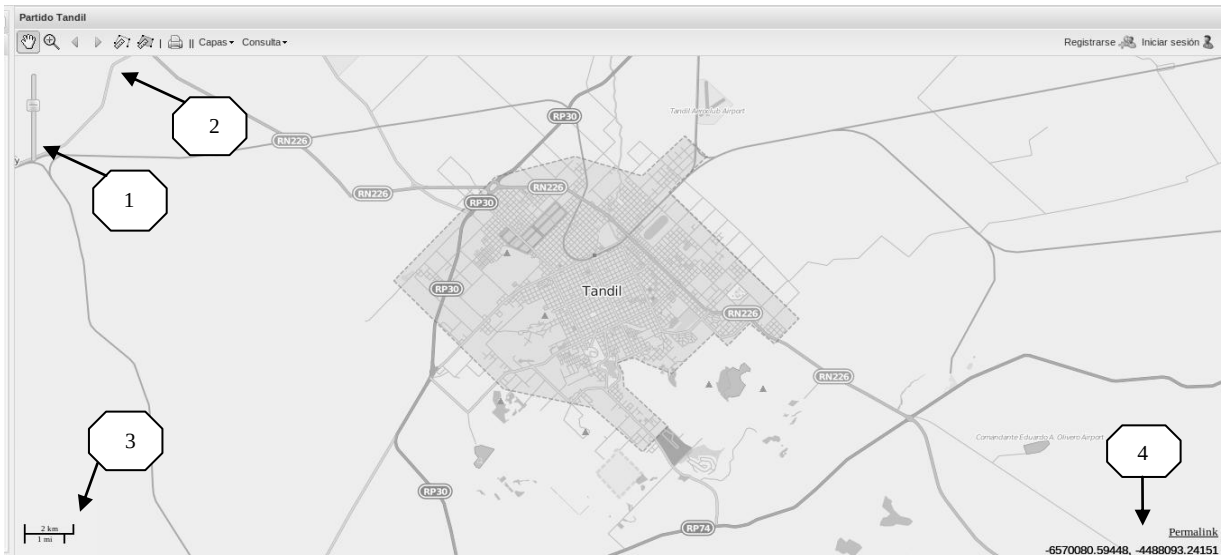


Fig. 3. Sector del mapa

su adecuación a los requerimientos (técnicos y funcionales). Esta evaluación se lleva a cabo en cada entrega modular, hasta finalizar el desarrollo.

Por otro lado debe considerarse la validación de los objetivos para el cual la IDE ha sido desarrollada: las potencialidades de la IDE Académica establecidas en nuestra Introducción. La metodología de validación de los objetivos se ha propuesto mediante la correlación de una serie de variables y sus respectivos indicadores y/o índices. El instrumento de "medición" será el propio sistema a través de cuestionarios impartidos a los usuarios en el formulario de registro (datos pre) y evaluaciones estadísticas de los usos que le den a la IDE al cabo de un lapso establecido (datos pos). La correlación entre ambos tipos de datos permitirá realizar la evaluación de la eficiencia real de este instrumento y hacer una discusión más amplia de los resultados obtenidos.

Conclusión

El proyecto ha dado lugar a un fluido intercambio de conocimientos entre las tres facultades correspondientes a distintas universidades, fortalecido por la financiación que otorga el programa Redes Interuniversitarias de la Secretaría de Políticas Universitarias del Ministerio de Educación.

La IDE desarrollada se encuentra operativa, funcionando en un servidor de la Facultad de Ciencias Humanas de la UNCPBA. En la actualidad se están diseñando otros módulos que permitirán ampliar sus funcionalidades y servicios.

Este trabajo ha puesto de manifiesto la potencialidad de las IDE como una plataforma para difundir los resultados y datos generados en las investigaciones llegando a la comunidad académica y a la sociedad en su conjunto, aportando a un proceso de socialización y democratización de la información científica.

Referencias

[1] Olaya, V., *Sistemas de información geográfica*, Creative Common Atribucion, 2011.

[2] Bernabé Poveda, M. A., López Vázquez, C. M, *Fundamentos de las Infraestructuras de Datos Espaciales*. UPM-Press, Serie Científica., 2012.

[3] Valencia Martínez de Antoñana, J., *Pasado, presente y futuro de las infraestructuras de datos espaciales*, Bubok Publishing S.L., 2008.

[4] del Boque González, I. et al., *Los Sistemas De Información Geográfica y la Investigación en Ciencias Humanas y Sociales*, Confederación Española de Centros de Estudios Locales, 2012.

[5] Guasp Giner, L., "Las Infraestructuras de Información Geográfica Científica: conceptualización y análisis de casos existentes", in *Jornadas Ibéricas de Infraestructuras de Datos Espaciales*, Madrid, España, 2012.

[6] Granell, C. et al., 2006, "Contribuciones de una IDE a la e-Ciencia: Proyecto AWARE", *Avances en las infraestructuras de datos espaciales*, Publicacions de la Universitat Jaume 1, pp. 73-83.

[7] Abarca, O. and Bernabé Poveda, M., "Propuesta de un modelo conceptual para el desarrollo de una Infraestructura de Datos Espaciales (IDE) orientada a la gestión de estaciones experimentales de investigación, extensión, docencia y producción", *GeoFocus (Informes y comentarios)*, nº 8, 2008, pp. 1-22.

[8] Nicolás, C. and Miraglia, M., "Creación de una infraestructura de datos espaciales para el conurbano bonaerense", in *XIV Conferencia Iberoamericana de Sistemas de Información Geográfica*, Universidad Nacional Autónoma de Honduras, Tegucigalpa, Honduras, 2013.

[9] Hjelmager, J. et al., "Developing a modeling for the spatial data infrastructure", in *International Cartographic Conference*, La Coruña, España, 2005.

[10] Kumbaugh, J., Jacobson, I. and Booch, G., *El Lenguaje Unificado de Modelado: Manual de Referencia*, Pearson Educación S. A., 2000.

[11] Conallen, J., *Building Web Applications with UML Second Edition*, Addison-Wesley, 2002.

[12] Gamma, E., Heln, R. and Vlissides, J., *Patrones de Diseño. Elementos de software orientado a objetos reutilizable*, Addison-Wesley, 2003.

[13] Comité ISOTC 211, *Guía de normas. Información Geográfica / Geomática*. Creative Common, 2010.

[14] Pons Rubio, J. L., *Adaptación de la Información Cartográfica Local a las Infraestructuras de Datos Espaciales*, Universidad de Córdoba, España, Tesis Doctoral, 2009.

Datos de Contacto

Horacio Omar Martín. Facultad Regional Trenque Lauquen, Universidad Tecnológica Nacional. Villegas 980, Trenque Lauquen. inghoraciomartin@gmail.com.

Evaluación de la Usabilidad en sistemas E-Cultura

Gallo, Fabio Rafael; Palavecino, Rosa Adela; Herrera, Susana Isabel
Universidad Nacional de Santiago del Estero, Facultad de Ciencias Exactas y Tecnologías
fabiorafael22@gmail.com – {rosypgg, sherrera}@unse.edu.ar

Abstract

En los últimos tiempos, las Tecnologías de la Información y de la Comunicación (TICs) han tenido impactos significativos en todas las actividades del hombre. Uno de los ámbitos afectados es la cultura; actualmente, personas de cualquier parte del mundo, usando las TICs, tienen acceso a contenidos de otras culturas de manera rápida y económica. Si bien esto facilita y promueve la divulgación cultural, también implica riesgos como la tendencia a la homogenización y a la globalización. Es en este aspecto donde los sistemas de e-cultura cumplen un rol fundamental, éstos permiten reforzar el sentido de identidad cultural de una comunidad y favorecer el diálogo intercultural. Sin embargo, existen cuestiones problemáticas en el desarrollo de este tipo de sistemas de información. Entre ellas se encuentran: la necesidad de utilizar arquitecturas web, la optimización de la usabilidad en cuanto a la capacidad de ser aprendido por otras culturas, poniendo énfasis en el diseño emotivo y en el uso de un lenguaje universal.

El objetivo general de este trabajo fue contribuir a la optimización de la usabilidad de sistemas de e-cultura. Para ello se definió una nueva metodología ágil de desarrollo denominada Desarrollo Guiado por Características para Sistemas de E-Cultura (DeGuiCSEC); la cual, a partir de Feature Driven Development (FDD), incorpora pautas de diseño emotivo y de usabilidad (estándar ISO/IEC 9126).

Con el propósito de evaluar la usabilidad de los productos desarrollados con DeGuiCSEC, se desarrolló un prototipo de Sistema Web de e-cultura Santiagueña usando dicha metodología. Su usabilidad fue evaluada aplicando un Cuestionario de Usabilidad para Sistemas de E-Cultura (definido a partir de cuestionarios estándares de usabilidad) a una muestra de usuarios multiculturales y siguiendo un procedimiento de cálculo basado en medias aritméticas. Se obtuvo como resultado un alto nivel de usabilidad.

1. Introducción

Los avances en el tratamiento de la información y los nuevos sistemas de comunicación de los últimos años, han propiciado lo que algunos denominan la nueva revolución social, con el desarrollo de la sociedad de la información [5]. Esto ha tenido impactos significativos en varios ámbitos. Uno de ellos es la cultura, lo cual dio lugar a un nuevo concepto, e-cultura, el cual abarca todos los procesos de expresión y reflexión en el dominio digital [6-8, 18, 23]. Aunque tiene efectos positivos, las características de la globalización y las TICs han venido afectando de alguna manera la identidad de las culturas del mundo [22]. Una manera de revertirla es aprovechar el mismo medio para la promoción de las culturas poco difundidas, como es el caso de la cultura santiagueña. Es por ello que surgió la iniciativa de desarrollar un prototipo de sistema web que difunda tal cultura.

Sin embargo, debido a que la cultura está ligada a las danzas, música, teatro, poesía, literatura, pintura, etc., y éstas a su vez están intrínsecamente vinculadas a la emoción, es necesario que el sistema de e-cultura siga las premisas que establece el diseño emotivo de los sistemas [1, 4, 16]. Para ello, en este trabajo se tuvo en cuenta la propuesta de diseño emotivo de Donald Norman [15].

Dado que esta propuesta parte del supuesto que la incorporación del diseño emotivo mejora la usabilidad de los sistemas web de e-cultura [11], se definió una metodología ágil para el desarrollo de este tipo de sistemas denominada DeGuiCSEC.

Se partió de una metodología ágil de desarrollo que involucra prototipación, método Feature Driven Development (FDD) [17], porque el grado de interacción que se requiere es alto y fue necesario un proceso de diseño iterativo que permita una rápida prueba y evaluación de distintas alternativas, impactando así en la calidad del sistema.

Para optimizar la interacción se tuvieron en cuenta los aspectos vinculados a la usabilidad, considerando las métricas del estándar ISO/IEC 9126-2 [13]. Además, debido a que FDD no hace énfasis en la recopilación de datos para definir el modelo global (primera etapa) sino en cómo se realizan las fases de diseño y construcción,

para la consecución de los mismos se implementan historias de usuario [3] y entrevistas a informantes calificados; y como se trata de una aplicación web se utilizan los modelos que propone Object Oriented Hypermedia Design Method (OOHDM) [2, 21] en la fase de diseño.

2. Interacción en sistemas de e-cultura

La digitalización está cambiando profundamente la forma de acceder a la cultura de una determinada región. Por ejemplo, se habla de la biblioteca digital, arte digital, universidad virtual, museo virtual para referirse a estos ámbitos afectados por las TICs. A través de estas tecnologías se está creando la más amplia red de información jamás existente.

Pero tecnología no significa necesariamente progreso; ofrece oportunidades pero también establece nuevas problemáticas. Una de las características de e-cultura es que la evolución en medios de comunicación digitales han llevado a un proceso de convergencia cultural, entre diferentes disciplinas y dominios en el campo de las artes y la cultura. En las sociedades puede causar una especie de tensión entre lo mundial y lo local (sin olvidarse de las propias raíces); y entre lo tradicional y el modernismo [19]. Los ciudadanos alrededor del mundo están cada vez más preocupados debido que los acelerados procesos de globalización e innovación tecnológica están llevando a la homogenización de la cultura [5].

Es notable la necesidad de desarrollar productos de e-cultura que permitan promover con diversos fines la cultura de una determinada región. Sin embargo, surgen diversas problemáticas asociadas al proceso de desarrollo de tales productos. Una de las cuestiones más enigmáticas fue definir una forma de lograr emotividad en un producto de e-cultura. Estudios previos [10-12] sostienen que para ello es necesario mejorar la interacción.

Ya que el diseño de un sistema influye en las actitudes e intenciones de los usuarios, el principal esfuerzo radica en lograr un diseño emotivo que provoque en los usuarios (sea cual fuese su cultura nativa) sensaciones análogas a las que causa a quienes viven y conocen la cultura santiagueña. Por ello fue necesario incorporar los principios del diseño emotivo (propuesto por Donald Norman) a un método de desarrollo, de manera que asegure una integración óptima a las actividades del mismo.

3. Cuestionarios de usabilidad

Una de las formas de medir la usabilidad que más éxito ha tenido es la de realizar estas medidas utilizando cuestionarios especialmente diseñados para tal propósito [14, 24]. Consiste en diseñar un cuestionario "tipo test", donde es necesario contestar una colección de preguntas que deben responderse entre un rango determinado de respuestas. El principal motivo para realizar estos cuestionarios es recoger respuestas concretas proporcionando datos comprobables, por ejemplo mediante estudios estadísticos. Los cuestionarios [14, 24] más relevantes en el ámbito de la usabilidad son:

- QUIS (Questionnaire for User Interface Satisfaction): Es una técnica de valoración subjetiva. Este cuestionario se enfoca en los aspectos de la interfaz del usuario y consta de 5 secciones, la primera de las cuales valora las reacciones del usuario mientras utiliza el sistema. Las secciones restantes valoran la pantalla, la terminología y el sistema de información, de aprendizaje y las capacidades del sistema.
- SUMI (Software Usability Measurement Inventory): Es un inventario de medidas de usabilidad que forma parte del proyecto global MUSiC 81. Este cuestionario valora la calidad de uso de un sistema o de un prototipo. Está referenciado en estándares de calidad ISO como el ISO 9241-10 y en el ISO 9126-2.
- WAMMI (Web Site Analysis and MeasureMent Inventory): Surge como extensión de SUMI para orientarlo hacia la medición de la usabilidad en la web. Este es un cuestionario basado en escenarios que trata de descubrir información acerca de lo que piensan los visitantes de los sitios web en cuanto a su calidad de uso.
- MUMMS (Measuring the Usability of Multi-Media Systems): Surge como una extensión de SUMI, trata de evaluar la usabilidad de los productos multimedia en general y es contestado por usuarios finales. Está enfocado en obtener el conocimiento adquirido por éstos. Fue desarrollado por un grupo de investigación de la Universidad de Cork, Irlanda.
- CSUQ (Computer System Usability Questionnaire): Desarrollado por IBM, se compone de 19 preguntas. Cada pregunta es una afirmación y se califica a cada una en una escala de siete puntos desde "totalmente en desacuerdo" hasta "totalmente de acuerdo".
- Words: Se basa en las 118 palabras utilizadas por Microsoft en sus “tarjetas de reacción de producto”. Cada persona que completa el cuestionario es libre de elegir tantas palabras como desee, teniendo en cuenta aquellas que describen mejor a su interacción con el sitio web.
- SUS (System Usability Scale): Fue desarrollado en 1986 por la Digital Equipment Corporation

como parte de la aplicación de ingeniería de usabilidad a los sistemas de oficina. Su propósito era proporcionar un test fácil de completar (número mínimo de cuestiones), fácil de puntuar y que permitiera establecer comparaciones cruzadas entre productos; es decir, que fuera posible comparar unas soluciones con otras y optar por la mejor.

4. DeGuiCSEC

Como resultado de la investigación, se elaboró una nueva metodología ágil para el desarrollo de sistemas web de e-cultura, basada en FDD y que incorpora pautas de diseño emotivo para mejorar la usabilidad. Esta metodología se denomina Desarrollo Guiado por Características para Sistemas de E-Cultura (DeGuiCSEC). En esta apartado se presentan sus principales aspectos.

4.1. Descripción del proceso de desarrollo

A continuación se describe cada una de las etapas, con sus respectivas entradas, actividades y salidas:

Etapa: Recolección de datos

- Entrada: Ninguna.
- Actividades:
    - Escoger los rasgos culturales que se van a considerar para el sistema de e-cultura. Para ello, se tiene en cuenta los 12 rasgos característicos de una cultura: creeres, hablares, cantares, bailares, contares, pensares, saberes, haceres, sentires, vivires, luchares y mirares [10-12].
    - Identificar a los informantes calificados (IC) de acuerdo a los rasgos escogidos. Estos son personas conocedoras de la cultura.
    - Entrevistar a los IC para obtener información asociada a los rasgos culturales.
    - Buscar y revisar documentación asociada a los rasgos escogidos (libros, publicaciones, notas periodísticas, videos, sitios web, etc.).
    - Seleccionar contenidos culturales relevantes, referidos a los rasgos escogidos.
    - Obtener historias de usuario para registrar las necesidades de interacción del usuario.
- Salidas:
    - Resumen y/o informe de contenidos culturales acorde a los rasgos escogidos
    - Conjunto de historias de usuario relacionadas con la interacción usuario-sistema.

Etapa: Desarrollar un Modelo Global

- Entrada: Resumen y/o informe de contenidos culturales acorde a los rasgos definidos en el alcance, e historias de usuario.
- Actividades
    - Identificar las clases, atributos y métodos a partir del informe de contenidos culturales y de las historias de usuario.
    - Establecer las relaciones y dependencias entre clases.
    - Definir el contexto del sistema de e-cultura.
    - Consultar nuevamente a los usuarios y/o informantes calificados para aclarar dudas u obtener información complementaria, cuando sea necesario.
    - Refinar el diagrama de clases (desde etapas posteriores) a medida que se tiene más conocimiento del sistema de e-cultura que se desea desarrollar.
- Salida: Diagrama de clases (aproximación inicial al sistema de e-cultura que posteriormente puede ser refinado).

Etapa: Construir Lista de Características

- Entrada: Resumen y/o informe de contenidos culturales acorde a los rasgos definidos en el alcance, historias de usuario y diagrama de clases del modelo global.
- Actividades
    - Definir una lista de características inicial. Una característica [17] es una pequeña función que es valorada por el cliente expresada en la forma <acción> <resultado> <objeto>, a partir del análisis de los elementos de entrada a la etapa.
    - Identificar las características que sean demasiada complejas para ser implementadas dentro de 2 semanas, y luego descomponerlas en varias características más pequeñas hasta que encajen en ese período de tiempo.
- Salida: Una lista de características del sistema de e-cultura, que resume las funcionalidades que se espera del sistema, donde cada una puede ser totalmente desarrollada en un período menor a 2 semanas.

Etapa: Planear por Característica

- Entrada: Lista de características e historias de usuario.
- Actividades

- Identificar las dependencias entre características, teniendo en cuenta la dependencia funcional o de implementación.
- Identificar las características vinculadas a las historias de usuario y asignarle las prioridades que se identificaron en las mismas.
- Establecer la secuencia de desarrollo dentro de un proceso iterativo, acorde a las prioridades y dependencias identificadas.

- Salida: Un plan de desarrollo de alto nivel, donde cada característica tiene identificada su dependencia, asignada una prioridad, y un orden en la secuencia de desarrollo dentro de un proceso iterativo.

Etapa: Diseñar por Característica

- Entrada: Diagrama de clases del modelo global, lista y planeación de características.
- Actividades
  - Elegir un grupo de características (según la secuencia y prioridades definidas en la planeación) y realizar el diseño Navegacional y diseño de Interfaz Abstracta [2, 21].
  - Inspeccionar el diseño para contrastarlo con los puntos de la Lista de Comprobación de Diseño Emotivo (LCDE).
  - Rediseñar (cuando sea necesario) para cumplir con los puntos insatisfechos de la LCDE, y luego inspeccionar nuevamente. Esto se repite hasta que se cumpla con todos los puntos de la LCDE.
- Salida: Diseño Navegacional (Esquema de Clases Navegacional y Esquema de Contextos Navegacionales) y Diseño de Interfaz Abstracta (Diagramas de Configuración ADV y Mapas AVD) [2, 21] que satisfacen los puntos de la LCDE.

Etapa: Construir por Característica

- Entrada: Diseño Navegacional y Diseño de Interfaz Abstracta para el sistema de e-cultura.
- Actividades
  - Codificar según lo diseñado en la etapa anterior.
  - Inspeccionar el código para verificar el cumplimiento de la Lista de Comprobación de Usabilidad (LCU).
  - Realizar pruebas unitarias.
  - Volver a la atapa de diseño por característica y codificar (cuando sea necesario) para cumplir con todos los puntos de la LCU, luego inspeccionar nuevamente. Repetir esto hasta que se cumpla con todos los puntos de la LCU.
  - Realizar pruebas de integración
  - Corregir errores de codificación y de implementación cuando sea necesario.
- Salida: Código fuente (en distintos formatos según la/s tecnología/s usada/s), versiones parciales sucesivas de la aplicación que cumple con los puntos de las LCDE y LCU. Finalmente se obtiene un sistema de e-cultura completo.

4.2. Listas de comprobación

En este apartado se presentan todos los puntos (o reglas) de verificación de las Listas de Comprobación de Diseño Emotivo y los principales puntos de la Lista de Comprobación de Usabilidad [9]. La lista de usabilidad consta en total de 136 reglas.

- Lista de Comprobación de Diseño Emotivo (nivel reflexivo):

ReglaDE1: ¿El diseño y su contenido no apelan al intelecto de los usuarios?

ReglaDE2: ¿Hay cohesión en el diseño con un foco y una visión? ¿Tiene un mensaje claro?

ReglaDE3: ¿Las representaciones de cuestiones culturales asociados a hechos tristes/desafortunados están acompañados de sonidos de tempos lentos y tonos menores?

ReglaDE4: ¿Las representaciones de cuestiones culturales asociados a hechos felices/agradables están acompañados de sonidos rápidos, música melódica que es bailable, con sonidos armoniosos y rangos relativamente constantes de tono y volumen?

ReglaDE5: ¿Las representaciones de cuestiones culturales asociados a hechos sombríos están acompañados de sonidos de tempo rápido, con discordancia y cambios abruptos de tono y volumen?

ReglaDE6: ¿Se utiliza música de fondo que no sea intrusiva en el pensamiento del navegante?

ReglaDE7: ¿El usuario tendrá siempre el control y sabrá qué está pasando en todo momento?

ReglaDE8: ¿Se proporciona al usuario un conjunto de opciones de personalización a la interfaz de la aplicación que permita darle toques individuales?

ReglaDE9: ¿Las opciones de personalización serán de fácil acceso, fácil de recordar y fácil de aprender para el usuario?

ReglaDE10:¿La personalización estará dada por la combinación de múltiples piezas sencillas?

ReglaDE11:¿Los usuarios podrán expresar sus pensamientos registrando de alguna manera sus opiniones, comentarios y/o pensamientos?

ReglaDE12:¿El usuario tendrá suficiente información previa antes de pasar a algún escenario distinto dentro del sistema?
ReglaDE13:¿Se utilizan imágenes, videos, películas, fotos, recortes periodísticos o cualquier tipo de imagen?

• Pautas de un buen diseño gráfico:

ReglaDE14:¿Existe una integración/interacción entre los elementos del diseño? ¿Existe una armonía en el lenguaje visual?
ReglaDE15:¿Se utiliza a lo sumo 3 familias tipográficas, usando cursivas y variaciones de tamaño?
ReglaDE16:¿Se intenta guiar el foco de atención en un orden?
ReglaDE17:¿La elección de colores es justificada?
ReglaDE18:¿Se evitó agregar elementos innecesarios? (regla de menos es más)
ReglaDE19:¿Se dejan espacios es blanco? ¿No se trata de llenar todo el espacio con algún elemento de diseño? ¿Se utilizan espacios en blanco como recurso de diseño?
ReglaDE20:¿No se utiliza tipografía demasiado grande ni con diseño extravagante? ¿La tipografía seleccionada es legible?
ReglaDE21:¿Las formas, colores e imágenes podrán ser interpretados por una amplia audiencia? ¿Se intentó ser universal?
ReglaDE22:¿Se crean contrastes visuales entre los elementos de diseño para crear la ilusión rectas, curvas y variaciones de densidad?
ReglaDE23:¿Se usan luces y sombras, y transiciones de color en los elementos de diseño?
ReglaDE24:¿Se tomaron decisiones justificadas y con convicción para el color, forma, posición y distancia (respecto de otros) de los elementos de diseño?
ReglaDE25:¿Las ilusiones ópticas obtenidas (diseño, proximidad entre objetos, formas, rectas y curvas, etc.) son las que inicialmente se pretendían?
ReglaDE26:¿Se usan imágenes propias y/o personalizadas en lugar de imágenes muy conocidas?
ReglaDE27:¿Se ignora la moda y no se diseña lo que audiencia espera?
ReglaDE28:¿Se evita el uso de una distribución plana de los elementos de diseño? ¿Se crea una ilusión de 3 dimensiones?
ReglaDE29:¿Se evita el uso de la simetría en los elementos de diseño?

- Lista de comprobación ad hoc de Usabilidad según ISO/IEC 9126-2

ReglaU1: ¿Las funciones (o tipos de funciones) son definidas de manera comprensible en la descripción del sistema?
ReglaU2: ¿Las funciones que requieren demostración cuentan con ella?
ReglaU3: ¿Las funciones están correctamente descriptas en la documentación de usuario y/o facilidad de ayuda?
ReglaU4: ¿Son consistentes las operaciones similares para que puedan ser llevadas a cabo?
ReglaU5: ¿Los mensajes del sistema podrán ser fácilmente entendidos?
ReglaU6: ¿Los usuarios podrán personalizar las funciones y procedimientos de operaciones para su conveniencia?
ReglaU7: ¿Se validan los datos ingresados por el usuario?
ReglaU8: ¿Las funciones tienen la capacidad de evitar operaciones incorrectas?
ReglaU9: ¿Los elementos de interfaz de usuario podrán ser personalizados en su apariencia?
ReglaU10: ¿Son atractivas las interfaces de usuario?

• Pautas de accesibilidad para usuarios con discapacidad física.

ReglaU11: ¿Se asegura que el orden del código fuente y el orden de la presentación visual es el mismo y que no ha sido alterado mediante hojas estilo?
ReglaU12: ¿Se complementa el texto del enlace con otro fragmento de texto que permanece oculto y que contenga una descripción más detallada del contenido del mismo?
ReglaU13: ¿Se establece las propiedades *name*, *description* y tabIndex a los objetos no textuales, ya sea desde el panel llamado "Accesibilidad" desde la herramienta de autor Macromedia Flash o mediante ActionScript 2.0/3.0?
ReglaU14: ¿Se proporciona validación cliente-servidor de los ingresos de usuario y se añade texto de error adyacente al ingreso del campo inválido?
ReglaU15: ¿Se utilizan atributos del lenguaje HTML para especificar el idioma en contenido Flash, ya sea con el atributo *lang* de manera global para el documento o aplicándolo solamente al flash embebido con la etiqueta *<object data="languages.swf" lang="..." xml:lang="...">?*
ReglaU16: ¿Se utilizan manejadores de eventos redundantes de ratón y teclado en Flash para asegurarse que los usuarios sean capaces de recibir la misma información independiente del dispositivo que utilice?
ReglaU17: ¿Se proporciona un control para desactivar el sonido que se reproduce automáticamente en Flash?
ReglaU18: ¿Se proporciona un script que advierta al usuario un tiempo límite para realizar alguna

tarea está por expirar y se proporciona una manera de extenderla si lo desea?

ReglaU19: ¿Se establece la propiedad *silent = true* a objetos que se desea que sean ignorados por tecnologías asistentes?

ReglaU20: ¿Se aplican subtítulos a contenido multimedia sincronizado pregrabado?

ReglaU21: ¿A todo contenido no textual se lo encierra entre etiquetas *<object>…</object>*, y se establece un nombre descriptivo en la propiedad *name* de dicha etiqueta?

ReglaU22: ¿Se crean contenidos que parpadeen menos de 5 segundos?

ReglaU23: ¿Se usan elementos semánticos de acuerdo a su significado y no según su apariencia?

ReglaU24: Cuando se utilizan variaciones de la presentación del texto para transmitir información ¿Se proporciona una manera adicional de transmitir la misma información que no dependa de la variación de la presentación del texto?

ReglaU25: ¿Se proporciona un audio que describa el mismo texto que se encuentra en la imagen de un CAPTCHA?

ReglaU26: ¿Se usa una herramienta para asegurar que el contenido no viola el umbral general de destello o destello rojo?

ReglaU27: ¿Se colocan imágenes animadas en formato *gif* para que dejen de parpadear después de n ciclos (dentro de los 5 segundos)?

ReglaU28: ¿Se reproducen sonidos sólo cuando el usuario lo solicita?

ReglaU29: ¿Se usa alguna tecnología para incluir contenido con destello que pueda ser desactivado mediante un agente de usuario?

ReglaU30: ¿Se proporcionan subtítulos optativos?

ReglaU31: ¿Se combinan imagen y texto adyacente cuando enlazan un mismo recurso?

ReglaU32: ¿Se utiliza en la etiqueta *<p>* el atributo *dir=ltr* para especificar la dirección del texto de izquierda a derecha, o *dir=rtl* para derecha a izquierda cuando se quiere mezclar la dirección de texto en una línea en código HTML?

ReglaU33: Cuando los posibles valores de un elemento *<select>* dependen del valor seleccionado en otro *<select>*. Para evitar el cambio de contexto ¿el evento *onchange* de la etiqueta html *<select>* independiente convoca una función JavaScript que actualice los posibles valores del *<select>* dependiente?

ReglaU34: Cuando se usa una imagen decorativa como parte de un hipervínculo ¿Se convocan funciones JavaScript mediante los eventos *onmouseover* y *onmouseout* para cambiar la imagen cuando el mouse se ubica por encima o se aleja del contenido de la página, y los eventos *onfocus* y *onblur* para cambiar la imagen cuando el elemento recibe o pierde el foco?

ReglaU35: ¿En lugar de utilizar las funciones *document.write* o *object.innerHTML* se utilizan las funciones del Modelo de Objetos del Documento (DOM por sus siglas en inglés) como *createElement(), createTextNode(), appendChild(), removeChild(), insertBefore()* o *replaceChild()*?

ReglaU36: ¿Se utilizan scripts para controlar el destello y/o parpadeo de contenidos y detenerlo antes de 5 segundos?

La primera lista se elaboró luego de revisar la obra de Donald Norman, en la cual el autor comenta y reflexiona sobre las características y pautas que son tomadas en cuenta en el diseño de productos fabricados en masa. Casi todos son productos físicos a los que hace referencia (como automóviles, indumentaria, muebles, etc.), es por ello que algunas de sus pautas no pueden ser aplicadas al diseño de algo intangible como son los sistemas basados en software (como el desgaste progresivo y los aromas de los productos por ejemplo). Por lo tanto, luego de analizar los lineamientos, se definieron pautas en la lista de manera que puedan ser aplicadas en el diseño de sistemas web, para que éste impacte en los usuarios a nivel reflexivo. Además, uno de los aspectos importantes que considera el autor es el "buen diseño grafico". Ya que esto es un concepto general, se recurrió al análisis y adaptación de las 20 reglas de un buen diseño gráfico que propone Samara [20]; con el objetivo de desglosar este concepto en puntos más específicos para la lista de comprobación.

En cuanto a la lista de usabilidad, se definió luego de analizar las métricas que establece la norma ISO/IEC 9126-2 [13]. Además, se decidió incorporar los Lineamientos de Accesibilidad a Contenido Web 2.0 (WCAG 2.0 siglas en inglés) del W3C [25] del nivel de conformidad "A", ya que unas de las métricas especifica que hay que tener en cuenta a usuarios con alguna discapacidad física. Para definir los puntos de la lista referente a esta norma de accesibilidad, se ingresó al sitio web oficial de W3C para identificar y discriminar cuales pautas corresponden a este nivel de conformidad, ya que en el sitio los lineamientos no están categorizados por nivel de conformidad sino por componentes de la programación a los que hace referencia (por ejemplo por pautas HTML, pautas FLASH, etc.).

5. Evaluación de usabilidad en sistemas de e-cultura

Para evaluar el impacto en la usabilidad que tiene DeGuiCSEC, se diseñó un nuevo cuestionario para sistemas de e-cultura para evaluar la usabilidad de un prototipo de sistema web de e-cultura santiagueña desarrollado con dicha metodología (el mismo puede accederse en http://ecultura-sde.6te.net/). Además, fue necesario planificar un experimento con una muestra de usuarios quienes registraron sus impresiones de uso en dicho cuestionario. Todo ello se detalla a continuación.

5.1. Cuestionario de usabilidad para sistemas de e-cultura

Muchas métricas externas de usabilidad son testeadas por usuarios tratando de usar una función. Los resultados podrían ser influenciados por las capacidades de los usuarios y las características del sistema anfitrión. Esto no invalida las mediciones ya que el software evaluado está corriendo bajo condiciones explícitamente especificadas por una muestra de usuarios representativa [13]. Además, DeGuiCSEC incorpora principios de diseño emotivo (en forma de lista de comprobación) con el objetivo de optimizar la interacción (enfocado en la usabilidad). Por lo tanto, para medir la eficiencia de la incorporación de estos principios se definió un cuestionario de usabilidad para sistemas de e-cultura (CUSEC), cuyo objetivo es registrar las impresiones de uso de una muestra de usuarios para evaluar la usabilidad de un sistema de e-cultura.

Para definir CUSEC se analizaron las preguntas de los cuestionarios de usabilidad estándares (en el capítulo 3), redefiniéndolas para que estén orientadas a otros aspectos en los que se enfocan las LCDE y LCU.

Algunos de estos cuestionarios estándar tienen preguntas similares entre sí, y algunas se enfocan en aspectos negativos (o no deseados) de la usabilidad. Para la definición del CUSEC todas las preguntas fueron redactadas apuntando a aspectos que contribuyen a la usabilidad con el objetivo de facilitar el análisis de las respuestas. A continuación se presentan las 19 preguntas que forman parte del CUSEC:

1) Es fácil navegar en esta aplicación.
2) Me siento satisfecho con la facilidad de uso de esta aplicación.
3) Es fácil encontrar lo que busco en esta aplicación.
4) Puedo hacer pequeños ajustes para hacer más confortable mi experiencia de uso.
5) Me sentí muy seguro al usar la aplicación, no se me presentaron dudas.
6) La aplicación se comportó en todo momento de manera esperada, respondiendo a las interacciones de manera predecible.
7) Usar por primera vez esta aplicación fue fácil porque es intuitiva.
8) Esta aplicación tiene todas las funciones, capacidades y contenidos que esperaba.
9) Creo que la aplicación podría ser usada por una amplia variedad de usuarios.
10) Cada vez que cometí un error usando la aplicación pude revertirlo fácilmente.
11) Las palabras cuyos significados son poco conocidos, son aclarados y/o definidos en la misma aplicación.
12) Es fácil recordar cómo se usa esta aplicación.
13) El uso de esta aplicación no es una pérdida de tiempo.
14) Recomendaría esta aplicación a personas conocidas.
15) La presentación de contenidos es consistente y no hay elementos innecesarios.
16) La aplicación es innovadora y original.
17) La aplicación no presenta contenidos visuales o sonoros que distraigan la atención.
18) Presenta tamaños de fuente adecuados.
19) La aplicación es visualmente atractiva.

Las preguntas del CUSEC se pueden dividir en dos grupos: las primeras 10 preguntas referidas a cuestiones relacionadas con la funcionalidad del sistema, y las 9 restantes referidas a cuestiones relacionadas con la estética y con los contenidos de la aplicación.

Para evaluar la usabilidad de un sistema de e-cultura, cada pregunta del CUSEC obligatoriamente debe ser contestada, usando la siguiente escala cualitativa: De acuerdo, Indeciso o En desacuerdo. A partir de las 3 opciones de respuestas se calcula un valor único que representa una medida cuantitativa de la usabilidad del sistema sometido a evaluación. A continuación se describe el cálculo. Como se mencionó anteriormente, cada una de las preguntas del cuestionario fue redactada para que refleje algún aspecto que contribuye a la usabilidad, por lo tanto se asignan los siguientes valores que se especifican en la Tabla 1.

Tabla 1. Valoración de respuestas del CUSEC.

Respuesta	Valor
En desacuerdo	1
Indeciso	2
De acuerdo	3

Luego, una vez contestado el cuestionario, para cada una de las 19 preguntas se calcula el promedio de los valores asociados a las respuestas de los usuarios, el cual siempre es un valor entre 1 y 3. Posteriormente, se suman

los 19 promedios correspondientes a cada una de las preguntas y el resultado de la sumatoria final siempre es un valor comprendido entre 19 y 57. En base a este valor, se determina el nivel de usabilidad del sistema conforme a los siguientes rangos:

- Si Suma ≥ 19 ∧ Suma < 35,2, entonces *nivel de usabilidad bajo*.
- Si Suma ≥ 35,2 ∧ Suma < 45,6 entonces *nivel de usabilidad intermedio*.
- Si Suma ≥ 45,6 ∧ Suma ≤ 57, entonces *nivel de usabilidad alto*.

Estos tres intervalos corresponden del 0%-40%, 40%-70% y 70%-100% respectivamente de los posibles valores de la sumatoria.

5.2. Planificación del experimento

Para evaluar la usabilidad del prototipo de sistema Web santiagueño se utilizó el CUSEC. El objetivo del cuestionario fue obtener un indicador cuantitativo de la usabilidad del prototipo; para evaluar el impacto de la incorporación de principios de diseño emotivo en la usabilidad.

Para aplicar el CUSEC, se escogió una muestra de usuarios multicultural, quienes luego de interactuar con el prototipo contestaron el mismo. Esta muestra fue seleccionada de acuerdo a 3 categorías de usuarios:

- Categoría 1: Usuarios que crecieron en la cultura santiagueña, quienes contestan el cuestionario según el grado de identificación con su propia idiosincrasia.
- Categoría 2: Usuarios provenientes de otras culturas de Argentina, quienes comparten el mismo territorio nacional y que tienen (probablemente) algunas nociones mínimas de la cultura santiagueña.
- Categoría 3: Usuarios provenientes de culturas de otros países, quienes probablemente conozcan muy poco, o casi nada, de la cultura de Santiago del Estero.

De acuerdo al estándar ISO/IEC 9126-2 [13] contar con una muestra de usuario de al menos 8 individuos es suficiente para obtener valores fiables en la evaluación de la usabilidad. Por lo tanto, se escogieron al menos 3 usuarios de cada categoría. Además se trató de contar con un grupo de edades heterogéneas, y con distintos niveles de familiaridad y destreza con Internet. Una condición fue que todos los usuarios, de los tres grupos, tengan dominio del idioma español ya que tanto el prototipo como el cuestionario en línea están disponibles en este idioma.

Por correo electrónico se envió a cada usuario un mensaje solicitándoles su colaboración y especificando la tarea y el objetivo. Además, se les indicó el enlace para acceder al prototipo (http://ecultura-sde.6te.net/) y el enlace al CUSEC en línea (https://docs.google.com/forms/d/1ajaBkCqlA5Z77N8MuwnHbsg1cxt-_difPcK9UDGwC5k/viewform). De esta forma se registraron sus impresiones de uso durante un período de evaluación de aproximadamente un mes.

5.3. Interpretación de los resultados

Luego de que todos los usuarios de la muestra completaron el cuestionario en línea, se procesaron sus respuestas. En base a ello se calculó el nivel de usabilidad del prototipo, de acuerdo al procedimiento también descripto en el apartado 5.1.

En esta sección se presenta el nivel de usabilidad calculado, y se realiza el análisis de valores extremos discriminando por categoría de usuarios.

En la Tabla 2 se reflejan todos los valores obtenidos en el procesamiento, desde la valoración de las respuestas de los usuarios hasta el nivel de usabilidad calculado.

Tabla 2. Análisis de las respuestas obtenidas mediante la aplicación del cuestionario.

N° de pregunta	Promedio de valores de respuestas de usuarios categoría 1	Promedio de valores de respuestas de usuarios categoría 2	Promedio de valores de respuestas de usuarios categoría 3
1	3	3	3
2	3	3	3
3	2,75	2,66	2,33
4	2	2	2,33
5	3	2,33	2,66
6	2,75	2,33	2,33
7	3	2,66	2
8	2,25	2,66	2,33
9	3	3	2
10	2,75	2	2,33
11	2,75	2,33	2,66
12	2,75	3	2,66
13	2,75	2,66	3
14	2,75	2,33	2
15	2,75	2,66	2,66
16	2,75	2,33	2,33
17	3	3	2,66
18	3	2,66	3
19	3	2,66	2,33
Sumatoria de promedios de la categoría	**53**	**49,33**	**47,66**

Primero, se analizaron en conjunto los valores extremos de promedios que coindicen entre las categorías de usuario; ellas son las preguntas N°1, N°2, N°4, N°7, N°9, N°17 y N°18.

Los 3 grupos registraron el máximo valor promedio igual a 3 para las preguntas N°1 (“es fácil navegar en esta aplicación”) y N°2 (“me siento satisfecho con la facilidad de uso de esta aplicación”). Esto se considera positivo ya que implica que todos los usuarios (de diversas culturas) estuvieron satisfechos con la facilidad de uso.

La categoría 1 y la categoría 2 de usuarios coinciden con máximo promedio igual a 3 para las preguntas N°9 (“creo que la aplicación podría ser usada por una amplia variedad de usuarios”) y N°17 (“la aplicación no presenta contenidos visuales o sonoros que distraigan la atención”). Estos usuarios que provienen de la cultura santiagueña (categoría 1) y otros que son de otras culturas del resto del país (categoría 2) tal vez consideran que no hay contenidos que distraigan la atención porque conocen, en mayor o menor medida, la cultura santiagueña e interpretan correctamente la esencia y/o significado de los rasgos culturales. Por otra parte, la categoría 3 registró el valor mínimo de promedio igual a 2 para la pregunta N°9, por lo tanto podría interpretarse que esos usuarios provenientes de culturas de otros países consideran que el prototipo no sería “universal” en cuando a la facilidad de uso.

Las categorías 1 y 2 coinciden con el valor mínimo igual a 2 en la pregunta N°4 (“puedo hacer pequeños ajustes para hacer más confortable mi experiencia de uso”); pero la categoría 3 para esa misma pregunta registró un valor un poco mayor con 2,33; es decir que los usuarios del grupo 3 valoraron un poco más las opciones de personalización o que tal vez les fueron más útiles.

También las categorías 1 y 3 de usuarios coinciden con el máximo promedio con valor igual a 3 en la pregunta N°18 (“presenta tamaños de fuente adecuados”) esto indicaría que estos grupos de usuarios no tuvieron inconvenientes en la lectura. Además, para la pregunta N°7 (“usar por primera vez esta aplicación fue fácil porque es intuitiva”) la categoría 1 registró el mayor promedio con valor 3 y para la misma pregunta la categoría 3 registró el menor promedio con valor 2; los usuarios de la categoría 3 (provenientes de otros países) podrían necesitar mayor información para que la facilidad de uso no esté sujeta a suposiciones de los usuarios.

Luego se consideraron las preguntas con valores de promedio extremos disparares (no coincidentes entre las categorías). Ellas son las preguntas N°5, N°10, N°12, N°13, N°14 y N°19.

Las usuarios de la categoría 1 registraron su mayor promedio igual a 3 en las preguntas N°5 (“me sentí muy seguro al usar la aplicación, no se me presentaron dudas”), y N°19 (“la aplicación es visualmente atractiva”). Podría estimarse que no tuvieron dudas al interactuar con el prototipo porque estos usuarios viven inmersos en la cultura santiagueña y tienen preconceptos; lo cual habría influenciado de manera positiva al momento de interactuar debido a la familiaridad del contenido.

Los usuarios de la categoría 2, el menor promedió igual a 2 se registró también para la pregunta N°10 (“cada vez que cometí un error usando la aplicación pude revertirlo fácilmente”). Se podría deducir que el prototipo a éstos usuarios no les resultó lo suficientemente intuitivo tal vez porque el mismo no proporcionó la suficiente información previa antes de realizar una acción. Sin embargo, los inconvenientes no habrían sido muchos ya que el valor de promedio coincide con el valor medio (recordar que el promedio toma valores reales entre 1 y 3). Además, la categoría 2 de usuarios registró el promedio máximo con valor 3 para la pregunta N°12 (“es fácil recordar cómo se usa esta aplicación”). Ya que manifestaron que es fácil de recordar podría deducirse que interactuaron más de una vez con el prototipo, que pudieron recordar su funcionamiento, y que además los contenidos culturales captaron su interés.

Los usuarios de la categoría 3 también registraron el mayor promedio con valor 3 en la pregunta N°13 (“el uso de esta aplicación no es una pérdida de tiempo”). Esto implicaría que los usuarios de otros países consideraron fructífero el tiempo invertido para conocer la cultura santiagueña, y que les resultó interesante. Además, esta categoría de usuarios también registró el mínimo promedio con valor 2 para la pregunta N°14 (“recomendaría esta aplicación a personas conocidas”). Estos usuarios en cierta medida estarían “indecisos” de recomendar la aplicación a personas conocidas, ya que para ellos no fue lo suficientemente intuitivo el prototipo, y supondrían que las personas allegadas a ellos (probablemente de su misma cultura) tendrán también pequeñas dificultades con la facilidad de uso.

Por último, se analizaron los niveles de usabilidad del prototipo según cada categoría. Estos valores pueden observase en la última fila de la Tabla 2:

- Para la categoría 1 se obtuvo el valor de sumatoria 53, por lo tanto, según los usuarios santiagueños el nivel de usabilidad es alto.
- Para la categoría 2 se obtuvo el valor de sumatoria 49,33, entonces según los usuarios del resto del país el nivel de usabilidad es alto.
- Para la categoría 3 se obtuvo el valor de sumatoria 47,66, por lo tanto, según los usuarios de otros países el nivel de usabilidad es alto.

Luego, según cada uno las 3 categorías de usuario el prototipo alcanza un nivel de usabilidad alto. Además, puede observarse que los 3 valores son decrecientes a medida que aumenta la distancia geográfica de los usuarios de la cultura que se trata (según el origen de los usuarios de cada categoría definida: Santiago del Estero – resto de Argentina – resto del mundo). Sin embargo se encontraron varias similitudes en los valores extremos

entre la categoría 1 y la categoría 2; pero la categoría 3 en general presentó valores diferentes respecto a las otras categorías de usuarios.

6. Conclusiones y trabajos futuros

Con esta propuesta se pretendió contribuir a la optimización de la usabilidad de los sistemas de información de e-cultura, lo cual se considera logrado. El principal resultado de este trabajo es la definición de una metodología de desarrollo ágil para sistemas de e-cultura denominada DeGuiCSEC. La misma formaliza la incorporación de principios de diseño emotivo y de usabilidad que influyen en la calidad del sistema. DeGuiCSEC considera los recursos de interacción apropiados para lograr emotividad en los usuarios, mediante actividades tendientes a la elección de contenidos y su presentación óptima (texto, imagen y sonido) a usuarios de diversas culturas.

A lo largo de esta investigación, se logró mostrar que la metodología DeGuiCSEC permite desarrollar sistemas de e-cultura con alto nivel de usabilidad. Para ello, se desarrolló un prototipo de sistema web de e-cultura usando dicha metodología. Y se evaluó la usabilidad del prototipo mediante el CUSEC y un procedimiento de cálculo basado en medias aritméticas. De acuerdo al análisis de resultados realizado descripto anteriormente, se determinó que el prototipo presenta un alto nivel de usabilidad. En forma complementaria, se obtuvo información importante referida a aspectos de usabilidad en función del diseño emotivo y de las categorías de usuarios.

Por otra parte, cabe destacar la importancia de haber obtenido una versión inicial de sistema web de e-cultura de Santiago del Estero que permita difundir y promocionar la santiagueñidad a través de Internet. Además de contar con un nivel alto de usabilidad, hace posible que miembros de otras culturas conozcan y comprendan algunos de los aspectos más esenciales de la santiagueñidad. Todo esto, valiéndose de las oportunidades que presentan los sistemas de e-cultura para reforzar el sentido de identidad cultural de los santiagueños. Esto es muy importante ya que no se contaba, a la fecha, con un sistema web de estas características.

Asimismo, cabe resaltar la importancia del CUSEC, definido para evaluar la usabilidad del prototipo desarrollado con DeGuiCSEC. El mismo sería útil para evaluar la usabilidad de cualquier sistema de e-cultura; independiente de la cultura de la que se trate y de la metodología empleada para su desarrollo. Esto se debe a que fue definido considerando cuestionarios estándares de usabilidad.

A continuación se enuncian futuras líneas de investigación y/o desarrollo relacionadas con la temática del presente trabajo:

- Desarrollar el Sistema Web de e-cultura de Santiago del Estero, a partir del prototipo incorporando los 12 rasgos culturales y el manejo de cuentas de usuario para ampliar las opciones de personalización.
- Aplicar DeGuiCSEC para el desarrollo de sistemas e-cultura de otras culturas diferentes a la santiagueña.
- Ampliar la metodología DeGuiCSEC con los siguientes aspectos:
    - Incorporar en su lista de comprobación de usabilidad pautas de accesibilidad del Nivel de Conformidad “AA” y “AAA” de W3C; validándola con un prototipo evaluado por usuarios que presenten distintos niveles de discapacidad visual.
    - Abordar técnicas de Gestión del Conocimiento como herramientas para recolectar el conocimiento de los informantes calificados (Etapa “Recolección de datos” de DeGuiCSEC).
    - Incorporar lineamientos para el desarrollo de aplicaciones móviles de e-cultura.
- Realizar estudios comparativos de la usabilidad de distintos sistemas de e-cultura desarrollados con diferentes metodologías, contrastando con sistemas desarrollados con DeGuiCSEC.

7. Referencias

[1] Cardoso, S. y Vaartjes, R, “Designing emotions in online travel”, Sotopia. 1º edición, ISBN 9-0902-1413-5, Holanda, 2008.

[2] Casteleyn, S., Daniel, F., Dolog, P. y Matera, M., “Engineering Web Applications”, Springer, ISBN 978-3-540-92200-1, Berlín, Alemania 2009, pp. 75-77.

[3] Cohn, M., “User stories applied for agile software development”, Addison-Wesley Professional, 1° edición, ISBN 0-321-20568-5, Estados Unidos 2004.

[4] Conejera, O., Vega, K. y Villarroel, C., “Diseño Emocional, Definición, metodología y aplicaciones”, Facultad de Humanidades y Tecnología de la Comunicación, Universidad Tecnológica Metropolitana, Chile 2005.

[5] Dragojević, S., Dodd, D., Cvjetičanin, B. y Smithuijsen, C., “eCulture: The European Perspective Cultural Policy, Creative Industries, Information Lag”, Institute for International Relations, ISBN 953-6096-36-6, Zagreb, Croacia 2003.

[6] Drigas, A., Koukianakis, L. y Glentzes, J., "A dynamical system for e-culture services", WSEAS TRANSACTIONS on SYSTEMS, Issue 1, Volumen 5, Atenas, Grecia 2006, pp. 125-130.

[7] Drigas, A., Koukianakis, L. y Glentzes, J., "An E-Culture - E-Museums Environment for Common Citizens and Disabled Individuals", Int. J. of Digital Culture and Electronic Tourism 2009 - Vol. 1, Núm. 4, Atenas, Grecia 2009.

[8] Freixa Riba, V., "e-Cultura otra manera de participar en la Cultura de la ciudad", Portal Iberoamericano de Gestión Cultural. Boletín GC: Gestión Cultural Nº10: Cultura Digital, ISSN: 1697-073X, Barcelona, España 2004.

[9] Gallo, F. R. y Palavecino, R. A.," E-Cultura: Prototipo de sistema web de la santiagueñidad", Quinto Simposio Internacional de Investigación; Interdisciplinariedad, Multidisciplinariedad y/o Transdisciplinariedad: En la búsqueda de respuestas desde la experiencia de la investigación, Universidad Católica de Santiago del Estero, Departamento Académico San Salvador, San Salvador de Jujuy, Jujuy, Agosto 20-22 2013.

[10] Herrera, S., Clusella Cornejo, M., Luna, P., Mitre, G. y Santillán, M. Santiagueñidad siglo XXI – Retroprospectivación sistémica de la cultura de Santiago del Estero (RA), Fundación Argentina para el talento y el ingenio, Lucrecia, ISBN 978-987-1754-81-6, Santiago del Estero, Argentina 2011.

[11] Herrera, S., Clusella Cornejo, M., Mitre, M., Santillán, M. y García, C., "An interactive information system for e-culture", International Institute Galileo Galilei & Universidad Nacional de Santiago del Estero & Universidad Católica de Santiago del Estero, Santiago del Estero, Argentina 2010.

[12] Herrera, S., Zuaín, S., Mitre, M., Gallo, F. y Ávila, H., "Emotion and Communicability in e-culture Applications", International Institute Galileo Galilei & Universidad Nacional de Santiago del Estero, Santiago del Estero, Argentina 2011.

[13] ISO/IEC. "Software engineering - Product quality - Part 2: External metrics", ISO/IEC 9126-2, 2003.

[14] Lorés, J., "Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad. MPIu+a", Grupo de Investigación en Interacción Persona Ordenador e Integración de Datos, 2010, disponible en línea en: <http://www.grihohcitools.udl.cat/mpiua/metricasusabilidad.htm> Consultado Noviembre 2012.

[15] Norman, D., "Emotional Design", Editorial B. Books, ISBN 0-465-05135-9, Estados Unidos 2004.

[16] Palacio García, J., "Chat Lang: Interfaces Afectivas Síncronas", Escuela de Ingeniería, Departamento de Ingeniería en Sistemas Computacionales, Universidad de las Américas, Puebla, México 2004, capítulo II.

[17] Palmer, S. y Felsing, J., "A practical guide to feature-driven development", Prentice Hall 1° edición, ISNB 0-13-067615-2, Estados Unidos 2002.

[18] Ronchi, A., "eCulture: Cultural Content in the Digital Age", Springer, ISBN 978-3-540-75276-9, Alemania 2009.

[19] Salgado, F., "Problemáticas asociadas a las TICs en la sociedad de la información", 2011, disponible en <http://www.youblisher.com/files/publications/17/98846/pdf.pdf> Consultado Abril 2013.

[20] Samara, T.," Desing elements - A graphic Style Manual", Rockport, ISBN 978-1-59253-261-2, Estados Unidos 2007.

[21] Schwabe, D. y Rossi, G., "Developing Hypermedia Applications using OOHDM", Proceedings of Workshop on Hypermedia development Process, Methods and Models, Hypertext´98, Estados Unidos 1998.

[22] Schwarz, M., "e-Culture: Crossovers and Challenges", Institud Intelligente- und Informationssysteme, Versión en Inglés, Holanda 2006.

[23] Sorgdrager, W., "From ICT to E-culture", Netherlands Council for Culture, Holanda 2003, disponible en <http://www.cultuur.nl/files/pdf/adviezen/E-cultuur_engels.pdf> Consultado Marzo 2011.

[24] Tullis, T. y Stetson, J., "A Comparison of Questionnaires for Assessing Website Usability", Human Interface Design Department, Fidelity Center for Applied Technology Fidelity Investments 2004, disponible en: <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf> Consultado Noviembre 2011.

[25] World Wide Web Consortium., "Web Content Accessibility Guidelines (WCAG) 2.0", 2008, disponible online en: <http://www.w3.org/TR/WCAG/> Consultado Julio 2013.

Utilización del sistema SIG para el diseño de políticas de Vinculación Tecnológica

de San Pedro María Eugenia, Lasso Marta, Serón Natalia, Ramos Luis, Montenegro Cristian, Carrizo Alejandra

LabTEm - Laboratorio de Tecnologías Emergentes
Instituto de Tecnologías Aplicadas
Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral

Abstract

En la actualidad, los Sistemas de Información Geográfica (SIG) son herramientas muy utilizadas en la resolución de los problemas ambientales y no está del todo claro cómo es que se incorporan en la toma de decisiones, lo que es necesario para entender como deberían ingresar dentro de la formación de las ciencias ambientales. Esto nos ha llevado a reconstruir todo el proceso de toma de decisiones, a diferenciar a un SIG como soporte o apoyo, entender y analizar cómo se enlaza con las ciencias ambientales y sobre la importancia de la percepción de la realidad que tienen y requieren. Luego se analizó e indagó al problema ambiental como fundamento que permite ordenar el contenido y el programa en el proceso de formación en ciencias ambientales. Todo esto lleva a considerar a los SIG, como una herramienta ideal pero compleja que es, útil en diferentes formas e intensidades dentro de todo el proceso de toma de decisiones y a considerar a las "problemáticas ambientales" como una base fuerte para estructurar un proceso de formación en ciencias ambientales y por lo tanto útil para considerar dentro de este proceso a los SIG como herramienta para toma de decisiones.

Como uno de los objetivos de dicho proyecto se ha comenzado a trabajar en la construcción del paisaje productivo de la Región Norte de la provincia Santa Cruz, a efectos de detectar oportunidades y redes tecnológicas.

Palabras clave: *Sistema de Información Geográfica, Ciencias Ambientales, Paisaje Productivo. Percepción Remota.*

Introducción

La línea de investigación que se aborda en este proyecto, comenzó como una actividad de Vinculación Tecnológica relacionada con el medio local.[Serón et al. 2008], [de San Pedro et al. 2009], [Serón et al. 2010a], [Serón et al. 2010b], [Serón et al. 2011], [de San Pedro et al. 2012], [Serón et al. 2012a], [Serón et al. 2012b]

Con el objetivo de continuar en esta línea, y en el marco de la cooperación académica que la UNPA tiene con la Universidad Nacional Autónoma de México (UNAM), en particular con el Centro de Investigación en Geografía Ambiental (GIGA), se presentó un proyecto en el programa de Investigación en Ciencia y Tecnología de la Universidad Nacional de la Patagonia Austral (UNPA), con el objetivo general de la investigación y el desarrollo en nuevas tecnologías basadas en Sistemas Inteligentes y Software Libre, en el área de las ciencias ambientales.

Este nuevo proyecto donde el aporte de los SIG, en las ciencias ambientales, resulto de interés para el grupo, se está desarrollando en el Laboratorio de Tecnologías Emergentes (LabTEm), de la Unidad Académica Caleta Olivia (UACO). Dicho aporte está vinculado principalmente al tipo de información que éstos manejan y a la perspectiva de la realidad que nos proporcionan.

El proyecto se enmarca en una de las líneas de acción del área de Vinculación Tecnológica de la UNPA-UACO con el fin de analizar y diseñar lineamientos para el establecimiento de las políticas de construcción de vinculación derivadas de un estudio y registro del sector productivo de la Zona Norte de Santa Cruz, Patagonia, Argentina. Asimismo intenta constituirse en la matriz para el desarrollo de investigaciones futuras como soporte e integración de varios proyectos relacionados, dentro de la UNPA-UACO.

Sistemas de Información Geográfica

La Informática Social (IS) es un campo de conocimientos, teorías, metodologías y técnicas que contribuyen a comprender y examinar los aspectos sociales que intervienen en el éxito o fracaso de los diseños, gerencia/gestión y usabilidad de las Tecnologías de la Información y la Comunicación (TIC) al interactuar con contextos organizacionales y culturales en proceso de digitalización o virtualización.

Una de las más amplias y complejas aplicaciones de los SIG, es sin lugar a dudas en las ciencias ambientales y al mismo tiempo una de las más recurrentes junto a la planificación del territorio y la administración de los recursos naturales (Martin, 1985) (Bocco, 2000). De modo que el aprendizaje de la técnica y el uso de la información a través de los SIG, resultan importantes en la formación de una disciplina medio ambiental.

En definitiva, dice Chuvieco (2002), “los SIG son sólo herramientas para la mejor gestión de la información disponible del territorio. Lo cual implica las diferentes y diversas aproximaciones que se pueden tener sobre el mismo, algo que resulta bastante diverso y complejo”. Por otro lado es precisamente el manejo de este tipo de información “territorial”, por el que principalmente un SIG se diferencia de otras herramientas de soporte a la toma de decisiones.

Como confirma Gladstone (1999), los SIG son o pueden ser considerados como Sistemas de Soporte de Decisión y Planificación Integrado (SSDPI). Es decir que son herramientas de utilidad en parte o en la totalidad del proceso de toma de decisiones, que puede incluir la fase de inteligencia, diseño y elección. Pero particularmente el carácter espacial de la información con la que trabaja un SIG, hacen que estos sean llamados también Sistemas de Soporte de Decisión Espacial (SSDE) (Ascough, 2002).

De hecho no podemos considerar a un SIG como un producto cerrado sino más bien un compuesto de elementos diversos: ordenador, digitalizador, trazador gráfico, impresoras, y distintos programas computacionales orientados a una finalidad específica; aunque la tendencia, desde ya hace varios años, en

algunos productos sobre todo los comerciales, es tratar de recoger todos estos productos bajo una misma estructura (Chuvieco, 2002).

El aporte de los SIG, en las ciencias ambientales está vinculado principalmente al tipo de información que éstos manejan y a la perspectiva de la realidad que nos proporcionan. Son estas dos propiedades las que han permitido hacer comparables en un mismo sistema (espacial) información de la más diversa naturaleza. Y aunque esto se hace efectivo en todas las aplicaciones SIG, en las ciencias ambientales toman más relevancia debido al carácter holístico e integrador de su objeto de estudio “El medio ambiente”.

La definición de ciencias ambientales de la Organización de las Naciones Unidas para la Ciencia, la Educación y la Cultura (UNESCO), dice textualmente “El término ciencias ambientales implica más bien una orientación hacia la articulación de las ciencias naturales y sociales para abordar operacionalmente los problemas de medio ambiente” (UNESCO, 1989).

Cuando observamos la forma en que se abordan los problemas ambientales, incluso si éstos son del mismo tipo o tienen casi el mismo nombre, nos encontramos con una panacea de técnicas, métodos, disciplinas, tiempos, insumos, valoraciones y por supuesto decisiones.

Esta condición peculiar en las ciencias ambientales, desarrollada por el aporte e influencia de diferentes campos del conocimiento, resulta problemática cuando se trata de formalizarla (de manera consensuada) en el campo de investigación y sin confundirse y compararse con otros campos. En el aspecto formativo definir sus contenidos y competencias son prácticamente y naturalmente la expresión de la misma condición.

Al otorgar centralidad a las problemáticas ambientales en los contenidos dentro de este proceso formativo, podemos asumir que entre otros, los SIG pueden retomar esta centralidad para plantearlos en un ámbito formativo. Es decir “los SIG como herramienta para la toma de decisiones” se enseñarían a partir de su aplicabilidad a una problemática ambiental previamente definida.

Una de ellas tiene que ver con la planificación territorial, como una necesidad urgente para ordenar las actividades humanas, de modo de evitar riesgos, contener y frenar el deterioro de los ecosistemas y el efecto negativo multiplicador del impacto ambiental, a la vez de proteger la vida humana y procurar mayor bienestar a las poblaciones asociadas a espacios específicos.

En casos donde existe heterogeneidad ambiental y social es necesario realizar investigación enmarcada en temas como el reconocimiento de la vulnerabilidad, el riesgo, y el crecimiento urbano y la precariedad social y ambiental que genera, y que son inherentes a la planificación del uso del territorio y a la prevención de situaciones futuras no deseadas.

Otro grupo de problemáticas tiene que ver con el manejo del territorio, que hace referencia a las acciones directas que se realizan sobre territorios específicos, sean éstos urbanos, peri-urbanos o rurales. Estas acciones pueden ser de apropiación o uso de los bienes y servicios provistos por la naturaleza, de restauración o rehabilitación para compensar el daño ocasionado por acciones pasadas, o de conservación para prevenir la pérdida a futuro de biodiversidad o funciones del ecosistema que son vitales para la vida humana. Las decisiones que los actores locales toman para relacionarse con el entorno natural y social específico que constituye su territorio, están definidas por aspectos sociales, tales como la organización y las reglas internas; culturales como visiones, acceso a información, creencias y tradiciones; históricos que se expresan en herencias ecológicas, productivas; económicos, como los mercados y opciones de ingreso; o políticos, es decir impulsados por programas de gobierno. En este aspecto, las contribuciones del CIGA están enfocadas a comprender cómo operan estos factores para establecer modelos de manejo y cómo estos modelos de manejo se reflejan e impactan en el uso del territorio. Este entendimiento es crucial para ofrecer propuestas y alternativas para un manejo adecuado que evite la pérdida de áreas conservadas, la alteración hidrológica en cuencas, la erosión genética y cultural, la corrupción social y la pobreza.

Otra problemática tiene que ver con la evaluación y comprensión del cambio ocurrido en territorios específicos. La evaluación y comprensión del cambio en la cobertura de la vegetación y el de uso del suelo es fundamental para entender qué está

ocurriendo sobre la superficie terrestre en términos de su cobertura, y especialmente explorar y comprender los procesos sociales que definen las presiones humanas sobre los diferentes componentes del paisaje.

La evaluación del cambio de cobertura y de uso del suelo es posible gracias a las nuevas tecnologías geoespaciales y de cómputo que se han desarrollado, que permiten interpretar imágenes remotas (aéreas y satelitales) y generar modelos para explicar de manera más completa los cambios ocurridos y las tendencias.

La percepción remota permite obtener información de un área u objeto por medio del análisis de los datos que se han adquirido mediante algún dispositivo que no está en contacto físico con el objeto, área o fenómeno investigado. Es una ciencia que medirá las propiedades físicas de un objeto utilizando un sensor que esté a cierta distancia desde el objeto.

Los SIG y la percepción remota son una herramienta importante en la planificación ambiental y ordenamiento territorial. Las imágenes que proporcionen la percepción remota permiten obtener información actualizada sobre amplias áreas geográficas. Los datos permitirán obtener patrones y medir procesos a gran escala en forma directa, en lugar de inducirlo mediante mediciones a escala local.

La utilización de la percepción remota se fundamenta en que se puede lograr: la Radiación Electromagnética de cada objeto, Interacción con la atmosfera, Interacción con el objeto observado, y como Sensor para recibir la energía electromagnética reflejada o emitida.

Otras aplicaciones de la percepción remota pueden ser incluidas en: a) Inventario regional del medio ambiente para preparar estudios de impactos ambientales. b) Cartografía geológica para la exploración mineral y petrolífera. c) Inventario del agua superficial. d) Verificación de contenidos de salinidad en las principales corrientes de agua. e) Investigaciones forestales. f) Cartografía de la cobertura vegetal del suelo.

1.1. Utilización del sistema SIG para el diseño de políticas de VT

Como se señaló anteriormente, este trabajo se enmarca en una de las líneas de acción del área de Vinculación Tecnológica de la UNPA-UACO con el fin de analizar y diseñar lineamientos para el establecimiento de las políticas de construcción de vinculación derivadas de un estudio y registro del sector productivo de la Zona Norte de Santa Cruz, Patagonia, Argentina. Asimismo intenta constituirse en la matriz para el desarrollo de investigaciones futuras como soporte e integración de varios proyectos relacionados, dentro de la UNPA-UACO.

La metodología aplicada se basó en la difusión del proyecto hacia dentro de la Unidad Académica, más específicamente la puesta a disposición de la posibilidad de dar soporte a través del sistema GIS, a los proyectos relacionados con temáticas específicas de “Planificación territorial”. “registro de hitos socio-culturales-ambientales de la región”, “identificación de puntos de conflictos ambientales”, “actividades de ordenamiento territorial”.

Teniendo en cuenta los objetivos del presente proyecto, se procedió a establecer espacios interdisciplinarios de trabajo y se estableció como línea de trabajo la consulta y la interacción permanente entre las áreas de gestión de la Unidad.

Resultados y Objetivos

Los SIG como herramienta en la toma de decisiones pueden introducirse de manera adecuada en un proceso formativo dentro de las ciencias ambientales por medio de su aplicabilidad a una problemática ambiental.

Algunas de estas capacidades complementarias, son las de poder trabajar en equipos multidisciplinarios, conocer métodos y técnicas participativas y contar con habilidades en la coordinación de grupos.

En la Unidad Académica el desarrollo de las carreras, está íntimamente ligado a la demanda profesional de la zona norte de la Provincia de Santa Cruz y fuertemente asociada al desarrollo humano de la región.

Se encuentra atravesada por la dimensión histórica y ambiental, por lo que brinda un panorama novedoso para el desarrollo de nuevos campos de la ciencia, especialmente en la formulación de investigaciones interdisciplinarias. La UACO-UNPA posee carreras de las Ciencias Sociales, Exactas y Naturales, y este tipo de investigación, aportaría al desarrollo de una mejor comprensión de las necesidades regionales, los espacios de vacancia y favorecería la comprensión de la construcción social del paisaje de la región.

A través de un proyecto de Vinculación Tecnológica denominado "Construcción del paisaje productivo de la Región Norte de Santa Cruz. Oportunidades y redes tecnológicas.", se está trabajando básicamente, y como objetivo primario, identificar territorialmente las diferentes actividades que se desarrollan en torno a las actividades de producción extractivas.

La aplicación del sistema SIG permite el diseño de clusters según el tipo de actividad, localización y potencialidades de vinculación y transferencia.

El proyecto sobre esta línea tiene tres fases fundamentales:

La primera que estuvo ligada al establecimiento del soporte de construcción de la base de datos, tipos de datos y sistematización de los datos cualitativos y cuantitativos, recolectados por el área de VT. También se trabajó con los indicadores del proyecto, para la visualización de resultados y análisis comparativo con las otras dimensiones de la base.

Este proyecto se encuentra aún en desarrollo y en su segunda fase, que tiene por objetivo la puesta en marcha del "demo" y su evaluación con el equipo técnico del Proyecto de Vinculación.

Como proyecto de desarrollo se abordan aspectos sobre la adecuación del SIG que pueda abordar y satisfacer a las necesidades del proyecto de VT.

Por otra parte en el eje de la investigación, el grupo aborda desafíos de desarrollo de herramientas del propio sistema en adecuación a las necesidades del proyecto en cuestión.

Se trabaja en conjunto con los encuestadores y responsables del mismo a los efectos de que puedan también emplear herramientas del sistema, la transferencia entre los dos grupos de trabajo constituye uno de los objetivos de fortalecimiento del campo de trabajo hacia dentro del grupo y hacia afuera como grupo consolidado.

En cuanto al eje de extensión, el grupo trabaja en la comunicación de los resultados a modo de elaborar puentes entre el grupo SIG y los demás grupos de investigación.

2. Trabajos Futuros

Como una de las metas principales del grupo de investigación, apunta al desarrollo de actividades que permitan la integración de áreas multidisciplinares a efectos de poder brindar información a los diferentes sectores.

Como trabajos futuros, existen avances concretos con el INTA EEA Santa Cruz, con el sector Manejo de Recursos Hídricos, para el análisis la problemática hídrica en la zona de chacras de Caleta Olivia y Puerto Deseado; en un nuevo proyecto de vinculación tecnológica con el área de Patrimonio Cultural y Turístico de la zona norte de la provincia; y con vinculación académica, trabajar en la construcción de una matriz de la relación UNPA - Organizaciones Públicas / UNPA – Organizaciones Privadas.

3. Formación de Recursos Humanos

En el marco de este proyecto, se prevé insertar a integrantes alumnos de la carrera Ingeniería en Sistemas en la temática, lo cual posibilitaría llevar adelante el desarrollo del Proyecto Final para obtener la titulación de Ingeniero en Sistemas.

Capacitar a más personas para la incorporación del SIG como herramienta de análisis de sus proyectos de investigación.

Proporcionar una base del sistema SIG sobre el área de influencia de la UNPA, para su utilización libre

Formalizar un grupo de trabajo que asista los requerimientos de los usuarios.

Formar a los estudiantes en este sentido para la continuidad de esta tarea.

Referencias

[Ascough, J. 2002] Multicriteria Spatial Decision Support Systems:Overview, Applications, and Future Reserch Directions. Reserch Hydraulic Engineer and Associates.

[Bocco, G .2000] Ciencia, comunidades indígenas y manejo de recursos naturales – Un caso de investigación participativa. Nterciencia.

[Chuvieco, E. 2002] Teledetección ambiental – La observación de la Tierra desde el espacio, Ariel.

[de San Pedro et al. 2009] de San Pedro Maria Eugenia, Serón Natalia, Cristian Montenegro - Sistema de información geográfica aplicado a turismo y patrimonio histórico y cultural - 11° Workshop de Investigadores de Ciencias de la Computación WICC'09, pp.438-441 ISBN 978-950-605-570-7, Universidad Nacional de San Juan, San Juan – Mayo 2009.

[de San Pedro et al. 2012] de San Pedro M.E., Lasso M., Serón N., Carrizo, A., Montenegro C., Moyano A. - "Sistema de Información Geográfica integrado con un sistema logístico basado en el Problema de Ruteo de Vehículos" - XIV Workshop de Investigadores de Ciencias de la Computación - WICC 2012, Universidad Nacional de Misiones, Posadas – Abril 2012

[Gladstone, B. 1999] Geological Hazards: Their assessment, avoidance and mitigation. E & FN Spon, London

[Martin, F. 1985] Using ageografic information system for forest land mapping and magnament. Photogramemetric Engineering and Remote Sensing.

[Serón et al. 2008] Serón N., Montenegro C., Vidal P., Villagra S., Orozco S., Valdéz J., Díaz F., de San Pedro M. – Implementación de un Sistema de Información Geográfica en Municipios de la Zona Norte de la Provincia de Santa Cruz – 10° Workshop de Investigadores de Ciencias de la Computación WICC'08, pp. 530-533 ISBN 978-950-863-101-5, Universidad Nacional de La Pampa, General Pico, La Pampa – Mayo 2008.

[Serón et al. 2010a] Serón N., Montenegro C., Valdez J., de San Pedro M.E. – Sistema de Información Geográfica Aplicado al Turismo y Patrimonio Histórico y Cultural - II Jornadas de Extensión Universitaria UNPA, Caleta Olivia, 18 de Agosto de 2010.

[Serón et al. 2010b] Serón N., Montenegro C., Valdez J., de San Pedro M.E. – Sistema de Información Geográfica Aplicado al Turismo y Patrimonio Histórico y Cultural - III Jornadas RedVITEC, Mendoza, 25 y 26 de Noviembre de 2010.

[Serón et al. 2011] Serón N., Moyano A., de San Pedro M.E. – Una mirada de la ciudad a través de un Sistema de Información Geográfica - III Jornadas de Extensión Universitaria UNPA, Puerto San Julián, 3 y 4 de Noviembre de 2011.

[Serón et al. 2012a] Serón N., Montenegro C., de San Pedro M.E. – Un Sistema de Información Geográfica que permita Gestionar la Accesibilidad a la Información sobre Caleta Olivia - IV Jornadas de Extensión Universitaria UNPA, Río Turbio, 27 y 28 de septiembre 2012.

[Serón et al. 2012b] N. Serón, C. Montenegro, J. Valdéz, M. de San Pedro - "Sistema de Información Geográfica para Gestionar la Accesibilidad a la Información sobre Caleta Olivia". - XIV Workshop de Investigadores de Ciencias de la Computación - WICC 2012, Universidad Nacional de Misiones, Posadas – Abril 2012

[UNESCO, 1989] Glosario de términos sobre medio ambiente. UNESCO.

Configuración Eficiente de Controladores Fuzzy

Martínez Sergio L.
Facultad de Ingeniería
Universidad Nacional de Jujuy
smartinez@fi.unju.edu.ar

Tarifa Enrique E.
Facultad de Ingeniería
Universidad Nacional de Jujuy
eetarifa@fi.unju.edu.ar

Gutiérrez Jorge J.
Facultad de Ingeniería
Universidad Nacional de Jujuy
jorgejgutierrez@outlook.com

Resumen

Los controladores basados en lógica fuzzy utilizan heurísticas de control derivadas del conocimiento de un experto que no requieren normalmente de un modelo del proceso. El diseño específico de controladores fuzzy sigue un procedimiento general que deja algunas consideraciones bajo el criterio del diseñador. En este trabajo se establece como criterio principal, que un controlador fuzzy genere una curva de transferencia prácticamente lineal, para no interferir con el proceso de control aplicado a plantas generalmente no lineales. Se analizan diferentes tipos de funciones de pertenencia, alcances de las variables fuzzy y el efecto que producen las distribuciones no uniformes en las particiones fuzzy para obtener una curva de transferencia prácticamente lineal, considerando que el proceso de inferencia fuzzy es alineal. Las consideraciones resultantes se aplican para diseñar controladores fuzzy eficientes, que se incorporan en un modelo de control de un mezclador de caudales en línea, produciendo excelentes resultados.

1. Introducción

Los controladores basados en lógica fuzzy[1] utilizan un modo de razonamiento aproximado que se asemeja al procedimiento que efectuaría un operario experto, cuyo conocimiento es sustentado en un conjunto de reglas lingüísticas [1]. La base de un controlador fuzzy es un sistema de inferencia (Figura 1), que contiene los siguientes componentes básicos:

- Módulo de fuzzyficación: En este módulo las variables de control convierten sus valores numéricos instanciados en valores de pertenencia dentro de un intervalo [0, 1] a través de un conjunto de funciones de pertenencia asociadas para cada variable. Esta descripción puede considerarse como una descripción del concepto fuzzyficar.
- Base de reglas: Conjunto de reglas condicionales que establecen las relaciones fuzzy entre las variables controladas y las variables de control, de acuerdo a la dinámica del proceso y según el criterio del experto.
- Motor de inferencia: Estructura lógica que relaciona los hechos iniciales del proceso, a través de las reglas fuzzy, para determinar las precondiciones que asumirán las variables de control, aún en estado fuzzy.
- Módulo de defuzzyficación: Estructura que convierte los resultados fuzzy de las variables internas del sistema de inferencia, generados por el motor de inferencia, en valores numéricos nítidos, para ser aplicados a la acción de control.

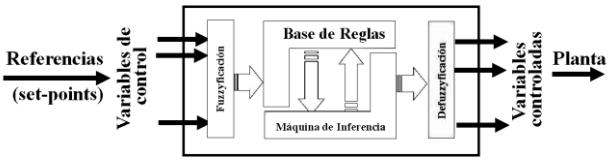


Figura 1. Sistema de inferencia fuzzy configurado como controlador.

De acuerdo a la configuración del controlador fuzzy, puede adoptar el esquema que responde a los tipos básicos del control convencional: Proporcional (P), Integral (I) y Derivativo (D); y las combinaciones más usuales: Proporcional–Integral (PI), Proporcional–Derivativo (PD) y la más general Proporcional–Integral–Derivativo (PID), cuya elección dependerá de la naturaleza del sistema a controlar, sus requerimientos o la interpretación del diseñador del sistema de control [2].

2. Diseño de Controladores Fuzzy

Los sistemas a controlar (planta), de mediana o mayor complejidad, operando a lazo cerrado, son generalmente no lineales. En tal caso, es importante que la función de transferencia del controlador asociado no introduzca en el lazo de realimentación una alinealidad adicional, que puede producir distorsiones inesperadas en el proceso de control. Éste es entonces el criterio principal a tener en cuenta en el diseño de los controladores fuzzy que se plantean.

[1] Si bien se usan traducciones para el término fuzzy (por ejemplo difuso, borroso), se considera que no son totalmente representativas del concepto, por lo que en este trabajo, se utiliza el término original y sus derivaciones adaptadas a este idioma (fuzzyficar, defuzzyficar), que son de uso común en la literatura pertinente en español.

El procedimiento general para diseñar un controlador fuzzy, se puede resumir en los siguientes pasos [3]:

1. Identificar las variables (entradas, estado y salidas) de la planta.
2. Segmentar el alcance de cada variable en un número conveniente de particiones, asignando a cada una de ellas una etiqueta lingüística. Usualmente tales alcances se utilizan normalizados a [0, 1] ó [–1, +1].
3. Asignar una función de pertenencia a cada partición.
4. Establecer las relaciones entre las variables de entrada o de estado fuzzy con las variables de salida fuzzy, para formar la base de reglas.
5. Seleccionar los factores de escala apropiados de las variables de entrada y salida, para normalizar tales variables a los intervalos [0,1] ó [–1, +1].

2.1. Selección de Funciones de Pertenencia

Uno de las elecciones de mayor criticidad en el diseño, es el indicado en el paso 3. En principio no hay un criterio concreto que indique el tipo de función de pertenencia a utilizar. De hecho, en diversas publicaciones relacionadas, los autores aplican diferentes funciones [4], [5], [6], siendo las más usuales de tipo triangular, trapezoidal, gaussiana y sigmoide.

Por la complejidad relacional del sistema de inferencia fuzzy, no es sencillo establecer analíticamente la función de transferencia de sistema. Para mostrar el efecto de cada tipo de función, se utiliza un modelo SISO (Single Input – Single Output) de controlador fuzzy tipo P, con las características que se indican:

- Funciones de pertenencia utilizadas: gaussiana, trapezoidal y triangular.
- Alcance de las variables para entrada y salida [–1, +1].
- Partición uniforme y solapamiento del 50%. Esquema idéntico para entrada y salida.
- Método de implicación por intersección (Mamdani).
- Método de defuzzyficación por centroide.
- En todos los casos se disparan dos reglas como máximo.

Utilizando la interfaz gráfica fuzzy de Matlab®, se puede visualizar la curva de transferencia del controlador fuzzy para las funciones de pertenencía consideradas, como se muestra en las figuras siguientes:

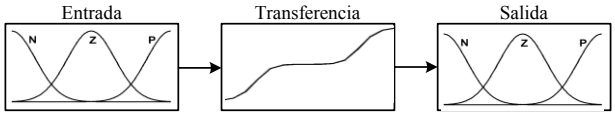


Figura 2. Curva de transferencia para entrada - salida gaussiana.

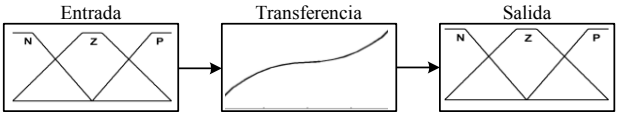


Figura 3. Curva de transferencia para entrada - salida trapezoidal.

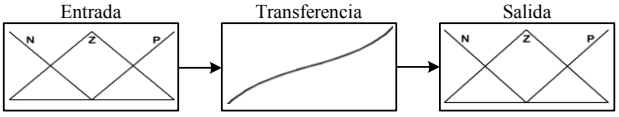


Figura 4. Curva de transferencia para entrada - salida triangular.

Como se observa en las figuras anteriores, las funciones de pertenencia definen curvas de transferencia alineales, situación que resulta más pronunciada para las funciones gaussianas, y con tendencia más lineal para las funciones triangulares.

Como ventaja adicional para las funciones de pertenencia lineales (especialmente triangulares) y considerando que este tipo de sistema siempre trabaja en un esquema iterativo exigente, el esfuerzo computacional para resolver funciones lineales siempre es menor.

2.2. Redefinición del alcance de la salida

El paso 2 del procedimiento de diseño, que sugiere un alcance normalizado para las variables fuzzy de entrada y salida del controlador, genera una situación que es parcialmente responsable de las alinealidades de la curva de transferencia del controlador fuzzy, mostradas en el apartado anterior. Se refiere al alcance definido que segmenta a las funciones de pertenencia, en los extremos del alcance de la variable de salida. Es decir, cuando la entrada toma un valor extremo (–1 ó +1) o cercano a los extremos, y la regla disparada afecta a una de las funciones de pertenencia extrema, su formato no es el mismo.

Considerando las funciones triangulares (originalmente definidas como triángulos isósceles), para la variable de salida, en los extremos se comportan como triángulos rectángulos (Figura 5), de modo que la composición y defuzzificación –en este caso por centroide–, de figuras diferentes, no mantiene linealidad en el entorno. Situación similar ocurre para los otros tipos de funciones de pertenencia consideradas.

El otro inconveniente que se desprende por el cambio de forma de las funciones de pertenencia extremas, obligado por los límites del alcance, es que el valor defuzzyficado de la salida por método del centroide, nunca alcanza los límites originalmente definidos para esta variable (–1 ó +1 en este caso). En el caso de las funciones triangulares, sólo llega al 67% del valor máximo de salida (1/3 de la base de un triángulo rectángulo, desde el lado verti-

cal). Si se cambiara el método de defuzzyficación, por ejemplo a LOM (Large of Maximum) o SOM (Small of Maximum), la situación no se corrige totalmente, porque siempre quedaría un error en uno de los extremos. Esto se puede observar en el esquema gráfico (Figura 5) de operación de la base de reglas sobre la interfaz gráfica fuzzy de Matlab®, con defuzzyficación por centroide.

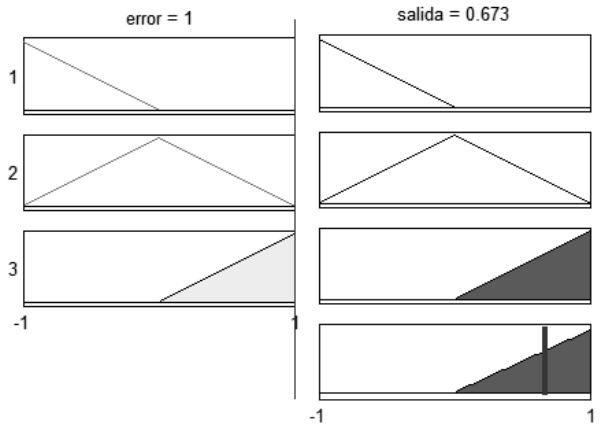


Figura 5. Base de reglas: Alcance restringido de la salida para valor máximo de error de entrada.

Esta situación se puede superar redefiniendo el alcance de la variable de salida, de modo que cubra la totalidad de las funciones de pertenencia extremas, para que el proceso de defuzzyficación por centroide acceda a una figura simétrica completa (Figura 6).

Si bien esta extensión del alcance no forma parte de la definición original del controlador, los segmentos adicionados nunca serían alcanzados durante el trabajo del controlador, debido al límite máximo permitido por la variable de entrada. La Figura 6 muestra la operación de la base de reglas fuzzy en forma gráfica, para la situación de un valor de entrada extremo.

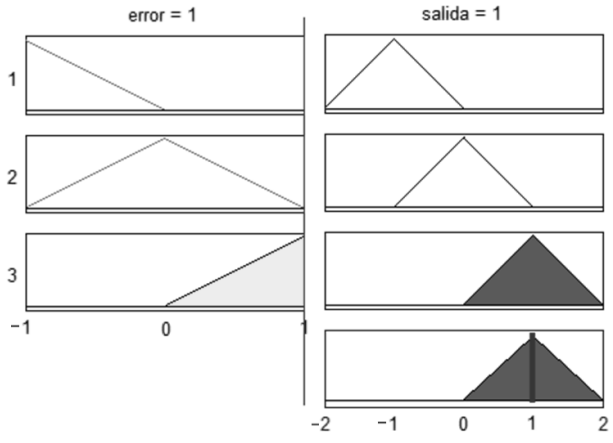


Figura 6. Base de reglas: Alcance completo de la salida para valor máximo de error de entrada.

2.3. Distribución de las particiones

Este proceso, que se encuentra implícito en el paso 2 del procedimiento de diseño detallado anteriormente, se puede considerar como el principal responsable de las alinealidades más significativas de la función de transferencia de un controlador fuzzy.

Si bien, algunos autores proponen en sus trabajos pertinentes, distribuciones poco convencionales de las particiones en las funciones de entrada y salida para los controladores fuzzy [7], [8], [9], [10], la mayoría opta por una distribución regular de tales particiones y sus funciones de pertenencia asociadas.

En las figuras que siguen, se muestra cómo afecta al desarrollo de la curva de transferencia de un controlador fuzzy, diferentes esquemas de distribución de particiones. Para la muestra, se utilizaron funciones triangulares, que se infirieron como la mejor alternativa en las discusiones anteriores.

Para este experimento se utilizó también un controlador fuzzy tipo P, con mayor cantidad de particiones en las variables de entrada y salida, que permiten apreciar mejor el efecto de distribución de las particiones. Los datos son los siguientes:

- Funciones de pertenencia triangulares.
- Alcance de la variable de entrada [–1, +1]. Alcance original de la variable de salida [0, +1], extendida a [–0.25, +1.25].
- Partición uniforme de cinco segmentos para la salida. Particiones no uniformes de cinco segmentos para la entrada.
- En todos los casos, solapamientos del 50% que asegura el disparo de dos reglas fuzzy como máximo.
- Método de implicación por intersección (Mamdani).
- Método de defuzzyficación por centroide.

1º caso: distribución densa en los extremos. Este es un caso bastante desfavorable, como lo muestra la curva de transferencia (Figura 7).

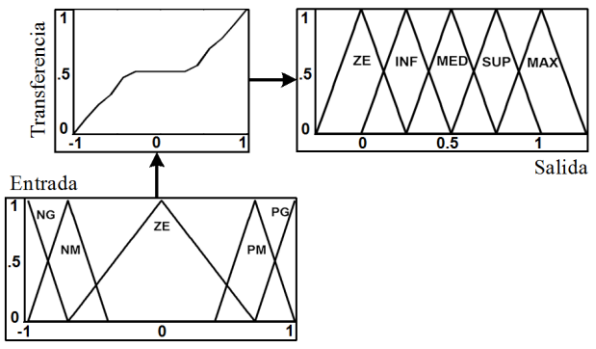


Figura 7. Curva de transferencia para entrada densa en los extremos y salida uniforme extendida.

A pesar que en los extremos el sistema es más sensible, debido a que la curva de transferencia tiene mayor pendiente, en la parte central, el segmento horizontal –de casi el 50% del total– representa un intervalo que se conoce como "zona muerta", donde la salida del controlador fuzzy no responde a las variaciones de las entradas. En definitiva, no es una buena distribución.

2º caso: distribución densa en el centro. Este caso se muestra en la Figura 8 y representa una situación tanto o más desfavorable que la anterior. Se disponen las particiones con segmentos más pequeños al centro del alcance de la variable, de modo que allí se concentran las funciones de pertenencia. Esto produce que la parte central de la curva de transferencia sea aproximadamente lineal, pero la "zona muerta" se traslada ahora hacia los dos extremos, donde el controlador no responderá a las variaciones de la entrada.

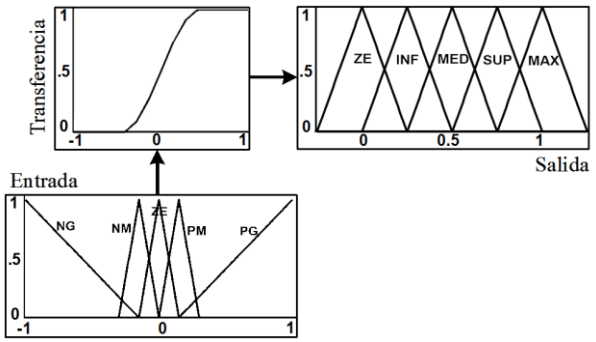


Figura 8. Curva de transferencia para entrada densa en el centro y salida uniforme extendida.

En este caso, ambos segmentos insensibles representan un poco más del 50% del alcance total de la entrada. En definitiva, esta distribución tampoco es buena.

3º caso: distribución simétrica a la entrada y salida. Este caso se muestra gráficamente en la Figura 9.

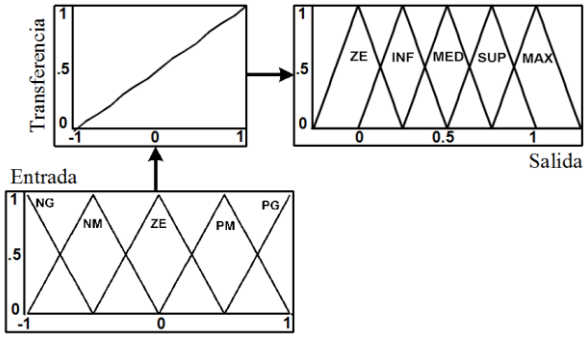


Figura 9. Curva de transferencia para entrada uniforme y salida uniforme extendida.

Se puede observar que la función de transferencia es casi lineal en toda la extensión de las variables. Las pequeñas oscilaciones pueden ser minimizadas aumentando la cantidad de particiones para las variables, especialmente a la salida, donde tiene lugar el proceso de composición y defuzzyficación.

En este caso es fácil deducir que la ganancia del controlador es de 0.5, que se puede modificar redefiniendo la relación de alcance entre las variables de salida y entrada. Para el caso de controladores fuzzy, resulta más conveniente modificar la ganancia con factores externos al controlador y dejar el diseño con alcances normalizados para las variables de entrada y salida (paso 5 del procedimiento de diseño).

Considerando el criterio inicial establecido en este trabajo, para un controlador fuzzy esta configuración resulta la más conveniente, dado que no introduciría alinealidades adicionales a un eventual proceso de control y fundamentalmente no generaría zonas muertas.

3. Implementación de un sistema de control

Para mostrar la actividad de un sistema controlador–planta, operando bajo las condiciones de eficiencia en el diseño de los controladores propuestos en este trabajo, se utiliza como planta un mezclador de caudales en línea que posee dos variables de entrada y dos de salida, como se muestra en el esquema de la Figura 10. Los dispositivos conocidos como mezcladores de flujos o de caudales (flow mixers) mezclan corrientes de fluidos para producir una nueva corriente con propiedades específicas (caudal, presión, temperatura, composición, etc.) [11].

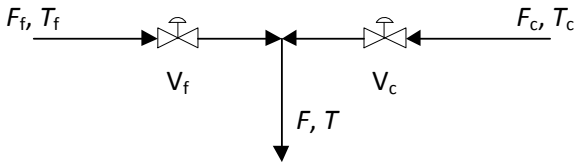


Figura 10. Esquema simplificado de un mezclador de caudales en línea.

El modelo de este mezclador, responde a las ecuaciones siguientes:

$$F = x_f F_f + x_c F_c \qquad 0 \le x_f, x_c \le 1 \qquad (1)$$

$$T = \frac{x_f F_f T_f + x_c F_c T_c}{x_f F_f + x_c F_c} \qquad 0 \le x_f, x_c \le 1 \qquad (2)$$

donde el caudal F y la temperatura T de la corriente de salida dependen de las condiciones de las corrientes de entrada, como ser caudales de las corrientes fría y caliente (F_f y F_c) y sus respectivas temperaturas (T_f y T_c); además de las aperturas x_f y x_c de sus respectivas válvulas.

Se asume que los sensores de las variables controladas están cerca del punto de mezcla, por lo cual el retardo de las mediciones es despreciable. Además, considerando que el objetivo del trabajo es mostrar la linealización de los controladores fuzzy bajo un nuevo enfoque en la configuración de las particiones y funciones de pertenencia, el modelo de la planta se ha simplificado, utilizando válvulas proporcionales puras, sin retardo, que responden a las ecuaciones siguientes:

$$x_f = \frac{F_{sp}\left(T_c - T_{sp}\right)}{F_f\left(T_c - T_f\right)} \quad \text{con} \quad \begin{cases} T_f \le T_{sp} \le T_c \\ 0 \le F_{sp} \le F_{max} \end{cases} \tag{3}$$

$$x_c = \frac{F_{sp}\left(T_{sp} - T_f\right)}{F_c\left(T_c - T_f\right)} \quad \text{con} \quad \begin{cases} T_f \le T_{sp} \le T_c \\ 0 \le F_{sp} \le F_{max} \end{cases} \tag{4}$$

donde (x_f, x_c ∈ [0, 1]) son las variables de apertura de las válvulas fría y caliente respectivamente y (F_{sp}, T_{sp}) corresponde a las variables de referencia del sistema.

El modelo experimental se ha instanciado con los siguientes parámetros:

- Corriente fría: caudal de entrada F_f = 100 l/min, temperatura T_f = 25 ºC.
- Corriente caliente: caudal de entrada F_c = 100 l/min, temperatura T_c = 70 ºC.

Como la planta tiene dos variables de entrada (la regulación de las válvulas x_f y x_c) y dos variables de salida (caudal de la mezcla F y temperatura de la mezcla T), en un esquema de control a lazo cerrado requiere de un controlador tipo MIMO (Multiple Input – Multiple Output).

3.1. Configuración del controlador

La planta propuesta, requiere un controlador MIMO por la cantidad de variables de entrada y salida. A tal efecto, se selecciona el tipo PI (proporcional –integral), bajo un criterio de simplicidad en la generación de la base de reglas fuzzy, como se describe más adelante. Tal controlador se configura con el acoplamiento de cuatro controladores fuzzy SISO también PI, para mantener la configuración general.

El formato de un controlador convencional incremental tipo SISO, en configuración PI, responde a la siguiente ecuación de tiempo discreto:

$$\Delta u(k) = \underbrace{K_p\,\Delta e(k)}_{\substack{\text{segmento} \\ \text{proporcional}}} + \underbrace{K_I e(k)}_{\substack{\text{segmento} \\ \text{integral}}} \tag{5}$$

donde $\Delta u(k)$ es la variación de la acción de control en tiempo discreto k, $e(k)$ es el error de control, la variación del error calculada como $\Delta e(k) = e(k) - e(k–1)$ y (K_p, K_I) son las contantes paramétricas –o ganancias– de esta configuración.

Luego, los cuatro controladores fuzzy SISO requeridos, tienen la configuración mostrada en el esquema de la Figura 11 y se han diseñado siguiendo los pasos indicados en el apartado de diseño; las variables de entrada ($e(k)$, $\Delta e(k)$) y salida ($\Delta u(k)$) se han configurado de acuerdo a las condiciones de eficiencia[2] propuestas en este trabajo:

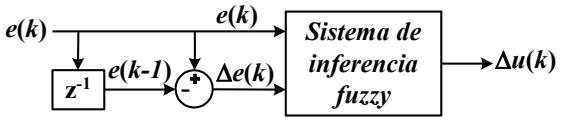


Figura 11. Esquema de controlador fuzzy SISO tipo proporcional – integral.

- Utilizando una cantidad mínima y apropiada, se establecen tres particiones por cada variable, normalizadas a [–1, +1] y asociadas con funciones de pertenencia triangulares, solapamiento del 50 % y extensión del alcance de la variable de salida, como se observa en la Figura 12.

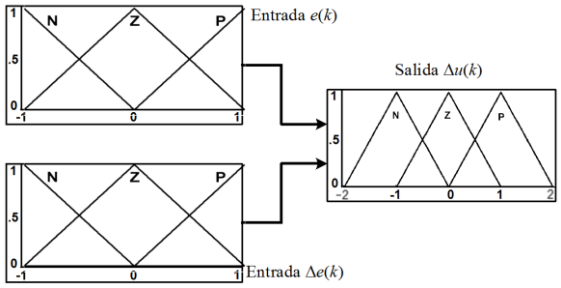


Figura 12. Esquema de controlador fuzzy SISO tipo proporcional – integral.

- Base de reglas configuradas de acuerdo a las tablas de decisión de MacVicar–Whelan[3] [1], con nueve reglas, como se observa en la Figura 13.

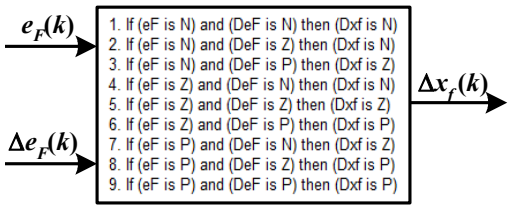


Figura 13. Base de reglas del controlador fuzzy SISO tipo PI para $\Delta x_f(k) = f((e_F(k), \Delta e_F(k))$.

[2] Se considera que el nuevo enfoque propuesto en las condiciones de diseño de los controladores fuzzy permiten obtener un producto más eficiente y con mayor simplicidad respecto del procedimiento tradicional.

[3] Estas tablas se utilizan para el diseño de la base de reglas de los cuatro controladores fuzzy SISO requeridos, bajo un criterio de simplicidad, puesto que no requieren de un conocimiento experto exhaustivo; tan solo conocer si la variable de entrada varía en forma directa con la variable de salida. En caso contrario se aplica la tabla en forma invertida.

- La base de reglas de la figura anterior, corresponde particularmente al controlador fuzzy tipo PI que relaciona el error y variación de error de caudal (($e_F(k)$, $\Delta e_F(k)$) con la variación de la acción de control sobre la válvula de la corriente fría ($\Delta x_f(k)$).

Los cuatro controladores fuzzy SISO, se acoplan, para configurar un controlador fuzzy MIMO, según las ecuaciones (6) y (7):

$$\Delta u_j(k) = \sum_{i=1}^{N} K_{Pi}\,\Delta e_i(k) + K_{Ii}\,e_i(k) \qquad \text{con} \quad j=1,...,M \quad i=1,...,N \tag{6}$$

siendo M la cantidad de variables manipuladas (aperturas de las válvulas fría x_f y caliente x_c), N la cantidad de variables controladas de la planta (caudal F y temperatura T). Luego, la acción de control completa para cada una de las variables manipuladas se puede calcular como:

$$u_j(k) = u_j(k-1) + \Delta u_j(k) \qquad \text{con} \quad j=1,...,M \tag{7}$$

con $u_j(k)$ la acción de control completa en el instante k, $u_j(k–1)$ la acción de control en el instante anterior y $\Delta u_j(k)$ la variación de control calculada con la ecuación (6).

4. Operación del sistema

La implementación del sistema controlador–planta se presenta en la Figura 14, donde se observa el bloque gris de mayor tamaño conteniendo al sistema mezclador de flujos, y los bloques grises más pequeños correspondientes al conjunto de cuatro controladores fuzzy SISO, acoplados, para conformar el controlador fuzzy MIMO de acuerdo a las ecuaciones (6) y (7). El conjunto dispone además de dos bloques (identificados como "control manual") que permite –mediante conmutadores– el ingreso directo del valor de las aperturas de las válvulas x_f y x_c para una eventual operación manual directa.

5. Resultados

Con el modelo experimental de la Figura 14, se pueden desarrollar diferentes pruebas y verificaciones. En este caso, se ejecutaron pruebas clásicas como se indica:

1º prueba – Variación escalón de ambas referencias: Durante una corrida de 100 min, se aplica simultáneamente una variación escalón descendente en t_k=40 min a la referencia de caudal (F_{REF}) y variación escalón ascendente en t_k=60 min a la referencia de temperatura (T_{REF}), para verificar cómo responde el controlador en conjunto (Figura 15).

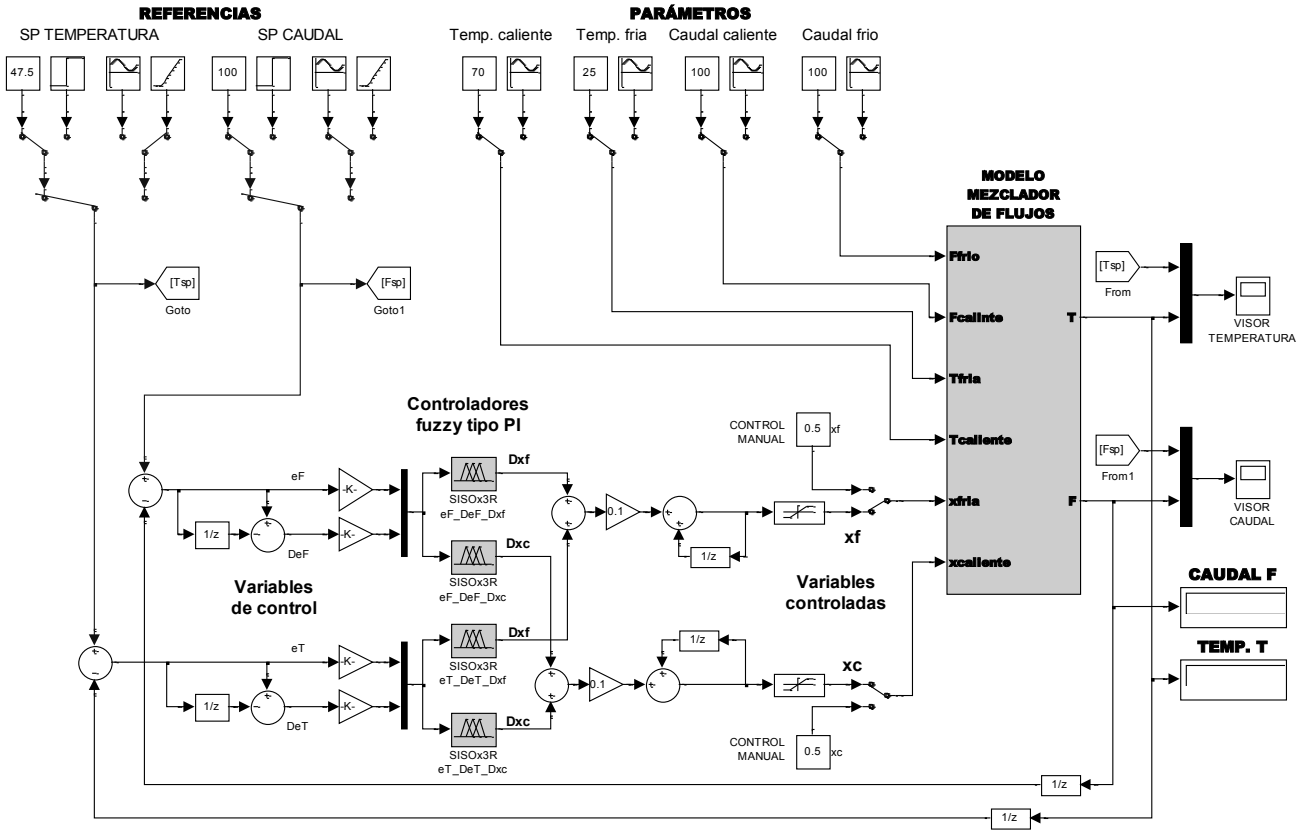


Figura 14. Modelo experimental en Matlab® – Simulink® para el sistema mezclador de caudales con control fuzzy MIMO tipo PI.

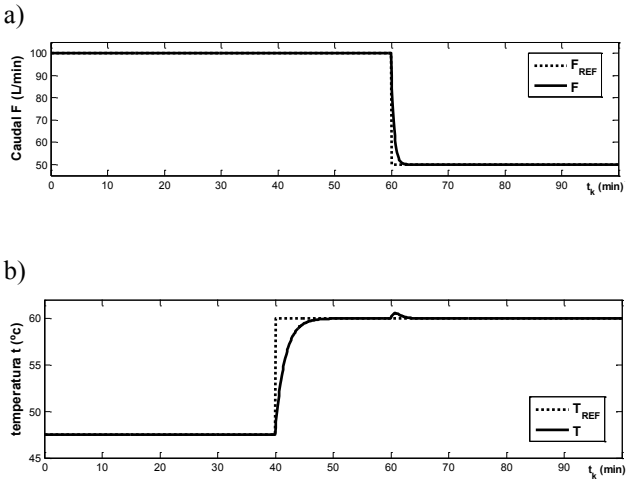


Figura 15. Respuesta a variación escalón para a) la referencia caudal (F_{REF}) y b) para la referencia temperatura (T_{REF}).

Se observa que el controlador responde muy bien a las variaciones de las referencias, alcanzando los valores finales sin offset por acción de la componente integral del controlador. La variación de caudal produce una pequeña perturbación sobre la respuesta de temperatura, debido a la variación del caudal en t_k=60 min, que se disipa rápidamente.

2º prueba – Variación tipo rampa: Este tipo de prueba permite explorar los límites del sistema. Mientras el caudal de referencia (F_{REF}) se mantiene constante en 100 L/min, se produce una variación tipo rampa sobre la referencia de temperatura (T_{REF}), con valor final no limitado (Figura 16).

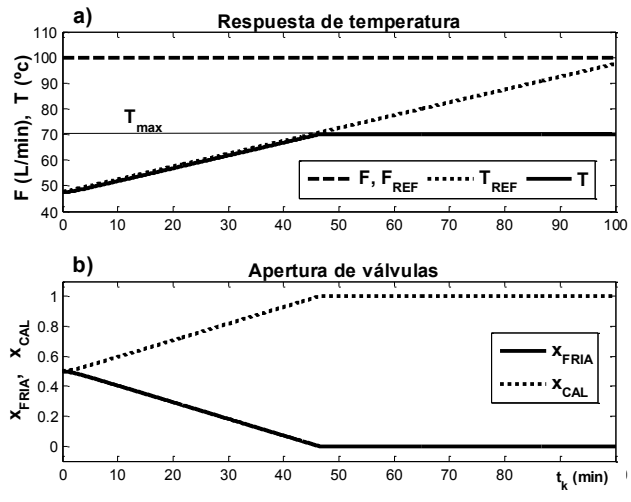


Figura 16. Respuesta a variación tipo rampa para a) la referencia temperatura (T_{REF}). b) Variación de apertura de las válvulas.

En la Figura 16 (a), las gráficas muestran cómo el controlador logra que la planta genere una temperatura de salida (T) que sigue a la rampa de la temperatura de referencia (T_{REF}), en régimen permanente, con un error cuadrático medio ECM = 47.5 (con 466 puntos computados), hasta el instante t_k=46.5 min, donde el sistema se satura al haber alcanzado la temperatura máxima T_{max}=70 ºC. Dicho de otra forma, el controlador genera respuesta con un retardo de 1,4 min, hasta el instante de saturación.

En la Figura 16 (b), se muestra cómo son controladas las válvulas de la corriente fría (x_f) y la corriente caliente (x_c), variando linealmente, para seguir la variación –también lineal– de temperatura de referencia. En el instante t_k=46.5 min, el sistema se satura cuando la válvula x_c alcanza su apertura máxima y la válvula x_f queda totalmente cerrada; el sistema no puede aumentar más su temperatura.

Durante todo el proceso, el caudal de referencia (F_{REF}) y el de salida (F) se mantienen constantes frente a las variaciones de la temperatura (T_{REF}). El controlador fuzzy MIMO tipo PI ha respondido como se esperaba.

6. Conclusiones

En este trabajo se ha presentado un nuevo enfoque para configurar los controladores fuzzy, utilizando como soporte de análisis un controlador fuzzy SISO tipo proporcional. El criterio seguido fue conseguir una función de transferencia lineal, para lo cual se analizaron el tipo de función de pertenencia más conveniente, la distribución y alcances de las particiones, mostrando los efectos que causa sobre la curva de transferencia distribuciones no regulares de tales particiones.

De las consideraciones presentadas se concluye que las funciones de pertenencia triangulares, con una distribución regular y un alcance extendido para las variables de salida, producen los mejores resultados.

Luego, siguiendo el procedimiento de diseño presentado, junto con los criterios de eficiencia determinados, se han diseñado cuatro controladores fuzzy SISO tipo proporcional–integral, convenientemente acoplados para configurar un controlador fuzzy MIMO, que se aplica a un modelo de mezclador de corrientes líquidas en línea, utilizando el entorno de simulación gráfica Simulink® de Matlab®.

Los resultados obtenidos con el modelo experimental controlador–planta, muestran un excelente desempeño de los controladores fuzzy diseñados, en base a las pruebas realizadas.

7. Referencias

[1] Martínez, S., Tarifa, E. y Sánchez Rivero, V., "Derivación de reglas de control MIMO para un controlador fuzzy sin conocimiento experto", *Experiencias Innovadoras en Investigación Aplicada,* ISBN 978-987-26314-4-4, V. 1, 2011.

[2] Chen, J., He, Z.F. and Qi, X., "A new control method for MIMO first order time delay non-square systems", *Journal of Process Control*, V 21(4), pp. 538–546, 2011.

[3] Ross, T. J., *Fuzzy Logic with Engineering Applications*, John Wiley & Sons Ltd., 3º ed., 2010.

[4] Sharma D., *"Designing and Modeling Fuzzy Control Systems"*, International Journal of Computer Applications. V. 16(1), 2011.

[5] Zulfatman and Rahmat, M. F., "Application of self-tuning fuzzy PID controller on industrial hydraulic actuator using system identification approach", *International Journal on Smart Sensing and Intelligent Systems*, V. 2(2), 2009.

[6] Precup, R. and Preitl, S., "Stability and Sensitivy Analysis of Fuzzy Control Systems", *Mechatronics Applications. Acta Polytechnica Hungarica*, V. 3(1), 2006.

[7] Diaz, J. y Muñoz, F., "Comparación entre un Controlador Difuso y un Controlador PID para un Motor Universal", ENGI, V. 1(2), 2013.

[8] Lorandi Medina, A., Hermida Saba, G., Ladrón de Guevara Durán, E. y Hernández Silva, J., "Controladores PID y Controladores Difusos", *Revista de la Ingeniería Industrial*, V. 5(1), 2011.

[9] Montiel, O., Sepúlveda, R., Melin, P., Castillo, O., Porta, M. and Meza, I., "Performance of a Simple Tuned Fuzzy Controller and a PID Controller on a DC Motor", *FOCI 2007 – IEEE,* 2007.

[10] Reznik, L., "Fuzzy Controllers Handbook: How to Design Them, How They Work", Ed. Newnes, 1º ed., 1997.

[11] Martínez, S., Tarifa, E. y Nuñez, A., "Diseño, simulación y comparación de controladores inteligentes aplicados a un proceso industrial", *Investigaciones en Facultades de Ingeniería del NOA*, V. 1 (1), 2013.

Un lenguaje para el modelado y simulación de procesos de morfogénesis de sistemas biológicos

José Oscar Angelini,
Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (FIUNER), carrera de Bioingeniería, Cátedra de Biomateriales y Biocompatibilidad.
Universidad Tecnológica Nacional, Facultad Regional Santa Fé (UTN-FRSF)
Ruta 11 km 10, Oro Verde, Entre Ríos
jangelini@bioingenieria.edu.ar

Ernesto Martínez,
Instituto de Desarrollo y Diseño INGAR (CONICET- UTN)
Avellaneda 6357, Santa Fe, Argentina
ecmarti@santafe-conicet.gob.ar

Carlos H. G. Ramírez,
Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (FIUNER), carrera de Bioingeniería, Cátedra de Biomateriales y Biocompatibilidad.
Ruta 11 km 10, Oro Verde, Entre Ríos
chramirez@bioingenieria.edu.ar

María de los Milagros Gutiérrez
CIDISI Universidad Tecnológica Nacional Facultad Regional Santa Fe (UTN-FRSF)
Lavaise 610, Santa Fe
mmgutier@frsf.utn.edu.ar

Abstract

Los modelos son los principales elementos usados por investigadores para organizar y comunicar ideas, evaluar hipótesis, representar en forma abstracta comportamientos, etc.. Para que un modelo cumpla con su objetivo es necesario contar con un lenguaje de modelado que sea claro, que contenga los conceptos necesarios para poder representar las características más sobresalientes de los modelos del dominio. En relación al modelado y simulación del proceso de morfogénesis de sistemas biológicos, se presenta la necesidad de contar con un lenguaje adecuado orientado a dicho dominio. En este trabajo se muestran los avances realizados en la definición de un lenguaje para el modelado y la simulación de este tipo de sistemas, de manera de ocultar detalles de implementación y ejecución de una simulación.

1. Introducción

Los Modelos se utilizan en los sistemas biológicos para organizar y comunicar ideas, evaluar hipótesis, representar comportamiento, etc. La forma de esos modelos pueden ser muy variadas y los elementos usados para construirlo deben ser los adecuados de manera de poder representar las características del dominio. Las representaciones tradicionales, como por ejemplo de redes de reacciones biomoleculares han utilizado narraciones en lenguaje natural complementadas con diagramas en bloques y flechas, que si bien son útiles para describir las hipótesis sobre los componentes de un sistema y sus interacciones, resultan inadecuadas para la comprensión de la dinámica de estos sistemas, altamente no lineales y complejos. En cambio, los modelos computacionales que sustituyen estos diagramas estáticos se utilizan como integradores del conocimiento, y sirven como la pieza central del ciclo de modelización y simulación científica. Al describir sistemáticamente cómo las entidades y procesos biológicos se interrelacionan y se desarrollan, y por la adopción de normas de cómo se definen, representan, manipulan e interpretan, estos modelos permiten una comparación significativa entre las consecuencias de las ideas básicas y los hechos empíricos.

Si bien la experimentación ha sido clave en el desarrollo del conocimiento biológico, el avance solo se puede lograr mediante la elaboración de teorías, hipótesis que puedan ser luego, comprobadas experimentalmente.

La biología del desarrollo es una rama de la biología que estudia el proceso biológico que causa que un organismo desarrolle una forma determinada. Este proceso se denomina morfogénesis y es utilizado tanto en medicina regenerativa como en ingeniería de tejidos. Actualmente, existen dos paradigmas que enfocan el estudio de estos sistemas de manera muy diferente: el reduccionismo y el emergentismo. El reduccionismo estudia un fenómeno complejo a través del análisis, es decir la división en partes o elementos componentes.

Dentro de los métodos emergentistas están los modelos discretos en los que distintos autores tratan de explicar el fenómeno considerando la conducta y

dinámica de las células. Entre ellos, el más usado es el Modelo de Glazier, Graner y Hogewek (GGH) [4,5] conocido también como Modelo Celular de Potts. Este modelo es una variante del método de Montecarlo basado en la Hipótesis de Adhesión Diferencial (HAD) creada por Steingber en 1963 [12].

Recientemente, ha surgido el concepto de agentes para modelar sistemas biológicos, esta metodología denominada ABM (Agent-based Modeling) simula la interacción de entidades autónomas (agentes), entre sí y con su ambiente, para predecir patrones emergentes tanto en su distribución espacial, como en su distribución temporal.

La metodología ABM se ha aplicado a sistemas biológicos como un medio para modelar: (i) la estructura y conducta individuales de diferentes entidades de un sistema biológico (heterogeneidad); (ii) la estructura del entorno y su dinámica en escalas temporales y espaciales diferentes; (iii) Las interacciones locales entre los agentes biológicos y su entorno [8]. Con esta metodología se pueden modelar los mecanismos que influencian tanto la conducta de las células como la dinámica del ambiente extracelular [8].

Varios autores han aplicado la metodología ABM en los últimos años: Grant y cols. [6] estudiaron la morfogénesis del tejido epitelial in vitro. Schaller y col. (2007) [11] investigaron los procesos que controlan la estabilidad de la epidermis. Sun y cols. [14] estudiaron la formación de colonias celulares de las células de la epidermis o queratinocitos. Adra y cols [1] desarrollaron un modelo tridimensional de la epidermis humana para estudiar la influencia del factor (palabra equivalente a sustancia) de crecimiento TGF-Beta1 (Transforming Growth Factor Beta 1) durante el proceso de cicatrización de heridas. A excepción de este último, los autores mencionados desarrollaron modelos considerando sólo aspectos geométricos, como el número de vecinos cercanos o la distancia a un conjunto células madres (nicho) para modelar los procesos por los cuales las células se reproducen o se diferencian a otros tipos celulares. Por otra parte Adra y cols. [1] se restringieron a simular experimentos in vivo, y no incluyeron experimentos in vitro. Se debe destacar aquí que tener en cuenta la influencia de los factores de señalización y crecimiento es clave para entender adecuadamente la morfogénesis y el desarrollo de cualquier tejido generado tanto in vitro como in vivo.

El presente trabajo trata de contribuir en esta área, con la creación de un lenguaje que permita modelar los conceptos involucrados y su simulación utilizando agentes para la visualización y predicción de morfogénesis de tejidos biológicos, teniendo en cuenta también los mencionados factores de señalización y crecimiento.

El trabajo se ha estructurado de la siguiente manera: En la sección 2 se dan los fundamentos del método de modelado empleado y los conceptos biológicos utilizados en la elaboración del lenguaje. En la sección 3 se explica el metamodelo desarrollado, la definición de los agentes, su ambiente y el entorno de simulación. En la sección 4 se ejemplifica como caso de estudio, la morfogénesis de una valva cardíaca, creada en vitro y se muestra su implementación en Netlogo, junto con los resultados obtenidos. Finalmente, en la sección 5, se detallan las conclusiones a las que se arriba.

2. Fundamentos

Se considera agente a toda entidad autónoma que percibe su entorno, mediante sensores, y actúa sobre el mismo mediante efectores [10]. La figura 1 muestra las principales propiedades de un agente. Su comportamiento es no lineal, y en consecuencia no se puede utilizar el principio de superposición para predecir su respuesta. Tienen la capacidad de adaptarse a su entorno o de evolucionar en respuesta a los estímulos que percibe del ambiente. La filosofía subyacente del ABM es que los agentes individuales responden a un conjunto de reglas mediante las cuales toman decisiones que determinan su comportamiento particular [9]. En la interacción de los diferentes comportamientos, los agentes definen el estado del sistema en un instante dado, y con el tiempo, producen la aparición (emergencia) de patrones observables a más alto nivel. Con la inclusión de varios tipos de agentes es posible la exploración de comportamientos más complejos [13].



Figura 1. Propiedades de los agentes.

Los sistemas biológicos se pueden considerar constituidos por agentes. Estos agentes pueden ser

células o microfibrillas biológicas. Las células pueden subdividirse creando nuevas células o pueden diferenciarse a otras células más especializadas y también pueden crear microfibrillas que pasan a integrar el ambiente intercelular como por ejemplo microfibrillas de colágeno que se unen unas a otras configurando las fibras resistentes de los tejidos blandos. En las etapas tempranas del desarrollo humano son las células las que crean los tejidos y órganos biológicos y luego de crearlos se acomodan en ellos para poder repararlos de los daños producidos por la actividad diaria, es decir actúan como "ingenieros de mantenimiento". En el organismo humano, hay células indiferenciadas llamadas células madres que tienen la capacidad de convertirse en otras células altamente especializadas que cumplen funciones muy específicas como, por ejemplo, las neuronas, los queratinocitos, los glóbulos rojos o los miocardiocitos (células musculares del corazón). A este tipo de células especializadas se las denomina *diferenciadas*. Para sufrir divisiones sucesivas (mitosis) o crear células diferenciadas, las células madres necesitan ser estimuladas mediante sustancias químicas llamadas *factores de señalización y crecimiento*.

3. Metamodelo

3.1. Definición del lenguaje

Se definieron una serie de conceptos para modelar el dominio bajo estudio (figura 2).

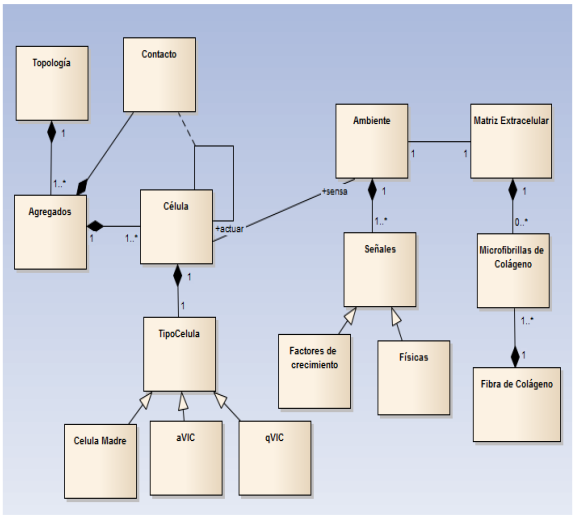


Figura 2. Diagrama del metamodelo realizado.

Hay dos conceptos principales que forman el eje del metamodelo: Células y Ambiente que representa el entorno donde la celula vive. Las células son la unidad funcional de la vida; sensan el medio ambiente y actúan en él. La célula tiene un ciclo de vida: nacer, se reproduce y muere. De acuerdo con la configuración del ambiente, la célula puede ser de diferentes tipos tales como células madres, células valvulares intersticiales activas o células intersticiales de válvula inactivas, entre otros. Durante su vida, la célula puede cambiar de tipo, por ejemplo, una célula nace como una célula madre, pero luego crece y se puede transformar en una célula intersticial de la válvula cardíaca. Este fenómeno se conoce como diferenciación. Cada tipo de células tiene un comportamiento y propiedades diferentes. En el metamodelo, se ha representado el concepto principal de *célula*, y el concepto *TipoCélula* asociado a él. Esta representación permite modelar el cambio de comportamiento específico de manera tal que la célula mantenga sus características, ya que la misma nunca cambiar su esencia pero si su comportamiento, este cambio esta representado por el concepto *tipoCelula*. Durante su ciclo de vida de la célula puede cambiar su tipo y seguir siendo la misma célula. Para ello se delega en *TipoCélula*, la representación de todos los comportamientos.

El ambiente representa el medio donde se desarrolla la visa social de las células. Para modelarlo se definió el concepto de *Ambiente*. La célula interactúa con el *Ambiente* a través de dos relaciones: (i) Percibir: recibe un conjunto de valores del ambiente y (ii) actúa: a través de este vínculo la célula le informa que acción ha decidido efectuar. El concepto *Ambiente* está relacionado con el concepto *MatrizExtracelular*, que representa el soporte físico que permite la vida de las células, sirve como medio sobre el que hay señales. En este concepto *Matriz Extracelular* (MEC), se representa la estructura de soporte que permite la deposición de moléculas. La MEC está compuesta por microfibrillas de colágeno que se unen y forman fibras de colágeno capaces de modificar la rigidez de la MEC. Sobre el *Ambiente* actúan señales. El concepto *Señales* representa no solo a *Factores de crecimiento* sino también otros tipos de señales como por ejemplo *fuerzas físicas* que actúan como perturbaciones externas que modifican el estado de la MEC.

La *Célula* tiene la capacidad de comunicación con otras células mediante señales depositadas en el ambiente o por contacto directo. Las señales depositadas en el ambiente son mayoritariamente los factores de señalización y crecimiento. Cuando las células, tienen contacto directo pueden crear uniones a través de las cuales se comunican, por eso se definió el concepto *Contactos*. Estos contactos son uniones entre las células las cuales forman *Agregados*, que son conjuntos de células con una cierta densidad y orden en el espacio. El conjunto de agregados conforma una *Topología*, o patrón característico de cada tejido biológico.

3.2. Caso de estudio: Morfogénesis en vitro de una válvula cardíaca

Con el objetivo de demostrar la aplicabilidad de los conceptos vertidos en el metamodelo en el modelado y simulación del proceso de morfogénesis biológica, tomamos como caso de estudio el proceso de creación por ingeniería de tejidos de una válvula cardíaca en vitro. Este proceso parte depositando células madres mesenquimales en un soporte temporal, junto con el agregado de factores de crecimiento y señalización. Este modelo sirve para visualizar cómo se desarrolla este proceso ante la variación de la concentración de factores de crecimiento. En la bibliografía, existen modelos que consideran la dinámica de un nicho de células madres en vivo [3], pero se basan en aspectos geométricos y no consideran las señales que intercambian las células, en particular los factores de crecimiento. El uso de los estos es un aspecto clave en ingeniería de tejidos, ya que es la forma en que las células determinan que tejido crear.

3.3.1 Descripción del sistema

Una válvula cardíaca está compuesta de dos o tres membranas resistentes llamadas valvas. La resistencia de estas valvas está determinada por el número y densidad de las fibras colágenas que a su vez están formadas por el agregado de microfibrillas colágenas.

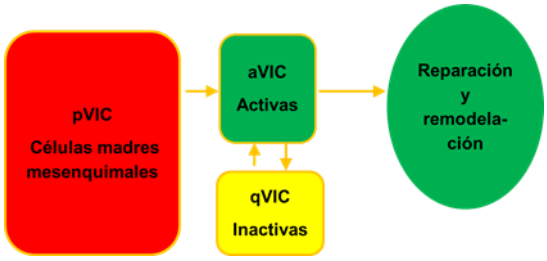


Figura 3: Relaciones entre los distintos tipos celulares del sistema bajo estudio

En este trabajo las principales células que componen una valva cardíaca se denominan células VICs (Valvular Intersticial Cells). Estas células se originan de un tipo celular indiferenciado que se denomina pVIC se obtienen normalmente de la médula ósea. Esta célula en estado pVIC puede dividirse generando más células pVIC indiferenciados o bien diferenciarse a otro tipo celular designado como aVIC (en biología se dice que ha cambiado el fenotipo) y que se parece a un tipo celular denominado miofibroblasto. En estado aVIC la célula tiene la propiedad de poder generar otra célula distinta como ser las moléculas de colágeno (figura 3). Además la célula aVIC puede desactivarse al fenotipo qVIC muy similar a un fibrocito y dejar de producir microfibrillas colágenas [7]. Las fibrillas Colágeno pueden unirse con otras fibrillas configurando fibras de colágeno más gruesas y resistentes que son las que confieren resistencia a las valvas.

El ambiente en que se desenvuelven las células y las fibrillas de colágeno es lo que se denomina espacio intercelular. En su ambiente natural, el espacio intercelular se denomina matriz extracelular y lo construyen las mismas células, pero en la experimentación biológica in vitro se crean ambientes artificiales que permiten el cultivo de las células o pueden servir como soporte temporal mientras las células recrean su propia matriz extracelular. Las células secretan a ese espacio intercelular, natural o artificial, sustancias como factores de señalización y crecimiento o microfibrillas de colágeno. Estas microfibrillas se autoensamblan y forman fibras más resistentes cuando reciben señales mecánicas del medio. De manera que el ambiente en que se desenvuelven influencia sobre su comportamiento, el cual estará sujeto a variables como la concentración de factores de señalización y crecimiento o cargas direccionales de tensiones mecánicas

En este ejemplo, se considerará la utilización de células madres mesenquimales (MSCs o pVICs) que se encuentran en la médula ósea. Para que una pVIC se reproduzca y diferencie a una célula aVIC, necesita que en el ambiente se encuentre el factor TGF-beta1 (Transforming Growth Factor Beta 1). Otro factor involucrado en el sistema es el FGF-2 (Factor de crecimiento fibroblástico 2) que se encarga de indicarle a la célula madre pVIC que se mantenga indiferenciada. Los factores de crecimiento tienen la propiedad de activar a las células sólo a partir de alcanzar una concentración umbral, por debajo de la cual su presencia no tiene efectos. Los valores de estos umbrales son seleccionados al inicio de la simulación.

3.3.2. Modelo de Implementación usando Agentes

Con el fin de poder ejecutar los modelos generados con el lenguaje presentado de manera de interpretar su comportamiento dinámico, los conceptos del metamodelo se implementaron usando agentes de software que puedan ser usados en un sistema de simulación. Para mapear los conceptos del metamodelo que interpretan el dominio del experto al de dominio de un modelo basado en agentes, es necesario identificar qué conceptos pueden ser representados como entidades autónomas usando el concepto de agente y de cómo representar su entorno.

El concepto Célula se implementó usando un agente que se llamó *Agente VIC*. Este agente tiene un contador, cuyo valor determina su posición en el ciclo celular. Una célula necesita tiempo para reproducirse y diferenciarse. Para especificar su conducta, se considera en su estado el

tipo de célula correspondiente. Además tiene una variable que indica su estado y está vinculada con una implementación del concepto *TipoCelula*. Se modelaron los 3 tipos intervinientes, Célula Madre Mesenquimal (pVIC), Célula Valvular Intersticial Activada (aVIC) y célula valvular inactiva (qVIC). De esta manera se flexibiliza la posibilidad de cambiar de un tipo a otro cambiando su comportamiento sin perder su identidad como célula. La clase TipoCelula, es una clase abstracta, de la que se derivan, por herencia clases que implementan las características y conductas de cada tipo célular considerada. En consecuencia, el proceso de diferenciación implica un cambio de estado que conlleva la instanciación de TipoCelula específico. La figura 4 muestra el cambio de estado posible de una célula para el caso de los tipos considerados.

Hay moléculas que actúan de forma autónoma. Ese es el caso de las microfibrillas de colágeno. Estas son consideradas como *agentes Co* que pueden conformar un agente de compuesto *FibraColageno* que implementa el comportamiento de la fibra de colágeno.

Si bien los factores de crecimiento influyen en la dinámica del sistema no se consideran como agentes, ya que el metamodelo no considera su mecanismo de acción, por ser un proceso intracelular, además su comportamiento es fácilmente modelado por ecuaciones de difusión. Tienen un valor umbral para activar las acciones celulares.

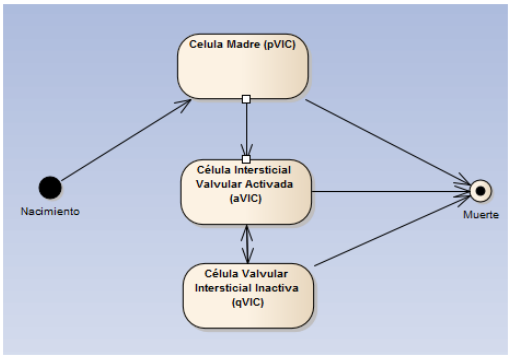


Figura 4: Diagrama de estado del *agente VIC*.

El agente VIC tiene como estado inicial pVIC, es decir de célula madre mesenquimal o proliferativo, durante el cual tiene como principal función crear nuevas células. De su estado inicial pasa a estado aVIC donde crea agentes Co. Puede pasar bidireccionalmente, al estado qVIC.

El agente VIC tiene un contador interno que se incrementa con cada ciclo de simulación, que indica en que parte de su ciclo de vida está, hace las veces de reloj biológico y tiene 3 tipos asociados (figura 4) en los que se definen la conducta de los tipos celulares involucrados. Consideramos que el ciclo de reproducción se inicia a partir de *cReproduccion* Ciclos de simulación, donde *cReproduccion* es un número entero que se mantiene constante y ha sido fijado en 30 ciclos, También se considera un número de ciclos a partir del cual puede cambiar el tipo de agente. Se define *Dif-qVic*, como el número de ciclos a partir del cual el tipo aVIC puede cambiar a qVIC, como una constante igual a 25. Se denomina *cDif-Avic* a la constante para el cambio de pVIC a aVIC que se fija en 20 ciclos.

El estado *pVIC* define la conducta del agente para simular la de las células madres (tabla 1), se lo denomina *pVIC* por ser un estado principalmente proliferativo, sin embargo puede pasar al estado *aVIC*, según el valor del contador interno y las reglas que definen su conducta.

Las reglas que definen la conducta de la célula *pVIC* son las siguientes:

Tabla 1. Reglas que definen el comportamiento del agente celular pVIC.

Nro	Variable Sensada	Regla	Acción
P1	Factores de crecimiento TGF-Beta1	Si el Número de TGF-beta1 es mayor a su umbral y su contador interno es mayor a cReproduccion Ciclos	Crea nuevas células madres (tipo pVIC)
P2	Factores de crecimiento FGF-2	Si el Número es mayor al umbral de FGF-2 > cReproduccion Ciclos	Crea nuevas células madres (tipo pVIC)
P3	Factores de crecimiento TGF-Beta1 y FGF-2	Si su contador interno está por encima de cDif-Avic ciclos y FGF-2 menor que su umbral y TGF-beta1 es mayor que su umbral	Se diferencia a aVIC (cambia a tipo aVIC)
P4	Número de Células vecinas	Si N es > 8	No se reproduce
P5	Número de Células vecinas y Factores de crecimiento FGF-2	Si N <= 8 y FGF-2 < su umbral	Secreta FGF-2

Las reglas P1 a P5 definen la conducta del agente en estado pVIC, consideran el efecto de los factores de crecimiento y del espacio en que se halla. Si no tiene espacio no puede reproducirse (regla P4). Si recibe una cantidad de TGF-Beta 1 por encima de su umbral y si su

contador interno está por encima de cReproduccion ciclos, parámetro prefijado, se reproduce. Esta regla interpreta el hecho que una célula madre no puede reproducirse inmediatamente, necesita llegar a la madurez, y lo hará si recibe como señal el factor TGF-beta1 o FGF-2. Si el FGF-2 esta por encima de su umbral se mantiene su tipo, reciba o no TGF-beta1. Si recibe solo TGF-beta1 y su contador interno está por encima de cDif-avic ciclos, entonces cambia de tipo a qVIC.

En el estado aVIC, el agente aplica las reglas mostradas en la tabla 2.

Tabla 2. Reglas que definen el comportamiento del agente celular aVIC.

Nro	V. sensada	Regla de decisión	Acción
A1	Nro de Vics a una distancia D ; Deformación de la MEC (ξ) ; Ciclo de simulación	Si tiene >= que 8 VICs vecinas y la Deformación es media y su contador interno está por encima de Dif-qVic ciclos	Se diferencia a qVIC (Cambia su tipo a qVic)
A2	Ciclos de simulación	En cada ciclo de simulación	Crea microfibrillas de colágeno con dirección aleatoria
A3	Deformación de la MEC (ξ) y Numero de Factores de Crecimiento TGF-beta 1	Si ξ > 0 y TFG-beta1 > su umbral (sin importar otras reglas)	Mantiene su su tipo

El tipo aVIC representa la conducta de los miofibroblastos. Estas células son responsables de la creación de microfibras que luego forman las fibras. La conducta de la microfibrilla es interpretada por el agente fibrilar *Co*. En consecuencia el Agente *VIC* creará en cada ciclo de simulación un agente *Co*. El resto de las reglas consideran los cambios de tipo del agente celular que se mantiene activo si está ante la presencia de TGF-beta1 o bien se diferencia a qVIC si entra en contacto con células vecinas (desactivación por contacto).

La célula qVIC emite prolongaciones y cuando entra en contacto con otras del mismo tipo se une a ellas. Es muy difícil saber con precisión qué información comparten esas células, una hipótesis es que entre otras cosas comparten el valor de deformación del ambiente. La célula no sensa directamente la presión ya que no tiene sensores de tensión, lo hace indirectamente a través de la deformación, para lo cual si tiene sensores. La deformación es clasificada como baja; media o alta, si es alta la célula se (re)activa.

Las reglas para el estado qVIC son las mostradas en la tabla 3.

Tabla 3. Reglas que definen el comportamiento del agente celular qVIC.

Nro	V. sensada	Regla de decisión	Acción
1	Vics a una distancia D	Si están a una distancia D	Emitir una prolongación (Se conecta con las vecinas)
2	Deformación promedio del MEC (ξ) Factores de crecimiento TGF-Beta1	Regla 2.a Si ξ es alta Regla 2.b Si el en Numero de TGF-beta1 > su umbral y ξ promedio es >= baja	**Cambia su tipo a aVIC**

El Agente Co, tiene una función simple pero determinante en el comportamiento del sistema. Tiene dos posibles estados: direccionado y no direccionado. Cuando el agente VIC lo crea, el agente Co, se ubica en la posición del mismo con orientación aleatoria y estado no direccionado. Luego sensa la dirección de la tensión aplicada al ambiente y se orienta en su dirección cambiando su estado a direccionado. Si encuentra otro agente Co a una distancia cercana, se une con él por las puntas formando segmentos más largos o bien lateralmente formando segmentos más gruesos.

Definición del ambiente de simulación

En el metamodelo definimos que la *MatrizExtracelular* se vincula con el *ambiente*, En la misma se desarrolla la vida social de las células y es a partir de la misma donde se deben obtener las percepciones y aplicar las acciones indicadas por las células. Por ello partiremos de considerar cómo de implemente este concepto. La *MatrizExtracelular* es modelada como una superficie cuadrada de 100 x 100 micrómetros, al que se lo divide en sectores de 5x5 micrómetros. Simula el comportamiento de un sólido anisotrópico (cuando se le aplica una tensión se deforma de manera no lineal y con valores diferentes para distintas direcciones). Pero además el ambiente debe operar cíclicamente actualizando su estado, para ello se utiliza el el ciclo de simulación de la figura 5.

```
por cada agente en agentes hacer
        Percep [agente] ← Obt-Percep(agente, estado)
fin
por cada agente en agentes hacer
        Acción[agente]← Programa[agente](Percep[agente])
fin
estado ← fn-actualización(acciones, agentes, estados)
ponderaciones ← fn-perfomance(ponderaciones, agentes,
                              estado)
```

Figura 5: Ciclo de simulación.

Al inicio de cada ciclo, el ambiente suministra la información que perciben los agentes.

Las percepciones definidas para este caso de estudio son las siguientes:

a) VICs vecinas:

El ambiente determina, utilizando el método de Manhatan, cuáles son los agentes del mismo tipo que se encuentran a una distancia igual o menor a D. Este conjunto se incluye en un vector que se envía al agente del tipo qVIC. La distancia D es un número entero que representa el número de sectores.

b) Número de VICs vecinas:

El medio debe contar el número de células VICs que están a una distancia menor o igual a un número entero D.

c) Células interconectadas:

El ambiente guarda información de las células interconectadas en un vector bidimencional con dos componentes por cada fila. En cada componente se ubica el ID (número identificatorio de cada célula) de la célula que pidió la conexión y el ID de la célula a la cual se destinó esa conexión.

d) Módulo de elasticidad del sector:

En cada sector el módulo de elasticidad se estima considerando el valor inicial más la rigidez aportada por las microfibrillas de colágeno. Es un número real. La rigidez aportada por las fibrillas de Colágeno se considera solamente cuando el agente está orientado en dirección de la tensión y se calcula con la siguiente ecuación (1).

$$Ec = Eo + nCo*Eco \quad (1)$$

Donde:

Ec es el módulo de elasticidad del sector considerado en Giga Pascales (GPa),

Eo es el módulo de elasticidad inicial del ambiente también en GPa,

Eco es el módulo de elasticidad de la microfibrilla de colágeno, que se considera un valor de 1GPa.

e) Deformación promedio:

La deformación promedio recibida por cada agente VIC, se calcula promediando la deformación del sector en el que se ubica con las deformaciones de los sectores en los que haya células interconectadas. La deformación de cada celda se estima considerando la tensión y dividiéndola por su módulo de elasticidad, ese valor es promediado con las deformaciones de los sectores en que se encuentran los otros agentes interconectados. Es un número real adimensional

f) Cantidad de factores de crecimiento (GF):

Para cada agente VIC ubicado en una posición i,j dentro del ambiente se cuenta el número de moléculas de FGF-2 y de TGF-beta1 (Números enteros).

g) Dirección de la tensión:

El agente Co se orienta en dirección de la tensión. El ambiente de simulación está sometido a tensión, la que puede ser descompuesta en dos componentes ubicados sobre los ejes coordenados. La dirección está dada por el ángulo entre las componentes, que se calcula con la función arco tangente del cociente entre la tensión Y y la tensión en X. Este ángulo, es un número real expresado en grados.

h) Ciclo celular:

El ambiente le indica a cada agente cuando se ha iniciado un nuevo ciclo de simulación para que lo almacenen en sus contadores internos.

Dentro de la actualización de estado del ambiente se considera la difusión de factores de crecimiento, cuya cantidad se calcula utilizando las funciones de difusión que el entorno se simulación provee.

4. Implementacion en NetLogo

El caso de estudio fue implementado en Netlogo. Una interface del mismo se observa en la figura 6

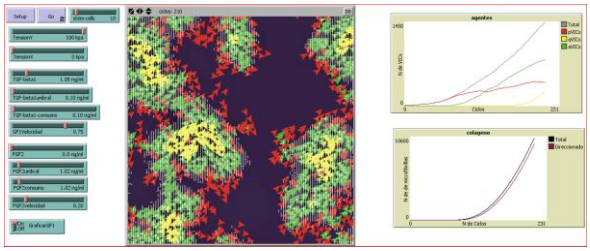


Figura 6: snapshot de netlogo

La interface cuenta con 2 botones, Setup y Go. El primero inicializa la simulación y el segundo la corre. Además tiene una serie de controles deslizantes que permiten fijar la concentración de factores de crecimiento, su umbral de acción y su consumo. Los factores de crecimiento FGF-2 y TGF-beta tienen sus controles respectivos. Se pueden fijar el número de células madres que se depositan sobre el ambiente aleatoriamente. Hay 2 gráficas sobre la derecha de la imagen, una representa el número de agentes en función de los ciclos de simulación. Con trazo negro se representa el total de agentes, con trazo rojo los agentes de tipo pVIC, en verde los agentes del tipo aVIC y en amarillo el tipo qVIC.

La otra grafica representa el número de agentes Co en función de los ciclos se simulación. Con trazo azul el número total de agentes Co, con trazo violeta el número de agentes en estado direccionado.

El ambiente de simulación se ubican los agentes VIC con forma triangular y su color indica el tipo. Los agentes Co tienen forma de barra su color siempre es blanco. Sobre el ambiente se representa la concentración de 1 factor de crecimiento de acuerdo a una escala de colores. Es posible optar, mediante un control que factor de crecimiento se representa.

4.1. Escenarios de prueba y resultados de las simulaciones

Se han considerado los siguientes escenarios:

- Escenario 1: Células madres sembradas sin factores de crecimiento ni tensiones
- Escenario 2: Células madres sembradas con TFG-Betal, sin tensión
- Escenario 3: Células madres sembradas con TFG-Beta1, con tensión en dirección Y

A continuación se describen cada uno de estos escenarios y se muestran sus resultados.

4.1.1. Escenario 1: Células madres sembradas sin factores de crecimiento ni tensiones

Para este caso de estudio, se sembraron 10 células madres. Las condiciones iniciales fueron: FGF-2 = 0; FGF-2 umbral = 1,02 ng/ml; TGF-beta1 = 0. Resultados: (figura 8) grafica superior izquierda, 100 ciclos de simulación, se observan que las células secretan FGF-2 mediante el color de los sectores vecinos. Cuando la concentración de FGF-2 está por encima de su umbral las células se reproduce. Gráfica superior derecha, 1200 ciclos de, las células se reproducen y forman colonias. Figuras al medio izquierda, 1500 ciclos de simulación las colonias crecen más aceleradamente. Figura al medio derecha, 2500 ciclos de simulación las células han poblado todo el ambiente. La gráfica inferior muestra el número de agentes (eje Y) en función del número de ciclos, eje x. Se puede observar que la curva llega a una meseta máxima, por encima de la cual no hay nuevos incrementos en la población celular.

En este caso, se cumple lo indicado por Bianco y cols [2] que indica que las células madres, se pueden reproducir para aumentar su número y forman colonias antes de llegar al estado de confluencia (el estado de confluencia se da cuando no se produce un aumento mayor de la población celular.

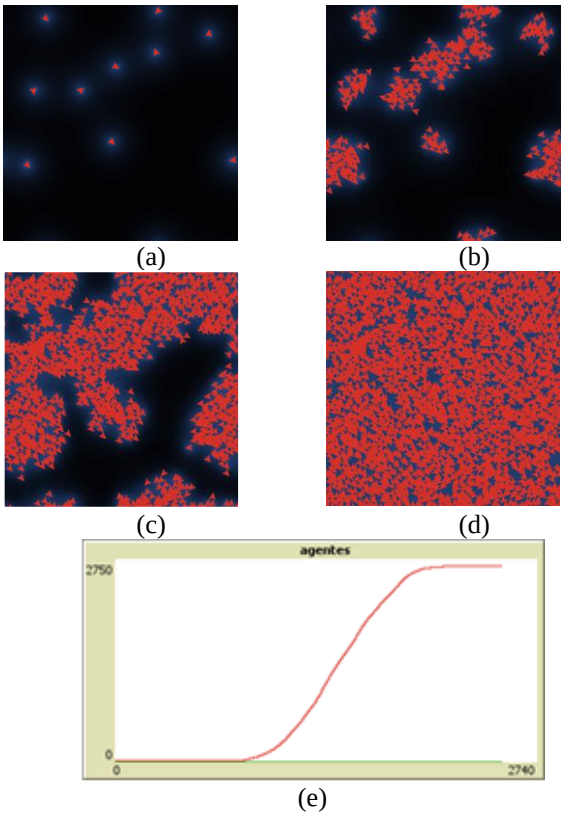
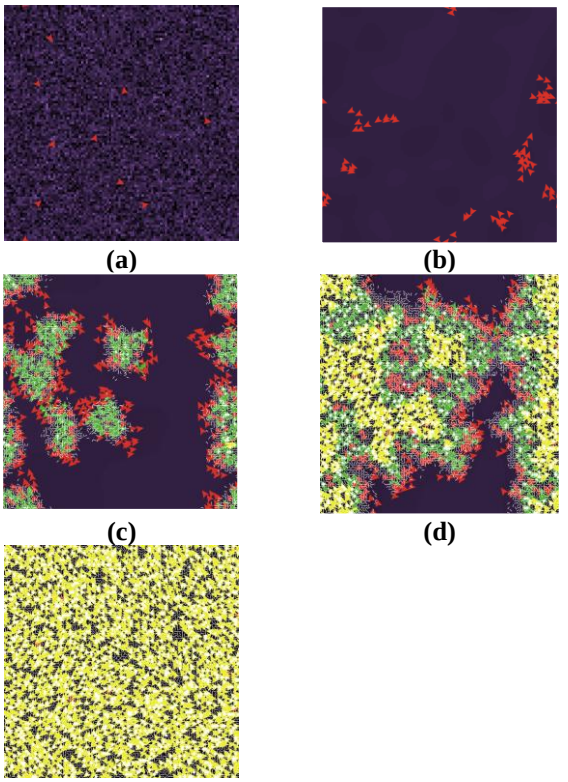


Figura 7. Simulación del caso 4.2.1.

4.2.1 Células madres sembradas con TFG-Beta1, sin tensión.

Los resultados fueron obtenidos (figura 8) con las siguientes condiciones iniciales de simulación: células 10, Tensión X = 0; Tensión Y = 0; FGF-2 = 0 ; FGF-2 umbral 1,02 ng/ml ; FGF-2 consumo = 1,02 ng/ml ; FGF-2-velocidad = 0.2. TGF-beta 1 = 0,97 ng/ml ; TGF-beta 1 umbral = 0,1 ng/ml ; TGF-beta1 consumo = 0,1 ng/ml; TGF-beta1 velocidad = 0,75. Figura 9 (a) estado inicial; Figura 9 (b) 50 ciclos de simulación. Figura 9 (c) 150 ciclos de simulación; Figura 9 (d) 250 ciclos de simulación. Figura 9 (e) estado de confluencia, 600 ciclos de simulación.

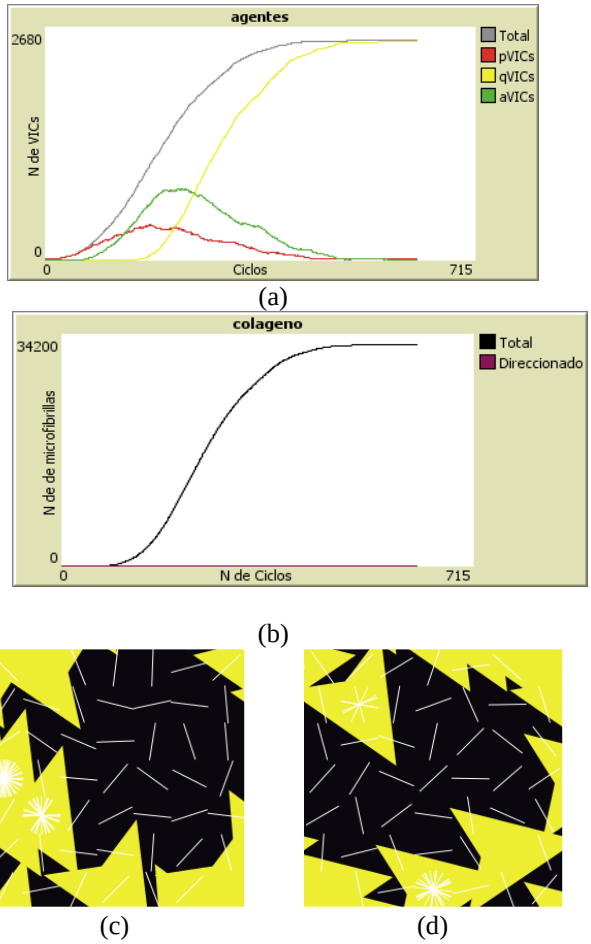
Se muestra, gráfica superior la evolución del número de agentes Vics. Grafica al medio indica la evolución de la producción de las microfibrillas de Colágeno y las imágenes inferiores su direccionamiento aleatorio en 2 sectores diferentes.



(a) (b) (c) (d) (e)

Figura 8. Simulación del caso 4.2.2. Patrones recreados

En las figuras 8, se puede observar que los agentes pVics se reproducen y diferencian a aVIC. Sin embargo la síntesis de Colágeno se hace sin el proceso de remodelación (figuras 9) ya al no haber tensión microfibrilas no se direccionan ni se unen.



(a)

(b)

(c) (d)

Figuras 9. Caso 4.2.4, alineamiento de Colageno.

4.2.4 Células madres sembradas con TFG-Beta1, con tensión en dirección Y.

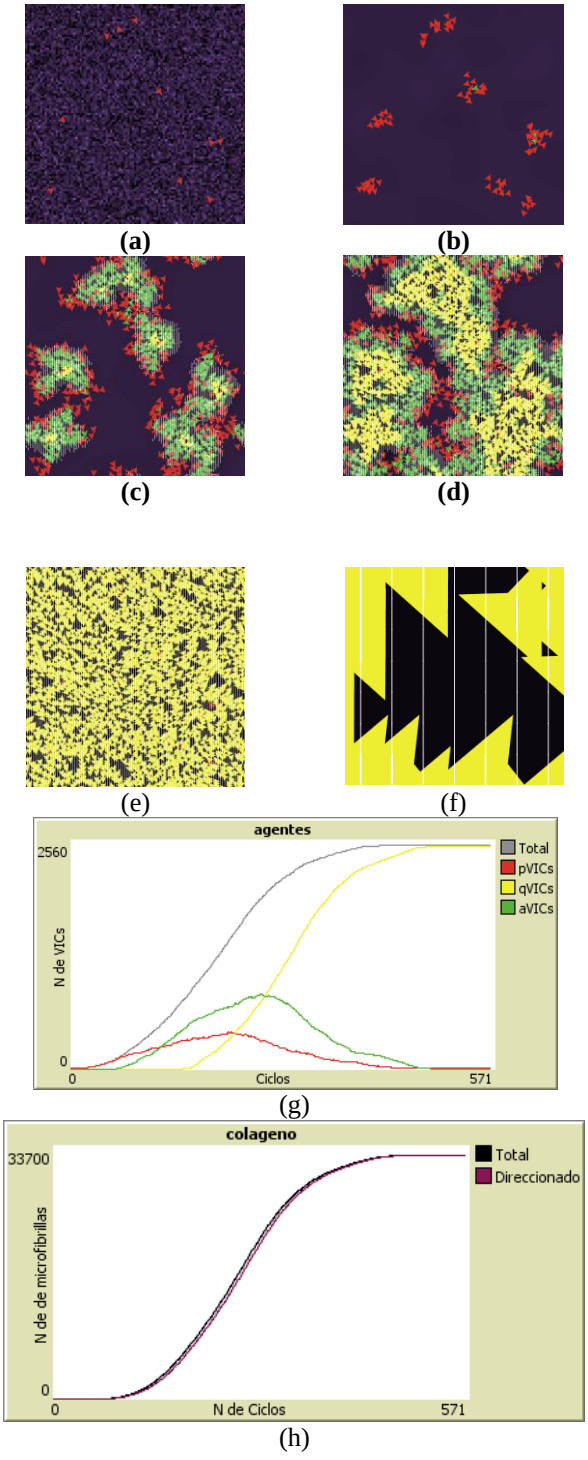


Figura 10. Simulación del caso 4.2.3.

En las figura 10 (a,b,c,d), se observa que los patrones recreados son similares al caso anterior, pero se ha alineado el colágeno, figura inferior derecha en dirección de la tensión aplicada. La figura 10 (e) muestra que se ha llegado al estado de confluencia, mientras que la figura 10 (f) indica que todo el colágeno se ha direccionado siguiendo la tensión.

5. Conclusiones

El lenguaje definido representa los conceptos del dominio, mediante su implementación en Netlogo fue posible recrear el proceso regenerativo de una válvula cardíaca. El modelado en agentes es una herramienta que facilita el proceso y permite realizar un mapeo desde los conceptos del dominio a entidades de manera simple. Las reglas que definen la conducta de los agentes son adecuadas para recrear la dinámica del proceso, si consideramos que las entidades biológicas son no lineales, el método modela de manera adecuada estos sistemas, que en general se encuentran en la frontera del caos y son sensibles a sus condiciones iniciales. El modelo creado ejemplifica que mediante simulación que es posible recrear fenómenos emergentes haciendo hincapié en la conducta de los componentes del sistema y su interacción. El modelo permite predecir situaciones que suceden en experimentos en vitro y es una herramienta útil para la elaboración de hipótesis y comprobación de teorías que puedan ser comprobar experimentalmente. El lenguaje creado permite ocultar detalles de implementación y ejecución de una simulación.

Como trabajo futuro, se pretende encontrar las reglas de transformación que permitan mapear los conceptos vertidos en el metamodelo a un sistema basado en agentes de manera automática. De manera tal, que los expertos del dominio creen modelos basados en el lenguaje y visualicen los resultados del proceso morfogenético sin tener que implementar tareas de programación.

6. Referencias

[1] Adra, S., Sun, T., MacNeil, S., Holcombe, M., & Smallwood, R.. “Development of a three dimensional multiscale computational model of the human epidermis”. PloS one, 5(1), e8511. 2010.

[2] Bianco, P., Robey, P. G., & Simmons, P. J.. “Mesenchymal stem cells: revisiting history, concepts, and assays”. Cell stem cell, 2(4), 313-319. 2008.

[3] d’Inverno, Mark, and Rob Saunders. "Agent-based modelling of stem cell self-organisation in a niche." In Engineering Self-Organising Systems, pp. 52-68. Springer Berlin Heidelberg, 2005.

[4] Glazier J.A., Graner F., “Simulation of the differential adhesion driven rearrangement of biological cells”. Phys. Rev. E 47, 1993 ,pp 2128-2154.

[5] Graner F., Glazier J.A., "Simulation of biological cell-sorting using a two-dimensional extended Potts model". Phys. Rev. Lett. 69, 1992, pp 2013-2016. [Online]. Available: < http://graner.net/francois/publis/graner_hydres.pdf >

[6] Grant, M. R., Mostov, K. E., Tlsty, T. D., & Hunt, C. A.. "Simulating properties of in vitro epithelial cell morphogenesis". PLoS computational biology, 2(10), e129. 2006

[7] Liu, Amber C., Vineet R. Joag, and Avrum I. Gotlieb. "The emerging role of valve interstitial cell phenotypes in regulating heart valve pathobiology." The American journal of pathology 171.5 (2007): 1407-1418.

[8] Montagna Sara, Omicini Andrea, Ricci Alessandro. "A Multiscale Agent-based Model of Morphogenesis in Biological Systems". WOA 2010. 2010 [Online] Available: http://ceur-ws.org/Vol-621/paper04.pdf.

[9] Railsback, Steven F., and Volker Grimm. "Agent-based and individual-based modeling: a practical introduction". Princeton University Press, 2011.

[10] Russell, S. and Norvig P.. "Artificial Intelligence: A Modern Approach". Prentice Hall Pa. 2010

[11] Schaller, G., & Meyer-Hermann, M.. "A modelling approach towards epidermal homoeostasis control". Journal of theoretical biology, 247(3), 554-573. 2007.

[12] Steinberg, Malcolm S.. "Reconstruction of Tissues by Dissociated Cells," Science141 (3579), 401. 1963. [Online]. Avalilable: http://dx.doi.org/10.1126/science.141.3579.401

[13] Sun, T., McMinn, P., Coakley, S., Holcombe, M., Smallwood, R., & MacNeil, S. "An integrated systems biology approach to understanding the rules of keratinocyte colony formation". Journal of the Royal Society Interface, 4(17), 1077-1092. 2007.

[14] Thorne Bryan C., Bailey Alexander M., De Simone Douglas W., Peirce Shayn M..."Agent-based modeling of multicellmorphogenic processes during development,." Birth Defects Research Part C: Embryo Today. Volume 81, Issue 4, pages 344–353. 2007.

Técnicas metaheurísticas para resolver una variante de aplicación industrial del problema de flowshop

Gabriela Minetti
Facultad de Ingeniería
Universidad Nacional de La Pampa
Email: minettig@ing.unlpam.edu.ar

Carolina Salto
Facultad de Ingeniería, CONICET
Universidad Nacional de La Pampa
Email: saltoc@ing.unlpam.edu.ar

Resumen

En este trabajo, proponemos algoritmos metaheurísticos basados en trayectoria para resolver el problema de flowshop *flexible híbrido con secuencias dependientes del tiempo de* setup*, problema que se puede encontrar en muchos ambientes industriales. Las dos primeras propuestas son algoritmos de enfriamiento simulado que utilizan diferentes operadores de movimiento: intercambio e inserción. La tercera y última propuesta es una variante del algoritmo de búsqueda local iterada, propuesto en la literatura, que considera el operador de intercambio en la búsqueda local en lugar del de inserción. Los experimentos numéricos comparan el rendimiento de las distintas propuestas utilizando un amplio conjunto de datos de prueba de uso actual en la literatura. Los resultados muestran que las variantes algorítmicas que aplican el operador de intercambio son más eficientes que las que usan el de inserción para solucionar el problema en estudio.*

1. Introducción

La planificación de la producción es un proceso de toma de decisiones en el nivel operacional. Se puede definir como la asignación de tareas a recursos de producción disponibles y luego secuenciarlas en cada recurso en el tiempo de una manera eficiente. Tal problema de programación ocurre con frecuencia en el ámbito industrial y es conocido en su forma más simple como *flowshop*. Este problema es difícil de resolver debido a su naturaleza combinatoria [3], [7].

En este trabajo nos centramos en una variante del flowshop donde existen varias etapas en la producción con múltiples máquinas (idénticas o no) por etapa, además cabe la posibilidad de que algún trabajo no necesite ser procesado en algunas de las etapas. Esta variante se conoce como problema de flowshop flexible híbrido (*hybrid flexible flowshop problem*, HFFSP). Además, en varias industrias (farmacéutica, metalúrgica, automotriz, entre otras) se consideran tiempos de puesta a punto (*setup*) de las máquinas entre dos trabajos diferentes, los cuales aportan mayor dificultad a los problemas de planificación de la producción. Estos tiempos de setup pueden o no depender de la secuencia. Considerando estas restricciones, el problema considerado en este trabajo es el HSSFP con tiempos de setup dependientes de la secuencia (*sequence dependent setup time*, SDTS/FSSFP), cuyo objetivo es minimizar el tiempo de finalización.

Como veremos, el problema SDST/HFFS no ha sido ampliamente estudiado en la literatura. Sin embargo, se han propuesto algunas heurísticas [10] y metaheurísticas tal como algoritmos genéticos [11], [19], un sistema inmune [22], un algoritmo de búsqueda local iterada [15], entre otros.

En este trabajo proponemos algoritmos metaheurísticos basados en trayectoria para resolver el problema SDST/HFFS. En una primera etapa se presentan dos variantes del algoritmo de enfriamiento simulado, donde se analizarán dos operadores de movimiento para generar el vecindario del estado actual, a fin de determinar cuál de los dos permite una mejor exploración del espacio de soluciones del SDST/HFFS ya que tiene un fuerte impacto en la efectividad del método. Luego, se formula una mejora al algoritmo de búsqueda local iterada propuesto por Naderi et al. [15]. Con el fin de evaluar nuestro aporte, estas propuestas algorítmicas se compararán con algoritmos que representen el estado del arte para el SDST/HFFS.

El artículo se organiza de la siguiente manera. En la Sección 2 se describe el problema y se detallan los algoritmos que lo tratan en la literatura. La Sección 3 explica los algoritmos propuestos en este artículo. La Sección 4 especifica la parametrización utilizada en la experimentación mientras que en la Sección 5 se presentan y analizan los resultados de la experimentación realizada. Finalmente, la Sección6 resume nuestras conclusiones y esboza el trabajo futuro.

2. Descripción del problema

En esta sección se realizará una descripción detallada del problema en cuestión: flowshop flexible híbrido con tiempos de puesta a punto (*setup*) (SDST/HFFS). La sección concluye con un breve estado del arte sobre el mismo.

El problema HFFS se caracteriza por un conjunto de N trabajos, $N = \{1, \ldots, n\}$, disponibles para su procesamiento en tiempo cero en una serie de M etapas, $M = \{1, \ldots, m\}$. En cada etapa i, $i \in M$, se tiene un conjunto de $M_i = \{1, \ldots, m_i\}$ máquinas paralelas idénticas. Cada máquina en cada etapa puede procesar cualquier trabajo. Cada trabajo tiene que ser procesado por sólo una de las M_i máquinas paralelas idénticas. Sin embargo, algunos trabajos pueden saltar algunas etapas. F_j indica el conjunto de etapas que el trabajo j ($j \in N$) tiene que visitar ($1 \leq \mid F_j \mid \leq m$). Sólo se permite saltar etapas

pero no es posible alterar el orden de visita de las mismas. El tiempo de procesamiento del trabajo j en la etapa i se indica por p_{ij}. Finalmente, S_{ijk} es el tiempo de setup entre el trabajo j y el k ($k \in N$) en la etapa i. El tiempo de finalización del trabajo j se indica por C_j, y es el tiempo de finalización del trabajo en la última etapa visitada. El criterio de optimización es minimizar del tiempo de terminación máximo o $makespan$, el cual se calcula como $C_{max} = \max_{j \in N} C_j$.

Un caso especial del SDST/HFFS es el problema de flowshop con múltiples procesadores con sólo dos etapas ($m = 2$). Gupta [4] demostró que este último es NP-duro, lo cual sugiere que el SDST/HFFS pertenece a la misma clase de problemas [9].

Los primeros pasos en la resolución del SDST/HFFS fue el trabajo de Kurz y Askin [10] al introducir reglas de despacho basadas en métodos *greedy*, heurística de múltiple inserción y una adaptación de la regla de Johnson. Los mismos autores propusieron un algoritmo genético (*Genetic Algorithm*, GA) con representación *random keys* (RKGA) [11] cuyos resultados mejoraron los publicados en [10]. Zandieh et al. [22] propusieron un sistema inmune artificial (*artificial inmune system*, IAS) utilizando una representación real de los individuos y el operador de cruce tradicional *order crossover* (OX). Los resultados muestran que el IAS mejora los resultados del RKGA. Mirsanei et al. [14] propusieron un algoritmo de enfriamiento simulado (*Simulated Anneling*, SA) que utiliza una combinación de dos operadores de movimiento (inversión e intercambio), superando los resultados del RKGA [11] y del IAS [22]. Naderi et al. [15] presentaron una regla heurística de despacho dinámica y un búsqueda local iterada (*iterated local search*, ILS). Estos algoritmos se compararon con otros siete algoritmos existentes, siendo la metaheurística búsqueda local iterada (*Iterated Local Search*, ILS) la que mejores resultados obtuvo. Choong et al. [2] propusieron un algoritmo híbrido basado en ejecutar la técnica de optimización por enjambre de partículas (*Particle Swarm Optimization*, PSO) y luego mejorar el resultado obtenido con SA o búsqueda tabú. La mejor combinación híbrida resultó ser PSO con SA, pero sin mejorar el estado del arte en la resolución del problema. Un trabajo reciente es el de [19] que plantearon un GA y analizaron su comportamiento utilizando varios operadores de cruce específicos del problema. Los autores muestran que su GA mejora al ILS [15]. Este GA no se incluirá en la comparación con los algoritmos propuestos en este trabajo, debido a la falta de detalle en su descripción al momento de replicar su funcionamiento.

3. Metaheurísticas para SDST/HFFSP

En esta sección describiremos los algoritmos basados en trayectoria, SA e ILS, propuestos para resolver el problema SDST/HFFS. Para ello es necesario explicar como determinar la secuencia de trabajos al comienzo de cada etapa y la forma de asignar trabajos a cada máquina en cada etapa.

En el problema de flowshop tradicional, sólo es necesario una permutación de trabajos por cada máquina, al menos para los criterios basados en el tiempo de finalización. En cambio en el problema HFFS, mantener una única permutación de trabajos en todas las etapas resulta en una planificación poco eficiente. Además, en el flowshop tradicional no es necesario asignar trabajos a las máquinas dado que existe una sola máquina por etapa. En cambio, HFFS con más de una máquina por etapa puede ser visto como una serie de problemas de máquinas paralelas.

Esto nos lleva a adoptar una regla de despacho que contemple estas dos características, por ende elegimos la regla heurística modificada de despacho dinámico (*Modified dynamic dispatching rule, MDDR*), propuesta por Naderi et al. en [15]. MDDR se trata de un método muy rápido y efectivo que no separa las decisiones de secuenciamiento de trabajos y asignación de máquinas. Comienza con la etapa 1, considerando todos los trabajos que visitan esta primera etapa. Para estos trabajos se calcula el mínimo tiempo de finalización más temprano (*earliest completion time, ECT*) para todas las máquinas paralelas en esta etapa. El trabajo con el mínimo ECT se planifica y asigna a la máquina que produzca el valor mínimo de ECT. Este proceso se repite hasta que se asignen todos los trabajos en la etapa uno. Luego, el proceso continua con la segunda etapa, pero esta vez considerando los tiempos de finalización de los trabajos en la etapa anterior. Esto se repite en las restantes etapas. Esta regla de despacho se considera dinámica ya que el valor de ECT puede variar luego de planificar cada trabajo.

Para una mejor comprensión de los algoritmos propuestos, es necesario explicar previamente los métodos de codificación de soluciones y de generación de la solución inicial utilizados. Si bien durante la última década se han empleado muchos métodos de codificación, hemos adoptado uno de los más utilizados denominado representación basada en trabajos (*job-based representation, JBR*) [5]. En JBR, la secuencia de trabajos en la etapa uno se representa por una simple permutación de trabajos que indica el orden en el cual se ejecutan los mismos. En tanto que, la secuencia para las siguientes etapas se determina usando la regla de despacho MDDR.

Habitualmente, la generación de las soluciones con las cuales una metaheurística inicia su búsqueda es aleatoria. Sin embargo, considerar "buenas" soluciones iniciales en los primeros pasos del método ayuda a aprovechar las regiones prometedoras del espacio de búsqueda [13], [18], [21]. Para ello empleamos una adaptación de la bien conocida técnica NEH [16], denominada NEHH, ya que se trata de la heurística con mejor desempeño en el tratamiento de HFFS [17].

3.1. Enfriamiento Simulado

Enfriamiento simulado (*Simulated Annealing, SA*) es un método Monte-Carlo simple y de propósito general desarrollado para optimización combinatoria en la década de 1980 por Kirkpatrick et al. en [8]. SA es un algoritmo metaheurístico basado en una analogía con los sistemas físicos de enfriamiento. Esto es un proceso térmico para la obtención de los estados de baja energía de un sólido en un baño de calor.

En los primeros pasos (a alta temperatura), SA acepta soluciones con mayores costos en virtud de una cierta pro-

babilidad, con el fin de explorar el espacio de búsqueda y para escapar de óptimos locales. Durante el proceso de enfriamiento esta probabilidad disminuye de acuerdo a la temperatura de enfriamiento, la intensificación de la búsqueda y la reducción de la exploración con el fin de explotar una zona restringida de un espacio de búsqueda. En los últimos pasos la exploración es nula (o casi nula) y la explotación es total (o casi total). SA evoluciona a través de una secuencia de transiciones entre estados; las cuales son generadas por probabilidades de transición. En consecuencia, SA puede ser matemáticamente modelado por cadenas de Markov, donde se genera una secuencia de cadenas usando una probabilidad de transición cuyo cálculo involucra a la temperatura actual.

En otras palabras, SA inicia con una solución S_0 y a continuación genera una solución vecina S_1 mediante la aplicación de un operador de movimiento. Si S_1 mejora el valor de la función objetivo se acepta con probabilidad 1; en caso contrario se calcula la probabilidad para aceptarla mediante el criterio de Boltzman ($exp((E_1 - E_0)/T)$) que involucra el parámetro correspondiente a la temperatura actual. Este parámetro se actualiza cuando la longitud de la cadena de Markov es nula siguiendo algún esquema de decaimiento.

SA es una metaheurística fácilmente adaptable a problemas de planificación de la producción [1], [6], ya que es posible adecuar su diseño a una representación por permutación. Esto significa, por un lado, poder usar como función de evaluación la regla de despacho MDDR, y por otro lado, utilizar operadores especialmente diseñados y ampliamente usados, para este tipo de representación tales como el operador de intercambio y el de inserción. Sin embargo, la facilidad de adaptación al problema no es la única razón por la que se elige SA, sino que también se lo escoge por su capacidad de explorar el espacio de búsqueda en etapas tempranas localizando las regiones más prometedoras para luego explotarlas. De esta forma, surgen SAint y SAins, dos variantes basadas en SA que utilizan el operador de intercambio y el de inserción, respectivamente, para generar las soluciones vecinas. En el algoritmo 1, se muestra un pseudo-código común a SAint y a SAins.

Algoritmo 1 Pseudo-código de SAint y SAins

```
k = 0;
inicializar T; {inicializar la temperatura, T}
NEHH(S0); {generar la solución inicial, S0, usando NEHH [17]}
evaluar S0 en E0;
repeat
    repeat
        k = k + 1;
        generar S1 desde S0; {generar la solución vecina, S1, usando los operadores
        de intercambio o inserción}
        evaluar S1 en E1;
        {si la nueva solución es mejor que la actual, S1 es aceptada}
        if (E1 − E0) ≤ 0 then
            S0 = S1;
            E0 = E1;
            {si la nueva solución es peor que la actual, S1 es aceptada bajo la
            probabilidad: exp((E1 − E0)/T)}
        else
            if exp((E1 − E0)/T) > random[0, 1) then
                S0 = S1;
                E0 = E1;
            end if
        end if
    until (k mod Long. Cadena de Markov) == 0
    actualizar T;
until condición de corte sea satisfecha
return S0;
```

3.2. Búsqueda local iterada

La búsqueda local iterada (*Iterated Local Search, ILS*) [12], [20] es una metaheurística basada en un concepto simple pero muy efectivo. En cada iteración, la solución actual es perturbada y a esta nueva solución se le aplica un método de búsqueda local para mejorarla. Este nuevo mínimo local obtenido por el método de mejora puede ser aceptado como nueva solución actual si pasa un *test de aceptación*. La importancia del proceso de perturbación es obvia: si es demasiado pequeño puede que el algoritmo no sea capaz de escapar del mínimo local; por otro lado, si es demasiado grande, la perturbación puede hacer que el algoritmo sea como un método de búsqueda local con un reinicio aleatorio. Por lo tanto, el método de perturbación debe generar una nueva solución que sirva como inicio a la búsqueda local, pero que no debe estar muy lejos de la actual para que no sea una solución aleatoria. El criterio de aceptación actúa como contra-balance, ya que filtra la aceptación de nuevas soluciones dependiendo de la historia de búsqueda y de las características del nuevo mínimo local.

En [15], Naderi et al. proponen una ILS para SDST/HFFSP que resulta ser efectiva y eficiente y, que de ahora en adelante denominaremos *ILSins*. Este algoritmo comienza la búsqueda con una solución inicial generada por NEHH, luego una búsqueda local (*Local Search, LS*) basada en el operador de inserción se aplica a la solución candidata x. Esta búsqueda se detiene cuando se encuentra una primera mejora. Cuando no se encuentra ninguna mejora, se incrementa un contador. Este contador vuelve a cero cada vez que la búsqueda local mejora la solución actual. Si este contador supera un determinado umbral, se aplica un mecanismo de perturbación (*Giant Leap*, GL) basado en una colección de movimientos de inserción, con el fin de redireccionar la búsqueda hacia regiones más prometedoras del espacio de soluciones.

Con el objetivo de incrementar la calidad de las soluciones, en este trabajo se propone una variante de la metaheurística propuesta por Naderi, denominada *ILSint*. Dicha variante consiste en modificar el proceso de búsqueda local. Esta nueva LS se basa en el operador de intercambio y procede de la siguiente forma: el trabajo en la primera posición de x, denominado x_1 se intercambia por el trabajo ubicado en una posición diferente elegida al azar en la permutación. Si esta nueva permutación x' resulta con un mejor makespan, la solución actual x es reemplazada por x' y la búsqueda local termina. De lo contrario, el procedimiento continúa con x_2. La búsqueda itera en la mayoría de los trabajos, sin repetición. El fundamento de este procedimiento es realizar una búsqueda local muy rápida. En el algoritmo 2 se describe un pseudo-código común a ILSint y a ILSins, y en el algoritmo 3 se detalla el esquema general de la búsqueda local basada en intercambio.

Algoritmo 2 Pseudo-código de ILSins e ILSint

```
Procedure ILS
contador = 0;
Inicializar x; {ILSins e ILSint usan NEHH durante la inicialización}
x_best = x;
while condición de corte no sea satisfecha do
    x' = ls(x); {LS basada en inserción o intercambio según corresponda}
    if f(x') <f(x) then
        x = x';
        if f(x') <f(x_best) then
            x_best = x';
        end if
    else
        contador = contador + 1;
        {Aplicación del operador de perturbación, GL}
        if contador >umbral then
            contador = 0;
            for r=1 to nu_move do
                s̃(r) = GL(x);
            end for
            x = s̃_best;
        end if
    end if
end while
return x;
```

Algoritmo 3 Esquema general de la LS propuesta para ILSint

```
Procedure LS {LS basada en el operador de intercambio}
i = 0;
while i ≤ n do
    x' = la secuencia producida al intercambiar x_i con el trabajo ubicado en una
    posición diferente elegida al azar;
    if C_max(x') <C_max(x) then
        x = x';
        break;
    end if
    i = i + 1;
end while
return X;
```

4. Diseño experimental

En esta sección se muestran los valores paramétricos utilizados para evaluar el rendimiento de las metaheurísticas basadas en trayectoria descriptas en la Sección 3 para resolver el problema SDTS/FSSFP. Las variantes propuestas son:

- SAint: SA con operador de intercambio para generar la solución vecina.
- SAins: SA con operador de inserción para generar la solución vecina.
- ILSint: ILS con una LS basada en el operador de intercambio.
- ILSins: ILS con una LS basada en el operador de inserción (originalmente propuesto en [15].

Los parámetros específicos del SA se fijaron en los siguientes valores. La temperatura inicia con un valor de 0.99 y el decaimiento se planifica usando el enfriamiento proporcional. En tanto que el número de iteraciones entre dos cambios consecutivos de temperatura (longitud de la cadena de Markov) se fija en 30. Este valor surgió de una experimentación previa realizada tomando valores de iteraciones en el conjunto 10, 30 y 60, resultando 30 el número de iteraciones más adecuado para el proceso, con el fin de alcanzar el cuasi equilibrio en cada nivel de temperatura.

Los valores de los parámetros para las variantes de ILS se describen a continuación. La cantidad máxima de iteraciones sin observar mejoras entre perturbaciones (umbral) se fija en 30. El operador de perturbación GL utiliza la operación de inserción para generar 15 (nu_move) soluciones perturbadas, siendo 2 la cantidad de trabajos reubicados. Estos parámetros son los adoptados en [15].

Cuadro I
VALORES PARAMÉTRICOS USADOS POR LAS VARIANTES DE SA Y DE ILS.

	Parámetro	***Valor***
SA	Longitud de la cadena de Markov	30
	Temperatura Inicial	0.99
ILS	Nro. de iteraciones sin mejora (umbral)	15
	Nro. de perturbaciones (nu_move)	30
SA e ILS	Representación de la solución	Basada en permutación
	Método para generar la solución inicial	NEHH
	Condición de terminación	$n^2 \times m \times 1{,}5ms$

Todas las variantes algorítmicas planteadas utilizan la heurística NEHH para generar la solución inicial. A fin de que la comparación entre las distintas metaheurísticas resulte confiable, el criterio de parada fue configurado en $n^2 \times m \times 1{,}5ms$ de tiempo de CPU. Este criterio es el adoptado en varios trabajos [15], [19]. El cuadro I muestra un resumen de los parámetros utilizados en este trabajo.

El conjunto de prueba utilizado está formado por 960 instancias (disponibles en http:\\soa.iti.es). Las instancias resultan de la combinación de varios valores de n y m, donde $n = \{20, 50, 80, 120\}$ y $m = \{2, 4, 8\}$. Los tiempos de procesamiento están generados desde una distribución uniforme [1, 99]. Los tiempos de setup pertenecen a cuatro distribuciones distintas ([1, 25], [1, 50], [1, 99] y [1, 125]) que permiten analizar los efectos que tiene el tiempo de setup en los resultados; se corresponden con un 25 %, 50 %, 100 % y 125 % del tiempo de procesamiento, respectivamente. Se consideran grupos de instancias con dos máquinas paralelas en cada etapa y otros grupos donde la cantidad de máquinas por etapas se obtiene de una distribución uniforme en el rango [1, 4]. La probabilidad de que un trabajo salte una etapa es de 0.1 y 0.4. En el cuadro II se presenta un resumen de estas instancias.

Los algoritmos fueron implementados en C++. Para la experimentación se utilizó un cluster de PCs con procesador Intel Core i7 a 3.90 GHz y 4GB de RAM. Cada instancia fue ejecutada 30 veces para cada configuración algorítmica. Para comparar los distintos algoritmos se utiliza la desviación porcentual relativa (DPR) de las mejores soluciones conocidas, esta métrica se describe en la Ecuación 1, donde Alg_{sol} es el mínimo C_{max} obtenido por un determinado algoritmo para

Cuadro II
INFORMACIÓN DE LAS INSTANCIAS.

Factores	Valores
n	20, 50, 80, 120
m	2, 4, 8
m_i	Constante: 2
	Variable: U(1, 4)
p_{ij}	U(1, 99)
S_{ijk}	U(1, 25), U(1, 50)
	U(1, 100), U(1, 125)
S_p	0.1, 0.4

una instancia del problema en las 30 ejecuciones y min_{sol} es el mínimo C_{max} de los algoritmos bajo comparación para una determinada instancia. Obviamente, son deseables valores pequeños de DPR para un algoritmo, ya que indican una desviación porcentual menor de sus soluciones respecto del mínimo.

$$DPR = \frac{Alg_{sol} - min_{sol}}{min_{sol}} \times 100 \quad (1)$$

La experimentación realizada ha sido extensa: las cuatro variantes algorítmicas en estudio se evaluaron con cada una de las instancias, en total suman 3840 combinaciones (4 algoritmos $\times$ 960 instancias). Debido a la naturaleza estocástica de los algoritmos, cada una de ellas se ejecutan 30 veces para obtener una muestra confiable. En consecuencia se llevaron a cabo 30 $\times$ 3840 ejecuciones, totalizando 115200 ejecuciones.

Hemos realizado un análisis estadístico de los resultados a fin de determinar si existen diferencias significativas entre los algoritmos comparados. Como paso previo, se determina si los datos siguen una distribución normal al aplicar la prueba de Shapiro-Wilks y homogeneidad de varianzas (homocedasticidad). En el caso de que los datos resulten con una distribución normal, se realiza la prueba de ANOVA. En caso contrario, se usa el test no paramétrico Kruskal Wallis. El valor de α se fija en 0.01, para indicar un nivel de confianza del 99 % en los resultados.

5. Análisis de los resultados

Esta sección está dedicada al análisis de los resultados obtenidos en la experimentación llevada a cabo sobre los variantes de SA e ILS mencionadas en el apartado anterior.

En las siguientes subsecciones, se medirán en primer instancia parámetros básicos tales como DPR (ver ecuación 1) y la cantidad de veces que un algoritmo encuentra la mejor solución (#min), cuyos objetivos es ofrecer un estudio de las variantes algorítmicas propuestas desde el punto de vista de la calidad de las soluciones encontradas. Por último, se realiza un análisis del esfuerzo numérico y tiempo empleado por los algoritmos para hallar una solución, esto permite estudiar la velocidad de convergencia de cada algoritmo para encontrar la mejor solución.

5.1. Análisis de calidad

Comenzaremos el análisis con la comparación del rendimiento de los algoritmos SA propuestos con dos métodos para generar el vecindario. El cuadro III muestra los valores promedios de DPR (columnas 2 y 3) y de *#min* (columnas 4 y 5) para cada grupo de instancias discriminados para las variantes de SA propuestas. Los resultados de los experimentos están promediados para cada combinación de n y m. A fin de verificar la validez estadística de los resultados, la última columna de este cuadro muestra los valores *p* obtenidos del estudio estadístico realizado. La última fila del cuadro presenta valores promedios de las dos métricas y se resaltan en negritas los mejores valores para cada una.

Cuadro III
VALORES PROMEDIOS DE DPR Y #MIN OBTENIDOS POR CADA SA EN CADA GRUPO DE INSTANCIAS.

Inst.	*DPR*		*#min*		*Valor p*
	SAint	*SAins*	*SAint*	*SAins*	
20x2	**1.85**	6.36	**64**	16	1.42E-09
20x4	**0.75**	5.36	**66**	14	1.93E-15
20x8	**0.68**	4.73	**60**	20	5.66E-12
50x2	**0.79**	6.82	**69**	11	<2.20E-16
50x4	**1.57**	5.23	**66**	14	<2.17E-15
50x8	**0.13**	5.25	**74**	6	<2.20E-16
80x2	**2.19**	5.75	**67**	13	4.02E-15
80x4	**0.72**	4.11	**68**	12	<2.20E-16
80x8	**0.17**	3.70	**70**	10	<2.20E-16
120x2	**0.79**	6.02	**75**	5	<2.20E-16
120x4	**1.01**	4.14	**69**	11	<2.20E-16
120x8	**0.25**	3.10	**68**	12	<2.20E-16
Promedio	**0.91**	5.05	**68**	12	–

Como se puede observar en el cuadro III, los valores promedios de DPR para SAint son menores que los de SAins para todas las instancias, indicando que el primer algoritmo encuentra soluciones con valores de makespan menores que el segundo. Además, los valores de *#min* de SAint superan, en promedio, en más de cuatro veces el valor correspondiente a SAins. Los valores *p* son menores que α, entonces con un 99 % de nivel de confianza se puede afirmar que hay diferencias significativas entre la calidad de soluciones obtenidas por SAint y SAins. De esta manera, se puede inferir la influencia del operador de movimiento en la calidad final de los resultados. Es decir que, la operación de intercambio le permite al SA explotar con mayor eficacia el espacio de búsqueda que en el caso de usar el operador de inserción.

Dado que SAint resultó ser el algoritmo basado en SA que encuentra soluciones de mayor calidad, a continuación se comparan sus resultados con los obtenidos por las variantes ILS. En el cuadro IV se muestran los valores promedio de DPR y de #min obtenidos por estos tres algoritmos. De dicha comparación surge que los algoritmos basados en ILS obtienen soluciones con mejores makespan finales que SAint, debido a que los DPR son notablemente menores a los calculados para este último. Como muestra la última columna, el análisis de varianza avala las diferencias antes mencionadas, resultando ILSint el algoritmo que obtiene los valores mínimos de C_{max}.

Al contrastar las dos versiones de ILS observamos que ILSint presenta valores de DPR inferiores a ILSins para los tres primeros grupos de instancias ($n \in \{20, 50, 80\}$), mientras que para las instancias con $n = 120$ los valores son similares, obteniendo ILSins una ventaja mínima. Para confirmar estas últimas observaciones hemos realizado pruebas estadísticas considerando sólo las dos versiones de ILS, las cuales indican que las diferencias entre ILSint y ILSins son significativas (valores p cercanos a 0) para $n \in \{20, 50, 80\}$, mientras que para $n = 120$ los resultados son estadísticamente similares (valores p mayores al nivel de significancia α).

Analizando la cantidad de valores de #min de cada algoritmo por grupo de instancias, ILSint tiene valores considerable-

mente mayores (cercanos al total de instancias dentro de cada grupo) para los grupos de instancias con $n \in \{20, 50, 80\}$, seguido por SAint y ILSins. Para las instancias con $n = 120$, los valores de #min son muy similares entre los distintos algoritmos.

Del análisis anterior se desprende que, SAint e ILSint fueron los mejores algoritmos dentro de cada tipo de metaheurística. Esto da soporte a la idea de que el operador de intercambio es el más adecuado para el problema en estudio. La razón de esto es que el operador de intercambio permite conservar más bloques sin modificaciones dentro de una solución que en el caso del operador de inserción. Esto sucede porque la inserción de un trabajo en una posición aleatoria desplazan todos los trabajos a partir de tal posición, ocasionando una disrupción muy importante en las soluciones y saltos aleatorios en el espacio de búsqueda con el consecuente deterioro de la calidad.

En resumen, los algoritmos basados en ILS presentan el mejor comportamiento en cuanto a la calidad de las soluciones, especialmente cuando se utiliza el operador de intercambio como operador de búsqueda local. Es decir, que ILSint es el que obtiene los valores mínimos de makespan.

5.2. Análisis de desempeño

El análisis de desempeño se lleva a cabo considerando el esfuerzo numérico realizado y el tiempo empleado por SAint, SAins, ILSint e ILSins. El esfuerzo numérico se mide teniendo en cuenta el número promedio de evaluaciones realizadas para que cada algoritmo halle la mejor solución, y por el promedio del total de evaluaciones realizadas durante el proceso completo de búsqueda. Finalmente, se analiza el tiempo promedio empleado por cada opción algorítmica para encontrar la mejor solución, el cual es medido en segundos. Cabe aclarar que no es necesario comparar el tiempo total dedicado por cada algoritmo, dado que se estableció como condición de corte el mismo tiempo de ejecución variando de acuerdo al tamaño de las instancias.

En primer lugar, procederemos con el estudio del esfuerzo para localizar la mejor solución. Es decir, la cantidad de evaluaciones promedio de la función objetivo necesarias para alcanzar las soluciones con menor makespan con cada uno de los algoritmos. El cuadro V muestra los valores obtenidos para esta métrica. Se observa que las variantes de ILS tienden a necesitar un menor número de evaluaciones que las variantes de SA. Para confirmar estas observaciones se han usado pruebas estadísticas, las cuales indican que las diferencias entre los distintos algoritmos son significativas (valores p cercanos a 0 en la última columna del cuadro V). En tanto que, en la figura 1 se advierten dos grupos bien diferenciados: por un lado el formado por las variantes de SA y el otro, el conformado por las variantes de ILS. Este último reduce aproximadamente un 50 % la cantidad de evaluaciones necesarias para alcanzar las mejores soluciones cuando se lo compara con el desempeño de SA. Siendo ILSint el algoritmo que necesita en promedio el menor esfuerzo para alcanzar las mejores soluciones en 8 de los 12 grupos de instancias.

Continuamos el estudio del esfuerzo numérico teniendo en cuenta también las evaluaciones realizadas durante todo el proceso se búsqueda. Como se muestra en la figura 2, las cuatro técnicas algorítmicas realizan un número total de evaluaciones estadísticamente similar. Si, además, consideramos que el tiempo de ejecución total es igual para todas ellas, deducimos

Cuadro IV
VALORES PROMEDIOS DE DPR Y #MIN OBTENIDOS POR SAINT, ILSINT, ILSINS EN CADA GRUPO DE INSTANCIAS.

Inst.	*DPR*			*#min*			*Valor p*
	SAint	*ILSint*	*ILSins*	*SAint*	*ILSint*	*ILSins*	
20x2	13.93	**0.00**	1.22	5	**74**	1	<2.20E-16
20x4	8.23	**0.00**	1.00	4	**75**	1	<2.20E-16
20x8	6.68	**0.06**	0.64	3	**68**	9	<2.20E-16
50x2	6.30	**0.06**	2.29	9	**70**	1	<2.20E-16
50x4	8.07	**0.04**	1.83	5	**73**	2	<2.20E-16
50x8	5.74	**0.06**	1.64	2	**76**	2	<2.20E-16
80x2	7.75	**0.16**	2.14	14	**62**	4	<2.20E-16
80x4	6.32	**0.04**	1.69	7	**70**	3	<2.20E-16
80x8	5.62	**0.02**	1.56	1	**77**	2	<2.20E-16
120x2	4.41	0.88	**0.87**	**33**	27	20	6.23E-01
120x4	5.75	0.70	**0.61**	20	27	**33**	7.08E-10
120x8	3.572	0.48	**0.45**	12	26	**42**	<2.20E-16
Promedio	6.87	**0.21**	1.33	9.58	**60.41**	10	–

Cuadro V
NÚMERO PROMEDIO DE EVALUACIONES REALIZADAS POR CADA ALGORITMO PARA ENCONTRAR LA MEJOR SOLUCIÓN EN GRUPO DE INSTANCIAS.

Inst	*Algoritmos*				*Valor p*
	SAint	*SAins*	*ILSint*	*ILSins*	
20x2	714960	747799	**284359**	361750	<2.20 E-16
20x4	562998	587730	**277894**	281207	<2.20 E-16
20x8	490962	503395	**227554**	253201	<2.20 E-16
50x2	681551	718550	**339083**	375352	<2.20 E-16
50x4	672317	692112	**314303**	373033	<2.20 E-16
50x8	692937	704472	373581	**352159**	<2.20 E-16
80x2	783941	825918	**425485**	438931	<2.20 E-16
80x4	795645	817104	**437034**	469658	<2.20 E-16
80x8	804433	815757	452669	**401275**	<2.20 E-16
120x2	851031	889680	438444	**423376**	<2.20 E-16
120x4	870691	890372	**449977**	455684	<2.20 E-16
120x8	885035	895697	518881	**515507**	<2.20 E-16
Promedio	733875	757382	**378272**	391761	–

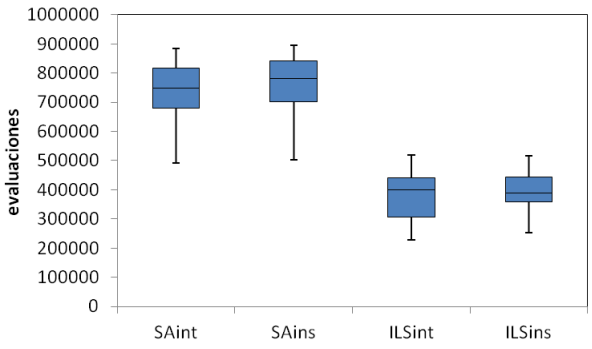


Figura 1. Boxplot de las evaluaciones para obtener las mejores soluciones para todos los algoritmos.

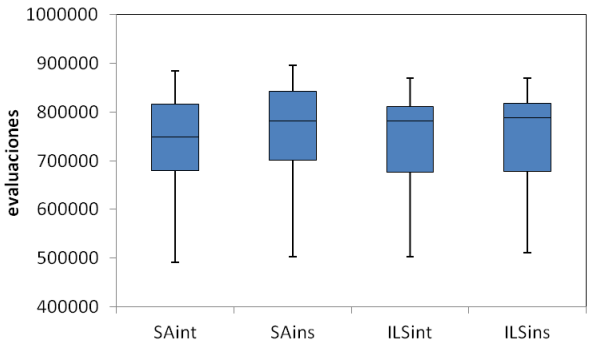


Figura 2. Boxplot del total de evaluaciones realizadas por los algoritmos durante todo el proceso de búsqueda.

Cuadro VI
TIEMPO PROMEDIO EMPLEADO POR CADA ALGORITMO PARA ENCONTRAR LA MEJOR SOLUCIÓN EN CADA GRUPO DE INSTANCIAS.

Inst.	*Algoritmos*				*Valor p*
	SAint	*SAins*	*ILSint*	*ILSins*	
20x2	1.21	1.25	**0.31**	0.48	2.00E-16
20x4	2.10	2.18	0.93	**0.88**	2.00E-16
20x8	4.08	4.25	**1.81**	2.00	2.00E-16
50x2	7.05	7.11	**3.40**	3.78	2.00E-16
50x4	14.06	14.08	**6.58**	7.89	2.00E-16
50x8	29.05	29.11	15.98	**15.01**	2.00E-16
80x2	19.06	19.08	**10.18**	10.50	2.00E-16
80x4	38.04	38.13	**21.16**	22.66	2.00E-16
80x8	76.03	76.03	43.78	**38.71**	2.00E-16
120x2	43.03	43.05	22.01	**21.21**	2.00E-16
120x4	86.06	86.04	**44.78**	45.51	2.00E-16
120x8	172.00	172.06	102.90	**102.09**	2.00E-16
Promedio	40.98	41.03	22.82	**22.56**	–

que el esfuerzo computacional realizado en cada ciclo de la búsqueda es similar. La razón de esto es que todas trabajan con una única solución por iteración y utilizan operadores de movimiento con características similares. En tanto que, el proceso de perturbación en ILS es compensado por el número de iteraciones (longitud de la cadena de Markov) que realiza SA entre dos cambios consecutivos de temperatura. Esto, también, nos lleva a inferir que el esfuerzo computacional realizado por los operadores de intercambio y de inserción no presentan diferencias importantes.

Finalmente, analizamos la otra métrica interesante para medir el desempeño de una metaheurística que es el tiempo de ejecución necesario para alcanzar la mejor solución y se muestra en el cuadro VI. Como se puede observar, las variantes más rápidas son las correspondientes a ILS, con tiempos menores (de aproximadamente el 50 %) a los presentados por los algoritmos basados en SA, independientemente de la combinación de n y m que se analice. Esta es una situación esperada en función a lo observado previamente al analizar el esfuerzo numérico de cada variante. Cabe destacar que, todas las diferencias marcadas son estadísticamente significativas (valores p cercanos a cero en la última columna del cuadro VI).

Resumiendo, las técnicas basadas en ILS se desempeñan con casi un 50 % más de eficiencia que las variantes de SA para hallar los menores valores de makespan. Siendo nuestra propuesta, ILSint, la que presenta un mejor comportamiento en 7 de las 12 instancias.

6. Conclusion

En este trabajo se presentaron tres algoritmos para resolver el problema NP-duro SDST/HFFS: SAint, SAins e ILSint. En los dos primeros casos se trata de dos versiones de SA que utilizan los operadores de intercambio e inserción, respectivamente, para generar soluciones vecinas. En tanto que, ILSint resulta de una mejora realizada al ILS presentado en [15], renombrado en este trabajo como ILSins. Esta mejora consiste en modificar el operador de la búsqueda local. Es decir, se reemplazó el operador de inserción usado en ILSins por el de intercambio, surgiendo de esta manera ILSint.

Del análisis de los resultados obtenidos por estos cuatro algoritmos, en primer lugar se desprendió que las técnicas que usan el operador de intercambio realizan una explotación mucho más eficaz del espacio de búsqueda que al aplicar el operador de inserción. La razón principal de esto es la gran disrupción que genera el operador de inserción en las soluciones. En segundo lugar, se determinó con sustento estadístico, que los algoritmos basados en ILS obtuvieron soluciones de mejor calidad que los basados en SA y el esfuerzo computacional para lograrlo fue significativamente menor. Por último, es importante destacar que la versión de ILS propuesta en este trabajo obtuvo los mejores resultados empleando el menor esfuerzo computacional en un alto porcentaje de las instancias.

Como futuras líneas de investigación, intentaremos en primer lugar diseñar e implementar operadores para generar soluciones vecinas que utilicen información heurística del problema en cuestión. Estos operadores se podrán utilizar luego en metaheurísticas tales como SA, ILS y GA con el fin de desarrollar algoritmos aún más eficientes que los actuales en la resolución de SDST/HFFS. Como una segunda línea de investigación, pretendemos incursionar en otras variantes del problema HFFS, dado que resulta necesario en diferentes actividades económicas desarrolladas en la región.

Referencias

[1] U.K. Chakraborty. Minimizing total flow time in permutation flow shop scheduling with improved simulated annealing. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 158–165, May 2009.

[2] F. Choong, S. Phon-Amnuaisuk, and M.Y. Alias. Metaheuristic methods in hybrid flow shop scheduling problem. *Expert Systems with Applications*, 38:10787–10793, 2011.

[3] E. Coffman and J. Bruno. *Computer and job-shop scheduling theory*. John Wiley & Sons, New York, 1976.

[4] J. Gupta. Two-stage, hybrid flowshop scheduling problem. *Journal of the Operational Research Society*, 39(4):359–364, 1988.

[5] C.W. Holsapple, V.S. Jacob, R. Pakath, and J.S. Zaveri. A genetics-based hybrid scheduler for generating static schedules in flexible manufacturing contexts. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(4):953–972, Jul 1993.

[6] P. Jarosław, S. Czesław, and Ż. Dominik. Optimizing bicriteria flow shop scheduling problem by simulated annealing algorithm. *Procedia Computer Science*, 18(0):936 – 945, 2013. 2013 International Conference on Computational Science.

[7] A. Rinnooy Kan. *Machine Scheduling Problems: Classification, complexity and computations*. Martinus Nijhoff, The Hague, 1976.

[8] S. Kirkpatrick, C.D. Gelatt Jr, and M.P. Vecchi. Optimization by simulated annealing. *Science*, (220):671–680, 1983.

[9] T. Kis and E. Pesch. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. *European Journal of Operational Research*, 164(3):592–608, 2005.

[10] M. Kurz and R. Askin. Comparing scheduling rules for flexible flow lines. *International Journal of Production Economics*, 85(3):371–388, 2003.

[11] M. Kurz and R. Askn. Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research*, 159(1):66–82, 2004.

[12] H. R. Lourenço, O. Martin, and T. Stützle. Iterated local search. In *Handbook of Metaheuristics*, pages 321–353. Kluwer Academic Publishers, 2002.

[13] G. Minetti, E. Alba, and G. Luque. Seeding strategies and recombination operators for solving the dna fragment assembly problem. *Inf. Process. Lett.*, 108(3):94–100, 2008.

[14] H. Mirsanei, M. Zandieh, M. Moayed, and M. Khabbazi. A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 22(965-978), 2011.

[15] B. Naderi, R. Ruiz, and M. Zandieh. Algorithms for a realistic variant of flowshop scheduling. *Computers & Operations Research*, 37(2):236 – 246, 2010.

[16] M. Nawaz, E. Enscore Jr, and I. Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91 – 95, 1983.

[17] R. Ruiz and C. Maroto. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3):781–800, March 2006.

[18] C. Salto, E. Alba, and J. M. Molina. *Optimization Techniques for Solving Complex Problems*, chapter Greedy Seeding and Problem-Specific Operators for GAs Solving Strip Packing Problems, pages 361–378. John Wiley & Sons, Inc., 2009.

[19] A Sioud, M. Gravel, and C. Gagne. A genetic algorithm for solving a hybrid flexible flowshop with sequence dependent setup times. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2512–2516, June 2013.

[20] T. Stützle. Local search algorithms for combinatorial problems analysis, algorithms and new applications. Technical report, DISKI Dissertationen zur KÄunstliken Intelligenz, Sankt Augustin, Germany, 1999.

[21] C.H. Westerberg and J. Levine. Investigation of different seeding strategies in a genetic planner. In *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*, pages 505–514, London, UK, UK, 2001. Springer-Verlag.

[22] M. Zandieh, S. Fatemi Ghomi, and S. Moattar Husseini. An immune algorithm approach to hybrid flow shops scheduling with sequencedependent setup times. *Applied Mathematics and Computation*, 180(1):111–127, 2006.

Visualización de imágenes médicas de alta resolución mediante una aplicación zero footprint

Emanuel J. C. Arguiñarena
PLADEMA
Facultad de Cs. Exactas,
Universidad Nacional del Centro.
CONICET
earguina@exa.unicen.edu.ar

Mariana del Fresno
PLADEMA
Facultad de Cs. Exactas,
Universidad Nacional del Centro.
CIC-PBA
mdelfres@exa.unicen.edu.ar

Jose M. Massa
INTIA
Facultad de Cs. Exactas,
Universidad Nacional del Centro.
jmassa@exa.unicen.edu.ar

Pedro P. Escobar
INTELyMEC
Facultad de Ingenieria,
Universidad Nacional del Centro.
pescobar@fio.unicen.edu.ar

Martin A. Santiago
IFAS
Facultad de Cs. Exactas,
Universidad Nacional del Centro.
CONICET
msantiag@exa.unicen.edu.ar

Resumen

En la actualidad los dispositivos de captura de modalidades médicas llegan a generar imágenes digitales de gran resolución o estudios con múltiples cortes que pueden ocupar algunos gigabytes. Para la gestión y distribución de las imágenes en formato Digital Imaging and Communication in Medicine (DICOM), las instituciones de salud recurren a la utilización del Picture Archiving and Communication Systems (PACS). Existe un número creciente de aplicaciones que requieren la transmisión remota de las imágenes generadas con fines de diagnóstico. Para lo cual, se han desarrollado Web-PACS que posibilitan la visualización de estos estudios mediante el uso de navegadores Web. La mayoría de los navegadores no permiten la visualización de imágenes DICOM, requiriendo la instalación de alguna aplicación en el cliente Web.

Este trabajo presenta el desarrollo de un visor de imágenes DICOM que funciona con un Web-PACS y posee características zero footprint, es decir que se ejecuta en el navegador Web sin requerir ningún tipo de instalación en el cliente. Además se contempla un esquema de transmisión de imágenes progresivo optimizado, a fin de reducir el tiempo requerido para la visualización y que preserva la calidad diagnóstica. El visor implementado es capaz de ejecutarse en cualquier navegador que soporte HTML5. La performance del algoritmo de trasmisión de imágenes progresivo ha sido evaluada mediante la comparación del Time To Display (TTD) contra la transmisión de imágenes comprimidas con pérdida e imágenes crudas (RAW).

1. Introducción

En los últimos años, la tecnología de imágenes ha avanzado notablemente, tanto en variedad y calidad de las modalidades de captura, como en el desarrollo de métodos para procesamiento y análisis de las mismas. En particular, las imágenes médicas cumplen un rol esencial en el diagnóstico y el monitoreo de enfermedades, así como también en la planificación y simulación de tratamientos e intervenciones [1].

Debido a la creciente disponibilidad de imágenes digitales (como ecografías, mamografías, tomografías computadas, etc.), se ha vuelto necesaria la incorporación de sistemas de gestión de los estudios generados en las instituciones de salud. Hoy en día, está ampliamente aceptado que el uso de PACS es la opción más apropiada para el archivo y distribución de imágenes médicas digitales [2]. Los servidores PACS permiten el almacenamiento de los estudios médicos en formato de archivo DICOM.

La comunicación de imágenes médicas en entornos de PACS comúnmente ocurre dentro de las redes locales de los centros de salud, si bien existe un crecimiento constante del número de aplicaciones que requieren el acceso remoto al PACS. En este contexto, se han desarrollado una variedad de Web-PACS [3], accesibles por medio de un Web browser a través del protocolo Hypertext Transfer Protocol (HTTP).

Un Web-PACS puede lograr un acceso simplificado para el usuario; sin embargo, se deben solventar algunos

inconvenientes para optimizar el acceso remoto y la posterior visualización de estudios. La mayoría de los Web browsers no tienen capacidad para poder visualizar imágenes DICOM, lo que implica que el visor debe ser instalado de alguna manera en el cliente Web. Algunos Web-PACS incluyen un visor embebido, que se descarga con el estudio. Sin embargo, sea cual sea el caso, en general la visualización de la imagen no es posible hasta que el archivo se ha descargado completamente.

Los primeros visores de imágenes médicas web fueron desarrollador en base a Java Applets [4]. La utilización de Java brinda varias ventajas, aunque también tiene algunas falencias. En particular, el intérprete de Java puede funcionar lento en algunos casos y, por otro lado, se debe tener cuidado con la selección de las librerías que se utilizan para desarrollar el Applet a fin de evitar incompatibilidad con la versión instalada en el cliente [4]. También hay algunos visores DICOM basados en tecnología Flash [5]–[7], que permite un deployment mucho más sencillo que en el caso de los Applets de Java ya que la máquina virtual de Flash se encuentra instalada en la mayoría de las computadoras con conexión a internet y no tiene el problema que presenta Java con las librerías externas. Los desarrollos alcanzados en este tipo de visores han logrado aplicaciones robustas y amigables [5], [7]. Sin embargo, la tendencia actual se orienta al desarrollo de aplicaciones para dispositivos móviles, en los que no es posible la instalación de Adobe Flash, al menos en los equipos de Apple [8].

Debido al crecimiento y maduración de las APIs de JavaScript y la adopción completa de HTML5 en la mayoría de los navegadores, se han propuesto algunos visores DICOM basados en HTML5 y código JavaScript que se ejecutan completamente en el navegador Web. Este tipo de aplicaciones, denominadas zero footprint [9], no requieren ningún tipo de instalación en el cliente y pueden correr en navegadores con soporte HTML5[10].

Debido a las continuas mejoras tecnológicas en los dispositivos de adquisición, un aspecto importante a tener en cuenta en el desarrollo de aplicaciones para la visualización de estudios médicos es el tamaño y la cantidad de imágenes que los equipos médicos modernos suelen generar, las cuales pueden llegar a ocupar varios gigabytes. La visualización remota por Web requiere de la transmisión del estudio desde el PACS hasta el cliente Web. Por esta razón, es crítica la implementación de estrategias que permitan ahorrar el uso de ancho de banda y minimizar el tiempo requerido para la transmisión y visualización de las imágenes, preservando su calidad diagnóstica. Un enfoque usual consiste en la utilización de algoritmos de compresión. Las técnicas sin pérdida de información pueden lograr una reducción en el tamaño de la imagen en un factor de 3 contra tasas de compresión mucho mayores que pueden alcanzar las técnicas con pérdida. Sin embargo, en el ámbito médico, los profesionales son reticentes a basar sus diagnósticos en imágenes comprimidas con pérdida [11].

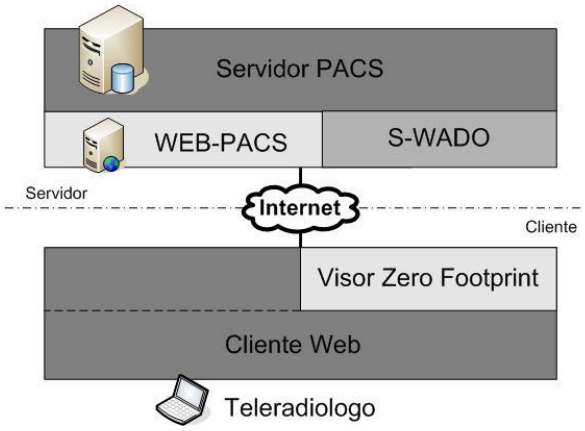


Figura 1. Arquitectura del sistema e interacción de los diferentes componentes

Diferentes estrategias se han implementado en distintos contextos con el fin de optimizar el uso de los recursos y minimizar el tiempo entre el requerimiento y la visualización de la imagen (Time To Display - TTD) [12], [13]. En particular, se han propuesto esquemas basados en transmisión progresiva de imágenes (Progressive Image Transmission - PIT) [14] y también diferentes estrategias que contemplan el manejo de regiones de interés [15].

En este trabajo se presenta un prototipo de un visor de imágenes médicas digitales zero footprint que funciona en conjunto con un Web-PACS. Este visor implementa un algoritmo de transmisión progresiva de imágenes basadas en regiones de interés, que pueden ser seleccionadas de forma interactiva por el usuario, sin comprometer la calidad diagnóstica. El visor es capaz de ejecutarse en cualquier navegador que soporte HTML5 y se descarga cuando se solicita la visualización de la imagen.

El resto del trabajo se organiza de la siguiente manera. En la sección 2 se describe la arquitectura y el desarrollo de la solución. En la sección 3 se presenta la discusión sobre los resultados obtenidos. Finalmente, en la sección 4 se presentan las conclusiones y algunas líneas futuras de trabajo.

2. Elementos del trabajo y metodología

En el diseño de un visor web de imágenes médicas se deben tener en cuenta distintas restricciones asociadas al tipo de aplicación involucrada y su vinculación con las posibilidades tecnológicas. En esta sección, se revisan diferentes aspectos de diseño y las características de la tecnología utilizada para el prototipo desarrollado.

Una característica de la visualización de imágenes de gran resolución (como mamografías, RX, microscopia, etc.) es que, durante su inspección, los profesionales suelen revisar inicialmente el estudio en forma general, para luego centrarse en algunas regiones específicas. Por esto, sólo algunos sectores de la imagen pueden ser requeridos en toda su resolución. Durante este proceso, al construir un visor Web, se presenta el desafío de transmitir las imágenes con calidad diagnóstica, minimizando el TTD [16]. Teniendo esto en consideración, se planteó un algoritmo para realizar PIT, el cual utiliza la información que el usuario genera a partir de los movimientos dentro de la imagen, para definir las regiones que se deben transmitir con mayor calidad. Por esta razón, se ha denominado al algoritmo de visualización IOPIT (por Interactive and Optimized PIT) [17].

2.1. Arquitectura del Sistema

El sistema desarrollado está compuesto por dos aplicaciones: un Visor zero footprint (VZFP) y un Servidor S-WADO. El VZFP permite visualizar en el host remoto los estudios DICOM almacenados en el servidor PACS a través del web browser, mientras que el S-WADO tiene acceso directo a los archivos DICOM almacenados en el servidor PACS en respuesta a los requerimientos del cliente. La comunicación entre el cliente y el servidor se realiza mediante el protocolo HTTP, de acuerdo a la especificación WADO (Web Access DICOM Object) del estándar DICOM [18]. En este caso se utilizó el servidor PACS open source Dcm4Chee [19], pero podría adaptarse a otra implementación. En la Figura 1 se muestra el esquema general de la solución implementada.

Mediante el Web-PACS el cliente puede listar y buscar los estudios almacenados en el Servidor PACS, y eventualmente, seleccionar y visualizar la imagen mediante el visor VZFP.

2.2. IOPIT

Considerando que muchas de las modalidades médicas poseen alto nivel de resolución, la estrategia propuesta para la transmisión de los datos considera la descarga, bajo demanda, de aquellas porciones de la imagen que se van requiriendo en el área de visualización y con una resolución no mayor que la de la pantalla del cliente. Para esto, el algoritmo divide virtualmente el área de visualización en bloques de tamaño fijo denominados Tiles, cuya cantidad es calculada en base al área de visualización y además contempla la discretización de la imagen en imágenes de menor tamaño formando una pirámide donde en los niveles superiores se encuentran las imágenes de menor tamaño (Figura 2). Esta división

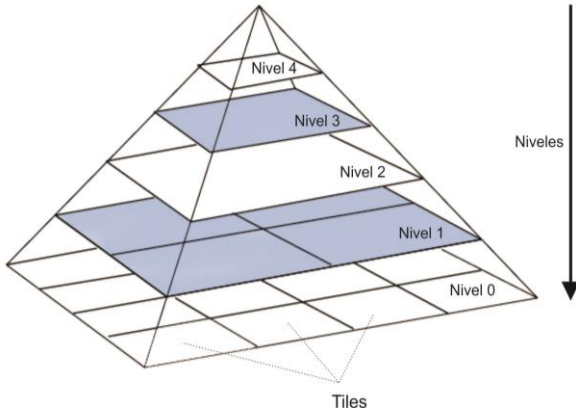


Figura 2. División virtual de la imagen por tiles y por niveles

virtual es conocida como Pyramid Tiling [20] y es la estructura base en la cual el algoritmo determina la información que debe transmitir.

A medida que el usuario explora la imagen mediante las herramientas interactivas de zoom y pan (mover), va definiendo aquellas partes de la imagen que desea ver con mejor definición. De esta forma, el algoritmo determina los bloques requeridos para mostrar la imagen buscada.

Cuando el usuario solicita visualizar una imagen por primera vez, el algoritmo calcula la cantidad de niveles necesarios para mostrar la imagen de forma eficiente en base al tamaño de la misma, el área de visualización del cliente remoto y el tamaño de los tiles. La visualización del estudio comienza con la imagen correspondiente al nivel más alto de la pirámide y conforme el usuario utiliza el zoom, se mueve en los diferentes niveles de la pirámide. Cuando se define una nueva región de visualización, el algoritmo calcula los tiles requeridos para mostrar el área de visualización de forma completa y los solicita al servidor.

La organización en niveles discretos se realiza para poder reutilizar las porciones de la imagen descargada, por lo que si el usuario selecciona un nivel de zoom intermedio, se considerará el nivel de zoom inmediatamente más profundo.

En la siguiente sección se describirá el VZFP que implementa el algoritmo IOPIT.

2.3. Visor zero footprint

El VZFP es un visor de imágenes DICOM que provee un entorno de visualización en el cliente Web, el cual está compuesto por un área de visualización simple e incluye diferentes opciones para manejo de las imágenes (Figura 3). El VZFP fue desarrollado en Java utilizando



Figura 3. Vista inicial (arriba), vista con zoom de área (abajo)

la librería Google Web Toolkit (GWT) [21], la cual luego de una compilación produce una aplicación JavaScript que puede ser ejecutada en cualquier navegador con soporte de HTML5. El VZFP es descargado desde el servidor y ejecutado en el navegador de forma transparente al usuario cuando este selecciona el estudio que necesita visualizar del Web-PACS.

En este prototipo se han implementado las herramientas básicas de zoom-in, zoom-out y pam (mover). Cuando se realiza una operación de zoom o pan en el área de visualización, el algoritmo IOPIT determina las peticiones de tiles necesarias, que antes de ser enviados al servidor S-WADO, son ordenadas para solicitar primero las de mayor relevancia espacial. Una vez en el cliente, los tiles recuperados se utilizan para construir la vista de la imagen en el área de visualización. El VZFP se encarga de realizar las interpolaciones necesarias a fin de mostrar las imágenes al usuario correctamente. Todas las peticiones de tiles realizadas por el cliente, acordes a la especificación WADO.

A fin de optimizar la performance del cliente, solo permanecen en memoria los bloques correspondientes al

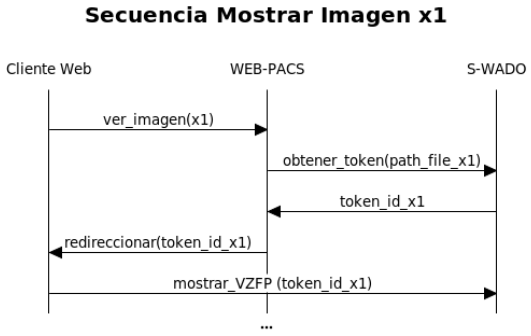


Figura 4.Interaccoin entre Cliente, Web-PACS y S-WADO

área de visualización. De esta forma, el visor siempre utiliza el mínimo de memoria requerida y reduce los cálculos necesarios para en el manejo de las imágenes. En caso de requerir nuevamente alguna de las imágenes descartadas, estas se pueden recuperar de la cache del navegador Web sin necesidad de hacer nuevamente la petición al servidor.

2.4. Servidor S-WADO

El S-WADO es un servidor HTTP que ofrece un esquema para acceso seguro a las imágenes almacenadas en el PACS e implementa la especificación WADO para responder peticiones de imágenes y sub-imágenes que realizan los clientes remotos.

El esquema de seguridad se basa en la utilización de un identificador temporal llamado token que es creado y destruido por el Web-PACS. El token da acceso a cualquier cliente que solicite información durante su tiempo de vida. Como se muestra en la Figura 4, ante la solicitud de un estudio por parte de un usuario remoto previamente autenticado, el servidor Web-PACS solicita al S-WADO una autorización para ver un estudio específico que es respondida con un token. Luego el Web-PACS conecta al cliente Web con el S-WADO mediante una redirección en la que especifica el token asignado. A partir de ese momento el cliente web descarga y ejecuta el VZFP de forma automática donde se visualiza el estudio. Cuando el usuario finaliza la sesión con el Web-PACS, este elimina todos los tokens asignados evitando todo acceso posterior a las imágenes. Esta estrategia basada en tokens para identificación temporaria permite proveer un nivel aceptable de seguridad que puede ser mejorado implementando las reglas de seguridad HIPAA en el Web-PACS [22].

Como se especificó arriba, una vez que el algoritmo IOPIT determina los tiles necesarios para mostrar el

https://SWADO/request?requestType=WADO&studyUID=1.2.250.1.59.40211.12345678.678910&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2&contentType=image/jpeg&columns=400&rows=300&region=0.3,0.4,0.5,0.5

Figura 5. Ejemplo de URL WADO

estudio en el VZFP, estos son solicitados al S-WADO utilizando la especificación WADO, la misma es una Uniform Resource Locator (URL) en la cual se debe especificar el identificador de estudio (studyUID), el identificador de serie (seriesUID), el identificador de objeto (objectUID), el número de corte (frameNumber), la región de la imagen definida desde x1,y1 hasta x2,y2 donde los valores son reales entre 0 y 1 (region), la cantidad de columnas (columns) y filas (rows) de la imagen resultante y el tipo de objeto de retorno (contentType) como se puede ver en la Figura 5.

Para mejorar la entrega de tiles por parte del S-WADO, se utilizaron algunas optimizaciones. En particular se crearon tres niveles de cache dentro del servidor, para lo que se utilizó la librería ehcache [23], se utilizaron threads para responder eficientemente a peticiones simultáneas y se implementó un pool de objetos que permite la reutilización de objetos creados en memoria y reducir los tiempos de creación y eliminación de los mismos [21].

S-WADO ha sido desarrollado utilizando Java 2 Platform Enterprise Edition (J2EE), el Servlet Container Jetty [24] y el toolkit dcm4che2 versión 2.0.18 para interactuar con los archivos DICOM [25]. Se reutilizó parte del código del servidor "DCM-Ar Server" desarrollado en trabajos previos [5].

3. Resultados y discusión

Una de las tendencias en teleradiología es la visualización de imágenes desde cualquier dispositivo móvil o desde cualquier computadora. Esto requiere el desarrollo de aplicaciones que funcionen en diferentes sistemas y no requieran la instalación de aplicaciones específicas, complementos o máquinas virtuales. Para evaluar la factibilidad de este tipo de aplicaciones se han realizado pruebas del VZFP en una Tablet Galaxy Tab 2 con Android 4.1 y en un iPad 2 con iOs, constatando el funcionamiento correcto de la aplicación en ambos dispositivos. Estas pruebas se realizaron con una versión básica del VZFP que no incorpora el manejo de gestos táctiles.

Otro de los objetivos del trabajo ha sido mejorar la experiencia del teleradiólogo mediante la minimización de los tiempos de latencia al visualizar imágenes médicas a través de redes públicas. A fin de determinar la calidad del algoritmo IOPIT en la reducción del TTD, se hicieron pruebas utilizando diferentes estrategias de transmisión. Inicialmente se transmitieron desde el Web-PACS imágenes de alta resolución sin comprimir (RAW), en el segundo caso se transmitieron las mismas imágenes comprimidas sin pérdida, en formato PNG, y finalmente las imágenes fueron transmitidas utilizando el algoritmo IOPIT. En todos los casos, la imagen fue visualizada en el visor y se obtuvieron las siguientes tasas relativas de TTD: 14:5:1 (para las opciones RAW, PNG y IOPIT, respectivamente). Las pruebas se realizaron con imágenes de radiología digital con un tamaño entre 10MB y 30MB.

Como fue descripto anteriormente, el VZFP solicita al servidor S-WADO tiles con partes de la imagen original en diferentes resoluciones. En todos los casos, el algoritmo IOPIT permite al teleradiólogo acceder a las imágenes con la calidad original sin incrementar el TTD.

Este trabajo muestra la factibilidad de la utilización de aplicaciones zero footprint para visualización progresiva de imágenes médicas en distintos dispositivos.

4. Conclusiones y trabajos futuros

En este trabajo se presentó el desarrollo de un prototipo para la visualización remota de imágenes médicas implementando un algoritmo de visualización progresiva de imágenes. El sistema consiste en un cliente web que interactúa con el usuario para mostrar las imágenes de estudios remotos y en un servidor de imágenes que implementa la especificación WADO definida en el estándar DICOM. El visor cliente, que se descarga al momento de visualizar las imágenes, es una aplicación zero footprint que no requiere ningún tipo de instalación. Este visor ha sido probado tanto en computadoras personales como en dispositivos móviles basados en iOs y Android. El algoritmo de transmisión progresiva implementa una estrategia interactiva basada en una división de la imagen en bloques (tiles), con un esquema piramidal. Estas características lo hacen especialmente apropiado para la visualización de estudios médicos de gran tamaño y para aplicaciones de telemedica. Este esquema permite reducir el tiempo requerido para la visualización (TTD), sin comprometer la calidad diagnóstica de las imágenes.

Como trabajo futuro se prevé la migración del prototipo a una implementación basada en las especificaciones del estándar Jpeg2000 para visualización progresiva de imágenes.

5. Agradecimientos

Este trabajo es financiado parcialmente por la Agencia Nacional de Promoción Científica y Tecnológica (PICT 2010-1287).

6. References

[1] W. J. Chimiak, "The digital radiology environment," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 7, pp. 1133–1144, 1992.

[2] J. R. Pilling, "Picture archiving and communication systems: the users' view," *Br. J. Radiol.*, vol. 76, no. 908, pp. 519–524, Aug. 2003.

[3] J. Zhang, J. Sun, and J. N. Stahl, "PACS and Web-based image distribution and display.," *Comput. Med. Imaging Graph.*, vol. 27, no. 2–3, pp. 197–206, Jan. 2003.

[4] J. Fernàndez-Bayó, O. Barbero, C. Rubies, M. Sentís, and L. Donoso, "Distributing medical images with internet technologies: a DICOM web server and a DICOM java viewer.," *Radiographics*, vol. 20, no. 2, pp. 581–90.

[5] E. J. C. Arguinarena, J. E. Macchi, P. P. Escobar, M. Del Fresno, J. M. Massa, and M. A. Santiago, "Dcm-ar: a fast flash-based Web-PACS viewer for displaying large DICOM images.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2010, pp. 3463–6, Jan. 2010.

[6] "MedDream." [Online]. Available: http://www.softneta.com.

[7] "Oviyam." [Online]. Available: http://oviyam.raster.in/.

[8] J. Honeyman-Buck, "The Magical iPad," *J. Digit. Imaging*, vol. 23, no. 4, pp. 514–515, Jun. 2010.

[9] D. A. Gutman, W. D. Dunn, J. Cobb, R. M. Stoner, J. Kalpathy-Cramer, and B. Erickson, "Web based tools for visualizing imaging data and development of XNATView, a zero footprint image viewer.," *Front. Neuroinform.*, vol. 8, p. 53, Jan. 2014.

[10] S. Fulton and J. Fulton, *HTML5 Canvas*. O'Reilly Media, 2011, p. 654.

[11] P. Suapang, K. Dejhan, and S. Yimmun, "Medical image compression and DICOM-format image archive," pp. 1945–1949.

[12] S. Hu, V. Tao, and H. Wu, "Use image streaming technologies to present high resolution images on the internet," *... Arch. Photogramm. Remote Sens. ...*, 2004.

[13] V. Tuominen and J. Isola, "Linking whole-slide microscope images with DICOM by using JPEG2000 interactive protocol," *J. Digit. Imaging*, 2010.

[14] J. Villanueva-Oller, R. J. Villanueva, and S. Díez, "CASANDRA: a prototype implementation of a system of network progressive transmission of medical digital images.," *Comput. Methods Programs Biomed.*, vol. 85, no. 2, pp. 152–64, Feb. 2007.

[15] I. Baeza, J. -a. Verdoy, J. Villanueva-Oller, and R.-J. Villanueva, "ROI-based procedures for progressive transmission of digital images: A comparison," *Math. Comput. Model.*, vol. 50, no. 5–6, pp. 849–859, Sep. 2009.

[16] M. Pietsch, a Schlaefke, T. J. Vogl, and B. Bergh, "Development and evaluation of different methods to assess download and display time of image web systems.," *J. Digit. Imaging*, vol. 19, no. 4, pp. 336–45, Dec. 2006.

[17] P. Escobar, M. Santiago, M. Fresno, J. Massa, and E. Arguiñarena, "IOPIT: A FAST WAY FOR LARGE DICOM FILES TRANSMISSION OVER INTERNET," in *The International eHealth, Telemedicine and Health ICT Forum for Educational, Networking and Business*, 2011, pp. 513–516.

[18] F. Text, "Digital Imaging and Communications in Medicine (DICOM) Supplement 148 : Web Access to DICOM Persistent Objects by Means of Web Services Extension of the Retrieve Service (WADO Web Service)," no. August 2011, 2008.

[19] M. Warnock, C. Toland, and D. Evans, "Benefits of using the DCM4CHE DICOM archive," *J. Digit. ...*, no. October, pp. 125–129, 2007.

[20] E. E. Hemayed, E. M. Farghal, and S. I. Shaheen, "Pyramid-based multiresolution tiling for interactive viewing of remote sensing images," in *2010 International Computer Engineering Conference (ICENCO)*, 2010, pp. 128–133.

[21] R. Dewsbury, *Google Web Toolkit Applications*. Addison-Wesley Professional., 2007, p. 608.

[22] F. Cao, H. K. Huang, and X. Q. Zhou, "Medical image security in a HIPAA mandated PACS environment.," *Comput. Med. Imaging Graph.*, vol. 27, no. 2–3, pp. 185–96, Jan. 2003.

[23] I. Terracotta, "Ehcache an open source, standards-based cache for boosting performance," 2013. [Online]. Available: http://ehcache.org/.

[24] Eclipse Fundation, "Jetty a Java web servlet," 2013. [Online]. Available: http://www.eclipse.org/jetty/.

[25] Dcm4che, "Dcm4che, an open source collection of applications for healthcare IT," 2013. [Online]. Available: http://www.dcm4che.org/.

Automatización del modelo electoral de participación ciudadana

Istvan, Romina Mariel; Antonini Sergio Andrés

Universidad Tecnológica Nacional, Facultad Regional La Plata
Laboratorio de Ingeniería en Sistemas de Información, LINES
Av. 60 s/n° esquina 124, CP 1900, La Plata, Buenos Aires, Argentina.
ristvan@frlp.utn.edu.ar; antonini@frlp.utn.edu.ar

Abstract
En este trabajo se plantea un sistema de votación que haciendo uso de la tecnología existente, automatice la gestión del proceso de votación, respondiendo a las necesidades legislativas y sociales vigentes.
Luego de explorar los modelos y experiencias realizadas a nivel mundial y nacional, se obtienen las ventajas y desventajas que estos sistemas ofrecen [1]; y se propone, a partir de ellas, un sistema que automatiza el modelo electoral de participación ciudadana con distribución de electores y boleta única.
Se explicitan todos y cada uno de los pasos del proceso de votación, en los que se incluyen diseño y confección del padrón electoral, identificación del votante, pasos del proceso de votación, emisión, cierre y recuento de votos, transmisión y totalización de resultados y soportes formales para eventuales auditorías. Asimismo se presenta un diseño preliminar de las boletas únicas en los que se explicitan los datos mínimos requeridos por la Junta Electoral.
Se espera que los beneficios principales que provea minimicen los tiempos de espera para los votantes al contar con un padrón único por centro, y maximicen los beneficios de mantener la boleta como soporte formal. De esta manera, se intenta automatizar una gran parte del proceso electoral como signo de modernidad y seguridad, respondiendo al reto tecnológico y manteniendo, a su vez, la boleta papel como soporte real para la comprobación de resultados electorales.
Se posibilita su aplicación a múltiples usos como ser encuestas y votaciones en general, además de votaciones municipales, provinciales, y nacionales.

Palabras Clave
Voto electrónico, Votación electrónica, Boleta única, Escáneres de alta velocidad

Introducción

Hoy en día, tecnología existente presenta numerosos beneficios para llevar a cabo el proceso de votación, de modo tal que se aseguren los principios sobre los que se apoya el sistema electoral argentino [2].
El objetivo de este trabajo es analizar y diseñar un modelo factible de votación que haciendo uso de la tecnología existente permita automatizar una gran parte del proceso electoral como signo de modernidad y seguridad, respondiendo al reto tecnológico, social y legislativo.
La metodología a utilizar parte de los modelos y experiencias realizadas a nivel mundial y nacional y particularmente de los sistemas que utilizan boleta única y escáneres de alta velocidad para la lectura de votos.
Los componentes principales que fueron investigados son: confección del padrón, autenticación del votante, selección de candidatos, proceso de votación, emisión cierre y recuento de votos, como así también velocidad en el acto eleccionario, rapidez en el recuento de los votos, la utilización de escáneres de alta velocidad para el conteo de los votos y la posibilidad de transmisión de los resultados a un centro totalizador y auditorías [3].
El presente artículo se organiza de la siguiente manera. En la sección dos, se realiza una breve descripción de los antecedentes y experiencias del equipo de investigación. En la sección tres, se

presentan los sistemas de votación existentes. En la sección cuatro, se plantea la propuesta elegida, en la que se incluyen diseño del padrón único por centro de votación, boceto de las boletas únicas y pasos a seguir en el proceso de votación. Luego de esto, se analizan las diferencias y ventajas del modelo propuesto con el sistema de votación tradicional en Argentina. En la sección cinco, se presentan los resultados esperados. Y por último, en la sección seis, se expone la conclusión y trabajos futuros.

Elementos del Trabajo y metodología

Luego del estudio de los sistemas existentes y análisis de experiencias llevadas a cabo a nivel mundial, se propone un sistema de votación que automatiza gran parte del proceso electoral. El modelo propone básicamente la distribución de electores y boleta única, contemplando padrón único por centro, varios boxes asociados a las mesas de votación y la utilización de escaners de alta velocidad para el recuento de votos.

Padrón electoral

El padrón único electrónico contiene todos los electores por centro de votación, en él se registra nombres, apellido, tipo y número de documento, ejemplar (original/duplicado/triplicado …), mesa originaria, estado de votación y firma.

Las mesas originarias determinan la mesa en la cual el elector debe realizar su voto en caso de que ocurra algún incidente durante el acto eleccionario que impida la correcta actualización de los padrones por centro, de manera de poder continuar con el proceso eleccionario. Para tal fin, y en caso de ocurrir una contingencia, se imprimirá desde el servidor central, una copia para cada mesa de votación con los datos de los electores de su mesa originaria que ya realizaron su voto, para su chequeo manual, de forma no permitir nuevamente la votación a electores que ya emitieron su voto. Cada votante en estos casos, sólo podrá realizar la votación acercándose a su mesa originaria.

Los estados de votación que se preveen son:

- "HABILITADO": para identificar a los electores que aún no emitieron sus votos;
- "VOTANDO": si el elector se encuentra en el proceso de votación;
- "VOTÓ": luego de que el votante haya emitido su voto

Boleta única

Los datos básicos a ser incluidos en el diseño de las boletas únicas, según requerimientos de la Junta Electoral para categorías de elecciones de fórmula, tales como Presidente – Vicepresidente y Gobernador / Vicegobernador son:

- ✓ Distrito, Circuito, Sección y Mesa de Votación
- ✓ Partido y escudo de la fórmula
- ✓ Número de Lista
- ✓ Foto y Nombre y Apellido de los candidatos de fórmula
- ✓ Sección adicional para Voto en Blanco
- ✓ Firma del Presidente de Mesa

Sin embargo, estos datos se preveen configurables para cada elección, posibilitando su adaptación a los diferentes requerimientos que puedan surgir.

Un boceto preliminar de una boleta única, para una categoría de Gobernador/Vicegobernador es el siguiente:

DISTRITO:
CIRCUITO:
SECCIÓN:
MESA DE VOTACION:

ELECCIONES DE GOBERNADOR

PARTIDO *Escudo*	LISTA *Nro.*	CHECK	GOBERNADOR *Imagen*	GOBERNADOR *Apellido y Nombre*	VICEGOBERNADOR *Imagen*	VICEGOBERNADOR *Apellido y Nombre*
		☐				
		☐				
		☐				
		☐				
		☐				
		...				

☐ VOTO EN BLANCO

FIRMA DEL PRESIDENTE DE MESA:

Los datos básicos para categorías de listas como DIPUTADOS y SENADORES,

también configurables para cada elección, son:

- ✓ Distrito, Circuito, Sección y Mesa de Votación
- ✓ Partido y escudo de la fórmula
- ✓ Número de Lista
- ✓ Lista de candidatos
- ✓ Sección adicional para Voto en Blanco
- ✓ Firma del Presidente de Mesa

DISTRITO: CIRCUITO: SECCIÓN: MESA DE VOTACION:			
ELECCIONES DE DIPUTADOS			
PARTIDO *Escudo*	LISTA *Nro.*	CHECK	LISTADO DE CANDIDATOS
		☐	
		☐	
		☐	
		☐	
		☐	
		...	
		☐	VOTO EN BLANCO
FIRMA DEL PRESIDENTE DE MESA:			

Proceso de votación

El proceso de votación queda siguiendo este modelo resumido en los siguientes pasos:

1. Identificación del elector
   - ✓ El elector elije la mesa de votación en la cual desea emitir su voto.
   - ✓ Se presenta ante el presidente de mesa, el cual verifica los datos del votante, chequeando los mismos contra el padrón electrónico.
   - ✓ Una vez hallado y verificado, realiza una marca que indica que el elector está por empezar el proceso de votación, pasando su estado de "HABILITADO" a "VOTANDO", en esta instancia el elector queda momentáneamente deshabilitado para presentarse en otras mesas de votación.

2. Votación
   - ✓ El presidente de mesa habilita uno de los boxes disponibles (ya que, cada mesa tiene varios habilitados) y entrega una boleta única por categoría al votante.
   - ✓ El elector marca el voto sobre sus boletas únicas. Una vez que completa las boletas con las opciones deseadas.

3. Emisión de voto
   - ✓ Luego de esto, retorna a la mesa de votación, deposita cada boleta en la urna correpondiente a cada categoría y firma digitalmente. A partir de aquí, el presidente de mesa efectúa una nueva marca en el padrón, pasando de "VOTANDO" a "VOTÓ", indicando que el proceso de votación de ese elector particular, ha culminado.
   - ✓ Dicha información se actualiza inmediatamente de manera centralizada, de forma que una vez que emitió su voto en una mesa, no podrá realizar nuevamente una votación, ya que el nuevo estado "VOTÓ" se verá reflejado en los padrones de todas las mesas.
   - ✓ El hecho de que existan varias marcas, es para evitar los siguientes conflictos, como ser que el elector salga del box y no llegue a colocar el voto en la urna. En ese caso, el proceso de votación quedaría rechazado, haciendo que el elector vuelva a quedar habilitado para votar.
   - ✓ El presidente de mesa hace entrega del comprobante correspondiente junto con la devolución del DNI.

4. Cierre y Recuento de votos
   - ✓ Cada centro de votación contará con escaners de alta velocidad que realizarán la contabilización de los votos una vez cerrada la votación de cada mesa.
   - ✓ Para ello, los presidentes de cada mesa, una vez abiertos los sobres, introducirán las boletas a fin de que puedan ser contabilizadas por los escaners.
   - ✓ En caso de que alguna boleta no pueda ser correctamente leida, la

misma será retornada en una bandeja a diferente, posibilitando la interpretación del presidente de mesa.
- ✓ Los resultados de cada mesa de votación serán almacenados en un dispositivo extraíble.

5. Transmisión y Totalización
   - ✓ Una vez cerrada la votación, de manera automática se dispone de los resultados que son transferidos de manera encriptada e inmediata al al Centro de Cómputos de la Junta Electoral para el registro y totalización de los resultados finales.

6. Auditorias
   - ✓ Las boletas se mantienen como soporte formal para futuras auditorías.
   - ✓ Asimismo, la información que resulte de la votación será almacenada en un dispositivo extraíble que será remitido a la Junta Electoral para eventuales auditorias, junto a las boletas papel.

Beneficios del modelo propuesto

Este sistema de votación reúne la mayor cantidad de ventajas y beneficios, adaptado a las leyes electorales y decretos reglamentarios existentes en nuestro país [5] [6] [7].

Se espera que el sistema propuesto minimice los tiempos de espera al elector, al ofrecerle la posibilidad de elegir la mesa de votación con menor cantidad de personas.

El presidente de cada mesa, asimismo dispone de varios boxes habilitados para la votación, de manera de poder asignar dinámicamente uno a cada elector. Se espera de esta manera que varios electores efectúen su voto en simultáneo, reduciendo aún más la espera para el acceso a la urna.

Requiere técnicamente un único padrón por centro de votación y mesas conectadas en red localmente.

Se presentan imágenes donde se contrapone el sistema de voto tradicional con el sistema propuesto:





Figura 1: Sistema de Votación Tradicional Figura 2: Sistema de Votación Propuesto

En la Figura 1 se muestra el Sistema de Votación Tradicional. El sistema actual asigna al elector una mesa predeterminada en la cual votar. De esta manera, ocurre que en determinadas mesas se encuentran grandes cantidades de electores mientras que otras permanecen prácticamente vacías por importantes períodos de tiempo.

En la Figura. 2 se muestra el Sistema de Votación Propuesto, el cual, a diferencia del Sistema Tradicional, permite que en cada centro de votación exista un padrón electoral único, posibilitando al elector elegir la mesa con la menor espera posible, logrando que las colas se autorregulen.

A continuación, se presentan las diferencias más notorias entre el sistema actual y el sistema propuesto.

Sistema de votación actual
Existencia de boletas papel por para cada partido. Ocasiona eventualmente problemas de faltantes en los cuartos oscuros.
Único cuarto oscuro asociado a una mesa. Único votante por mesa en un instante de tiempo determinado.
Elector asignado a una única de votación.
Contabilización de votos manual.
Unico padrón por mesa.
Cada mesa cuenta con información que no es compartida al resto de las mesas.
Sistema de votación propuesto
Una boleta única en papel que contiene a todos los candidatos.
Varios boxes asociados a una mesa. Posibilidad de votaciones en simultáneo.
Posibilidad elección de la mesa por parte del elector.
Escaners de alta velocidad para el conteo de votos.
Padrón electrónico único por establecimiento.
Todas las mesas contienen la misma información y toda actualización será vista por las demás mesas de manera automática.

Tabla 1: Diferencias sistemas de votación

Resultados

Se espera que el sistema propuesto minimice los tiempos de espera para los votantes, al contar con un padrón único por centro, lo que permitirá a cada elector elegir la mesa en la cual emitir su voto. Los tiempos de recuento también se verán disminuidos, debido a la existencia de escaners de alta velocidad que suplirán ese proceso.

El presidente de cada mesa, asimismo dispondrá de varios boxes habilitados para la votación, de manera de poder asignar dinámicamente uno a cada elector. Se espera de esta manera que varios electores efectúen su voto en simultáneo, reduciendo aún más la espera para el acceso a la urna.

La utilización de la boleta única maximizará los beneficios de mantener un soporte formal y real para eventuales auditorias, facilitando la aceptación del modelo propuesto a nivel social.

Discusión

Se espera que el sistema de votación propuesto con distribución de electores, boleta y padrón único y la utilización de escáneres de alta velocidad, permita automatizar y agilizar una gran parte del proceso electoral, como signo de modernidad y seguridad respondiendo al reto tecnológico, pero manteniendo la boleta, como único soporte .

Los tiempos se espera serán minimizados, al ofrecerle la posibilidad de elegir la mesa de votación con menor cantidad de personas y la disposición de varios boxes asociados a cada mesa.

Se espera que el proyecto permita aplicarse a múltiples usos, como ser votaciones municipales, provinciales y/o nacionales u otros tipos de votaciones, incluyendo las encuestas.

Conclusión

El trabajo explora los sistemas de votación existente a nivel mundial y nacional. De estos modelos se tomaron casos de estudio, que fueron implementados en Argentina (elecciones municipales y provinciales en la provincia de Buenos Aires) [8] [9], como así también a nivel internacional, en países como España (urna electrónica Demotek) [10], Estados Unidos (sistema e-slate de la empresa Hart Inter Civic) [11] y Brasil (pantallas táctiles desarrolladas por la empresa Diebold) [12].

De ellos se evaluaron, las ventajas y desventajas que ofrecen respecto a cuatro puntos esenciales: autenticación del votante, selección de candidatos, proceso de votación y recuento de votos. Como así también la velocidad en el acto eleccionario, la eliminación de las boletas papel, la rapidez en el recuento de los votos y la posibilidad de transmisión de los resultados a un centro totalizador.

Junto con este se define una propuesta de sistema de votación con distribución de electores y con boleta única, que permita automatizar una gran parte del proceso electoral como signo de modernidad y seguridad respondiendo al reto tecnológico, pero manteniendo la boleta, como único soporte formal y real para comprobar los resultados electorales.

Los principales beneficios que se esperan obtener con el sistema propuesto están asociados a la transparencia en la obtención de los resultados, la minimización de los tiempos de espera para los votantes, la rapidez en la votación del elector y recuento de votos.

Asimismo, se prevé la posibilidad de que el proyecto pueda aplicarse a múltiples usos como ser encuestas y votaciones en general, además de votaciones municipales, provinciales, y nacionales.

Se plantea como trabajo futuro el desarrollo e implementación del sistema propuesto.

Referencias (Times New Roman, 10, negrita).

[1] ALEUY Mahmud, La votación electrónica, 2007.

[2] TULA María Inés, ed. Voto electrónico: entre votos y máquinas, las nuevas tecnologías en los procesos electorales. Ariel, 2005.

[3] RIAL, Juan. "Posibilidades y límites del voto electrónico." Fernando Tuesta Soldevilla, 2004.

[4] BARRIENTOS DEL MONTE Fernando, El voto electrónico: contexto, experiencias y dilemas, 2006.

[5] Ley Electoral 26.571: Ley de democratización de la representación política, la transparencia y la equidad electoral. (Modificaciones a las Leyes 23.298, 26.215 y 19.945 - Código Electoral).

[6] Código Nacional Electoral. Ley 19.945 y modificatorias.

[7] DECRETOS REGLAMENTARIOS LEY 26.571. CÓDIGO ELECTORAL NACIONAL: Decreto 935/2010.

[8] http://www.sgp.gov.ar/contenidos/inap/investigacion docs/ponencias_x_redmuni/Tecnologias/Mesa%202/ AntoniniRedMuni2009.pdf

[9] http://www.voto-electronico.org/reportes/argentina.pdf

[10] http://www.infoelectoral.mir.es/Modernizacion/visit a_MAE_infoelectoral.htm

[11] https://www.verifiedvoting.org/

[12] http://www.votoelectronico.org/reportes/brasil.pdf

Una Aproximación al Estado del Desarrollo de Tecnologías Móviles en la Región de La Plata

Mirta del Carmen Peñalva
Departamento de Ingeniería en Sistemas de Información
Facultad Regional La Plata, Universidad Tecnológica Nacional
Av.60 esq. 124 s/n - CP 1900 - La Plata - Buenos Aires
E-mail:mirpenalva@hotmail.com

Abstract

El desarrollo de las comunicaciones globales y la tecnología inalámbrica sumadas al masivo uso de dispositivos móviles han impactado en la industria del software dando lugar a un capítulo especializado en el desarrollo móvil. La diversidad de aspectos a tener en cuenta en una solución móvil, la ha convertido en una entidad compleja. Un usuario en movimiento, e introduciendo la tecnología en todos los ámbitos de su vida, exige nuevos servicios a través de los cuales las tecnologías móviles sean adaptativas a todas estas capas entrelazadas del contexto. Dado el incremento a nivel mundial de estos desarrollos nos planteamos investigar el estado del arte del desarrollo regional de tecnologías móviles en el municipio de La Plata y áreas de influencia, con el fin de poder establecer puntos de encuentro con el contexto nacional y mundial, como así también detectar nuevas oportunidades de servicios locales.

En la primera sección se aborda un marco teórico de referencia para detallar en qué consiste una aplicación móvil atenta al contexto, luego se presenta la investigación en el área local con el análisis de resultados, posteriormente se presentan ejemplos de soluciones de tecnología móvil producidas en la región y para finalizar las conclusiones generales y visiones a futuro.

Palabras claves: Tecnología móvil, posicionamiento, context-awareness, desarrollo móvil regional.

1. Introducción

El desarrollo de las comunicaciones globales, la tecnología inalámbrica, la diversidad y uso masivo de dispositivos móviles, sumados a la capacidad de procesamiento, almacenamiento local y en la nube, han revolucionado el vínculo persona-tecnología, extendiendo sus servicios a ámbitos y situaciones no previstas en paradigmas anteriores. Se ha ido trasladando el uso de e-aplicaciones a las m-aplicaciones, siendo estas últimas, versiones de las primeras adaptadas a la tecnología móvil, aunque no todas adscriben al posicionamiento.

Con el advenimiento de la movilidad, la sensibilidad al contexto imprime una complejidad mayor a aquellas aplicaciones que la consideran.

En etapas iniciales de la industria, los usuarios usaban aplicaciones de facto, paulatinamente se ha revertido esta situación, hoy en día las aplicaciones deben tener la capacidad de ser elegidas. Conforme el usuario ha co-evolucionado con las TICs, se ha jerarquizado su rol, demanda funcionalidad (RF)[1] revestida de características no funcionales (RNF)[2] , que más allá de enriquecerlas le dan sentido. Se ha evidenciado de esta forma, lo expresado en [1] la necesidad de que los RF y RNF se desarrollen conjuntamente, es así como los RNF cobran un lugar central para que la aplicación brinde funcionalidad adecuadamente a un usuario que exige calidad de servicios.

Actualmente la movilidad permite saber dónde está el usuario, su orientación, altura, etc. y poder reaccionar y brindar servicios asociados a cambios en su ubicación, entrelazándolos con preferencias de su perfil, información de su red de vínculos personales y su entorno social.

Otro aspecto destacable es la jerarquización de la visualidad y las capacidades de procesamiento de las imágenes, que han hecho que la navegación y selección en las aplicaciones, estén dirigidas por imágenes habiendo desplazado al texto. Por cierto, cada vez se acoplan más servicios de diferentes proveedores, la integración y calidad (QoS)[3] de los mismos es clave en el desempeño de estas aplicaciones.

Los sectores que primero han recibido aportes de esta nueva generación de aplicaciones son el turismo, (bookmarking), los juegos interactivos, logística de

[1] RF Requerimientos Funcionales
[2] RNF Requerimientos no Funcionales
[3] QoS Quality of Service

mercaderías, comercio, y prontamente van incorporándose otras: explotación de hidrocarburos y actividades a cielo abierto: ganadería y agricultura, meteorología y procesamiento de imágenes para información estratégica. No habiéndose explotado aún segmentos como podrían ser usuarios con una afinidad específica ubicada regionalmente, los cuales interactúan a través de redes sociales pero no con aplicaciones móviles, como podría ser la cartelera de espectáculos (con calificación de la población), pérdida y ubicación de mascotas, cortes o interrupciones del tránsito, paradero e identificación de personas en situación de calle, por citar algunas. En todos los casos, la obtención de conocimiento de todo tipo (individual o grupal) surgido de las experiencias de usuarios es fundamental para proveer retroalimentación mejorando así la precisión y adecuación de los servicios en tiempo real.

Recordemos que la celeridad y vigencia de los datos son una parte crítica de la calidad de los servicios en el campo móvil, ya que el usuario incorpora las respuestas del dispositivo a su propio devenir y si bien establece una dependencia del mismo, no estará dispuesto a esperar demasiado para actuar.

Como se desprende de las consideraciones mencionadas, la incorporación de los dispositivos móviles en todas las actividades humanas plantea un escenario donde entender, representar el dinamismo humano y darle soporte, constituyen tareas cada vez más complejas que imponen la colaboración e integración de varios campos disciplinares.

2. Marco teórico

Para poder comprender la naturaleza de las aplicaciones móviles recurrimos a la clasificación referida en [2] que describe dos grandes ejes y sus subclasificaciones:

Tecnologías y Características, que incluyen:

Técnicas de localización/orientación
Tipo de ambiente
Recuperación de datos
Implementaciones sensibles al contexto

Dispositivos y componentes del cliente, que incluyen:

Tipo de dispositivo móvil
Tipo de aplicación (arquitectura)
Funcionalidades soporte según experiencia de usuario
Aspectos especiales de interfaces de usuarios no aplicables a PC.

De acuerdo a lo expresado en [3] donde se sostiene que la computación debe ser adaptativa a los nuevos requerimientos y, conforme el objetivo fundamental de toda actividad ingenieril que es la contribución a mejorar la existencia del hombre, es necesario atender aspectos del usuario sujeto al contexto tales como su estado fisiológico y psicológico. De esta manera se evidencia claramente la complejidad de las aplicaciones móviles caracterizadas como 3a (anywhere, anytime, anyhow) [3] para atender a aspectos tan diversos y cambiantes de allí el vasto abordaje al tema de servicios context-awareness.

Que un sistema se adapte a un contexto (sea context-awareness) implica que estará atento a factores ambientales y reaccionará en consecuencia. La información que describe a una entidad del contexto se puede categorizar con respecto a: *individualidad* (natural, humana, artificial, grupo), *actividad, ubicación, tiempo*, *relaciones* (social, funcional, composición). Mientras que el contexto propio de las personas puede clasificarse en *físico* (Ej.: datos personales) y *mental-emocional* (Ej.: interés, humor, experiencia). Otras taxonomías los describen como contextos activos o pasivos dependiendo de cuánto influyen sus cambios en la adaptación de la aplicación, los primeros serán los que generan mayor oportunidad de proveer servicios frecuentes en lapsos de tiempo pequeños, lo que resulta estratégico para dar un mejor servicio centrado en el usuario.

Atento las preferencias del usuario que emergen de su subjetividad y experiencia, es éste quien debe establecerlas, aunque muchas aplicaciones contemplan esquemas predefinidos de perfiles como umbral de clasificación. Un factor a tener en cuenta es el estado emocional del usuario que traccionará hacia modos distintos de vivenciar.

Si bien la creciente disponibilidad de servicios permite la integración de componentes probados, conlleva también a un grado mayor de dependencia de sus proveedores. Otro aspecto relevante es la energía, la actualización constante desde servidor remoto, mayor procesamiento y uso de pantalla son consumidores de baterías. Para actividades expuestas al aire libre y alejado de fuentes de recarga, la autonomía es crítica en largas jornadas de trabajo. Otro problema de los dispositivos es su fragilidad en espacios de trabajo rudos expuestos al viento, tierra, humedad, golpes. Otros factores influyentes son el ancho de banda de comunicación, tráfico de datos en servidor, seguridad, cantidad de usuarios accediendo simultáneamente. Con tantos RNF se hace necesario identificar si existe entre ellos trade-off (Ej: velocidad de recuperación y calidad de la imagen) para lograr punto de acuerdo.

Otra situación que surge del carácter social del uso de las aplicaciones, se refiere a la influencia que tiene la información del grupo de afinidad que puede cambiar las preferencias personales.

Resulta que estas capas del contexto no trabajan aisladamente, además de captar información de ellas, será valioso para la aplicación contar con la habilidad de integrarlas para lograr la mejor adaptación a las necesidades del usuario. Para ilustrar se presentan ejemplos de adaptaciones al contexto:

Eventos ➔	Reacciones de la aplicación
Detectar grupo de afinidad (Ej:edad)	Proveer discurso con lenguaje ameno al grupo.
Establecer preferencia (Ej: recorrido libre o estandarizado)	Proveer información según cada formato.
Estado emocional (Ej: nivel empatía)	Proveer varios estilos de abordaje
Existe experiencia previa (Ej: camino ya recorrido)	Proveer diversas versiones de recorrido (generar sorpresa)
Movimiento del dispositivo (orientación)	Proveer diferentes enfoques visuales (o interpretaciones)
Cotas de tiempo del usuario	Proveer alternativas de recorrido adecuados al lapso de tiempo
Detectar niveles de luz / poca batería	Ajustar color y brillo de pantalla / servicios de bajo rendimiento
Detectar tipo o familia del dispositivo	Proveer transmisión de imagen bajando la resolución o bajar los cuadros x segundo en video
Detectar interacción grupal	Permitir comunicación, compartir información grupal
Detectar si se cumplen restricciones (Ej: servicio de guía gratis en museo si hay 10 o más personas)	Avisar si un servicio está condicionado

No toda la información proveniente del contexto es válida: puede ser imprecisa, desactualizada, contener errores, provenir de una fuente poco calificada, poseer una granularidad inadecuada. Con la posibilidad de mejoramiento gradual, la información debe considerarse y mientras que un usuario con experiencia hará siempre prevalecer su conocimiento para decidir, un usuario inexperto dependerá más de lo que la aplicación le ofrezca, la misma deberá adaptarse para reducir esta incertidumbre.

2.1. Arquitectura móvil

Complementariamente presentaremos un diagrama que identifica qué aspectos se deberían modelar en una solución móvil context-awareness, el mismo se esquematiza en Figura 1:

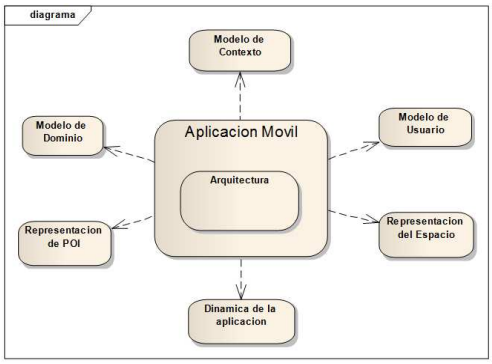


Figura 1- Integración de Modelos de aplicaciones móviles- Dependencias

El **modelo del dominio** representa qué funcionalidad va a tener. Podría configurarse, para lo que necesitará conectarse con el modelo de usuario para personalizar la aplicación (Ej.: usuario experto o novato).

El **modelo de contexto** representa las capas de aspectos del ambiente que son de interés, restricciones, tiempo, y que dinámicamente proveen información para que la aplicación reaccione en consecuencia adaptándose a nuevos valores de las variables. Estos valores pueden capturarse a través de sensores, métodos relacionados al uso de la cámara (códigos QR[4]) o en forma manual. Este modelo representa lo más cambiante debido al frecuente reposicionamiento del usuario con su dispositivo.

El **modelo del usuario** debe representar su posicionamiento y puede contener preferencias personales, es un modelo más estático. Expresa lo que la aplicación necesita conocer del usuario. El sensado de la posición puede ser directo (automático, invisible al usuario) o indirecto (interviene el usuario para informarla).

La representación de los **puntos de interés** (POI), éstos son fijos y pueden ser edificios, objetos, áreas, comercios, piezas de interés, etc. Cada POI tiene posición, contenido sobre cómo llegar a él y requiere ser ubicado en el espacio.

La **representación del espacio** es el soporte donde la aplicación va a dar servicio, es donde el usuario va a actuar, puede ser el mapa, esquema o plano. Dependiendo de la granularidad optaremos por una visión macro (outdoor) o micro (indoor).

Representar el espacio en 3d es más complejo y no bastará con una posición de latitud y longitud sino que necesitaremos implementar otros recursos como grillas, cuadrículas superpuestas o se podría optar por una representación vectorial. Todas estas representaciones generalmente se utilizan en espacios interiores.

La **dinámica de la aplicación** representa la forma cómo se va a recorrer el espacio físico, bajo un esquema predefinido, libre o combinaciones de ambas. Podría optarse por una estructura lineal cuando el recorrido esta preestablecido o podría usarse un grafo si hubiera varias opciones de acceso a un POI. El formato libre facilita acceder a puntos aislados sin una secuencia. En los casos donde las actividades del usuario incluyen un recorrido, es necesario definir un marco de referencia el cual sea capaz de proveer por ejemplo el "mejor camino".

El diseño de la **arquitectura de la aplicación** dependerá de las características del dispositivo móvil, de la red de comunicación, el tipo de aplicación, entre otras y puede tipificarse según las siguientes alternativas:

[4] Código de barras QR (Quick Response Barcode) es un sistema para almacenar información en una matriz de puntos o código de barras bidimensional creado por la compañía japonesa Denso-Wave en 1994

- Cliente: la aplicación móvil se instala y se ejecuta en el dispositivo.
- Cliente-Servidor: la aplicación móvil se desarrolla en dos partes, una se instala y se ejecuta en el dispositivo móvil y la otra se instala en el servidor, quedando disponible para que la aplicación del cliente la acceda.
- Servidor Web: la aplicación reside y se ejecuta en el servidor, se accede a ella a través de un contenedor Web instalado en el dispositivo móvil.

Al momento de elegir la arquitectura se deberá tener en cuenta la conectividad, abstracción de plataformas, objetivo de captar mayores usuarios, frecuencia de uso, volumen de información a compartir y reusar, entre otras.

Para integrar contenidos digitales en la aplicación es necesario diferenciar dos capas: una de posicionamiento y otra de contenidos asociados. Su actualización dependerá de los cambios de posicionamiento y/o orientación.

Las aplicaciones móviles pueden estar enriquecidas con Hipermedia Móvil, ésta cuenta con links y caminos, es el mismo concepto utilizado en aplicaciones Web, la diferencia en móviles es que puede vincularse con el camino real que el usuario está haciendo y al conocer la posición puede brindarle alternativas de otros POI.

Siguiendo el concepto de MVC[5], las capas de contenidos o concerns pueden superponerse en un mismo POI, podría mostrarse solo alguna parte de esa información o que el usuario elija cual navegar, integrando de esta forma las capas. Es importante diseñar una aplicación desacoplada haciendo uso de las buenas prácticas de la Ingeniería del Software para que pueda escalar y evolucionar. A continuación, en Figura 2 se muestra un modelo genérico vinculando todos los aspectos descriptos:

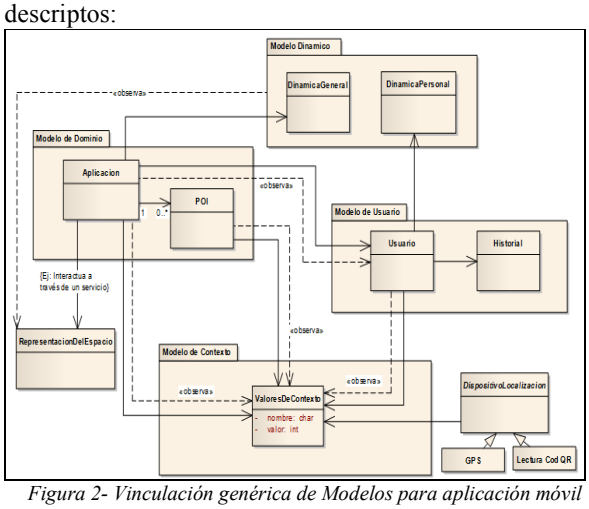


Figura 2- Vinculación genérica de Modelos para aplicación móvil

[5] MVC Model View Controller, este patrón propone la separación de la aplicación Web en distintos componentes (interfaz de usuario, el modelo de negocio y la lógica de control)

3. Estado del arte del desarrollo móvil

Hemos tomado como referencia un análisis del estado del arte sobre el desarrollo de aplicaciones móviles expuesto en [2], el mismo se basó en un conjunto bibliográfico sobre 120 piezas científicas producidas entre 1999 y 2011. Los autores sugieren una correlación entre la relevancia e interés en cada aspecto y la intensidad de investigación del mismo. Se tomó un subconjunto de información de los años 2005 y 2010 para denotar el avance en el quinquenio en los aspectos que se detallan:

Método de recuperación de datos: La literatura referente a este tema ha incrementado considerablemente en especial la Recuperación de datos Centralizados en forma remota, que se ha duplicado en cantidad.

Técnicas de Localización: Las técnicas de localización del dispositivo (persona) se pueden clasificar en *Directas* (Ej: GPS, triangulación Wifi, triangulación red telefónica) e *Indirectas* (Ej: QR). Las que mayormente fueron referidas en bibliografía son las Outdoor y dentro de ellas GPS, también se registran incrementos en literatura relacionada con localización por imágenes (asociadas al desarrollo de la realidad aumentada RA[6]). Si bien las formas de posicionamiento se combinan en las aplicaciones, la actual tendencia indica que la forma de posicionamiento absoluto GPS es la más popular por su uso masivo en celulares. Por otro lado la localización por lectura de códigos de barra bidimensional (Ej: QR) se sigue utilizando por sus bajos costos y porque brinda posición y sugiere orientación (al menos por un lapso de tiempo).

Software de base en Dispositivos móviles: Dentro de los dispositivos móviles podemos mencionar PDAs, celulares, smartphones, tablets, infokiosks, laptops, en ellos corren software de base como sistemas operativos diversos. En la literatura seleccionada en [2] los lenguajes de programación de aplicaciones móviles (en su conjunto) han casi duplicado su referenciación, mientras que los sistemas operativos han tenido un desarrollo exponencial debido a la inexistencia de literatura en 2005. Dentro de este grupo los más atendidos son Symbian, Apple-iOS y Android.

Implementaciones context-awareness: Este campo ha duplicando las referencias en el mismo período. Las categorías más investigadas han resultado ser las concernientes al usuario, siguiendo por las de orden

[6] Realidad Aumentada permite al usuario de aplicaciones móviles enriquecer el mundo real incluyendo objetos virtuales superpuestos a ella para mejorar su percepción. La RA no reemplaza la realidad sino que la suplementa.

social y de servicios. Dentro de las centradas en el usuario las más destacadas son las de localización, orientación (sin registros en 2005), las que escuchan el historial y las preferencias. Dentro del contexto social lo más indagado ha sido el relacionamiento e interacción grupal, mientras que en la capa de servicio lo más destacado ha resultado la disponibilidad.

Interfaces móviles: El abordaje de este tópico requiere un tratamiento especial para los dispositivos móviles ya que dependerá de las dimensiones de las pantallas. Los reducidos tamaños y las limitaciones en resolución pueden afectar la presentación de elementos de la realidad con excesiva calidad. El lenguaje en este caso deberá ser mínimo y claro, las interfaces intuitivas y consistentes, permitiendo al usuario configurar por ejemplo límites de recepción de información. Las funciones (si fueran muchas) no deben estar disponibles en una única pantalla, se deberá maximizar el uso de metáforas y mostrar en la pantalla principal sólo las funciones más importantes y las secundarias se obtendrán por navegación. Existen patrones de Diseño típicos como: despliegue de listas, despliegue de botones en tablas, menú, pestañas de navegación, migas de pan (desde la perspectiva de la Usabilidad representan la huella de navegación). Se pretende de esta manera facilitar la interacción del usuario al interactuar ante una interfaz pequeña, para lo que se recomienda un uso restringido de pestañas, el uso de migas de pan podría reemplazarse por listas y particionarse la navegación para evitar los recorridos profundos para acceder a la funcionalidad. La literatura en el campo de la Usabilidad, muestra un incremento notable (8 veces mayor) en aspectos de los dispositivos pequeños.

Otros servicios muy utilizados son los mapas interactivos, éstos son clave ya que muchas aplicaciones montan sus servicios sobre ellos. Algunos estudios exploratorios han revelado que los usuarios se ubican mejor en mapas de papel aunque sean de mayor tamaño. La clave estaría en que necesitamos ver el espacio real cuantificable donde se identifiquen límites de referencia.

La selección táctil está siendo usada con mayor asiduidad incluso existieron implementaciones con escasa repercusión como PC All in-one. Como futura tendencia se menciona el uso de interfaces multi-touch presentando una interacción más compleja. Existe incluso un protocolo de gestualización para dos dedos con sus respectivas semánticas de movimientos.

Otro canal de ingreso de información en expansión es la captura de la voz aunque se requiere de condiciones ambientales de silencio para que sea de calidad.

Con mayor intensidad las aplicaciones implementan interfaces de realidad aumentada (RA) aprovechando la cámara de los dispositivos, desplegando información asociada a una imagen luego de su reconocimiento.

Las capacidades de las interfaces móviles para adaptarse al contexto están todavía en desarrollo y a medida que se consoliden, se proveerán servicios más cercanos a las necesidades del usuario.

4. Trabajo de campo: investigación del desarrollo móvil regional

Hasta aquí hemos visto qué es una aplicación móvil sensible al contexto, cuál es el estado actual de la industria móvil, cuál ha sido su crecimiento, las ventajas y los problemas que todavía adolece, los dominios de aplicación con mayores servicios de este tipo y el modelo arquitectónico de aplicaciones móviles context-awareness. A partir de este marco global, es que se analizará un espectro de información focalizada regionalmente. Es por ello que se realizó una indagación sobre el estado del arte del desarrollo de aplicaciones móviles en el mercado de La Plata. Como hito inicial se diseñó una encuesta digital de carácter anónimo utilizando una versión libre de herramienta SurveyMonkey [18] y se dirigió a dos auditorios:

Sector Privado: Se realizó una búsqueda en empresas y consultoras desarrolladoras de software en la región, tomando como fuente guías en Internet o cámaras de afinidad. Se contactaron 16 empresas del Distrito Informático La Plata [15], y 5 empresas de mayor tamaño con sede local las cuales no respondieron el contacto y el resto fueron contactos a desarrolladores independientes.

Sector público: La información sobre e-goverment se canalizó a través de una reunión con el Director de la Dirección de Informática y Comunicaciones Provincial quien nos dio una visión global del desarrollo móvil en las distintas dependencias a su cargo.

Contenido de la encuesta: el cuestionario incluyó aspectos de producción de aplicaciones móviles, capacitación y especialización del personal en esta tecnología, descripción de las demandas sobre movilidad, tecnologías utilizadas: herramientas y entornos y metodologías de producción usados. Del total de contactados se obtuvieron 15 encuestas respondidas, se procesaron y se arribaron a los siguientes resultados:

- Pregunta 1: ¿Ha desarrollado o está actualmente desarrollando aplicaciones móviles? El 53,33 % de los encuestados respondieron afirmativamente, a través de este porcentaje tomamos conocimiento de que la producción móvil local está en estado inicial de explotación, pero a la vez denota que esa tecnología ya ha sido abordada.
- Pregunta 2: ¿Para qué sistema operativo (SO) ha desarrollado? Las respuestas indican que el 90% de las aplicaciones móviles desarrolladas fueron para SO Android, mientras que el 40% son para iOS y le siguen Windows Phone con el 20% y el 10% restante para

Blackberry OS. No hubo experiencias para sistema operativo Symbian tal como se grafica en Figura 3

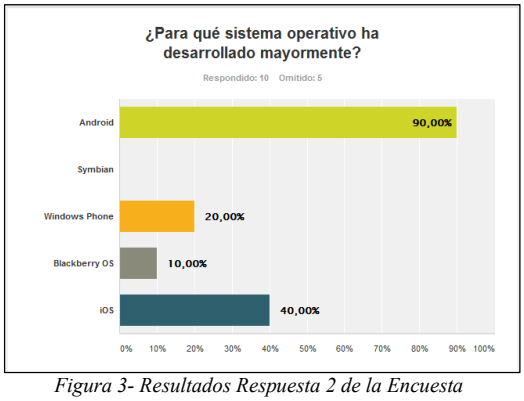


Figura 3- Resultados Respuesta 2 de la Encuesta

Se realizó la comparación entre información sobre el uso de SO para dispositivos móviles, el desarrollo local de aplicaciones por SO (encuesta) y bibliografía sobre Tecnologías Móviles. El objetivo fue buscar puntos de encuentro entre el desarrollo móvil local, nacional y el de escala mundial, para ver si los fenómenos en estos espacios van en la misma dirección.

Respecto de la fuente de información sobre del uso de SO para dispositivos móviles se tomó la investigación sobre ventas del mercado mundial en el período Mayo/Junio 2013 [8]. El mercado mundial total se divide en mayor fracción para iOS, en segundo lugar Blackberry, y en tercer lugar pero con una tendencia creciente Android [10]. Para el caso de la descripción de la bibliográfica sobre Tecnologías Móviles sobre sistemas operativos la fuente de datos es el artículo desarrollado en [2].

Según Gartner la venta a nivel mundial de teléfonos celulares en el período Mayo/Junio 2013 fue de 435 millones de unidades de los cuales el 51,7 % fueron teléfonos inteligentes. Esto indica que un gran conjunto de usuarios incorporados al consumo de servicios de tecnología móvil, los cuales impulsarán nuevos desarrollos.

Aclaramos que los smartphones sólo conforman un subconjunto del universo de dispositivos móviles, aún así los tomamos como representativos debido a su masividad y tendencia en expansión. Tal como se despliega en la Tabla 1, encontramos que los porcentajes son concordantes. En la segunda columna se cuantifican los desarrollos de aplicaciones móviles regionalmente según SO los cuales se distribuyen en el mismo orden de posicionamiento, comparándolos con el comportamiento del mercado el mundial (acotado a la referida investigación de [8]) descripto en la primera columna, aunque en proporciones distintas).

La tercera columna describe el porcentual de bibliografía referida a cada SO, si bien no respeta el mismo orden de posicionamiento, es importante destacar que la literatura concentra el 57,13 % en los tres SO más utilizados, es decir los mejor rankeados.

Tabla 1. Comparación ventas de dispositivos, desarrollo local de aplicaciones y producción bibliográfica sobre Tecnologías Móviles.

SO	% mercado mundial (Fuente: Gartner)	% desarrollo según SO (Fuente: Encuestas)	% Bibliografía (Fuente [2])
Android	79.0 %	90 %	14,28 %
iOS	14.2 %	40 %	28,57 %
Microsoft	3.3 %	20 %	14,28 %
BlackBerry	2.7 %	10 %	Sin datos
Bada	0.4 %	Sin datos	Sin datos
Symbian	0.3 %	0 %	28,57 %
Otros	Sin datos	Sin datos	14,3 %

Subsidiariamente en el informe de Gartner se resalta a América Latina y países del este de Europa que se posicionan en segundo lugar después de Asia y el Pacífico, como la región de mayor crecimiento de ventas de dispositivos de telefonía móvil. Otras fuentes de investigación [9], [16] confirman que Android es la plataforma dominante del mercado argentino debido a la firme posición de los fabricantes asociados a la misma. Estos argumentos refuerzan lo expresado en el cuadro anterior que indica que la producción local de software con tecnologías móviles respecto de los sistemas operativos va en la misma dirección nacional y mundial.

- Pregunta 3: ¿Qué tipo de tecnología ha usado para captar el posicionamiento? Los resultados muestran que el 60% tuvo experiencias Outdoor, mientras que el 30% ha desarrollado Indoor. Y casi en la misma proporción (20%) han utilizado lectura QR, orientación y han combinado varias tecnologías. Estos resultados son consistentes con lo que desprende de la literatura que indica que el posicionamiento por GPS es el de mayor tendencia y que seguirá incrementándose debido al desarrollo de red de telefonía móvil e incremento de dispositivos inteligentes conectados.
- Pregunta 4: ¿Qué tipo arquitectura ha implementado? La arquitectura Cliente-Servidor acapara un 80 % de las respuestas, seguida por Servidor Web el 50% y Cliente el 30%. Podríamos sugerir, a través de las dos más usadas (Cliente-Servidor y Servidor Web), que el tipo de aplicaciones móviles desarrolladas en la región contienen un alto grado de servicios no particularizados y que las capas de aspectos contextuales en su mayoría son estandarizadas. Indagaciones más profundas podrían hacernos conocer más evidencias para concluir si esta hipótesis es válida.

▪ Pregunta 5: La demanda proviene, ¿en su mayoría del ámbito local, nacional, extranjero, desarrollos propios sin demanda específica? A este respecto las cifras denotan que el 50% corresponde a requerimientos locales, el 40% del orden nacional y el 30% del exterior, mientras que el 20% corresponde a desarrollos propios. Esta estructura porcentual indica una importante demanda regional (empresas), los requerimientos nacionales e internacionales indican una buena conexión y posicionamiento de la producción local en esos mercados ya que la cuantificación es significativa. En especial el 30% de respuestas con experiencias de demandas del exterior, se relaciona con los informes de exportaciones de Software y Servicio Informáticos brindados por el Ministerio de Industria que muestra que en el período 2003-2010 hubo un crecimiento del 290% según [7]. El desarrollo de aplicaciones móviles sin una demanda específica asciende al 20 %, un porcentaje notable ya que esta producción generalmente se considera innovadora, que ha encontrado un nicho no explotado. Este último punto indica un importante esfuerzo dedicado a la innovación tecnológica, que a diferencia de la producción masiva de software, aporta mayor valor agregado en productos diferenciales [7] y contribuye a la formación de una matriz industrial competitiva.

▪ Pregunta 6: Sectores de la economía para los cuales ha desarrollado soluciones móviles, según se destaca en Figura 4.

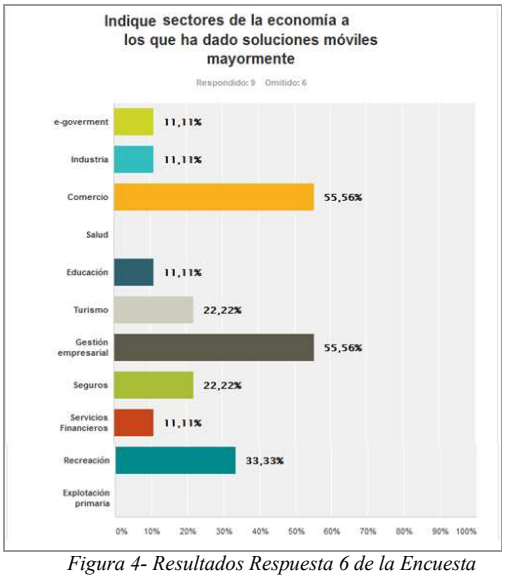


Figura 4- Resultados Respuesta 6 de la Encuesta

La investigación muestra en primer lugar al Comercio y Gestión Empresarial con un 55,56%, luego con 33,33% Turismo y Seguros, con un 11,1 % los sectores e-government, Industria, Educación y Servicios Financieros. Estos porcentajes indican qué sectores se han abierto a la movilidad en etapas tempranas, habría que indagar más esas aplicaciones para conocer qué tipo de funcionalidad implementan si son de carácter crítico, si conforman el corazón del negocio, si son meramente informativas, etc. Cabe aclarar que la sectorización de la economía presentada se tomó en base a la Clasificación Industrial Internacional Uniforme (CIIU) [6] [17], un nomenclador de todas las actividades económicas que generó ONU y el cual sirve de estándar mundial para comparar niveles de desarrollo de los países.

▪ Pregunta 7: ¿Qué tan adecuada considera la metodología de desarrollo utilizada? Esta pregunta intenta incursionar en algunos aspectos del proceso de producción. El 55,56 % los respondentes las consideran adecuadas aunque mejorables, mientras que el 33,33% las considera como muy adecuadas, implicando muy buen desempeño lo que podría ser un emergente de la utilización de buenas prácticas de la ingeniería y nivel de madurez alcanzado.

▪ Pregunta 8: ¿Cómo calificaría el costo de capacitación del personal involucrado en el desarrollo? La visión de los encuestados muestra que el 55,56 % considera a la inversión adecuada y el 11,11% lo respondió que fue poco costosa, de lo que se podría deducir que un total de 66,67% considera que los profesionales que calificaron produjeron satisfactoriamente retornando la inversión.

▪ Pregunta 9: ¿Considera el mercado local atraído por desarrollos móviles particularizados a problemáticas de la región? Según se desprende de Figura 5, las encuestas revelan un alto nivel de desconocimiento (46,15%) por parte de la industria. Esto conjuntamente con el 38,40% de opinión sobre que está poco atraído, estaría indicando un sector no estudiado en cuanto a sus comportamientos y potencialidad. Una indagación profunda revelaría el verdadero estado de situación respecto a esta cuestión.

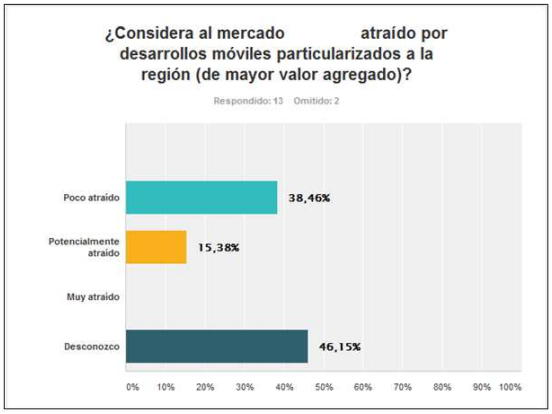


Figura 5- Resultados Respuesta 9 de la Encuesta

▪ Pregunta 10: ¿Qué aspectos podrían impulsar el desarrollo local de aplicaciones móviles? Se encontró que el auditorio no considera que alguna tendencia concentre

la posibilidad de desarrollo, no obstante la más destacada es el Seguimiento de productos (Tracking) con un 30,77%, le sigue la ampliación del ancho de banda con un 23,08% y en tercer lugar con el 15,38% el mayor uso de sensores y mayor autonomía energética. Aunque la domótica en la encuesta aparece con un escueto 7,69%, particularmente se considera que existe una oportunidad de desarrollo a futuro debido a la tendencia continua de incorporación de tecnología hogareña, la optimización del uso luz y gas, el monitoreo remoto por celular.

5. Algunos ejemplos de desarrollos propios de la región

Se ha identificado en la región un desarrollo por parte de la Municipalidad, es el caso del Sistema de Estacionamiento Público Medido *(SEM)* que actualmente dispone de captura de inicio y fin de servicio a través de envío de mensaje de texto pago por celular (uso típico). Posteriormente se implementó una solución móvil que reduce este gasto de los ciudadanos (detallado en Figura 6), aunque se desconoce la causa por la cual no ha tenido aún difusión masiva. La aplicación no incluye posicionamiento por lo que consideramos podría incrementar su valor estratégico obteniendo la ubicación del vehículo (al menos inicial y la de mayor porcentaje de permanencia), ya que actualmente no hay forma de registrarlo, además se podría incluir dentro de su funcionalidad la compra de crédito para estacionamiento articulando con servicios como *Botón de Pago* [19] . Esto constituye una línea de acción concreta para mejorar la calidad de este servicio al ciudadano.

Ahorro en estacionamiento diario por usuario (promedio):
3 hs. Precio Hora normal------ 7,5 $
3 hs. Precio Hora pico -------- 12 $ Total ---19,5 $
Costo SMS inicio servicio --- 0,25 $
Costo SMS fin servicio ----- 0,25 $ Total -- 0,5 $
El costo de envío de mensajes en este caso representa el 3%, mientras que en un estacionamiento de mínima fracción ½ hora simple el gasto en SMS representaría un 40%.

Figura 6 - Estimación del ahorro por el uso de la aplicación móvil en SEM.

Con el conocimiento de otras soluciones regionales, se realizó una búsqueda de aplicaciones gratuitas de servicios a la comunidad y se identificaron dos aplicaciones móviles desarrolladas en el ámbito de e-government y otra obtenida de la indagación del reservorio SEDICI – UNLP [11]. Las tres tienen la característica común de estar basadas en posicionamiento, aunque con diferentes mecanismos de captura de la posición y arquitecturas distintas. Cabe aclarar que no se dispone de información sobre la evaluación cuantificada de los beneficios de su utilización, se detallan en Tabla 2, 3 y 4.

Tabla 2. Síntesis sobre Aplicación Botón de Alerta.

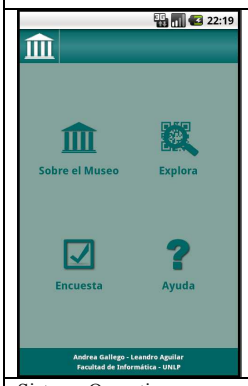
Nombre de la aplicación	**BOTON DE ALERTA** [12]
	La funcionalidad de esta aplicación móvil consiste en el envío de alertas ante una situación de pánico. Manda las coordenadas GPS de la ubicación del dispositivo, o voz, a una lista de contactos que puede incluir el 911, mails u otros teléfonos o un tweet. Esta lista es establecida por el usuario. El mensaje contiene la URL de Google Maps con las coordenadas GPS. Al llegar a destino se puede ejecutar y ver en el la ubicación. Permite la configuración local de la lista, preferencias, la muestra del mapa ubicando el punto y el envío de comentarios por mail para la mejora del servicio.
Sistema Operativo:	Android e iOS
Contexto:	No atiende capas del contexto excepto la posición del dispositivo.
Técnicas de Localización:	A través del GPS (Outdoor)
Diferenciación de perfil de usuario:	No implementa perfiles de usuarios diferenciados.
Preferencias de usuario:	Implementa en el cliente la lista de personas a contactar desde la agenda.
Puntos de interés (POI):	No aplica su uso.
Representación espacial:	A través de mapas de Google Maps
Arquitectura de software:	Cliente
Dinámica de aplicación/ Realidad aumentada	No implementa
Autoría: Gobierno de la Provincia de Buenos (Ministerio de Seguridad – Provincia Seguros)	
Observaciones: Requiere la toma de la posición GPS en forma precisa y estable (no siempre bien resuelta). Esto es crucial para que se pueda asistir correctamente al pedido. Podría en el futuro tipificar el origen de la alerta, diferenciando según perfil de usuario en hechos de violencia, urgencias médicas, adolescentes amenazados, adultos mayores perdidos y personas con discapacidad. En la ultima versión incluye la posibilidad de captura remota de la posición para el caso de personas perdidas.	

Tabla 3. Síntesis sobre Aplicación Donde pago.

Nombre de la aplicación	**DONDE PAGO** [14]
	La funcionalidad de esta aplicación web consiste en mostrar todos los centros de pago Bapro circundantes a la posición GPS donde se encuentra el dispositivo móvil o la posición de la IP de la Pc. Sistema Operativo: Indistinto, se accede a través de browser a URL en Internet Contexto: No atiende capas del contexto excepto la posición del dispositivo.
Técnicas de Localización:	En móvil a través del GPS (Outdoor)
Diferenciación de perfil de usuario:	No implementa perfiles de usuarios diferenciados.
Preferencias de usuario:	No implementa preferencias del

	usuario.
Puntos de interés (POI):	Implementa una capa de POI que son las bocas de pago Bapro sobre el mapa.
Representación espacial:	A través de mapas de Google Maps
Arquitectura de software:	Servidor Web
Dinámica de aplicación/ Realidad aumentada	No implementa
Autoría: Provincia NET – Nuevos Emprendimientos Tecnológicos.	
Observaciones: La pantalla muestra un área muy pequeña en la pantalla (limitaciones propias del tipo de interfaz) que dificulta ver los resultados claramente para personas con disminución visual.	

Tabla 4. Síntesis sobre Aplicación Museo de Física.

Nombre de la aplicación:	**MUSEO DE FISICA – UNLP** [4], [5], [13]
	La aplicación móvil permite al visitante del Museo de Física – UNLP investigar sobre el museo y acceder a información adicional que se dispone de los objetos que se exponen en él, pudiendo observar, videos del mismo en funcionamiento, explicaciones textuales y auditivas, juegos educativos sobre el tema y encuestas relacionadas para incentivar la participación del usuario. Aclaración: En asociación a este proyecto se desarrolló una aplicación Web que administra la información de objetos en y la generación de códigos QR.
Sistema Operativo:	Android
Contexto:	No atiende capas del contexto excepto la posición del dispositivo.
Técnicas de Localización:	A través de lectura códigos QR (Indoor)
Diferenciación de perfil de usuario/Preferencias de usuario	No implementa
Puntos de interés (POI):	Implementa un conjunto de POI pero no los dispone sobre un mapa.
Representación del espacio:	No aplica en este caso.
Arquitectura de software:	Cliente – Servidor
Dinámica de la aplicación:	Permite el recorrido libre que define el usuario, la aplicación reacciona ante la nueva lectura de código QR frente a un POI.
Enriquecida con Realidad Aumentada: Implementada a través de lectura de códigos QR, no pudo ser implementada en forma tradicional a través de los sensores del dispositivo, debido a condiciones ambientales del Museo (habitáculo sin ventanas, instrumentos poco iluminados en vitrinas cerradas con mucho reflejo que impide una captura clara de cámara fotográfica).[7]	
Autoría: Tesis de grado A.Gallego, L.Aguilar, Lic.Informática Fac. Informática, UNLP	
Observaciones: (Requiere software de lectura QR instalado). Al interactuar con la aplicación, resultó ser muy didáctica y aprehensible para todo público, en especial jóvenes y niños empapados en las nuevas tecnologías.	

[7] Los autores justifican el uso de QR indicando que esta interacción produce retroalimentación e intercambio de información que conlleva a aumentar aún más realidad.

6. Conclusiones

En el presente trabajo se abordó el marco teórico sobre Tecnología Móvil y en especial el desarrollo de Aplicaciones Móviles con posicionamiento. Conjuntamente se describió el estado del arte de los diferentes aspectos que integran un desarrollo móvil en base a una investigación sobre cuantificación de referencias a estos tópicos en reconocida bibliografía. De ese estudio se tomaron los años 2005 y 2010 para realizar comparaciones de evolución. Conjuntamente se abordó el modelado arquitectónico genérico de una solución móvil context-awareness. Con la intención de conocer acerca del desarrollo móvil en la región se realizó una investigación en base a una encuesta dirigida a empresas de desarrollo de software y se mostraron los resultados sobre aspectos de la producción de aplicaciones móviles en el ámbito regional. Como resumen de este análisis se arribó a la idea preliminar del perfil regional: desarrollo móvil en crecimiento, orientado mayormente a SO Android, con arquitectura predominante Cliente-Servidor, demanda de servicios en forma equilibrada entre mercado local, nacional e internacional. Con una representativa producción de software sin demanda específica, pudiéndola asimilar como desarrollo de innovaciones tecnológicas, prestando servicios mayormente al comercio y la gestión empresarial, en menor medida a entretenimientos. Las metodologías se calificaron como adecuadas, con la percepción de una justa inversión en capacitación de personal, con la concepción actual de un mercado local poco interesado y con la visión a futuro que la expansión de Tracking de productos, ampliación del ancho de banda, mayor uso de sensores y dispositivos con mayor autonomía energética, podrían impulsar el crecimiento de la demanda de aplicaciones móviles (con posicionamiento) a nivel regional.

Esta síntesis indica que la infraestructura del desarrollo móvil en la región, está en correlación con el ámbito nacional y las tendencias globales, aunque se requiere de estimulación del uso de aplicaciones móviles no masivas con alto valor local. Partiendo de esta primera aproximación de la situación regional, consideramos fundamental un abordaje más amplio en cuanto a empresas entrevistadas, con mayor profundidad para tipificar las aplicaciones móviles desarrolladas, como así también indagar si en los planes gubernamentales existen líneas de promoción local de la producción de este tipo de herramientas y proyectos de servicios a la comunidad aprovechando la masividad de la tecnología móvil actual.

Resaltamos en este sentido, el surgimiento de casos concretos de aplicaciones móviles basadas en posicionamiento construidas en la región, de uso libre, con las cuales tuvimos oportunidad de interactuar y expresar observaciones sobre la experiencia. Por último

se demarcó el caso de estacionamiento público medido cuya implementación móvil representa un ahorro considerable para los ciudadanos usuarios del mismo.

Adicionalmente como acción futura, se sugiere la elaboración de un anteproyecto para presentarlo ante el Municipio que contenga líneas de mejora en la difusión de los beneficios de esta aplicación (ahorro, facilidad de descarga) y otras complementariedades que podrían ser muy útiles a la gestión como la geo-referenciación de los vehículos estacionados y la adición de funcionalidad como compra de crédito por medio del aplicativo.

7. Agradecimientos

A las empresas y desarrolladores que estuvieron dispuestos a acceder a la entrevista de presentación del proyecto y luego tuvieron el interés de contestar la encuesta.

8. Bibliografía

[1] Chung L., Sampaio do Prado Leite J. C.: On Non-Functional Requirements in Software Engineering, Department of Computer Science, The University of Texas at Dallas, Dpto. Informática, Pontifícia Universidade Católica do Rio de Janeiro, (2009).

[2] Emmanouilidis C., Koutsiamanis A., Tasidou A., Mobile Guides: Taxonomy of Architectures, Context Awareness, Technologies and Applications

[3] Pernici B., Mobile Information Systems (Infrastructure and Design for Adaptivity and Flexibility) - Cap. 1 Ed. Springer

[4] Diaz F., Harari I., Gallego A., Aguilar L., Una Aplicación Móvil para el Museo de Física de la Universidad Nacional de La Plata, Facultad de Informática, UNLP.

[5] Gallego A., Aguilar L., Realidad Aumentada en el Museo, Tesis de grado de la Carrera Licenciatura en Informática, Licenciatura en Sistemas, Facultad de Informática, UNLP.

[6] Clasificación Industrial Internacional Uniforme de todas las actividades económicas (CIIU), Naciones Unidas, Nueva York, 2009, Departamento de Asuntos Económicos y Sociales División de Estadística- Revisión 4

[7] Documento de Trabajo sobre Formación de Recursos Humanos en Informática, RedUNCI, Agosto 2013.

[8] http://www.gartner.com/newsroom/id/2573415

[9] http://illuminatilab.com/blog/internet/mercado-movil-latam/

[10] https://www.mapbox.com/labs/twitter-gnip/brands/#2/30.3/-45.4

[11] http://sedici.unlp.edu.ar/ (Repositorio producción académica UNLP)

[12] http://www.provinciaseguros.com.ar/prensa/provincia-seguros-lanzo-boton-de-alerta/

[13] http://www.museo.fisica.unlp.edu.ar/ (Sitio oficial del Museo de Física de la Fac. de Cs. Exactas, UNLP)

[14] http://www.provinciapagos.com.ar/Sistema-Recaudacion/Donde_pago.aspx

[15] http://www.dilp.com.ar/

[16] http://www.comentariosblog.com.ar/2013/08/02/avance-coreano/ (Carrier & Asoc.)

[17] http://unstats.un.org/unsd/publication/seriesM/seriesm_4rev4s.pdf

[18] https://es.surveymonkey.com/s/6BX26GR (Encuesta digital)

[19] http://www.provincianet.com.ar/

La Arquitectura de Aplicaciones en un Modelo de Servicios Reutilizables para Gobierno Electrónico

Marcelo Antonio Castro, Víctor David Sánchez Rivero, Luis Alejandro Vargas,
José Humberto Farfán, María Concepción Aparicio & Daniel Castro
Investigación + Desarrollo en Gobierno Electrónico
Facultad de Ingeniería - Universidad Nacional de Jujuy
mcastro@fi.unju.edu.ar, vdsanchezrivero@fi.unju.edu.ar, lavargas@fi.unju.edu.ar,
jhfarfan@fi.unju.edu.ar, mcaparicio@fi.unju.edu.ar, ddcastro@ fi.unju.edu.ar

Abstract

El presente trabajo tiene por objeto exponer la importancia que reviste la arquitectura de aplicaciones en un modelo de servicios reutilizables, estableciendo la necesidad de automatizar los procesos relacionados con trámites y servicios que brinda el Estado a la ciudadanía, como así también la importancia que reviste el desarrollo de componentes reutilizables bajo el concepto de la Arquitectura Orientada a Servicios (SOA), implementando sus operaciones internas en forma de servicios web. En este contexto se plantea la conveniencia de crear súper componentes como medio para administrar eficientemente los datos suministrados por los componentes básicos, haciendo hincapié en la importancia que reviste la interacción entre los componentes reutilizables, el catálogo de trámites y servicios (CTS) y el administrador de movimiento de documentación (ADOC). Además se destaca el impacto que los componentes reutilizables ejercen sobre las aplicaciones locales, y finalmente se establece la necesidad de utilizar la replicación de base de datos, como tecnología, para propagar cambios en los datos generados entre distintos niveles jurisdiccionales del Estado, ya sea municipal, provincial o nacional.

1. Introducción

El desarrollo de un proyecto, que implementará una solución de Gobierno Electrónico (GE), parte de la premisa de tener en cuenta el problema a solucionar, la institución donde se aplicará tal solución y, esencialmente, la solución IT (Information Technology) que se implementará. En tal sentido, se hace necesario definir el diseño de una arquitectura que describa un conjunto de técnicas, tecnologías, actividades, mejores prácticas, herramientas y materiales a utilizarse para construir un conjunto de soluciones coherentes, eficientes y eficaces, para solucionar los problemas de una población objetivo [1]. La experiencia indica que existe una alta probabilidad de que las soluciones tecnológicas, sobre GE, fracasen sin un adecuado diseño arquitectónico.

Tal diseño deberá tener en cuenta un proceso que permitirá definirlo, y que se inicia con el planteamiento de una necesidad de la ciudadanía, para después identificar el servicio a prestar, para brindarle la solución pertinente. Como es de esperar, se requiere una solución institucional y que será efectuada por un actor institucional.

Una solución tecnológica gubernamental, de este tipo, requiere la aplicación de TIC's (Tecnologías de la Información y la Comunicación), con inversiones importantes relacionadas con aspectos organizativos, financieros, administrativos, sociales y políticos; constituyendo un proceso de GE una solución estratégica para todo ámbito estatal. Bajo esta perspectiva, y en función de una metodología de formalización para la aplicación de GE [2], se emplea una estructura basada en servicios digitales comunes reutilizables; los cuales serán provistos por organismos públicos para dar respuesta a requerimientos de servicios por parte de la ciudadanía.

La ejecución de tales servicios está prevista funcionar como "caja negra"; en donde no es necesario especificar detalles internos ya que es suficiente conocer qué interfaces maneja, los parámetros necesarios y los productos que se vayan a suministrar. Estos componentes software deben ser diseñados para ofrecer una disponibilidad permanente, asegurar la interoperabilidad entre los diversos procesos administrativos y la estandarización de la información entre los organismos del Estado [1].

La idea básica es gestionar la resolución de trámites administrativos desde un portal web, por ejemplo, para evitar inconvenientes tales como pérdidas de tiempo, procesos engorrosos, extrema burocracia y todo lo relacionado a la problemática de iniciar y terminar un trámite común. Es conveniente señalar que ya existen soluciones implementadas pero que solo involucran la información y necesidades de un organismo específico; lo cual se pretende mejorar con el empleo del paradigma de servicios digitales comunes.

De acuerdo a este enfoque, estos servicios deben asegurar que el beneficio de su aplicación no se debe limitar al organismo que los implementó sino que debe ser factible de beneficiar a otras instituciones públicas;

integrando los servicios de todos ellos y asegurar la interoperabilidad de los procesos y la información; una de las premisas del plan de modernización de Estado.

En consecuencia, se busca optimizar el control, el proceso, la transmisión y la centralización de la información crítica entre organismos estatales, aún a costa del factor presupuestario que puede afectar la provisión de los recursos tecnológicos necesarios, sobre todo debido a la existencia de múltiples plataformas informáticas.

En investigaciones anteriores, se demostró que es conveniente la aplicación de arquitecturas modernas para implementar servicios comunes reutilizables. En este sentido SOA (Service Oriented Architecture) es "un marco de diseño para integrar aplicaciones separadas de modo que se pueda acceder a su funcionalidad a través de los servicios de una red. Aunque la implementación de una arquitectura SOA no requiere necesariamente ser realizada a través de servicios web, éstos resultan ser la forma más acostumbrada de implementar SOA" [3]. Y más precisamente se recomienda la aplicación de SOAP/REST [4].

El modelo de servicios digitales comunes reutilizables se sustenta en una arquitectura de software, que brinde flexibilidad, adaptabilidad, disminución de costos de mantenimiento, desarrollo y control para disponer de soluciones de GE de gran rendimiento y potencialidad.

Este trabajo se basa en la arquitectura de datos, la arquitectura de aplicaciones y la arquitectura tecnológica, bases del modelo de servicios comunes.

La arquitectura de datos tiene en cuenta un análisis para definir los datos pertinentes a la gestión de las aplicaciones de un organismo; incluyendo su política de almacenamiento seguro y confiable.

En cuanto a la arquitectura de aplicaciones, se trata de proyectar una estructura de tal manera que se desarrollen aplicaciones integradas y que funcionen eficientemente. En particular, este tipo de arquitectura es el eje central de este trabajo y será analizado en párrafos siguientes.

Con respecto a la arquitectura tecnológica, se debe considerar la organización del software, las interfaces necesarias, la interacción, selección e integración de los componentes tecnológicos, involucrados en el proceso [1].

2. La Arquitectura de Aplicaciones

Para modelar la arquitectura de aplicaciones resulta indispensable tener en cuenta algunos aspectos que se encuentran relacionados con características tales como la comprensibilidad, la completitud y la estabilidad.

Con respecto a la comprensibilidad, es importante documentar la información que se utiliza en la elaboración de los modelos arquitectónicos. En relación a la completitud en la definición de las aplicaciones; se debe tratar de integrar, en una única definición, todas las aplicaciones del modelo, asegurando la integración y la consistencia y evitar duplicar la información.

La estabilidad, en la definición de las aplicaciones, se encuentra relacionada a que esta última debe ser independiente de la tecnología a utilizar, de aspectos relacionados al análisis y diseño o de aspectos relacionados a características individuales o perfiles del usuario final.

3. La Importancia de los Procesos de Trámites y Servicios en GE

La mayoría de las aplicaciones desarrolladas en el ámbito estatal y especialmente en el marco de Gobierno Electrónico, poseen un dominio específico relacionado con la automatización de trámites y servicios suministrados por el Estado a la ciudadanía.

Los procesos relacionados con aplicaciones de trámites y servicios podrán ser ejecutados en el entorno de una intranet gubernamental, tal es el caso de aplicaciones de uso interno; aunque también pueden ser ejecutadas a través de internet, como es el caso de las guías únicas de trámite. En ambos casos, cualquier ciudadano podrá consultar desde los requisitos que necesita para efectuar un trámite, hasta realizar el seguimiento de los mismos.

Para lograr una adecuada interoperabilidad procesal entre las diferentes áreas y unidades organizacionales del Estado es importante efectuar y legalizar la obligatoriedad de documentar digitalmente toda la información contenida en trámites procesales.

4. Los Componentes Reutilizables

La aparición de nuevos paradigmas en el ambiente de la programación, sobre todo el Orientado a Objetos, significó un cambio radical en la forma de desarrollar software. La reutilización de componentes software se puede ejemplificar completamente con el proceso seguido por una industria dedicada a la fabricación de productos similares, en el cual se reutilizarán uno o más pasos. En consecuencia, supone establecer un nuevo enfoque en el proceso de desarrollo de software.

Una definición representativa sobre el concepto de reutilización de software es: "la reutilización de software es un proceso de la Ingeniería de Software que conlleva al uso recurrente de activos de software en la especificación, análisis, diseño, implementación y pruebas de una aplicación o sistema de software" [5].

De acuerdo al objetivo de este trabajo, se considera que la finalidad de un proyecto de servicios comunes reutilizables es lograr que diversos organismos del Estado desarrollen componentes reutilizables, basados en los preceptos de la arquitectura SOA y los servicios web. Para ello se deben analizar funcionalidades básicas o "atómicas", a partir de las cuales se desarrollarán los citados componentes. En base a las conclusiones obtenidas, al investigar distintas

implementaciones en diversos países o jurisdicciones, resulta que el trabajo por delante es arduo y complejo. Se determinó que, en un principio, el desarrollo de un componente será el fruto de un proyecto individual de un organismo específico; a partir de allí se debe generar un efecto "cascada" de manera que otras reparticiones públicas también desarrollen sus componentes, sobre todo por la necesidad de integrar los datos gestionados por los distintos componentes, a través de funciones internas programadas para tal fin. En otras palabras ciertos componentes orientados a un organismo local (provincial o municipal) podrán ser distribuidos a un ámbito nacional, sin mayores esfuerzos. Como ejemplo puede citarse el trabajo del Registro Nacional de las Personas; cuyos procesos, en principio locales, se han integrado hacia una red nacional.

Sin embargo existirán componentes que estarán focalizados al ámbito local, y sus datos se destinarán al área del organismo que los utiliza; tal es el caso de la administración de expedientes, recursos humanos, seguridad, entre otros.

Pero también se hace necesario analizar la universalidad de los datos administrados por los componentes, de manera que puedan ser distribuidos a ámbitos regionales o mundiales.

En resumen, al integrar funciones para enlazar los datos administrados se deberían definir tablas en las que identifiquen los súper componentes (que integran datos), los componentes básicos, su descripción y, si es posible, la plataforma de desarrollo. Por ejemplo se podría definir un súper componente común, local y que aglutine componentes esenciales tales como diccionario de datos, catálogo de trámites y servicios, etc. [1].

5. Las Interacciones entre Componentes Reutilizables

El concepto de reutilizar software viene desde hace un tiempo, donde se recopilaba en librerías de subrutinas especializadas, cuyo el objetivo primordial era maximizar el uso recurrente de componentes de software existentes, dando lugar a la creación de sistemas de software a partir de software existente. Últimamente, la presión de reducir drásticamente el costo y el tiempo de desarrollo de sistemas y aplicaciones, sin afectar los niveles de calidad del producto, da lugar a un nuevo activo reutilizable denominado Componente de Software.

Según SEI (Instituto de Ingeniería de Software) un componente es "una implementación opaca de funcionalidad, sujeta a composición por terceros y que cumple con un modelo de componentes" [6].

La reutilización de software es el proceso de implementar o actualizar sistemas de software utilizando activos de software existentes, ya que entre el 40 y 60% del código fuente de una aplicación es reutilizable en otra aplicación similar. Solo el 15% del código encontrado, en varios sistemas, tiene características propias y novedosas.

El desarrollo de nuevas aplicaciones en base a componentes, se diseñan utilizando la integración o composición de componentes, donde [7] destaca como elemento fundamental las interfaces.

Las interfaces determinan la totalidad de operaciones que un componente puede ejecutar. Dichas operaciones también son denominadas servicios o responsabilidades. Las interfaces facilitan un mecanismo para que haya una interacción entre componentes y controlan las dependencias entre las mismas.

En la Figura 1, se puede llegar a comprender la composición como una relación Cliente-Servidor entre dos componentes. En el componente Cliente, en este caso el proceso de Gobierno Electrónico, el ciudadano solicita un determinado servicio, una solución a un problema, o realiza una petición al componente Servidor, el cual procesa la operación solicitada y devuelve los resultados, que el ciudadano consume. El componente Servidor, para dar respuesta al petitorio del cliente, puede solicitar un trámite o servicio en otra institución; es decir que a través del Administrador de Movimiento de Documentación (ADOC) consulta el Catálogo de Trámites y Servicios (CTS), y se vuelve a generar una nueva solicitud de un trámite, es decir prevalece el concepto de agrupar trámites y servicios en otros meta trámites.

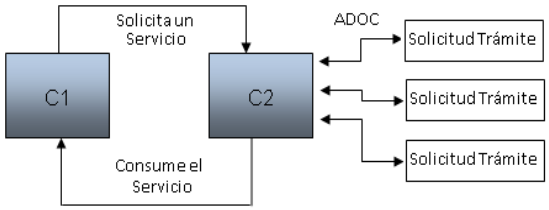


Figura 1. Interacción entre componentes.

La importancia de los componentes es que son reutilizables y como tal para su interacción, mínimamente, deben cumplir las siguientes características:

1. Identificable: para facilitar su catalogación y búsqueda en los repositorios.
2. Accesible solo a través de su interfaz: dar a conocer las operaciones que ejecuta y ocultar el detalle de la implementación.
3. Sus servicios son invariantes: las operaciones no deben variar, es decir no afectar la interfaz.
4. Documentado: debe tener una documentación adecuada, para su utilización.
5. Genérico: el componente puede ser utilizado en una gran variedad de aplicaciones.
6. Autocontenido: tener una menor dependencia de otros componentes.
7. Mantenido: que el componente se encuentre constantemente actualizado.
8. Independiente de la plataforma, del lenguaje, y de las herramientas de desarrollo.

9. Reutilizado dinámicamente: cargado en tiempo de ejecución.
10. Certificado: evaluado por alguna agencia de software.
11. Accedido uniformemente sin importar su localidad: invocar los servicios del componente independientemente de su ubicación.

6. Administración, conectividad y propagación de datos por parte de los componentes reutilizables

A continuación se describen, a modo de ejemplo, un conjunto de componentes reutilizables que se pueden encontrar en cualquier nivel jurisdiccional y el impacto que producen en la generación y utilización de aplicaciones, como así también la complejidad en la administración de los componentes reutilizables y su estructura de propagación:

- DD (Diccionario de Datos): este componente presenta una alta complejidad al momento de ser gestionado, motivo por el cual su administración está a cargo del GE central, y su entorno de aplicación será sobre todos los niveles jurisdiccionales. Por otra parte se efectuará su replicación en cada nivel jurisdiccional inferior para realizar consultas. En general el DD debe formar parte de un repositorio nacional o estadual por lo que se debe asegurar la conectividad en todo momento ya que afectaría a otros componentes reutilizables del modelo universal de datos, como por ejemplo a las Tablas de Convalidación, al Catálogo de Localidades o Agrupaciones Poblacionales, al Administrador de Expedientes Físicos y/o Digitales, a la Estructura de Entidades Organizativas, etc.
- CNV (Tablas de Convalidación): el conjunto de tablas de validación debe ser propagado desde un nivel jurisdiccional de menor nivel a un nivel superior, permitiendo la definición de tablas de uso limitado a un departamento u organización gubernamental definida. Asimismo su compleja administración será realizada por el gobierno electrónico central o por la entidad organizativa que la creó. La pérdida de conectividad a las CNV afecta a todos los componentes reutilizables que utilizan Tablas de Convalidación.
- SI (Seguridad Integrada): Este componente valida la legitimidad del usuario. Es creado y administrado por cada unidad organizativa, es un componente típicamente local, la falta de conectividad afecta a todos los componentes reutilizables y aplicaciones usadas en el modelo.
- AP (Catalogo de Localidades o Agrupaciones Poblacionales): Este componente es creado y administrado por los organismos estatales con un nivel jurisdiccional determinado, por ejemplo las Direcciones de Estadísticas y Censos provinciales o municipales. En general, su ámbito geográfico es determinado por el nivel jurisdiccional, con posibilidad de replicar información parcial hacia un nivel jurisdiccional superior; tal sería el caso del nivel municipal al provincial. La propagación de los datos debe ser a un nivel jurisdiccional superior, para el que se está generando la información, pero se debe permitir que cada organización, del nivel jurisdiccional inferior, pueda acceder a los datos.
- ADX (Administrador de Expedientes Físicos y/o Digitales): Este componente afecta a todos los organismos de la Administración Pública que administran trámites o servicios que generan expedientes físicos o digitales, su creación y mantenimiento está a cargo del administrador de cada centro departamental, perteneciente a cualquier nivel jurisdiccional.
- EO (Estructura de Entidades Organizativas): Estos componentes serán creados y administrado por el administrador de cada unidad o entidad organizacional, en su respectivo entorno y nivel jurisdiccional. Sus datos deben ser propagados a todo nivel jurisdiccional superior, pero también permite definir estructuras organizativas de uso limitado de la unidad operacional.
- RPN (Repositorio de las Personas Naturales): Este componente es creado y administrado por cualquier organismo estatal de cualquier nivel jurisdiccional. Por ejemplo la Dirección Provincial del Registro Civil y Capacidad de las Personas utiliza el modelo de repositorio, cuyos registros básicos son generados localmente para ser posteriormente agregados al repositorio nacional. La falta de conexión con este repositorio afecta a todas las aplicaciones de la Administración Pública, en los diferentes niveles jurisdiccionales que administran personas.
- ADOC (Administrador de la Documentación Entrante y Saliente de una Institución): Está creado y administrado por cada entidad organizativa y su ámbito de aplicación es en la misma organización estatal, pero no se propaga a otros niveles jurisdiccionales, su administración es de alta complejidad por el gran volumen de datos a manejar.
- CTS (Catalogo de trámite y Servicios del Estado): Este componente debe pertenecer a un reservorio general y propagado en todos los niveles jurisdiccionales, pero debe permitir definir trámites y servicios de uso limitado a cada unidad de organización gubernamental.
  La falta de conectividad afectaría a todas las aplicaciones de la Administración Pública que administran trámites o servicios.
- SDL (Administración de Sesiones de Órganos Deliberativos): Este componente es creado y administrado por cada entidad organizativa u organismo gubernamental perteneciente a un nivel jurisdiccional determinado, municipal o provincial, y no propaga a niveles superiores.

La ausencia de conectividad afecta a todas las aplicaciones de la Administración Pública con procesos deliberativos.

7. La Importancia de la Replicación de Bases de Datos como Medio Tecnológico de Propagación de Cambios en un Modelo de Componentes Reutilizables para GE

Teniendo en cuenta lo expuesto precedentemente, se torna indispensable implementar la replicación entre Base de Datos a fin de propagar los datos en todos los niveles jurisdiccionales que posea el Estado. Raúl Monge [8] define la replicación como "el proceso de copiar y de mantener los objetos de la base de datos en las múltiples bases de datos que incorporan un sistema de base de datos distribuida", es decir que la información de una base de datos la tenemos disponible en más de un nodo, tanto en el gobierno central como en los nodos llamados "esclavos".

Esta situación presenta innumerables beneficios [9] tales como:

- Rendimiento: ya que al tener varios nodos se procesa la consulta en los mismos, no dependiendo de una sola base de datos central.
- Versatilidad en caso de falla: ya que al producirse un error en el nodo maestro pero, al estar este sincronizado con los nodos esclavos, cualquiera de estos puede reemplazarlo; logrando de esta manera cluster de alta disponibilidad.
- Fiabilidad por la seguridad: porque como se realiza la replicación en más de un nodo esto puede evitar la generación de bloqueos.

Es importante analizar la forma en cómo se realizará la replicación, utilizando los siguientes casos:

- Componentes reutilizables que están ubicados en el gobierno central, nacional, estadual o de mayor nivel jurisdiccional y se los debe propagar hacia los Centros Informáticos de niveles jurisdiccionales de nivel inferior, ya sea provincial o municipal. Un ejemplo del mismo es el Diccionario de Datos (DD) el cual debe ser manipulado en el gobierno central y propagado a los niveles jurisdiccionales de nivel inferior, ya que el mismo tiene que ser administrado en forma centralizada. La misma definición de DD afirma que "es un medio centralizado poseer información sobre los atributos lógicos de los datos de la Base de Datos" [10] por la necesidad y las ventajas de tener un modelo universal de datos para toda la organización. Por el contrario si no existiera un único modelo de datos se producirían continuamente problemas en el diseño de las tablas, un ejemplo sería una tabla con distinto diseño en el nodo central y en el nodo esclavo con los innumerables inconvenientes que esto podría acarrear.
- Componente reutilizables que se replican desde los Centros Informáticos de niveles jurisdiccionales de nivel inferior hacia los servidores del gobierno central, un ejemplo de estos son los que forman la Seguridad Integrada (SI), tales como los usuarios del Estado – En este caso interviene el concepto de autenticación, el que se define como "el proceso mediante el cual se confirma que quien se conecta y solicita acceso a un servicio es realmente quien dice ser, es decir, el legítimo usuario" [11] los cuales son componentes locales de cada departamento.
- Componentes reutilizables que no son necesarios replicar porque son típicamente locales tales como el administrador de expedientes físicos y/o digitales (ADX), el administrador de la documentación entrante y saliente de una institución (ADOC), el administrador de sesiones de los órganos deliberativos (SDL) y el generador de documentos institucionales (GDI). Esto de ninguna manera afirma que no es posible la consulta del gobierno central a los Centros Informáticos de niveles jurisdiccionales de nivel inferior o viceversa, simplemente afirma que no es necesaria la replicación, lo cual libera recursos en los servidores correspondientes.
- Finalmente el cuarto caso son los componentes reutilizables que deben ser replicados en ambos sentidos tales como las tablas de convalidación (CNV) actualizando el catálogo central o departamental, teniendo cuidado en las aplicaciones descentralizadas de brindar la posibilidad de la creación de tablas locales y de solo poder incorporar valores en las tablas centrales, nacionales o estaduales. Otros componentes son el catálogos de localidades o agrupaciones poblacional (AP) en donde la replicación de los datos debe ser en ambos sentidos pero es conveniente que sea validada o controlada por algún organismo en particular; la estructura organizativa de la organización sea nacional o de niveles jurisdiccionales de nivel inferior, el repositorio de las personas naturales (RPN) y el catálogo de tramites o servicios del estado (CTS), en estos tres últimos componentes debe replicarse íntegramente en la base de datos central o nacional mientras que en sentido contrario puede ser selectivo.

En relación al concepto de replicación surgen ciertos aspectos técnicos que deben ser tenidos en cuenta, considerando que las bases de datos actuales utilizan el modelo relacional o modelos basados en éste, y que el modelo relacional exige que todas las tablas tengan una clave primaria única que identifique unívocamente cada tabla [12].

Se debe plantear una solución y la alternativa técnica para evitar el problema de duplicación de las claves primarias (por ejemplo dos localidades de distintos estados con el mismo número único que lo identifica); es por todo ello que se plantea la

generación de semillas en un campo autonumérico, donde cada semilla identifica además al gobierno central o cada uno de los Centros Informáticos de niveles jurisdiccionales de nivel inferior (por ejemplo el gobierno central inicia la primer localidad en 1001, la segunda en 1002 y así sucesivamente, mientras que el primer Centros Informáticos de niveles jurisdiccionales de nivel inferior inicia en 2001, la segunda en 2002 y así sucesivamente con los demás Centros Informáticos de niveles jurisdiccionales de nivel inferior.

8. Conclusiones

En el presente trabajo se puede apreciar la importancia que reviste la arquitectura de aplicaciones en un modelo de servicios reutilizables para gobierno electrónico, especialmente en lo relacionado con los criterios fundamentales que debe de cumplir la modelación de la citada arquitectura y que se encuentra relacionada con: la comprensibilidad de los modelos, la que documenta la lógica información utilizada para su elaboración; la definición de las aplicaciones la cual debe ser completa, única y sin traslape innecesario y finalmente la definición de las aplicaciones que debe ser estable, por lo que no debe depender de factores tecnológicos, de diseño, técnico o de las características individuales del usuario final.

Por otra parte se plantea la importancia que revisten procesos relacionados con trámites y servicios que brinda el estado, en un contexto de gobierno electrónico. En este marco de trabajo, se puede observar que la creación y utilización de componentes reutilizables son esenciales para lograr una adecuada utilización de las denominadas funcionalidades atómicas, bajo el concepto de SOA. Estos componentes han sido desarrollados para publicar sus operaciones internas en forma de Servicios Web, para ello hay que poner un énfasis especial en el seguimiento de la interacción de dichos componentes.

Además resulta indispensable plantear la necesidad de optimizar la administración, conectividad y propagación de datos por parte de los componentes reutilizables lo cual permitirá una adecuada gestión de los datos que pueden ser generados desde los niveles jurisdiccionales más bajos, como es el caso de municipio y provincia o estados, hacia los niveles jurisdiccionales más altos ya sea a nivel nacional o estadual. Para ello es necesario utilizar la replicación de bases de datos como medio tecnológico de propagación de cambios en un modelo de componentes reutilizables para GE, desde los niveles jurisdiccionales más bajos hacia los más altos, utilizando el concepto de repositorio único, el cual forma parte de la arquitectura de datos del modelo de servicios reutilizables para GE.

9. Referencias

[1] Chaix Y., "Propuesta de arquitectura para los gobiernos municipales electrónicos", 2008. Disponible en: http://www.gobenic.gob.ni/eventos/gobierno-digitalmunicipal/Propuesta_de_arquitectura_para_el_gobierno_electronico_municipal.V2.pdf/view. Última visita en Julio de 2014.

[2] Castro M, Farfán J, Sánchez Rivero D, Castro D., Cándido A, Lombardo D.: "Tic: sistematizando el proceso de gobierno electrónico". Actas del III SIE y 38 JAIIO. Mar del Plata. Argentina, (2009).

[3] Sánchez Rivero D., Castro M., Reinoso E., Aparicio M., Aragón F., Cazón L., Zapana J.: "Aplicando SOA en Gobierno Electrónico", Actas de las VIII Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA, 2012 (ISSN 1853-7871), (2012).

[4] Castro M., Sánchez Rivero D., Farfán J., Castro d., Cándido A., Vargas A.: "Aplicación de Servicios Web SOAP y REST para funcionalidades existentes en sistemas informáticos provinciales", VII SIE y 42° JAIIO, SADIO, Facultad de Matemática, Astronomía y Física de la Universidad Nacional de Córdoba, Córdoba, Córdoba, Argentina, (ISSN: 1850-2526), (2013).

[5] Jonas A., Montilva C., Arape N. & Colmenares J. A., "Desarrollo de Software basado en componentes". IV Congreso de Automatización y Control. Mérida. 12-14 Noviembre 2003.

[6] Bachmann F., Bass L., Buhman Ch., Comella-Dorda S., Long F., Robert J., Seacord R. & Wallnau K., "VolumeII: Technical concepts of component-based software engineering", 2nd edition. Technical report, Software Engineering Institute, Carnegie Mellon University, Julio 2000.

[7] Sametinger J. "Software engineering with reusable components". Springer Verlag, Agosto 1997.

[8] Monge, R. (2005), "Sistemas Distribuidos de Computación: Trabajo de Investigación", "Base de Datos Distribuidas: Replicación", Junio de 2005, Universidad Técnica Federico Santa María, Valparaíso, Chile. Disponible en http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r11294.PDF. Última visita en Julio de 2014.

[9] Wikipedia. (2014) "Replicación (informática)". Disponible en http://es.wikipedia.org/wiki/Replicaci%C3%B3n_(inform%C3%A1tica). Última visita en Julio de 2014.

[10] INACAP. (2002) "Manual Base de Datos 1", Dirección Área de Informática, Departamento de Ciencias de Computación, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Chile. Disponible en http://es.slideshare.net/squall835/man-base-datos-i. Última visita en Julio de 2014.

[11] INTECO – Instituto Nacional de Tecnologías de la Comunicación. (2012), "Medidas de Seguridad para transacciones ONLINE: Observatorio de la Seguridad de la Información", León, España, Disponible en http www.inteco.es/file/30d5U9m9KCNG7-0ggHlozQ. Última visita en Julio del 2014.

[12] Elmasri, R y Navathe, S. "Fundamentos de Sistemas de Bases de Datos". 5ta Ed., Madrid. Editorial Pearson Educación, (2007).

TDD - ¿Qué se le cuestiona? - Algunas respuestas a los principales cuestionamientos a la metodología

Ing. Pablo Andrés Vaca, Ing. Calixto Maldonado, Ing. Claudia Inchaurrondo, Ing. Juan Peretti, Ing. María Soledad Romero, Ing. Matías Bueno, Lic. Marcelo Cagliolo

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Resumen

El Test-Driven Development (TDD) como una metodología de diseño y construcción de software, cuenta con muchos beneficios y ventajas. A pesar de ello, se han realizado cuestionamientos sobre su utilidad. Se presenta y discute aquí una selección de dos de los más polemizados:

"Efecto de las pruebas unitarias en el diseño de software" y "Situaciones donde no es posible aplicar TDD". Se planteó la necesidad de estudiarlos contrastando evidencia a favor y en contra de la utilización de TDD con el objetivo de arrojar un poco de luz sobre ellos a fin de esclarecer el grado de validez de los mismos. Se concluye que ambos cuestionamientos hacia TDD parten de confundir su finalidad y se detalla por qué no son válidos. Este análisis permitirá estudiar y enmarcar este tipo de metodologías en la industria.

Palabras Clave

TDD, BDD, Pruebas Unitarias, Diseño, Software, Desarrollo de Software, Pruebas Automáticas, Programación Extrema, Desarrollos Ágiles, Unit Tests, Test Automation.

Introducción

En dos trabajos titulados "Beneficios y Desafíos para el Desarrollo de Software" [1] y "Test-Driven Development - Una aproximación para entender su utilidad en el proceso de desarrollo de Software" [2] se han expuesto los beneficios de la utilización de TDD y los problemas a los que se enfrentan los desarrolladores durante su implementación en diferentes proyectos.

A pesar de contar con muchos beneficios y ventajas, se han realizado cuestionamientos sobre la utilidad de TDD como metodología de diseño y construcción de software, por lo tanto nos hemos planteado la necesidad de estudiar algunos de esos cuestionamientos contrastando evidencia a favor y en contra de la utilización de TDD.

David Heinemeier Hansson, en un artículo publicado recientemente en su blog [3] y en conversaciones con Kent Beck y Martin Fowler, publicadas en el blog de este último [4] expone que TDD induce a daños en el diseño debido a la complejidad excesiva que produce la necesidad de generar maquetas de componentes y servicios que aún no existen. También expone que en algunos casos, las bases de datos se maquetan para lograr velocidad al momento de ejecutar las pruebas, o porque se quiere lograr un mayor aislamiento y que esto se traduce en falta de confianza por parte de los desarrolladores al momento de implementar TDD. En dicho blog, Hansson afirma que todos estos problemas se derivan de la implementación de numerosas

pruebas de unidad que realizan los fundamentalistas de TDD.

Martin Fowler y Kent Beck [4] relacionaron estos problemas más bien con decisiones de diseño que con TDD, y lo explicaron de diferentes maneras, una de las cuales es que TDD es un camino para obtener código más confiable, aunque no la única. Y que en muchos casos es necesario resignar un mayor acoplamiento en pos de una mejor cohesión del código. Por otro lado expresaron que TDD no está orientado a tener que maquetar todo, que existen otros caminos para eliminar las dependencias externas, como ser los resultados intermedios. Esto último puede verse como parte del resultado obtenido en un trabajo de Simo Mäkinen y Jurgen Münch [5] en el cual observaron que el acoplamiento era más alto en los casos que se aplicaba TDD, y en contrapartida la cohesión era superior.

Aunque como lo expresaron en su trabajo, estos resultados no eran concluyentes, porque en la mayoría de los casos estudiados no se medían estas propiedades del código. Esto también puede verificarse en la presentación basada en el trabajo realizado por David Janzen [6], en la cual se muestra que cuando se usa TDD se obtiene un mayor acoplamiento, pero también una mejor cohesión del código. Relacionado con lo expuesto anteriormente, Adnan Causevic, Daniel Sundmark y Sasikumar Punnekkat [7] encontraron que la ausencia de diseño o el diseño insuficiente es un factor que limita la implementación de TDD. A esto le suman otra limitante que es la experiencia de los desarrolladores. Los autores del presente trabajo ya habíamos señalado en [1,2] que la experiencia de los equipos de desarrollo es un aspecto importante en la utilización de TDD.

Por otro lado, Robert Martin planteó en su blog [8] situaciones en las cuales a su criterio es impráctico, inapropiado o directamente imposible la implementación de TDD. Él manifiesta que cuando el software interactúa con aspectos físicos del mundo real, por ejemplo una máquina, o el software dibuja algo en la pantalla, es poco práctico o imposible escribir primero las pruebas que verifiquen si el resultado es el esperado, y sugiere que en estos casos es mucho más aplicable el testing manual. En dicho blog, Martin expone un ejemplo sencillo para ilustrar esta situación, es el caso de un software que ante determinados eventos debe hacer sonar una campana o timbre en una máquina. Si bien es posible escribir una prueba que evalúe si el software envía la señal a la campana, si realmente se quiere saber que la campana emite el sonido y se desea automatizar esto, es preciso instalar un micrófono y escribir un software que evalúe lo que el micrófono recibe. Algo que resulta por demás complejo, teniendo en cuenta que para saber si la campana suena o no, solamente es necesario escuchar, es decir, probar en forma manual y no automática.

El equipo de investigadores que suscribimos este trabajo hemos podido apreciar hasta este momento, que hay dos cuestionamientos importantes a TDD. El primero plantea que al poner el acento en las pruebas unitarias y buscar que las mismas se ejecuten rápidamente, se deben desarrollar y crear servicios y objetos que emulan aspectos que están por fuera del código productivo propiamente dicho y que lo aíslan de interactuar con aspectos externos como las bases de datos. Según los cuestionamientos detectados estos servicios generar complicaciones de diseño que no aportan beneficios al software. El segundo, a su vez, expresa que no todo el software es posible probarse mediante pruebas automatizas, sino que existen pruebas que deben ser realizadas de forma manual. El objetivo que perseguimos entonces es aportar luz de manera crítica y profundizar sobre estos dos puntos con la finalidad de poner en claro el grado de validez de los mismos.

Elementos del Trabajo y Metodología

Para llevar a cabo esta propuesta se realizó una revisión sistematizada de la literatura existente. Se efectuó una consulta y revisión bibliográfica de publicaciones relevantes sobre el tema y otras anexas o secundarias. Para ello, se utilizó como soporte de guía un trabajo de un grupo de investigadores de Gran Bretaña [9] y otro trabajo conjunto de un grupo de investigadores españoles y croatas [10]. Además, se tomaron como modelos las guías que se proveen en el trabajo "PRESENTACIÓN POR ESCRITO DE LA REVISIÓN BIBLIOGRÁFICA" [11].

La búsqueda de información bibliográfica también incluyó la utilización de buscadores en la web. En esta tarea se utilizaron principalmente criterios de búsqueda definidos por palabras claves como: "TDD - Limitaciones a la implementación - Factores de éxito - Casos de éxito - TDD y el desarrollo de software - Test-driven development - Introducción al Test-Driven Development – Pros and Cons of TDD – Agile – Extreme Programming, XP – Detractores de TDD – Cuando no aplicar TDD – Deficiencias de TDD". El resultado de estas búsquedas fue una lista de documentos y blogs, sobre los cuales se realizó un trabajo de selección y recopilación de información. El criterio empleado consistió en realizar una lectura de cada uno de los textos con la finalidad de detectar aquellos que consideramos realizaban aportes significativos a nuestro trabajo. Hay que destacar que la mayoría de la documentación encontrada ha sido escrita por autores que son parte del desarrollo de esta tendencia, es decir, están involucrados en su difusión. Se ha encontrado escasa documentación que haga un estudio objetivo de los defectos de TDD o si existen inconvenientes en la aplicación de TDD en determinadas circunstancias, razón por la cual a esta información hemos adicionado algunos blogs online. En este caso se los inspeccionó y separó considerando para su clasificación los más recientes y aquellos que refieren a bibliografía conocida y que es utilizada en este trabajo, o que pertenecen a profesionales de la industria con destacada trayectoria en el medio.

Luego de esta revisión, selección y evaluación, se utilizaron los elementos resultantes como base y soporte para nuestro trabajo, permitiendo detectar posibles problemas o inconvenientes en la aplicación de TDD en el desarrollo de software expresados por algunos referentes de la industria. Partiendo de estos inconvenientes se buscó en el presente trabajo validar estos problemas planteados o por lo contrario, refutar los mismos y encontrar una explicación de por qué los problemas enunciados no son válidos.

Resultados y Discusión.

Las dos posturas expuestas, su análisis y nuestras experiencias recogidas a través de distintos desarrollos realizados nos permiten establecer que:

Del primer cuestionamiento, en donde lo importante es la rapidez de ejecución de las pruebas unitarias, se desprende un aumento en la complejidad de la arquitectura obtenida y el diseño del software se ve así sumamente afectado.

El segundo cuestionamiento revela que existen situaciones en las cuales TDD no podría implementar las pruebas antes de escribir el código y por lo tanto no se puede aplicar. Esto fue puesto de manifiesto por Robert Martin en su blog [8] en el cual enuncia situaciones o escenarios en los cuales no se puede aplicar TDD en desarrollo de software. Tal como se ejemplificó en la introducción, él expone que esto se da principalmente cuando el software interactúa con el mundo físico.

Remitiendo a la definición misma de TDD, [12,13] la cual expresa que es una

práctica iterativa de diseño de software, podemos inferir que esta práctica no está orientada a las pruebas, más allá de su nombre. El espíritu de la metodología es utilizar las pruebas como un camino para diseñar el software, lo cual no significa que se deben tener que escribir todos los casos de pruebas necesarios para garantizar la calidad del software y tampoco implica que todas las pruebas deban ser automatizadas. Por lo tanto, no se debe pensar la misma como la implementación de pruebas unitarias solamente, sino que se pueden utilizar por ejemplo pruebas de integración para guiar el diseño del software, en pos de disminuir la complejidad del mismo. En cuanto a la implementación de maquetas para emular servicios u otros componentes de software con la finalidad de acelerar la ejecución, Kent Beck planteó que esto se puede reemplazar utilizando resultados intermedios [4].

En cuanto al cuestionamiento que TDD no se puede aplicar siempre, acordamos con Robert Martin que cuando el software interactúa con el mundo real, se pueden escribir pruebas que guíen el diseño y verifiquen que el software envía las señales a los dispositivos externos utilizando los drivers correspondientes. Y tal como expresamos en un párrafo anterior, no todas las pruebas deben ser automáticas.

Andrew Dalke en un trabajo publicado en su blog [14] referencia como un problema el hecho de que TDD no cubre todos los casos, en concordancia con otros trabajos [1,2,12,13], el mismo Dalke explica que TDD se usa para el desarrollo del código, por lo tanto los test de unidad que se escriben son sólo un subconjunto y no todos los tests necesarios para probar el software. No se debe olvidar que la escritura del código es sólo una parte del proceso.

Un común denominador que se aprecia en los cuestionamientos a TDD, es plantearlo como una herramienta para ejecutar las pruebas del software y verificar su calidad. Si se considera a TDD solamente como una metodología para realizar pruebas, se parte no sólo desde un punto equivocado en el camino de su implementación, sino que también se contradice la intención de los autores al momento de enunciarla. Esto genera dudas sobre si como metodología de diseño no necesita ser revisada y mejorada, desde el mismo nombre, el cual genera muchas confusiones, hasta en la forma de aplicarla.

Se ha apreciado que se hace mucho énfasis en utilizar herramientas de pruebas unitarias para implementar TDD, y no se ha reparado lo suficiente en que se pueden utilizar pruebas de integración y de sistema para guiar el desarrollo. Se observa que muchos de los planteos están referidos a que en los sistemas web actuales no se pueden escribir pruebas unitarias de las interfaces de usuario o que con la finalidad de que la ejecución sea veloz no se realizan pruebas para guiar el diseño de los componentes que interactúan con una base de datos. Esto está alejado de la realidad, los problemas surgen cuando se utilizan las herramientas inadecuadas para cada uno de los distintos trabajos. TDD no especifica en ningún momento qué herramientas utilizar, debido a esta falta de definición aparecen dudas al momento de determinar su selección. Este es un aspecto que deberá ser profundizado en estudios y trabajos posteriores.

Conclusiones

En la introducción planteamos algunos cuestionamientos que se le realizan a TDD, uno de los cuales pone en duda si podemos utilizarlo en los diferentes tipos de desarrollo de software, enunciando que no es viable para aquellos softwares que interactúan con el mundo real. El segundo cuestionamiento va más allá y formula que TDD no sería aplicable en todos los desarrollos de software, según su autor, y

que utilizarlo genera problemas en el diseño del software.

Ambos cuestionamientos se inician con un razonamiento lógico, pero por lo que se pudo determinar, los mismos son parcializados y parten de una definición errónea: que TDD es solamente para probar software. Según su definición [1,2,12,13], TDD es una metodología para diseñar software, no para probarlo, la calidad del software es una etapa posterior al desarrollo del mismo. Las pruebas desarrolladas utilizando TDD pueden ser un punto de partida para las pruebas posteriores, pero no deben ser tomadas como las pruebas del software. Se vale de herramientas que son comunes con las pruebas automáticas, pero ese es el único punto de contacto. TDD no explicita qué tipo de pruebas se deben hacer para guiar el diseño, sólo expresa que deben existir antes del código. Tampoco indica que se deben escribir todas las pruebas antes de escribir el código, sólo son necesarias aquellas pruebas que ayuden al diseño del software.

También se detectó que existe la necesidad de profundizar en la explicitación de TDD como metodología de diseño. Un aspecto que se encontró como posible oportunidad de mejora es definir apropiadamente qué tipos de pruebas se sugieren para guiar el desarrollo de los diferentes componentes de software, por ejemplo, formular además qué tipos de herramientas se sugieren para desarrollar componentes web de una aplicación o componentes de persistencia de datos. Esto requiere realizar pruebas y evaluaciones primero en ambientes controlados y luego en ambientes industriales.

A modo de cierre, los autores del presente trabajo podemos afirmar que los cuestionamientos hacia TDD tratados aquí no son válidos. Se originan al confundir la finalidad del mismo. Esto puede estar motivado por los enunciados generalistas que se publican sobre TDD y que dan lugar a diversas interpretaciones.

Y por último, algo que sí plantean estos cuestionamientos son dudas en cuanto a que si TDD necesita o no una reformulación, lo cual abre nuevos caminos para explorar buscando respuestas a aquellos puntos débiles que se han encontrado a través de este estudio presentado.

Referencias

[1] - Ing. Pablo Andrés Vaca, Ing. Calixto Maldonado, Ing. Claudia Inchaurrondo, Ing. Juan Peretti, Ing. María Soledad Romero, Ing. Matías Bueno - Test-Driven Development - Beneficios y Desafíos para el Desarrollo de Software. Universidad Tecnológica Nacional – Facultad Regional Córdoba.

[2] - Ing. Pablo Andrés Vaca, Ing. Calixto Maldonado, Ing. Claudia Inchaurrondo, Ing. Juan Peretti, Ing. María Soledad Romero, Ing. Matías Bueno - Test-Driven Development - Una aproximación para entender su utilidad en el proceso de desarrollo de Software. Universidad Tecnológica Nacional – Facultad Regional Córdoba.

[3] – David Heinemeier Hansson "TDD is dead. Long live testing. By David Heinemeier Hansson on April 23, 2014" - Abril 2014 – http://david.heinemeierhansson.com/2014/tdd-is-dead-long-live-testing.html.

[4] – Martin Fowler – Is TDD dead? Artículo en blog personal publicado el 09-09-2014 - http://martinfowler.com/articles/is-tdd-dead/.

[5] - Simo Mäkinen and Jürgen Münch - “Effects of Test-Driven Development: A Comparative Analysis of Empirical Studies” - Universidad de Helsinki.

[6] – David Janzen - “Sobre la influencia de TDD en el diseño de Software” - PhD en Ciencias de la Computación - Conferencia sobre Ingeniería de Software. Universidad de Kansas.

[7] - Adnan Causevic, Daniel Sundmark, Sasikumar Punnekkat - Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review - Mälardalen University, School of Innovation, Design and Engineering, Västerås, Sweden. 2010.

[8] – Robert Martin y otros – When TD doesn’t Work - Artículo en blog personal publicado el 30 de abril 2014 - http://blog.8thlight.com/uncle-bob/2014/04/30/When-tdd-does-not-work.html.

[9] - Barbara Kitchenham (a), O. Pearl Brereton (a), David Budgen (b), Mark Turner (a), John Bailey (b), Stephen Linkman (a) - (a): Software Engineering Group, School of Computer Science and Mathematics, Keele University, Keele Village, Keele Staffs. ST5 5BG, UK. (b): Department of Computer

Science, Durham University, Durham, UK - "Systematic literature reviews in software engineering – A systematicliterature review". 2009.
[10] - Zlatko Stapic, Vjeran Strahonja -"Performing Systematic Literature Review in Software Engineering" Facultad de Organización e Informática, Universidad de Zagreb. Eva García López, Antonio García Cabot, Luis de Marcos Ortega, Departamento de Ciencia de la Computación, Universidad de Alcalá -. 2012.
[11] - Fernando Aranda Fraga - "PRESENTACIÓN POR ESCRITO DE LA REVISIÓN BIBLIOGRÁFICA" - Secretaría de Ciencia y Técnica - Universidad Adventista del Plata.
[12] Kent Beck, "Extreme Programming Explained: Embrace Change", Addison-Wesley Professional, 1999. ISBN: 978-0321278654
[13] Carlos Fontela - Trabajo Final de Especialización en Ingeniería de Software. "Estado del arte y tendencias en Test-Driven Development" - Junio de 2011.
[14]Andrew Dalke - "Problemas con TDD" - publicado en blog personal el 29 de dicembre de 2009.
http://dalkescientific.com/writings/diary/archive/2009/12/29/problems_with_tdd.html. .

Una Guía Para El Investigador Sobre Derechos De Autor

Litvak, Claudia S.
Universidad Nacional de La Matanza, DIIT
Universidad Nacional del Oeste, Escuela de Informática
clitvak@ing.unlam.edu.ar

De Giusti, Marisa
Universidad Nacional de La Plata, PREBI-SEDICI
Comisión de Investigaciones Científicas de la Pcia. De Buenos Aires (CICBA)
marisa.degiusti@sedici.unlp.edu.ar

Abstract

Todas las legislaciones del mundo referidas a propiedad intelectual resguardan el derecho de un autor sobre su obra, excepto que él mismo lo ceda con carácter exclusivo. A esta lógica no escapan los investigadores, quienes en el sistema científico tienen la necesidad de publicar a pedido de los organismos de investigación y/o las universidades. Paralelamente existe un fuerte desconocimiento del tema de derechos, lo cual hace que los investigadores, convertidos en autores, no resguarden adecuadamente sus derechos. Este trabajo pretende ser una guía sencilla al presentar una orientación sobre los distintos tipos de derechos que constituyen el derecho de autor, las diferentes opciones de publicación y los modos de proteger los derechos de autor en el ámbito de la investigación. Entre ellas se destacan las posibilidades de publicación en acceso abierto. También presenta el tratamiento de los derechos en los repositorios existentes en nuestro país a la fecha de realización del mismo.

1. Introducción

Cuando un investigador genera el resultado de su investigación en algún formato, esto conforma la producción intelectual de ese investigador, la cual será publicada a través de diferentes medios y editores.

Pero a quién pertenecen los derechos de autor de su producción intelectual, es un tema que los investigadores suelen descuidar. La obligatoriedad de publicar y la importancia de las editoriales que están detrás de las principales revistas del mundo, hacen perder de vista la importancia de conservar los derechos de autor.

Se suele publicar sin leer las restricciones del editor seleccionado, y sin conocer incluso que, tales condiciones podrían modificarse de manera consensuada.

Dependiendo de las políticas de cada editor, un autor puede hasta tener prohibido colocar un enlace a su trabajo en su propia página web.

Hoy en día, es fundamental considerar la posibilidad de brindar acceso abierto a través de Internet a los resultados de las investigaciones a partir del depósito en repositorios temáticos o institucionales de acceso abierto, que ofrecen acceso público e irrestricto a las obras.

Es por ello que se generó esta breve guía, que pretende orientar al investigador para que conozca las opciones que posee respecto a sus derechos de autor y pueda seleccionar los sitios de publicación con total conciencia de los derechos que cede en cada caso.

Entre las alternativas de publicación se resalta la trascendencia de publicar en abierto. Estudios previos han mostrado la importancia de la difusión en abierto de las obras [1-5].

Otros estudios presentan la propuesta de generar y diseminar el conocimiento y la información científica, tanto en el país como a nivel internacional, publicando en abierto [6].

Conocer sobre los derechos de autor, las relaciones existentes entre autor-editor y editor-autor, así como los temas de acceso abierto y depósito en repositorios, facilitará el cuidado de la propiedad intelectual de cada investigador.

2. Derechos de Autor: Producción Intelectual del Investigador

Cuando un investigador desarrolla un artículo científico, cualquiera sea la especialidad del mismo, la ley lo reconoce como autor y titular de los derechos de propiedad intelectual sobre la obra. Entonces, es el investigador quien es propietario de los derechos de autor y quien puede controlar todos los usos que se harán sobre la obra publicada.

2.1. Tipos de derechos de autor

La legislación Argentina separa los derechos del autor en derechos morales y patrimoniales. La mayoría de las legislaciones del mundo mantienen esta misma clasificación con excepción de las basadas en el derecho de autor inglés, es decir, Inglaterra y EEUU. La legislación anglosajona no considera la existencia de derechos morales.

2.1.1. Derechos morales: Otorgan al autor el derecho a ser reconocido como tal y son intransferibles, y algunos no caducan, como el reconocimiento de autoría e integridad. Estos derechos no existen en todas las legislaciones pero sí en la Ley.

2.1.2. Derechos patrimoniales o de explotación (copyright): Incluyen la publicación, distribución, reproducción, comunicación pública, traducción y transformación, así como la posibilidad de autorizar a otros a que ejerzan cualquiera de estos derechos. Los derechos de explotación son transferibles, es decir, pueden cederse en forma exclusiva o no exclusiva a editores o a lectores / usuarios. Esta clase de derechos tiene una vigencia temporal acotada y caducan luego de pasados muchos años desde la muerte de todos sus autores. Por ejemplo en Argentina, EEUU y Europa el plazo de caducidad de los derechos patrimoniales de una obra es de 70 años luego de la muerte del último de sus autores.

La Figura 1 muestra un esquema referido a estos derechos.

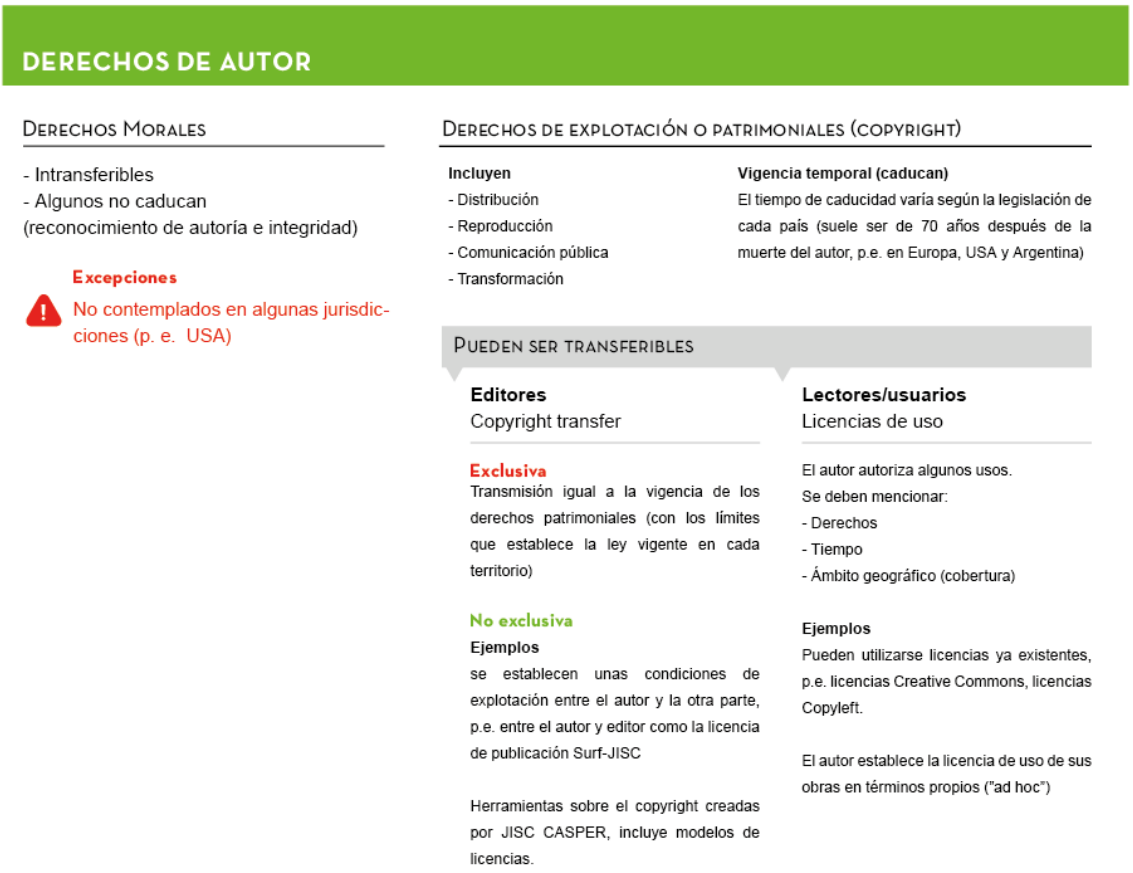


Figura 1. Derechos de autor

2.2. Relación editor - autor y autor - editor

Antes de publicar un artículo, los editores de revistas científicas, suelen pedir como requisito para la publicación, que se les cedan por contrato escrito los derechos de explotación sobre dicho artículo. Este contrato incluye entre sus cláusulas las condiciones de la cesión que se está realizando, el plazo de la misma, los responsables involucrados y los beneficios y obligaciones que recaen sobra cada uno.

Hay diferentes formas de ceder los derechos de explotación pero si la cesión fuera en exclusiva, el editor tiene el control total y exclusivo de la explotación de la

obra. Esta acción no tiene retorno ya que el autor habrá perdido estos derechos.

Debe considerarse que en el caso de existir acuerdos previos entre el autor y al institución financiadora en cuanto a sus trabajos, estos tendrán preeminencia.

Según las condiciones del acuerdo firmado con el editor, será la utilización que podrá hacer un autor de su propio artículo. Cuando un autor cede los derechos en exclusiva, deberá pedir permiso al editor para subir el artículo a su página web, subir una copia al campus virtual o repositorio de su universidad u otra entidad donde investiga, utilizar partes en otra obra, traducirlo, colocarlo en redes sociales, o en un repositorio temático.

2.3. Vías de publicación

Es el autor quien debe decidir dónde y cómo publicar sus investigaciones. Para ello debe conocer las diferentes vías, que son: a) Línea Roja o Paga, b) Línea Verde, c) Línea Dorada o Amarilla (Ver Figura 2) [7]. La o las vías de publicación que pueda elegir el autor estarán determinadas principalmente por la revista y editorial dónde publique.

2.3.1. Ruta Roja o Paga: consiste en publicar en revistas tradicionales a las que el público accede mediante pago por suscripción o por artículo. En este caso el autor envía un escrito que es revisado y luego de aceptado es publicado. Como ya se explicó, si la revista posee de forma exclusiva los derechos de explotación, el autor deberá pedir permiso para cualquier uso que quiera realizar con su artículo.

2.3.2. Ruta Verde: es el camino más simple y abierto, el autor deposita en un repositorio institucional o temático sus publicaciones para que sean accedidas de forma pública y gratuita. Según el contrato firmado con la editorial, el autor podrá depositar la obra final publicada, la revisada o el preprint sin revisión. (Ver Sección 2.5).

2.3.3. Ruta Dorada: corresponde a Revistas que exponen todos o algunos de sus artículos bajo el modelo de Acceso Abierto, es decir, accesibles pública y gratuitamente (Ver Sección 2.4). Se dividen en revistas genuinas e híbridas. Las revistas genuinas permiten ver todos los artículos sin cargos de suscripción para el lector. Las revistas híbridas permiten ver algunos artículos gratuitamente, pero en otros artículos el lector debe pagar.

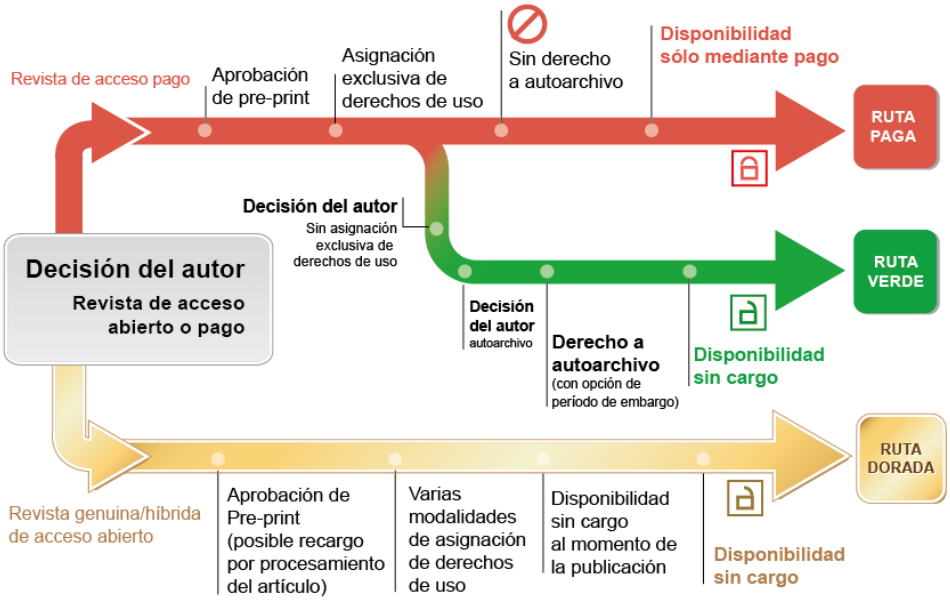


Figura 2. Vías de publicación

2.4. Iniciativa Acceso Abierto (Open Access)

El movimiento de Acceso Abierto (del inglés Open Access) tiene como objetivo dar acceso libre y gratuito a la producción científica para hacerla accesible sin restricciones y preservarla digitalmente como un bien común para la sociedad del presente y del futuro. El movimiento de Acceso Abierto comenzó formalmente en diciembre de 2001 en una reunión organizada en Budapest por la Open Society Institute, que generó la Declaración de la Iniciativa de Acceso Abierto, en febrero de 2002 [8]. Esta declaración fue luego complementada por las declaraciones de Bethesda en junio de 2003 [9] y la de Berlín [10] en octubre del

mismo año. En los años venideros, otras instituciones, redes y consorcios generaron declaraciones similares en diversos ámbitos y en apoyo al movimiento de acceso abierto

Gracias a iniciativas como Acceso Abierto, muchos editores permiten a los autores, realizar actos de comunicación pública de sus obras. Esto incluye poner copias de sus artículos en páginas web personales, y/o depositarlos en archivos digitales que se denominan, como ya mencionamos, repositorios (Ver Sección 2.5). Entonces podrán estar accesibles en abierto a través de internet.

2.5. Repositorios

Los dos tipos principales de repositorios son los institucionales y los temáticos.

Los repositorios institucionales contienen toda la producción científica de una universidad o centro de investigación. En realidad es toda la producción que quiera poner una institución u organización.

Los temáticos están especializados en una o varias disciplinas, y dan acceso a obras de autores de diferentes instituciones.

Cuando un autor realiza la acción de agregar su obra en un repositorio dicha acción es llamada autoarchivo. Pueden autoarchivarse las versiones preprint, que no ha sido revisado o el postprint, es decir, el artículo una vez que ha pasado la revisión por pares. En este caso puede ser la versión del autor o la del editor en pdf, según permita el contrato firmado con la editorial.

2.6. Adendas

Si un autor no está de acuerdo con las condiciones que establece un editor, o simplemente no puede aceptar las condiciones impuestas porque su institución se lo impide, puede solicitar la incorporación de una adenda al contrato firmado con la editorial. El término adenda proviene del latín y quiere decir “Las cosas que se han de añadir”. Allí se describen los derechos que un autor quiere retener.

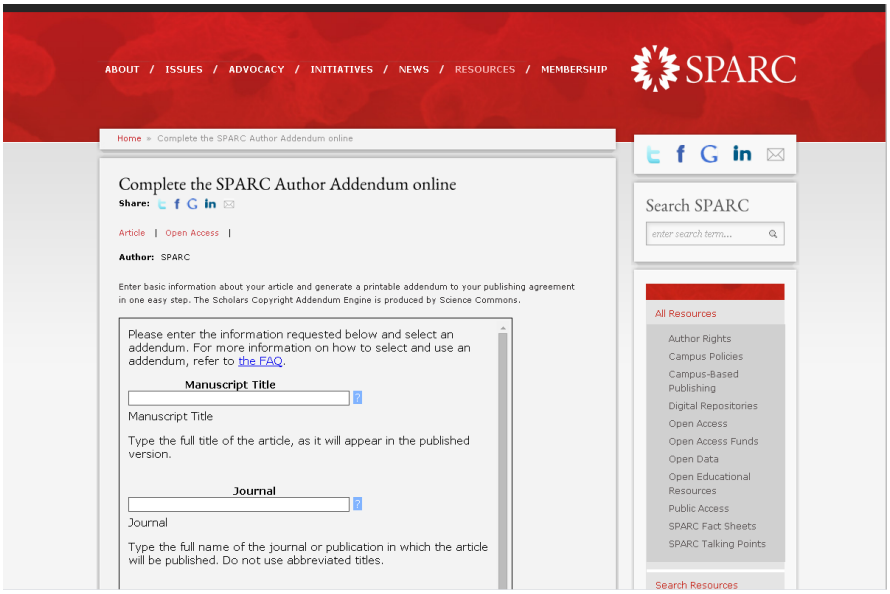


Figura 3. Generación de Adendas en la página web de Sparc

Existen modelos de adendas e incluso algunos de ellos pueden generarse en línea. Para la generación de adendas en línea se puede visitar la página de SPARC [11] (Ver Figura 3). Hay editores que aceptarán las adendas y otros que no, por lo que para mantener sus derechos el autor se verá forzado a cambiar el editor.

2.7. Licencias Creative Commons

Cuando el autor publica una obra bajo el modelo de acceso abierto puede regular el uso posterior de la misma bajo los términos de una licencia de uso, por ejemplo las licencias Creative Commons [12-13]. Estas licencias indican al lector los usos que tiene permitido realizar con la obra y pueden aplicarse a obras inéditas o previamente publicadas por el mismo autor, siempre y cuando cuente con los derechos de propiedad intelectual.

Cretive Commons es una organización internacional sin fines de lucro que ofrece 6 alternativas, que varían en cuanto al reuso que permiten de una obra. Es importante destacar que todas las licencias Creative Commons requieren la cita correcta al autor de la obra.

Estas licencias se componen a partir de la combinación de los siguientes criterios:

Atribución: En cualquier explotación de la obra autorizada por la licencia será necesario reconocer la autoría (es condición obligatoria en todas las licencias).

Uso Comercial: Se permite la explotación de la obra libre o se limita usos no comerciales.

Generación de obras derivadas: Se autoriza al lector a crear nuevas obras basadas en la presente.

Compartir Igual: Se debe compartir las nuevas obras que deriven de la presente utilizando la misma licencia Creative Commons o no.

Basándose en las condiciones anteriores se pueden generar las *seis* combinaciones que producen las licencias Creative Commons:

Atribución (*by*): Se permite cualquier explotación de la obra, incluyendo la explotación con fines comerciales y la creación de obras derivadas, la distribución de las cuales también está permitida sin ninguna restricción. Esta licencia es una licencia libre según la Freedom Defined.

Reconocimiento – Compartir Igual (*by-sa*): Se permite el uso comercial de la obra y de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original. Esta licencia es una licencia libre según la Freedom Defined.

Atribución – No Comercial (*by-nc*): Se permite la generación de obras derivadas siempre que no se haga con fines comerciales. Tampoco se puede utilizar la obra original con fines comerciales. Esta licencia no es una licencia libre.

Atribución – No Comercial – Compartir Igual (*by-nc-sa*): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original. Esta licencia no es una licencia libre.

Atribución - Sin Obra Derivada (*by-nd*): Se permite el uso comercial de la obra pero no la generación de obras derivadas. Esta licencia no es una licencia libre.

Atribución – No Comercial – Sin Obra Derivada (*by-nc-nd*): No se permite un uso comercial de la obra original ni la generación de obras derivadas.

Para informarse sobre estas licencias, el autor puede visitar la página de Creative Commons [12] o Creative Commons Argentina [13].

Para crear una Licencia Creative Commons se puede visitar en la página web del sitio, la opción de Licencias (Ver Figura 4).

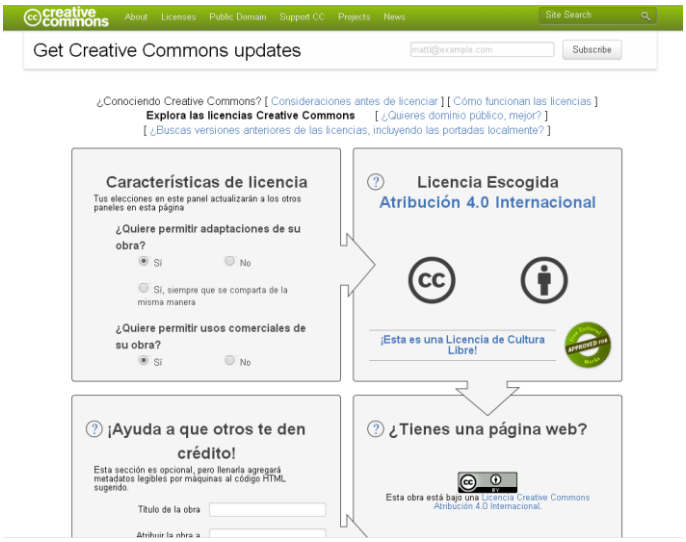


Figura 4. Licencias Creative Commons

2.8. Recomendaciones para el uso de Licencias Creative Commons

El licenciamiento de las obras intelectuales es un acto importante y requiere que el creador sepa qué hace y cómo libera sus obras. Por ello, es muy recomendable que el autor se informe adecuadamente.

Las licencias Creative Commons se otorgan a perpetuidad y son no-exclusivas: una vez que ha publicado su obra bajo esta licencia, la liberación del uso, copia, distribución y ejecución de la obra se da a perpetuidad y a personas indeterminadas. Si bien el autor puede retirar o cambiar la licencia esto no afectará a quienes tomaron la obra bajo las condiciones iniciales de uso.

3. Repositorios en Argentina y en el resto del mundo

Se presentarán a continuación, los repositorios existentes en Argentina (a la fecha agosto de 2014). En las siguientes secciones se mostrará información sobre los repositorios en América del Sur y en el resto del mundo.

3.1. Repositorios en Argentina

La Ley 26899 de Creación de Repositorios Digitales Institucionales de Acceso Abierto, Propios o Compartidos fue sancionada el 13/11/2013 [14], establece que las instituciones del Sistema Nacional de Ciencia y Tecnología y que reciban financiamiento del Estado Nacional, deben crear repositorios digitales institucionales de acceso abierto y gratuito en los que se depositará la producción científico tecnológica nacional.

Para conocer los diferentes repositorios, puede investigarse en El Directorio de Repositorios de Acceso Abierto: *The Directory of Open Access Repositories – OpenDOAR* [15], en la página http://www.opendoar.org/ (Ver Figura 5). Allí puede realizar una búsqueda por repositorios, buscar repositorios por contenido, listar todos los repositorios, o ver las estadísticas de cada repositorio.

En la búsqueda por repositorios, a su vez pueden buscarse por país o por continente. Se presenta la información de nuestro país.

Argentina cuenta con un total de 33 repositorios, a la fecha agosto de 2014. El sitio dice 34 repositorios, pero se halla repetido el repositorio de la Universidad Católica Argentina.

Se presentará cada uno de ellos y un resumen de las condiciones de depósito y licenciamiento de las obras.

En aquellos repositorios que no se mencionan dichas condiciones es debido a que dicho repositorio no se ha definido aún al respecto.

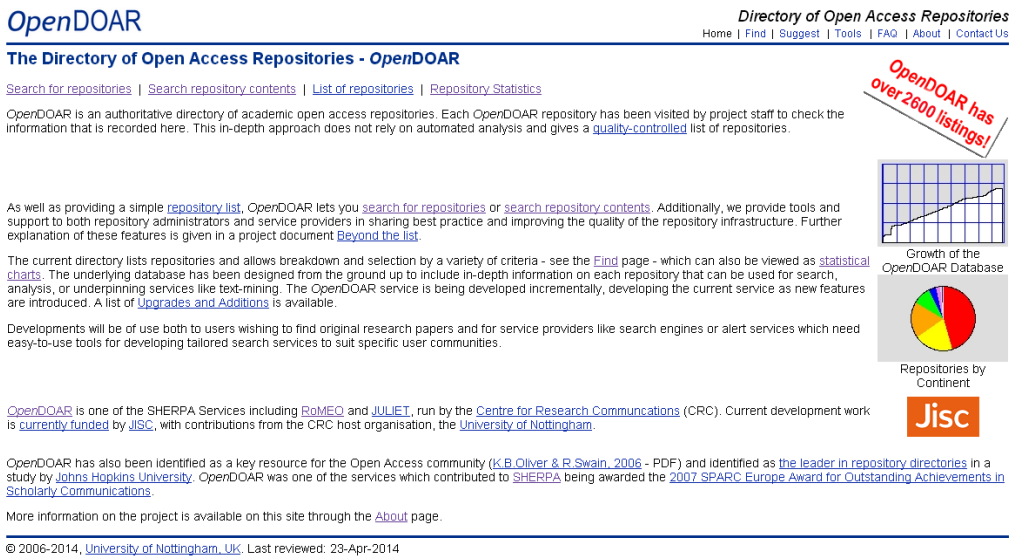


Figura 5. OpenDOAR

Ansenuza: Es el repositorio gratuito de acceso abierto, organizado por la Universidad Nacional de Córdoba, y el Ministerio de Educación de la provincia de Córdoba. El repositorio tiene como objetivo proporcionar material didáctico para todos los niveles educativos, con énfasis en las necesidades locales. Reciben el trabajo de los profesores universitarios y profesionales de la enseñanza. Todos los materiales están incluidos bajo una licencia

Creative Commons, y permite trabajos derivados, mientras que sólo en algunos se permiten usos comerciales. En todos los casos de obras derivadas, se deberá atribuir la autoría al autor de la obra original. La interfaz está en español.

Argos: Es el repositorio de la Facultad de Humanidades y Ciencias Sociales, Universidad Nacional de Misiones. Es el Repositorio Institucional del Ministerio de Investigación y Postgrado, que almacena y mantiene la información científica generada por el personal académico. La interfaz está en español y permite alertar a los usuarios de los nuevos contenidos.

Biblioteca Digital - Bolsa de Cereales: Este sitio proporciona acceso a la producción científica de la Bolsa de Cereales de Argentina. La interfaz está disponible en español. Es multidisciplinario, de Agricultura, Alimentación y Veterinaria; Ciencias Económicas y Empresariales. La interfaz está en español.

Biblioteca Digital Biblioteca Digital de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires (Biblioteca Digital FCEN-UBA) [16]. Este sitio proporciona acceso a la investigación y colecciones de la Facultad. La interfaz está en español. Su contenido es sobre ciencia en general y contiene artículos, tesis y elementos multimediales. Bajo el título Normativa, aparece el apoyo al movimiento de Acceso Abierto. Además mencionan la resolución CS No. 6323 del 13 de marzo de 2013, fecha de creación del Repositorio Digital Institucional de la Universidad de Buenos Aires, a cargo del SISBI, que cosecha los metadatos de los repositorios de las Facultades y ofrece una forma de ingresar a datos y documentos de las Facultades que no tengan repositorios. Se establece allí para la UBA el Acceso Abierto como política de apoyo e incentivo para elevar la visibilidad y el impacto de la producción cientifica-tecnológica institucional y en consecuencia, que los investigadores, docentes, becarios y estudiantes de posgrado cuya actividad haya sido financiada con fondos públicos, deben depositar o entregar a la Biblioteca para su depósito una copia digital de la versión final de su producción científico-tecnológica resultante del trabajo. Tiene además una solapa de *Autores y derechos* (Ver Figura 5), donde aparece una pestaña *Autorización*, que es la autorización que debe firmar un autor o tesista. Se ceden los derechos de publicación digital y gratuita, de modo no exclusivo autorizando los cambios de formato con fines de preservación.

Biblioteca Digital de la Biblioteca Nacional de Maestros: Este sitio proporciona acceso a las colecciones digitalizadas de la institución, incluyendo el trabajo de Leopoldo Lugones. La interfaz está disponible en español. Los temas contenidos son de Artes y Humanidades General, Derecho y Política.

Biblioteca Digital del programa Pro-Huerta (INTA-MDS) / Pro-Huerta's (INTA-MDS) Digital Library: Esta biblioteca digital contiene la producción de Argentina, el programa de producción y la seguridad alimentaria del Pro-Huerta. Contiene libros, gráficos y publicaciones audiovisuales. La interfaz está en español. El contenido corresponde a los temas Agricultura, Alimentación y Veterinaria; Ecología y Medio Ambiente.

Biblioteca Digital por la Identidad: Pertenece a la *Comisión Nacional por el Derecho a la Identidad (CONADI)* de Argentina. Este sitio proporciona acceso a contenidos digitalizados en relación con el derecho a la identidad y los derechos humanos en Argentina. La interfaz está en español. Los temas de esta biblioteca son Derecho, Política y Psicología. Permite seleccionar una colección, entre archivo fotográfico, recortes periodísticos, legislación y libros.

Biblioteca Digital Universidad Católica Argentina: Pertenece a *la Pontificia Universidad Católica Argentina*. Es el repositorio institucional que alberga la producción intelectual de la Universidad. Está integrado por seis colecciones: Tesis y Trabajos Finales seleccionados por cada Facultad, Revistas y Documentos de Investigación con el aporte de docentes e investigadores de las distintas Facultades, Ponencias presentadas en jornadas organizadas por la Universidad, Libros, editados por la Universidad o por editoriales externas y Contribuciones, colección que aporta material diverso de docentes de nuestra Universidad. La Biblioteca Digital adhiere a la filosofía del *acceso abierto a la literatura científica*, permitiendo descargar, distribuir, copiar e imprimir su material sin restricciones, asegurando así el cumplimiento su objetivo fundamental, que es el acceso irrestricto a la información, sin barreras económicas, legales o técnicas Es multidisciplinario y la interfaz está disponible en español. Tiene un link a Acceso Abierto.

Biblioteca Virtual UNL: Pertenece a la Universidad Nacional del Litoral. Este sitio proporciona acceso a la producción científica de la institución. Es multidisciplinario. La interfaz está principalmente disponible en español, sin embargo, las páginas de las tesis están disponibles también en inglés. La reutilización de metadatos está permitida para fines no lucrativos; y la reutilización de elementos de datos permitida también para fines no lucrativos.

Biblioteca para la Persona (Library for the Person): Pertenece a la *Asociación Civil Epimeleia Cuidando el Desarrollo Personal.* Trata los temas de salud, psicología y medicina. Este sitio proporciona acceso a la colección digitalizada de material relacionado con dicha asociación. Parte del contenido se mantiene en sitios web externos. La interfaz está disponible en español.

Cor-Ciencia: Pertenece al *Acuerdo de Bibliotecas Universitarias de Córdoba (ABUC).* Este es el primer repositorio creado por un acuerdo entre las bibliotecas universitarias. Contiene temas de investigación de ciencia

y tecnología de la provincia de Córdoba, y muchos artículos que no están a texto completo. Es multidisciplinario y en español. Dice expresamente "políticas no analizadas todavía".

FAUBA Digital: Pertenece a la *Facultad de Agronomía Universidad de Buenos Aires.* Proporciona el acceso abierto a los trabajos e investigaciones del personal y los estudiantes. La interfaz está disponible en una amplia variedad de idiomas. Los temas son Agricultura, Comidas y Veterinaria. Reutilización de elementos de datos permitida para fines no lucrativos.

LAReferencia - Red Federada de Repositorios Institucionales de Publicaciones Científicas Latinoamericanas. Presenta publicaciones de nueve países de América Latina: Argentina, Brasil, Chile, Colombia, Ecuador, El Salvador, México, Perú, Venezuela. Las licencias varían según los países. Las publicaciones de Argentina presentadas allí tienen todas licencias Creative Commons.

Memoria Académica: Pertenece a la *Facultad de Humanidades y Ciencias de la Educación (FaHCE), Universidad Nacional de La Plata(UNLP).* Este sitio proporciona acceso a los materiales de investigación y de aprendizaje de la institución, a ordenanzas y materiales administrativos. El contenido está disponible en formato HTML y PDF. La interfaz está en español y es un repositorio multidisciplinario. Memoria Académica utiliza la licencia Creative Commons Atribución-NoComercial-SinDerivadas 2.5 Argentina (**CC BY-NC-ND 2.5**), para todos sus documentos.

Naturalis: Pertenece a Facultad de Ciencias Naturales y Museo, *Universidad Nacional de La Plata (UNLP).* Este sitio proporciona acceso a la producción científica de la institución. La interfaz está disponible en español y es un repositorio multidisciplinario. Brinda algunos textos digitales completos, textos completos impresos que se encuentran en la universidad y en otros casos las referencias bibliográficas.

Nülan: Pertenece a *Facultad de Ciencias Económicas y Sociales, Universidad Nacional de Mar del Plata.* Es un portal de promoción y difusión pública del conocimiento académico y científico. Este sitio proporciona acceso a la producción académica de la institución. La interfaz está en español. Trata sobre los temas de las Ciencias Sociales en General; Gestión y Planificación; Empresariales y Económicas; Educación. Reutilización Comercial de Metadatos está permitida.

Producción Académica UCC: Pertenece a la *Universidad Católica de Córdoba.* Se trata de un repositorio institucional que permite el acceso al trabajo del personal, investigadores y estudiantes de la universidad. Incluye documentos de trabajo, artículos de revistas de la UCC, artículos en revistas publicadas por profesores externos, trabajos académicos de pregrado, maestría y tesis doctorales, trabajos de investigación, informes técnicos, fotos de modelos arquitectónicos, videos, conferencias y trabajos de taller. Trata sobre Salud y Medicina; Tecnología, Educación, Derecho y Política, Gestión y Planificación de Agricultura, Alimentación y Veterinaria; Química y Tecnología Química. Respecto a políticas del repositorio, explicita que no han sido definidas.

Red de Bibliotecas Virtuales de Ciencias Sociales de América Latina y El Caribean: Pertenece al Consejo Latinoamericano de Ciencias Sociales (CLASCO). Este sitio proporciona acceso a la producción científica del consorcio CLACSO, que abarca temas de ciencias sociales. El sitio también contiene un enlace a proyectos dentro del consorcio. La interfaz está disponible en español. CLACSO participó en el Comité de Expertos del MINCYT, quienes en mayo de 2013 aprobaron la ley de acceso abierto en Argentina., donde se redactó el proyecto de ley aprobado.

RPsico: Es el Repositorio de la *Facultad de Psicología, Universidad Nacional de Mar del Plata.* Este sitio proporciona acceso a la investigación de la facultad. La interfaz está disponible en español. Los temas tratados son Biología y Bioquímica, Ecología y Medio Ambiente, Educación, Psicología.

Repositorio de trabajos finales del Taller de Diseño Industrial (Cátedra Gálan) de la Carrera de Diseño Industrial: Pertenece a la *Fadu (Facultad de Arquitectura, Diseño y Urbanismo), Universidad de Buenos Aires (UBA).* Este sitio proporciona acceso a la producción científica de la institución. La interfaz está disponible en español y el repositorio es multidisciplinario. Reutilización Comercial de Metadatos permitida; reutilización de elementos de datos permitida para fines no lucrativos.

Repositorio Digital de la Universidad FASTA (REDI): Este es el repositorio institucional de la Universidad FASTA. La interfaz está en inglés y español. Es multidisciplinario. Todos los trabajos depositados en REDI deben continuar siendo propiedad de los autores. El REDI permite proporcionar licencias alternativas Creative Commons mediante el cual se puede liberar algunos de los derechos que se le asigne automáticamente por la ley de derechos de autor. Para agregar a REDI documentos previamente publicados se tomarán en cuenta las normas Sherpa/Romero y Dulcinea. En caso de no estar incluida la publicación dentro de estas normas se pedirá autorización a los editores para la publicación.

Repositorio Digital de la Universidad Nacional de Córdoba (RD-UNC): Pertenece a la *Universidad Nacional de Córdoba.* Es un espacio donde se almacena, organiza, preserva, provee acceso libre y procura dar visibilidad a nivel nacional e internacional, a la producción científica, académica y cultural en formato

digital, generada por los integrantes de la comunidad universitaria. Es multidisciplinario y la interfaz está disponible en español.

Repositorio Digital Institucional José María Rosa: Pertenece a la *Universidad Nacional de Lanús.* El Repositorio Digital Institucional "José María Rosa" contiene contenido académico, científico, institucional y administrativo de la Universidad Nacional de Lanús, incluyendo tesis de grado. La interfaz está en español. Los temas contenidos son Salud y Medicina; General de Tecnología, Artes y Humanidades General; Ciencias Sociales en General. Reutilización de elementos de datos permitida para fines no lucrativos, gozando de la protección que brinda la ley 11.723 de propiedad intelectual. Los documentos pueden ser visualizados y descargados en cualquier formato por cualquier persona, sin cargo económico alguno y sin necesidad de pedir permisos explícitos. La mencionada autorización comprende el uso de los documentos únicamente con fines educativos y de investigación. En ningún caso podrá usarse el material con fines de lucro. Toda vez que los documentos obtenidos en el repositorio se mencionen como referencia deberán ser citados enunciando los nombres de los autores, el título del documento y los detalles bibliográficos completos, así como su ubicación Web, o sea el enlace al registro en el repositorio. Prohíben modificar el formato digital, alterar los datos del autor y los del editor digital. Establecen que el contenido de los documentos no podrá ser traducido, adaptado o modificado en modo alguno, excepto con la expresa autorización del autor.

Repositorio Digital San Andrés: Pertenece a la *Biblioteca Max von Buch, Universidad de San Andrés.* Este sitio proporciona acceso a la salida de los estudiantes de la institución. La interfaz está disponible en español. Los temas contenidos son Matemáticas y Estadística; Artes y Humanidades Generales; Ciencias Sociales Generales, Ciencias Económicas y Empresariales; Educación. Trabajan bajo licencias Creative Commons.

RepHipUNR: Es el Repositorio Hipermedial de la *Universidad Nacional de Rosario.* Este sitio proporciona acceso a la investigación y la producción docente de la institución. Es multidisciplinario y la interfaz está en español.

Repositorio Institucional de la Universidad Nacional de Salta: Pertenece a la *Universidad Nacional de Salta (UNSa).* Es un repositorio institucional. Es multidisciplinario y la interfaz está en español. Trabajan bajo licencias Creative Commons.

Repositorio Institucional del Centro Atomico Bariloche y el Instituto Balseiro: Pertenece a la *Comisión Nacional de Energía Atómica.* Este sitio proporciona acceso a la producción científica de la institución. La interfaz está disponible en español e inglés. Trata sobre Ciencia en general y particularmente Energía Atómica. Trabajan bajo la iniciativa de acceso abierto, y en caso de artículos enviados previamente a una revista con referato, propone consultar los permisos de cada editorial respecto del autoarchivo/ autodepósito en el sitio de Sherpa/Romeo[19]

Repositorio Institucional del Ministerio de Educación de la Nación: Pertenece al *Ministerio de Educación de la Nación Argentina.* Este sitio proporciona acceso a la producción científica de la institución sobre temas educativos. La interfaz está disponible en español. Trata sobre Educación. Trabajan bajo licencia de Creative Commons Reconocimiento- NoComercial - SinObraDerivada 3.0.

Repositorio Institucional UCES: Pertenece a la *Universidad de Ciencias Empresariales y Sociales (UCES).* Este sitio proporciona acceso a la producción científica de la institución. Es multidisciplinario y la interfaz está en español. Presenta una opción para leer la información sobre los *Derechos de Autor*. Allí se indica "Los autores conceden una copia de su trabajo a la UCES para conservarla, difundirla y preservarla. La concesión de esta licencia no exclusiva permite a los docentes e investigadores continuar reteniendo sus derechos de autor para publicar en otros medios o disponer de su trabajo a voluntad. Le sugerimos para proteger su obra de usos indebidos suscribir una de las licencias Creative Commons que le permitirán autorizar el uso de la obra y que ésta continúe estando protegida. El autor escoge una de las licencias e identifica su obra con el símbolo CC y le adjunta la licencia de uso. Cuando los usuarios accedan al documento podrán identificar fácilmente cuáles son las condiciones que el autor ha establecido para su obra." Además presenta la posibilidad de leer sobre el depósito de una obra no publicada, donde explica en detalle las licencias Creative Commons.

Biblioteca Digital UNCuyo (Repositorio OAI Biblioteca Digital Universidad Nacional de Cuyo): Pertenece al *SID (Sistema integrado de Documentación), Universidad Nacional de Cuyo (UNCuyo)* [17]: Este sitio proporciona acceso a la producción científica de la institución. La interfaz está en español. Es Multidisciplinario; de Ciencia General, Salud y Medicina. Reutilización de Metadatos permitida para fines no lucrativos; reutilización de la información bibliográfica de datos permitidos para propósitos sin fines de lucro; las políticas de contenido definidos, políticas presentación definidos. En la página inicial presenta un link a *¿Cómo publicar?*, que permite imprimir los formularios de Creative Commons y los formularios para publicar Tesis de Grado y Posgrado, Libros, Informes de Investigación y Audiovisuales (Ver Figura 6).



Figura 6. Indicaciones de cómo publicar de la Biblioteca Digital UNCuyo

Repositorio Universidad de Belgrano: Pertenece a la Universidad de Belgrano. Este sitio proporciona acceso a la investigación y las colecciones de la Universidad. La interfaz está en español. Es multidisciplinario.

Scientific Electronic Library Online - Argentina (SciELO - Argentina) [18]*:* Pertenece a *CAICYT (Centro Argentino de Información Científica y Tecnológica).* Este sitio reúne una gran colección de publicaciones seleccionadas de revistas científicas argentinas. La interfaz del sitio está en inglés, español y portugués. Trata sobre Ciencias Generales, Salud y Medicina, y Ciencias Sociales. Las políticas de contenido están definidas; las políticas de presentación están definidas. *Scientific Electronic Library Online* es una biblioteca electrónica que conforma una red iberoamericana de colecciones de revistas científicas en texto completo y con acceso abierto, libre y gratuito. El proyecto *SciELO* está fundado en el desarrollo de una metodología modelo para la preparación, almacenamiento, diseminación y evaluación de la publicación científica en soporte electrónico. Es una iniciativa de *BIREME*, que desde sus inicios en 1997 cuenta con el financiamiento de la Fundación de Apoyo a la Investigación del Estado de São Paulo *(FAPESP).* A partir de 2002 se suma a este emprendimiento el *CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).* Argentina se integra al proyecto *SciELO* en junio de 2005 y en octubre de 2006 obtiene la certificación para operar regularmente como sitio oficial (Ver Figura 7). Aclaran en el sitio que "Los resultados obtenidos pueden ser impresos y/o salvados en formato HTML o ISO 2709. Debajo de cada una de las referencias obtenidas como resultado de las búsquedas se encuentran los enlaces a los formatos disponibles (HTML y/o PDF). En el formato HTML se brinda información acerca de cómo citar los artículos"

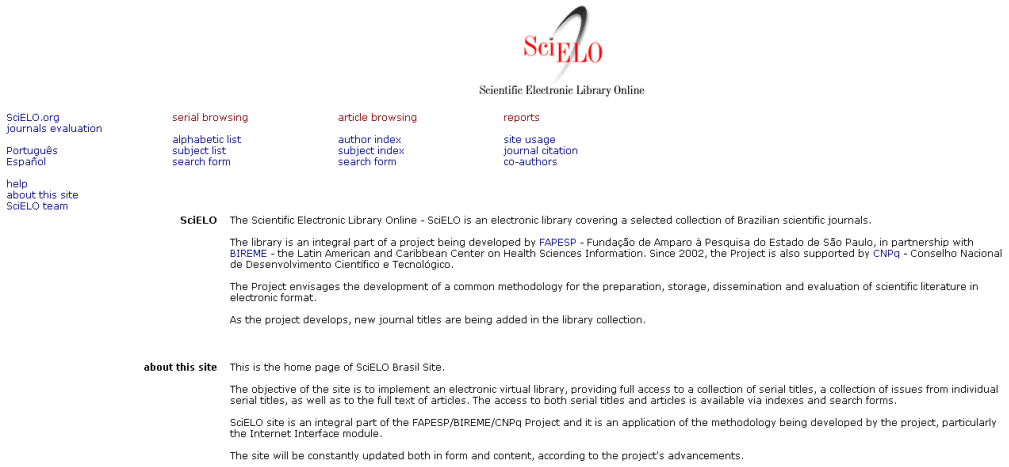


Figura 7. SciELO - Argentina

SEDICI (Servicio de Difusión de la Creación Intelectual): Pertenece a *Universidad Nacional de La Plata (UNLP)* [7]: Este sitio es un repositorio institucional que proporciona acceso abierto a la producción de investigación del personal de la institución y los estudiantes, los recursos históricos digitalizados, y otras publicaciones y recursos académicos que incluyen: postgrado, maestría y tesis de licenciatura, ebooks, artículos, objetos de conferencias, revistas, audio y más . La interfaz está disponible en Inglés, portugués y español.

Es multidisciplinario. Reutilización Comercial de Metadatos permitida; los Derechos varían para la reutilización de elementos de datos completos; las

políticas de contenido definidas; las políticas de presentación definidas. En la opción ¿Cómo cargo mis trabajos?, además de guiar al autor, muestran la *Licencia de distribución no exclusiva* y la elección de la licencia de uso pertinente (Ver Figura 8).

Las características de las licencias de depósito se especifican en http://sedici.unlp.edu.ar/static/resources/licencia_de_deposito.pdf.



Figura 8. SEDICI (UNLP)

3.2. Repositorios en América del Sur

Sudamérica cuenta con un total de 241 repositorios posibles. De esos repositorios Argentina tiene 33, Bolivia 2, Brasil 85, Chile 19, Colombia 34, Ecuador 25, Paraguay 1, Perú 25, Uruguay 2, y Venezuela 15.

3.3. Repositorios en el resto mundo

Como ya se mencionó anteriormente puede utilizarse la página de *OpenDOAR* [15] para revisar los repositorios de cada país (Ver Figura 4). En *OpenDOAR* se muestran las políticas de acceso [19] a sus ítems a texto completo en el conjunto de repositorios del mundo registrados en *OpenDOAR* [20]. Aunque existe un porcentaje desconocido, *OpenDOAR* reporta más de un 70% de los repositorios del mundo sin una política expresa de acceso a los contenidos a texto completo. Argentina se encuentra en cuanto a estas políticas en similares niveles que el resto del mundo.

Además la página de *SHERPA/ROMEO* [21] permite conocer las políticas de archivo de un listado de 1022 editores y las 19623 revistas. Se puede observar la categorización planteada por *SHERPA/ROMEO* que consiste en dar colores (Ver Figura 9):

Verde: Permite archivar el preprint y el postprint o publicar versión en pdf final.

Azul: Permite archivar y el postprint o versión final en pdf.

Amarillo: Permite archivar el preprint, la primera revisión del trabajo antes de la revisión por pares.

Blanco: No permite archivar o no está definido.

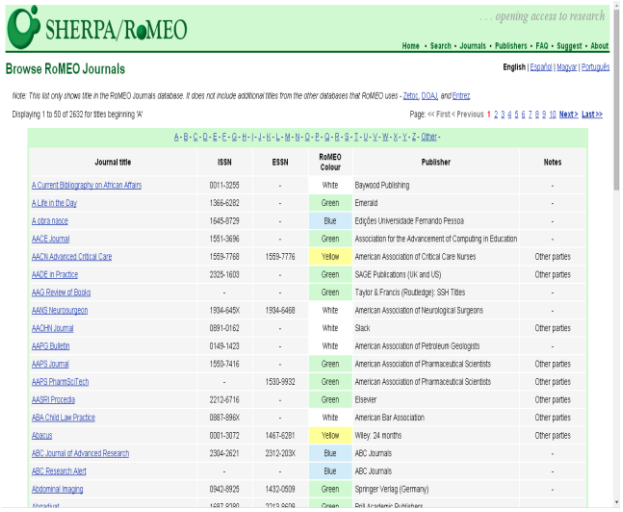


Figura 9. Sherpa/Romeo

Se aconseja verificar en cada editor antes de firmar, por si hubiera alguna diferencia. Observar que lo que es

válido es lo que se firma con cada editor, como documento o contrato.

4. Conclusiones

Los autores deben comprender la información referida a propiedad intelectual para poder proteger sus derechos y para proteger el saber y el acceso libre a una producción, que en el caso de las universidades, es generada en gran parte con fondos públicos. Se debe comprender que tanto el mundo editorial como las métricas de evaluación de los organismos científicos están cambiando y que se posee el derecho de utilizar adendas o elegir vías alternativas de publicación que no impliquen ceder en forma exclusiva la propiedad de las obras producidas.

5. Referencias

[1] De Giusti, M. R., "El desafío de la difusión abierta de las obras y los derechos de autor en las instituciones académicas" Semana Internacional del Acceso Abierto, Colombia, Octubre 2011, http://hdl.handle.net/10915/5564, revisado el 3/6/14.

[2] De Giusti, M. R., Lira, A. J., Villarreal, G. L., Texier, J., "Las actividades y el planeamiento de la preservación en un repositorio institucional", BIREDIAL - Conferencia Internacional Acceso Abierto, Comunicación Científica y Preservación Digital, Noviembre 2012, http://hdl.handle.net/10915/26045, revisado el 3/6/14.

[3] Bernal, I., "Digital.CSIC: impacto de la ciencia en acceso abierto (2013)", presentación que forma parte del curso impartido por la Oficina Técnica de Digital.CSIC "Digital.CSIC: desarrollo de contenidos. Gestión de Copyright. Impacto de la Ciencia en Acceso Abierto", Madrid, Marzo de 2013, http://hdl.handle.net/10261/73245, revisado el 6/8/14.

[4] Abadal, E, Acceso abierto a la ciencia, Editorial UOC, Colección El profesional de la información, 2012

[5] Open Access for UK Research - JISC's contributions, http://www.jisc.ac.uk/publications/programmerelated/2010/openaccessmainbrochure.aspx, revisado el 6/8/14.

[6] De Giusti, M. R., "LibLink; integración iberoamericana; promoción de la investigación; transferencia de tecnología", ISTEC, Espacio de Sesiones Virtuales WELS, Septiembre 2013, http://hdl.handle.net/10915/29347, revisado el 10/6/14.

[7] SEDICI, http://sedici.unlp.edu.ar, revisado el 25/7/14.

[8] Budapest Open Access Initiative (2/2002), www.soros.org/openaccess/read.shtm, revisado el 6/6/14.

[9] Bethesda Statement on Open Access Publishing, http://ictlogy.net/articles/bethesda_es.html, revisado el 6/6/14.

[10] Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities, http://www.geotropico.org/1_2_documentos_berlin.html, revisado el 6/6/14.

[11] SPARC, http://www.sparc.arl.org/resource/complete-sparc-author-addendum-online, revisado el 6/6/14.

[12] Creative Commoms, https://creativecommons.org/, revisado el 6/8/14.

[13] Creative Commoms Argentina http://www.creativecommons.org.ar/licencias, revisado el 6/8/14.

[14] Ley 26899, http://repositorios.mincyt.gob.ar/recursos.php, revisado el 6/8/14.

[15] OpenDoar, http://www.opendoar.org/, revisado el 6/8/14.

[16] FCEN- UBA, http://digital.bl.fcen.uba.ar/gsdl-282/cgi-bin/library.cgi, revisado el 1/8/14.

[17] UNCUYO, http://bdigital.uncu.edu.ar/paginas/publicar.php, revisado el 1/8/14.

[18] SciELO Argentina, http://www.scielo.org.ar/scielo.php, revisado el 1/8/14.

[19] OpenDoar, Políticas cubiertas, http://www.opendoar.org/tools/en/policies.php, revisado 7/8/14.

[20] OpenDoar, Gráfico de las políticas de acceso, http://www.opendoar.org/onechart.php?cID=&ctID=&rtID=&clID=&lID=&potID=3&rSoftWareName=&search=&groupby=pog.pogHeading&orderby=pog.pogID&charttype=pie&width=600&height=300&caption=Recorded%20[Full-text]%20Data%20Re-use%20Policies%20-%20Worldwide, revisado el 7/8/14

[21] SHERPA/ROMEO, http://www.sherpa.ac.uk/romeo/journalbrowse.php?fIDnum=|&mode=simple&la=en, revisado el 6/6/14

El problema de la cifra negra en los Delitos Informáticos en Latinoamérica y el Proyecto ODILA

Temperini, Marcelo Gabriel Ignacio
Becario Tipo I de CONICET, Doctorando en Derecho en la Facultad de Ciencias Jurídicas y Sociales de la Universidad Nacional del Litoral / AsegurarTe / mtemperini@asegurarte.com.ar

Borghello, Cristian Fabian
Licenciado en Sistemas, desarrollador, Certified Information Systems Security Professional (CISSP), CCSK (Certificate of Cloud Security Knowledge) y Microsoft MVP Security (Most Valuable Professional) / seguinfo@gmail.com

Macedo, Maximiliano
Analista en Informática Aplicada (Universidad Nacional del Litoral) / AsegurarTe / mmacedo@asegurarte.com.ar

Abstract

El paso de los años demuestra que el problema de los delitos informáticos sigue creciendo y cada vez a pasos más acelerados. En relación a la problemática puntual sobre la cifra negra planteada en el presente trabajo, se expresa la preocupación sobre la falta de estadísticas generales y oficiales sobre delitos informáticos, la cual impide, por ejemplo, determinar qué tipo de delitos son los más cometidos, los bienes jurídicos más afectados, determinar los tipos de objetivos de los delincuentes (empresas financieras, bases de datos personales, etc.), entre otras datos de interés y utilidad al momento de tomar decisiones de política criminal con respecto al problema. A nuestro entender, la cifra negra existente representa un aspecto sustancialmente problemático que impide desarrollar un trabajo serio de observación, análisis y elaboración de estrategias o planes a mediano o largo plazo orientados a combatir el cibercrimen. En este marco es presentado el Proyecto ODILA (Observatorio Latinoamericano de Delitos Informáticos), que busca construir un espacio de investigación y trabajo, especialmente dedicado a relevar y recolectar información sobre delitos informáticos ocurridos en Latinoamérica, con la finalidad de generar, sistematizar y difundir información sobre el problema de la cifra negra y la necesidad de fomentar la realización de denuncias por parte de los particulares que sean víctimas. ODILA asume el riesgo de equivocarse, pero de intentar, aportando una idea y un lugar de trabajo que, más allá de la precisión o calidad de los datos que puedan llegar a obtenerse con el paso del tiempo, siempre tendrá como finalidad última una intención de difusión y capacitación hacia la sociedad sobre la problemática de los delitos informáticos, un flagelo que sigue avanzando y sobre el cual estamos convencidos que para poder combatirlo eficazmente, es necesario el esfuerzo conjunto entre los distintos actores, públicos y privados.

1. Introducción

De acuerdo a un informe [1] elaborado por la empresa Symantec, las actividades informáticas delictivas están en crecimiento a nivel global, incluyendo por supuesto a América Latina. En las estadísticas citadas, las víctimas de los delitos informáticos aumentaron de un 10% a un 13% sólo entre el año 2011 y 2012. Tal como expresa Nir Kshetri en una de sus obras [2], "*El crecimiento meteórico del cibercrimen ha sido un tema de preocupación apremiante para nuestra sociedad*"[1]
Los delitos informáticos son una de las problemáticas de mayor complejidad de la actulidad, en las cuáles se concentran distintos tipos de desafios, tanto técnicos como jurídicos. Entre ellas se destaca que es un delito de tipo pluriofensivo, viéndose afectada la confidencialidad, integridad y disponibilidad de la información, así como la privacidad, el patrimonio, la reputación e imagen de las personas, entre otras. Por otro lado, encontramos desafíos tales como la internacionalidad en su posibilidad de comisión, la posibilidad de realización de ataques masivos, la posibilidad de anonimato en Internet, la

[1]Traducción propia, originalmente el autor expresó *"The meteoric rise in cybercrime has been an issue of pressing concern to our society"*.

dificultad en la obtención de evidencia digital, la falta de conocimiento de las victimas, la falta de recursos dedicados por parte de los organos competentes, sumado a la falta de estadísticas oficiales que permitan una buen dimensionamiento del problema. En esta oportunidad, trabajaremos sobre el ultimo de los elementos, es decir, el problema de la falta de estadísticas oficiales, o desde otro punto de vista, la problemática de la cifra negra de los delitos informáticos, un aspecto que a nuestro criterio es sustancialmente problemático, toda vez que impide desarrollar un trabajo serio de observación, análisis y elaboración de estrategias o planes a mediano o largo plazo orientados a combatir el cibercrimen. En este marco, se propone la creación del Observatorio Latinoamericano de Delitos Informáticos: ODILA. Este proyecto busca construir un espacio de investigación y trabajo en materia de delitos informáticos, especialmente dedicado a relevar y recolectar información sobre delitos informáticos ocurridos en Latinoamérica, con la finalidad de generar, sistematizar y difundir información sobre la realidad de esta problemática, así como fomentar la realización de denuncias por parte de las víctimas.

2. El alcance de los delitos informáticos

Entre las distintas precisiones que son necesarias realizar, es importante adelantar al lector que la variedad de denominaciones de los delitos por un lado, sumado a la multiplicidad de formas de criminalización de las conductas analizadas que se dan en los distintos países incluidos en el estudio, representan un importante desafío al momento de lograr un consenso sobre cuáles son los llamados “delitos informáticos”.

Existe una variedad de denominaciones de los delitos informáticos, así como una multiplicidad de criterios en las formas de criminalización de estas conductas de acuerdo a cada país. De acuerdo al Dr. Julio Tellez Valdes [3], estos delitos se pueden clasificar en sus formas típica y atípica, entendiendo por la primera a "las conductas típicas, antijurídicas y culpables en que se tienen a las computadoras como instrumento o fin" y por las segundas "actitudes ilícitas en que se tienen a las computadoras como instrumento o fin”. Si bien existen otros tantos conceptos en la doctrina, se toma el citado toda vez que su simple clasificación entre típicos y atípicos es útil a los fines de este trabajo. Aquí, solamente se consideran como “delito informático” a aquellos dentro de las categorías de los típicos, es decir, como una conducta penalmente sancionada por un país determinado.

En un segundo escalón de razonamiento del “delito informático”, se plantea una decisión que, dependiendo de la postura, los delitos informáticos son algunos pocos o se multiplican de una forma considerable. Nos estamos refiriendo a que por lo general, el concepto de “delito informático” está reservado para aquellos tipos penales específicos de la materia (por ejemplo, casos de hacking, cracking, denegación de servicios, etc.). Sin embargo, de acuerdo a una interpretación estricta del concepto ya descripto, si la conducta típica se realiza a través de un medio informático o telemático (a nuestro criterio esto es más correcto que mencionar “computadoras” como se expone en el concepto original) también pasaría a ser un delito informático. ¿Es ésto correcto? El autor Miguel Angel Davara Rodríguez [4] define el delito informático como “la realización de una acción, que reuniendo las características que delimitan el concepto de delito, sea llevada a cabo utilizando un elemento informático y/o telemático, o vulnerando los derechos del titular de un elemento informático, ya sea hardware o software”. Agregando luego: “Nos estamos refiriendo solamente, a la comisión de un delito por medios informáticos o telemáticos, ya que la comisión de otros delitos en los que alguna forma interviene un elemento informático, se encontrará sin duda dentro del Derecho Penal General”.

A nuestro parecer, la respuesta es positiva y negativa a la vez. Es decir, en principio debe dejarse en claro que la mera intervención de un elemento informático no convierte a un delito clásico en un delito informático. Sin embargo, es necesario reconocer que en determinados tipos penales, el ingrediente tecnológico es tan poderoso que hace necesario que para su persecución y sanción intervengan especialistas dedicados a los delitos informáticos. Es decir, hablamos de aquellos casos donde el tipo penal clásico es perfeccionado utilizando a las nuevas tecnologías como medio para su comisión, dotándolo de esa manera de muchos de los inconvenientes o desafíos clásicos de los delitos informáticos, tales como el anonimato, internacionalidad, dificultad en la obtención de evidencia digital, entre otros.

A modo de ejemplo, podemos citar delitos de los cuáles pocos dudan de su condición de “delito informático”, al punto que los mismos han sido incorporados por la Ley Nro. 26.388 de Argentina, y por otro lado, son considerados entre los delitos informáticos “obligatorios” según la consideración del Convenio de Cibercriminalidad de Budapest. Estamos hablando de los casos de la estafa electrónica y de la pornografía infantil, dos delitos que datan de muchísimos años antes de la irrupción de las nuevas tecnologías, afectando el bien jurídico del patrimonio por un lado, y la integridad sexual del menor por el otro. Sin embargo, es claro que la combinación de las nuevas tecnologías al momento de atacar los mismos bienes jurídicos, generó una nueva forma de ataque mucho más potente y masiva, generando la necesidad de adaptar ambas figuras penales a fin contar con herramientas aptas para combatir estas nuevas formas de cometer delitos clásicos.

A los fines del presente trabajo, consideramos importante

realizar estas apreciaciones iniciales toda vez que ellas son el fundamento de por qué si bien las injurias o amenazas no son técnicamente "delitos informáticos", si se decide incorporarlos entre las distintas opciones de "hechos" que pueden ser reportados por los usuarios al momento de informar su reporte en el Proyecto ODILA.

Por último, y antes de pasar a un desarrollo más pormenorizado de distintos desafíos que plantean los delitos informáticos, parece atinado citar al Dr. Horacio Fernández Delpech [5], quien en su última obra realiza un detalle sobre serie de circunstancias que hacen que este tipo de ilícitos penales sean de difícil represión:

1) La falta de una tipificación específica en la mayoría de las legislaciones de los delitos informáticos y cometidos a través de la red;
2) La trasnacionalidad de las conductas, que muchas veces se realizan en un país, pero cuyos resultados se producen en otro;
3) La falta de consenso internacional sobre la reprochabilidad de ciertas conductas;
4) Las permanentes innovaciones tecnológicas, que generalmente avanzan más rápido que las implementaciones de soluciones normativas;

3. El problema de las estadísticas

Si bien al comenzar el artículo se cita una de las estadísticas más reconocidas en el ámbito de la seguridad de la información, se deben realizar algunas consideraciones que se consideran pertinentes en relación a lo que se considera como el problema (o desafío) de las estadísticas en materia de cibercriminalidad.

La falta de estadísticas oficiales en la materia, es al menos para estos autores, un aspecto sustancialmente problemático, toda vez que ello impide un trabajo serio de observación, análisis y elaboración de estrategias o planes orientados a combatir el cibercrimen.

Hasta el momento, no se conocen en Argentina (ni en América Latina) estadísticas oficiales generales que permitan observar y cuantificar los delitos informáticos ocurridos en todo nuestro territorio.

En nuestro país, es posible citar un importante esfuerzo llevado adelante por el Dr. Ricardo Sáenz [6], Fiscal General ante la Cámara Nacional de Apelaciones en lo Criminal y Correccional de la Capital Federal, que en el año 2010 ha llevado adelante un estudio solicitando a las Fiscalías Criminales y Correccionales Federales de todo el país, Fiscalías Nacionales en lo Criminal de Instrucción, Fiscalías Correccionales y Fiscalías de Menores que informaran la cantidad de causas iniciadas con motivo de la sanción de la ley 26.388 hasta el 30 de junio del año 2010 .

Para este informe se requirió, asimismo, que comunicaran: el modo de inicio de las actuaciones (prevención, denuncia en cámara, denuncia en fiscalía, denuncia en sede policial, denuncia remitida desde el extranjero), si el trámite de la causa se encontraba delegado en la fiscalía, si el autor era conocido o tramitaba como NN; cuáles eran los delitos investigados; las dificultades más relevantes (ya fueran técnicas o jurídicas) en la investigación de esos delitos; el cuerpo técnico interviniente y, por último, la proyección internacional de los delitos informáticos y los obstáculos más significativos relacionados con esa cuestión.

De acuerdo a este estudio, los delitos más denunciados en esa etapa, fueron en primer lugar las estafas informáticas (art. 173 inc. 16), seguido por casos de pornografía infantil (art. 128) y finalmente en tercer lugar los casos de acceso ilegítimo (ar.t 153). Entre las conclusiones de dicho trabajo, se destaca lo siguiente:

"Enfocarnos únicamente en los delitos informáticos previstos en la ley 26.388 nos confirmó que, aunque existe un número elevado de casos en los que interviene la justicia penal, aún existe un gran número que tenemos que incluir dentro de la cifra negra de esta clase de delitos. En nuestra opinión, esta situación responde a la falta de conocimiento general en lo que a este tema se refiere, así como también a la carencia de normas que castiguen las nuevas conductas que revisten características suficientes como para incluirlas como accionares ilícitos."

"El relevamiento pone en evidencia, sin lugar a dudas, que la capacitación de los operadores de la justicia es indispensable para una mejor y más cabal comprensión de la ciberdelincuencia en todos sus aspectos legales. Las particularidades que esta nueva forma de delincuencia reviste presenta serias dificultades investigativas: la transnacionalidad del accionar ilícito pone en juego a los principios básicos de la ley penal aplicable; el anonimato atenta contra la efectividad en la identificación de los ciberdelincuentes; el ciberdelito se perpetra en cuestión de segundos y la evidencia digital que prueba su comisión y permite un primer rastreo de los datos que nos conduzcan a la obtención de información que de fundamento al inicio de la pesquisa, se elimina, se altera y se pierde, sin que pueda ser recuperada, en la mayoría de las veces, con la misma velocidad."

El citado estudio del Dr. Ricardo Saenz fue de vital importancia para la creación del Equipo Fiscal Especializado en Delitos y Contravenciones Informáticas, que actúa con competencia única en toda la Ciudad Autónoma de Buenos Aires, con el fin de investigar los delitos informáticos propiamente dichos, y aquellos que se cometen a través de internet, que por su complejidad en la investigación o su dificultad en individualizar a los autores, merecen un tratamiento especializado.

Dicho equipo, fue generado el 15 de noviembre de 2012, donde la Fiscalía General de la CABA dictó la

Resolución 501/12, a través de la cual creó como prueba piloto por el término de un año, el Equipo Fiscal Especializado en Delitos y Contravenciones Informáticas. A finales de 2013, dicho equipo (compuesto por Daniela Dupuy, Tomás Vaccarezza, Mariana Kiefer y Catalina Neme, entre otros), publicó un informe sobre el trabajo realizado durante el primer año de funcionamiento.

Antes de sacar algún tipo de conclusiones sobre estas estadísticas, es necesario mencionar que dicho equipo de trabajo no posee competencia sobre todo tipo de delitos informáticos, sino solamente sobre aquellos que expresamente se han delimitado, a fin de que la tarea del grupo no se encuentre desbordada (más aún en consideración de su intención de conformación como prueba piloto). Como puede observarse en el gráfico Nro. 1, estos delitos solamente son: Pornografía Infantil (art. 128 CP); Daño informático y daño agravado (art. 183 2do párrafo y art. 184 inc. 5to. CP), Amenazas (art. 149 bis CP); Hostigamiento, maltrato o intimidación (art. 52 Cód. Contravencional de CABA[2]); Suministro de material pornográfico a menores (art. 63 Cód. Contravencional de CABA[3]).

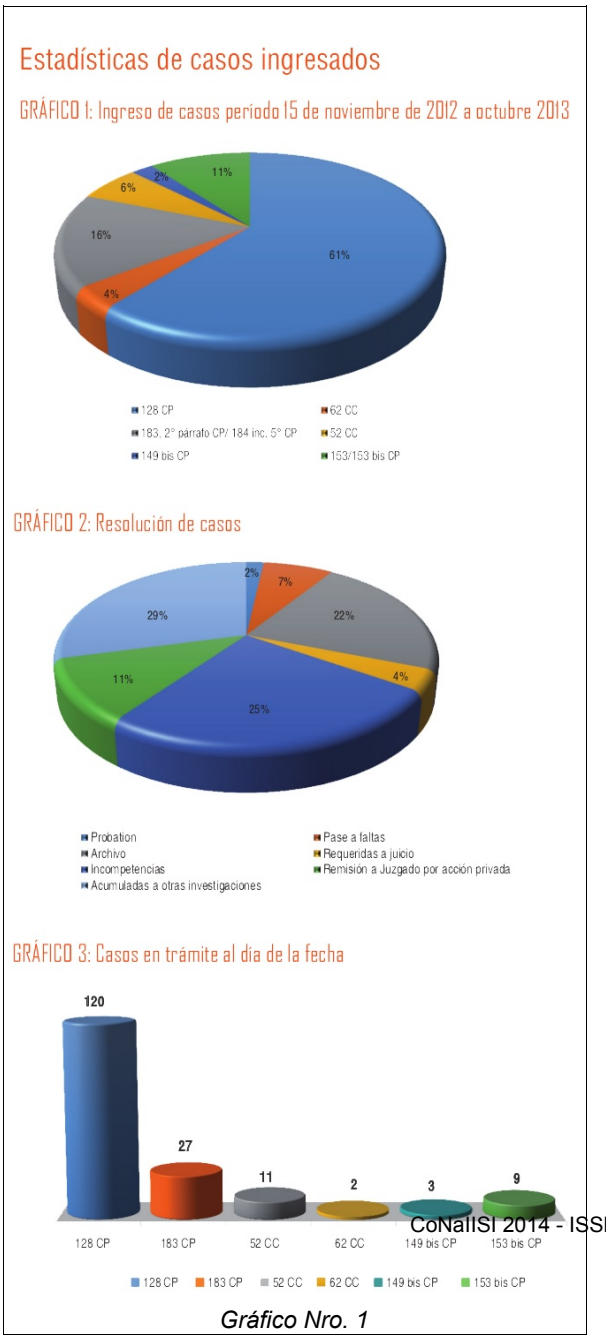


Gráfico Nro. 1

Entre las conclusiones de dicho informe, se destaca *"el incremento incesante de casos informáticos a la justicia local, el que ha superado ampliamente las expectativas iniciales. Nótese que actualmente contamos con ciento setenta y dos (172) casos de delitos informáticos en trámite, sin perjuicio que, además, investigamos otros doscientos cuarenta y cuatro (244) delitos y contravenciones comunes; lo que eleva a un total de cuatrocientos dieciséis (416) casos los absorbidos por este equipo fiscal. Asimismo, varios son los factores que justifican considerar que el ingreso de casos va a aumentar significativamente en un futuro inmediato"*.

Como se comentó inicialmente, lo publicado en el citado informe son las únicas estadísticas oficiales existentes, que sólo hacen referencia a un determinado conjunto de delitos, y sobre casos exclusivamente de la Ciudad Autónoma de Buenos Aires. Es decir, no existen estadísticas o cifras que permitan observar la magnitud de la problemática que representan los delitos informáticos a nivel nacional, o al menos, provincial.

Este vacío implica diversas dificultades para analizar seriamente el problema de la ciberdelincuencia en el país. La falta de estadísticas oficiales impide, por ejemplo, determinar qué tipo de delitos son los más cometidos, los bienes jurídicos más afectados, determinar los tipos de objetivos de los delincuentes (empresas financieras, bases de datos personales, etc.), entre otros datos de interés que brindarían un marco adecuado para tomar determinadas decisiones de política criminal.

4. Estadísticas corporativas e internacionales

Ante la inexistencia de estadísticas oficiales en la materia, podrá el lector válidamente preguntarse, ¿Qué es lo que actualmente se utiliza para trabajar en materia de Cibercrimen? A modo de respuesta preliminar, se deben reconocer que en la mayoría de los casos suelen utilizarse

[2] Ley 1472 CABA - Artículo 52 - Hostigar. Maltratar. Intimidar. Quien intimida u hostiga de modo amenazante o maltrata físicamente a otro, siempre que el hecho no constituya delito, es sancionado con uno (1) a cinco (5) días de trabajo de utilidad pública, multa de doscientos ($ 200) a un mil ($ 1.000) pesos o uno (1) a cinco (5) días de arresto. Acción dependiente de instancia privada.

[3] Ley 1473 CABA - Artículo 62 - Suministrar material pornográfico. Quien suministra o permite a una persona menor de dieciocho (18) años el acceso a material pornográfico es sancionado/a con uno (1) a cinco (5) días de trabajo de utilidad pública, doscientos ($ 200) a un mil ($ 1.000) pesos de multa o un (1) a cinco (5) días de arresto. La sanción se eleva al doble en caso que tal conducta se dirija a una persona menor de dieciséis (16) años. Admite culpa.

estadísticas realizadas por empresas privadas interesadas en el mundo de la seguridad de la información, tales como las publicadas por la empresa Symantec, referenciadas al comienzo de este artículo.

En el informe de la citada empresa (ref. 1), se afirma que los delitos informáticos producen una pérdida de 388 mil millones de dólares anuales (considerando el propio dinero robado más el dinero que les implica a las víctimas poder solucionar sus problemas), haciendo comparable cifras con la ocasionadas por el mercado negro de la marihuana y la cocaína, que es de 288 mil millones de dólares anuales.

Es legítimo preguntarse (y dudar) por la validez de dichas cifras, que son arrojadas a modo genérico y mundial, toda vez que quien las publica son empresas cuyo negocio es la venta de una de las principales "*armas*" para defenderse de estos ataques informáticos (antivirus, *firewall* y otras herramientas similares).

En un estudio sobre cibercrimen encargado por la Organización de Estados Americanos [7], se reconoció que el cibercrimen ha crecido (entre 2011 y 2012) entre el 8 y el 12 % en algunos países, y en el caso más extremo, de cerca del 40%. Más allá de la estadística, y en relación a la propia problemática de la recolección de datos que permitan analizar el estado de situación, en su introducción, ell estudio afirma que "o*bviamente, los incidentes cibernéticos incluidos en los informes de los gobiernos de los Estados miembros de la OEA representan solamente una fracción del número total de incidentes y otras formas de delincuencia cibernética que se llevan a cabo en la región. Pero* ***sigue siendo sencillamente imposible en este momento recopilar datos que permitan obtener una imagen verdaderamente exhaustiva y detallada de la extensión de todos estos incidentes y actividades en las Américas y el Caribe****, o en cualquier otro sitio. Como ya dijimos, el intercambio de información dentro de los gobiernos —incluso aquellos con la capacidad más avanzada en materia de seguridad cibernética— sigue quedando corto, en gran parte debido a las realidades prácticas de que múltiples organizaciones tengan que responder simultáneamente a una gama de amenazas y blancos en constante evolución. Y muchas empresas privadas y otras entidades no gubernamentales siguen mostrándose reacias a reportar ataques o violaciones.* ***Contabilizar el número de incidentes que afectan a los ciudadanos individuales plantea un desafío incluso mayor, en vista del porcentaje incluso más alto de ellos que pasan desapercibidos y no se reportan. Por último, la falta de colaboración generalizada y persistente entre las partes interesadas en todos los niveles dificulta todavía más recoger información sobre violaciones de datos.****"*[4]

Es decir, en un estudio cuyo objetivo era precisamente determinar el estado o la situación del cibercrimen a nivel regional, se reconoce oficialmente la dificultad que se tuvo en el trabajo al intentar recolectar información oficial de los propios organismos públicos y de las grandes empresas, reticentes a colaborar y dar información sobre los incidentes sufridos.

Por otro lado, también es posible citar el *Comprehensive Study on Cybercrime* [8] realizado por Naciones Unidas, en particular por la *United Nations Office on Drugs and Crime* (ONUDC), publicado en Febrero de 2013 y en el cuál se encuestaron a más de 82 países, incluyendo varios de Latinoamérica (entre ellos, Argentina).

Para la recopilación de información, la ONUDC elaboró un cuestionario que fue difundido entre los Estados Miembros, las organizaciones intergubernamentales y las entidades del sector privado. Además, teniendo en cuenta la necesidad de equilibrar la representación de las diferentes regiones, se consultó a los representantes del sector privado, incluidos los representantes de los proveedores de servicios de Internet, los usuarios de los servicios y otros actores pertinentes; así como a representantes del mundo académico, tanto de los países desarrollados y en desarrollo.

Entre otros datos importantes que incorpora este estudio integral, se revela que en la mayoría de los países el índice de cibercriminalidad es notablemente más alto que los de los delitos tradicionales, tales como el robo o el hurto común. Mientras estos últimos tienen índices inferiores al 5%, los delitos informáticos oscilan entre el 10 y el 17%. Estos números, pueden apuntalar con relativa facilidad las afirmaciones sobre el incremento de la delincuencia informática en los últimos años.

No obstante, y aun considerando que los delitos informáticos no reconocen límites de fronteras para su comisión (característica de este tipo de delitos), no se debe caer en la confusión de utilizar estadísticas generales de esta clase de estudios (realizados a nivel mundial) con las situaciones propias que vive cada país en particular, en nuestro caso, de Argentina.

5. El problema de la cifra negra

En estrecha relación con el desafío de la falta de estadísticas oficiales, se debe desarrollar la problemática de la propia falta de denuncias realizadas por las víctimas, que genera la llamada "cifra negra" de los delitos informáticos.

De acuerdo al Dr. Germán Aller [9] la cifra negra es lo más próximo numéricamente a la cantidad real de crímenes cometidos en una sociedad determinada. La relación de tensión existente entre delitos realmente cometidos y los efectivamente tratados por el aparato penal, engloba a la mayor cantidad de víctimas que ni siquiera serán atendidas, tratadas ni conocidas por el

[4] La negrita pertenece a los autores.

segmento penal, y a las cuales el Estado no da respuesta alguna. De acuerdo a Aller, tal proceso "empuja" a las personas a no denunciar los actos ilícitos, a no reconocerse a sí mismas como víctimas y, en consecuencia, a la impunidad que el infractor penal asume, puesto que en el acto desvalorado no vislumbra un referente social acompasado del penal, en tanto a su conducta es delictiva, pero el núcleo social o persona menoscabada por el delito que se ha cometido no pone en evidencia tal daño, y por ende, tampoco el segmento penal podrá operar en su contra.
De acuerdo a este autor, muchas víctimas no denuncian los delitos sufridos porque:

1) No se cree en la Policía ni en la Justicia Penal.
2) No se acepta la condición de víctima, debido a que implica pérdida de dignidad y falta de solidaridad.
3) No se quiere evidenciar la victimización individual ni colectiva.
4) El aparato penal carece de plataforma adecuada para abordar ni siquiera con un mínimo de éxito la solución del conflicto social base.
5) Se tiene miedo a la venganza o amenazas posteriores por parte del autor del delito.
6) Se quiere olvidar lo ocurrido.
7) Se desconoce que se haya cometido un delito.
8) La víctima se siente total o parcialmente culpable de lo sucedido.
9) Se ignora que puede pedir la intervención del Estado.
10)

A simple vista, se pueden reconocer en el listado de causas realizado por el Dr. Aller, varios aplicables al ámbito del cibercrimen. Puntualmente, se tomarán algunas de ellas, con el objetivo de un desarrollo personal sobre dichos aspectos.

a) Falta de confianza en la Policía o la Justicia: En nuestro país este aspecto es relevante en la mayoría de los casos, ya que existe un alto porcentaje de la población que tiene falta de confianza en la Justicia. Así lo demuestran desde hace años los estudios del Índice de Confianza en la Justicia [10] (ICJ) realizados por Fores, la Escuela de Derecho de la Universidad Torcuato Di Tella, y la Fundación Libertad. A Marzo de 2010, los índices de confianza fueron del 50,5% (en una escala donde 0 expresa el mínimo de confianza y 100 el máximo). El mismo estudio revela, por ejemplo, que "*En términos de eficiencia... la Justicia en Argentina es...*" poco confiable para el 55% de los encuestados, y nada confiable para el 22% de la población consultada.

b) El aparato penal carece de plataforma adecuada para abordar ni siquiera con un mínimo de éxito la solución del conflicto social base: Este punto, si bien tiene estrecha relación con el anterior, reviste sus diferencias. Puntualmente, es menester destacar que en muchos casos de delitos informáticos, para poder realizar una investigación eficaz tendiente a determinar y capturar al autor del mismo, es necesario poseer una infraestructura adecuada. Es decir, se precisa contar con los recursos técnicos necesarios, con personal disponible y capacitado, así como cierta agilidad en cuanto a coordinación y cooperación de distintos entes (relación ISP / Justicia). Todos estos aspectos no suele encontrarse (salvo excepciones) a lo largo de nuestro país, radicando aquí otra de las grandes problemáticas a ser abordadas.

c) Se desconoce que se haya cometido un delito: Desde la experiencia en el trabajo privado de consultoría en seguridad informática, se puede afirmar que muchas personas que escriben para consultar por los problemas que han tenido en Internet, desconocen totalmente que han sido víctimas de un delito informático tipificado en Argentina. Esta falta de conocimiento por parte del promedio de la sociedad sobre la existencia de estos "novedosos" delitos, hace que varios casos se pierdan dentro de la gran bolsa de la cifra negra.
Antes de finalizar en este punto, consideramos oportuno agregar un elemento nuevo, o al menos una causa no considerada por el Dr. Aller, que desde la experiencia se puede afirmar que es un aspecto decididamente relevante al momento de analizar la falta de denuncias sobre casos vinculados al cibercrimen.

d) Confidencialidad como requisito: Muchas víctimas de distintos tipos de delitos informáticos deciden voluntariamente no denunciar sus casos, bajo el razonamiento que la posibilidad de difusión pública ocasionaría un daño peor al ya efectivamente sufrido. En este sentido, dentro del ámbito privado de la seguridad informática, una de las principales virtudes que analizará un cliente, será precisamente el nivel de profesionalismo en relación a la confidencialidad de los casos. Dicha situación encuentra una lógica respuesta considerando que la víctima es consciente que de difundirse públicamente el incidente que ha sufrido (algo potencialmente probable en caso de denuncia ante las autoridades), las consecuencias hacia su imagen, buen nombre, reputación, etc., puede ser gravemente perjudicada, superando ampliamente el daño ya recibido por el ataque informático en sí mismo.
A modo de mejorar el sentido de este aspecto, se citan dos ejemplos para la facilidad de comprensión de los lectores. Supongamos el caso de una importante empresa dedicada al rubro de la salud (clínica por ejemplo), la cual es víctima de ataques informáticos que logran acceder a sus sistemas y hacerse con todas sus bases de datos, las cuales poseen información sensible sobre la salud de miles de personas de la región. La difusión de

dicho incidente podría ocasionar daños irreparables en cuanto a la confianza de los clientes, construida en base al trabajo de muchos años. Otro ejemplo puede ser el caso de un profesional reconocido en su ambiente (un periodista, un abogado, un psicólogo, etc.) el cuál es víctima de un caso de interceptación de comunicaciones electrónicas (cuentas "pinchadas"), permitiendo que terceros extraños accedan a todas sus comunicaciones privadas (y eventualmente difundidas). Fácilmente puede advertirse que los daños ocasionados por una noticia (que vale destacar que son generalmente buscadas atendiendo a la gran recepción que existe en la masa sobre este tipo de casos de problemas "cibernéticos") pueden ocasionar perjuicios irreparables, afectándose bienes jurídicos de difícil reconstrucción (como la imagen, honor o reputación de una persona física o jurídica), comparativamente mucho más graves al daño ya sufrido por el delincuente informático.

6. ODILA: Observatorio Latinoamericano de Delitos Informáticos

El Proyecto ODILA (Observatorio Latinoamericano de Delitos Informáticos) es un proyecto conjunto, llevado adelante por Segu-Info[5] y AsegurarTe[6], a través del Lic. Cristian Borghello (Segu-Info), el Abog. Marcelo Temperini (AsegurarTe) y el A.I.A. Maximiliano Macedo (AsegurarTe). Este proyecto busca construir un espacio de investigación, encuentro y trabajo sobre la realidad latinoamericana en materia de delitos informáticos y cibercrimen, especialmente dedicado a relevar información sobre incidentes o delitos informáticos ocurridos en los países de Latinoamérica.
La misión del Proyecto ODILA es brindar un ámbito virtual donde todas aquellas personas que hayan sido víctimas de algún tipo de un delito informático en Latinoamérica, puedan reportar e informar de manera 100% electrónica sobre el hecho ocurrido, con el fin de recolectar información que permita saber el estado de situación en materia de delitos informáticos en la región.
En otras palabras, ODILA nace a partir de la necesidad de dar a conocer el problema de la cifra negra de los delitos informáticos, informando a la sociedad sobre la legislación vigente y fomentando la necesidad de realizar las denuncias en los casos que ocurran este tipo de incidentes, convencidos que para poder avanzar en la lucha contra esta problemática, es necesario conocer la situación actual de los datos sobre el estado de situación en los países de nuestra región, algo que en la mayoría de ellos, no existe oficialmente.

6.1. Objetivo General

- Proponer una alternativa que permita combatir el problema de la cifra negra en materia de delitos informáticos en los países de América Latina.
- Generar, sistematizar y difundir información relevante para estudiar, investigar e incidir en la problemática de los delitos informáticos en países de América Latina.

6.2. Objetivos específicos

- Informar sobre el problema de la cifra negra en materia de delitos informáticos.
- Difundir consejos e información útil para las víctimas de delitos informáticos.
- Generar informes y estadísticas propias sobre la ciberdelincuencia en Latinoamérica.
- Fomentar en el usuario la realización de denuncias para los casos de delitos informáticos.

6.3. Metodología, Sistema y Funcionamiento

El Proyecto ODILA será llevado a cabo a través de su sitio oficial www.odila.org, en el cual cualquier ciudadano podrá encontrar información sobre el proyecto, legislación sobre delitos informáticos y, sobre todo, tendrá la posibilidad de reportar el incidente informático que ha sufrido, a fin de poder obtener una guía sobre si el hecho sufrido puede ser considerado o no, un delito informático de acuerdo a la legislación vigente en el país de residencia del usuario.
Como puede observarse en el gráfico nro. 2 de procesos que se acompaña, el sistema se desarrolla a través de distintas etapas:

[5]Segu-Info: www.segu-info.com.ar

[6]AsegurarTe: Consultora en Seguridad de la Información – www.asegurarte.com.ar

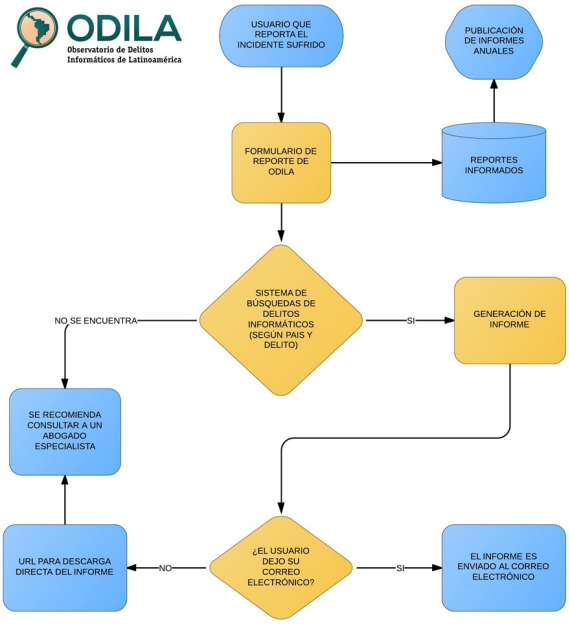


Gráfico Nro. 2

1) Recolección de información a partir del reporte realizado por el usuario.
2) El formulario es procesado por el sistema, por un lado enviándose a una base de datos de reportes, cuyo objetivo final es la publicación de futuros informes.
3) Por otro lado, parte de esa información (país de residencia y tipo de incidente) es enviada al sistema de búsquedas de delitos informáticos.
4) En el caso que el sistema encuentre que en el país indicado ese hecho pueda ser encuadrado en algunos de los tipos penales vigentes, el sistema procederá a generar un informe, indicándose dichos artículos así como algunas recomendaciones generales a tener en cuenta y sobre todo, información sobre donde poder realizar una denuncia formal para que se pueda investigar el hecho por parte de las autoridades pertinentes.
5) En el caso que el sistema no encuentre que en ese país, ese hecho pueda encuadrarse en algunos de los tipos penales vigentes, devolverá un mensaje informando sobre tal situación, sumado a la legislación vigente en dicho país (para que el usuario pueda consultarla de forma directa) y sobre todo, una recomendación de consultar y asesorarse con un profesional especializado en la materia.

6.4. Formulario de Reporte

El formulario de reporte que se brinda al ciudadano se compone de las siguientes preguntas, acompañadas de información de ayuda para facilitar la interpretación y las respuestas. A fin de lograr esto, en la mayoria de las preguntas, se ha dispuesta un sistema de opciones estilo múltiple choice, por un lado para que sea de más rápido llenado para el usuario, y por otro, para estandarizar el tipo de respuesta, facilitando la posterior construcción de estadísticas.

Las preguntas son las siguiente:

1- ¿Cuál es su país de residencia? *Esta información es útil para poder determinar si el incidente es considerado un delito informático en el país donde usted se encuentra. Sólo para Latinoamérica.* * Esta pregunta es obligatoria.

2- La víctima de un delito informático ha sido... *Esta información es opcional, pero si usted colabora nos ayudará a comprender cuáles son los sectores sociales más atacados por los ciberdelincuentes.*

- Persona física
- Empresa pequeña o mediana (PyME)
- Gran empresa (más de 100 empleados)
- Organismo gubernamental

3- ¿Qué tipo de incidente ha sufrido? *De acuerdo a su lugar de residencia, dicho incidente podrá ser considerado como un delito informático de acuerdo a su legislación vigente. Si usted lo desea, una vez finalizado el reporte, se le enviará un mail informando sobre la legislación existente en su país de residencia.* * Esta pregunta es obligatoria.

- **Hacking:** alguien accedió sin mi consentimiento a mis cuentas o sistemas.
- **Cracking:** alguien ha modificado, alterado o eliminado todo o parte de mis datos o sistemas informáticos.
- **Fraude o Estafa Informática:** alguien ha realizado alguna manipulación que me ocasionó un perjuicio económico.
- **Phishing:** alguien a través de un engaño me ha solicitado información confidencial -Nº de tarjetas de crédito, contraseñas, PIN, etc.
- **Suplantación de identidad digital:** alguien se hace pasar por mi a través de un medio electrónico (mails, redes sociales, etc.). También se lo conoce como "Robo de identidad"
- **Denegación de Servicio:** alguien ha realizado un ataque que me ha dejado sin poder acceder o prestar mi servicio informático o electrónico de forma normal.
- **Grooming:** un menor ha sido acosado y/o extorsionado con fines sexuales a través de algún medio informático.
- **Calumnias o Injurias:** alguien lo está calumniando o injuriando a través de medios electrónicos -redes sociales, correos

electrónicos, celulares, etc.

- **Amenazas:** alguien lo está amenazando o intimando a través de medios electrónicos -redes sociales, correos electrónicos, celulares, etc.
- **Pornografía Infantil:** alguien está difundiendo, comercializando o facilitando pornografía infantil a través de medios electrónicos.
- **Violación de Datos Personales:** alguien está ofreciendo, comercializando, interceptando o modificando datos personales sin autorización
- **Difusión de Malware:** alguien está produciendo, distribuyendo, vendiendo o propagando malware o software malicioso

4- ¿Cuándo ha ocurrido el incidente? *Aunque haya sucedido hace tiempo, repórtelo igual, a fin de poder colaborar con las estadísticas de los incidentes ocurridos.* * Esta pregunta es obligatoria. Formato DD-MM-AAAA. Por ejemplo 20-08-2014

5- ¿Ha denunciado el delito ante los organismos competentes en su país? *Ej: Denuncia policial, denuncia ante la fiscalía de turno, etc.* * Esta pregunta es obligatoria.

- Si, ya denuncie y la investigación está en curso
- Si, denuncie pero la investigación no avanzó
- Sí, denuncié y ya se ha condenado al o los culpables.
- No, no denuncié porque... (ver siguiente pregunta)

6- En el caso que NO haya realizado la denuncia, ¿Se debe a alguna de las siguientes causas? *Puede elegir una o más causas.* * Esta pregunta es obligatoria.

- Ya denuncié.
- No creo en la Policía ni en la Justicia Penal.
- No me considero víctima de un delito.
- No quiero difundir públicamente el incidente (pérdida de confidencialidad).
- Tengo temor de futuras represalias de parte del autor.
- No creo que la investigación tenga éxito.
- No estoy seguro de haber sido víctima de un delito penal.
- En parte me siento culpable por el incidente.
- No creo que la denuncia sea útil, porque el sistema penal no es apto para combatir el cibercrimen.
- Otros

7- Edad. *Esta información es opcional, pero si usted colabora nos ayudará a comprender cuáles son los sectores sociales más atacados por los ciberdelincuentes.*

- Menores de 21 años
- Entre 22 y 35 años
- Entre 36 y 45 años
- Más de 45 años

8- Género. *Esta información es opcional, pero si usted colabora nos ayudará a comprender cuáles son los sectores sociales más atacados por los ciberdelincuentes.*

- Masculino
- Femenino

9- Nivel de Instrucción. *Esta información es opcional, pero si usted colabora nos ayudará a comprender cuáles son los sectores sociales más atacados por los ciberdelincuentes.*

- Sin instrucción
- Primaria completo
- Secundaria completo
- Universitaria o Terciaria completo

10- Correo electrónico. *En el caso que usted quiera (es opcional), puede recibir una respuesta al presente formulario donde se le indicará si el incidente que usted ha reportado, configura un delito penal en su país de residencia. En el caso que lo sea y que usted aún no haya denunciado formalmente ante las autoridades de su país, se le brindará información de contacto para que usted pueda proceder a realizarla. En el caso que decida utilizar esta opción, se le informa que sus datos personales serán tratados de acuerdo a los principios de acuerdo a la Ley 25.326 de Protección de Datos Personales de Argentina. Ante cualquier duda, puede ponerse en contacto.*

6.5. Aclaraciones importantes

Es de importancia hacer destacar que el Proyecto ODILA **no pretende ser un asesoramiento para el usuario**, razón por la cual se deja en claro que el usuario debe siempre consultar ante un profesional especializado en la materia. Es decir, el sistema no garantiza la exactitud de los resultados en relación a sobre si el hecho efectivamente puede ser considerado un delito penal en el país indicado por el usuario, sino que simplemente brinda un acercamiento a la materia, destacando que su principal objetivo siempre es la concientización acerca de la importancia de dar conocer la problemática, y sobre todo, fomentar la realización de denuncias que permita un combate eficaz frente al problema de la cifra negra.

Por la magnitud del proyecto en cuanto a su intención latinoamericana, así como por las innumerables problemas que pueden presentarse al momento interpretar si un hecho descripto reúne todos los requisitos del tipo penal (algo que podría ser desarrollado por cualquier fiscal o juez), así como las propias complejidades que posee en cada tipo penal en relación a sus requisitos jurídicos, es imposible garantizarle al usuario la exactitud o precisión en los resultados del sistema. Por lo tanto, nuevamente se insiste en que el Proyecto ODILA solamente pretende ser una guía para

orientar al usuario (víctima), brindar información sobre la materia y fomentar la realización de denuncias.
En relación a la recolección de datos, se aclara que en dicho proceso no es necesario que el usuario brinde sus datos personales, es decir, que el reporte puede ser realizado de forma anónima. Excepcionalmente, el usuario tiene la posibilidad de informar en el reporte su cuenta de correo electrónico, a fin de que los resultados y consejos puedan ser posteriormente enviados a esa cuenta. Por último, destacar que se pretende que el formulario diseñado, junto con sus preguntas, resulten lo más sencillos posible, a fin de facilitar su comprensión y por lo tanto, su utilización.
Por último, destacar que los datos recolectados, y de acuerdo al caudal de datos que puedan ser incorporados a través del Proyecto ODILA, serán publicados en informes anuales.

7. Conclusiones

El paso de los años demuestra que el problema de los delitos informáticos sigue creciendo y cada vez a pasos más acelerados. Si bien en la mayoría de los países ya se ha dado el primer paso, sancionando penalmente en mayor o menor medida a los delitos informáticos, esto no es suficiente.
A modo general, se expresa una posición que considera necesario el avance sobre reformas y modificaciones sobre la legislación procesal, a fin de que se arbitren medios más idóneos para poder perseguir investigar, perseguir y condenar de forma efectiva a los delincuentes informáticos. Una legislación procesal más adecuada y flexible permitirá una mejor recolección de la evidencia digital, elemento vital y necesario para avanzar sobre la problemática.
Es también necesario contar con la colaboración de los ISP, empresas privadas que en la mayoría de los casos de delitos informáticos, son poseedores de información (datos de tráfico) esencial al momento de la presentación judicial de las pruebas de los ilícitos.
En relación a la problemática puntual sobre la cifra negra planteada en el presente trabajo, se expresa la preocupación sobre la falta de estadísticas generales y oficiales sobre delitos informáticos, la cual impide, por ejemplo, determinar qué tipo de delitos son los más cometidos, los bienes jurídicos más afectados, determinar los tipos de objetivos de los delincuentes (empresas financieras, bases de datos personales, etc.), entre otras datos de interés y utilidad al momento de tomar decisiones de política criminal con respecto al problema.
A nuestro entender, la cifra negra existente representa un aspecto sustancialmente problemático que impide desarrollar un trabajo serio de observación, análisis y elaboración de estrategias o planes a mediano o largo plazo orientados a combatir el cibercrimen.
En este marco es presentado el Proyecto ODILA (Observatorio Latinoamericano de Delitos Informáticos), que busca construir un espacio de investigación y trabajo, especialmente dedicado a relevar y recolectar información sobre delitos informáticos ocurridos en Latinoamérica, con la finalidad de generar, sistematizar y difundir información sobre el problema de la cifra negra y la necesidad de fomentar la realización de denuncias por parte de los particulares que sean víctimas.
Por último, destacamos que los autores de esta iniciativa, somos conscientes de las limitaciones y errores que pueden darse en el proceso del proyecto planteado, algunas de ellas ya desarrolladas en entre las aclaraciones importantes del proyecto, y otras que seguramente se irán develando a partir de la puesta en marcha del proyecto. ODILA asume el riesgo de equivocarse, pero de intentar, aportando una idea y un lugar de trabajo que, más allá de la precisión o calidad de los datos que puedan llegar a obtenerse con el paso del tiempo, siempre tendrá como finalidad última una intención de difusión y capacitación hacia la sociedad sobre la problemática de los delitos informáticos, un flagelo que sigue avanzando y sobre el cual estamos convencidos que para poder combatirlo eficazmente, es necesario el esfuerzo conjunto entre los distintos actores, públicos y privados.

8. Referencias

[1] SYMANTEC CORPORATION, “2012 Norton Cybercrime Report”, septiembre de 2012; http://www.norton.com/2012cybercrimereport Consulta: 15/04/2014

[2] KSHETRI, Nir, “The Global Cybercrime Industry”, C Springer-Verlag Berlin Heidelberg 2010, ISBN 978-3-642-11521-9

[3] TÉLLEZ VALDÉS, Julio, "Derecho Informático", 3ª.ed., Ed. Mc Graw Hill, México, 2003, Pág. 8

[4] DAVARA RODRIGUEZ, Miguel Ángel, Manual de Derecho Informático, Thomson Aranzadi, 10º edición, pág. 358 y 359.

[5] FERNANDEZ DELPECH, Horacio, Manual de Derecho Informático, 1ra. Edición, Ed. Abeledo Perrot, Argentina, 2014, Pág. 194

[6] SAENZ, Ricardo. "Informe sobre el relevamiento de causas en la que se investigan delitos informáticos", 2010.

[7] TREND Micro; “Tendencias en la seguridad cibernética en América Latina y el Caribe y respuestas de los gobiernos”, 2013, Secretaría de Seguridad Multidimensional de la OEA. ISBN 978-0-8270-6061-6

[8] NACIONES UNIDAS, New York, 2013. UNODC. Comprehensive Study on Cybercrimen: http://www.unodc.org/documents/organized-crime/UNODC_CCPCJ_EG.4_2013/CYBERCRIME_STUDY_210213.pdf Consultado: 27 de Noviembre de 2013

[9] ALLER, Germán. “Cuestiones Victimológicas de Actualidad: Origen de la Victimología, Seguridad, Cifra Negra, Personalización del Conflicto y Proceso Penal”. Revista ILANUD Nro. 27, 2006.

[10] FORO DE ESTUDIOS SOBRE LA ADMINISTRACIÓN DE JUSTICIA, “Indice de Confianza en la Justicia”. http://j.mp/1rE9SiB Consultado: 27 de Noviembre de 2013

Caracterización de los perfiles biográficos digitales en Facebook de adolescentes de Rafaela y Sunchales

María Eugenia Balbiano, Noelia Castillo, Román Pablo Zenobi, Dario Karchesky, Fernando Bircher, Micaela Bressan
Universidad Católica de Santiago del Estero, Departamento Académico Rafaela
Bv. Irigoyen 1502 (2300) Rafaela, Santa Fe, Argentina

eugeniabalbiano@gmail.com, noe_sfe@hotmail.com, rozenobi@hotmail.com, dario.karchesky@ucse.edu.ar, fer.bircher@gmail.com, micabressan@gmail.com

Abstract

Proyecto de investigación de un equipo interdisciplinario, abarcando las carreras de Licenciatura en Psicología, Abogacía e Ingeniería en Informática. La propuesta es describir las características de los perfiles biográficos digitales en Facebook de adolescentes entre los 16 y los 18 años de Rafaela y Sunchales indagando sobre los siguientes aspectos: las personalidades y el desarrollo de los perfiles de personas, redes sociales y perfiles virtuales, la incidencia en el desarrollo de la personalidad, la forma de exposición de las personas en un ambiente virtual y la relación con los conceptos de seguridad de la información, la garantía de privacidad y autenticidad de los contenidos publicados, la vulnerabilidad de los datos presentados y publicados.

1. Introducción

El ser humano es social por naturaleza y necesita relacionarse con sus pares para desarrollarse plenamente. Cada persona es única y tiene una personalidad que la identifica: su forma de comunicarse, su manera de relacionarse con otros, de afrontar problemas, de actuar frente a determinadas situaciones, el modo de hablar, y demás cuestiones.

El desarrollo de Internet, y el avance de la llamada Web 2.0, hizo que los contenidos digitales no sólo se consuman sino que sean las personas las generadoras de material y las impulsoras del intercambio entre todos. Hoy se tiene un rol netamente activo frente a los contenidos de Internet, y el crecimiento es exponencial minuto a minuto. Uno de los motores de dicha expansión son las Redes Sociales sobre Internet, espacios virtuales donde cada persona lleva su perfil personal e interactúa con otros individuos.

Analizando el caso particular de Facebook, nos preguntamos: ¿estamos frente a la posibilidad de tener un perfil psicológico real y un perfil virtual? ¿Estos dos perfiles son similares, contrapuestos o no existe vínculo entre ellos? ¿Hay una relación directa entre la forma en que una persona se relaciona con sus pares de manera real (cara a cara) que cómo lo hace en una red social virtual? ¿Es posible categorizar esté vínculo y obtener indicadores o indicios de esta relación supuesta?

Éstos y otros interrogantes nos llevan a plantear el desarrollo de esta investigación. Como equipo interdisciplinario, abarcando las carreras de Licenciatura en Psicología, Abogacía e Ingeniería en Informática, nos proponemos indagar los aspectos vinculados a cada parte involucrada en este planteo: las personalidades y el desarrollo de los perfiles de personas, las redes sociales y los perfiles virtuales, la incidencia en el desarrollo de la personalidad, la forma de exposición de las personas en un ambiente virtual y la relación con los conceptos de seguridad de la información, la garantía de privacidad y autenticidad de los contenidos publicados, la vulnerabilidad de los datos presentados y publicados.

En concordancia con el párrafo anterior el objetivo general del proyecto es describir las características de los perfiles biográficos digitales en Facebook de adolescentes entre los 16 y los 18 años de Rafaela y Sunchales. Para la concreción del mismo, se tiene en cuenta una serie de objetivos específico:

- Analizar el perfil biográfico digital, mediante su revisión en Facebook, la observación y el registro.
- Analizar el perfil psicológico-social, a partir de la entrevista personal que se realizara a los adolescentes.
- Analizar si existen elementos o indicadores que describan la relación entre el perfil psicológico-social y el perfil biográfico digital.
- Determinar las características de los perfiles biográficos digitales frente a parámetros de seguridad informática y el marco jurídico regulatorio.

2. Marco teórico y estado actual del conocimiento en el problema a investigar

Nos parece importante iniciar esta sección citando el siguiente texto:

El perfil de un usuario forma parte de su identidad digital. Según Hernández y Torruella [1],

en su trabajo Identidades Digitales: límites poco claros, *"La llegada de Internet cambió las condiciones tradicionales de creación y representación de identidad. Especialmente en los inicios, en el entorno virtual, la presencia física se disocia de los encuentros sociales. Este aspecto, junto a la condición de permanecer en el anonimato, permitió la opción de no revelar información sobre el propio aspecto físico o sobre la biografía personal. De esta manera, como nunca antes en la historia, la red hizo posible la reinvención de nosotros mismos, a través de la creación de nuevas identidades; las herramientas nos permitieron proyectar nuestra imagen de una forma más completa"*. También en este artículo se presenta una cuestión muy relacionada a las intenciones que se intentan descubrir en este trabajo, ellos afirman: *"Se supone, muchas veces, que los perfiles en redes sociales crean y comunican una imagen idealizada de nosotros, es decir, que fingimos en Internet lo que no somos. De acuerdo con esta hipótesis sobre la identidad virtual idealizada, los propietarios de perfiles reflejan características idealizadas que no corresponden a sus personalidades reales"*. Otra postura que los autores plantean como alternativa sostiene que, sí reflejamos nuestras verdaderas identidades en Internet, las redes sociales constituyen, ya, parte de nuestro contexto social extendido y son un lugar para expresar características de personalidad reales.

Una Red Social es un conjunto de entidades que se conectan entre sí por alguna relación particular: parentesco, amistad, trabajo, casualidad, descubrimiento; y dónde esas relaciones pueden ir evolucionando a otros estados. Generalmente esas entidades son personas u organizaciones.

Una definición de Red de la Real Academia Española (RAE) es "conjunto de elementos organizados para determinado fin". Si tenemos en cuenta este concepto, podemos decir que en la Red Social las personas se organizan en torno a un determinado fin, de contactarse y relacionarse socialmente.

Si nos referimos al ámbito de la informática, la RAE dice que una Red es "conjunto de computadoras o de equipos informáticos conectados entre sí que pueden intercambiar información".

En este caso también podemos aclarar que una Red Social virtual necesita de una gran red de equipamiento informático de soporte para mantener su funcionamiento. Así mismo, podemos pensar que cada perfil virtual de una persona tiene atrás una computadora de soporte.

El concepto Social, está vinculado al término sociedad. Según C. Sluzki, la red social personal es la suma de todas las relaciones que un individuo percibe como significativas o define como diferenciadas de la masa anónima de la sociedad. Esta red contribuye sustancialmente a su reconocimiento como individuo y constituye una de las claves centrales de la experiencia individual de identidad, bienestar, competencia y protagonismo. [2]

Para este proyecto asumiremos como concepto que una Red Social es un conjunto de personas que se relacionan entre sí de manera organizada con el fin de desarrollar vínculos.

Particularmente nos vamos a focalizar en el ambiente de las tecnologías de la información y la comunicación para trabajar sobre las Redes Sociales Virtuales, es decir, aquellas que se desarrollan sobre Internet.

Para ser más precisos, según Tenzer, S. y otros [3], *"una red social virtual es una estructura social de relaciones entre usuarios a través de internet"*

La red social virtual es una de las herramientas que las nuevas tecnologías han aportado como medio de comunicación y socialización, y no está exenta de las características que describen las relaciones sociales que configuran los vínculos "cara a cara".

Actualmente la red social virtual más grande del mundo es Facebook contando con más de 1100 millones de usuarios. La podemos considerar la "madre" de las redes sociales, la que marcó tendencia en el uso de los perfiles digitales. Por este motivo el presente trabajo la usará como caso de estudio.

Un usuario para poder participar de la misma debe definir su perfil biográfico digital, el cual para el caso de Facebook, está constituido por:

- Datos personales: Nombre y apellido, fecha de nacimiento, fecha de ingreso a la red social, ciudad de origen y actual de vivienda, estudios y trabajo.
- Contactos: grupo de "amigos" que forman parte de la red de este individuo.
- Intereses personales: gustos sobre música, libros, cine, páginas web, otros perfiles digitales, etc.
- Comentarios: es el espacio donde el protagonista de la red escribe sus pensamientos, ideas o simplemente mensajes con otros contactos.

Facebook se compone de un panel principal denominado "muro" que representa el espacio que un usuario dispone para expresar sus opiniones. Este espacio es básicamente la primera imagen del perfil biográfico digital de la persona, mostrando el nivel de privacidad que se desea tener para con otras personas. Es decir, por medio del muro podemos determinar el nivel de exposición pública que un contacto puede tener frente a la red social.

El contenido que un usuario puede publicar en Facebook se compone de: textos de propia autoría, como ser opiniones sobre un determinado tema, estados de ánimo, alguna situación que quiera comentar de su vida, etc.; imágenes y videos que

pueden ser generados por el propio usuario o buscado de otras fuentes; enlaces a páginas o material que otro usuario de la red haya publicado; y diversos contenidos que estén presentes en otra plataforma pero que permitan su vinculación con el perfil de Facebook. Al mismo tiempo, sobre las publicaciones o comentarios de los "amigos" en su red social, puede emitir opiniones indicando "Me gusta" o compartiendo el contenido con otros. Todo este material que el usuario administra y comparte en la red social, forma parte de las características de su perfil biográfico digital y muestra lo que realmente le interesa divulgar para con otros contactos.

A partir de las características mencionadas existe una actividad conocida como "rastreo de perfiles" que consiste en navegar por distintos espacios de integrantes de la red (conocidos o no) para evaluar el nivel de privacidad y así poder obtener información para distintos fines.

El nivel de privacidad en Facebook es determinado por el propio usuario. Existe un panel especial en la página para configurar los parámetros del perfil.

Cuando se define una cuenta en Facebook en primer lugar se solicita una validación de la edad del solicitante para que entre en un rango permitido. Al generar el espacio en la red social, Facebook puede recibir la siguiente información de cada usuario:

- Información de registro: datos personales como nombre, apellido, correo electrónico, fecha de nacimiento y sexo.
- Información que se desea compartir: es la que el usuario desea compartir de manera intencional.
- Información que otros usuarios publican: comprende toda la información que otros contactos pueden publicar de otros "amigos".
- Información que Facebook recopila: datos de la computadora, el navegador, páginas web visitadas, amigos agregados, ubicación por medio de coordenadas GPS, etc.

Facebook trabaja con el concepto de "amigos". Esto es el conjunto de personas que un determinado usuario considera como contactos conocidos y con los que desea tener interacción de alguna manera. Cada usuario en Facebook se determina por un nombre de usuario también conocido como URL de Facebook. Esto es así ya que cada perfil tiene una dirección que incluye este nombre de usuario. Por ende, conociendo a un nombre de usuario se puede acceder automáticamente al perfil "público" de dicho miembro. Según Facebook, la información que siempre es pública para cada usuario es: nombre, foto de portada y perfil, sexo, redes (cantidad de amigos) y nombre de usuario.

La red social personal puede inhibir o favorecer el desarrollo de una persona en cualquier momento de su ciclo vital, especialmente durante las transiciones de una etapa a otra de la vida [Estévez y Aravena, 1988 (Arón A.; Nitsche R.; Rosenbluth A. 1995)][4]

En el caso de los adolescentes, se postula que la red social influye en la forma en que éstos enfrentan al mundo y en su visión de éste y de sí mismos. La adolescencia ha sido descripta como una etapa de transición entre la niñez y la adultez. Es un período crítico, de adquisición de nuevos roles, necesidades y demandas, durante el cual la necesidad de recibir apoyo, así como el tipo de apoyo necesitado, pasa por cambios dramáticos dado que los adolescentes comienzan a alejarse de sus padres y a orientarse hacia sus pares [Vaux, 1985, (Arón A.; Nitsche R.; Rosenbluth A. 1995)]

Durante esta etapa se comienza a articular un repertorio de identidades sociales, siendo la tarea principal formar nuevas redes sociales que reflejen estas nuevas identidades. El adolescente participa activamente en la formación de su red social.

En relación a esto, se ha observado que las redes de los adolescentes se van transformando, volviéndose más densas, extensas y complejas que en la infancia [Bryant, 1985; Hirsch, 1985 (Arón A.; Nitsche R.; Rosenbluth A. 1995)].[4]

Conociendo que las redes sociales de los adolescentes, en general, van aumentando rápidamente de tamaño, especialmente en el número de pares, y a partir del saber popular y la experiencia de los integrantes del equipo de trabajo, los jóvenes son los usuarios que más "amigos" tienen en su red virtual de Facebook.

En virtud de lo cual se realizó una búsqueda de publicaciones y antecedentes sobre la temática. Se encontraron escasos documentos e investigaciones de validez científica sobre el tema a abordar. Se publican con relativa periodicidad informes sobre trabajos de investigación personales, publicaciones de tipo académicas y artículos periodísticos en revistas técnicas o de contenido general a modo de opiniones o análisis desde distintas ópticas tanto en la Argentina como en el exterior.

A modo ejemplo se citan algunas conclusiones:

- En el documento Modelos de interacción de los adolescentes en contextos presenciales y virtuales, sus autores concluyen: "Las TIC ofrecen a las familias un vehículo de comunicación que requiere flexibilidad y adaptación además de un aprendizaje continuo. Todos los miembros de la familia se enfrentan a un proceso de aprendizaje a medida que adoptan nuevas tecnologías. En este proceso, tanto adultos como adolescentes tienen como tarea el crear espacios de conversación que desenfatizen la necesidad de gestionar o controlar la interacción adolescente-tecnología…temas como la confidencialidad, el control de la privacidad y la seguridad no se resuelven con apretar un par de teclas del ordenador. Se requiere de un proceso reflexivo que puede equilibrar los

riesgos de la creatividad, la imaginación y el descubrimiento, con los de la seguridad, la estabilidad y la previsibilidad.". (Prendes, M. ; Sánchez, M. ; Serrano, J., 2011)[5]

- Gregorio C. y Ornelas L. (2011) [6] en su libro Protección de datos personales en las redes sociales digitales: en particular de niños y adolescentes, enuncian: "A los niños, niñas y adolescentes se les debe formar, en un lenguaje fácil, que permita la comprensión del espíritu de la protección de la vida privada de ellos y los demás, en: el uso responsable y seguro del Internet y las redes sociales digitales; en que el Internet es un espacio con normas y que las acciones allí tienen consecuencias, en particular debe hacérseles conocer que: la distribución de pornografía, el acoso, la discriminación, la promoción del odio racial, la difamación, la violencia son ilegales y penados por la ley; en el respeto a la vida privada, buen nombre e intimidad de terceras personas; en las posibles responsabilidades civiles, penales o administrativas derivadas del uso abusivo del Internet; debe formarse a niños, niñas y adolescentes en el uso responsable y seguro del Internet, en particular sobre las políticas de privacidad, seguridad y de alertas de las distintas redes y en la "incertidumbre" que rodea a la veracidad de la información en Internet y a buscar y discriminar las fuentes; y, debe recordarse que el uso de seudónimos es aceptado, siempre que estos no sirvan para el engañar o confundir sobre la identidad real, pero ello debe informarse sobre los riesgos de ser engañados sobre la identidad de la otra persona y los robos o la suplantación de identidad."

3. Justificación y relevancia social y/o académica de investigar el problema

Desde el punto de vista social la temática despierta el interés del equipo de trabajo y la sociedad en general. Ante la presencia de un fenómeno que se expande a gran velocidad en un mundo mediado por las tecnologías, consideramos importante conocer y comprender su impacto y estar preparados para responder las inquietudes de los distintos actores de la sociedad: jóvenes, padres, docentes, profesionales de la salud, etc. Por tal motivo se decidió considerar a los adolescentes como muestra del presente proyecto entendiendo que se encuentran en un período crítico, de adquisición de nuevos roles, necesidades y demandas, durante el cual recibir apoyo, así como el que se necesite brindar, pasa por cambios dramáticos dado que los adolescentes comienzan a alejarse de sus padres y a orientarse hacia sus pares, como ya se mencionó anteriormente.

En lo que respecta al ámbito académico, el conocimiento generado y sistematizado, servirá como insumo de trabajo, especialmente para los profesionales del ámbito de la psicología y abogacía que contarán con nuevos elementos de estudio y análisis para las cátedras y/o las prácticas profesionales de los alumnos.

Asimismo los resultados podrán ser utilizados como disparadores de nuevos proyectos de investigación que continúen profundizando sobre la temática.

4. Metodología

En este proyecto de investigación se utiliza fundamentalmente una metodología cualitativa, respondiendo de esta manera al encuadre social del mismo. Complementando, de ser necesario, con aportes provenientes de lo cuantitativo.

En función de los objetivos definidos se propone una investigación de tipo exploratoria-descriptiva. La investigación descriptiva es aquélla que busca desarrollar una imagen o fiel representación (descripción) del fenómeno estudiado a partir de sus características.

Técnicas, instrumentos y procedimientos para el relevamiento, procesamiento y análisis de la información

Características y tamaño de la Muestra

Se constituye por un grupo de 30 alumnos de entre 16 y 18 años que se encuentran realizando la educación media en una escuela secundaria pública de la ciudad de Rafaela y otra de la ciudad de Sunchales.

Realización de Entrevistas

Se realizaron entrevistas semiestructuradas, a partir de la conformación de una guía de preguntas abiertas. La etapa de definición de la entrevista constituyó una tarea clave para el trabajo. Las preguntas versan sobre distintos temas y a modo ejemplo se listan algunos ejes temáticos o dimensiones en base a las siguientes categorías:

Tabla 1. Categorías y dimensiones entrevistas

CATEGORIA	DIMENSIONES
Enfoque Psicológico	Social (A)
	Personalidad (B)
Enfoque Informático	Seguridad de la Información (C)
	Conocimientos técnicos sobre el uso de Facebook (D)

Tabla 2. Ejes según dimensiones

	EJES
(A)	Grupos de referencia. Participación. Modos de relacionarse.

(B)	Características personales. Modos de comunicación e interacción. Temas de interés. Rol dentro de un grupo.
(C)	Conocimiento de conceptos de seguridad, confidencialidad, autenticidad. Mecanismos para el resguardo de la seguridad.
(D)	Configuración de su perfil. Configuración de la privacidad. Solicitudes de contacto. Administración del perfil biográfico en general.

Cada una de las entrevistas fue grabada por medio de un dispositivo portátil. Luego, por medio de su escucha se realizó la trascripción.

Rastreo de Perfiles de Facebook

En base a los datos recolectados en las entrevistas, se efectuó un rastreo de los perfiles de Facebook de cada uno de los jóvenes, revisando varios aspectos de su configuración pública. Luego se confeccionó una tabla general para la comparación de la información recabada de manera de facilitar su análisis.

Los parámetros que se analizaron de cada uno de los perfiles están representados en la siguiente tabla:

Tabla 3. Parámetros perfiles

Fotos y Amigos	
Foto Perfil	Cantidad de Fotos
Foto Portada	Cantidad de Fotos en las que aparece
Cantidad de Fotos Perfil	Cantidad Amigos
Periodicidad de Cambio Foto Perfil	
Información	
Formación y Empleo	Información Básica
Lugares de Residencia	Acontecimientos Importantes
Relación	Información de Contacto
Familia	
Publicaciones	
Comentarios Fotos Amigos	Comentarios sobre otras Publicaciones (Imágenes, textos, noticias, links, etc.)
Comentarios Generales Muro	Contenido Compartido
Documentos Subidos	
Intereses	
Describe Películas	Describe Eventos a los que asiste
Describe Música	Describe Grupos a los que pertenece
Describe Libros	Describe "Me Gusta"
Describe Deportes	Sigue a otros Perfiles / Cuántos
Describe Aplicaciones y Juegos	
Acceso Perfil Público	
Acceso a Fotos	Aceptó Solicitud Amistad
Acceso a Lista de Amigos	Demora en Aceptar la Solicitud (en días)
Publicaciones	Lenguaje empleado
Intereses	

Para llevar a adelante cada uno de los rastreos de los perfiles siguiendo los parámetros indicados, se implementó la siguiente metodología:

- Para ubicar el perfil de Facebook de la persona en primer lugar se utilizó el buscador que otorga el sitio de manera básica. Por medio del ingreso del nombre, apellido o dirección de correo electrónico se podía dar con el perfil indicado. En caso de no encontrarlo se hizo uso de los filtros como: ciudad, escuela, edad, etc. Si hasta ese momento no aparecía la persona en cuestión, la técnica se orientó a las fotos de sus compañeros de escuela para observar si el alumno entrevistado participaba de etiquetas, comentarios y/o indicaciones "me gusta". En ciertas ocasiones, si un perfil era dudoso a causa del uso de datos ficticios, la url (dirección del perfil) permitió despejar las dudas, ya que en la misma aparece el nombre y apellido real de la persona. Para tener la certeza de estar rastreando el perfil correcto siempre se escuchaban las entrevistas.
- Para contar la cantidad de fotos de perfil y/o portada, se ingresó al álbum "Fotos de Perfil" y/o "Fotos de Portada" de la persona y se contabilizó una por una.
- Para determinar la periodicidad de cambio de foto de perfil se realizó el siguiente cálculo:
- Cantidad de Fotos de Perfil / (Fecha de última foto de perfil – Fecha de primer foto de perfil)
- Para contar la cantidad de fotos en las que aparecía una persona se ingresó a la sección "Fotos en las que aparece…" y se registró una por una.
- La cantidad de amigos se obtuvo por medio de un código especial usado en el navegador web. A través de una dirección particular, usando el nombre de usuario como parámetro, se obtuvieron parámetros como la "cantidad total de amigos".[1]
- Para la sección de "Información" se ingresó a la pestaña "Información" de la

1 Detalle del código: http://siddharta.esdebian.org/49578/obtener numero-total-amigos-facebook-traves-terminal

biografía de la persona y se buscó cada uno de los datos suministrados.

- En cuanto a las "Publicaciones", se ingresó a la biografía de la persona y se observó el tipo de información de publicación, videos, textos, fotos, frases, etc. Además se consideró el lenguaje empleado a la hora de redactar cada uno de los comentarios.

Toda la información recabada de los rastreos fue consolidada en una planilla para poder tener una óptica cuantitativa y que de esa manera facilite su análisis posterior.

Como complemento a esta actividad, se crearon dos perfiles de Facebook ficticios, uno de una joven y otro de un joven para enviar solicitudes de amistad a cada uno de los entrevistados de manera de registrar la aceptación o rechazo de la misma.

En cada uno de los perfiles creados se usaron fotos de "dominio público" sin relación con personas existentes en la realidad. En cuanto a los nombres para las dos personas, los mismos fueron probados en primer lugar para ver la cantidad de similitudes en Facebook.

5. Resultados

- **La perspectiva Informática: Rastreo de Perfiles**

La primera categoría de análisis para los resultados fue el rastreo de perfiles. Como resultado de los mismos, se desarrolló una planilla con los datos de cada evaluación.

De 30 perfiles revisados obtuvimos acceso a 1441 fotos de "perfil". Esto da un promedio de 48 fotos por usuario. Es importante destacar que las mismas tienen acceso público lo que significa que cualquier persona con una cuenta en Facebook puede tener ingreso a ellas. Esto pone en riesgo la privacidad de los datos de los usuarios e indica que la configuración que se le efectuó a ese perfil puede dejar vulnerable varios aspectos de cada usuario.

En relación al tipo de foto seleccionada, la mayoría, 20 casos, se orientó a fotografías de la cara y/o el cuerpo completo.

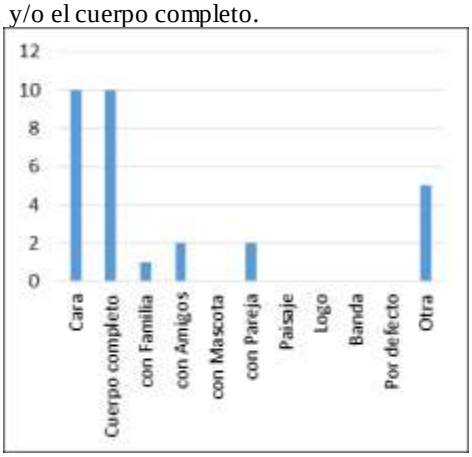


Gráfico 1. Tipos de fotos de "perfil"

Esta característica de selección del tipo de foto evidencia que la persona puede ser fácilmente reconocida en una primera búsqueda del perfil, ya que tanto la foto de "perfil" como la de "portada" es la carta de presentación a la hora de encontrar un usuario en esta red social.

Otro dato analizado fue la periodicidad con que se actualizan las fotos de "perfil" para cada usuario. Por medio del siguiente gráfico mostramos el resultado:

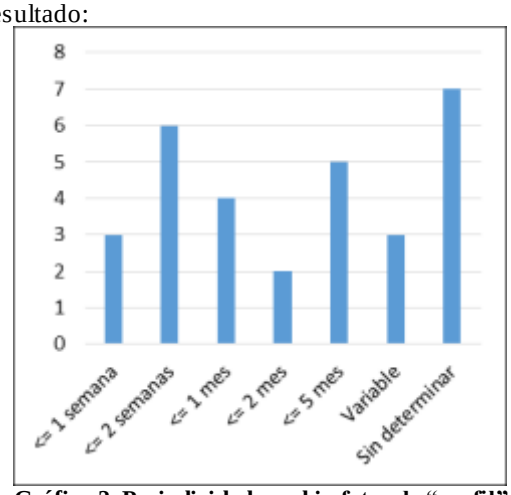


Gráfico 2. Periodicidad cambio fotos de "perfil"

Lo que notamos es que la mayoría de los usuarios analizados cambian sus fotos de "perfil" como máximo en un tiempo de 5 meses.

Con más precisión podemos decir que dentro de ese período, cada dos semanas aproximadamente es la tasa de actualización más usada.

En el caso de las fotos de "Portada", que ocupan la parte superior de la página de cada perfil de Facebook, evidenciamos que la mayoría de los usuarios opta por una en la que esté su grupo de amigos o por otra que no ingresa dentro de nuestras categorías propuestas. Es decir, este tipo de foto representa más su red social real, pero también un espacio donde colocar una "imagen" de tema libre. Esto es distinto al caso de la foto de "perfil" dónde se evidencia más la tendencia a centrarse concretamente en la persona misma.



Gráfico 3. Tipos de fotos de "portada"

Para continuar con el tema de las fotos, un aspecto que analizamos fue la cantidad total que

cada usuario tiene en su perfil para acceso completamente público. Un total de 10353 fotos están subidas para todos los involucrados en nuestra muestra. Esto da un promedio de 345 fotografías para cada perfil. Es un valor alto, considerando que es contenido de acceso masivo por cualquier usuario de Facebook. En este punto vuelve a aparecer el concepto de privacidad de los datos de los usuarios y los posibles riesgos que pueden surgir por permitir el acceso público a un gran número de fotos.

En base a la cantidad de fotos mencionadas anteriormente, realizamos un análisis para detectar en cuántas de ellas está presente el dueño del perfil. Llegamos a la suma de 2323 fotos con la presencia del usuario, dando un promedio de 77 para cada uno. Si consideramos el total de fotografías subidas para cada perfil podemos indicar que aproximadamente un 22% de las mismas tienen presente a la persona.

La cantidad de amigos de cada perfil de usuario fue otro aspecto de análisis. De los 30 perfiles revisados, se tuvo la cuenta de 27701 "amigos" enlazados. Esto es un promedio de 923 amigos por usuario. El hecho de poder ver la cantidad de amigos de un perfil significa que se puede conocer el nombre de esa persona y por lo menos su foto de "perfil" y "portada". De esto podemos desprender el concepto de efecto dominó; es decir, desde un perfil conocido por nosotros tener la posibilidad de ir accediendo a otros contactos desconocidos para evaluar su perfil. De lo anterior, podemos decir que prácticamente no hay límites a la hora de encontrar usuarios (personas) en una red social como Facebook; llevará más tiempo, más esfuerzo, pero las condiciones están dadas para conocer aspectos de otros participantes.

Una característica de un perfil en Facebook es que se puede configurar si se permite el acceso a la lista de amigos de cada usuario. De los 30 alumnos que tomamos como muestra, en 17 casos pudimos ingresar al detalle completo de amigos de su perfil. El resto (13) tenía la opción bloqueada. De éstos últimos, en 6 casos la lista de amigos no estaba accesible de ninguna manera pero en los 7 casos restantes, se podía tener acceso a los "amigos" en común. Para nuestro caso de análisis, al ser una muestra de desconocidos, no tuvimos esta característica en común.

Consideramos un dato importante que más del 50% de los perfiles analizados tenían acceso completo a los "amigos" siendo estos una cantidad promedio de 923 aproximadamente.

Una de las secciones de un perfil en Facebook es la relacionada con la "información" del usuario. A continuación mostramos los resultados del análisis de los perfiles de los alumnos de nuestra muestra:

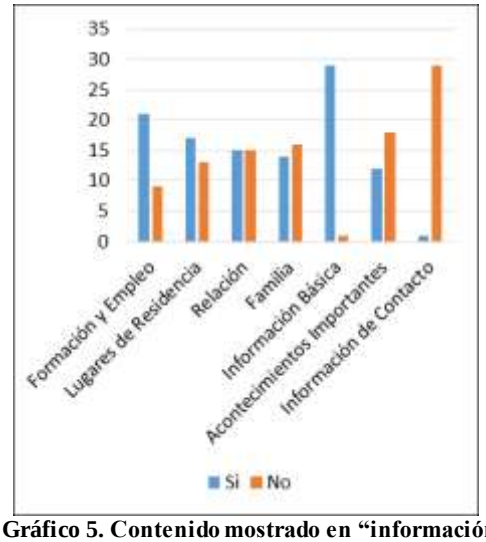


Gráfico 5. Contenido mostrado en "información"

Como se puede observar, la tendencia de los perfiles analizados muestra que la "información básica" es lo que más se publica. De todos modos es importante destacar que el "lugar de residencia" se muestra en más del 50% de los perfiles y la "formación y empleo" también. En este último caso, lo que se publica es la escuela a la que pertenecen.

Siguiendo con el análisis de los resultados del rastreo de perfiles, mostramos a continuación lo relacionado a "intereses" de los usuarios:

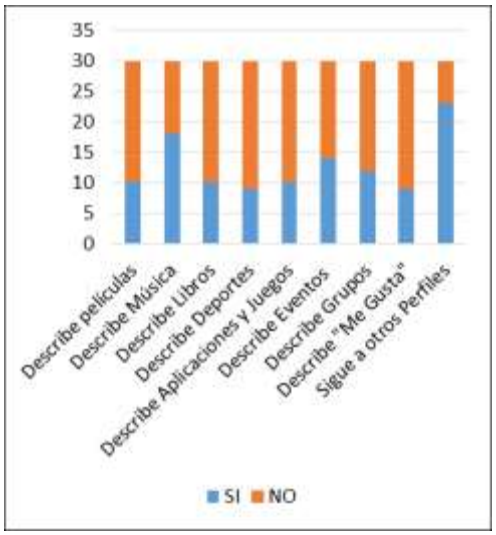


Gráfico 5. Tipos de "intereses"

Podemos destacar que de entre los intereses que se muestran, lo relacionado con los "perfiles" que sigue un usuario representa la mayor incidencia frente a los alumnos analizados. En gran medida esto sucede porque es un práctica común "seguir" perfiles de famosos, músicos, actores, etc.

Recolectamos en total 282 perfiles "seguidos" lo que da un promedio de 12 perfiles por cada uno de los alumnos vinculados.

Para culminar el desarrollo de los resultados parciales, vamos a mostrar el comportamiento de

los entrevistados frente a las solicitudes de amistad realizadas desde dos perfiles de Facebook ficticios. Desde cada uno de ellos se mandaron pedidos de amistad a la cantidad de alumnos involucrados en la muestra. En total fueron 30 solicitudes desde cada perfil. De lo anterior podemos decir qué como resultados parciales:

- Para el perfil de una joven se logró la aceptación de 9 usuarios. Del resto, 2 eliminaron el pedido y los demás están pendientes de respuesta.
- Para el perfil de un joven se consiguió la aceptación de 12 usuarios. Del resto, uno eliminó el pedido y los demás están pendientes de contestar.

Como se puede observar se logró un interesante número de solicitudes aceptadas. La aceptación se dio sobre perfiles desconocidos y que no existen en la realidad.

- **La perspectiva de la Psicología: Análisis de Entrevistas**

Desde el punto de vista de la Psicología, el desarrollo tiene que ver con la interpretación de cada una de las entrevistas efectuadas a los alumnos. Las mismas están enfocadas en las dimensiones descriptas en la Tabla 1. Por eso la tarea principal es agrupar las respuestas dentro de estos ejes/categorías con el objetivo de encontrar patrones de respuesta que orienten la definición de los perfiles biográficos de cada adolescente.

A partir de las entrevistas realizadas, se analizaron las características de los adolescentes teniendo en cuenta, aspectos sociales y personales. Dentro de estas dimensiones se incluyeron: características personales (emocionales, conductuales, caracterológicas) modos de comunicación e interacción, roles y temas de interés, además de los grupos de referencia, participación y modos de relación; definidas por los propios adolescentes.

Las preguntas realizadas movilizaron una actividad de introspección, entendida como la capacidad para la reflexión y conocimiento del sujeto sobre los propios sentimientos y emociones, es decir, de su mundo interno observándose de este modo algunos casos donde hubo mayor dificultad para este registro y observación.

La información se organizó en función de una tipología basada en dos variantes caracterológicas: el tipo introvertido y el extravertido. El tipo introvertido se centra en su mundo interior, no tiene facilidad para establecer relaciones sociales. Se lo concibe, en general, como distante de las personas y aislados o retraído en sí mismo. Tienden a basarse fuertemente en su evaluación interna a la hora de formular sus juicios. El extravertido está centrado en el mundo exterior, tiene gran facilidad para las relaciones sociales, tienden a utilizar la relación con el mundo externo como fuente de gratificación de sus necesidades básicas. Los frecuentes y a veces excesivos intercambios afectivos es lo que más los distingue, ya que muestran afecto hacia su entorno, más fácilmente que los introvertidos.

Por lo tanto, se realiza una recopilación de los datos que implican no solo la conformación de su red social (familia, y amigos) sino también las características que definen estos dos tipos caracterológicos.

De las 30 entrevistas realizadas se pudieron obtener 13 perfiles de personas con características extravertidas reuniendo las condiciones de sociable, comunicativo, abierto, expansivo.

Del total, se obtuvo 8 adolescentes con características introvertidas, que se definen por presentar signos de timidez, retraimiento, falta de comunicación y expresión de sentimientos y emociones.

Los restantes 9 perfiles de personas no pudieron determinarse en función de un tipo específico. Según las manifestaciones que presentan los adolescentes de este grupo las características oscilan en introvertidas - extravertidas.

En función de este análisis se pretende conocer si existe alguna vinculación entre el modo de presentación de cada adolescente en un contexto individual y social, en comparación al contexto virtual. Para lo cual se analiza el uso de la red social Facebook, la actividad que realiza en las mismas y la configuración de privacidad.

De esto se desprende que de los adolescentes con características introvertidas, se presentan 2 casos donde la utilización de las redes sociales no se realiza con demasiada frecuencia. Se mantiene la frecuencia y el uso de la red social Facebook en los restantes 7 casos. De lo cual podemos inducir, que gran parte de los adolescentes con estas características, realiza un uso frecuente en las redes sociales. Además, 5 adolescentes de este grupo tienen otras redes sociales, como ser Twitter.

De los perfiles con características extravertidas, de un total de 13 adolescentes, en 6 casos se presenta un uso activo y diario. Los restantes 7 adolescentes declaran no usarlo con mucha frecuencia, su uso no es diario, depende de las circunstancias. De los adolescentes de este grupo, 10 tienen otras redes sociales, Twitter y 2 casos donde tienen más de dos redes sociales.

De las 8 personas con características introvertidas, en la elección de la foto de perfil 2 adolescentes indicaron que prefieren no estar solos en la foto. Los restantes indicaron que eligen su foto en función de la apariencia, si salen bien, si le gusta a la gente.

De las 13 personas con características extrovertidas, en 10 casos manifiestan poner fotos, donde salgan bien, también consideran que sean fotos decentes.

A la hora de elegir foto de portada la gran mayoría coincide en colocar, foto con amigos, familia, club favorito, frase, banda de música.

En el perfil de Facebook existe la opción de describir los propios intereses entre los cuales se encuentran: películas, música, programas de TV, libros, deportes, aplicaciones y juegos, entre otros. En 17 casos, del total, describen intereses y al menos 13 casos coinciden con alguno de los intereses que se describen en la entrevista.

Del grupo de extravertidos (13 casos), al menos 8 describe intereses. Del grupo de introvertidos (8 casos), en 6 casos se describen los intereses.

Tanto en el uso de la herramienta y como en las actividades que realizan en Facebook, los adolescentes entrevistados señalan un modo comportamiento anterior que se diferencia del actual en relación a la configuración de la privacidad. Este cambio se asocia a una mayor conciencia de riesgo.

Se registró que en 27 casos del total, se establecen condiciones cuando acepta o solicita amistades. Dentro de las condiciones se encuentran como las más relevantes: conocer a la persona (de vista, personalmente, de la escuela, si es pariente) y la cantidad de amigos en común. En un solo caso se establece como condición el sexo de la persona, aceptando solo al mismo género lo cual tiene que ver con un acuerdo con su pareja por inconvenientes surgidos a partir de Facebook.

De estos 27 casos corroborados en el perfil biográfico digital, 8 adolescentes aceptaron la solicitud de los perfiles creados por el equipo de trabajo. Se registró en 3 casos, la aceptación del perfil del sexo opuesto, y en los restantes se acepta ambos sexos.

De esta manera podemos constatar que no se cumple con la condición de "conocer a la persona" que están agregando, quedando un total de 19 adolescentes que cumplen con las condiciones y no aceptan perfiles desconocidos.

En un solo caso se registró un impedimento, siendo "no permitida" la solicitud, verificando de este modo la configuración de un perfil privado que impide recibir solicitudes de amistad.

De los 3 casos restantes (de género masculino) declaran que a la hora de aceptar personas, no ponen condiciones, pero si esta persona les resulta extraña o los molesta, optan por eliminarlo o bloquearlo. Se confirma este dato siendo que en 2 casos se aceptan ambos perfiles, y en el restante se acepta y luego se elimina a la persona agregada.

En todos los casos, el envío de solicitudes de amistad se realiza por alguna motivación, como en el caso de una persona que conoció y que aún no la tiene en Facebook.

Podemos concluir que existe conciencia sobre el riesgo de publicar determinadas fotos, datos o información. Se hace mención a los peligros que implica, publicar dando información sobre lo que se encuentran haciendo en el momento, y también sobre los movimientos de cada persona. Casi el total de los entrevistados reconoce que se pueden cometer delitos mediante Facebook u otras redes sociales. En un solo caso se manifiesta no estar al tanto, y por lo tanto que le gustaría recibir información. Al momento de tomar una medida en función de considerarse posibles víctimas de delitos, en 21 casos se hace mención a las herramientas que ofrece Facebook: bloquear o denunciar (2 adolescentes la han utilizado). De los restantes, 3 no saben utilizar las herramientas, en otros 4 casos recurrirían a la policía o a un abogado, 2 casos en donde alertarían a sus amigos y se resguardarían bajo condiciones de privacidad de perfil.

Del total, 5 adolescentes respondieron que se lo comunicarían a alguno sus familiares.

Por último, resulta interesante conocer lo que opinan los adolescentes acerca del uso de los demás en Facebook. La gran mayoría (23 casos) se manifiestan en desacuerdo o molestos cuando se publica sobre la vida privada, mostrando fotos, publicando sus problemas, generando conflictos a partir de comentarios.

También, se hace evidente reconocer que los adolescentes en su mayoría han manifestado que declinó el uso de la herramienta, en función de la aparición de nuevas formas de comunicación. En 21 casos se sostiene que no es fundamental, puede ser reemplazado, o que si bien antes no existía, puede seguir sin existir.

Con respecto a las formas de comunicación con sus pares, la gran mayoría de los adolescentes de esta muestra optan hoy en día, por mensajes instantáneos, utilizando el programa WhatsApp. En 24 casos de los 30 analizados, se privilegia esta vía como forma de comunicación. Algunos de los motivos de esta elección tienen que ver con la disponibilidad de la función en el teléfono, y en donde Facebook requiere en algunos casos de una computadora. De esos 24 casos 9 también utilizan el recurso de Facebook. En menor medida (6 casos) prefieren realizar llamadas o utilizar los mensajes. Se presenta un solo caso donde ni el adolescente, ni sus amigos tienen celular, solicitando a que alguien de la familia sea el portador de la comunicación por vía de mensajes telefónicos.

Los restantes 9 casos, del total, sostienen que es una herramienta fundamental, considerando que para algunos sirvió para encontrar familiares o reencontrarse con personas que formaron parte de la vida de cada uno (4 casos) y sigue conformando un medio importante para la comunicación.

- **La perspectiva del Derecho: Cuestiones de Seguridad**

En cuanto a la óptica del Derecho, se buscaron identificar perfiles biográficos digitales que tengan una situación de vulnerabilidad explícita que

puedan llevar a que se comentan delitos informáticos. Por medio del marco jurídico se buscó brindar información sobre los mecanismos legales para hacer frente a estas situaciones que pongan en riesgo las condiciones que hacen a un perfil en Facebook seguro. Para esta sección se tuvo en cuenta el material obtenido en las participaciones de miembros del equipo en el "VII Congreso de Derecho Informático" de la UNL Santa Fe y la "Jornada de Seguridad Informática" de la UCSE DAR (Anexo V).

A continuación se describen características o particularidades sobre la temática luego de analizado el material y la legislación al respecto.

Particularmente vinculado al Derecho Penal, la Privacidad es un bien jurídico por él tutelado (se incorporó expresamente con la ley 26.388, la cual la agregó al epígrafe original de Capítulo III, del título V, Libro II del Código Penal, el cual es actualmente: Violación de Secretos y de la Privacidad").

Según la Corte Suprema de Justicia de la Nación, el derecho a la privacidad e intimidad, fundado en el artículo 19 de la Constitución Nacional, protege jurídicamente en relación directa con la libertad individual un ámbito de autonomía personal, así como acciones, hechos y datos que, conforme a las formas de vida acogidas por la sociedad, están reservadas al individuo, y cuyo conocimiento y divulgación por extraños implica peligro real o potencial para la misma intimidad. (CSJN, 11-12-84, E. D. 112-239).

Asimismo la protección de la intimidad se volcó específicamente al Código Civil, en su artículo 1071 bis, agregado por la ley 21.173, en la siguiente forma: "El que arbitrariamente se entrometiere en la vida ajena publicando retratos, difundiendo correspondencia, mortificando a otro en sus costumbres o sentimientos, o perturbando de cualquier modo su intimidad, y el hecho no fuere un delito penal, será obligado a cesar en tales actividades, si antes no hubieren cesado, y a pagar una indemnización que fijará equitativamente el juez, de acuerdo con las circunstancias; además, podrá éste, a pedido del agraviado, ordenar la publicación de la sentencia en un diario o periódico del lugar, si esta medida fuese procedente para una adecuada reparación"

Con relación a las redes sociales el abanico de posibilidades de infracción a los derechos de intimidad y privacidad es muy amplio, ya sean estos ilícitos cometidos por otros usuarios de las redes o por terceros.

En un Estudio sobre la Privacidad de los datos personales y la seguridad de la información en las redes sociales online de la Agencia Española de Protección de Datos se señaló que con relación a la Privacidad personal: a pesar de que sean los usuarios los que voluntariamente publican sus datos, los efectos sobre la privacidad pueden tener un alcance mayor al que consideran en un primer momento ya que estas plataformas disponen de potentes herramientas de intercambio de información, la capacidad de procesamiento y el análisis de la información facilitada por los usuarios. Y por su parte con relación a la Privacidad de terceros: es esencial que los usuarios tengan en cuenta que la publicación de contenidos con información y datos respecto a terceros no puede ser realizada si éstos no han autorizado expresamente su publicación, pudiendo solicitar su retirada de forma inmediata.

Algunas de las problemáticas jurídicas que afectan a las redes sociales son:

- Protección de los derechos al honor:

  "El honor es la propia personalidad entendida como la suma de cualidades físicas, morales, jurídicas, sociales y profesionales, valiosas para la comunidad, atribuibles a las personas. Cuando el que atribuye esas cualidades es el propio interesado se habla de honor subjetivo u honra de la persona. Cuando los que le atribuyen esas cualidades al interesado son los terceros, se habla de honor objetivo o crédito de la persona". (Nuñez Ricardo C., Manual de Derecho Penal, parte especial, 2da. Edición actualizada por Victor F. Reinaldo, Ed. Marcos Lerner, año 1999)

  La ley argentina, a través del Código Penal en su capítulo referido a "Delitos contra el Honor" tutela el derecho al honor, el decoro y la reputación de las personas, es decir su personalidad moral.

  Injurias: La injuria es la ofensa genérica al honor ajeno; que puede ser a la honra de la persona (honor subjetivo), y en este caso es una lesión al derecho que tienen las personas a que los terceros respeten las cualidades que ellos le asignan a su personalidad. O bien puede ser una ofensa al crédito de la persona (honor objetivo), y en este caso es una lesión al derecho de las personas a que no se perjudique la opinión que sobre su personalidad tengan o puedan tener los terceros. Es la fama o reputación.

  Calumnias: La calumnia se encuentra regulada en el artículo 109 del Código penal, siendo la figura agravada de los delitos contra el honor. Dicha figura penal es definida por aquel cuerpo normativo como "la falsa imputación de un delito que dé lugar a la acción penal pública". En este orden, podemos afirmar que los usuarios que realicen actividades difamatorias mediante la utilización de las redes sociales online, pueden ser pasible de este tipo de delitos.

- El derecho a la propia imagen:

  La imagen es la figura, la fisonomía que la persona tiene, como individuo único e irrepetible.

  El derecho a la propia imagen atribuye al individuo la capacidad de ejercer un control sobre la captación, grabación, uso y difusión de su imagen entendida como representación gráfica de la figura humana, y también de su voz. El derecho a la propia imagen no sólo atiende a los aspectos más concretos y definitorios del mismo, la facultad de consentir en la captación o difusión de imágenes que reproduzcan la figura humana, sino también a la información que éstas revelan y a su directa relación con las intromisiones en la vida privada. De hecho, debe considerarse que es esta relación con la vida privada la que dota de relevancia constitucional a la protección de la imagen y, en su caso, de la voz.

  En este sentido, las redes sociales online son un ámbito susceptible en donde se pueden cometer gran cantidad de infracciones sobre el derecho a la imagen. Simplemente si se piensa que una persona publica una fotografía, y esta inmediatamente se empieza a distribuir por toda la red de usuarios, llegando a contactos o publicaciones no deseadas por el usuario. En muchos casos hay actuaciones de los usuarios o de las propias redes que realizan respecto de la imagen y otros datos que se escapan de la propia esfera de la protección de datos, como por ejemplo las cesiones ilegales de imágenes para finalidad comercial sin consentimiento previo e informado del titular.

- Derecho a la intimidad y privacidad de los usuarios:

  El derecho a la intimidad tiene por objeto la protección de la esfera más íntima de la persona, y se encuentra estrechamente ligado a la protección de la dignidad del individuo. Es el derecho que tiene toda persona humana a que sea respetada su vida privada y familiar, el derecho a no ser objeto de injerencias arbitrarias en la zona espiritual íntima y reservada de una persona o de un grupo, especialmente de una familia.

  Entre las posibles situaciones de riesgo para la protección de la intimidad, cabe señalar:

  • En el momento del registro de alta como usuario, en la medida en que no sea configurado correctamente el nivel de privacidad del perfil, así como por el hecho de que sea publicada información sensible desde el inicio de la actividad en la red.

  • En el momento de participación en la red como usuario, en el supuesto que el grado de información, datos e imágenes publicados puedan ser excesivos y afectar a la privacidad, tanto personal como de terceros.

  Por último, es importante tener en cuenta que en la gran mayoría de ocasiones, las redes sociales permiten a los motores de búsqueda de Internet indexar en sus búsquedas los perfiles de los usuarios, junto con información de contacto y de perfiles amigos, lo que puede suponer otro riesgo para la protección de la privacidad, además de dificultar el proceso de eliminación de su información en Internet.

  En estos casos, la persona afectada podrá reclamar los daños y perjuicios ocasionados mediante una acción judicial.

- Protección de los datos personales:

  Ley 25.326 de Protección de Datos Personales protege los datos personales en bases de datos ya sean de acceso público o privadas destinadas a publicar informes.

  Esta ley no protege la autoría o propiedad de la base de datos en sí, sino el derecho del titular de los datos (persona a quien refiere los datos) al acceso, honor, intimidad y correcto uso y tratamiento de los mismos según lo establece el artículo 43 de la Constitución Nacional.

  Las redes sociales generalistas o de ocio cuentan con un nivel de riesgo superior al de las redes sociales profesionales, dado que los usuarios exponen no sólo sus datos de contacto o información profesional (formación, experiencia laboral), sino que se pueden exponer de manera pública las vivencias, gustos, ideología y experiencias del usuario, lo que conlleva que el número de datos de carácter personal puestos a disposición del público es mayor que en las redes sociales de tipo profesional.

  Entre las principales situaciones, cabe señalar las siguientes:

  • Existe un problema derivado de la falta de toma de conciencia real por parte de los usuarios de que sus datos personales serán accesibles por cualquier persona y del valor que éstos pueden llegar a alcanzar en el mercado. En muchos casos, los usuarios hacen completamente públicos datos y características personales que en ningún caso expondrían en la vida cotidiana como ideología, orientación sexual y religiosa, etc.

  • Los datos personales pueden ser utilizados por terceros usuarios malintencionados de forma ilícita.

  • Existe la posibilidad de que traten y publiquen en Internet información falsa o sin autorización del usuario, generando situaciones jurídicas perseguibles que pueden llegar a derivarse de este hecho.

• El hecho de que, a través de las condiciones de registro aceptadas por los usuarios, éstos cedan derechos plenos e ilimitados sobre todos aquellos contenidos propios que alojen en la plataforma, de manera que pueden ser explotados económicamente por parte de la red social.

- Protección de la propiedad intelectual:

La facilidad de reproducción y distribución de contenidos hacen de Internet uno de los principales medios de crecimiento para los contenidos de propiedad intelectual, al tiempo que supone uno de los principales retos en lo que respecta al control y protección de los derechos de autor, en la medida en que los contenidos se encuentran en formato digital y, por tanto, su distribución y comunicación pública es mucho más sencilla que en otro tipo de formato. (Estudio sobre la Privacidad de los datos personales y la seguridad de la información en las redes sociales online. Agencia Española de Protección de Datos).

El modelo de generación de contenidos ha variado en gran medida respecto al existente antes del surgimiento de la Web 2.0, dado que hoy en día los contenidos no son generados por los propios autores en exclusiva, sino que cualquier usuario tiene la capacidad de generar y difundir sus obras de propiedad intelectual, convirtiéndose así en autor y potencial productor y distribuidor.

En la legislación Argentina se encuentra la Ley 11.723 de Propiedad intelectual que protege toda producción científica, literaria, artística o didáctica sea cual fuere el procedimiento de reproducción, así como también los programas informáticos.

Desde el punto de vista de los posibles riesgos que se pueden producir contra la protección de la propiedad intelectual en Internet, en general, y en los servicios de redes sociales y plataformas colaborativas, en particular, deben diferenciarse dos situaciones en origen:

• De un lado, se ven afectados los contenidos que son titularidad de terceros y que el usuario decide publicar dentro de la red social sin autorización de los titulares del derecho de propiedad intelectual.

En estos supuestos el usuario se encuentra violando derechos de autor, y en consecuencia deberá responder por los daños y perjuicios.

• De otro lado, las implicaciones jurídicas sobre las obras que sean titularidad de los propios usuarios y que éstos deciden compartir o hacer públicas a través de estas redes y plataformas.

Una problemática que se presenta en relación a este punto, es cuando los usuarios deciden dar de baja su suscripción o "cuenta" a la red social, siendo que en ese caso se debería dejar de difundir y publicar los contenidos de su autoría.

La realidad es que en la práctica esto no ocurre de esta manera, ya que los sitios web de redes sociales siguen publicando los contenidos del usuario argumentando que se estos deben quedar visibles para los demás usuarios de la red con quien se compartieron en su momento.

A causa de este tipo de situaciones, Facebook.com decidió modificar unilateralmente sus términos y condiciones estableciendo que los usuarios cedían y licenciaban de manera irrevocable y perpetua sus contenidos a la empresa , argumentando la necesidad de seguir contando con esos contenidos online en caso de que el usuario diera de baja su cuenta.

6. Referencias

[1] Hernández D.; Torruella G. (2011) *Identidades digitales: límites poco claros. Cuadernos De Pedagogía*. Recuperado en mayo de 2013 en http://ci20.pbworks.com/f/74794642 identidades digitales.pdf

[2] Dabas, E. (2001) *Redes sociales: niveles de abordaje en la intervención y organización en red*. Recuperado en junio de 2013 en http://www.santafe.gov.ar/index.php/web/content/download/71292/345905/

[3] Tenzer, S. ; Ferro, O. ; Palacios, N. (2009*) Redes Sociales Virtuales: personas, sociedad, empresa*. España Recuperado en mayo de 2013 en http://salonvirtual.upel.edu.ve/pluginfile.php/19830/mod resource/content/0/Redes_Sociales/Redes.pdf

[4] Arón A.M., Nitsche R. Rosenbluth A. (1995) *Redes sociales de adolescentes: un estudio descriptivo comparativo*. Revista Psikhe Pontificia Universidad Católica de Chile. Recuperado en mayo de 2013 en http://media.wix.com/ugd/e54253_da2b8b135e95d7b0901cf8aa97f70c1.pdf

[5] Prendes, M.; Sánchez, M.; Serrano, J. (2011) *Modelos de interacción de los adolescentes en contextos presenciales y virtuales*. EDUTEC, Revista Electrónica de Tecnología Educativa. Recuperado en mayo de 2013 en http://edutec.rediris.es/revelec2/revelec35/ISSN: 1135 9250.

[6] Gregorio C. y Ornelas L. (Comp.) (2011) *Protección de datos personales en las redes sociales digitales: en particular de niños y adolescentes*. México. Recuperado en mayo del 2013 http://www.libros.metabiblioteca.org/bitstream/001/307/9/978-968-5954-59-4.pdf

[7] Hernández Sampieri, R. ; Fernández Collado, C. ; Baptista Lucio, P. (2006) *Metodología de la Investigación*, México: Mc.Graw Hill.

Los Derechos Personalísimos en las Redes Sociales Rastros Digitales y Captura de Evidencia

María del Carmen Becerra, Pedro Daniel Zárate
Instituto de Informática
Universidad Nacional de San Juan
San Juan, Argentina
marisabecerra2005@yahoo.com.ar, pzarate@iinfo.unsj.edu.ar

Abstract

Como surge de la investigación, los datos de las personas se acumulan a cada instante en las computadoras personales, en los teléfonos celulares y en cuando dispositivo electrónico lo permita. La información, las fotografías, los videos, las filmaciones transitan a cada instante por las redes sociales, provocando innumerables conflictos que atentan la protección de la vida privada en Internet y específicamente en las Redes Sociales. Todo ello ha puesto en jaque los derechos personalísimos, pero pocos los casos que han llegado a la justicia son de difícil solución, debido a deficiencias en la producción de la prueba que no toma en cuenta el torrente de información personal que se deja al interactuar con soportes electrónicos o al navegar por internet. Al participar en redes sociales se dejan verdaderas huellas tecnológicas que tienen validez, como medio de prueba en la justicia y debe velarse por su preservación dado que permiten resolver situaciones cotidianas y dar soporte a las decisiones de los jueces., Dado que la informática forense es una herramienta imprescindible para formar su convicción en los fallos, en este trabajo, se presenta el resultado de un procedimiento para capturar la evidencia digital contenida en el disco duro de una computadora. Se obtuvo una imágen de la partición de datos y se analizaron exhaustivamente sus archivos, mediante el uso de herramientas forenses de licencia libre.

Palabras Claves:
Las Redes sociales e Internet. Derechos personalísimos. La identidad digital y la reputación en línea. Los rastros digitales. Procedimiento para capturar evidencia digital.

1. Introducción

Inicialmente se partió del estado del arte de Internet y se analizó el crecimiento contínuo de los servicios de redes sociales que han experimentado gran auge entre el público. Las redes sociales (Facebook, Twitter, LinkedIn, Pinterest, Instagran y otras) permiten a los usuarios generar perfiles y articular conexiones y vínculos con distintas personas. Como se muestra en la fig. 1, es mayor el crecimiento de las redes donde los usuarios pueden crear y modificar sus fotos y videos[1].

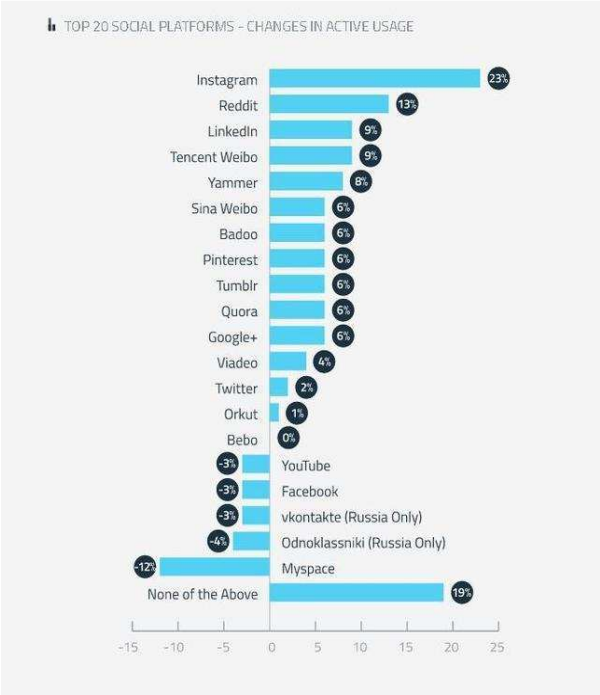


Fig.1.Crecimiento de las Redes Sociales

[1] http://www.abc.es/tecnologia/redes/20140121/abci-instagram-crece-rapido-facebook-201401211726.html

Las investigaciones de las consultoras sobre los hábitos online son consecuentes en arrojar los primeros lugares a los argentinos en las encuestas sobre los usuarios de Internet que más utilizan las redes [1]. Actualmente los Usuarios de Internet en general son adeptos al uso estas redes y resulta que ser un usuario de Internet es casi un sinónimo de participar en una red social.

Se nota entre otras cosas, que estos servicios ofrecen medios de interacción basados en perfiles personales que generan sus propios usuarios registrados, lo que propicia un nivel sin precedentes de divulgación de información de carácter personal de los interesados (y de terceros).

Muchos han nacido en Internet hace pocos años, sin embargo existe una generación de Internet que están tratando de definir múltiples disciplinas, el uso de las redes sociales se está extendiendo con rapidez, sin embargo, el dominio sobre todos los aspectos que implican su uso no es generalizado.

Además las redes sociales generalistas o de ocio cuentan con un nivel de riesgo superior al de las redes sociales profesionales, dado que los usuarios exponen no sólo sus datos de contacto o información profesional (experiencia laboral), sino muestran de manera pública las vivencias, gustos, ideología y experiencias, lo que conlleva que el número de datos de carácter personal puestos a disposición del público es mayor que en las redes sociales de tipo profesional.

Así el tratamiento de datos especialmente protegidos, supone un mayor nivel de riesgo para la protección de los datos personales y, por ende, del ámbito de la privacidad e intimidad de los usuarios. Entre las principales situaciones que se detectan, cabe señalar las siguientes:

• Existe un problema derivado de la falta de toma de conciencia real por parte de los usuarios de que sus datos personales serán accesibles por cualquier persona y del valor que éstos pueden llegar a alcanzar en el mercado. En muchos casos, los usuarios hacen públicos datos y características personales que en ningún caso expondrían en la vida cotidiana como ideología, orientación sexual y religiosa, etc.

• Los datos personales pueden ser utilizados por terceros usuarios malintencionados de forma ilícita.

• Existe la posibilidad de que se trate y publique en la Red información falsa o sin autorización del usuario, generando situaciones jurídicas que pueden llegar a derivarse de este hecho, por ejemplo la falsa imputación de un delito que dé lugar a una acción penal pública. En la ley argentina, estas conductas están tipificadas en el Código Penal en su capítulo referido a los “Delitos contra el Honor”, que tutela el derecho al honor, al decoro y la reputación de las personas, es decir su personalidad moral. En este orden se puede afirmar con la doctrina, que los usuarios que realicen actividades difamatorias mediante la utilización de redes sociales online, pueden ser pasibles de este tipo de delitos[2].

• El hecho de que, a través de las condiciones de registro aceptadas por los usuarios, éstos cedan derechos plenos e ilimitados sobre todos aquellos contenidos propios que alojen en la plataforma, de manera que pueden ser explotados económicamente por parte de la red social. Es importante tener en cuenta que en muchas ocasiones, las redes sociales permiten a los motores de búsqueda de Internet indexar en sus búsquedas los perfiles de los usuarios, junto con información de contacto y de perfiles amigos, lo que puede suponer otro riesgo para la protección de la privacidad, además de dificultar el proceso de eliminación de esa información en Internet, dejando sin posibilidad la concreción del derecho al olvido.

Del muestreo de relaciones surgió que aunque los servicios de redes sociales aportan un amplio abanico de oportunidades de comunicación, como es el intercambio en tiempo real de todo tipo de información, la utilización de estos servicios puede plantear riesgos para la privacidad de sus usuarios (y de terceras personas).

Los datos personales relativos a las personas son accesibles de forma pública y global, de una manera y en una cantidad sin precedentes, sin perjuicio de las distintas actividades delictivas que se pueden llevar a cabo usando estas redes como medios para la comisión de ilícitos (Delitos Informáticos, Delitos contra el Honor, contra la Propiedad Intelectual, Industrial, etc.) [2].

En síntesis, el abanico de posibilidades de infracción a los derechos de intimidad y privacidad en las redes sociales es muy amplio, ya sean estos ilícitos cometidos por otros usuarios de las redes o por terceros [3].

Si bien los Derechos personalísimos o de la personalidad y las leyes que los protegen nos insertan en la doctrina y la jurisprudencia del Siglo XIX y son consagrados en normas de carácter internacional como La Declaración de Derechos Humanos, el Pacto de San José de Costa Rica, que se tradujeron en tratados y Pactos Internacionales, entre los diferentes límites fijados en los alcances de la investigación, se debe destacar que la misma recoge solamente la normativa vigente en el país respecto de los derechos personalísimos implicados en el Derecho a la Intimidad, Derecho a la Imágen y al Honor [4].

Se considero necesario destacar el concepto de identidad digital, como toda la actividad que generamos en Internet. Su construcción está ligada a nuestras habilidades tecnológicas, así como a una actitud activa,

[2] www.gonzalezfrea.com.ar/derecho-informatico/...redes.../265/

participativa, abierta y colaborativa en la Red. Las redes sociales permiten disponer de diferentes identidades digitales, y todas ellas quedan fijadas en el ciberespacio, resultando complicado eliminar su rastro. Recientemente la comisión Europa ha puesto en marcha un proceso normativa para regular el derecho al olvido digital.



Fig.2 Ejemplo de Identidad Digital

Como se muestra en la fig. 2, A partir de Gamero (2009)[3] se presentan una serie de herramientas gratuitas y accesibles en la red, a través de las cuales cualquier persona puede vertebrar una identidad digital. Los blogs que en la navegación se referían al diario de un barco. Extrapolado a nuestros días, un blog ha pasado a ser un diario que tanto puede ser personal como corporativo. Los microblogs son una herramienta similar al blog, con la diferencia que tienen un número limitado de caracteres y que se pueden publicar a través de diversas aplicaciones, como vía SMS o por correo electrónico. Los usuarios de los microblogs, aparte de escribir micromensajes, pueden seguir a otras personas y conocer lo que hacen y comentan. Los portales de noticias y sitios web pueden ser una plataforma a partir de la cual construir una identidad digital. Los sitios de redes sociales genéricas o especializadas, tales como Facebook, LinkedIn, XING o Pleiteando (esta última especializada en el mundo jurídico).

En relación a la gestión de la propia identidad, muchos sitios de redes sociales permiten que no se muestren algunos de los datos del perfil[4]. Del mismo modo, la red Facebook permite restringir el acceso a ciertas informaciones, usuarios concretos o a grupos de usuarios mediante la configuración de cada perfil, además también permite quitar una etiqueta de un tuit específico, a su vez Twitter permite deshabilitar la función de etiquetar desde las opciones de configuración[5].

El conjunto de todos estos pasos en Internet forma parte de la identidad digital de una persona, de quien posteriormente se pueden buscar y recuperar gran parte de las acciones, comentarios y opiniones que ha dejado en la red. En la identidad digital convergen muchos aspectos de carácter sociológico, cultural e incluso psicológico.

Los datos expuestos son los referidos a información personal como números de identificación y seguridad, credenciales, cuentas bancarias, patrones de compra, direcciones y otros altamente redituales al venderse las bases de datos. El robo de identidad digital también es una realidad en nuestro país, no obstante todavía no existir una tipificación del delito como tal, hay en Argentina una cruzada para tipificar el delito de suplantación de identidad digital y como lo expresa la doctrina es uno de los delitos de mayor crecimiento [5].

Como producto de la presente investigación, se ha detectado un vasto mar de información digital se graba a través de una gruesa red de sensores, desde teléfonos hasta dispositivos GPS (Sistema de posicionamiento global) o tarjetas de identificación electrónicas, que partan nuestros movimientos e interacciones. Junto con la información ya obtenida de fuentes como de navegación en la Web y tarjetas de crédito, estos datos forman la base para un nuevo campo llamado inteligencia colectiva[6].

Los textos, fotografías o vídeos en la red, con Google Docs, Picasa, Flickr, YouTube o Vimeo todas las actividades en la red (visitas a la web, clics en un enlace, comentarios en un blog, colgar una foto o un vídeo...) quedan registradas y difícilmente se borran7.

Los expertos en seguridad informática aseguran que los teléfonos móviles de última generación son nuestro principal semillero de rastros digitales, así como el mayor boquete de seguridad informática con el convivimos. Los celulares pueden proporcionar una serie de datos: Llamadas realizadas (fecha, hora, duración), Llamadas recibidas (fecha, hora, duración), Ultimo número marcado (LDN-Last Daialed Number),

[3] Gamero, Ruth (2009). "La construcción de la identidad digital". *EnterIE*, 131 (junio 2009). <http://www.enter.ie.edu/enter/mybox/cms/11569.pdf>. [Consulta: 02/10/2014].

[4] http://bid.ub.edu/24/giones2.htm

[5] https://twitter.com/settings/security

[6] http://www.lanacion.com.ar/1076576-dejando-un-rastro-digital-que-sucede-con-nuestra-privacidad

[7] http://www.elperiodico.com/es/noticias/sociedad/huella-imborrable-1392784

Lista de contactos, Mensajes de texto, fotografías, Archivos, Archivos borrados, Espacio desperdiciado, Videos, Agenda, Correo electrónico, Tonos de timbres, Ubicación, establece la ubicación física de una persona o su dirección de traslado o viaje, Tarjeta SIM (Subscriber Identity Modelue).

A su vez las tarjetas SIM contienen un procesador de memoria no volátil, se utiliza como dispositivo de almacenamiento de información relacionada con el suscriptor, incorporada a la red global de celulares GSM (Global Systems for Mobile communications) y pueden contener: a) Identificador de área local, identifica donde está ubicado el celular. Este valor permanece aunque este apagado el celular, b) Número de serie se puede obtener sin tener el PIN (Personal Identification Number), c) número de Cliente, que permitirá mediante la ayuda del proveedor de servicio identificar al propietario del celular) Número de teléfono celular [6].

Como lo señala el autor Gustavo Daniel Presman, los integrantes de una red social intercambian información, pudiendo estar en línea o desconectados, utilizando diferentes dispositivos de conexión como computadores, celulares, encontrándose en el centro de todas esas comunicaciones los servidores de redes sociales [7].

a) El historial del chat del Black Berry Messenger de Facebook o WhatsApp, nos muestran que las comunicaciones con la lista de contactos revelan la portabilidad extrema a la que han llevado servicios Facebook, Twister y Linkedin.

b) Los comentarios (post) subidos por miembros de la red social que pertenecen a los contactos del usuario comentado, que pueden contener archivos adjuntos de videos, audio, imágenes (con metadatos propios).

c) El correo electrónico del mismo modo que no se borra el rastro en la red, en general, tampoco se borran los mensajes de correo electrónico a pesar de que estén protegidos con una contraseña. Hay proveedores de correo electrónico que personalizan la publicidad que aparece en la pantalla a partir del contenido de los mensajes electrónicos. Son una prueba de fácil acceso mediante un pedido de informe judicial al proveedor de servicio de Internet o un peritaje informático [6].

d) Él envió de notificaciones al correo electrónico cada vez que son mencionados en una red social.

e) La búsqueda de contactos y agregarlos a la lista actual.

f) Publicación de eventos.

g) Los “tweets”, esto es, mensajes de hasta ciento cuarenta caracteres que incluyen hasta 10 fotos.

h) Un “me gusta” en una foto de una mujer o de un hombre es otra cosa. No tiene efectos jurídicos, pero suma en el conflicto emocional, además si en el muro del imputado se colocan frecuentemente esos “me gusta” como una conducta de flirteó u necesidad de agradar, máxime cuando se mantienen abiertos los perfiles de Facebook, favoreciendo el acceso público vía buscador[8].

i) Las fotos publicadas en Facebook también son utilizadas como indicios. Ni hablar si alguno de los presentes filmo con su celular hechos ocurridos en una reunión o etiqueto fotografías de los presentes, la etiqueta puede afectar derechos personalísimos.

j) No hay que olvidarse de los mensajes de texto “Un ordenar lo cuenta todo. Solo necesitamos hacer una copia del disco duro y crear una línea de tiempo de su uso para saber quién ha hecho que en cada momento, incluso si ha habido intentos de borrar las huellas”, explica Daniel Creus.

No se ha juzgado aun en Argentina la eventual responsabilidad civil y penal de una red social por contenidos producidos por terceros, sin embargo, son innegables las consecuencias civiles y penales que impone el control sobre el respecto a la persona humana que se plasman en derechos como pueden ser entre otros los relativos al Derecho a la Intimidad, a la Imágen y a la Identidad y los reclamos que han llegado a la justicia son de difícil solución debido a la dificultad que presenta el manejo de la prueba por lo que en este trabajo se presenta un procedimiento para la recuperación de datos

2. Elementos del trabajo y metodología

En un primer momento de la investigación se destino a la revisión y relevamiento de antecedentes, bibliografía y referencias de estudios e investigación en la Red, vinculadas a las redes sociales y especialmente a Facebook.

Inicialmente se utilizo el método de observación para recoger datos de las redes sociales y se analizo la incorporación de los datos de las personas que conforme el derecho se consideran especialmente protegidos.

Si bien el contenido de las redes sociales puede encontrarse o bien en computadores personales o en los dispositivos móviles de los usuarios participantes, existe mucho material generado o descargado de estos nodos y mayoritariamente almacenado en los servidores a aplicación de la red social y en algún lugar de la nube.

A los efectos de limitar el alcance de la investigación se tomo como objeto de estudio los datos personales del perfil público de alumnos, contenidos en Facebook. Esta información fue obtenida a partir de los discos rígidos de distintas pc´s que forman parte de un gabinete de capacitación con el que cuenta el Instituto

[8] www.tendencias21.net

de Informática, por el cual continuamente pasan diferentes usuarios (alumnos) utilizando software de variadas aplicaciones, se seleccionaron un par de pc´s de dicho gabinete y se procedió a la recolección de datos de los alumnos, accesibles en forma pública, global y de naturaleza pública expuestos en la web, en fin, todo lo que no requiera violar una clave para obtener los datos, tarea ésta que se desarrolló en el Laboratorio de Informática Forense del Instituto de Informática de la Facultad de Ciencias Exactas, Físicas y Naturales de la UNSJ.

Cabe señalar que al momento de identificar y registrar la evidencia se consideraron las guías generales existentes en nuestro país, con el fin de identificar los rasgos y características esenciales sin ánimo de realizar juicios que invaliden el procedimiento. Estas guías se basan en el método científico o para concluir o deducir algo acerca de la información. Plantean una serie de etapas para recuperar la mayor cantidad de fuentes digitales con el fin de asistir en la reconstrucción posterior de eventos.

El objeto del caso de estudio se planteo fundamentalmente, teniendo en cuenta el protocolo de actuación de los peritos informáticos en la justicia[9] y que están autorizados por la legislación vigente para realizar este tipo de análisis, las tareas periciales se realizaron (Teniendo en cuenta de obligatoriedad de informar a las partes sobre el lugar y el momento de realización de tales actos para asegurar la posibilidad de presenciarles, y por ende, el legitimo derecho de defensa y el debido proceso constitucionalmente protegido) [7]. Luego se Labro un acta ante la cátedra.

En cuanto a la metodología, utilizada para la recolección de evidencias en redes sociales se ha trabajado con tres métodos: de recolección física, de recolección lógica y de recolección a partir de archivos de resguardo.

Con el primer método se accedió a la memoria del computador y se obtuvo una imágen completa de la memoria, en ella se guardo no solo el sistema de archivos sino también la información de usuario. El segundo se refiere a la copia del sistema de archivos activo desde un computador a un dispositivo de esta forma los datos almacenados se recuperan se pueden analizar posteriormente. Mediante el tercer método los datos a analizar han sido almacenados previamente a través de una copia de resguardo desde el dispositivo a la computadora del usuario. El resguardo de datos incluye, fotografías, videos y filmaciones.

[9] ww.jusneuquen.gov.ar/.../ProtocoloActuacionPericiasInformaticas.pdf

3. Resultados.

En este trabajo se muestra como, una vez obtenida la imágen (en este caso los datos personales de los alumnos), se procedió a trabajar sobre ella, llevando a cabo una exploración exhaustiva sobre la imágen forense obtenida con el fin de extraer la información relacionada con el planteo del problema. Se identifica una imágen obtenida de un usuario de Facebook, para lo cual se utiliza de una herramienta forense de libre distribución, capaz de identificar y decodificar los archivo jpg.

En este caso se utilizó **CAINE** (Computer Aided Investigative Environment), sobre una distribución de Linux basado en Ubuntu 10.04 CAINE cuenta con una gran selección de software, una interfaz gráfica amigable y un soporte receptivo [8]. Además CAINE trae un Mount Manager, que nos permite montar como de solo lectura el disco al que debemos efectuar la imágen forense, con el objeto solo de examinarlo. Estas herramientas poseen licencia de uso libre GNU/GPL, pudiéndose descargar del sitio sin inconvenientes [9].

Para comenzar la tarea en la estación de trabajo de Informática forense, se abrió el menú de análisis, se examino la imágen del dispositivo para verificar el contenido, los archivos borrados y fragmentos de archivos, los metadatos, etc., en particular la información de navegación satelital. La técnica de análisis de las imágenes implica recuperar archivos sin importar en que estructura de archivos se encuentren. Para ello se utilizo una herramienta que efectúa la extracción de fragmentos de datos a partir de un gran conjunto de archivos, en partículas de aquellas áreas del sistema de archivo que contiene espacio no asignado. El análisis se efectuó utilizando los valores de encabezado (inicio del archivo) y pie (fin de archivo) del tipo de archivo conocido.

Cada archivo posee una firma o estructura de encabezado que lo reconoce como un determinado tipo de archivo, por ejemplo: jpg, doc, tga. De esta forma se pueden recuperar archivos que no tienen una estructura de metadatos que apunten hacia ellos, ya sea porque las entradas a su sistema de archivo no se encuentran corruptas. La búsqueda se realizo bloque por bloque de datos desperdiciados que coincidan con el encabezado y pie de determinado bloque de archivos. Mientras los datos no sean sobrescritos o borrados de forma segura los datos borrados con cualquier tipo de dispositivo de almacenamiento pueden recuperarse con esta técnica.

Lo primero que se hizo fue clickear la opción **“File Type”**. Seleccionando luego **“Sort Files by types”** en la izquierda de la pantalla. Luego se hace click en Ok. Una vez terminado el análisis de los tipos de archivo, se nos mostrara un resúmen con la información que se

encuentra dentro de la imágen del disco, como audio, video, imágenes, aplicaciones, etc.
Copiamos la dirección que nos ofrece el análisis, y la pegamos en una nueva pestaña del navegador. Luego de hacer esto, se mostrara el mismo resumen de tipos de archivos pero más interactivo. En nuestro caso, como nuestra prioridad son las imágenes, hacemos click en **"images.html".**
Se abrirá un reporte con todas las imágenes que se encuentran en el disco, sin importar su extensión o si fueron eliminadas. Además muestra el tipo de la imágen, la versión, el tamaño y la ruta.
Los archivos de imágenes se pueden adquirir a partir de la recuperación de las imágenes borradas, efectuando la recolección de fragmentos de los archivos eliminados que aún se encuentran en la estructura del sistema de archivo. En la recuperación se pueden obtener imágenes con pérdida de fragmentos o con la falta de encabezado del archivo.
La búsqueda de fragmentos de archivos se realiza a partir del encabezado conocido del archivo hasta el fin o pie del archivo. La recuperación a partir de los fragmentos puede recomponer la imágen con fragmentos que no son secuenciales. En el análisis de la imágen, es importante determinar si ha sido modificada o es el original. Las imágenes pueden ser capturadas desde diferentes dispositivos como celulares, computadoras, escáner, o creadas directamente por computadora a través de un programa específico.

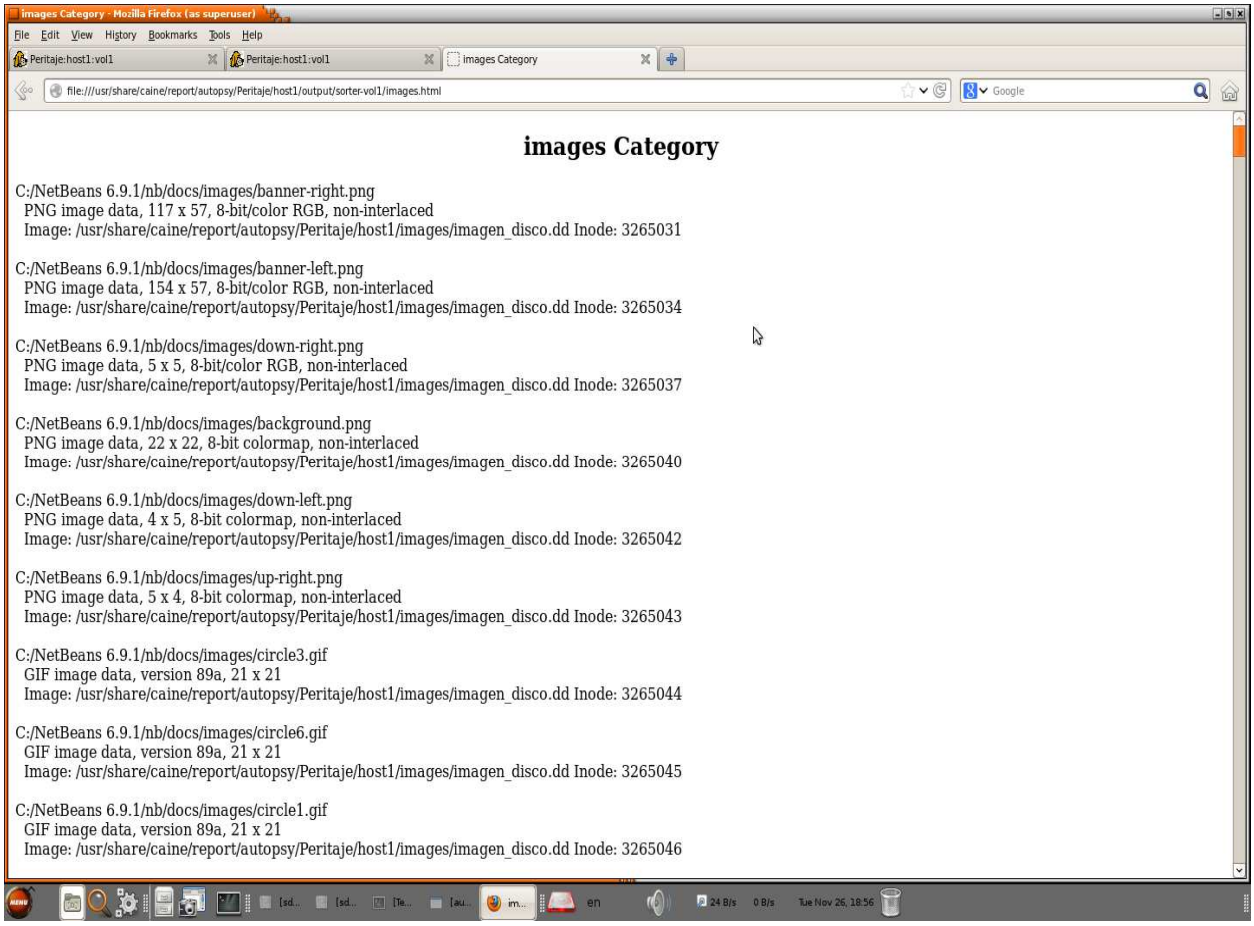


Fig.3 Muestra de categoria de imágenes

Se observa del reporte que la extensión de los archivos coincide con el tipo de imágen del mismo, por lo que se deduce que no hubo renombramiento de archivos. También comprobamos que existe una carpeta que podría contener imágenes privadas, por lo que concluimos que el reporte nos servirá como evidencia.
Al comprobar la existencia de una carpeta que contiene imágenes, determinamos hacer un análisis profundo sobre el disco.

Volvemos a la pestaña principal y hacemos click en **“File Analysis”**, para comprobar los archivos que se encuentran en la imágen del disco. Una vez que se termina el análisis, se puede apreciar que los archivos de color azul son los que se encuentran en el disco y los de color rojo los que han sido eliminados. **Autopsy** trabajará con todos ellos, esta es una herramienta forense incluida en el paquete de software de análisis forense CAINE. Continuando con el análisis debido a que el estudio de los datos debe ser profundo, podemos buscar archivos por carpetas. Teniendo en cuenta el reporte buscaremos inicialmente en las carpetas que consideramos sospechosas.

En el reporte podremos ver información como el Path, la fecha de generación del archivo, su tamaño, etc.

Dentro del menú **“File Analysis”**, en la parte izquierda de la pantalla, debajo de **“Directory Seek”** introducimos el nombre de la carpeta a buscar. Una vez dentro de la carpeta, navegamos haciendo click en el nombre de las carpetas que se encuentran en el centro de la pantalla de color azul. Se distinguen de los archivos porque su terminación es “/”. En este caso accedemos a la carpeta “forenses”.

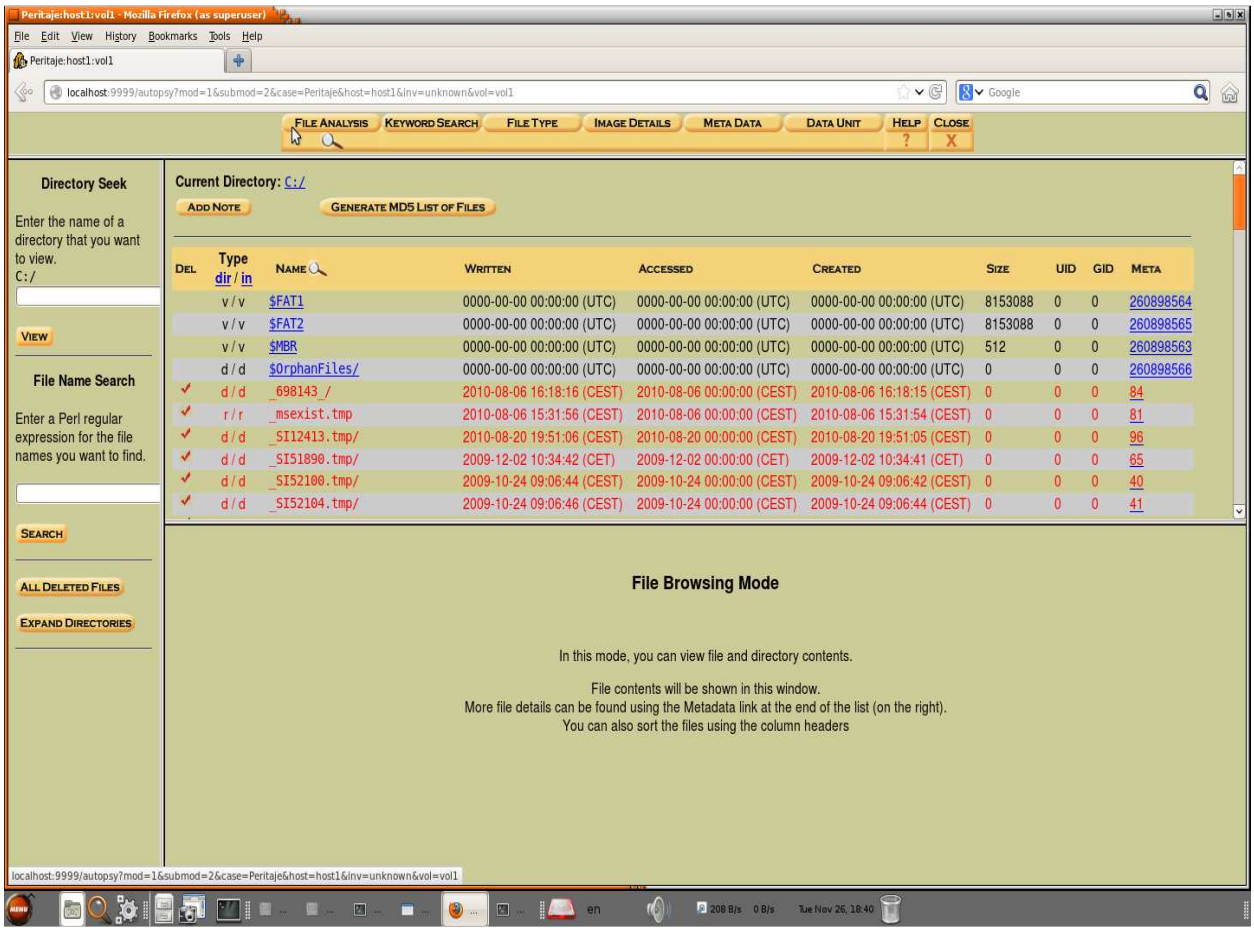


Fig.4 Muestra archivos en la imágen del disco

En el cuadro debajo de **“File Name Search”** que se encuentra a la izquierda de la pantalla, ingresamos la palabra clave a buscar en los archivos. En nuestro caso, como se trata de imágenes, escribimos cualquier extensión de las que aparezcan en el reporte, en la búsqueda que vamos a realizar escribimos **“jpg”** y hacemos click en **“search”**. Se muestra una lista con todos los archivos con extensión jpg. Un click derecho sobre la ruta de una imágen específica, se abre un menú desplegable dentro del cual hacemos click en **“Open Link in a new Tab”**. Se abrirá en una nueva pestaña del navegador la vista previa de la imágen junto a varias opciones, haciendo click en **“report”**, se genera un informe con toda la información de la imágen.

Dentro de la carpeta, encontramos archivos de formato **html**, en los cuales podemos hacer click para acceder a

su información. Encontramos varios archivos sospechosos, dentro de los cuales estaban varios archivos **html** e imágenes **jpg**. Haciendo click en **"report"**, obtenemos la información perteneciente a la imágen.
Podemos generar un reporte de la misma forma que lo hicimos con las imágenes clickeando en **"report"**. Haciendo click en **"Export",** podremos descargar el archivo para verlo en su formato original.

Una vez guardado podremos abrirlo desde el navegador. Nos encontramos que era una página de facebook guardada para acceder a la información personal de un usuario. En este caso nos dio un error por haberse abierto sin todos los archivos que forman el html como imágenes, etc.

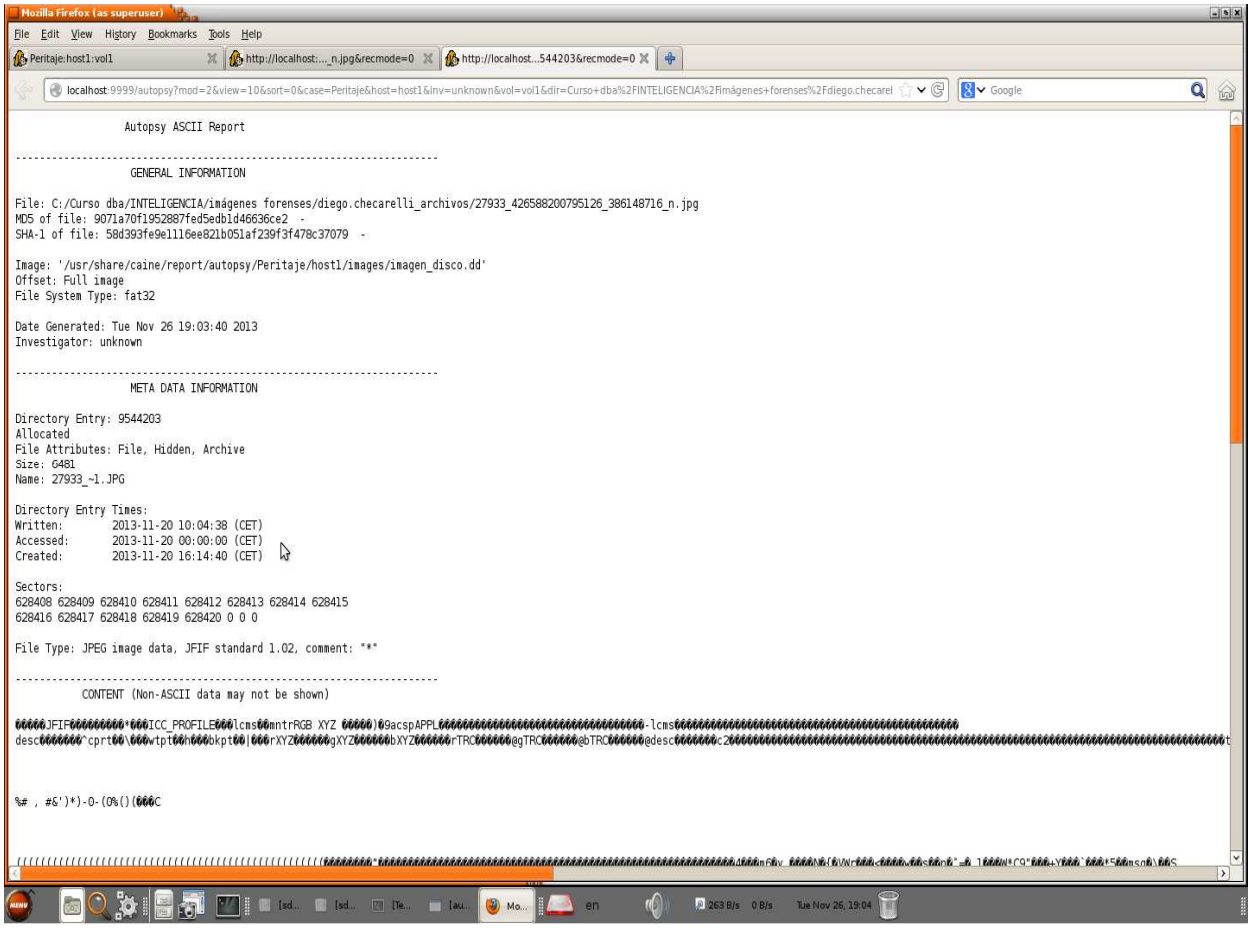


Fig.5 Informe con la informacion de las imágenes

Como aporte se muestra el resultado de un procedimiento que permite capturar la evidencia digital contenida en el disco duro de una computadora, generando una imágen de la partición de datos para analizar los archivos contenidos en la imágen mediante el uso de herramientas forenses de uso libre.

4. Discusión.

Se considera que se cumplió con los objetivos particulares de la ponencia, debido a que se realizo un trabajo para identificar las circunstancias y elementos a considerar al momento de realizar una intervención pericial en la vida real. La justicia necesita el auxilio de la informática forense para el esclarecimiento de los hechos y su persecución, los procedimientos a aplicar presentan particulares dificultades, por lo que en este trabajo se propuso el uso de herramientas forenses a los efectos de discutir con éxito la integridad de la evidencia digital obtenida.

5. Conclusión.

A lo largo de la investigación se pudo comprobar en cuanto a la relación entre derecho a la imágen y las redes sociales han surgido una serie de figuras y nuevos retos jurídicos e incluso nuevos riesgos con respecto a nuestros derechos personalísimos.

Se propuso y se aplicó una herramienta en la que se presentaron los resultados de un meticuloso análisis exploratorio sobre el futuro trabajo pericial de los alumnos de la carrera, se demostró que es posible recolectar evidencia en las redes sociales en las computadoras, celulares y otros dispositivos que hayan participado en los servidores de las redes.

Se estableció un marco referencial base para cualquier investigador forense digital, donde se tomaron en cuenta aspectos técnicos, y el marco legal necesario procurando la generación y el fortalecimiento de iniciativas multidisciplinarias para la modernización y el avance de la administración de justicia, en el contexto de la sociedad digital y de la información.

Desde nuestra cátedra en la Universidad Nacional de San Juan y para dar respuesta a la justicia ayudamos así a formar expertos que aporten certeza en la convicción de los jueces en las causas que requieran el auxilio de la tecnología para el esclarecimiento de los hechos y su persecución.

6. Referencias

[1] www.clarin.com/redes_sociales/argentinos-quintos-uso-redes-sociales

[2] Temperini, M; Borghello, C. *La captación ilegitima de datos confidenciales como delito informático*. 41 JAIIO.SID 2012.

[3] Almark, D. R. y Molina Quiroga E. *Tratado de Derecho Informático. Internet y Responsabilidad de los Proveedores de Servicios*. Tomo III. La Ley 2012.

[4] Borda, G. J., *"Las redes sociales y los derechos de la personalidad en la Internet"*, La Ley, 16/09/2010.

[5]Tomeo F. *Redes Sociales y Tecnologías 2.0. Reputación Online*. Astrea.2013.

[6] Darahuge, M.E., Arellano Gonzalez, L. E.. *Manual de Informática Forense II*. Errepar. Buenos Aires. 2012.

[7] Presman, G. *Investigación Forense en redes sociales*. XV Congreso Iberoamericano de Informática y Derecho. Bs.As.2011.

[8] http://software-libre-if.blogspot.com.ar/p/tutorial- de-linux-caine.html (última visita 01/08/2014)

[9]http://caine-live.net (Fecha de la última visita 01/08/2014)

RSE, SISTEMAS DE GESTION COMPUTARIZADOS Y CALIDAD DE VIDA

López, Carlos Alberto
Universidad Tecnológica Nacional, Facultad Regional La Plata

Abstract:
Cada vez más frecuentemente se habla de la responsabilidad social de las organizaciones, de las empresas y demás. Se ha avanzado mucho, pero en el componente atención al cliente, bajo un aparente avance por medio de la tecnología, considero que se está en un franco e indolente retroceso.
Es comprensible que temáticas involucradas como derechos humanos, medio ambiente, sustentabilidad y otras ocupen mayoritariamente los debates en foros y en las prácticas empresariales. Pero que a un cliente no se le reintegre lo que corresponde, no pueda acceder en horarios razonables a una página de internet que se sabe saturada o que aguarde infructuosamente en una línea telefónica, aunque sea un 0800, para recibir por respuesta "todos nuestros operadores están ocupados, intente más tarde", todo eso socava su calidad de vida.
Las normas que regulan el intercambio entre productores y consumidores, exhiben un vacío con relación a que los sistemas computarizados asociados a la comercialización y los servicios incluyan facilidades de resolución a conflictos manifiestos.
Recorrida la profusión de denuncias en Areas de Defensa al Consumidor, más del 85% no hubiesen existido o habrían tenido pronta y adecuada solución de haber mediado conceptos solidarios en la concepción de los sistemas computarizados.
Un marco legal adecuado que obligue a respetar los intereses de todos los actores involucrados y la puesta en operaciones de acordes sistemas de gestión, transformaciones no costosas ni que implican grandes cambios, permitirán mejorar ciertamente la calidad de vida.

Palabras Clave:
RSE, solidaridad, sistemas, defensa al consumidor, factor humano, aplicación organizacional, calidad de vida.

Introducción

Durante la Jornada "Responsabilidad Social Empresaria: Actualidad y Experiencias" en la Cámara Argentina de Comercio, siendo que el cliente es un capítulo de la RSE, consulté acerca de que acciones concretas se están emprendiendo para resolver la típica frase "se cayó el sistema", cuando existiendo soluciones prácticas, no se implantan, siendo que ello contribuiría a la calidad de vida.

Obtuve respuestas teóricas, elípticas, cargadas de buenos deseos y esperanza, pero nada en concreto; salvo una: *"Desarrollar un plan de RSE es casi una obligación, construye valor, colabora con el éxito y minimiza la probabilidad de crisis."*[1]

Crisis, palabra de uso en variados contextos, asociada a: "*Mutación importante….Momento decisivo…. Escasez, carestía. Situación dificultosa o complicada*."[2]

Y de eso se trata, de minimizar las ocurrencias crisis, de la necesidad de provocar cambios importantes en los actuales sistemas de gestión computarizados asociados a transacciones en general, propendiendo a la equidad en las aspiraciones de todos los actores.

Esa falta de atención provoca en algunos la reacción sanamente guerrera de lograr que alguien resuelva su problema y en el peor de los casos, en otros y desafortunadamente los más, la resignación a una realidad a la que prefieren restarle importancia. O acaso nunca escucharon expresiones tales como: *"Total, me descontaron sólo un mes de más", "Por esta diferencia en mi contra no voy a regresar al supermercado", "Me prometieron que no va a volver a ocurrir", "Me pidieron mil disculpas pero ya lo habían pasado a descuento en cuotas", "Hace más de un año que no me mandan el resumen de la tarjeta, entonces voy al banco, lo pido, lo pago y chau" "Dicen que no llega por culpa del correo (pero quién lo contrató?)", "Es preferible dejar pasar que*

[1] Maglier Pablo, Director de RR.II., S. Rural Argentina
[2] Diccionario de la lengua española (DRAE). 22.ª edición.

hacer el reclamo , uno se amarga porque nadie atiende y la página está colgada permanentemente".

Autopista y colectora con piquete incluido, es la sensación que se siente cuando al escuchar hablar de los postulados de la RSE se los compara con los relatos de la realidad cotidiana de los usuarios.

En el CONAIISI 2013 expuse el trabajo "Sistemas y solidaridad". En su Abstract escribí: "*La consabida frase "se cayó el sistema" continúa cobrando víctimas día a día y la padecemos todos por igual, a punto tal que ya ni los victimarios se salvan. Es tiempo de tomar conciencia y obrar en consecuencia, centrando la atención en la importancia del factor humano….Los sistemas deben brindar satisfacción a todos los actores que se vean involucrados con los mismos y esto es posible siempre que se respeten los derechos de todos en un pie de igualdad. Buenos sistemas y solidaridad es el camino a transitar.*"[3]

Convencido de la necesidad de motivar a estos cambios actitudinales en las organizaciones de toda índole, que ciertamente pueden concretarse pues aptitudes y equipamiento existen y de las mejores calidades, agrego entonces a esa línea de pensamiento y accionar esta contribución académica, centrada en llamar la atención a quienes promueven y ejercen la responsabilidad social empresarial, a quienes dictan normas que regulan las relaciones transaccionales y a quienes estudian y difunden las bondades de una adecuada calidad de vida, simplemente, a fin de lograrla.

Elementos del Trabajo y metodología

Los ejes conceptuales de la presente contribución son: 1) Definiciones, 2) Normativa, 3) Realidad actual y Propuestas de solución, aportes concretos. En todos ellos, se incluyen comentarios a fin de mantener el hilo conductor de las ideas que sostenemos.

[3] López Carlos Alberto, "Sistemas y solidaridad", anales CONAIISI 2013

1) Definiciones:

Si bien un glosario es siempre necesario a fin de evitar ambigüedades, cuando las temáticas que se abordan poseen un fuerte componente social, deviene más necesario aún porque definiciones y sentimientos no siempre se lucen isomorfas.

Pasaremos revista entonces a opiniones diversas acerca de 1.1. RSE y asociadas, 1.2. Sistemas de gestión computarizados y 1.3. Calidad de vida.

1.1. RSE y asociadas (RSO y RSC):

En Google se tiene acceso a 11.100.000 de ocurrencias para "Responsabilidad Social Empresarial", 5.940.000 para "Responsabilidad Social en las organizaciones" y 562.000 para "Responsabilidad Social Corporativa", de modo tal que bien podemos ilustrarnos. Dada la alta coincidencia existente, bastan unas pocas definiciones para entender el concepto:

"*Es la contribución … hacia la sociedad … en pos de mejorar … la calidad de vida de toda la comunidad."*[4]

"C*ultura de negocios basada en principios éticos…, respetuosa de las personas… que contribuye al… bienestar general...*"[5]

"*Buen gobierno, gestión ética…, compromisos de carácter voluntario que una empresa adquiere para gestionar su impacto social, … tratando de hacer compatibles el objetivo financiero tradicional de obtención del máximo beneficio con la generación de beneficios para el conjunto de la sociedad."*[6]

"*Ahora …, las empresas están más conscientes de que cada decisión debe considerar las implicaciones sociales … el impacto sobre todos los públicos interesados de la empresa."*[7]

Se caracteriza por una práctica empresarial responsable, iniciativas voluntarias que van más allá de lo exigido por la legislación y por las obligaciones contractuales adquiridas, actuaciones beneficiosas para los diversos grupos de interés y obtención

[4] EOI- Escuela de Organización Industrial - España
[5] CentraRSE - Centro para la Acción de la RSE Guatemala
[6] Xunta de Galicia, Consellería de trabajo y bienestar
[7] CentraRSE – Opt. Cit.

simultánea de beneficios, debiendo las empresas estar más conscientes de que cada decisión debe considerar las implicancias sociales y el impacto sobre todos los públicos interesados de la empresa.

"En síntesis, que la RSE no sea una moda pasajera, sino un nuevo modo de entender el papel de la empresa en la sociedad."[8]

A efectos de no sobreabundar la extensión del presente, sugiero consultar conceptos clave como Gestión Ética y Socialmente Responsable, Política de Gestión Ética y Responsabilidad Social en www.oreti-ca.org; misma recomendación sobre Satisfacción de expectativas y necesidades y Sensibilización en http:// observatorio rsc.org /inicio /quienes -somos/...

1.2. SISTEMAS DE GESTIÓN COMPUTARIZADOS

Hemos detectado y preferimos creer que por abuso del lenguaje y no por ignorancia, que cuando se alude a sistemas de gestión, en realidad sólo se tienen subsistemas computarizados; entonces, fallas no soportadas por los equipos involucrados, sean computadoras, dispositivos ó líneas de transmisión y/o interrupciones de energía, provocan la total inacción de la gestión por no estar previstos seguros, simples y elementales procedimientos alternativos de solución, relegándola hasta la puesta en servicio del subsistema transaccional computarizado, ocasionándole perjuicios sólo a los clientes.

Transcribimos algunas consideraciones donde se aprecia que si bien se declaran los intereses de todos los actores involucrados, se diseña y se construye para satisfacer las necesidades de quienes financian el ciclo de vida de los sistemas informáticos; está claro entonces que no lo hacen para los clientes.

"L*os diseñadores de sistemas de información deben diseñarlos entre otras con las siguientes características: Que reflejen las necesidades burocráticas y políticas de los sistemas con características para dar cabida a intereses diversos."*[9]

"El paradigma informacional incluye una nueva división del trabajo con tres dimensiones: a) creación de valor,… b) toma de decisiones,… y c) creación de relaciones, que "atañe a la relación entre una organización determinada y su entorno, incluidas otras organizaciones."[10]

"Se plantean algunas perspectivas y conflictos a considerar en el desarrollo de un SI, entre el personal de informática y los decisores de los diferentes niveles jerárquicos; sin olvidar las necesidades de información que requieren los beneficiarios de los servicios públicos.

La mayoría de los diseños de SI son formulados desde la perspectiva empresarial,…; omitiendo información destinada a los entes externos de la organización."[11]

"Las tradicionales alternativas de desarrollo de sistemas deben sustituirse o mejorarse por otras más orientadas al usuario final."[12]

1.3. CALIDAD DE VIDA

El relacionamiento que necesariamente se da en los marcos transaccionales, cuales quiera sean estos, siempre conlleva asociado un componente emocional derivado de la forma y el fondo de la atención que recibimos. Hace a las necesidades del humano, cuyas jerarquías: fisiológicas, de seguridad, sociales, de reconocimiento o estima y de auto superación fueron enunciadas entre otros, por Maslow: *"concierne al respeto de las demás personas: la necesidad de atención, aprecio…"*[13] Esto, entre otros factores, hace a nuestra calidad de vida.

Una profusión de autores han reflexionado definiendo cinco áreas diferentes a fin de evaluar la calidad de vida: Bienestar físico, material, social**,** desarrollo y emocional.

[8] WBCSD - Consejo Empresarial Mundial para el Desarrollo Sostenible

[9] Laudon y L. (1996). Administración de los Sistemas de Información. 3ra. edición. México: Editorial Prentice Hall.

[10] Castells, Manuel. La era de la información: economía, sociedad y cultura. Madrid: Alianza, 3era. Edición 2006.

[11] Sistemas de información: ¿Adecuación a los cambios tecnológicos o herramienta de gestión?, Arellano et all. Rev. de Ciencias Sociales v.14 n.3 Maracaibo dic. 2008

[12] García Bravo, Dl et all., "Cambios en la cultura y estructura de las empresas frente a las TI. Una aproximación empírica", Univ. de Alicante, 2000.

[13] Maslow Abraham, A theory of human motivation, 1943

"Cuando hablamos de calidad de vida, ..., estamos haciendo referencia a todos aquellos elementos que hacen que esa vida sea digna, cómoda, agradable y satisfactoria. En el caso de los seres humanos, los elementos que contribuyen a contar con una calidad de vida pueden ser tanto emotivos, como materiales como culturales."[14] *"... es un estado de satisfacción general, derivado de la realización de las potencialidades de la persona. Posee aspectos subjetivos y objetivos. Es una sensación subjetiva de bienestar físico, sicológico y social. Incluye como aspectos subjetivos la intimidad, la expresión emocional, la seguridad percibida, la productividad personal y la salud recibida. Como aspectos objetivos, el bienestar material, las relaciones armónicas con el ambiente físico y social y con la comunidad y la salud objetivamente percibida."*[15] *"... es una medida compuesta de bienestar físico, mental y social, tal como la percibe cada individuo y cada grupo, y de felicidad, satisfacción y recompensa."*[16] *"...es la evaluación subjetiva del carácter bueno o satisfactorio de la vida como un todo."*[17] *"... es la sensación subjetiva de bienestar del individuo."*[18] *"Es el indicador multidimensional del bienestar material y espiritual del hombre en un marco social y cultural determinado."*[19]

Cerrando esta síntesis, presentamos tres definiciones:

De la Organización Mundial de la Salud, por su carácter abarcativo y universal: *"... es la percepción de los individuos ..., en un contexto cultural y en un sistema de valores respecto a sus metas, expectativas, estándares e intereses. ...concepto amplio influenciado por ... el estado psicológico, las relaciones sociales...."*[20]

De la Lic. Choren, por involucrar lo que deseamos acerca del estilo de desarrollo de un país relacionado con el bienestar integral de sus habitantes: *"... es el objetivo al que debería tender el estilo de desarrollo de un país que se preocupe por el ser humano integral. Este concepto alude al bienestar en todas las facetas del hombre."*[21]

Del Prof. Mihály Csíkszentmihályi, porque reflejan nuestra visión acerca de cómo encarar los desafíos y afrontar las adversidades que, con la debida licencia por la expresión coloquial, suelo resumir así "*fui forward, acostumbrado a cobrar y a seguir para adelante*", actitudes que marcaron mi estilo desde mis años jóvenes.

"De todas las virtudes que podemos aprender no hay otra característica más útil, más necesaria para la supervivencia y con más probabilidades de mejorar la calidad de vida que la capacidad de transformar la adversidad en un desafío que pueda proporcionarnos disfrute."[22]

Queda claro entonces que los factores emocionales condicionan ciertamente la calidad de vida de los miembros de una sociedad, llegado aún a producir deterioros en la salud. Por tanto, debiera ser interés de las organizaciones, cualquiera sea su naturaleza, tratar de minimizar las áreas de conflicto con sus clientes y responsabilidad del Estado, que esto se halle garantizado en los marcos legales regulatorios.

2. Normativa:

Incursionamos ahora en los aspectos normativos atinentes a la RSE dentro y fuera del país, al intercambio comercial entre productores y consumidores y la estructura de la defensa al consumidor en Argentina.

2.1. LEGISLACIÓN ARGENTINA SOBRE RSE

[14] http://www.definicionabc.com/social/calidad-de-vida.php

[15] Ardila, R. Calidad de vida: una definición integradora. Rev. Lat. de Psic., 2003, vol 35, nro.002. Bogotá, Col.

[16] Levy et all (1980). La tensión psicosocial. Población, ambiente y calidad de vida. México: Manual Moderno.

[17] Szalai & Andrews, The quality of life. Comparative studies.1980, London, Sage.

[18] Chaturvedi, S (1991). What is important for quality of life to Indians in rel. to cancer? Sl Sc. and Med. ,vol. 33.

[19] Quintero, G. (1992). Comunicación a J. Grau (1996).

[20] Wold Health Organization (1997). WHOQOL: Measuring quality of life. Geneve, Switzerland. WHO.

[21] Choren Susana, en http://www.cricyt.edu.ar /enciclopedia/terminos/CaliVida.htm

[22] Csíkszentmihályi Mihály, Flow, the psychology of optimal experience, 2008, Kairos, Bs As, Argentina

El documento "Avances de la legislación argentina sobre RSE" producido en la Universidad Nacional de Córdoba por Blazquez y otros, tiene por objetivo "*reflejar las acciones legislativas que se han emprendido desde el estado Argentino … en materia de responsabilidad social empresaria. Teniendo en cuenta que no existe una legislación integral de carácter nacional, se toman como ejemplo algunas legislaciones e iniciativas de alcance local."*[23]

De sus conclusiones destaco y transcribo: *"Los conceptos de responsabilidad social empresaria, en sus diferentes denominaciones e instrumentos que la gestionan son un ejemplo más de un proceso social positivo en cuanto a calidad de vida…"* y *"Temas como el presente son también parte de la agenda oficial y estamos convencidos que si bien los intentos analizados aún son muy embrionarios ponen de manifiesto una tendencia que debiera generalizarse…"*[24], de donde se sigue que esta temática hace a la calidad de vida, al par que el Estado debe tener presencia eficaz.

2.2. NORMATIVA INTERNACIONAL EN MATERIA DE RSCDircom es una asociación profesional que agrupa a directivos y a profesionales de la comunicación de empresas, instituciones y consultoras en España. En http://rsc.-dircom.org/Normativa/normativa.html se halla disponible una exhaustiva recopilación internacional.

Hemos analizado la misma sin poder hallar una sola referencia a las necesidades que hoy día y cada vez más están demandando los clientes, aún cuando, según el Observatorio de Responsabilidad Social Corporativa

2.3. REGULACIONES AL INTERCAMBIO E/ PRODUCTORES Y CONSUMIDORES

Las normas que al presente regulan el intercambio son 6 Leyes (19.511, 22.802, 24.240, 25.065, 25.542, 26.361), 16 Resoluciones (134/98, 313/98, 616/98, 906/98, 387/99, 461/99, 678/99, 54/2000, 75/2002, 8/2003, 26/2003,37/2003, 53/2003, 102/ 2003, 09/2004, 54/2004), 1 Decreto (561/99) y 1 Disposición (3/2003).

A su tiempo, deberán adicionarse los resultados de la creación de un fuero especial en la Justicia para la Defensa del Consumidor.

Sin embargo, no existe una sola alusión a la obligación de que los sistemas computarizados asociados a la comercialización y los servicios resuelvan las zonas manifiestas de conflicto. Existe entonces un vacío legal y por ende procedimental que tiene solución, no es costoso ni implica grandes cambios en los sistemas de gestión de organismos y empresas.

2.4. ESTRUCTURA DE LA DEFENSA AL CONSUMIDOR EN ARGENTINA

Analizamos la estructura que defiende al consumidor cuando se ven vulnerados sus derechos. En todos los niveles gubernamentales existen organismos con distintas jerarquías: Secretarías, Subsecretarías, Direcciones Generales, Direcciones, Oficinas municipales y denominaciones similares que incluyen términos como defensa, información, protección, usuario y consumidor, todas con idénticos propósitos.

A modo de ejemplo, hemos transcripto de la página de la Secretaría de Comercio dependiente del Ministerio de Economía y Finanzas Públicas su política: *"La política de Defensa del Consumidor ... es garantizar … derecho a un trato digno y equitativo sin abusos…, a que se cumplan las condiciones y garantías de los productos…".* Sus áreas de trabajo son: Sistema Nacional de Arbitraje de consumo, Lealtad Comercial, Metrología Legal y Educación y Capacitación.

Si alguno de los lectores han tenido oportunidad de pasar por la experiencia de recurrir a instancias gubernamentales, habrá podido comprobar cuán complejo resulta

[23] Cuadernos de Gestión Vol. 10. Nº 1 (Año 2010), ISSN: 1131 - 6837

[24] Blazquez et all, op. cit.

[25] http://observatoriorsc.org/inicio/quienes-somos/#sthash.UosqhXC3.dpuf

hallar una solución justa y rápida a un legítimo reclamo, cuánto tiempo invertido, además, nadie nos resarce de los gastos colaterales y otras penurias. La pregunta es saber si los sistemas transaccionales hubiesen podido evitar estas instancias. Y la respuesta es sí, si en su diseño se hubiesen considerado mandatorios en un pie de igualdad los derechos del consumidor.

3. Realidad actual, propuestas de solución, aportes concretos:

Definiciones elaboradas, abarcativas, precisas y que reflejan un estado intelectual de convergencia absoluta cualquiera sea el ámbito institucional o geográfico. Visiones, misiones y objetivos meditados e impecablemente redactados. Valga esta síntesis: *"Creemos conveniente desde nuestra perspectiva definir la responsabilidad social como el compromiso de la empresa a contribuir al desarrollo sostenible, con la participación de sus grupos de interés, a fin de mejorar la calidad de vida de la sociedad en su conjunto."*[26]

Antes de pasar al acápite Denuncias, que tal vuestra ánimo con relación a los servicios brindados (¿?) por la telefonía móvil? (permítanme una sonrisa!)

3.1. DENUNCIAS, ANALISIS Y CONCLUSIONES

Hemos analizado tan solo las primeras siete páginas de denuncias de un total de 6097 que se hallan disponibles en http://www.consumidor.gov.ar/category/reclamos-denuncias/page/, tan sólo 45 denuncias de más de 40.000!.

Allí se denuncian procederes indebidos, algunos incluso dolosos, de empresas prácticamente de toda índole, servicios médicos, telefonía móvil y fija, bancos, tarjetas de crédito, correos, supermercados, mutuales, distribuidoras eléctricas y mega tiendas; la mayoría con un sólido posicionamiento en el mercado.

Un ordenamiento alfabético no exhaustivo incluye a ADT, AGE Servicios Financieros S.A., AMTAE, AYSA, Banco Francés suc. 999 catedral / tarjeta Visa, Banco HSBC, Cablevisión, Carrefour, Claro AR, Creditia, Día Argentina S.A., Direct TV, Edenor, Edesur, Falabella, Lan, Lanpass, Mercado-libre, Movistar, Musimundo, Oca, Personal, Red Megatone, Royal Sun Alliance, Telecentro, Telecom, Telefónica de Argentina, Telefónica móviles Argentina S.A., Visa, Vital y Walmart Argentina.

Las 108 anomalías denunciadas han podido ser agrupadas en 37 breves descripciones que hemos consolidado en tres grandes componentes con relación:

a. Al perjuicio económico (60 = 56%):

Baja de ofertas sin causa, Cambio compulsivo de plan a uno más costoso/Incremento de abonos sin preaviso, Cobros indebidos, Daños y perjuicios sin resarcimiento, Disparidad de precios entre lo publicitado en góndolas y los del sistema, Imposibilidad de dar bajas, Imposibilidad de reintegrar un producto no deseado, Inclusión de servicios sin consentimiento, Incumplimientos de garantías y servicios técnicos, Inexistencia de servicios ofrecidos pero facturados, Intento de cobros indebidos, Negativa a reintegrar lo debitado/abonado de más, imputándolo a pagos futuros, No acreditación de pagos, Ofrecimiento de resarcimientos menores a lo que corresponde, Pérdida de promociones por errores en el sistema de registro, Perjuicios económicos directos por mala liquidación, dilación en reintegros y/o acreditación de montos menores a los que corresponden, Reclamos monetarios indebidos, Servicio no concretados pero facturados en cuotas, Sobrefacturación.

b. A la desconsideración para con las personas (29 = 27%):

Abandono de persona, Amenazas / Hostigamiento a los clientes / Destrato, Deficiencias múltiples en la atención al cliente, Fallas reiteradas en servicio sin solución, Falta de atención al cliente (personal, telefónica, vía web),

[26] Consejo Empresario Argentino para el Desarrollo Sostenible (CEADS) 2003. Capítulo local del World Business Council for Sustainable Development (WBCSD)

Imposibilidad de comunicación con las empresas, Mentiras al informar que ya están solucionados los desperfectos, Múltiples reclamos sin respuesta, Negativa a entregar comprobante de pago, Respuestas automáticas falaces, No hacerse cargo de inconvenientes ocasionados.

c. A las falencias en las prestaciones de servicios (19 = 17%):

Baja de servicios sin aviso, Cancelación indebida de servicios, Falencias en tarjetas de crédito, Falta de servicios sin preaviso ni resarcimiento, Falta de soluciones a los problemas denunciados, Incumplimientos de la más variada índole, Interrupción prolongada de servicios esenciales.

Del total, el 85% es posible de solucionar a través de los sistemas computarizados asociados y seguramente todas las empresas denunciadas disponen de servicios de análisis, diseño e implementación de soluciones ofimáticas.

De modo tal que no existen excusas dentro del campo de la honestidad para no cesar las anomalías detectadas/denunciadas y las más de las veces no resueltas satisfactoriamente para el cliente. Asimismo, la prelación en los porcentajes acaso sugieren algo?

3.2. PROPUESTAS DE SOLUCIÓN, APORTES CONCRETOS

A los siguientes ejemplos reales le adicionamos propuestas de solución para mostrar la sencillez de las mismas aunque "*Fácil y sencillo no es lo mismo*."[27]

Caso 1: "*El día 19, Cablevisión me ofrece gratis un paquete premium hasta fin de mes que en caso de no desear su continuidad (a $200,00 por mes) debía solicitar su baja el día 30. Llamé el 30 pero no aceptan la baja porque se receptan hasta el día 15!. Manifesté mi desagrado y me dijeron que no se podía hacer nada porque el sistema es así, al mes siguiente me vino descontado, pero sólo un mes"*. Propuesta: Otro modelo de gestión ético y equitativo tan sólo con una sencilla modificación a la misma llamada *"Si Ud. desea el servicio, el día 30 lo solicita"*

Caso 2: *"Concurrí al banco a pagar un servicio al valor del primer vencimiento pero se había caído el sistema. Al día siguiente me cobraron el monto correspondiente al segundo vencimiento, expresé que ayer no hubo sistema, igual era poco".* Propuesta: Otro modelo fácil, sencillo, algunos equipos portátiles con UPS registran el pago y cuando se repone el sistema transfieren las transacciones.

Caso 3: "Casi *nunca mandan los resúmenes de la tarjeta o llegan fuera de término, reclamé pero igual me cobraron punitorios, dicen que es culpa del correo".* Propuesta: Solución por demás simple, emplear herramientas de trazabilidad en los envíos, si el retraso efectivamente es atribuible al correo, a este le cargan los punitorios.

Y así siguiendo, pero peor si se requiere un servicio crítico de emergencia médica. En cada reiteración, el operador (que las más de las veces no es el mismo) requiere de nuevo todos los datos para *"verificar la autenticidad del cliente".* Quisiera saber si quien solicitó el diseño del sistema, quienes lo desarrollaron / operan tuviesen a su madre o a un hijo en esas circunstancias verían imprescindible tanta *"confirmación de datos"* o admitirían mansamente que le mientan durante horas informándole que la ambulancia "*está en camino*". Referencias concretas, OSDE y VITAL, aunque tal vez no sean las únicas empresas con estos procederes. Propuesta: Simple, se declara una alerta en el sistema y se le asigna prioridad de atención.

Valgan estos ejemplos para mostrar el camino a seguir.

Resultados

Al proponer y desarrollar doctrina, se debe perseverar y tener templanza. Insistir en cada ocasión posible, recopilando nuevos casos, difundiendo, persuadiendo y explicando en la cátedra, en nuestros lugares de trabajo, con amigos, a quienes demandan asesoría y acompañamiento para el cambio de paradigmas organizacionales, en las colas en supermercados, bancos y

[27] Clausewitz, Carl von, "Vom Kriege",Dümler, Berlín, 1991 (original, 1832)

oficinas públicas. No desesperar ni desanimarse si los cambios deseados no se producen en cantidad y con la velocidad deseadas, saber aguardar y no olvidar expresiones tales como *"Quien pierde la paciencia pierde la guerra", "Para encontrar un diamante hay que trabajar muy duramente y remover cientos de toneladas de tierra y piedras."*[28]

Al presente se ha logrado introducir el concepto de sistemas solidarios en los trabajos de la cátedra de Habilitación Profesional en la Facultad Regional La Plata - UTN. Del mismo modo han reaccionado las empresas y organizaciones con las que hemos tenemos vinculaciones.

Nos hallamos incursionando en la temática de Responsabilidad Social Empresarial teniendo previsto, además de participar en las actividades de divulgación de la misma, mantener reuniones con empresarios, funcionarios y legisladores.

Asimismo nos interiorizaremos acerca de la creación de un fuero especial en la Justicia en Defensa del Consumidor para que los usuarios y las empresas diriman sus diferencias. Incluirá la creación de la instancia de mediación obligatoria y un nuevo observatorio de precios dentro de la órbita de la Secretaría de Comercio. Para eso se reformarán el Decreto Ley N° 22056 - Ley de abastecimiento-, la Ley N° 22802 - Lealtad Comercial- y la de Ley N° 26362 - Defensa del Consumidor-.

Discusión

Al presente no se observa una decidida campaña en la dirección de poner en valor los derechos de los clientes en un pie de igualdad con el resto de los actores involucrados en el escenario de las aplicaciones informáticas que los vinculan.

Los marcos regulatorios centran su objetivo en las prestaciones de servicios, la comercialización de productos y las transacciones comerciales asociadas, pero en ninguno se hace alusión expresa a la calidad de la atención. Quienes implantan en sus organizaciones subsistemas computarizados, lo hacen a fin de solucionar casi excluyentemente sus necesidades.

La 5ta. Edición del PMBOK (Project Management Body of Knowledge) del 2013, incorpora una nueva área de conocimiento denominada "Gestión de los stakeholders del proyecto", cuyo cometido es incluir los procesos involucrados en identificar a los interesados del proyecto o stakeholders, así como la planificación, gestión y control de sus expectativas sobre el proyecto. Esto demuestra que existen incipientes iniciativas en la problemática planteada. Aguardaremos resultados. Mientras tanto, las experiencias que hemos desarrollado al presente muestran resultados concretos, altamente positivos, de sumamente bajos costos de implantación, replicables y obviamente perfectibles, que a más de hacer sentir a los clientes que son debidamente respetados, mejora las imágenes institucionales y lejos de implicar pérdidas económicas, contribuyen a la tan perseguida "fidelización del cliente". Sintetizando, redunda en beneficios tangibles e intangibles para todos.

Conclusión

Un tema central es "no hacer como que hacemos", vale decir, emplear la sinergia de la RSE para finalmente aplicarla en un solo sentido. *"… me pregunto si cuando hablamos de RSE… estamos hablando de algo distinto, de un cambio de paradigma, de una mirada fresca, no puramente económica de ver a la empresa, o es más de lo mismo."*[29]

Responsabilidad social de/en las organizaciones y calidad de vida deben ser caras de la misma moneda. Al presente son conceptos muy bien enunciados y en algunos casos adecuadamente aplicados, pero carentes de valor al igual que ambas caras separadas si orbitan independientemente. Consideramos que su unión es necesaria y posible. Necesaria, porque las personas están llegando al

[28] Gandhi Mohandas Karamchand, el "alma grande"

[29] Willi Alberto_IAE_Business_School en http://comunicarse web.com.ar ?RSE%2C_de_que_estamos_hablando

hartazgo que cubre el amplio espectro que va desde la resignación silenciosa hasta la respuesta airada que genera conflictos improductivos. Posible, al provenir de una adecuada reingeniería de los subsistemas computarizados que, incluyendo pocas modificaciones al código y por ende de costo mínimo, posicione adecuadamente al cliente con sus legítimos derechos, colocándolo en el centro de nuestra atención y dedicación para el bien de todos, a fin de cuentas, todos somos clientes. Como catalizador deberá actuar una legislación acorde que garantice solución a esta problemática, pues tornaría de cumplimiento obligatorio el componente decisión, que al presente, por ser voluntario, está faltando en la tríada necesaria para resolver problemas, conformada por decisión, conocimiento y recursos, donde estos dos últimos están más que disponibles, tanto por la excelente formación de los profesionales involucrados como por el equipamiento informático, los aplicativos y las herramientas más utilizadas para la gestión de la RSE tales como AA 1000, GRI Report, ISO 26000, SA 8000 y SGE 21. Por lo expuesto, si bien "*Fácil y sencillo no es lo mismo*"[30], el éxito total se puede lograr aplicando a todo lo expresado mi modelo de las 3P (persuadir, propender y punir). Nuestra misión: persuadir y propender.

Agradecimientos

A mi familia toda, por el afecto que permanentemente me prodigan, cada uno con su estilo. A mi esposa y compañera de toda la vida siempre a mi lado, a mis hijos y a Bautista, Delfina y Justito, que ya imaginarán quienes son.

A quienes comparten esta visión y tratan de operativizarla.

Referencias

[1] Maglier Pablo, Director de RR.II., S. Rural.
[2] Diccionario de la lengua española (DRAE). 22.ª edición.
[3] López Carlos Alberto, "Sistemas y solidaridad", analesCONAIISI 2013
[4] EOI- Escuela de Organización Industrial - España
[5] CentraRSE - Centro para la Acción de la RSE Guatemala
[6] Xunta de Galicia, Consellería de trabajo y bienestar
[7] WBCSD - Consejo Empresarial Mundial para el Desarrollo Sostenible
[8] Laudon y L. (1996). Adm. de los Sistemas de Información. 3ra. edición. México: Ed. Prentice Hall.
[9] Castells, M. La era de la información: economía, sociedad y cultura. Madrid: Alianza, 3era. Ed. 2006.
[10] Arellano et all. Sistemas de información: ¿Adecuación a los cambios tecnológicos o herramienta de gestión?, Rev. de Ciencias Sociales v.14 n.3 Maracaibo dic. 2008
[11] García Bravo, D. et all., "Cambios en la cultura y estructura de las empresas frente a las TI. Una aproximación empírica", Univ. de Alicante, 2000.
[12] Maslow A., A theory of human motivation, 1943
[13] http://www.definicionabc.com/social/calidad-de-vida.php
[14] Ardila, R. Calidad de vida: una definición integradora. Rev. Lat. de Psic., 2003, v. 35, Bogotá, Col.
[15] Levy et all. (1980). La tensión psicosocial. Población, ambiente y calidad de vida. México.
[16] Szalai & Andrews, The quality of life. Comparative studies.1980, London, Sage.
[17] Chaturvedi, S (1991). What is important for quality of life to Indians in rel. to cancer? Sl Sc. and Med. ,vol. 33.
[18] Quintero, G. (1992). Comunicación a J. Grau (1996).
[19] Wold Health Organization (1997). WHOQOL: Measuring quality of life. Geneve, Switzerland. WHO.
[20] Choren Susana, en http://www.cricyt.edu.ar /enciclopedia/terminos/CaliVida.htm
[21] Csíkszentmihályi Mihály, Flow, the psychology of optimal experience, 2008, Kairos, Bs As, Argentina
[22] Cuadernos de Gestión Vol. 10. Nº 1 (Año 2010), ISSN: 1131 - 6837
[23] Blazquez et all, op. cit.
[24] http://observatoriorsc.org/inicio/quienes-somos/#sthash.UosqhXC3.dpuf
[25] Consejo Empresario Argentino para el Desarrollo Sostenible (CEADS) 2003. Capítulo local del World Business Council for Sustainable Development (WBCSD)
[26] Clausewitz, Carl von, "Vom Kriege",Dümler, Berlín, 1991 (original, 1832)
[27] Gandhi Mohandas Karamchand, el "alma grande"
[28] Willi Alberto_IAE_Business_School en http://comu nicarse web.com.ar ?RSE%2C_de_que_estamos_hablando

[30] Clausewitz, Carl opt. cit.

Datos de Contacto:
Carlos Alberto López. Universidad Tecnológica Nacional – Facultad Regional La Plata. Calle 60 y 124 – 1900 la Plata. clopez@frlp.utn.edu.ar

Internet de las cosas y RFID: ¿políticas públicas o auto-regulación de privados? Un análisis a la luz de la participación de los gobiernos en Net mundial

Analía Aspis
Instituto de Investigaciones Jurídicas Ambrosio Di Gioja
Facultad de Derecho y Cs. Sociales
Universidad de Buenos Aires – CONICET
analia.aspis@gmail.com

Abstract

La presente ponencia tiene como fin analizar las tendencias en materia regulatoria de la tecnología de identificación por radiofrecuencia en el marco del Internet de las Cosas, a fin de indagar el rol que los gobiernos pueden ejercer en marco de dicha gobernanza y las consecuencias en cuanto a la adopción de políticas auto-regulatorias o de servicio público.

1. Introduction

Las tecnologías digitales permiten que las vidas de los consumidores y ciudadanos se encuentren expuestas, cada vez más, a un nivel de conocimiento sobre sus elecciones particulares, tanto por parte del sector público como del privado. Es innegable que para obtener información cada vez más precisa sobre sus ciudadanos y usuarios, las tecnologías de la Información y Comunicación (TICs) ofrecen un poderoso medio para conocer muchos aspectos de la vida y hábitos de quienes interactúan con ellas. En este contexto, la tecnología de identificación por radiofrecuencia (*RFID* por sus siglas en inglés) facilita una variedad de propósitos, sobre todo comerciales: desde la identificación y gestión de productos para controlar la cadena de suministro hasta la protección de la autenticidad de una marca como así también la posibilidad de brindar beneficios a largo plazo en la gestión empresarial y un aumento de eficiencia en las transacciones personalizadas con potenciales clientes y consumidores. En este contexto, la posibilidad de conexión de objetos físicos y dispositivos móviles entre sí junto a la disponibilidad de datos abiertos (*open data*) en un contexto de Internet de las Cosas (*IoT* por sus siglas en inglés) por parte de gobiernos y empresas, conlleva a indagar y analizar los aspectos regulatorios tendientes a la protección de los individuos y las posiciones de los gobiernos respecto de las decisiones nacionales en tal sentido. Es por ello que el propósito de la presente ponencia consiste en presentar un análisis sobre las propuestas de regulación del RFID en el marco de la IoT y las políticas internacionales de la gobernanza de internet, tomando en consideración el rol de los Estados en la generación de normas jurídicas que tengan en consideración el derecho de los usuarios y consumidores que se relacionen con dicha tecnología como así también las consecuencias respecto del modelo regulatorio que pudiese elegirse.

2. Tecnología RFID y el Internet de las Cosas

La tecnología RFID utiliza ondas de radio[1] para identificar de manera inalámbrica, sin ningún tipo de contacto, sin visibilidad y en forma automática, objetos o personas vinculadas con un *tag*[2] incorporado a un objeto, en su empaquetado o inclusive en una persona. El denominado "sistema RFID" se encuentra caracterizado por la interacción de sus dos elementos principales: un tag que contiene un número de identificación único y un lector que funciona como un escáner que emite una secuencia de radio para captar el número de identificación emitido por el tag. Este número será ulteriormente utilizado para el procesamiento de información en una o más base de datos y en tal sentido, el sistema se encuentra diseñado para que ningún tipo de intervención humana sea llevada a cabo. De esta manera, para que el sistema RFID funcione correctamente,

[1] Su funcionamiento consta de la utilización de una porción del espectro electromagnético mediante el cual es posible la identificación única de objetos. La atribución de bandas y la asignación de frecuencias se encuentra a cago de la dirección de la Comisión Nacional de Telecomunicaciones. Para mayor detalle consultar en www.cnc.gov.ar/ciudadanos/espectros.

[2] El vocablo tag hace referencia un microchip. La tendencia actual indica un ascenso en el remplazo de los códigos de barras por tags RFID. Para mayor información consultar http://www.epcglobalinc.org/home, última visita 9 de Julio 2013.

etiquetas y lectores deben estar sintonizados en la misma frecuencia ya que, de otra forma, estos elementos no son capaces de entenderse y la lectura de la información contenida en el tag resulta imposible, sin perjuicio de los casos de pequeñas implantaciones donde la tecnología de radiofrecuencia se emplea sólo de forma interna o para aplicaciones específicas (sistemas de RFID propietarios). No obstante, cuando se utilizan en procesos logísticos completos o en transacciones entre varias empresas a escala internacional, es preciso coordinar políticas de gestión tendientes a la adopción de una misma frecuencia. Con miras a dicho objetivo y con el fin de impulsar al máximo la estandarización de la gestión de sistemas RFID - tanto de los códigos utilizados en la identificación de artículos como de las frecuencias- surge la creación en 2003 de EPCglobal, una entidad sin fines de lucro resultado de la fusión entre *European Article Number* (EAN), organización mundial de estándares de código de barras y *Uniform Code Council* (UCC), entidad responsable de tales códigos en Estados Unidos.[3] Para fomentar esta evolución, EAN-UCC han reglamentado y administrado el Número Mundial de Artículo Comercial (GTIN), el cual brinda estándares de identificación única a diferentes productos tales como materias primas, productos terminados, insumos y hasta servicios. Por su parte EPCglobal tiene como fin continuar la tarea llevada a cabo por el consorcio EAN-UCC-GTIN para el desarrollo de la fase siguiente en materia de identificación de productos por medio del RFID y es por ello que se prevé que las empresas mantendrán sus sistemas de códigos de barras y añadir nueva infraestructura EPC. Para tales fines, EPCglobal ha S adjudicado un contrato de VeriSign para mantener el directorio raíz de servicio de nombres de objetos (ONS) que tendrá a su cargo la redirección de la información almacenada o leída desde los tags a la base de datos o *middleware.* El ONS es un servicio de red automatizado similar al servicio de nombres de dominio (DNS) que señala computadoras a sitios en la World Wide Web. Cuando un interrogador lee una etiqueta RFID, el Código Electrónico de Producto (EPC) se pasa al middleware, lo que, a su vez, va a un ONS en una red local o de Internet para encontrar la que se almacena información sobre el producto y luego el ONS señala el middleware a un servidor donde se almacena un archivo sobre ese producto. El middleware recupera el archivo (después de la autenticación correcta), y la información sobre el producto en el archivo se puede remitir a inventario o la cadena de suministro de aplicaciones de la empresa.[4]

Así, el EPC identifica los productos individuales mediante la escritura de la información útil en un lenguaje nuevo estándar llamado el marcado de *Physical Markup Language* (PML) el cual se basa en el *Extensible Markup Language* (XML). Debido a que está destinado a ser un estándar universal para la descripción de todos los objetos físicos, procesos y ambientes, el PML será amplio y abarcará todas las industrias. Además, proporcionará un método común para la descripción de los objetos físicos y será ampliamente jerárquico. Así, una lata de Coca-Cola podría ser descrita como una bebida carbonatada, que caería bajo la subcategoría "refresco" y que a su vez se encontraría en una subcategoría "alimentos".[5]

Además de la información del producto que no cambiará (tales como la composición del material), PML incluirá datos con cambios constantes (datos dinámicos)[6] y los datos que cambiarán sólo con el tiempo (datos temporales). Al hacer que toda esta información se encuentre disponible en un archivo de PML, las empresas serán capaces de utilizar la información en formas nuevas e innovadoras. Una compañía podría, por ejemplo, diseñar o acceder a conjunto desencadenado de información del precio un producto o información relacionada con el mismo a través de diferentes enfoques o bien terceros proveedores de logística podrían ofrecer contratos de nivel de servicio que indiquen que los bienes serán almacenados a una temperatura determinada.

3.1 Internet de las Cosas

Sin embargo, la tecnología RFID no se limita en sí misma, sino que puede combinarse con otras como ser sensores, actuadores[7] y redes 4-8G. El nuevo diseño de red se denomina Internet de las Cosas (IoT) y puede definirse como una red donde todo lo que la integra está conectado entre sí: entre personas, entre objeto (cosas) y el entorno donde físicamente se encuentran. En consecuencia, es un paso más allá de Internet tradicional debido al pasaje de la interconexión virtual a la interconexión física, aunque intangible y dada su relación de especia a género, lo denominaremos RFID/IoT.[8] Dada

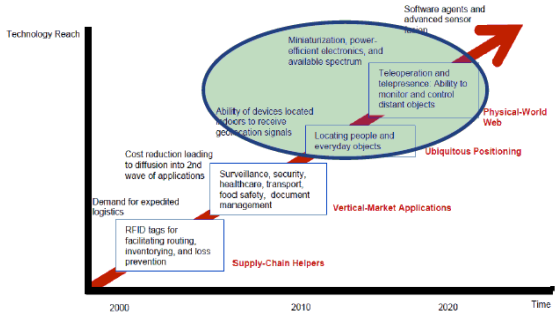
[3] Desde 2005, la organización opera bajo las siglas GS1.

[4] El ONS se encargará de muchas más solicitudes que de nombres de dominio de servicio de la Web por lo tanto es probable que las empresas mantener servidores ONS localmente, que almacenará la información para la recuperación rápida. Así que un fabricante puede almacenar los datos de la ONS de sus proveedores actuales en su propia red, en lugar de tirar de la información fuera del sitio Web cada vez que llega un envío a la planta de montaje. A su vez, el sistema también tendrá redundancias incorporadas: por ejemplo, si un servidor con información sobre un determinado producto estrella, el ONS será capaz de señalar con el middleware RFID a otro servidor donde se almacena la misma información.

[5] EPCglobal está confiando en el trabajo ya realizado por los organismos de normalización, tales como la Oficina Internacional de Pesas y Medidas (Le Sistema Internacional de Unidades - SI) y el Instituto Nacional de de Estándares y Tecnología de Estados Unidos

[6] Los datos dinámicos en un fichero PML pueden incluir la temperatura de un cargamento de fruta, o los niveles de vibración de una máquina. Los datos temporales cambia discretamente y de forma intermitente a lo largo de la vida de un objeto.

su funcionalidad e identidad la Unión Internacional de Telecomunicaciones ha definido a esta nueva red como una red donde las cosas que tienen identidades y personalidades virtuales que operan en espacios inteligentes utilizando interfaces inteligentes para conectarse y comunicarse dentro de sociales, contextos ambientales, y de los usuarios"[9][10].



Fuente: Digital Scotland 2020: Achieving World-Class digital infrastructure: a final report to the Scottish

Como fuera señalado, la adición de dispositivos al RFID combinados con una amplia gama de nuevos datos generados por sensores aporta un significativo aumento en el número de las comunicaciones, datos de intercambio, atributos, posibilidades técnicas, empresariales y de gobiernos como ser información de lugares, personas, contextos y objetos determinación del alcance y posibilidades de análisis disponibles; información logística y de flujos de datos, detección de operaciones en el mundo real, recopilación de datos remoto y automatizado y almacenamiento, incluyendo flujos multimedia, funcionamiento remoto a través de cualquier geografía, ofreciendo a su vez la posibilidad de añadir inteligencia de diferentes tipos en función de las necesidades. Las proyecciones de las investigaciones indican que cualquier objeto tendrá una forma única de identificación en el futuro, lo que se conoce comúnmente en el campo de las ciencias como *Unique Direction 1*, creando una continuidad de direccionamiento de datos entre computadoras, sensores, actuadores, teléfonos móviles y por supuesto, entre personas. Todos estos objetos serán capaces de intercambiar información y, si es necesario procesar activamente la información de acuerdo con esquemas predefinidos. Es por ello que también se refiere al IoT como una red mundial de objetos interconectados de forma única direccionable, basada en protocolos de comunicación estándar

4. Principales derechos susceptibles de ser vulnerados en un entorno RFID/IoT

La integración de los dispositivos señalados en redes híbridas inalámbricas y de sensores que se caracterizan por la modularidad, fiabilidad, flexibilidad, robustez y escalabilidad han llamado desde hace un tiempo al interés de diferentes sectores, no sólo el sector público y empresarial sino también para cibercriminales. Los datos alojados en el tag, pueden ser de carácter financiero o personal hasta incluso información importante para la seguridad nacional (e.g. pasaportes) como ser tratamiento de datos por medio de la ocultación de tags; tracking y monitoreo; *snooping*; *cracking; cloning; sniffing; malware* y denegación de servicio. Agrava esta situación la posibilidad que el *malware* RFID/IoT podría causar más daño que el malware tradicional basado en computadoras normales: esto es así puesto que el malware utilizado para atacar un RFID tiene también efectos en el mundo offline y en consecuencia los objetos tagueados son susceptibles de daño. Tal como lo ha señalado la *Organization for Economic Co-operation and Development* (OCDE), los sistemas RFID se encuentran, tal como cualquier otro tipo de sistema de información, sujetos a riesgos de seguridad.[11] En este contexto, el RFID, como una herramienta más para identificar personas u objetos, toma también protagonismo para aquellos que pretendan cometer un delito informático. Desde el punto de vista del consumidor, si bien no todos

7 En inglés, esta tecnología se conoce como Wireless Sensor Networks (WSN) y, también, como Wireless Sensor and Actuator Networks (WSAN)

[88] La investigación académicas y los trabajos doctrinarios comenzaron a abordar la temática RFID con antelación al IoT. A los fines del presente trabajo, y a fin de respetar dicha consecuencia histórica, se ha decidido hacer referencia a tal relación como RFID/IoT, lo cual no implica desconocer que el último concepto incluye otras tecnologías que, combinadas con la primera, brindan un espectro mucho más amplio de funcionalidades.

9 Una definición diferente, que pone el foco en la perfecta integración, podrían formularse como objetos interconectados tener un papel activo en lo que podría llamarse la Internet del Futuro.

10 En línea con lo global compromiso con el logro de la Objetivos de Desarrollo del Milenio (ODM), la Cumbre Mundial sobre la Sociedad de la Información (CMSI) se centra el desarrollo de las TIC a través de la creación de estrategias electrónicas nacionales, la garantía de acceso universal, ubicuo, equitativo y asequible a la tecnología y la más amplia difusión y el intercambio de información y conocimiento. CMSI compromisos van más allá de la difusión tecnológica - hay un compromiso para acción común para reducir la pobreza, la mejora del potencial humano y el desarrollo global a través de las tecnologías de comunicación y relacionados tecnologías emergentes. En este sentido, la tecnologías subyacentes de Internet de la oferta Cosas muchos beneficios potenciales.

11 Organisation For Economic Co-operation and development, RFID Radio Frequency Identification, OECD Policy Guidance A Focus on Information Security and Privacy Applications, Impacts and Country Initiatives, OECD

[11]Ministerial Meeting on the Future of the Internet Economy, Seoul, Korea, 17-18 June 2008, p. 4, disponible en http://www.oecd.org/sti/ieconomy/40892347.pdf, última visita 10 de Julio de 2013.

los usos del RFID devienen en implicancias relacionadas con la privacidad, lo cierto es que al tener la habilidad dichos sistemas de recolectar o procesar información relativa a un individuo identificado o identificable, transforma su status en vulnerable para su vida íntima debido a que pueden realizarse sin su consentimiento.[12] Si un delincuente accediera o tomara control del sistema RFID utilizado por una empresas y en consecuencia de los datos transmitidos por las etiquetas , tendrían la posibilidad de podrían determinar que los productos a compras de los consumidores, con qué frecuencia se utilizan estos productos, e incluso cuando el producto – y por extensión el consumidor - viajes. Mediante la agregación de datos para formar perfiles de consumidores, las empresas o un tercero podrían manipular los datos para hacer suposiciones deductivas sobre los ingresos de los consumidores, la salud, estilo de vida, hábitos de compra, y la ubicación. Y esas bases de datos, e incluso la etiqueta, pueden ser objetivos de hackeo. Estas preocupaciones son incluso más pertinentes cuando las tecnologías se combinan, integrando y conectando, de forma invisible y remota la computación ubicua. La comprensión de la cibercriminalidad, con referencia a la necesidad de la protección de la intimidad digital, reconoce que los datos personales son los activos intangibles de gran valor, tanto para empresas (como personas jurídicas) como para los delincuentes. A su vez, desde una perspectiva estatal, la soberanía y la infraestructura crítica de un país puede verse sometida a ataques a una o más partes de la estructura RFID/IoT teniendo consecuencias para toda una población de alcances de gravedad. Cabe destacar asimismo las implicancias relacionadas con la neutralidad[13] en la red máxime cuando en un mercado libre con operadores o en caso de industrias competitivas como así también la importancia en materia de ciberdefensa en ocasión de un tagueado masivo de infraestructura crítica y sus consecuencias para la soberanía de un Estado.

3. La gobernanza de Internet

El término gobernanza de internet (GI) hace referencia en sentido amplio al conjunto de actividades necesarias para el normal funcionamiento de internet -incluyendo el conjunto de acciones requeridas para su desarrollo en determinado sentido- que son llevadas a cabo por una multiplicidad de actores.[14] Si bien internet muta constantemente por decisiones de carácter técnico-político, creando posibilidades y restringiendo otras, lo cierto es que no existe aún un organismo único que vigile y coordine su funcionamiento, sino que una serie de actores (generalmente no estatales) realizan diferentes gestiones administrativas[15] por sí mismos o junto a otros, pero siempre en forma coordinada debido a la enorme complejidad que la tarea representa. Así, los actores intervinientes se relacionan entre sí en un modelo de múltiples partes interesadas o *multistakeholder* (MS), divididos por lo general en cinco categorías: Estados, organizaciones intergubernamentales, representantes del sector privado, comunidad técnica y académica, y miembros de la sociedad civil. La literatura indica que actores participan en igualdad de condiciones en los diferentes espacios de diálogo convocados al efecto[16], y las tensiones surgen frecuentemente a partir de lo que cada uno considera como rol propio, y por el rol que cree que los demás deberían ocupar.

4. La gobernanza del IoT

Se define a la gobernanza de Internet de las Cosas (GIoT) como aquel ámbito de la GI destinado a generar mecanismos y reglas de gobierno de los dispositivos,

[12] El debate sobre el consentimiento expreso o tácito de un consumidor al ingresar a una tienda comercial merece un tratamiento especial en doctrina que excede el propósito de la presente ponencia.

[13] Aunque RFID no es una tecnología nueva, su utilización en la cadena de suministro y las implementaciones basadas en el consumidor permanecen en una etapa relativamente incipiente. Desde implementaciones de RFID se están produciendo principalmente en la cadena de suministro de negocios y no todavía en los puntos de venta al consumidor, hay que preguntarse cuál es la "Internet de las cosas" en realidad es hoy y lo que, en su caso, los principios de gobierno deben aplicarse. Más importante aún, antes de aplicar los principios de gobernanza, es necesario examinar la naturaleza de las distintas redes de RFID que constituirían el Internet de las Cosas. Dependiendo de su arquitectura, la seguridad y los modos de interconexión, nuevas redes de RFID entre las entidades comerciales se podrían considerar "redes privadas" o "grupos cerrados de usuarios" (redes privadas que utilizan TCP / IP se abordan en el RFC 1918 que igualmente reconoce estas distinciones entre lo privado y las redes públicas). Redes privadas y grupos cerrados de usuarios son generalmente exentos de las regulaciones de telecomunicaciones tradicionales, ya que no se interconectan con la "red pública de telecomunicaciones" u otras redes públicas abiertas como Internet. En algunos contextos, la cuestión de si una red privada está sujeta a una obligación reglamentaria se convierte en realidad en la forma en que las interconexiones de red privada con la red pública.

[14] La definición referida sigue la línea conceptual propuesta por el Grupo de Trabajo sobre Gobernanza de internet, incorporada en el parágrafo 34 de la Agenda de Túnez, uno de los documentos finales de la fase de Ginebra de la Cumbre Mundial sobre la Sociedad de la Información (2005).

[15] Una de las pocas clasificaciones puntuales, que desarrolla la actual GI en 44 aspectos administrativos diferentes, puede encontrarse en DENARDIS, L. y RAYMOND, M. (2014), "Thinking Clearly About Multistakeholder internet Governance", disponible en http://www.phibetaiota.net/wpcontent/uploads/2013/11/Multistakeholder-Internet-Governance.pdf, última vista 4 de octubre de 2014.

[16] Véase por ej. la formulación de los artículos 72, 73 y 74 de la Agenda de Túnez mediante los cuales se crea el Foro de la Gobernanza de internet (IGF).

infraestructura y funcionamiento de loT. Se considera que los principios generales de gobernanza de Internet, tal como se establece en la Cumbre Mundial de la Sociedad de la Información, los estatutos de ICANN y las declaraciones del Foro de Gobernanza de Internet (IGF)[17], brindan orientación suficiente para el desarrollo y la implementación de nuevos servicios y aplicaciones IoT. En este contexto y en relación con la administración y gobierno de los ONS existe quienes sostienen que los mismos debieran organizarse de una manera diferente que el DNS - a fin de reparar ciertas debilidades de los primeros- mientras que otras posturas lo hace distinción alguna o no ven ninguna necesidad de utilizar la ONS para "reparar" algunas debilidades del DNS. Otra línea regulatoria sostiene que es dable la creación de una regulación desde cero, con orientación de mercado y gestión de ONS diferente a la realizada actualmente para los DNS. De esta manera, y más allá de los aspectos relacionados con la gestión antedicha, en términos generales es posible identificar dos propuestas de gobierno del IoT, a saber:

i. *Creación de un mecanismo especial de G IoT* tanto en el mundial, así como en el nivel regional y nacional, incluida la opción de un intergubernamental IoT organización del tratado en / o en relación con las Naciones Unidas.

ii. *Apoyar el sistema vigente GI o ecosistema de internet* (IETF, ICANN, RIR, ISOC, IEEE, IGF, W3C, etc) ya que el mismo contempla un espacio suficiente para lidiar con los temas especiales IoT relacionados con cuestiones de gobernanza de recursos tecnológicos. Se sostiene que debido a que internet es abierto, cambiante e interoperable donde la privacidad, la seguridad y la competencia han sido identificados como los principales temas relacionados con IoT son temas de GI, las mismas no deben ser discutidos en una forma separada o aislada sino por el contrario, incluirlas en la dinámica del modelo multistakeholder[18] y en los ámbitos de participación correspondiente. En este contexto resulta importante contemplar un acuerdo común sobre la terminología y conceptos es necesaria y una buena comprensión de la propia Internet por todas las partes interesadas no se puede asumir.[19]

Según estas dos posturas, un grupo ve la gobernanza de IoT como una cuestión separada especial, otros definen la IoT como "otra aplicación" en la parte superior del DNS (junto a otras aplicaciones de Internet, como el correo electrónico, las redes sociales y así sucesivamente). Asimismo existen controversias sobre la comprensión del ONS (en comparación con el DNS), y políticas de asignación de identificadores, incluyendo direcciones IPv6 (IGF 2012), y a su vez ambos modelos otorgan diferentes cualidades a la presencia de los Estados en materia regulatoria.

5. Los gobiernos y su participación en la cumbre Net Mundial – El rol de los Estados en las posturas de regulación.

La iniciativa Net Mundial[20] tuvo como fin convocar a las múltiples partes interesadas a discutir temas de relevancia en relación a la GI. De esta manera, puede considerarse como el primer foro internacional que buscaba lograr dos resultados concretos: una lista de principios sobre la GI y una hoja de ruta para la futura evolución de su ecosistema [21] Si bien el encuentro mencionado recibió 188 contribuciones, la presente ponencia tomará en consideración aquellas que fueron enviadas por gobiernos, las cuales representan un total de veintitrés (veintidós gobiernos de países y la comunidad europea). Todas ellas centraron su aporte escrito ya sea en relación a la realización de propuestas de principios de GI, de hoja de ruta para la gobernanza de internet o bien, ambos. En relación con el contenido de las contribuciones, es posible identificar una variedad de temas de referencia, que se concentran en tres ejes temáticos y un grupo de

[17] La IoT se encuentra en la agenda de las conversaciones informales en el IGF. La Coalición Dinámica sobre IoT (DC IoT) fue creado, que tuvo su primera reunión inaugural en Nairobi, octubre de 2011, habiendo ya informado que todavía hay grandes controversias acerca de la comprensión básica y definición de la gobernanza de la IoT

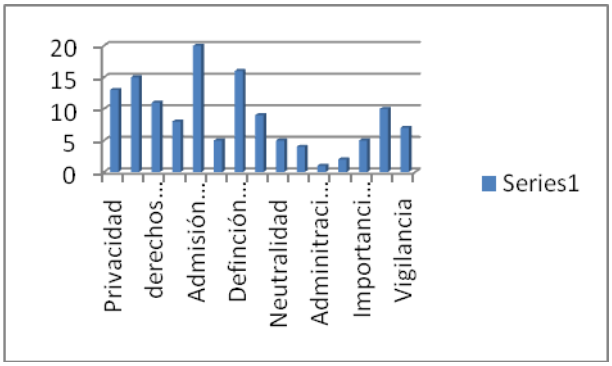
[18] El gobierno de un multistakeholder podría ser define como la suma de las muchas maneras los individuos y las instituciones, públicas y privadas, manejan sus asuntos comunes. Es un proceso continuo a través del cual los conflictos de intereses o diversas pueden ser acomodados y se puede tomar la acción cooperativa. Incluye las instituciones formales y los regimen facultados para exigir el cumplimiento, así como los arreglos informales que las personas y las instituciones han acordado o percibir ser de su interés.

[19] Por ejemplo, una gran cantidad de atención en el IGF permanece en "recursos críticos de Internet", que, hasta la fecha, ha significado los nombres de dominio, servidores raíz y administración de direcciones IP, y sólo recientemente se comienzan a hacer referencias a cuestiones relativas al IoT. Si el diálogo IGF es proporcionar elementos básicos de análisis para la aplicación de los principios de gobierno a las nuevas tecnologías y en evolución, hay una prima en el diálogo IGF conseguir el conceptual adecuado marco.

[20] Llevada a cabo el 23 y 24 de abril de 2014.

21 Si bien es el único foro en su especie comparte ciertas generalidades con otros encuentros mundiales de la temática. NetMundial se diferenció por su carácter y por su origen. Se apartó del modelo de foro dirigido exclusivamente a un tópico determinado y contó con considerable participación de todo tipo de partes interesadas. En cuanto a su origen, no se creó a partir del apoyo exclusivo de algún organismo ya existente, y fue promovido principalmente por un gobierno latinoamericano. Cabe destacar que la lógica de foros mundiales en donde las múltiples partes interesadas se reúnen a debatir sobre temas específicos (aún sin producir documentos concretos o vinculantes) no es nueva. El IGF tuvo ocho ediciones desde el 2006. También se desarrollan con asiduidad foros preparatorios y/o específicos a niveles nacionales y regionales.

casos especiales (CE), los cuales pueden sistematizarse en derechos fundamentales (E1), modelo de gobernanza (E2) y cuestiones relativas a la infraestructura de la red (E3). Para el presente artículo destacar que el interés temático menos importante para los gobiernos fueron los temas de infraestructura y de hecho, un menos de uno 15% consideró que tales temas debieran obtener una tasa alta de participación gubernamental en el diseño de estrategias, políticas, gestión y control del tráfico en la red.



Fuente: Elaboración propia del autor según las contribuciones publicadas en www.netmundial.br

Dicha postura puede ser extendida al análisis de la regulación RFID/IoT. En esencia, la mayoría de los actores parecen creer que no es necesario un nuevo marco de gobernanza, dedicada al IoT. Se considera que las cuestiones de política-IoT específicos que pueden ser tratados en el marco de las plataformas de gobernanza de Internet existentes y que el curso IoT proceso de múltiples partes interesadas que tienen lugar en las plataformas existentes para generar políticas sobre la IoT.

6. El rol de los Estados respecto de los recursos de Internet y su relación con el RFID/IoT

Como se destacara, respecto de la IoT, la comunidad global se encuentra dividida actualmente entre los países que se ven principalmente su gobernanza en el marco de organizaciones con múltiples partes interesadas para hacerse cargo de los aspectos importantes de la Internet, como ICANN (DNS y IANA) y IETF (elaboración de normas), y otros que depositan su confianza en las instituciones impulsadas por los gobierno como la UIT[22] o tratados internacionales especializados en la materia.[23] Cuando se considera que uno de los temas centrales en el debate de la gobernanza del IoT radica entre cómo poder lograr el equilibrio adecuado entre la funcionalidad e interoperabilidad del sistema (a cargo de los actores del sector y no gubernamentales privadas) junto con la necesidad de legitimidad con mayor participación de los gobiernos, dos opciones se presentan posibles en tal sentido:

a. Otorgar plena potestad a los Estados para la gobernanza de los recursos de la infraestructura de la IoT;

b. Designar claramente en el marco del modelo multistakeholder roles precisos con injerencia y supervisión detallada por parte de los gobiernos de aquellos actores designados para la administración de los recursos.

Sin embargo, ya sea una u otra opción, luego de Netmundial pareciera ser que la mayoría de los Estados que han enviado contribuciones a dicha Cumbre no se encuentran interesados en participar en tales actividades. Las explicaciones pueden obedecer a diferentes razones: ya sea porque tradicionalmente los sectores técnicos han llevado adelante una administración "abierta" de la red (aunque no supervisada con la consecuencia rendición de cuenta por parte de los Estados hacia sus ciudadanos) , o

[22] Quienes sostienen la necesidad de concentrar en un organismo internacional la regulación de la IoT adscriben al debate inicial de internet y la zona raíz. Desde los primeros días de la Cumbre Mundial de la Sociedad de la Información en 2002, la posibilidad teórica de la eliminación de los nombres de dominio de otros países - mediante la supresión de nombres de dominio del país a través de Internet - ha inspirado a muchos países a criticar el papel clave de los EE.UU. en la gestión de la base de datos de zona de las raíces. Una abrumadora mayoría de los países argumentan que la zona de la raíz en una Internet como una infraestructura global debe ser gestionada por una organización internacional que no está bajo el control o la jurisdicción de un solo país. Este argumento se formula a menudo como una cuestión de igualdad soberana, y la equidad en las relaciones internacionales que se ocupan de Internet Una de las opciones podría estar haciendo la base de datos del servidor y raíz inviolable, en particular, de cualquier jurisdicción nacional. Esta posibilidad abre la cuestión de dónde se encuentra el servidor raíz para asegurar la inmunidad de cualquier jurisdicción nacional.

[23] Como parte del debate sobre un modelo apropiado de la gobernanza de Internet, las críticas de la ICANN han surgido en dos frentes. Una crítica refleja la tensión entre los gobiernos nacionales y los procesos de funcionamiento y de gobernanza actuales de ICANN, mediante el cual los gobiernos se sienten carecen de influencia adecuada sobre las decisiones de ICANN que afectan a una amplia gama de políticas de Internet cuestiones. La otra crítica es alimentado por las preocupaciones de muchas naciones que el Gobierno estadounidense mantiene sus influencia indebida legado y el control sobre ICANN y el sistema de nombres de dominio.

por una tradición no-técnica como acción asociada a los roles del Estado, por no existir una política pública respecto del alcance de la IoT y tecnologías conexas o quizás por considerar que la mejor gobernanza es la auto-regulación del mercado (aún a costas de ignorar que existen intereses superiores ciudadanos que son susceptibles de ser vulnerados por la actividad privada), por la naturaleza dinámica o mutable del IoT que hace imposible su regulación, por impedir la innovación, o bien porque resulte difícil concebir el rol de los gobiernos de manera descentralizada en relación a estos temas y que por lo tanto es lógico que las empresas tomen ese lugar.

7. La importancia de la participación de los Estados en el diseño, administración y gestión de los recursos de RFID/IoT

Si el IoT representa una extensión y expansión de Internet tradicional para incluir los objetos físicos, cualquier legislación aplicable al internet probablemente será aplicable a la IoT, al menos hasta cierto punto ¿Qué rol debieran representar los Estados en dicha regulación? La respuesta tendrá relación con la necesidad de una evaluación del estado actual o las tecnologías más avanzadas a fin de poder brindar, por parte de los gobiernos, indicaciones claras de cómo se implementará el IoT en un nivel universal en los próximos años, así como indicar los aspectos importantes que necesitan ser más estudiados y desarrollados, lo cual conllevará un trabajo importante en el ámbito de la gobernanza. Sin un enfoque estandarizado es probable que una proliferación de arquitecturas, esquemas de identificación, protocolos y frecuencias desarrolladas por separado y sin una integración operativa. Esto llevará inevitablemente a una fragmentación de la IoT, lo que podría obstaculizar su popularidad y convertirse en un obstáculo importante en su despliegue. La interoperabilidad es una necesidad, y la comunicación inter-tag/sensor es una condición previa para que la adopción de la IoT para ser extendida. Por ello, resulta necesaria la creación de un marco común para la interacción entre una amplia gama de dispositivos, necesarios para conservar la apertura y evolución de la IoT. En tal sentido se considera que la predominancia de los Estados en la gobernanza del RFID/IoT y en consecuencia debe fomentarse un mayor involucramiento en diálogos intergubernamentales a tal respecto que fomenten la apertura, interoperabilidad[24], competencia, aseguren el respeto de los derechos de consumidores y ciudadanos como así también generen los mecanismos legales necesarios para el resguardo de los derechos afectados.

En la actualidad, las normas aplicables a la IoT son producidas por una serie de organismos donde los Estados, o la mayoría de ellos, tienen poca o ninguna injerencia. Es por ellos que se fomenta la interacción con los representantes de los organismos de normalización sugieren que el proceso de creación de normas y refinamiento combina la cooperación dentro de un organismo de normalización específica) con una fase competitiva (entre las diferentes normas). Especialmente en el caso de las normas (a menudo equivalente a la elección de un cuerpo de estándares) la comunicación y la interoperabilidad, son requisitos a tener en cuenta para evitar todo obstáculo al desarrollo integral y la coherencia del conjunto normas de disponibles como así también al resguardo del usuario. La mayoría de los estándares de la IoT a la fecha han sido creados por organismos liderados por la industria, con una orientación de ingeniería y relativamente pequeño avance y/o representación del sector público y la sociedad civil. Esto ha generado .preocupación por las posibles implicaciones competitivas de estandarización impulsadas por la industria: muchos han llamado la atención sobre el potencial de las normas impulsadas por las empresas dominantes para excluir la competencia, especialmente en vertical (por ejemplo, limitando las maneras dispositivos individuales pueden interactuar con y sobre plataformas comunes) peer-to-peer en lugar de la comunicación cliente-servidor.

Hoy en día EPCglobal[25] se define como un consorcio neutral, basado en el consenso para el desarrollo de normas impulsadas por la industria y apoyar la identificación por radio frecuencia en las redes de hoy en día bajo el directorio raíz de la ONS[26] Este escenario, donde los Estados no demuestran interés para intervenir,

[24] Por ejemplo, el diseño de estándares para controlar una amplia gama de aspectos de la RFID/IoT, de normas técnicas relativas a los dispositivos físicos y uso del espectro electromagnético con las normas para la codificación de la información y protocolos para la comunicación entre dispositivos. Muchos de ellos se heredan de o se aplican también a otros dominios en los que estos dispositivos funcionan - esto a su vez crea el potencial para conflictos de objetivos de la normalización y procedimientos incompatibles para definir, acordar, hacer cumplir y supervisar las normas. En este sentido, la discusión más importante es sobre la necesidad de normas específicas de IoT-, ya que por ahora sólo se han creado iniciativas de estándares específicos para incorporar y adaptar las normas existentes pero no una regulación específica sobre la materia.

[25] Sus raíces se encuentran en el trabajo que se ha realizado por el centro de AutoID, un consorcio liderado habían organizado por el Instituto de Tecnología de Massachusetts. El marco arquitectónico EPCglobal se basa en el Servicio de Información EPC, el cual está compuesto por la información proporcionada por el fabricante y las distintas partes interesadas. Ver: http://www.verisign.com/static/039451.pdf

[26] Un servicio de búsqueda en el centro, la raíz de la ONS se puede controlar y / o bloqueado por una sola empresa o un país, a diferencia del sistema DNS

deja abierta la posibilidad de una administración de recursos críticos tecnológicos en mano del sector privado, técnico e incluso mixto, donde los gobiernos, o no tienen participación o interés alguno en participar o bien debe restringir su incidencia o compartirla con otros sectores. El peligro de un sistema unipolar es que la empresa que controla la ONS tiene el poder de aislar a las empresas de productos, y la obtención de información vital (para los competidores) sobre la circulación de mercancías, recursos relacionados la defensa nacional, privacidad de los ciudadanos, etc. Los gobiernos debieran participar no como autoridades de supervisión sino como principales partidarios de un marco jurídico común que mantenga la rendición de cuentas en un modelo multilateral apropiado en un enfoque de múltiples interesados en la que todos los grupos implicados participen en igualdad de condiciones para llevar a cabo su función[27] con políticas de intervención bien enfocadas y específicas.

8. Conclusiones

Con el nuevo desarrollo de las tecnologías de radio frecuencia y el uso de la IoT, el debate alrededor de la Gobernanza de Internet demanda la definición de nuevas reglas para la GIoT. Se considera importante el RFID en tal contexto y por ello se estima que el gobierno de RFID/IoT o debe continuar a través de un enfoque multilateral donde los gobiernos tenga un papel activo en la gobernanza de los aspectos técnicos de la red pero no en calidad de asesores, sino con el debido reconocimiento y teniendo en cuenta que ningún gobierno retenga un legado exclusivo para dominar que la gobernanza. Esto no obsta a que puedan llevar adelante acciones para coordinar integrar y supervisar los organismos encargados de los aspectos técnicos y operativos, donde futuras iniciativas políticas deben orientarse usuario, impulsado por el mercado y necesaria para mejorar la privacidad de los usuarios finales, la seguridad y la competencia leal. Cuestiones de derechos humanos, privacidad, ciberseguridady ciberdefensa son ejes relacionado con el entorno RFID/IoT, y es por ello que no puede dejarse librado al sector privado o técnico su regulación o falta de organización. El Estado tiene que estar presente porque tiene interés público, las empresas sólo tienen intereses privados; es por ello que urge una mayor interés y expresión de posición por parte de los Estados respecto de su rol como representante de los intereses ciudadanos en el marco de la administración de los recursos e infraestructura RFID/IoT. Cuando la tecnología es susceptible de transformar la sociedad, el desafío político es inevitable El marco legal respecto a la privacidad y la seguridad deben adaptarse a una nueva realidad, renovando la democracia y desafiando las estructuras de poder existentes. Por tanto, es importante que la investigación socio-económica y el debate político sobre el Internet de las cosas vayan de la mano con la investigación tecnológica y los avances gubernamentales en la materia.

.

[27] Quienes no comparten este argumento sostienen que la gobernanza por medio de la regulación gubernamental es perjudicial y que el "reglamento mata la innovación", aduciendo que dada sus características y potencia expansión, el IoT debe ser gobernado por las corporaciones exitosas. Pareciera ser que el derecho internacional y las relaciones entre Estados no pudieran ser parte de la dinámica del entorno RFID/IoT .

9. Referencias

[1] Albergant i, Michel, Sous l'oeil des puces : la RFID et la démocratie, Paris, Actes Sud, 2007.

[2] Albrecht , Katherine. "Supermarket Cards: The Tip of the Retail Surveillance Iceberg." Denver University Law Review, Volume 79,Issue 4, Summer 2002.

[3] Bohn, L. et al., 'Social, Economic, and Et hical Implicat ions of Ambient Intelligence and Ubiquitous Comput ing', Inst itute forPervasive Comput ing, ETH Zurich, Switzerland, 2004.

[4] Borking J. and Raab C., 'Laws, PETs and Other Technologies for Privacy Protect ion', Refereed art icle, 2001 (1) The Journal ofInformat ion, Law and Technology (JILT).<http://elj.warwick.ac.uk/jilt /01-1/borking.html/>.

[5] Colin J.Bennett and Rebecca Grant , " Visions of P rivacy: P olicy Choices for the Digit al Age", Toronto, Universit y of Toronto P ress 1999.

[6] Electronic Privacy Information Cent er, "Radio Frequency Ident ificat ion (RFID) Systems", Washington, D.C., August 11, 2003,http://epic.org/privacy/rfid/

[7] Fisher, Joy, Data t ransmission and privacy, Boston, ed. by Dennis Campbell, 1994.

[8] Floerkemeier, C., Schneider, R., Langheinrich, M., 'Scanning with a purpose – Support ing the Fair Informat ion Principles in RFIDProtocols', Zurich, 2004. Disponible en : http://interval.huberlin.de/downloads/rfid/prevention/floerkem_scanning%20with%20a%20purpose.pdf

[9] Garfinkel, S.L., Juels, A., P appu, R., 'RFID privacy: an overview of problems and proposed solut ions', IEEE Security and Privacy, Vol. 3, No. 3, IEEE Computer Society Press, May-June 2005.

[10] Garfinkel, S.L., RFID: Applicat ions, Security, and Privacy, Boston, Rosenberg B. (eds), 2005.

[11] Federal Trade Commission Radio Frequency Identification, Applications and Implications for

Consumers. Disponible en http://www.ftc.gov/os/2005/03/050308rfidrpt .pdf.
[12] FIDIS, A Holist ic Privacy Framework for RFID Applicat ions D12.3. Disponible en at http://www.fidis.net/resources/deliverables/hightechid/d123-aholistic-privacy-framework-for-rfidapplicat ions
[13] Hunter, Richard, World without secrets: business, crime, and privacy in the age of ubiquitous comput ing, New York, J. Wiley & Sons, 2002.
[14] Hutto Julie and Robert D. Atkinson, Radio Frequency Identification: Litt le Devices Making Big Waves, Progressive Policy Institute, Policy Report , October 6, 2004. http://www.ppionline.org/documents/RFID_1006.pdf
[15] Juels, A., Rivest , R. L., Szydlo, M., 'The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy', 8th ACM Conference on Computer and Communicat ions Security, ACM Press, Washington, 2003.
[16]Final Report of the EU IOT Task Force on IOT Governance. Disponibel en www.consilium.europa.eu/uedocs/.../117236.pdf
[17] Comisión Europea. Report on the public consultation on IoT Governance. Disponible en ec.europa.eu/information.../cf/.../document.cfm?

Relevamiento de los factores que afectan al pleno desarrollo de la industria del software y servicios informáticos en el interior de la Provincia de Córdoba

Luciano Báez Moyano, Fernanda Barbero, Ana Carolina Ferreyra, Ricardo Medel,
Mariano Emilio Rodríguez, Silvana Sánchez Fernández
Secretaría de Investigación
Universidad Empresarial Siglo 21
Calle de los Latinos 8555, Córdoba, Argentina
lucianobaez@outlook.com, fernabel@gmail.com, ana_carolina_ferreyra@hotmail.com,
ricardo.h.medel@gmail.com, merodriguez@gmail.com, sanchezfernandezsilvana@gmail.com

Resumen

La expansión de las telecomunicaciones, especialmente de Internet, y el incremento en el uso del software ha dado paso a la así llamada "economía digital". En este contexto, el desarrollo de software puede realizase, en principio, desde cualquier lugar del mundo que posea conectividad y recursos humanos calificados. Sin embargo, en los últimos años la industria del software y servicios informáticos en el interior de Argentina, fuera de las grandes ciudades, no ha mostrado un crecimiento similar al producido en los grandes centros poblados. En este trabajo presentamos los resultados iniciales de nuestra indagación en las razones por las cuales la industria del software aún no crece a las tasas esperadas en el interior del país. Nuestro objetivo es definir una estrategia que permita superar las limitaciones encontradas y fortalecer los factores, quizás únicos a esta región geográfica, que favorecen el crecimiento de esta industria.

1. Introducción

Los avances tecnológicos de las últimas décadas del siglo veinte y de principios de este siglo han modificado la producción de bienes y servicios, favoreciendo la globalización de dichas actividades. Aunque la instalación de fábricas en países menos desarrollados tomó auge en los años '70, con la combinación de las tendencias llamadas *outsourcing*[1] y *offshoring*[2], recién en los años '90 el desarrollo de software y la prestación de servicios informáticos comenzó a expandirse hacia los países en vías de desarrollo, alejándose geográficamente de su mayor mercado de consumo, los países con economías con mayor desarrollo [5].

En sus comienzos el *outsourcing/offshoring* en el área informática estuvo dominado por grandes empresas (por ejemplo, EDS e IBM) instalando sus centros de desarrollo (*software factories*) en países periféricos, especialmente en India. Más recientemente, el advenimiento de la "economía digital" [7], con su eje central en la expansión global de las telecomunicaciones, en particular de Internet, ha permitido la proliferación de empresas locales de países en vías de desarrollo que prestan servicios y desarrollan software para mercados globalizados [8]. La economía digital, apuntalada por la creación de tiendas de aplicaciones[3] [9, 10] y la difusión de sitios web donde ofrecer servicios como desarrollador independiente o *free-lance* (por ejemplo, www.guru.com ofrece contactos con un millón y medio de desarrolladores independientes), ha impulsado el desarrollo independiente de software. Tanto para la demanda como para la oferta de bienes y servicios, internet permite hasta cierto punto liberar a las personas de las limitaciones de su entorno [1].

Por otra parte, el desarrollo de software es considerado desde cierto punto de vista como una forma de arte [2, 3, 4]. Por ejemplo, la Ley 11.723 de Propiedad Científica, Literaria y Artística, de la República Argentina equipara al software con las restantes expresiones artísticas, lo que permite reconocer el derecho de autor de los programadores sobre dichas obras. Aún sin llegar a reconocer el software como una forma de expresión puramente artística, muchas de las

[1] *Outsourcing*: subcontratación de terceros para hacerse cargo de actividades complementarias a la actividad principal de la empresa. Actividades que originalmente estaban en manos de la empresa.

[2] *Offshoring*: subcontratación de procesos de negocios de un país a otro, usualmente en busca de reducir costos.

[3] Tienda de aplicaciones o *app store*: servicio en línea que permite adquirir software y descargarlo directamente vía web.

actividades incluidas en el desarrollo de software requieren de habilidades creativas y artísticas [6].

La condición de industria artística que le asignamos a la industria del software, o al menos a parte de ella, nos permite suponer que buena parte de sus practicantes, al igual que buena parte de los artistas tradicionales, gustarían de desarrollar sus carreras en pequeñas comunidades alejadas de las ciudades. El estilo de vida de estas comunidades, alejadas de los problemas normalmente asociados a las grandes urbes, tales como contaminación, caos de tránsito, largos tiempos requeridos para el transporte desde y hacia el hogar, inseguridad y crimen, supone un fuerte atractivo para las mentes artísticas [4].

Es por todas estas condiciones (la posibilidad de desarrollar software de manera independiente o distribuida, proveyendo servicios a través de internet, y la calidad de artista que cierta parte de la industria tiene) es que podríamos hipotetizar que los desarrolladores de software querrían desarrollar sus carreras desde pequeñas comunidades alejadas de los grandes centros urbanos. Sin embargo, a pesar de que, en principio, la distribución de las actividades de desarrollo y prestación de servicios informáticos podría ser totalmente distribuida geográficamente, se advierte un fenómeno de concentración de estas actividades en grandes ciudades y polos tecnológicos.

Nuestra hipótesis es que si bien hay factores que benefician el asentamiento de desarrolladores independientes y pequeñas empresas de desarrollo de software en comunidades pequeñas, existen otros factores que impiden o dificultan dicho proceso. En este artículo presentamos los pasos que hemos realizado para comenzar a explorar esta cuestión, con el objetivo de definir estrategias de superación de las limitaciones y fortalecimiento de los factores benéficos para la radicación de la industria de desarrollo de software y servicios informáticos en pequeñas localidades, en particular de la provincia de Córdoba.

2. Metodología

Los resultados parciales que presentamos en este artículo han sido obtenidos por medio de actividades realizadas en el marco de nuestro proyecto de investigación, el primero en nuestra universidad con temática referida al software.

Como se mencionó anteriormente, la hipótesis sobre la que trabajamos es que las localidades pequeñas (fuera de las capitales provinciales y otros polos tecnológicos establecidos) se benefician económica y culturalmente si allí se establecen desarrolladores de software independientes y pequeñas empresas de desarrollo de software y prestación de servicios informáticos. Dichas localidades tienen ventajas sobre ciudades grandes, lo que las hace atractivas para cierto número de participantes de esta industria. Sin embargo, hay factores que impiden o dificultan este proceso de radicación y crecimiento.

A fin de relevar tanto los factores positivos como negativos se decidió realizar una encuesta lo más abarcativa posible entre desarrolladores independientes, dueños de PyMES informáticas y empleados ya radicados en localidades pequeñas. Por razones de recursos se decidió relevar el interior de la provincia de Córdoba, dejando de lado la capital provincial, un conocido polo de desarrollo industrial y de software, con empresas tanto locales como internacionales radicadas desde hace algunos años.

La provincia de Córdoba se ubica en el centro de la República Argentina, y si bien es la quinta provincia en superficie (165 mil km2), es la segunda más poblada del país con 3.308.876 habitantes (según censo 2010). El 40% de su población está concentrada en su ciudad capital, también llamada Córdoba, con 1.329.604 habitantes. Esta ciudad es considerada, por su importancia, la segunda ciudad del país, después de la Ciudad Autónoma de Buenos Aires. Tiene una importante presencia industrial, especialmente en los rubros automotor y metalúrgico. La ciudad capital concentra seis universidades, dos de ellas públicas, incluyendo la Universidad Nacional de Córdoba, la más antigua del país, fundada en 1613. Otras cuatro universidades se distribuyen en la provincia: Universidad Nacional de Río Cuarto, Universidad Nacional de Villa María y las Facultades Regionales San Francisco y Villa María de la Universidad Tecnológica Nacional.

La provincia se divide político-administrativamente en 26 departamentos, cada uno con una ciudad cabecera. Pero teniendo en cuenta la distribución demográfica de la provincia, y a fin de realizar una toma de datos lo más balanceada posible, para este estudio la dividimos en cinco sectores geográficos: Norte, Este, Traslasierra, Sur y Sudeste, según se muestra en la Figura 1. Asimismo, se caracterizó a la población muestral en diferentes roles: desarrollador de software o prestador de servicios independiente (INDEP), empresa de desarrollo de software o prestación de servicios informáticos (PYME), departamento de TI de una empresa cuyo producto o servicio final no es informático (DPTO) y, por último, empleado, departamento o laboratorio de una empresa (nacional o transnacional, no radicada completamente en la localidad) cuyo producto o servicio final es informático (LAB).

A modo exploratorio, y previo a la confección de la encuesta, se realizaron entrevistas guiadas a un número reducido de individuos pertenecientes a la población muestral. El objetivo de estas entrevistas fue experimentar con preguntas posibles para la encuesta y descubrir temas relevantes que no hayan sido

contemplados en el primer diseño de nuestra encuesta. El guión utilizado para las entrevistas se muestra en el Anexo A. Se relevaron el tamaño de la empresa tanto respecto de la cantidad de empleados, su rol y la formación académica de los mismos como la cantidad y tipo de hardware utilizado, también se relevaron las tecnologías y metodologías de desarrollo, los productos ofrecidos al mercado y las formas de contactarse con sus clientes, el uso de telecomunicaciones, las redes de contactos profesionales y empresariales, y el acceso a financiamiento a través de subsidios o préstamos gubernamentales.

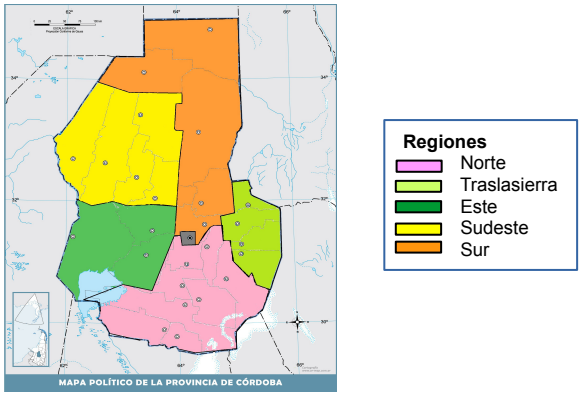


Figura 1. Regiones de la Provincia de Córdoba.

Los individuos elegidos para esta primera entrevista de exploración fueron seleccionados del pequeño conjunto de contactos disponible entonces, de modo que, como muestra la Tabla 1, se cubrieran todas las zonas geográficas en que se dividió la provincia y todos los roles profesionales.

Tabla 1. Características de entrevistados.

Individuo	Zona	Rol
1	Traslasierra	INDEP
2	Traslasierra	INDEP
3	Norte	PYME
4	Norte	INDEP
5	Norte	INDEP
6	Sudeste	PYME
7	Norte	INDEP
8	Sur	DPTO
9	Sur	PYME
10	Este	PYME
11	Este	PYME

Las entrevistas se realizaron personalmente, por videoconferencia (utilizando Skype) o por chat. El entrevistador tomó nota de las respuestas a la entrevista utilizando una computadora portátil. Cabe destacar que se hizo hincapié en que el entrevistado pudiera extenderse libremente en sus respuestas, dejando que la libre asociación de ideas nos permitiera descubrir más información sobre la situación de esta industria en el interior de Córdoba.

En las siguientes secciones se analizan los resultados de estas entrevistas. Estos resultados nos permitirán definir la encuesta de modo de poder relevar los factores más importantes en el desarrollo de la industria del software en el interior de la provincia de Córdoba. Las encuestas se realizarán sobre un subconjunto de la población total de profesionales de informática del interior de la provincia de Córdoba, lo cuales serán contactados a través de relaciones personales y/o profesionales previas y/o convocados a través de colegios profesionales y clústeres de las distintas regiones de la provincia, tratando de balancear la cantidad de encuestados para cada región geográfica.

3. Resultados parciales

Los emprendimientos están formados por un único emprendedor (5 casos) o por un pequeño grupo de profesionales que forman una microempresa ya sea formalmente o de hecho. La cantidad de personas trabajando en las empresas entrevistadas varió entre 2 y 7 personas, con la mayoría de ellas dedicadas al diseño y desarrollo de software. Solamente tres empresas tenían un empleado dedicado exclusivamente a la parte comercial o de marketing. En las otras empresas, al igual que en el caso de los independientes, este rol lo cumplía un profesional informático.

De las 27 personas involucradas en los emprendimientos relevados, 8 tienen títulos de grado en informática (de la Universidad Nacional de Río Cuarto y de las Facultades Regionales San Francisco y Córdoba de la Universidad Tecnológica Nacional), 4 tienen estudios de tecnicatura o analista y 4 tienen estudios universitarios incompletos. Lo cual indica que más de la mitad de los involucrados tienen estudios formales en informática. Del resto sólo 2 personas tienen estudios universitarios completos (Contador Público) y otras 2 tienen estudios de tecnicatura en diseño. Los restantes individuos son autodidactas y no tuvieron estudios formales más allá del secundario (en un solo caso ni siquiera ha completado el secundario).

Los productos desarrollados y servicios ofrecidos son muy variados. Aparentemente los ofrecimientos se definen en base a las necesidades del mercado antes que en base a los conocimientos, experiencia o preferencia de los profesionales. Ocho de los 11 emprendimientos ofrecen, entre otros productos, sistemas web de diferente complejidad. Los más complejos son una plataforma de e-publishing y otra de e-learning, y los más simples son páginas web estáticas. Luego de software web, los sistemas administrativos (adaptados a diferentes clientes) son los más ofrecidos. Sólo una empresa ofrece marketing digital y “community management”, y otra

ofrece auditorías de calidad. Cuatro empresas ofrecen servicio técnico de hardware y software, incluyendo un profesional independiente que ofrece servicio técnico de software vía internet, sin asistencia personal al lugar de trabajo.

Las tecnologías utilizadas marcan tanto la complejidad de los desarrollos como la orientación de la oferta de productos y servicios. HTML5, JavaScript, CSS3, PHP, Joomla y JQuery son las plataformas más utilizadas para desarrollo web, con MySQL, SQL y Oracle utilizadas para las bases de datos. Las tecnologías .Net se utilizan en pocos casos, y en un caso aún utilizan Visual Fox Pro. En cuanto a metodologías de desarrollo, o bien no se utilizan o se utilizan metodologías ágiles adaptadas al equipo de trabajo.

El hardware disponible es reflejo exacto del tamaño de la empresa, con los trabajadores independientes utilizando una o dos computadoras (de escritorio o portátiles) y las empresas más grandes (de 3 a 7 empleados) utilizando una o dos computadoras por empleado y uno o más servidores, usualmente Linux.

Las telecomunicaciones fueron uno de los puntos más criticados, debido a la baja calidad del servicio. Las empresas proveedoras incluyen a todas las empresas de telecomunicaciones nacionales y algunos proveedores locales, y las tecnologías incluyen a todas las disponibles en el mercado: wireless, cable y ADSL.

Respecto a la forma de atraer clientes, la mayoría confía en el "boca a boca" y las buenas referencias de sus clientes anteriores. Lo cual define un mercado casi exclusivamente local o zonal. Solamente un emprendedor independiente y una empresa tienen clientes en el extranjero. Tres empresas tienen clientes en otras provincias, incluyendo la Ciudad Autónoma de Buenos Aires. Los pagos del extranjero se reciben por giros (Western Union) y de otras provincias a veces se realizan por Mercadopago.

En cuanto a redes de contactos profesionales, solo un emprendedor participa activamente en un foro virtual. El resto de los emprendedores o empresas no participan activamente en foros (aunque tres de ellos leen los temas publicados en ciertos foros) ni participan en colegios o clústeres. La mayoría de los entrevistados no mencionó la falta de oportunidades de formación contínua como un problema. Cuando tienen dudas o necesitan aprender algún tema, generalmente realizan consultas profesionales a conocidos o buscan por respuestas publicadas en internet. Solo dos emprendedores dijeron aprovechar oportunidades de formación continua en sus localidades, uno de ellos participando en las actividades de posgrado de la UTN Facultad Regional San Francisco. Cabe destacar que un entrevistado indicó que no podía aprovechar las oportunidades de educación virtual ya que para él una de las supuestas bondades de los cursos online, esto es la flexibilidad de estudiar en el tiempo que el alumno pueda hacerlo, era una gran desventaja: él prefería tener horarios preestablecidos de clases, de preferencia presenciales.

En cuanto a acceso a financiamiento a través de planes de desarrollo gubernamentales (tales como FONSOFT o FONTAR), todos, excepto un emprendedor, conocían de su existencia, aunque sólo 3 empresas los han utilizado.

4. Conclusiones y trabajos futuros

El tamaño de los emprendimientos es el esperado, de profesional independiente a una decena de empleados, y los productos desarrollados, principalmente desarrollos web y sistemas administrativos, también son los que se suponía encontrar. Sin embargo, se halló una notable tendencia hacia el mercado local, cuando se esperaba encontrar más relación con clientes de otras provincias y países. Es necesario indagar las causas de este fenómeno, que puede tener relación con la forma de promocionar el emprendimiento y las limitaciones en cuanto al idioma inglés.

Debemos indagar por qué, a pesar de conocer la existencia de subsidios y ayudas financieras de las administraciones públicas tanto nacionales como provinciales y municipales, las PyMES no aprovechan estas herramientas en su pleno potencial.

La mayoría de los graduados universitarios o terciarios entrevistados están trabajando en la misma ciudad donde estudiaron (8 de 10). Esto indica que se detectó una relación relativamente baja de profesionales que estudiaron en universidades (ubicadas en las ciudades más grandes) y que hayan vuelto a emprender sus actividades profesionales en sus pueblos de origen, comparado con profesionales que no asistieron a ninguna universidad. Por lo tanto deberíamos indagar por qué buena parte o la mayoría de quienes van a estudiar a las ciudades no vuelven. Aún más, en una fase posterior de este proyecto podríamos encuestar a profesionales del interior que luego de estudiar en alguna universidad se quedó en esa ciudad para desarrollarse profesionalmente y relevar las razones de su decisión.

Las tecnologías utilizadas son modernas y acordes a los productos y servicios ofrecidos. Sin embargo, hay una falta de uso de metodologías de desarrollo y gestión que puede ser preocupante. Sería interesante entender por qué no se usan las metodologías propuestas y cómo adaptarlas para que puedan ser útiles a estos emprendimientos.

La principal falencia que encontramos en esta exploración previa es la casi nula participación en foros, colegios o clústeres profesionales o empresariales y la falta de posibilidades de formación contínua. Quizás haya una relación entre esta falta de comunicación y los escasos conocimientos de inglés, que impiden acceder a

la mayoría de los cursos y foros que existen en la internet.

En conclusión, este relevamiento inicial nos permitió descubrir cuáles son los puntos fuertes y cuáles los más débiles en la incipiente industria del desarrollo de software y servicios informáticos del interior de la provincia de Córdoba. La información recabada nos ha permitido reformular la encuesta de modo de poder utilizarla para indagar más profundamente en los temas que aparentan ser las mayores falencias en este ámbito.

Utilizando contactos personales y profesionales, así como los contactos provistos por el Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba, el Colegio de Ingenieros Especialistas, el Cluster Córdoba Technology, el Cluster Tecnológico Río Cuarto y el Cluster TIC de Villa María, en los próximos meses realizaremos una encuesta al mayor número posible de emprendedores y empresas de desarrollo de software y servicios informáticos del interior de la provincia de Córdoba. Los resultados de esta encuesta nos permitirán avanzar en el descubrimiento de los factores que impiden un mayor desarrollo de esta industria en la región y comenzar a diagramar una estrategia que permita superarlos.

5. Referencias

[1] Bell, D.R., "*Location Is (Still) Everything: The Surprising Influence of the Real World on How we Search, Shop, and Sell in the Virtual One*", Amazon Publishing/New Harvest, Julio de 2014.

[2] Cringley, R.X., "*Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date*", HarperBusiness, 1996.

[3] Feijen, W., van Gasteren, A.J.M., Gries, D., Misra, J., "*Beauty Is Our Business: A Birthday Salute to Edsger W. Dijkstra*", Texts and Monographs in Computer Science, Springer, 1990.

[4] INTELI, "*Creative-based Strategies in Small and Medium-sized Cities: Guidelines for Local Autorities*", INTELI - Inteligência em Inovação, Centro de Inovação, Junio de 2011.

[5] Kozul-Wright, Z., Howells, J., Hamdi, M., "*Changing Dynamics of Global Computer Software and Services Industry: Implications for Developing Countries*", Technology for Development Series, United Nations Publication, 2002.

[6] McConnell, S., "*The Art, Science, and Engineering of Software Development*", IEEE Software, Vol. 15, No. 1, Enero/Febrero de 1998.

[7] OECD, "*ICT Skills and Employment: New Competences and Jobs for a Greener and Smarter Economy"*, OECD Digital Economy Papers, No. 198, OECD Publishing, 2012.

[8] VisionMobile, "*App Economy Forecasts 2013-2016*", Julio de 2013. http://www.visionmobile.com/product/app-economy-forecasts-2013-2016

[9] VisionMobile, "*The European App Economy 2013: Creating Jobs and Driving Growth*", Septiembre de 2013. http://www.visionmobile.com/product/the-european-app-economy/

[10] VisionMobile, "*The European App Economy 2014: How the App Economy is Creating Jobs and Driving Economic Growth in the European Union*", Julio de 2014. http://www.visionmobile.com/product/european-app-economy-2014/

Anexo A

Proyecto de I+D: Relevamiento de la industria del software en el interior de la Provincia de Córdoba y definición de estrategias para su crecimiento.

Entrevista:
Orientada a desarrolladores de software, profesionales de software-informática-sistemas, empresarios de PyMEs de desarrollo de software y prestación de servicios informáticos.

Instrucciones:
Presentar el proyecto, explicar que queremos relevar la industria de desarrollo de software y servicios informáticos en el interior de la Provincia de Córdoba, y que estamos haciendo estas entrevistas para ayudarnos a entender esta industria y poder armar una encuesta que nos permita hacer el relevamiento más preciso.

Indicar explícitamente que los datos son confidenciales y que no serán publicados, solamente los resultados estadísticos.

Recuerde que es una entrevista abierta y el entrevistado puede explayarse cuanto quiera en torno a los temas que se tratan. Trate de cubrir todos los temas, pero no se limite a las preguntas propuestas para cada tema.

Identificación del entrevistado y su empresa o grupo y localización
Nombre:
Cargo:
Empresa:
Localidad:

Producto/Servicio y Mercado (resumen histórico)
¿Qué producto y/o servicio ofrecen? ¿A qué mercado apuntan (localidad, región, provincia, país, países limítrofes, exportación en general, etc.)? ¿Ha cambiado recientemente el producto o servicio?

Tecnologías utilizadas
- Entornos de desarrollo y gestión
¿Qué software utilizan para desarrollar? ¿Para gestionar los proyectos (bugs, tareas, etc.)?

- Tecnologías de comunicaciones
¿Qué servicio de telecomunicaciones/internet utilizan? ¿Con qué capacidad? ¿De cuál proveedor? ¿Está conforme? ¿La prestación es suficiente para las necesidades de su emprendimiento?

- Infraestructura tecnológica
¿Qué hardware utiliza para su emprendimiento?

Recursos Humanos (cantidad, niveles de estudio, origen)
¿Cuántas personas trabajan en su emprendimiento? ¿Cuál es la formación académica de cada uno? ¿Dónde estudiaron? Averiguar en lo posible si son nacidos y criados en la localidad, si se fueron a estudiar a otra localidad, si volvieron, etc.

Comunicación con los clientes
¿Cómo llega a los clientes potenciales? ¿Cómo se comunica con los clientes?

Contratos formales
¿Realiza contratos formales antes de realizar el trabajo? ¿Por qué? Si no los hace, ¿planea hacerlo en un futuro?

Redes y vinculación
¿Cómo se vincula con sus pares? ¿Pertenece a un clúster, colegio o foro profesional? ¿Cómo aprende nuevos conocimientos, qué posibilidades de formación continua tiene, etc.?

Financiamiento/Subsidios
¿Utiliza o ha utilizado subsidios o aportes de gobiernos (nacional, provincial, municipal, etc.)? Si sí, ¿cuáles y cuándo lo hizo? Si no, ¿por qué?

Consideraciones generales
Preguntar al entrevistado si quiere hacer algún comentario extra o aclarar algún punto de los discutidos.

Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación para la Formalización de Necesidades en Proyectos de Almacenes de Datos

Avalos, Verónica Nathalí[1] & Diván, Mario José [2, 3]
[1]*Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires*
[2]*Facultad de Ingeniería, Universidad Nacional de La Pampa*
[3]*Facultad de Ciencias Económicas y Jurídicas, Universidad Nacional de La Pampa*

Abstract
Este trabajo propone una Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación, cuyo objetivo es identificar y mantener la trazabilidad de los requerimientos en forma temprana en proyectos de almacenes de datos, a partir de la formalización de los procesos de negocios, y de la definición formal del proyecto de medición y evaluación que soportará el proceso de toma de decisión. Esto permite avanzar sobre la delimitación del alcance del proyecto, la identificación temprana de sus riesgos, a la vez que establece un mecanismo de trazabilidad entre las decisiones y artefactos que se generan a lo largo de su ciclo de vida. Esto representa un complemento importante sobre ciclos de vida como el propuesto por Kimball, en donde la fase de requerimientos no está formalizada, y no existe una estrategia por la cual se defina claramente aquellos aspectos que se desean cuantificar y-o a analizar a los efectos de soportar un proceso de toma de decisión.

Palabras Clave
Almacén de datos. EIbPREME. Procesos. Medición y Evaluación.

1. Introducción

Los proyectos de almacenes de datos permiten abordar el análisis de los datos desde diferentes perspectivas, centrado en el negocio, y bajo diferentes esquemas de granularidad. De este modo, y si bien la implementación de un almacén de datos es afectada por diversos factores tecnológicos, ello no implica que un almacén de datos deba definirse como un proyecto de corte netamente tecnológico [1,2], ya que existen aspectos no tecnológicos tales como la precisión de la información, los niveles de seguridad definidos sobre los datos, la veracidad y consistencia de la fuente de datos, entre otros que giran alrededor de la calidad de datos [3], y constituyen los cimientos de cualquier proyecto de este tipo. Así, la participación de los especialistas del negocio constituye un aspecto esencial, para poder guiar las necesidades que subyacen al proceso de toma de decisiones que cualquier estrategia de este tipo pretenda soportar.

Ahora bien, a los efectos de llevar adelante la construcción de un almacén de datos, pueden mencionarse al menos tres tendencias metodológicas [4,5]: *i) La orientada a datos, ii) La orientada a objetivos y iii) La orientada a las necesidades del usuario o requerimientos de los clientes*. El punto es que no todos los enfoques metodológicos suelen incorporar al especialista del negocio al proyecto, como en el caso de las metodologías orientada a datos [6], y por otro lado aquellos que lo incorporan, como es el caso de las orientadas a las necesidades del usuario, reconocen en la gestión de requerimientos cierto grado de complejidad que puede constituirse en una limitante del proyecto [7]. Por otro lado, aquellas orientadas a objetivos intentan cuantificar procesos de negocios, pero no explicitan cómo formalizar tales procesos, ni tampoco cómo hacer para que el proceso de medición sobre los procesos de negocios sea comunicable, repetible y comparable [4]. De este modo, la Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación (EIbPREME) surge como complemento de la fase de gestión de requerimientos planteada por Kimball [7] en su ciclo de vida del proyecto. Así, se propone que la fase de gestión de requerimientos sea especializada en tres: *a) Modelado de procesos de negocios, b) Gestión de requerimientos, y c) Formalización del proyecto de Medición y Evaluación (en adelante: M&E)*. En tal sentido, la idea es formalizar los procesos de negocios en base a un meta-modelo de procesos como SPEM (*Software & Systems Process Engineering Metamodel*) [8], lo que permitiría dotarlos de formalismo y hacerlos comunicables. Posteriormente, se avanza sobre la identificación de las entidades bajo análisis en base a ellos, permitiendo generar los requerimientos para la toma de decisiones en el marco del proyecto de almacén de datos. Finalmente, definido los procesos, identificadas las entidades bajo análisis y especificados los requerimientos, se formaliza el proceso de medición sobre un marco formal específico como C-INCAMI (*Context-Information Need, Concept model, Attribute, Metric and Indicator*) [9,10], lo que incrementaría la consistencia del proceso de medición, a la vez que permite la comparabilidad y repetitividad de las

medidas. De este modo, y continuando con el ciclo de vida de Kimball para almacenes de datos, antes de avanzar al modelado multidimensional o al diseño arquitectural, se tendría un conocimiento más consistente y completo sobre los aspectos que se pretenden cuantificar, y sobre cómo llevar adelante el proceso de M&E de forma que se garantice su repetitividad y comparabilidad a lo largo del tiempo. De este modo *EIbPREME* contribuye al ciclo de vida definido por Kimball [7], *en i) Emplea un meta-modelo de procesos que permite la formalización de los procesos de negocios, haciéndolos comunicables, extensibles y comprensibles, ii) Permite la integración natural de los interesados del negocio desde la etapa de definición de los procesos, hasta incluso en cuanto a cómo se pretende medir y evaluar cada una de las medidas, iii) Incorpora las prácticas de gestión de requerimientos de la ingeniería del software como un aspecto natural del proyecto, articulando la misma con un meta-modelo de procesos a partir del cual definirá los requerimientos, y de un marco de M&E al que abastecerá para poder definir cómo medir y evaluar las entidades bajo análisis, y iv) Permite definir el proyecto de M&E en base a la formalización de los procesos de negocios e incorporar formalismo, comunicabilidad y trazabilidad a la gestión de requerimientos, como así también a sus cambios.*

Este artículo se compone de seis secciones. La sección 2 resume el meta-modelo de procesos SPEM y el marco de medición y evaluación C-INCAMI. La sección 3 sintetiza los aspectos centrales de los principales enfoques metodológicos para la construcción de un almacén de datos. La sección 4 introduce la Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación. La sección 5, discute los trabajos relacionados, y por último, se resumen las conclusiones y trabajo a futuro.

2. Formalización de Procesos. Medición y Evaluación

La idea sintética de EIbPREME como desglose y complemento de la fase de requerimientos dentro del ciclo de vida de Kimball es esquematizada en la Figura 1 .

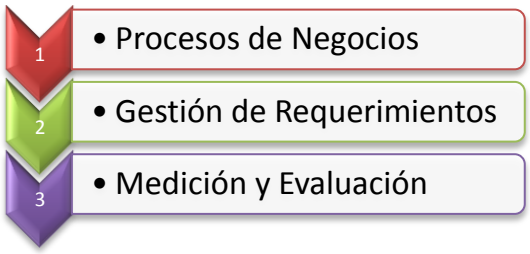


Figura 1. Síntesis de la idea asociada con EIbPREME

De este modo, EIbPREME se estructura del siguiente modo: 1) Formaliza los procesos de negocios, ya que a través de ellos se logrará formalizar el sistema de información, haciéndolos comunicables y extensibles, 2) A partir de la formalización de los procesos, la identificación de interesados, responsables y requerimientos es transparente, ya que el proceso representa el sistema de información sobre el que se pretende decidir, 3) Definidos los procesos e identificados los responsables, una definición consistente del proyecto de medición y evaluación es necesaria para garantizar la comparabilidad, consistencia y repetitividad del mismo.

Así y como puede apreciarse en la Figura 1, la comunicabilidad de los procesos de negocios requiere de un meta-modelo subyacente que permita su descripción. Luego, la medición y evaluación, a los efectos de garantizar su consistencia y perdurabilidad en el tiempo, requiere de un formalismo propio de un marco conceptual de medición y evaluación. De este modo, el meta-modelo de Ingeniería de Procesos de Software y Sistemas, conocido por sus siglas en inglés SPEM [8], es una especificación de la OMG (acrónimo en inglés para Object Management Group) que permite formalizar los procesos de software. Por otro lado y a los efectos de poder garantizar la repetitividad, comparabilidad y consistencia del proceso de medición de los procesos, un marco formal de M&E es requerido. En tal sentido, C-INCAMI [9,10] se basa en un enfoque en el cual la especificación de requerimientos, la medición y evaluación de entidades y la posterior interpretación de los resultados están orientadas a satisfacer una necesidad de información particular.

Así, la presente sección se divide en dos sub-secciones. En la primera sub-sección se da un panorama de SPEM, como herramienta para la formalización de procesos; mientras que en la segunda sub-sección, se brinda un síntesis de C-INCAMI, como fundamento para la definición de un proyecto de M&E.

2.1 Panorama de SPEM

El meta-modelo SPEM [8], es una especificación de OMG que permite formalizar los procesos de software. En tal sentido, se entiende por proceso de software a un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software [11]. SPEM está basado en un meta meta-modelo de la OMG conocido como Meta Object Facility [12], referenciado comúnmente a través de su abreviatura en inglés MOF. El meta-modelo MOF es una especificación que define cuatro niveles conceptuales: el primer nivel, o nivel 0 (más detallado), se asocia con el proceso representado o instanciado, el nivel 1 se vincula con el modelo del proceso instanciado, el nivel 2 se

asocia con el meta-modelo del proceso instanciado (aquí se ubicaría SPEM), mientras que el nivel 3, corresponde a un meta meta-modelo y es allí donde se emplea la capacidad representacional de MOF. De este modo, SPEM sería para los procesos lo mismo que UML (Unified Model Language) [13] es para los sistemas de software, ya que UML es un meta-modelo que sirve para representar diferentes modelos de sistemas informáticos, mientras que SPEM es una meta-modelo capaz de representar modelos de procesos de software. Así, entre los diferentes niveles existen relaciones de instanciación, de manera que un elemento de un nivel N-1 es una instancia de un elemento del nivel superior N, por ejemplo RUP (Rational Unified Process) [14] del nivel M1, es una instancia de SPEM del nivel M2.

En términos de SPEM versión 2 (en adelante, SPEM 2), la representación de los procesos se sustenta sobre tres elementos básicos, a saber: *a) el rol, b) el producto de trabajo y c) la tarea* (Ver Figura 2).

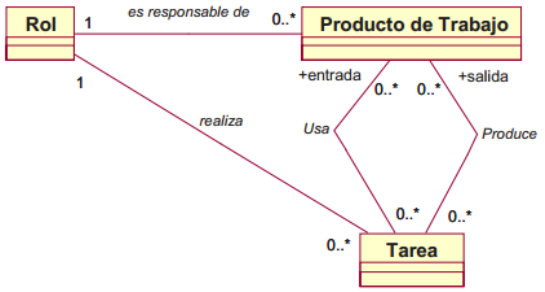


Figura 2. Elementos básicos de SPEM 2 [11]

En este sentido, las tareas representan el esfuerzo a realizar, los roles se asocian con el perfil de quien hace la tarea, y finalmente, los productos de trabajo se vinculan con las entradas que se utilizan en las tareas, como así también con las salidas que se producen mediante las mismas.

Ahora bien, la notación de procesos de negocios, conocida por sus siglas en inglés BPMN (*Business Process Model and Notation*) [15] también representa una alternativa para el modelado de los procesos de negocios, aquí se ha optado por SPEM ya que SPEM posee mayor cobertura semántica que BPMN en relación a la ontología de procesos definida en [16].

2.2 Panorama de C-INCAMI

C-INCAMI es un marco conceptual [10,9] que define los módulos, conceptos y relaciones que intervienen en el área de M&E. Se basa en un enfoque en el cual la especificación de requerimientos, la medición y evaluación de entidades y la posterior interpretación de los resultados están orientadas a satisfacer una necesidad de información particular. Está integrado por los siguientes componentes principales: 1) Gestión de Proyectos de M&E; 2) Especificación de Requerimientos no Funcionales; 3) Especificación del Contexto del Proyecto; 4) Diseño y Ejecución de la Medición: y 5) Diseño y Ejecución de la Evaluación. En adelante se resaltará el texto con itálica cuando se mencionen términos y atributos que forman parte de estos componentes. La mayoría de los componentes están soportados por los términos ontológicos definidos en [9]. En la Figura 3 se muestra un diagrama con los principales conceptos y relaciones para los componentes de requerimientos, contexto y medición. Los conceptos de la Figura 3, definen la estructura de los metadatos que permiten llevar adelante el proceso de M&E de una o más entidades bajo análisis.

El componente *Definición y Especificación de Requerimientos no Funcionales* (paquete *requirements en* Figura 3), permite especificar de forma concisa la necesidad de información (Information Need) de un proyecto de medición y evaluación. Generalmente surge de un objetivo específico a nivel de proyecto u organización. La necesidad de información describe el propósito (*purpose*) y el punto de vista de usuario (*userViewpoint*). A su vez, identifica un concepto calculable (*CalculableConcept*), y especifica la categoría de las entidades a evaluar (*EntityCategory*). Un concepto calculable puede ser definido como una relación abstracta entre atributos de un ente y una necesidad de información; en efecto, la calidad externa, costo, etc., son instancias de un concepto calculable. Un concepto calculable puede ser representado por un modelo (*ConceptModel*). En C-INCAMI, la calidad es medida y evaluada mediante la cuantificación de conceptos de menor nivel de abstracción como son los atributos (*Attribute*) de un ente.

El componente *Especificación del Contexto del Proyecto* (paquete *context en* Figura 3), permite describir de forma estructurada el contexto relevante (*Context*) para una necesidad de información por medio de propiedades (*ContextProperty*) (ver más detalle sobre contexto en [10]).

El componente *Diseño y Ejecución de la Medición* (paquete *measurement en* Figura 3), permite especificar las métricas utilizadas en la medición y registrar los valores medidos de los atributos de cada ente. Un atributo puede ser cuantificado por muchas métricas, pero, en un proyecto de medición específico, sólo se utiliza una métrica. Una métrica (*Metric*) define el método de medición o cálculo (*Measurement Method* o *Calculation Method*) para obtener el valor del atributo, y la escala (*Scale*) de los valores. Un método de medición se aplica a una métrica directa (*Direct Metric*); por el contrario, un método de cálculo (en el cual interviene una fórmula) se aplica a una métrica indirecta (*Indirect Metric*). Una vez que las métricas fueron seleccionadas, se utiliza su definición para realizar la

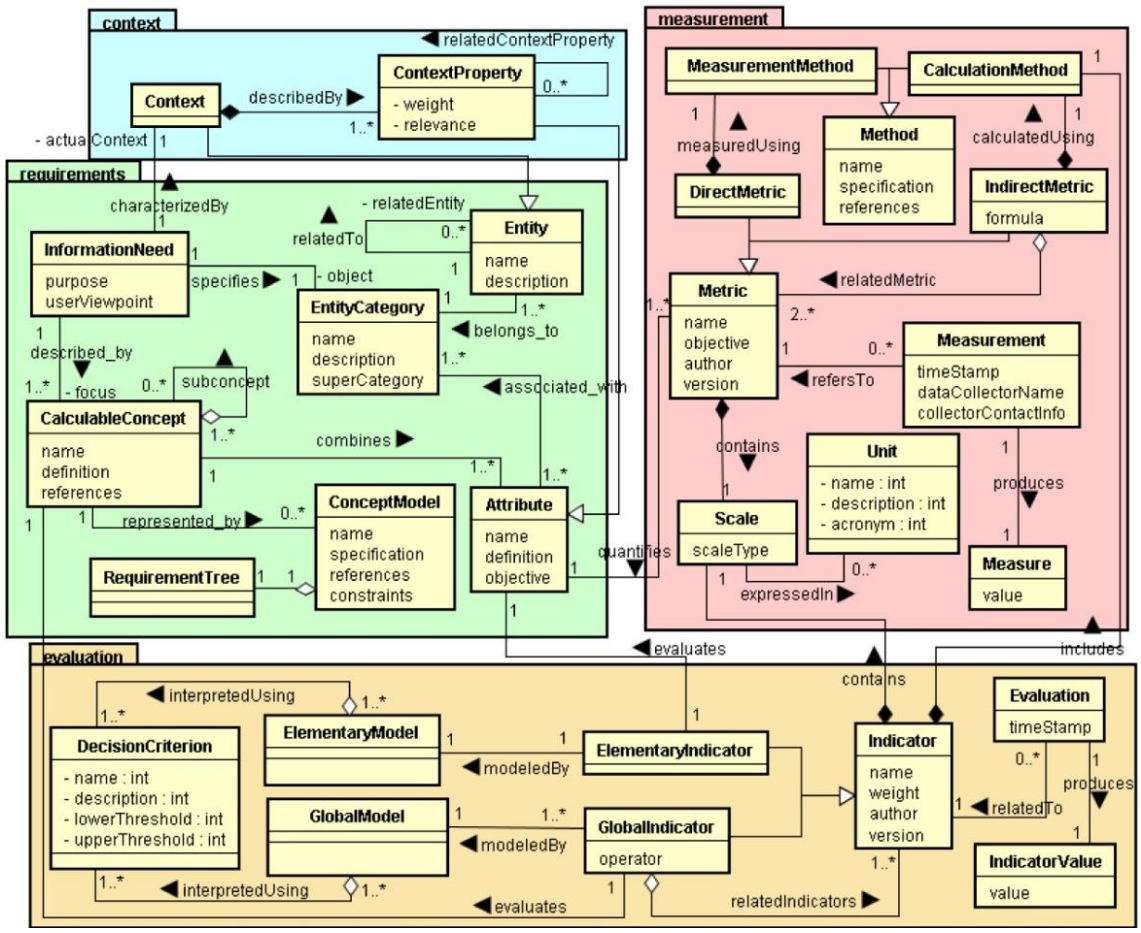


Figura 3. ***Principales conceptos y relaciones de los componentes Especificación de Requerimientos no Funcionales, Especificación del Contexto y de la Medición.***

medición (*Measurement*) y así producir una medida (*Measure*) para cada atributo. Sin embargo, el valor obtenido por una métrica particular no representa el nivel de satisfacción de un requerimiento elemental (atributo) por lo que se necesita realizar una nueva correspondencia utilizando indicadores. El componente *Diseño y Ejecución de la Evaluación* (paquete *evaluation en* Figura 3), permite especificar indicadores, que son la base para la interpretación de atributos y conceptos calculables. Hay dos tipos de indicadores: elementales y globales. Particularmente, definimos un indicador elemental (*Elementary Indicator*) como un indicador que no depende de otros indicadores para evaluar o estimar un concepto de más bajo nivel de abstracción como son los atributos. Por otro lado, un indicador parcial o global (*Global Indicator*) es un indicador que es derivado de otros indicadores para evaluar o estimar un concepto de alto nivel de abstracción (como son los subconceptos y conceptos calculables, por ej. de un modelo de calidad). El valor del indicador global finalmente representa el grado de satisfacción global de los requerimientos para dicha necesidad de información.

De este modo, en EIbPREME a partir de los procesos formalizados, es posible identificar los sistemas de información sobre los que se pretende decidir. A partir de ellos, es posible especificar cada uno de los requerimientos en base a un contexto con alcance limitado. Finalmente, por cada requerimiento que tenga asociada una necesidad de toma de decisión, es posible formalizar desde las métricas que se aplicarán, cómo se medirá, cómo se definen los indicadores y hasta cómo deben ser interpretados, entre otras cuestiones propias del proceso de M&E.

3. Principales enfoques metodológicos para la construcción de un almacén de datos

El ciclo de vida de un proyecto de Data Warehouse está dado por las siguientes fases: *Inicio, Planificación, Ejecución y Cierre* [17]. Dichas fases contemplan la adopción de una metodología cuyo principal objetivo es definir una serie de etapas que permitan planificar y controlar el proceso de implementación de un sistema de información [18]. Actualmente si bien no existe una metodología común para la implementación de un Data Warehouse [19], las existentes se han desarrollado principalmente en tres tendencias: i) Orientada a datos ii) Orientada a objetivos iii) Orientada a las necesidades del usuario o requerimientos [4].

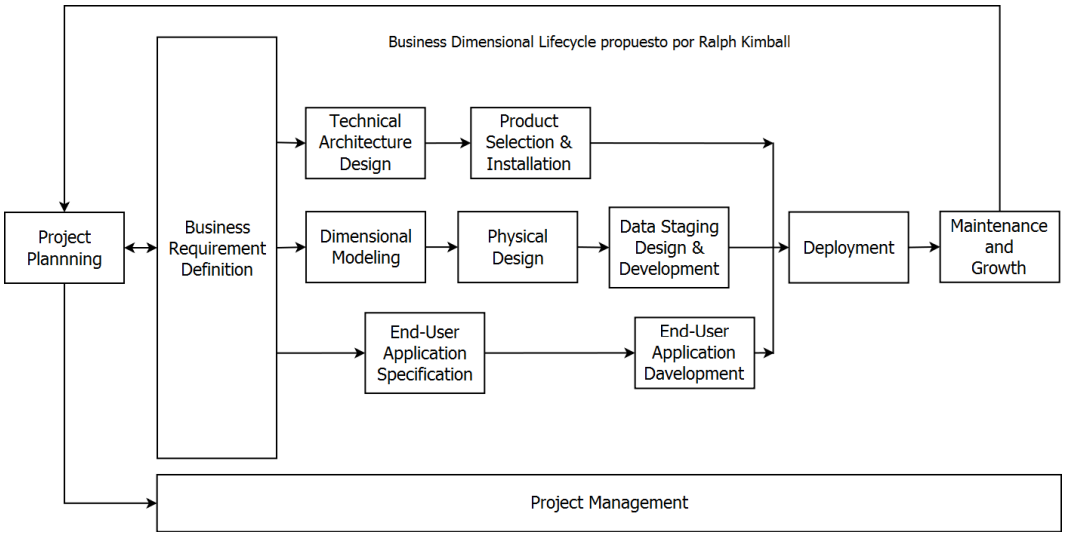


Figura 4. Ciclo de vida de un almacén de datos según Kimball & Ross [7]

Por otro lado, empresas como Teradata, Oracle, IBM, Sybase, SAP, Informatica, entre otras, dedicadas a la implementación de proyectos de Data Warehouse, han desarrollado aspectos particulares al ciclo de vida del proyecto [20]. De este modo, aquí se presenta una síntesis del estudio comparativo desde dichas empresas, con el fin de identificar diversos aspectos plausibles de mejora. Esta sección se subdivide en 4. La sección 3.1, resume la idea del enfoque orientado a datos para la construcción de almacenes. La sección 3.2, introduce la construcción del almacén desde la perspectiva de las necesidades del usuario. La sección 3.3, presenta la idea de construcción de almacenes de datos basados en objetivos; y la sección 3.4, resume un estudio comparativo sobre diferentes metodologías para la construcción de almacenes de datos.

3.1 Enfoque basado en los datos

La idea de este tipo de abordajes fue planteada originalmente por Bill Inmon [6]. Se trata de un enfoque basado en datos para el diseño del almacén de datos, el cual considera que: *a) Los analistas de gestión de la información de una empresa, con frecuencia entienden sus necesidades, y los datos que disponen, b) Los modelos tradicionales de desarrollo de software no son aplicables a este tipo de soluciones, y c) Los requerimientos o necesidades de información es la última fase a ser considerada dentro del ciclo de vida de un almacén de datos.* En tal sentido, Inmon plantea que el desarrollo de los sistemas operacionales es gestado alrededor de un ciclo de vida que inicia con los requerimientos y termina con el código, mientras que el procesamiento para el soporte a la toma de decisiones, es iniciado con los datos y es culminado con los requerimientos. Así, Inmon introduce el término *Corporate Information Factory* (*CIF*), como interpretación conceptual del ecosistema de repositorios consistentes que debe soportar el proceso de toma de decisiones. De este modo, se proponen una secuencia de fases para el desarrollo de un almacén de datos, a saber: *a) Desarrollo de sistemas operacionales:* A partir de los datos, se siguen las actividades clásicas de desarrollo de sistemas operacionales basado en ciclos de vida en cascada para esquematizar el sistema de información, desarrollándose actividades tales como la identificación de requerimientos, determinación y fases del proyecto, especificaciones de procesos mediante diagrama de flujos de datos (DFD), entre otras, *b)Desarrollo del almacén de datos:* Se inicia con el análisis del modelo de datos, sigue con la valoración técnica, la preparación del entorno técnico, el análisis de sistemas fuente, y el diseño del almacén de datos. En tal sentido, Inmon aclara que esta fase puede ser repetitiva siempre que se apliquen los pasos del Desarrollo de Sistemas Operacionales. *C) Procesamiento heurístico: Su* objetivo reside en describir el uso del almacén de datos para la toma de decisiones y el análisis de su información. Es decir, sugiere actividades para la determinación de los datos que se requieren analizar, recomienda la construcción de aplicaciones para la utilización de la información desde el almacén de datos y adicionalmente, propone una actividad de institucionalización de la información mediante la elaboración de reportes de forma repetitiva.

3.2 Enfoque basado en las necesidades del usuario

Este enfoque pretende delimitar en forma temprana el alcance de un proyecto de almacén de datos en base a la identificación de requerimientos. En la Figura 4 se ilustra el enfoque del ciclo de vida según Kimball [7], en donde la etapa de gestión del proyecto y requerimientos tiene como objetivo principal, delimitar el alcance y especificar los requerimientos del cliente en función de sus necesidades por lo que la especificación de

requerimientos y la información son la base de todo almacén de datos. Sin embargo, se reconoce que esta tarea de gestionar los requerimientos no es sencilla o natural dentro de un proceso de desarrollo de almacenes de datos. De este modo y en contraposición a Inmon, Kimball sostiene que intentar desarrollar un proyecto de almacén de datos desconociendo los requerimientos y expectativas del usuario en forma temprana (Ver Figura 4), representan un riesgo elevado y podría conducir al fracaso del proyecto [7].
Mitigar riesgos asociados con requerimientos no es algo nuevo en Ingeniería del Software [21,22,23], y Kimball lo intuye en forma temprana dentro de su ciclo de vida en contraposición al enfoque de Inmon. No obstante, el ciclo de vida de Kimball carece de documentación específica y el formalismo de la gestión de requerimientos, a los efectos de poder garantizar la comunicabilidad de los procesos sobre los que se quiere decidir, o bien, sobre la cuantificación de las métricas, indicadores y criterios de decisión que se emplearán para soportar el proceso de toma de decisión en sí.
Las siguientes etapas del ciclo de vida propuesto por Kimball se refieren al diseño (modelado dimensional) y arquitectura de la solución. La arquitectura y diseño de un almacén de datos deben reflejar los requisitos de negocios de la empresa, mientras que el modelo de datos, la estructura, los componentes y los metadatos deben basarse en las necesidades del negocio. De este modo, se propone construir un modelo dimensional a partir de un modelo conceptual gráfico soportado en una metodología semi-automática, que transforme un esquema entidad relación operacional a un modelo dimensional, dicha metodología es conocida como Hecho - Dimensión [7]. No obstante, dicha metodología no garantiza que el esquema entidad relación sea lo suficientemente robusto para cubrir todas las necesidades de negocio, por cuanto las entidades y relaciones cubiertas en un esquema entidad relación transaccional, están orientadas a cubrir en su mayoría requerimientos operativos. Una vez finalizada la definición de la arquitectura y el diseño de la solución, las siguientes fases del ciclo de vida abordan aspectos tecnológicos tales como: la elección e instalación del producto, la especificación de la aplicación analítica, su implementación y mantenimiento.

3.3 Enfoque basado en objetivos

Este enfoque se centra en la construcción de un almacén de datos a partir de los procesos de negocios, con el objetivo de poder medirlos y evaluarlos [4]. En forma similar al enfoque basado en las necesidades del usuario, se plantea una etapa temprana de gestión de requerimientos. Así, este enfoque lleva adelante las siguientes actividades [4]: *a) Identificar al patrocinador del proceso de negocio, b) Establecer metas u objetivos asociados al proceso de negocio por cada patrocinador, c) Seleccionar metas y objetivos que se consideran relevantes para la organización, y d) Cada meta es medida por al menos un indicador, los indicadores apuntan a la consecución de objetivos a largo plazo*. A diferencia de los enfoques introducidos anteriormente, esta perspectiva incorpora la idea de indicadores y procesos de negocio como elementos necesarios para la construcción de un almacén de datos.

3.4 Síntesis de un estudio comparativo sobre metodologías para la construcción de almacenes de datos

Aquí se resume un estudio comparativo basado en quince metodologías diferentes [20], empleadas por empresas dedicadas a la construcción de almacenes de datos a los efectos de aportar una perspectiva pragmática de la problemática. En el estudio se analizan características tales como la gestión de requerimientos, el ciclo de vida del proyecto, el diseño, la arquitectura y la gestión de cambios. Como aspecto destacable del estudio respecto a la gestión de requerimientos, se identifica que el 93% de las empresas utilizan técnicas estándar como las entrevistas y los foros para la obtención de requerimientos de negocio. Se afirma en el estudio que las metodologías evaluadas, hacen énfasis en la identificación, captura e interpretación de las necesidades de negocio.
En referencia a los tipos de ciclo de vida comúnmente utilizados en proyectos de construcción de almacenes de datos, se afirma que en su mayoría, las empresas han adoptado el desarrollo iterativo e incremental (prototipado) debido a que les permite mostrar resultados a corto plazo.
Por otro lado y respecto a la gestión del cambio en materia de requerimientos, ninguna de las metodologías de las analizadas en el estudio, permite dilucidar un abordaje detallado de esta problemática. Los cambios que se producen en las empresas con respecto a los procesos de negocio, la información, las nuevas tecnologías, el replanteamiento de objetivos y las metas de negocio, se producen de forma natural y permanente por lo cual es necesario contar con un esquema de trazabilidad que permita evaluar el impacto del cambio en el producto, de lo contrario se tornarían en un riesgo del proyecto por su impacto potencial en el alcance y su presupuesto.

4. Estrategia Integrada basada en Procesos, Requerimientos, Medición y Evaluación

El propósito de EibPREME consiste en guiar la construcción de un almacén de datos a través de una estrategia basada en procesos comunicables, con

énfasis en la gestión de requerimientos, que permitan formalizar el proceso de medición y evaluación a los efectos de promover la consistencia, repetitividad y comparabilidad de las medidas.

En la primera fase de EIbPREME, denominada "*Constituir el Proyecto de Almacén de Datos" (Código de identificación Eib-F1)*, se instaura el equipo del proyecto (integrado por un patrocinador, el equipo de especialistas de negocio y los especialistas de información) y se efectúa una aproximación inicial de las expectativas informales con respecto a la toma de decisiones, con el objetivo de delimitar el alcance inicial del almacén de datos. Adicionalmente se identifican los usuarios que participarán en el proyecto, se determinan sus perfiles y responsabilidades. Luego, con el objetivo de formalizar los acuerdos y lineamientos que regirán el Proyecto de Construcción del Almacén de Datos, se elabora el "*Acta de Constitución del Proyecto (ACP)"* y se da lugar la reunión de arranque del Proyecto (en inglés, Kick-off) con los involucrados para dar por finalizada la fase inicial. A seguir y como puede apreciarse en la Figura 5, el equipo del proyecto es el responsable de "*Identificar las Áreas de Negocio"* a fin de circunscribir el alcance en la organización.

EIbPREME

Presentation Name	Index	Pre...
EIbPREME	0	
Eib-F1: Constituir el Proyecto de Almacén de Datos	1	
Identificar al Equipo del Proyecto	2	
Constitución del Equipo del Proyecto	10	9
Recolectar expectativas informales	11	10
Formalizar Acta de Constitución de Proyecto	16	11
Reunión de Kick-off del Proyecto de Almacén de Datos	17	2,1
Identificar Áreas de Negocio	18	17
Identificar Áreas de Negocio	19	
Identificar Misión, Visión y Objetivos por Área de Negocio	20	
Eib-F2: Delimitar el Contexto de Negocio y sus Necesidades	21	18,66
Formalizar Proceso de Negocio y Necesidades de Decisión [1..n]	22	
Aprobación de diagrama de proceso de negocio en SPEM	28	
Identificar Necesidades por Proceso [1..n]	29	28
Aprobación de documentos de necesidades de decisión del proceso	34	
Unificar Necesidades de Decisión por Proceso [1..n]	35	34
Certificación de Matriz Necesidad-Proceso-Solicitante	40	
Especificación de Requerimientos	41	
Certificar Matriz de Requerimientos, Fuentes e Interesados	49	41
Eib-F3: Medir, Evaluar y Visualizar	50	
Definir el Proyecto de Medición y Evaluación utilizando GOCAME	51	
Certificar Proyecto de Medición y Evaluación	59	
Elaborar el Prototipo de la Visualización [1..n]	60	59
Certificar Prototipo de Visualización	65	
Eib-F4: Gestionar el Cambio de Requerimientos	66	
Evaluar Control de Cambios	67	73
Pre-Aprobación del Cambio	71	
Pre-Aprobación	72	71
Recomendar Cambios	73	
Evaluar el impacto del Cambio	74	72
Aprobar el Control de Cambios	77	
Aprobación Control de Cambios	78	77
Rechazar el Control de Cambios	79	
Verificación de las condiciones necesarias para pre-aprobar	80	67
Elaborar informe de Control de Cambios	81	79,78
Certificar Gestión de Requerimientos	82	

Figura 5. Estructura de Trabajo Desglosada de EIbPREME con notación SPEM (Salida del software Eclipse Process Framework Composer)

La Figura 5 expone la Estructura de Trabajo Desglosada para la estrategia EIbPREME con notación SPEM 2. De este modo, los productos de esta actividad, se constituyen en una "Lista de Chequeo de Áreas de Negocio" y una "Matriz de Interesados, Poder e Influencia por Área de Negocio" que son el principio de la fase para "Delimitar el contexto de Negocios y sus Necesidades". Luego, y a partir del "Matriz de Interesados, Poder e Influencia", por cada área de negocio detectada se inicia la fase de "*Delimitar el Contexto de Negocio y Necesidades" (Ver* Figura 6, *Código de identificación Eib-F2)*, tomando como referencia las directrices establecidas en el Acta de Constitución del Proyecto (en adelante, ACP) producto de la fase "*Constituir el Proyecto de Almacén de Datos" (Ver* Figura 5, *Eib-F1).* El equipo del proyecto es el responsable de gestionar las tareas y actividades que delimitarán con mayor precisión las necesidades y el contexto del negocio del proyecto. Dentro de esta fase se propone "*Formalizar los Procesos de Negocio", lo* que permite encaminar la elaboración del documento de necesidades de decisión por proceso de negocio (en adelante, NEDEP). Dicha fase posee diversas componentes iterativas, y por ende contará con tantas iteraciones como áreas y procesos de negocio se definan en el contexto del proyecto especificado en la *"Lista de Chequeo de Áreas de Negocio"*. Los responsables del negocio y ejecutores de cada proceso unificarán y acordarán las necesidades de decisión establecidas para cada proceso de negocio, a partir de la elaboración de la matriz entre el origen de las necesidades de información, el proceso de negocio y el solicitante (en adelante, NEPS). Esto último, permitirá facilitar la formalización de los procesos de negocio mediante SPEM, haciéndolos comunicables y extensibles. A partir de la formalización de procesos de negocio mediante el meta-modelo SPEM y sus necesidades asociadas, se da lugar a la actividad de "*Especificación de Requerimientos"*. La matriz entre el origen de las necesidades de información, el proceso de negocio y el solicitante, permite orientar las definiciones funcionales y no funcionales, facilitando la recolección de requisitos, y acotando de este modo el alcance de las necesidades del proceso y del negocio. Es parte de la fase "*Delimitar Contexto de Negocios y Necesidades*" identificar los diferentes patrones de uso de la información, definir los criterios de aceptación e identificar las fuentes de información por cada proceso. La delimitación del contexto de negocio y sus necesidades es iterativa por cuanto se va dando para cada proceso, y a su vez, cada proceso va retroalimentando a los anteriores, con lo que el esquema tenderá a converger a un conjunto de criterios de aceptación y fuentes en común. Al finalizar la fase, el resultado es una matriz de requerimientos, fuentes e interesados (en adelante, MFI), anteriormente no definida en las estrategias de construcción de almacenes de datos analizadas. Dicho artefacto, hace más consistente la trazabilidad bidireccional entre requerimientos, fuentes de información e interesados, y adicionalmente, permitiría garantizar que todos los

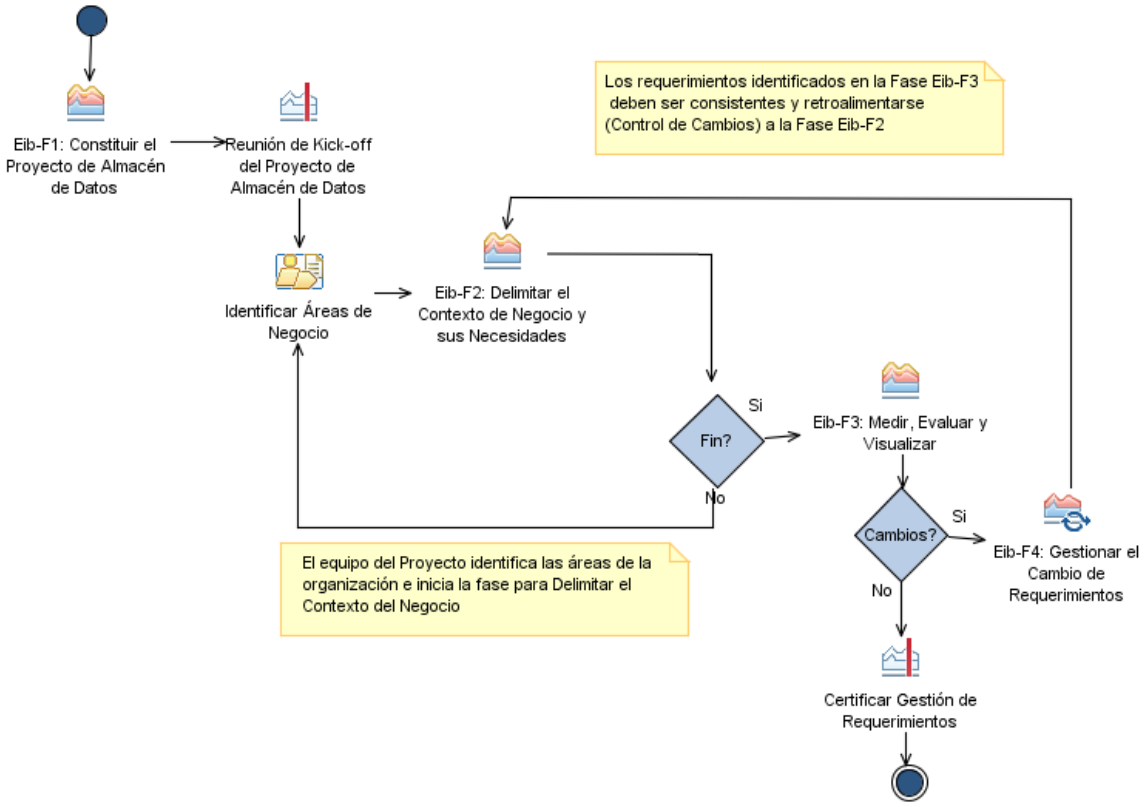


Figura 6. Diagrama de actividad global con notación SPEM 2 de EIbPREME

requerimientos sean tratados y evaluados desde el nivel más alto de granularidad (necesidad), hasta el nivel más bajo de detalle (solución técnica o implementación). Por otro lado, la matriz permite al equipo del proyecto evaluar el impacto de cada cambio en las fases posteriores, identificar las posibles inconsistencias o la ambigüedad entre los requisitos, y determinar las fuentes de información e interesados permitiendo aplicar acciones correctivas y puntos de control a tiempo.

Culminada la fase Eib-02, se realiza la *"Certificación de contexto de negocio y necesidades"* hito que tiene como objetivo garantizar la consistencia de inicio a fin de los artefactos generados durante la fase. Si eventualmente se certifica, se avanza a la fase "*Medir, Evaluar y Visualizar" (Ver* Figura 6, *Código de identificación Eib-F3)*, caso contrario se vuelve a efectuar la identificación de las áreas de negocio.

Luego, una vez certificada la delimitación del contexto de negocio y sus necesidades, se propone la fase "*Medir, Evaluar y Visualizar"*, que inicia con la *Formalización del Proyecto de Medición y Evaluación* utilizando la estrategia GOCAME [24]. Los artefactos generados en la fase anterior, como por ejemplo el Documento de Especificación de Requerimientos y la Matriz MFI, son utilizados como entrada para identificar la necesidad de información, definir la entidad bajo análisis, definir los requerimientos, determinar el modelo conceptual a utilizar, definir los atributos de la entidad bajo análisis a cuantificar mediante las métricas, delimitar el contexto, y definir las métricas e indicadores. Inicialmente, GOCAME propone definir e implementar las métricas, que en nuestro caso se analizarán y estarán guiadas en base a la matriz MFI. Luego, se diseñará e implementará la evaluación según propone GOCAME, para posteriormente ejecutar la tarea de análisis y recomendación correspondientes en base a las medidas y los indicadores definidos. Finalmente, el equipo del proyecto elabora el "*Documento de Proyecto de Medición y Evaluación*" con el objetivo de formalizar el esquema de medición y evaluación en base a las necesidades del negocio, lo que hace más consistente, comparable, y repetible al proceso de medición asociado. Adicionalmente y en esta fase, se propone "*Elaborar el Prototipo de la Visualización"* cuyo objetivo es modelar la visualización de indicadores y métricas para facilitar su lectura y consumo para la toma de decisiones. Con este fin, además se plantea la posibilidad temprana de elaborar el diseño básico del prototipo, construir un prototipo analítico visible al usuario, contrastar la visualización con el usuario final en términos de expectativas, ajustando el prototipo hasta tanto satisfaga sus requerimientos, lo que permitiría minimizar riesgos en términos de lo que el usuario espera, necesita y desea. Por este último motivo, es que dicha fase es entendida como una fase iterativa hasta que el producto (el prototipo) cumpla con los requerimientos que espera el usuario.

El equipo del proyecto a lo largo de las distintas fases será responsable de validar la consistencia de los requerimientos, lo cual dará lugar a fases iterativas de *Gestión de Cambios de Requerimientos (Ver* Figura 6, *Código de identificación Eib-F4)*, cuyos resultados deben retroalimentar la fase *Delimitar el Contexto de Negocio y Necesidades (Ver* Figura 6, *Código de identificación Eib-F2)* y ajustar los artefactos que se generan como son la Matriz NEPS (Necesidades, Procesos y Solicitante) y MFI (Fuentes - Interesados), entre otros.
Una vez que los requerimientos han sido consensuados, la información expuesta en los prototipos, y los prototipos han sido validados por el usuario final, se procede a dar por certificada la gestión de requerimientos, permitiendo continuar con la Fase II propuesta por Kimball [7] en relación a la arquitectura y el diseño multidimensional. En tal sentido, debe remarcarse que la estrategia EibPREME es complementaria al ciclo de vida propuesto por Kimball y tiene por objeto especializar su fase de "Gestión de Requerimientos" basado en procesos, requerimientos y un marco de Medición y Evaluación.

5. Trabajos Relacionados

En la sección 3 se presentaron los principales enfoques en cuanto a metodologías o estrategias de construcción de almacenes de datos. Por un lado, el enfoque orientado a datos [6] emplea DFD para representar flujos de información lo que dificulta la semántica de los procesos, los conceptos tales como responsables, tareas, actividades o fases no son discriminados lo que hace complejo el análisis de granularidad, el DFD dificulta la comunicabilidad del proceso que intenta representar. Adicionalmente, nada se plantea sobre M&E. Nuestra propuesta, especializa la fase de gestión de requerimientos de Kimball, sustentándose en un meta-modelo de procesos conocido y comunicable como SPEM, a la vez que emplea un marco de M&E como C-INCAMI para hacer consistente, comparable y repetible el proceso de medición.
El enfoque orientado a objetivos [4], si bien se orienta a los requerimientos del usuario e incorpora la idea de indicadores, no establece un mecanismo formal que permita comunicar sus procesos, ni tampoco un marco de M&E que permita hacer comparables los resultados de las medidas, tal y como plantea EIbPREME.
Con respecto al enfoque basado en las necesidades del usuario planteado por Kimball, la fase de gestión de requerimientos es guiada por entrevistas u otros esquemas no formales, que dificultan la comunicabilidad de los procesos, a la vez que nada indican sobre cómo soportarán el proceso de toma de decisión en términos de medidas e indicadores. EIbPREME especializa la fase de gestión de requerimientos, mediante el empleo de SPEM como meta-modelo de procesos, lo que permite generar matrices de trazabilidad para con los requerimientos, y finalmente, permite constituir formalmente un proyecto de M&E sustentado en un marco específico como C-INCAMI para garantizar la consistencia, comparabilidad y repetitividad del proceso de medición.
Por otro lado, Goede ha planteado anteriormente como alternativa para encarar el desarrollo de proyectos de almacenes de datos, el empleo de metodologías de desarrollo de software [19] en forma complementaria al ciclo de vida de Kimball. En tal sentido, el objetivo es guiar a los usuarios involucrados e interesados en general, a los efectos de garantizar la comprensibilidad, alcance, patrocinio y aceptación del proyecto. Nuestra propuesta se diferencia en que establece a priori un orden y priorización basado en el modelado del sistema de información a partir de sus procesos de negocios, empleando para ello un meta-modelo formal y específico como SPEM. Posteriormente determina las necesidades en relación al sistema de información modelado mediante sus procesos, generando un esquema de trazabilidad con respecto al origen, a la vez que constituye la entrada de la definición del proyecto de medición y evaluación. Adicionalmente, nuestra propuesta permite formalizar desde la etapa de procesos, hasta la medición y evaluación, lo que aporta consistencia, comunicabilidad y extensibilidad en los procesos, a la vez que se garantiza la consistencia, repetitividad y comparabilidad sobre las medidas a partir del marco formal C-INCAMI empleado a tal fin.

6. Conclusiones y Trabajo a futuro

El presente artículo ha introducido *EIbPREME* como estrategia complementaria del ciclo de vida definido por Kimball, con el objetivo de especializar la fase de gestión de requerimientos. Dicha especialización se sustenta en el modelado de procesos, a partir de los cuales se definen y monitorean los requerimientos, para finalmente definir la estrategia de M&E a partir de un marco especializado como C-INCAMI. De este modo, EIbPREME ha planteado el modelado de procesos mediante SPEM como punto de partida, haciendo los procesos comunicables, extensibles y comprensibles, a la vez que promueve la integración natural de los interesados del negocio desde la etapa de definición de procesos, hasta incluso en cuanto a cómo se pretende medir y evaluar cada una de las medidas. Por otro lado se incorporan las prácticas de gestión de requerimientos de la ingeniería del software como un aspecto natural del proyecto, articulando la misma con un meta-modelo de procesos a partir del cual definirá los requerimientos, y de un marco de M&E al que abastecerá para poder definir cómo medir y evaluar las entidades bajo análisis junto con un esquema de prototipado visual.

Como trabajo a futuro, se está trabajando en documentar un caso de estudio completo en base a una aplicación actual de EIbPREME en el Ministerio de Educación de Ecuador. Ello permitirá contar con una retroalimentación importante de la estrategia en la práctica.

7. Agradecimientos

Este trabajo forma parte del plan de Tesis de la Maestría en Explotación de Datos y Descubrimiento del Conocimiento (Universidad de Buenos Aires). Adicionalmente, está relacionado con el proyecto acreditado por resolución 066/2012 de la Facultad de Ciencias Económicas y Jurídicas (UNLPam), y con el proyecto PICTO-2011-0277 (FONCYT, Resolución 452/12) de la Facultad de Ingeniería.

8. Referencias

[1] Ruhul, A., Taslim, A.: The Empirical Study on the Factors Affecting Data Warehousing Success., 138-142 (2010)

[2] Yeoh, W., Koronios, A.: Critical Success Factors For Business Intelligence Systems., 23-32 (2010)

[3] Lucas, A.: Corporate Data Quality Management. In : International Conference on Management and Service Science, pp.1-6 (2011)

[4] List, B., Bruckner, R., Machaczek, K., Schiefer, J.: A Comparison of Data Warehouse Development Methodologies Case Study of the Process Warehouse., 203-215 (2002)

[5] Sen, A.: A Model of Data Warehousing Process Maturity., 336-353 (2012)

[6] Inmon, W.: Building the Data Warehouse 4th edn. John Wiley & Sons, Inc, Indianapolis. Canadá (2005)

[7] Kimball, R., Ross, M.: The Data Warehouse Toolkit 2nd edn. John Wiley & Sons, Inc., Toronto. Canada (2008)

[8] SPEM: Software Process Engineering Meta-Model Specification., Object Management Group (OMG) (2008)

[9] Olsina, L., Papa, F., Molina, H.: How to Measure and Evaluate Web Applications in a Consistent Way. In Rossi, G., Pastor, O., Schwabe, D., Olsina, L., eds. : Ch. 13 in Web Engineering. Springer (2007) 385–420

[10] Molina, H., Olsina, L.: Towards the Support of Contextual Information to a Measurement and Evaluation Framework. In : International Conference on the Quality of Information and Communications Technology (QUATIC), Lisboa (Portugal), p.154–163 (2007)

[11] Ruiz, F., Verdugo, J.: Guía de Uso de SPEM 2 con EPF Composer., Castilla, La Mancha (2008)

[12] Object Management Group, Inc. - OMG : Meta Object Facility. Versión 2.4.1. OMG (2011)

[13] Object Management Group, Inc - OMG: Information Technology -- Object Management Group Unified Model Language (OMG-UML) ISO 19505 1st edn. International Organization for Standardization (ISO) (2012)

[14] Ambler, S., Nalbone, J., Vizdos, M.: Enterprise unified process, the: extending the rational unified process. Prentice Hall Press, Upper Saddle River, NJ, USA (2005)

[15] Object Management Group, Inc - OMG: Business Process Model and Notation. Versión 2.0.2. OMG, Massachusetts (2013)

[16] Guizzardi, G., Falbo, R., Guizzardi, R.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE Software Process Ontology. In : XI Iberoamerican Workshop on Requirements, Recife, pp.127-140 (2008)

[17] Project Management Institute: A Guide to the Project Management Body of Knowledge (Pmbok Guide) 5th edn. Project Management Institute, Newtown Square, PA. USA (2013)

[18] Brojt, D.: Project Management. Granica Adelphi (2007)

[19] Goede, R.: The Use of Soft Systems Methodology for the Development of Data Warehouses. In Callaos, N., Lesso, W., Sánchez, B., Siddique, M., Tremante, A., eds. : The 2nd International Conference on Information and Communication Technologies and Applications (ICTA), Orlando, Florida. USA, pp.1-6 (2012)

[20] Sen, A., Atish, P.: A comparison of data warehousing methodologies. Communications of the ACM 48(3), 79-84 (2005)

[21] Hull, E., Jackson, K., Dick, J.: Requirements Engineering 2nd edn. Springer, Londres, UK (2010)

[22] Dragicevic, S., Celar, S., Novak, L.: Roadmap for requirements engineering process improvement using BPM and UML. Journal of Advances in Production Engineering & Management 6(3), 221-231 (2011)

[23] Ellis, K., Berry, D.: Quantifying the Impact Requirements Definition and Management Process Maturity on Project Outcome in Business Application Development., Waterloo, Canada (2011)

[24] Becker, P., Papa, F., Olsina, L.: Enhancing the Conceptual Framework Capability for a Measurement and Evaluation Strategy. UNLPam, La Pampa, Argentina (2013)

Conectando Gambas 3 con Firebird/Interbase SQL Server

Cherencio, Guillermo Ruben
Universidad Tecnológica Nacional, Facultad Regional Delta

Abstract
Gambas 3 es un entorno de desarrollo de software de código abierto basado en el lenguaje BASIC que posee extensiones de objetos similares a Microsoft Visual Basic®; permite el desarrollo de aplicaciones con interfaz gráfica de usuario (GUI) para sistemas Linux. Todo el sistema esta formado por un compilador, un intérprete, componentes para el desarrollo de GUI y un entorno de desarrollo. Es un entorno propicio para el desarrollo rápido de aplicaciones client-server; este tipo de aplicaciones requieren de drivers o componentes propios o de terceros para interactuar con un sistema gestor de base de datos (SGBD). Gambas 3 provee el componente gb.db que le permite interactuar con SGBD's tales como: PostgreSQL, MySQL, SQLite2, SQLite3, etc., no obstante, no posee ninguna forma de interacción con un SGBD Firebird/Interbase SQL Server (FB). El objetivo de este trabajo es el desarrollo de un módulo de software que permita conectar a Gambas 3 con FB. Se utilizó la librería libfb[6]; se hicieron cambios sobre la misma y se desarrolló un módulo de software en Gambas 3 para realizar el vínculo con libfb. Este trabajo ha logrado conectar a Gambas 3 con FB, de esta forma, ahora es posible desarrollar aplicaciones client-server utilizando FB y Gambas 3. Las pruebas realizadas han sido satisfactorias en cuanto a la velocidad de ejecución y su facilidad de uso. Todo el software se distribuye bajo licencia LGPL.

Palabras Clave
Gambas3 Basic Linux Firebird Interbase SQL Server

Introducción

La librería *libfb* ha logrado simplificar la interfase de programación de aplicaciones (API) del Sistema Gestor de Base de Datos[1] (SGBD) Firebird/Interbase basado en el modelo relacional [2], permitiendo a programadores ANSI C desarrollar aplicaciones *client-server* de una forma más simple y productiva. Este trabajo surge como una ampliación del desarrollo [6] continuando con las futuras líneas de investigación allí propuestas.

La plataforma Linux es muy popular entre los desarrolladores ANSI C, pero no así entre los desarrolladores Visual Basic (VB). La alternativa propuesta a los desarrolladores VB en Linux es Gambas 3.

Gambas 3 no pretende ser un clon de VB, sino que provee extensiones de orientación a objetos, programación orientada a eventos, un entorno simple y productivo que facilita el diseño de interfases gráficas de usuario de forma rápida e intuitiva. Incorpora muchas características de Microsoft Quick-Basic®.

Gambas 3 posee una arquitectura abierta que permite la incorporación de librerías dinámicas para extender el lenguaje. Esta característica permitió el vínculo entre *libfb* y *Gambas 3*.

El módulo software *gb.db* incluido en *Gambas 3*, permite utilizar varios SGBD's populares en el mundo de código abierto (PostgreSQL, MySQL, SQLite2, SQLite3, etc), pero no permite utilizar FB debido a que el módulo *gambas-gb-db-firebird* (incluído en *Gambas 2*) tiene varios programas bajo licencia IPL (Interbase Public License) y esta licencia no sería compatible con la licencia GPL de Gambas 3[7], por lo tanto, el módulo fue retirado. Este inconveniente impide a una amplia comunidad de desarrolladores *Gambas 3* la posibilidad de usar y conocer FB y motiva el presente trabajo.

La posibilidad de *Gambas 3* de vincularse con librerías dinámicas compartidas ha motivado la modificación

de *libfb* para convertirla en una librería tanto estática como dinámica [5] y adaptarla para su uso desde *Gambas 3.*

Todo el proyecto *libfb* es bajo licencia LGPL, así como también la presente extensión, por lo tanto, no se considera ningún impedimento para el uso de esta nueva versión de *libfb* en Gambas 3.

Hasta el momento de la presentación de este trabajo, *libfb* se presenta en *Gambas 3* como una alternativa de librería dinámica compartida, provista por un tercero. No obstante, sería posible la presentación del presente trabajo al equipo de desarrollo de *Gambas 3,* para que evalúe su incorporación al módulo *gb.db.* De esta forma, el acceso a FB sería de forma nativa desde *Gambas 3*.

El presente trabajo pretende ser una contribución a la difusión y uso de FB, como base de datos de código abierto, en el campo académico y profesional, apoyando a todo el movimiento propuesto por la *Free Software Foundation* y que la misma sea accesible de forma simple por todos los desarrolladores *Gambas 3*, sin necesidad invertir grandes cantidades de tiempo, disminuyendo la curva de aprendizaje que implicaría trabajar directamente con la API de FB; así como también el ahorro de dinero en la adquisición de licencias o software adicional para este tipo de proyectos, ya que, la librería propuesta estará disponible para toda la comunidad bajo licencia LGPL, denominada proyecto *libfb* en *sourceforge*.

Al mismo tiempo, esto podría ampliar aún más la comunidad de desarrolladores *Gambas 3*, permitiendo el acceso de programadores con experiencia en FB. Ampliar la comunidad *Gambas 3*, es también una forma indirecta de ampliar el mundo Linux, ofreciendo a los desarrolladores BASIC una alternativa cada vez más completa, en una plataforma de código abierto que les permita desarrollar en un nuevo entorno o bien migrar aplicaciones propietarias a un entorno libre de licencias.

Elementos del Trabajo y metodología

Todo el trabajo se desarrolló en las siguientes etapas, tomando los lineamientos generales de Blanchette[3]:

1. Requerimientos
2. Análisis de Arquitectura *Gambas 3*.
3. Modificación de *libfb* para generar nueva versión de librería dinámica compartida[5].
4. Diseño e implementación de módulo de vinculación *libfb – Gambas 3*.
5. Implementación de programas de ejemplo y Testing.
6. Empaquetado, publicación y distribución.

1. Requerimientos

Los requerimientos son genéricos, pues el usuario destinatario de este trabajo es muy amplio, no obstante, se consideró que el mayor requerimiento a cumplir estuvo centrado en la *usabilidad* de esta librería desde el punto de vista de un programador BASIC, en especial el programador no experto, el cual requiere de abstracciones simples y fáciles de comprender [4]. La librería *libfb* soporta los tipos de datos básicos[1] de FB y deberá permitir una representación de los mismos en el lenguaje de *Gambas 3*. Permitir una forma simple de conexión a una base de datos FB, ejecutar una consulta, recuperar los datos de cada tupla y cerrar la conexión. Permitir activar/desactivar los mensajes de error o advertencias de la librería; permitir

1 Se refiere a los tipos FB: INTEGER, CHAR, VARCHAR, DATE, DECIMAL

recuperar errores SQL (códigos de error, mensajes, etc).

2. Análisis de Arquitectura *Gambas 3*.

La principal dificultad fue la falta de información técnica detallada en cuanto a la arquitectura de *Gambas 3*, en su página introductoria [8] se menciona la posibilidad de extensión del lenguaje a través del desarrollo de librerías dinámicas compartidas (*shared libraries*)[5] que agreguen nuevas clases al lenguaje. La mayoría de la bibliografía abarca temas introductorios y esta orientada al aprendizaje de *Gambas 3*. Se optó por el desarrollo de una simple librería ANSI C dinámica compartida, desarrollada acorde con [5]; se creó un simple proyecto *Gambas 3*, se instaló la librería en carpeta `/usr/lib` ; se creó un módulo de código en el proyecto, se utilizó la instrucción `Library` para vincular la librería con *Gambas 3* y se declararon las funciones utilizando la sentencia `Public Extern` ; luego se invocaron estas funciones desde el código *Gambas 3*, se experimentó con distintos tipos de datos, de forma tal, de poder cubrir todos los requerimientos de *libfb* en ese aspecto.

Se resolvieron todos los problemas de representación, por ejemplo, los punteros fueron resueltos a través del tipo de dato `Pointer`; `int` por `Integer`, etc.. Las estructuras que requiere *libfb* como `query` y `rquery` fueron implementadas de la siguiente forma dentro de `ModFb3.module`:

```
Public Struct Rquery
  '/** buffer que contiene arreglo de apuntadores a
char * de longitud
  ' * query.cols con el contenido de las columnas * /
  col As Pointer
  '/** puntero a siguiente tupla */
  sig As Struct Rquery
  '/** puntero a tupla anterior */
  prev As Struct Rquery
End Struct

Public Struct query
  '/** cantidad de filas */
  rows As Integer
  '/** cantidad de columnas */
  cols As Integer
   '/** buffer que contiene arreglo de apuntadores a
char * de longitud
   ' * query.cols con los nombres de las columnas */
   colname As Pointer
   '/** flag de Error True - > Error , False - > no
Error */
   fb_error As Integer
   '/** si hay Error , en este miembro queda el codigo
sql de Error */
   SQLCODE As Long
   '/** ultimo status del ultimo fetch realizado */
   FETCHCODE As Long
   '/** buffer con mensajes de Error */
   errmsg As String
   '/** cuantas filas fueron actualizadas */
   rows_updated As Integer
   '/** cuantas filas fueron insertadas */
   rows_inserted As Integer
   '/** cuantas filas fueron borradas */
   rows_deleted As Integer
   '/** cuantas filas fueron recuperadas,
seleccionadas */
   rows_selected As Integer
   '/** cuantas filas fueron obtenidas(fetched)en cada
ejecucion del query */
   rows_fetched As Integer
   '/** puntero a la primer tupla en memoria */
   top As Struct Rquery
   '/** puntero a la ultima tupla en memoria */
   bottom As Struct Rquery
End Struct
```

Las constantes utilizadas en *libfb* fueron implementadas de la siguiente forma:

```
'fb_do_query() event types:
Public Const FB_EXECUTE_QUERY_OK As
Integer = 1
Public Const FB_EXECUTE_QUERY_ERROR As
Integer = 2
Public Const FB_SET_QUERY_INPUT As Integer
= 3 'set Input parameter
Public Const FB_SET_QUERY_OUTPUT As
Integer = 4 'set Input parameter
Public Const FB_START_QUERY_ERROR As
Integer = 5 'Error trying start query, we can 't
continue
```

```
Public Const FB_MEMORY_QUERY_ERROR As Integer = 6 'Error allocating Memory For query, we can 't continue
Public Const FB_FETCH_RECORDS As Integer = 7 'cursor ready To fetch records
'call back function fb_do_query() return types:
Public Const FB_CONTINUE As Integer = 1 'commit, execute a parametrized query
Public Const FB_ABORT As Integer = 2 'rollback this query, but Continue trying other pendings queries
Public Const FB_ABORT_ALL As Integer = 3 'rollback this query And cancel all pendings queries
```

Las pruebas realizadas durante esta etapa permitieron establecer la viabilidad del proyecto, pues, la librería *libfb* descripta en [6] no tuvo como objetivo ser utilizada como una extensión de *Gambas 3*.

3. Modificación de *libfb* para generar nueva versión de librería dinámica compartida.

A partir de la lectura de [5] se realizó la modificación de los *scripts* de compilación y *linkedición* de *libfb* para que la misma, además de ser una librería estática, también se distribuya como una librería dinámica compartida.

La librería compartida se instala en `/usr/lib` bajo el nombre `libfb.so`. Dentro de un módulo *Gambas 3* puede vincularse con la librería `libfb.so` usando la instrucción `Library "libfb"` luego de ello, se puede utilizar la instrucción `Public Extern` para vincular cada una de las funciones de la librería *libfb*.

Se agregaron las siguientes líneas al archivo `Makefile` de *libfb* para generar la librería compartida y permitir el uso del proyecto de ejemplo *Gambas 3* provisto con la librería:

```
...
@echo "Compiling libfb.so ..."

@$(CC) -shared -Wl,-export-dynamic -Wl,-soname,libfb.so -o libfb.so libfb.o -lc

…
@echo "For gambas 3 users: libfb.so and libfbclient.so must be in /usr/lib"

@cp libfb.so /usr/lib

...
```

4. Diseño e implementación de módulo de vinculación *libfb – Gambas 3*.

El uso de algunas funciones *libfb* requieren de asignación dinámica de memoria; en muchos casos, se trata de estructuras y su tamaño en bytes puede variar según el modelo de memoria adoptado por *Gambas 3*. El tamaño de una estructura en ANSI C no es el mismo en todas las plataformas en donde la misma se compile, por lo tanto, se resolvió el problema incorporando nuevas funciones *libfb* que devuelvan el tamaño en bytes de ciertas estructuras requeridas para ser asignadas dinámicamente desde *Gambas 3*:

```
int  fb_sizeof_query();
int  fb_sizeof_rquery();
int  fb_sizeof_fb_blob_info();
int  fb_sizeof_fb_query_info();
int  fb_sizeof_fb_db_info();
```

De esta forma, desde *Gambas 3*, utilizando la función `Alloc()` (que permite la asignación dinámica de memoria), podrían escribirse líneas de código tales como:

```
...
 Dim pdb_len As Integer
 Dim ret As Integer
...
 pdb_len = fb_sizeof_fb_db_info()
 pdb_info = Alloc(pdb_len)
```

una vez asignada la memoria, la misma puede inicializarse con valores

nulos (cero binario, `\0`), utilizando la función `gmbfb_clear_mem()`:

```
Public Sub gmbfb_clear_mem(ByRef p As Pointer,
ByRef plen As Integer)
  Dim fp As Stream
  Dim b As Byte
  Dim i As Integer
  fp = Memory p For Read Write
  Seek #fp, 0
  b = 0
  For i = 1 To plen
    Write #fp, Chr(b), 1
  Next
  Close #fp
End
```

libfb también posee variables estáticas, cuyos valores afectan a su comportamiento, como es el caso de la variable `FB_SHOW_MESSAGES`, la cual, por defecto, esta asignada con el valor `1`, lo cual indica que deben mostrarse todos los mensajes por consola (ello podría ser un problema en una aplicación CGI[9]); si esta variable se asigna con el valor `0`, los mensajes serán omitidos. Para estos casos, se diseñaron dos funciones *Gambas 3:* una para obtener (*get*) y otra para establecer (*set*) el valor de estas variables:

```
Public Function gmbfb_getint_symbol(ByRef
symbol As String) As Integer
  Dim p As Pointer
  Dim i As Integer
  p = System.GetExternSymbol("libfb", symbol)
  If p <> Null Then
    i = Integer@(p)
  Else
    i = -1
  Endif
  Return i
End

'
'Enable to set an integer value to libfb library (i.e.:
set a value into global int variable)
'
Public Sub gmbfb_setint_symbol(ByRef symbol As
String, symbol_value As Integer)
  Dim p As Pointer
  Dim i As Integer
  Dim fp As Stream
  p = System.GetExternSymbol("libfb", symbol)
  If p <> Null Then
    fp = Memory p For Read Write
    Write #fp, symbol_value As Integer
    Close #fp
  Endif
End
```

Con estas funciones, podemos controlar el comportamiento de *libfb* desde *Gambas 3;* por ejemplo, para habilitar los mensajes de error deberíamos escribir:

```
ModFb.gmbfb_setint_symbol("FB_SHOW_MESSA
GES", 1)
```

Para establecer una conexión a FB (y otras operaciones sobre la base de datos) se requiere previamente de la asignación dinámica de memoria de la estructura ANSI C `fb_db_info` y pasar su dirección de memoria entre distintas funciones de la librería *libfb*, para ello se creó la función Gambas 3 `gmbfb_connect()` :

```
Public Function gmbfb_connect(ByRef dbname As
String, ByRef user As String, ByRef pwd As String,
ByRef role As String) As Pointer
  Dim pdb_info As Pointer
  Dim pdb_len As Integer
  Dim ret As Integer

  pdb_len = fb_sizeof_fb_db_info()
  pdb_info = Alloc(pdb_len)
  ModFb.gmbfb_clear_mem(pdb_info, pdb_len)
  fb_pre_connect(pdb_info, dbname, user, pwd, role)

  ret = fb_do_connect(pdb_info)
  If ret = 1 Then
    Return pdb_info
  Else
    Return 0
  End If
End
```

Se realizaron pruebas hasta lograr la correcta representación de la función de *callback*[10] utilizada por *libfb* en la ejecución de consultas de la siguiente forma:

```
Public Extern fb_do_query(dbinfo As Pointer,
queryId As Integer, sql_stmt As String, onDoQuery
As Pointer, buffer As Pointer) As Integer
```

```
…
Public Extern onDoGenericQuery(eventType As Integer, dbinfo As Pointer, buffer As Pointer) As Integer
```

Todo el código *Gambas 3* esta contenido en los módulos: `ModFb.module` y `ModFb.module`; ambos forman parte del proyecto de ejemplo que se distribuye con *libfb*, para que el mismo pueda ser reutilizado en otros proyectos *Gambas 3*[2]:

```
Library "libfb"
...
Public Extern fb_do_connect(dbinfo As Pointer) As Integer
Public Extern fb_pre_connect(dbinfo As Pointer, dbname As String, user As String, passwd As String, role As String)
Public Extern fb_do_cmd(dbinfo As Pointer, sql_stmt As String) As Integer
Public Extern fb_do_db(dbinfo As Pointer, sql_stmt As String) As Integer
Public Extern fb_do_disconnect(dbinfo As Pointer) As Integer
Public Extern fb_do_exec_query(dbinfo As Pointer, sql_stmt As String, pinput As Integer, poutput As Integer, onDoQuery As Pointer, buffer As Pointer) As Integer
Public Extern fb_do_query(dbinfo As Pointer, queryId As Integer, sql_stmt As String, onDoQuery As Pointer, buffer As Pointer) As Integer
Public Extern fb_error(fb_db_info As Pointer, fb_query_info As Pointer) As Integer 'the 2 nd parameter may be Null
Public Extern fb_fprintf(FILE As Pointer, query As Pointer)
Public Extern fb_free(query As Pointer)


Public Extern fb_sizeof_query() As Integer
Public Extern fb_sizeof_rquery() As Integer
Public Extern fb_sizeof_fb_blob_info() As Integer
Public Extern fb_sizeof_fb_query_info() As Integer
Public Extern fb_sizeof_fb_db_info() As Integer

Public Extern fb_stream_msg_close()
Public Extern fb_wait_events(fb_db_info As Pointer, events As Pointer, flag As Pointer, call_back_fname As Pointer)
```

2 Por razones de comprensión, sólo se indican las funciones más importantes.

```
Public Extern onDoGenericQuery(eventType As Integer, dbinfo As Pointer, buffer As Pointer) As Integer
...
```

5. Implementación de programas de ejemplo y *Testing*.

A medida que se completaba el punto 4, las funciones de ambos módulos eran utilizadas y probadas.

El proyecto de ejemplo *Gambas 3* esta formado por un simple formulario en donde el usuario puede ingresar los datos de conexión con el SGBD FB; los datos requeridos son: ubicación física de la base de datos, usuario, contraseña, rol. A la derecha de los campos de ingreso de datos se encuentran 4 botones etiquetados como *“Fb Query 1”, “Fb Query 2”, “Fb Query 3” y “Fb Query 4”*; los cuales permiten ejecutar 4 *queries* distintos en la base de datos FB, a modo de ejemplo, mostraremos aquí el código correspondiente a *“FB Query 1”*, en donde se realiza la conexión a la base de datos FB, se ejecuta el *query* y luego se desconecta:

```
Public Sub btnQry1_Click()
'Using libfb from gambas 3 !
'This demo is equals to libfb/examples/fbquery1.c
'
'Regards, Cherencio, Guillermo,
grchere@yahoo.com
'
  Dim ret As Integer
  Dim csql As String
  Dim pdb_info As Pointer
  Dim pdb_len As Integer
  Dim sm As Stream
  Dim pMyQuery As Pointer
  Dim pMyRQuery As Pointer
  Dim scol1 As String
  Dim scol2 As String

  'redirect libfb messages to file fbmessages.txt, libfb error messages will be sent to stderr
  sm = Open User.Home & "/fbmessages.txt" For Write Create
```

```
  'libfb messages redirected to <user
home>/fbmessages.txt
  ModFb.fb_set_stream_msg(sm.Handle, "w")

ModFb.gmbfb_setint_symbol("FB_SHOW_MESSA
GES", 1)
  'Message.Info("FB_SHOW_MESSAGES=" &
ModFb.gmbfb_getint_symbol("FB_SHOW_MESS
AGES"))

  pdb_info =
ModFb.gmbfb_connect(txt_dbname.Text,
txt_user.Text, txt_pwd.Text, txt_role.Text)
  'Message.Info("FB_SHOW_MESSAGES=" &
ModFb.gmbfb_getint_symbol("FB_SHOW_MESS
AGES"))
  If pdb_info <> 0 Then
    pMyQuery = Alloc(ModFb.fb_sizeof_query())
    ModFb.fb_init(pMyQuery)

    If ModFb.fb_do_query(pdb_info, 1, "SELECT
CODIGO, DESCR FROM TBL_PLANE ORDER
BY DESCR ASC",
      ModFb.onDoGenericQuery, pMyQuery) > 0
Then
      Print "Execute query! get query data"

      'get the first row
      pMyRQuery =
ModFb.fb_get_query_top(pMyQuery)
      While (pMyRQuery <> 0)
        'get first column value
        scol1 = ModFb.fb_get_col(pMyQuery,
pMyRQuery, 0)
        'get second column value
        scol2 = ModFb.fb_get_col(pMyQuery,
pMyRQuery, 1)
        'print column values
        Print "[" & scol1 & "][" & scol2 & "]"
        'point to next record in memory
        pMyRQuery =
ModFb.fb_rquery_next(pMyRQuery)
      Wend

      Print "Free Query memory!"
      ModFb.fb_free(pMyQuery)

    Else
      Print "Error executing Query!\nError fb " &
ModFb.fb_get_query_fberror(pMyQuery) &
        " sql code " &
ModFb.fb_get_query_sqlcode(pMyQuery) &
        " last fetch code " &
ModFb.fb_get_query_fetchcode(pMyQuery) &
        "\nMessage: " &
ModFb.fb_get_query_errmsg(pMyQuery)
    Endif
...
    ret = ModFb.fb_do_disconnect(pdb_info)
    Print "fb_do_disconnect ret=" & ret

    Free(pdb_info)
  Else
    Message.Info("Connection Error!")
  Endif

  'close message file
  ModFb.fb_stream_msg_close()
  sm.Close
End
```

6. Empaquetado, publicación y distribución.

La librería *libfb* se distribuye bajo licencia LGPL[3], se incluyen los programas fuentes, ejemplos y documentación, en una estructura de directorios comprimidos que el usuario puede descargar, descomprimir y compilar utilizando la utilidad `make`[4] sobre la carpeta principal. La distribución y publicación de este trabajo se realiza a través del portal SourceForge[5] bajo el nombre *libfb*.

Resultados

Las pruebas fueron realizadas sobre una plataforma Linux Debian Wheezy amd64, ya que *Gambas 3* es un entorno sólo disponible en Linux.

Se ha realizado una comparación entre el proyecto *Gambas 3* distribuido con *libfb* con respecto a la ejecución de los programas compilados en ANSI C equivalentes. Los tiempos obtenidos se inclinan a favor de ANSI C y ello se debe más que nada a la diferencia en los tiempos de ejecución entre un programa compilado (ANSI C) y un programa interpretado (*Gambas 3)*. No obstante, si sólo se consideran los tiempos asociados con las

3 http://www.gnu.org/licenses/lgpl.html
4 http://linux.die.net/man/1/make
5 http://www.sourceforge.net

operaciones propias de la base de datos FB, las diferencias entre ANSI C y *Gambas 3* son mínimas.

A diferencia del código ANSI C distribuido con *libfb* que ha sido utilizado en otras aplicaciones, aun no se tiene conocimiento del uso del código *Gambas 3* en otros proyectos y ello se debe a su reciente incorporación a *libfb*.

Las pruebas realizadas en Linux han sido satisfactorias, permitiendo el desarrollo client-server desde *Gambas 3* utilizando una base de datos FB conectada a través de *libfb*. Se ha podido establecer y mantener conexiones con bases de datos FB, en servidor local y en servidores remotos. La ejecución de los distintos tipos de consultas han sido libre de errores y con tiempos de ejecución aceptables. De esta forma, se considera alcanzado el objetivo propuesto por este trabajo.

No se han realizado pruebas en ambientes *multithreading* o con cierto nivel de concurrencia.

Discusión

El proyecto *Gambas 3* propuesto a modo de ejemplo esta orientado a eventos, pero el mismo no incorpora clases u orientación a objetos. Esto último permitiría una mejor integración con `gb.db` y posiblemente podría tratarse de una futura línea de trabajo.

Las pruebas realizadas han indicado un rendimiento aceptable en la conexión, ejecución de consultas y recupero de resultados de la base de datos FB provista como ejemplo con la librería *libfb*. Por lo tanto, ello, sumado a las facilidades propias de *Gambas 3* para el desarrollo de GUI's y la facilidad de programación de este lenguaje, hace que esta extensión sea una opción viable y hasta el momento, única, para aquellos que desean desarrollar aplicaciones client-server utilizando el SGBD FB.

Desde el punto de vista didáctico, podrían plantearse proyectos desde asignaturas tales como: Sistemas Operativos, Bases de Datos, Gestión de Datos, etc.

La implementación actual esta orientada a resolver consultas que recuperen un número de tuplas almacenable en memoria principal, es de suponer, que si se pretende recuperar un volumen de datos que supere la memoria principal disponible (a pesar de los mecanismos de virtualización de memoria principal con los cuales pudiera contar el sistema operativo), el procesamiento de este tipo de consultas tendría un impacto negativo en el rendimiento general del sistema. La arquitectura actual no impide que se desarrollen nuevas implementaciones que contemplen esta situación y permitan el uso de cursores y almacenamiento secundario para disminuir los requerimientos de memoria principal.

Conclusión

Gambas 3 es una herramienta propicia para el desarrollo de aplicaciones client-server por parte de programadores BASIC en plataforma Linux, la versión 3 introduce mejoras que no podrán ser aprovechadas por aquellos usuarios que realicen desarrollos client-server que utilicen una base de datos FB, pues *Gambas 3* no provee conectividad con FB. Las aplicaciones que interactúen con FB escritas en *Gambas 2*, no podrán ser migradas a *Gambas 3*. El presente trabajo permite dar solución a los problemas antes planteados a partir de la modificación de la librería *libfb* y del desarrollo de dos módulos de software que vincula a *libfb* con *Gambas 3*.

Los módulos Gambas 3 `ModFb.module` y `ModFb3.module` -ambos implementados a partir de este trabajo- incluidos en el proyecto *Gambas 3* distribuido junto con *libfb*, permiten el vínculo entre *Gambas 3* y la librería *libfb* para lograr la interacción con el SGBD FB desde Gambas 3.

En el proyecto se incluye código de ejemplo para la ejecución de distintos tipos de consultas, las cuales abarcan las necesidades básicas de toda aplicación client-server.

El código *Gambas 3* que se debe escribir para lograr una conexión con una base de datos FB, la ejecución de una consulta, el recupero de tuplas y la desconexión a la base de datos es muy simple; no obstante, también es posible, gracias a la arquitectura de *libfb,* la ejecución de consultas y procesos complejos desde *Gambas 3* contra una base de datos FB.

Este trabajo abre nuevas lineas de investigación en cuanto a mejoras que puedan introducirse en los módulos *Gambas 3* provistos, como así también en la posibilidad de reusar este código en otros proyectos *Gambas 3*, permitir nuevos desarrollos que aporten nuevas funcionalidades.

Agradecimientos
Al Ing. Luis H. Perna de la Universidad Tecnológica Nacional, Facultad Regional Delta, por motivarme a la presentación del presente trabajo.

Referencias

[1] Castaño Miguel, Adoración de; Velthuis Piattini, Mario; Martinez, Esperanza Marcos, Diseño de Bases de Datos Relacionales, Editorial Ra-Ma, Madrid, 2000, ISBN 84-7897-385-0, Pag 5.
[2] Silberschatz, Abraham; Korth, Henry F.; Sudarshan S., Fundamentos de Bases de Datos, 4ta.Ed., Ed. Mc Graw Hill, Madrid, 2002, ISBN: 0-07-228363-7, Pag. 17.
[3] Blanchette, Jasmin, The Little Manual of API Design, Trolltech, a Nokia company, June 19, 2008, disponible en http://www4.in.tum.de/~blanchet/api-design.pdf
[4] Jackson, Daniel, Software Abstractions, MIT Press, 2006.
[5] Wheeler, David A., Program Library HOWTO, disponible en http://tldp.org/HOWTO/Program-Library-HOWTO/
[6] Cherencio, Guillermo, Envolviendo la API de Firebird/Interbase SQL Server, Primer Congreso Nacional de Ingeniería Informática, CoNaIISI, 2013, UTN FRC, Córdoba, República Argentina.
[7] Gambas Forum, Item #2373, Topic #727, disponible en http://whiteislandsoftware.com/forum/index.php?page=topicview&id=gambas-news%2Ffirebird-database
[8] Gambas Project, disponible en .http://gambas.sourceforge.net/en/main.html
[9] Common Gateway Interface, disponible en http://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn
[10] Funciones de Callback, disponible en http://es.wikipedia.org/wiki/Callback_(inform%C3%A1tica)

Datos de Contacto:
Cherencio, Guillermo Ruben. Universidad Tecnológica Nacional, Facultad Regional Delta. San Martin 1171, CP 2804, Campana, Provincia de Buenos Aires, República Argentina. E-mail: grchere@yahoo.com.

Construcción y carga de un DataMart y su aplicación para la caracterización de incidentes informáticos.

Corso, Cynthia Lorena
García, Mario Alejandro
Ciceri, Leonardo Ramón
Romero, Fernando
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

Este trabajo forma parte de la difusión del proyecto "Generación de Modelo Descriptivo para la prevención de incidentes informáticos en el contexto de laboratorio de sistemas" –código UTN1683 homologado por la Secretaria de Ciencia y Tecnología de la Universidad Tecnológica Nacional.
El contexto de la investigación se sitúa en el Laboratorio de Sistemas de Información, cuya dependencia es el Departamento de Ingeniería en Sistemas de Información, perteneciente a la Universidad Tecnológica Nacional Facultad Regional Córdoba.
El objetivo principal de este estudio es la detección de los posibles factores que influyen en la presentación de incidentes en los equipos informáticos del laboratorio de sistemas.
Este trabajo presenta una propuesta para la construcción de un Data Mart perteneciente al Área Técnica del Laboratorio de Sistemas, tomando como base la metodología propuesta por Kimball. A partir de este diseño se considera como punto de partida para la definición de los procesos de ETL, implementados por la herramienta conocida como Kettle. En este caso en particular los datos de origen provienen de un único motor que es MySql que serán utilizados para la integración y carga del Data Mart.
Los resultados expuestos se utilizarán como fundamento para el personal perteneciente al área técnica que optimizará la gestión y mantenimiento de los incidentes informáticos.

Palabras Clave
Data Mart, OLAP, Laboratorios Informáticos, Incidentes de equipos informáticos.

Introducción

La gestión de las actividades relacionadas con el área de mantenimiento es clave para que cualquier unidad de negocio pueda desarrollar sus actividades de manera eficiente. Por ello es fundamental el establecimiento de un sistema que permita la detección de acciones de carácter preventivo con el objetivo de minimizar la presencia de incidentes. En el caso de que estos incidentes se presenten, es fundamental la capacidad de proporcionar una respuesta adecuada y rápida. La presentación de incidentes frecuentes en cualquier unidad de negocio u organizacional tiene un impacto negativo desde el punto de vista operacional, pero por sobre todo en el ámbito económico.
Es por ello que se considera necesario y significativo el estudio que permita el desarrollo de una metodología que permita el descubrimiento factores influyentes que propician la presentación de incidentes en maquinas, como así también detectar si existen relaciones entre los mismos.

[1] El crecimiento acelerado del volumen de datos que generan las organizaciones; exige a los administradores de los sistemas de información actuales desarrollar estrategias de análisis dirigidas a la toma de decisiones que beneficien a la organización. Esta actividad se complica aún más, si tenemos en cuenta que en la actualidad los volúmenes de datos que generan los procesos operacionales sobrepasan el alcance de las estructuras que soportan los sistemas de información.

En la actualidad al existir un volumen significativo de información y de diversos formatos, uno de los desafíos para las personas de negocios es tener la suficiente capacidad para analizar la información en

conjunto, proceso que se ve limitado justamente por el alto volumen. En muchos casos en diferentes organizaciones se ha aplicado como alternativa para paliar estas desventajas el concepto de Data Mart.

Un Data Mart se puede conceptualizar como un repositorio de información referido a un área o departamento especifico de una organización, por ejemplo compras, ventas, recursos humanos etc. A diferencia de los data warehouse cuyo alcance es mayor, es decir que considera toda la organización.

[2] Existen dos tipos de Data Marts:

Data Marts dependientes los datos que se utilizan para poblar el data mart provienen del data warehouse. Esta estrategia es particularmente apropiada cuando el data warehouse crece muy rápidamente y los distintos departamentos requieren sólo una pequeña porción de los datos contenidos en él.

Data Marts independientes los datos que se utilizan para poblar el data mart provienen de los sistemas operacionales y/o fuentes externas. Una de las limitaciones de los data marts independientes

- El hecho de tener varios data marts independientes, sin ningún tipo de integración, puede hacer que las tareas de administración y mantenimiento se conviertan en un lastre.
- Se puede dar el caso de que algunos data marts necesiten los mismos datos para dar respuesta a determinadas preguntas, por lo que tenemos redundancia de datos, ya que cada data mart se alimenta con sus procesos de extracción y transformación propios.

Entre las principales ventajas de implementar un Data Mart se puede citar:

- Acceso una herramienta para la toma de decisiones en cualquier área funcional, basándose en información integrada y global del negocio.
- Facilita la aplicación de técnicas estadísticas de análisis y modelización para encontrar relaciones ocultas entre los datos del almacén; obteniendo un valor añadido para el negocio de dicha información.
- Capacidad de aprender de los datos del pasado y de predecir situaciones futuras en diversos escenarios.
- Simplifica dentro de la empresa la implantación de sistemas de gestión integral de la relación con el cliente.
- Supone una optimización tecnológica y económica en entornos de Centro de Información, estadística o de generación de informes con retornos de la inversión espectaculares.

El modelado de los datos contenidos en un data mart muy difundido y de gran aceptación es el modelo Multidimensional.
[3] Los modelos multidimensionales se benefician de las relaciones inherentes en los datos para la carga de matrices multidimensionales llamadas cubos de datos. El rendimiento de la consulta en las matrices multidimensionales puede ser mucho mejor que en el modelo relacional.

Existen varios esquemas para el modelado de los datos en un Data Warehouse o Data Mart siendo los más utilizados:

Esquema de Estrella: bajo este esquema la tabla de hechos está en el centro de la estrella y están relacionadas con ella de forma radial todas las tablas de dimensiones, las cuales no se relacionan entre sí. No existen caminos alternativos en las dimensiones.

Esquema de Copo de Nieve: es similar al esquema de estrella pero existen jerarquías en las dimensiones. Las tablas de dimensiones pueden estar relacionadas, o sea, existen caminos alternativos en ellas.

Esquema Mixto: es una combinación de los dos esquemas anteriormente citados.

En la próxima sección se presenta los fundamentos de la metodología de Kimball que fue la adoptada para el diseño del Data Mart.

Elementos del Trabajo y metodología

[4] Ralph Kimball, se considera el padre de la Inteligencia de Negocio (Business Intelligence), pues él ha codificado el esquema estrella y las estructuras de datos llamadas copo de nieve, ha definido los conceptos de Data Mart, jerarquías de dimensiones, métricas base y agrupadas, drilling, entre otras. La selección de esta metodología es el grado de aceptación que ha tenido en diferentes contextos de aplicación, como así también propicia un ámbito de fácil implementación ya que parte de enfoque de menor a mayor, muy versátil. Como así también permite el acceso de diversas herramientas para la construcción de un Data Mart o Data Warehouse.

La primera etapa propuesta por Kimball es la definición de requerimientos, en esta etapa es necesaria una entrevista exhaustiva dirigida al personal de negocio y técnico, con el propósito de interiorizarse acerca de la terminología del negocio como así también las necesidades. El resultado de esta etapa se resume en la siguiente tabla.

Id Req.	Requerimiento
1	Cantidad de incidentes informáticos por turno de trabajo.
2	Cantidad de incidentes informáticos por tipo de componente.
3	Cantidad de incidentes informáticos por aula.
4	Cantidad de incidentes informáticos por año.
5	Cantidad de incidentes informáticos por antigüedad de equipo.
6	Cantidad de incidentes informáticos por semestre.
7	Cantidad de incidentes informáticos por origen de equipo, proveedor y antigüedad.
8	Cantidad de incidentes informáticos por aula y año.
9	Tiempo de inactividad de equipo por aula.
10	Tiempo de inactividad de equipo por semestre.

Tabla 1. Detalle de requerimientos relevados

Otro aspecto fundamental que se apoya esta metodología es la creación de un modelo denominado Modelo Dimensional, que es un proceso ágil y altamente iterativo. En la figura 1 se visualizan sus cuatro pasos.

1. Selección del proceso de negocio: en este caso de estudio nuestro proceso de negocio es la gestión y mantenimiento de incidentes informáticos que se presentan en el laboratorio de sistemas.

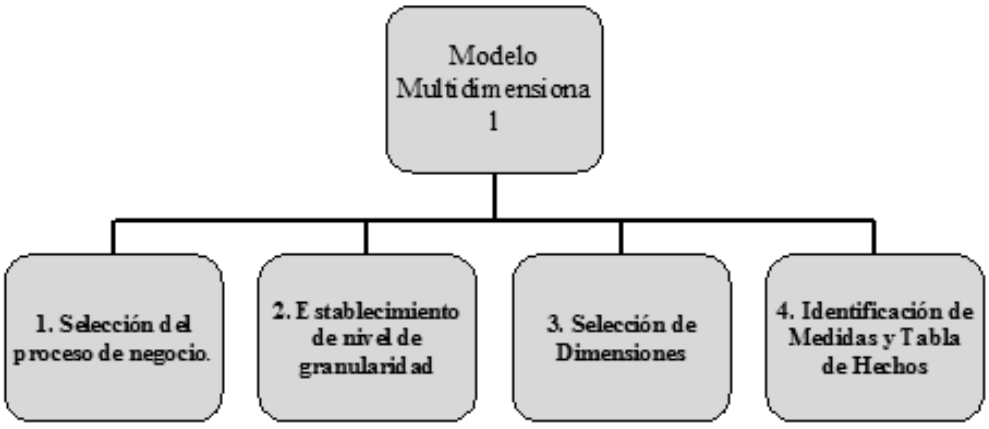


Figura 1. Pasos del Modelo Dimensional

2. Establecimiento de nivel de granularidad: En esta fase del proceso se trata de establecer cuál ha de ser el grado de detalle de las celdas que compondrán del cubo olap. En una primera aproximación se pretende establecer una granularidad que permita el nivel de detalle más preciso que los datos puedan ofrecer y de hecho esto es un principio a tener en cuenta en un buen diseño. Un exceso de detalle puede provocar que el sistema tenga que manejar una cantidad de datos que bien sea excesiva desde un principio, o bien crezca de una manera que provoque que las consultas se vuelvan desasido lenta y poco operativa.
Es necesario por tanto buscar un equilibrio entre nivel de detalle y carga del sistema.

En este caso de estudio los incidentes informáticos están registrados por **día y hora**. Según un primer análisis de las consultas requeridas pueden ser llevadas a cabo con la granularidad original de los datos tal como son facilitados por el laboratorio informático.
Teniendo en cuenta esta aproximación, la granularidad de los datos disponibles, la granularidad por día no supone una carga excesiva del sistema ya que el volumen de datos disponible no es muy numeroso.

3. Selección de Dimensiones: Las dimensiones generalmente surgen de la discusión entre los integrantes del equipo de trabajo, facilitada por el nivel de granularidad establecido.

Posteriormente se intenta determinar si ha quedado fuera del análisis algún candidato que debería ser tenido en cuenta, y esto se puede hacer estudiando si las diferentes instancias que se produzcan de nuestros hechos están perfectamente definidas y delimitadas con las dimensiones seleccionadas, y finalmente se trata de determinar si alguna dimensión se podría representar como una combinación de algunas de la otras, si es así, se está ante una dimensión fuertemente correlacionada con otra u otras y podemos prescindir de ella para nuestros propósitos.

En la siguiente tabla se especifican las dimensiones para este caso de estudio.

Id Dimensión	Dimensión
1	Área Reporte
2	Componente Real
3	Componente Posible
4	Gravedad
5	Incidente real
6	Lugar
7	Equipo
8	Prioridad
9	Tiempo
10	Tipo de Incidente
12	Tipo de Reparación
13	Turno de Trabajo

Tabla 2. Dimensiones seleccionadas

4. Identificación de Medidas y Tabla de Hechos: este último paso consiste en identificar las medidas que surgen de los procesos de negocio. Una medida es un atributo (campo) de una tabla que se desea analizar, agrupando sus datos, usando los criterios de corte conocidos como dimensiones. Las medidas generalmente se vinculan con el nivel de granularidad.
Las tablas de dimensiones contienen, generalmente, una llave o clave simple y atributos que la describen. En dependencia del esquema de diseño que se asuma pueden contener llaves foráneas de otras tablas de dimensión. Existe una dimensión fundamental en todo Data mart, la dimensión tiempo. Esto ocurre porque todo registro que se incluya constituye la ocurrencia de un fenómeno en un instante de tiempo definido. Dicha dimensión es la que establece uno de los objetivos fundamentales de la construcción de un data warehouse o data mart, la conservación de un "histórico". Los atributos dimensionales son fundamentalmente textos descriptivos, estos juegan un papel determinante porque son la fuente de gran parte de todas las necesidades que deben cubrirse, además, sirven de restricciones en la mayoría de las consultas que realizan los usuarios.

[5] La tablas de hechos representan la ocurrencia de un determinado proceso dentro de la organización y no tienen relación entre sí. Y además contienen como atributos medidas de un determinado proceso organizacional, teniendo en cuenta los requerimientos. La llave o clave de una tabla de hecho, es una llave compuesta, debido a que se forma de la composición de las llaves primarias de las tablas dimensionales a las que está unida.

Id Hecho	Hecho	Medidas
1	Incidente Informático	Tiempo de Inactividad
2	Mantenimiento	Antigüedad Reincidencia de Mantenimiento. Diagnostico Acertado.

Tabla 3.Hechos detectados

En la Figura 2 se ilustra el modelo conceptual obtenido reflejando el grado de relaciones entre la tabla de hechos y dimensiones.

El modelo conceptual resultante del proceso de diseño presenta un esquema mixto, mayormente responde a un esquema estrella y posee un copo de nieve para la dimensión Equipo Informático que deriva en la dimensión Proveedor.

Finalizado el proceso de diseño del data mart, se procede a la definición de los procesos ETL (Extracción, Transformación y Carga). Para cumplir este propósito hemos usado la herramienta Kettle, o también conocido como Spoon. La misma permite la realización de actividades de transformaciones de datos, realizar trabajos completos sobre la fuente de origen y extracción de datos de archivos como por ejemplo Access o planillas de cálculo entre otras. Para el caso de estudio que se presenta en este trabajo se citan algunas características de esta herramienta.

[6] Las conexiones de base de datos describen el método mediante el cual Kettle se conectará a una base de datos. Se pueden

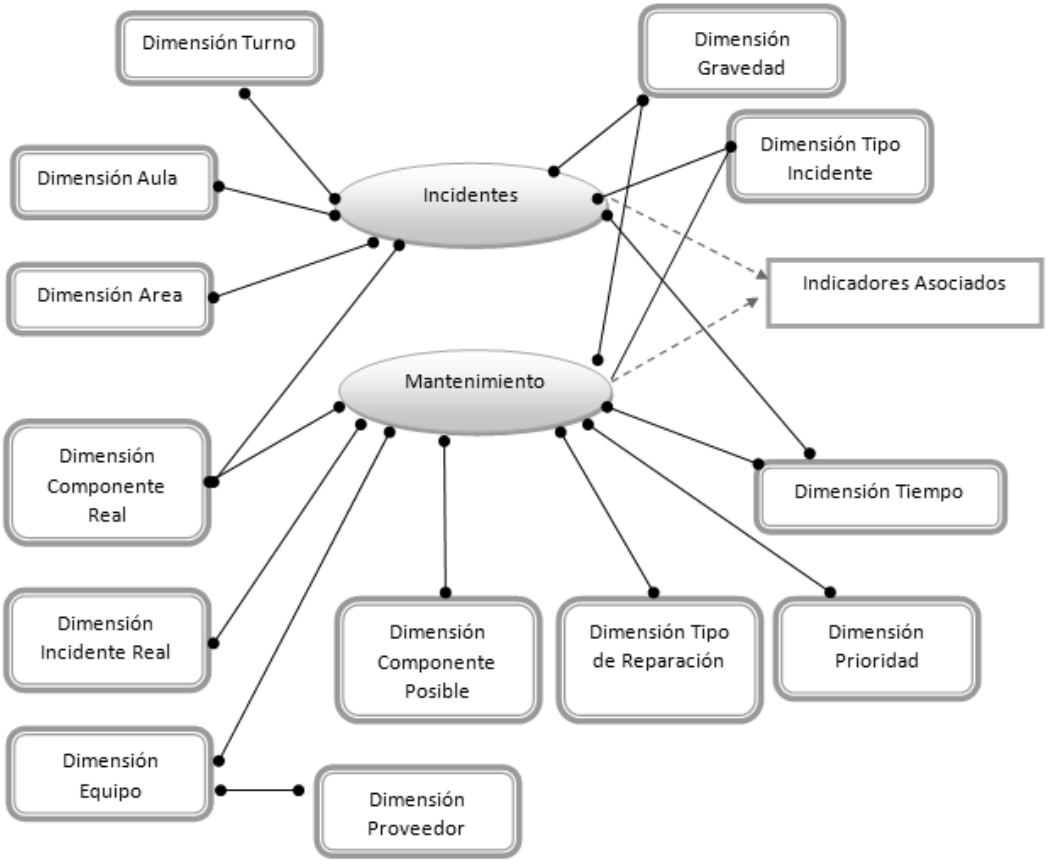


Figura 2. Modelo Conceptual resultante correspondiente al diseño de Data Mart.

crear conexiones específicas para un trabajo o transformación o guardarlas en el catálogo de Kettle para reutilizar las en múltiples transformaciones o trabajos.

[7] Un salto conecta un paso de transformación o entrada de trabajo con otro. La dirección del flujo de datos del salto se indica con una flecha en el panel de vista gráfica.

La entrada tabla permite disponer de una entrada de datos para una gran cantidad de motores de bases de datos; el asistente de configuración de este paso especifica el motor a conectar y a raíz de esto se especifican un conjunto de datos necesarios como por ejemplo el nombre del servidor que contiene la base de datos, aumentaciones para el acceso a esta etc.

Seleccionar/Renombrar valores nos permite hacer ajustes a los datos extraídos en componentes de entrada de tabla, entre los cambios que deben ajustarse a los datos se encuentra el renombrar los campos, cambiar el tipo de datos para compactarlo de un motor a otro; cambio en la longitud máxima de los campos etc.

El procedimiento realizado para la carga del Trabajo se resume en la figura 3.

El proceso ETL para la carga del modelo multidimensional planteado se compone de tres transformaciones, en cada una de ellas se ejecutan los scripts para cada etapa del proceso.

Regenerar BD: esta transformación tiene la responsabilidad de ejecutar un scripts que elimina las tablas de dimensiones y hechos y las vuelve a crear.
Cargar Dimensiones: la ejecución de este script facilita la carga de las tablas de dimensiones del modelo multidimensional planteado.
Cargar Tablas de Hechos: la transformación permite la ejecución de un scripts de las tablas de hechos del modelo multidimensional.

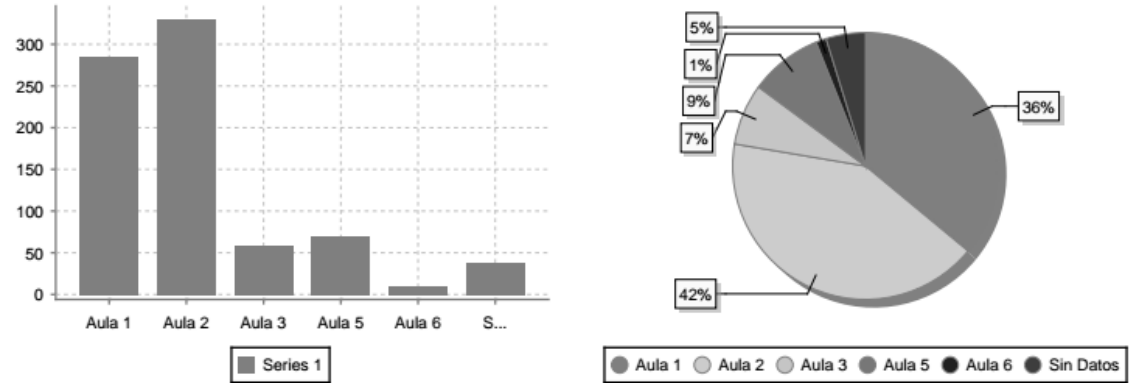
Finalizada el proceso de ETL (Extracción, Transformación y Carga) se ha usado una de las herramientas de Pentaho denominada Reporting Consulting. Uno de los objetivos principales del uso de esta herramienta es poder encontrar o detectar situaciones anormales que permita focalizar el estudio relacionado con la gestión y mantenimiento de incidentes. Como así también poder dar respuesta a los requerimientos relevados en la primera etapa.

Resultados

Los resultados obtenidos después la implementación del proceso de extracción, transformación y carga de datos como se ha expuesto en la sección anterior toman significado para dar respuestas concretas a los requerimientos formulados antes de iniciar este proceso. Algunos de los resultados se resumen en la figura 4 y figura 5.



Figura 3. Implementación del proceso ETL en Kettle.



De los

Figura 4. Gráfico de Reporte: Cantidad de incidentes

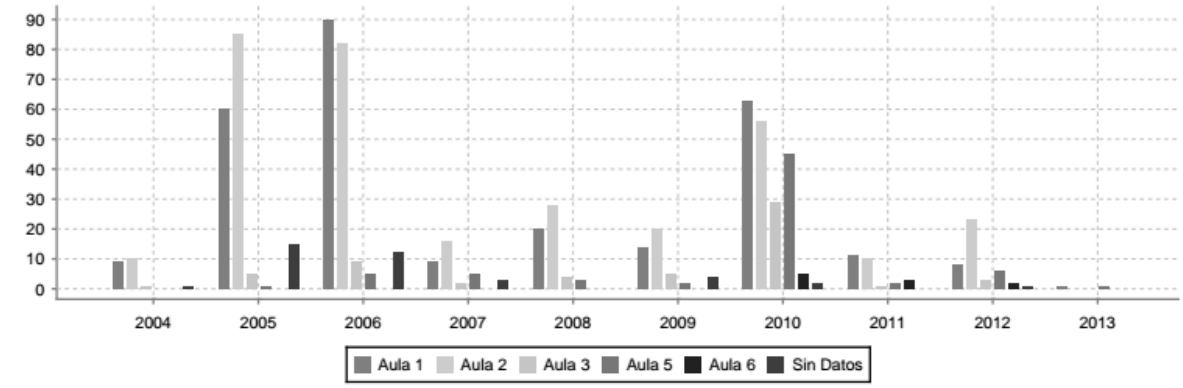


Figura 5. Gráfico de Reporte: Cantidad de incidentes

reportes obtenidos del data mart diseñado se puede concluir que:

- El periodo en que se evidencia más presentación de incidentes informáticos es desde el 2005 al 2006 inclusive y el año 2010.

- Los componentes que han sido reportados en mayor proporción como causante de incidentes en equipos son las lectoras de disco y fuentes. Justamente en el periodo de mas reportes de incidentes informáticos citado en el párrafo anterior.

El turno de trabajo que

- tiene presenta mayor número de incidentes informáticos es el turno de trabajo Mañana.

- Las aulas que presentan mayor presentación de incidentes informáticos son el Aula1 y Aula2.

- En el periodo de estudio considerado se han detectado la existencia de un grupo de incidentes informáticos que han sido reportados, pero que en realidad no lo eran. Es decir al llegar el reporte de incidentes al área Técnica se confirma que no hay incidente para un determinado equipo informático. La mayor proporción de incidentes informáticas con esta característica se concentra en los años 2006 y 2010.

Discusión

De las conclusiones expuestas en la sección anterior se considera profundizar el análisis y estudio de ciertas situaciones que se han detectado mediante los informes generados

por la herramienta Report Designer de Pentaho. Alguna de ellas es dar respuesta a los siguientes interrogantes como:

Porque hay existe alta tasa de reporte de incidentes en los años 2005 al 2006 y 2010?

Porque las lectoras de disco y fuentes son la causa más frecuente de presentación de incidentes en los equipos? Tiene alguna relación con la antigüedad, proveedor o el uso en una determinada aula?

Que características presentan en común los incidentes informáticos que finalmente no fueron incidentes? Hay un determinado factor o conjunción de factores que propician su presentación?

Con el propósito de dar respuestas a estos interrogantes, usaremos técnicas de Minería de Datos que permitan caracterizar y brindar posibles causas de estos interrogantes. Para esto se prevé la utilización de una de las herramientas de Pentaho que facilita la aplicación de dichas técnicas.

Conclusión

La metodología de Kimball proporciona una base empírica y metodológica perfectamente adecuada para la implementación de data warehouse y data mart como en este caso, debido a su gran versatilidad y su enfoque ascendente y progresivo.

La importancia de la construcción del data mart y es lo que le da gran importancia a este trabajo radica en el interés de las organizaciones de tomar decisiones sobre los hechos que acontecieron en algún instante de tiempo, para lograr ese objetivo se necesita entonces un buen análisis.

La herramienta de inteligencia de negocios Kettle fue de vital importancia para la población del data mart diseñado. Como así también la posibilidad de dar respuesta a los requerimientos iniciales de la organización negocio bajo estudio.

La metodología descripta en este trabajo puede llegar a ser útil y considerada como base; en el caso de situaciones de mayor complejidad donde la definición de proceso ETL requiere la integración de diversos orígenes de datos.

Referencias

[1] Calabria Sarmiento Juan Carlos, Construcción y poblamiento de un datawarehouse basado en el paradigma de bases de datos objeto relacional, Prospect. Vol. 9, No. 1, Enero - Junio de 2011, págs. 69-77

[2] Silberschatz Abraham; Korth Henry F., Fundamentos de Base de Dato. Cuarta Edición, Mc Graw Hill, 2002.

[3] Ramez Elmasri, Shamkat Navathe,Fundamentos de Sistemas de Base de Datos 5ta Edición, Pearson Educación S.A, Madrid.2007

[4] Kimball, Ralph. The Data Warehouse Toolkit. John Wiley & Sons. 2006.

[5] S. Carrera. “Transformación de Esquemas Multidimensionales Difusos del Nivel Conceptual al Nivel Lógico”. Tesis para optar al grado de Magíster en Ciencias de la Computación. Universidad de Concepción. Chile. 2009.

[6] Inmon , William . The Father of Data Warehousing. Inmon Consulting Services. 2007.

[7] Kimball, Ralph. The Data Warehouse Toolkit. John Wiley & Sons. 2006.

[8] Página official de la herramienta Pentaho, http://community.pentaho.com/projects/data-integration/

[9] C. Ballard, D. Herreman, D. Schau, R. Bell, E. Kim and A. Valncic. “Data Modeling Techniques for Data Warehousing”. SG24-2238-00. IBM Red Book. ISBN number 0738402451. 1998.

[10] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta and S. Paraboschi. “Designing Data Marts for Data Warehouses”. ACM Transactions on Software Engineering and Methodology. Vol. 10, Issue 4, pp. 452-483. October, 2001.

[11] Sierra Araujo Basilio, Aprendizaje Automático: conceptos básicos y avanzados. Aspectos prácticos utilizando el software Weka, Pearson Prentice Hall, Madrid, 2006.

[12] Fernandez-Medina, Trujillo J., Villaroel R., Piattini M., Developing secure data warehouses with UML extension, Decision support Systems, 2007.

[13] Fernandez-Medina, Trujillo J., Villaroel R., Piattini M., Access control and audit model for the multidimensional modeling of data warehouses. Decision support Systems, 2006.

Datos de Contacto:

Cynthia Lorena Corso. Universidad Tecnológica Nacional Facultad Regional Córdoba. Maestro M. López esq. Cruz Roja Argentina. E-mail: cynthia@bbs.frc.utn.edu.ar

Modelos para la Estimación de una Cota Superior del Número Cromático de Grafos Aleatorios Usando Regresión Lineal

Gustavo Dejean, Departamento de Computación
Universidad Nacional del Oeste
San Antonio de Padua, Pcia. de Buenos Aires
email: dejean2010@gmail.com

Resumen

La contribución principal del presente trabajo es mostrar cómo, usando técnicas del análisis de regresión lineal, se pueden establecer cotas superiores a los números cromáticos de los grafos aleatorios.
Se analizaron 3690 grafos aleatorios entre los rangos del 1% al 90 % de adyacencia y entre 100 nodos a 500 nodos. Para cada uno de ellos, se obtuvo una aproximación a su número cromático con un algoritmo simple y optimizado, para obtener un mejor acercamiento a su número cromático. Se obtuvo una superficie tridimensional: % de adyacencia – cantidad de nodos y mejor coloración obtenida. Por último se usaron métodos de análisis de regresión lineal para obtener un buen Modelo estimador. Se probaron muchas variables candidatas. Se determinó que el grado mínimo de los grafos aleatorios es un buen estimador de la cota superior de su número cromático usando Regresión Lineal Simple. Otro modelo más preciso, usando Regresión Multilineal, mejoró el anterior modelo. Ambos Modelo son una mejora sustancial en comparación a la cota del teorema de Brooks [1]. Los resultados son válidos dentro de los rangos probados.

1. Introducción

Los grafos aleatorios constituyen una importante rama de la Teoría de Grafos. Los primeros estudios datan de principios de la década de 1950 [2,3], casi diez años después, los investigadores P. Erdös y A. Renyi descubriendo importantes propiedades en esta clase de grafos, introduciendo el término "Teoría de Grafos Aleatorios" a esta área del conocimiento [4,5]. El objetivo de Erdös, fue estudiar el Método Probabilístico, sin embargo, actualmente, los grafos aleatorios, tiene aplicaciones muy diferentes a los objetivos originales de su creador, y se utilizan en diversas áreas del conocimiento como ser: las Redes Sociales, Modelos de Epidemias, de Comunicaciones y Biológicas. Para cada aplicación, pueden surgir diferentes variantes de los grafos aleatorios de Erdös: Grafos Geométricos Aleatorios, Regulares Aleatorios, Planares aleatorios y árboles aleatorios. El Modelo básico de Erdös, sigue teniendo plena vigencia y es campo de estudio en la actualidad. La coloración de grafos consiste en asignar un color a cada nodo de un grafo de manera tal que no queden dos nodos adyacentes con el mismo color. A la cantidad mínima de colores necesarios para colorear un grafo se la llama el número cromático del grafo, denotado por χ (G) [1]. Cada grafo tiene un número cromático asociado, que puede tomar valores dentro del rango que va desde 1 hasta N, ($1 \leq \chi$ (G) $\leq$ N) siendo N la cantidad de nodos del grafo. Un grafo completamente conexo es un grafo simple, donde cada par de vértices está conectado por una arista. Los grafos completamente conexos necesitan de N colores y un grafo sin aristas necesita solamente un color. La cantidad de arcos que tiene un grafo completamente conexo es n (n-1) / 2.
Hallar el número cromático de un grafo es un problema NP-Hard, por lo cual, es importante tener un modelo para estimar su valor. Un gafo aleatorio G(N, p), es un gafo donde cada arco tiene una probabilidad p de existir. Si p = 0.5, la cantidad promedio de arcos que habrá en los grafos G(N, 0.5) será la mitad de la máxima cantidad de arcos posible, esto es: n x (n-1)/ 4. (donde n es la cantidad de nodos que tiene el grafo). Un Grafo regular G(N,k) es un grafo donde cada nodo tiene exactamente grado k. El porcentaje de adyacencia de un grafo se calcula en función de la cantidad máxima posible de arcos:

$$\%ady = \text{cantidad de arcos} \times 100 \times 2 / n \times (n - 1)$$

Se define la densidad de un grafo como el coeficiente entre la cantidad de arcos que tiene y la cantidad máxima de arcos posibles:

$$densidad = 2 \times \text{cantidad de arcos} / n \times (n - 1)$$

En los grafos aleatorios, la densidad es aproximadamente igual a la probabilidad p. Por lo tanto, a p lo podemos expresar como:

$$p \approx densidad$$

La cota superior de χ (G) más usada y conocida es la del Teorema de Brooks, de 1941 [1,6] que dice: Sea G un grafo conexo que no es ni completo ni un ciclo impar. Entonces χ (G) $\leq \Delta$ (G); donde Δ (G) es el grado máximo de G. Otro teorema menos restrictivo es el Teorema de Vizing (1964) [1, 6] y dice: Para todo grafo G se tiene χ (G) $\leq \Delta$ (G) +1. La cota inferior más usada está relacionada a la existencia de subgrafos completamente conexos o cliquer, Un k-cliquer necesita

exactamente k colores para colorearlo. Si un grafo G contiene un k-cliquer, entonces se tiene que su número cromático no podrá ser inferior a k, por lo tanto será una cota inferior, esto es: $k \leq \chi(G)$. Esta cota inferior, en la práctica, es poco útil pues determinar si un grafo contiene un k-cliquer es también un problema NP-completo. Otra cota inferior está relacionada con la saturación de aristas y también son malas cotas para los grafos regulares y aleatorios.

La contribución principal del presente trabajo es mostrar cómo, usando técnicas del análisis de regresión lineal, se pueden establecer cotas superiores a los números cromáticos de los grafos aleatorios.

En las secciones siguientes, se describen la construcción del lote experimental, la selección y mejoras del algoritmo básico usado para obtener una cota superior de $\chi(G)$ y la construcción del conjunto de datos para el análisis de Regresión Lineal. Luego, se muestra un gráfico tridimensional donde se observa como es la superficie que describe el comportamiento de las cotas superiores halladas del número cromático en grafos aleatorios. Por último, se detalla el análisis realizado sobre el conjunto de datos experimentales y dos modelos de ajuste obtenidos con su correspondiente error.

2. Elementos del Trabajo y metodología

2.1. Construcción de un lote de casos experimentales

Para hallar una cota superior del número cromático primeramente se construyó un lote de datos experimental, consistente en 3690 archivos, cada uno de ellos representa un grafo aleatorio con una determinada probabilidad pi y una determinada cantidad de nodos Ni. Para crear el lote experimental, se preparó un algoritmo generador de grafos aleatorios, dándole como input 6 parámetros: las cantidades iniciales y finales, de nodos y de probabilidad, y el incremento para ambos parámetros. Los mismos, se inicializaron con N= 100 hasta N=500 con un incremento de 10; y desde p=0,01 hasta p=0,90 con un incremento de 0.01. Esto permitió obtener un lote de casos experimentales que abarcó un rango grande de valores posibles de N y p. El programa se desarrolló en Java y se usó la función rand para crear dichos grafos.

2.2. Elección del algoritmo para obtener una cota superior de $\chi(G)$ para cada grafo

A esta fase se la puede dividir en tres etapas: estudio del comportamiento de tres algoritmos conocidos, pero que en la literatura no se encuentran referencias a su comportamiento relativo. 2da etapa: mejoramiento de los algoritmos con técnicas pseudo-probabilísticas o por saturación de combinaciones aleatorias y análisis de resultados obtenidos. La tercer etapa fue adaptar el mejor algoritmo para hacerlo más eficiente y obtener una cota superior del $\chi(G)$ para cada grafo. Los algoritmos básicos implementados fueron los siguientes: Wells-Powell, Matula y el secuencial o aleatorio. Téngase presente que en esencia se tratan del mismo algoritmo y que solamente se diferencian en la heurística usada para procesar su input. En general, es de uso práctico obtener una cota del número cromático tomando un resultado de alguno de estos algoritmos [10] para luego optimizarlo con algoritmos que usan técnicas avanzadas como ser simulated annealing [11] o Ants algorimos [14] y branch and cut [13]. Para elegir la mejor cota, se estudió el desempeño relativo de los tres algoritmos básicos o heurísticas. Se ejecutó cada uno de estos algoritmos 1000 veces para colorear tres grafos con p= 50, 75 y 80% de adyacencia. Al graficar los resultados en tres histogramas (uno por cada grafo) se concluyó que los mejores resultados (para este tipo de grafos) siempre son del algoritmo de Powell, seguido por el aleatorio. También se observó que al ejecutar el mismo algoritmo muchas veces se alcanzan mejoras que llegan al 10% para cualquiera de los tres algoritmos. A esta técnica la podemos considerar como buscar una saturación de coloraciones posibles al ejecutar miles de veces cada algoritmo. Una vez seleccionado el algoritmo de Powell, se lo optimiza para que use sus propios resultados intermedios como poda en cada iteración, a fin de poder procesar la mayor cantidad de veces cada grafo en un tiempo mínimo. Se usó como corte el resultado obtenido en la primera pasada y luego cada vez que se obtuvo una menor cantidad de colores, se tomó este nuevo valor para cortar cualquier otra ejecución que requiera más de ese número. Se realizaron 10 mil corridas para cada uno de los 3690 grafos del lote experimental. La mejor coloración que generó para cada grafo (menor número), se grabó en un archivo junto con los datos físicos del grafo. A parte, en otro archivo, se guardó el grafo coloreado. Todos los resultados se comprobaron con otro programa probador para verificar el correcto funcionamiento del algoritmo de coloreo.

2.3 Preparación del conjunto de datos para el análisis

El conjunto de datos se formó de la siguiente manera: una fila por cada archivo del lote experimental (3690 filas) y teniendo como variables (columnas) los datos físicos de cada archivo del lote experimental, esto es, cantidad de nodos, cantidad de arcos, porcentaje de adyacencia, cantidad de arcos, grado máximo, grado mínimo, promedio entre el grado máximo y grado mínimo, densidad, densidad al cuadrado, densidad al cubo, grado máximo multiplicado por la densidad, y muchas combinaciones más, la variable objetivo o clase es el menor número obtenido en la coloración. Obsérvese que muchas variables son recalculables a partir de otras, pero lo que interesa aquí, es encontrar "la mejor variable" para predecir a la variable objetivo, así es que

en el trabajo, se probaron muchas otras variables que se fueron dejando de lado a medida que se demostraba que eran no significativas. Se utilizaron dos programa específicos para el estudio del análisis de regresión lineal: SPSS® y JMP®.

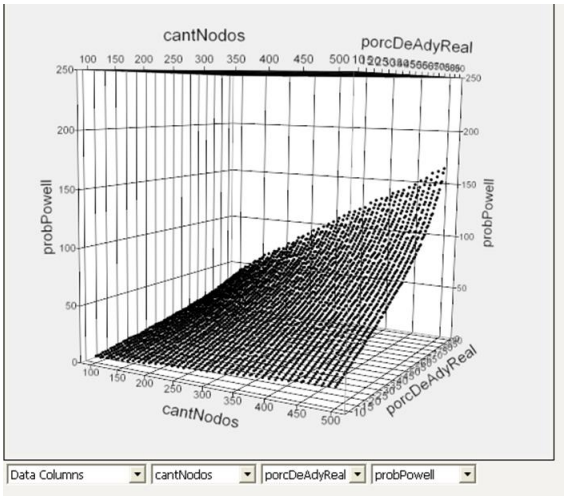


Figura 1*: cotas superiores halladas para cada grafo del lote experimental luego de ejecutar el algoritmo de Powell diez mil veces por cada grafo.*

En la Figura 1, se muestra como se distribuyeron los resultados de las diez mil corridas para cada archivo del lote experimental. Puede observarse que en el rango del 10% de adyacencia al 80% de adyacencia los resultados se mantienen aproximadamente lineales con respecto al porcentaje de adyacencia y cantidad de nodos. Mientras que a valores superiores al 80% de adyacencia, los valores se disparan notablemente.

3. Resultados

Primeramente se encontró un Modelo Lineal Simple, fácil de usar y calcular. Para ello, se analizaron numerosas variables. El análisis visual de las gráficas de dispersión, probó que el grado mínimo de un grafo aleatorio es un buen estimador para el rango de valores: $0{,}01 \leq p \leq 0{,}80$ y $100 \leq N \leq 500$. El Modelo Lineal Simple (MLS) hallado para ese rango de valores es:

$$\text{estimación}_{(MLS)} \text{ de } \chi(G) = 6{,}198 + 0{,}302 * \text{grado mínimo}$$

Este modelo tiene un valor para el estadístico de R cuadrado de 0,991; con un error estándar de la estimación de 2,321. Los residuos mínimos y máximos son: -7,89 y 9,40 respectivamente y el residuo medio es igual a cero. El p-value es igual 0. En la figura 2, se muestra el gráfico de dispersión y la recta obtenida y en la figura 3, se muestra como se distribuyen los residuos de la estimación. Lo anterior indica que se trata de un muy buen ajuste. Este Modelo predice, por ejemplo, que un grafo aleatorio G(N,p) con N, p dentro del rango de valores especificado, su número cromático está acotado por:

$$\chi(G) \leq \text{estimación}_{(MLS)} + 3*\text{error estándar}$$

donde la estimación se obtiene aplicando el modelo y el error estándar es 2,321; Se debe sumar a la estimación del Modelo, el triple de su error estándar para obtener una cota superior confiable al 99,74 %. Todo esto, gracias a que se demostró que los residuos siguen una distribución Normal. Formalmente se expresa como una probabilidad:

$$\mathbf{P}_{[\chi(G) \leq \text{estimación (MLS)} + 3*\text{error estándar}]} \approx 0{,}9974$$

Por ejemplo: Si se quiere obtener la cota superior de un grafo G(N,p) con N y p dentro del rango de valores especificado, y cuyo grado mínimo sea igual a 78, se calcula:

$$\text{estimación}_{(MLS)} + 3*\text{error estándar} = 29{,}75 + 6{,}96 = 36{,}71$$

o sea: $\mathbf{P}_{[\chi(G) \leq 36{,}71]} \approx 0{,}9974$

Por el contrario, la probabilidad de que dicho grafo tenga un número cromático superior a los 36 colores es de 0.0026, esto es de: (1 – 0.9974). En el lote experimental, se tienen 22 casos con grado mínimo igual a 78 y todos cumplen con la desigualdad ≤ 36.

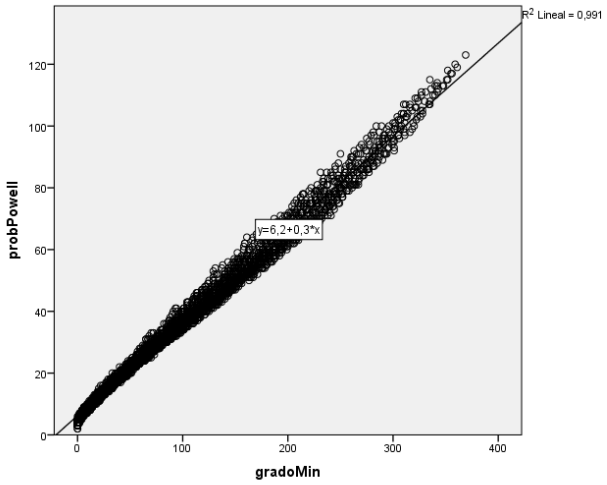


Figura 2*. Gráfico de dispersión y la recta que lo ajusta.*

El resultado queda expresado en forma independiente a los valores de N y p, y se observan casos donde, con distintos valores de N y de p, si su grado mínimo coincide, la estimación es aproximadamente la misma para todos. Todos los supuestos del Modelo lineal simple se cumplen menos el de Durbin-Watson (1951) ya que el valor de este estadístico da 0,86 y esto indica que los residuos no son totalmente independientes [7,8]. Efectivamente, los errores se hacen más grandes a medida que aumenta la variable dependiente. A pesar de este incumplimiento, en el lote experimental se verificó que, solamente en 59 casos de los 3239 casos, la estimación más 3*error estándar es inferior a la

cantidad de colores obtenida con el algoritmo, esto represente que solamente el 0,018% de los casos sobrepasaron la cota superior propuesta. En la práctica, no hace falta sumar tres veces su error estándar, pues como ya se dijo, el algoritmo usado para construir el lote experimental es básico y existen muchos algoritmos que usan técnicas avanzadas y producen resultados que se encuentran muy por debajo de los utilizados aquí.

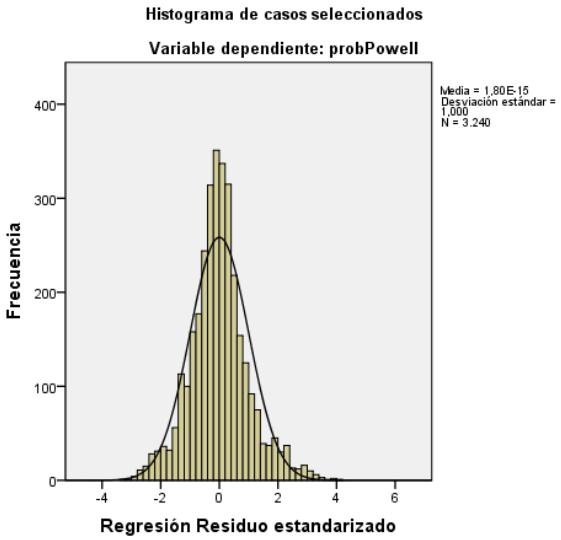


Figura 3. *Histograma de los de los residuos para el modelo regresión Lineal Simple*

Para mejorar el error de la estimación, se halló un segundo Modelo Lineal de Regresión Múltiple (MLRM):

estimación $_{(MLRM)}$ de $\chi(G)$ =

$$35,215d - 85,204d^2 + 70,905d^3 + 0,115\sqrt{cant.Arcos} + 0,227dgradoMax$$

donde: d = densidad y dgradoMax = d * grado_max

Obsérvese que en particular se trata de un modelo polinómico cúbico, pero haciendo las transformaciones triviales se puede escribir como un modelo Lineal Múltiple [7]. Este modelo, da un mejor ajuste que el modelo lineal simple para el rango de valores ya considerado. El estadístico R cuadrado es de 0.999, y el error estándar de la estimación es de 0,703. Los residuos mínimos y máximos son de -3,707 y 3,378 respectivamente, con una media igual a cero. En la figura 4 se muestra como se distribuyen los residuos de la estimación usando el Modelo de Regresión Lineal Múltiple y en la figura 5, se muestra la gráfica de probabilidad Normal. Esto demuestra que el Modelo cumple el supuesto de Normalidad de los Residuos. El estadístico Durbin-Watson que proporciona información sobre el grado de independencia entre los residuos es igual a 1,733; por lo tanto, se puede afirmar que el Modelo cumple con el supuesto de Independencia de los Residuos. Todo esto dentro del rango de valores ya mencionado. Si se reemplaza el valor de d por el de p, se obtiene una aproximación aceptable.

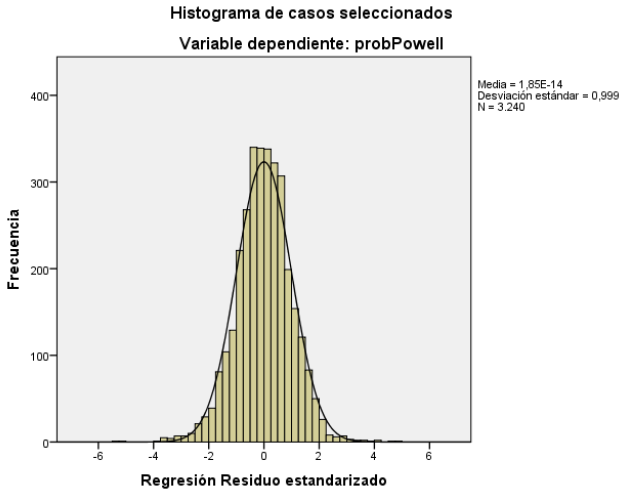


Figura 4. *Histograma de los de los residuos para el modelo regresión Múltiple*

Análogamente a como se procedió en el MLS, se tiene:

$$\mathbf{P}_{[\chi(G) \leq \text{Estimación (MLRM)} + 3*\text{error estándar}]} \approx 0,9974$$

Se verificó que en el lote experimental, solamente en 23 casos de los 3239 casos, la cota calculada con el Modelo fue inferior a la cantidad de colores obtenida con el algoritmo, esto represente que solamente el 0,007% de los casos, sobrepasaron la cota superior propuesta.
Por ejemplo, si tenemos un grafo aleatorio G(N=300,p=0.4) y con grado máximo igual a 96, su número cromático está acotado superiormente, con una probabilidad de 0.9974 por 35 colores:

$$\chi(G) \leq \text{ estimación}_{(MLRM)} + 3 * \text{error estándar} = 35,78$$

En este caso, la estimación aplicada al Modelo es 33,68 y el error estándar es igual a 0,703. En la Tabla 2, tercera fila, se muestra este caso, que pertenece al lote experimental. Obsérvese que la cota del Teorema de Brooks, para este caso, da 147 (su grado máximo), y queda lejos de la cota obtenida aplicando el Modelo. Nuevamente es oportuno aclarar que, en la práctica, no hace falta sumar tres veces su error estándar por las mismas razones ya mencionadas.
Una limitante del modelo, era que su validez estaba solamente dentro del rango de valores específico. Para validar su poder de predicción fuera de ese rango, se construyó otro lote experimental compuesto por 187 grafos aleatorios que van desde $0 \leq p \leq 0,80$ y $500 \leq N \leq 1000$, con paso de 0.05 para p y de 50 para N. A estos grafos se les ejecuto el mismo algoritmo que a los anteriores y también con 10000 corridas, para cada grafo. Al nuevo conjunto de datos obtenido, se le aplicaron ambos modelos, sin aplicar la suma del error estándar. En el caso del Modelo lineal de regresión

simple (MLRS), las estadísticas descriptivas de los residuos fueron: cantidad de casos= 176; mínimo valor: -9,7; máximo valor: 26,16; suma= 1353; media= 7,69;

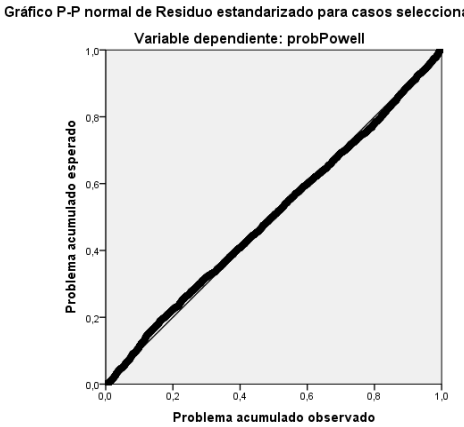


Figura 5. Gráfico P-P, que prueba visualmente la distribución Normal de los residuos

desviación estándar= 7,51. Es decir, que los residuos no siguen una distribución Normal ni su media es cero. Sin embargo, puede notarse que como cota superior es válida, pues los residuos están desplazados hacia los valores positivos, esto es porque la pendiente verdadera para este nuevo rango de valores es de 0.281 o sea que es menor que la del MLRS usado. En el caso del Modelo Multilineal, las estadísticas descriptivas de los residuos dan mejores resultados: cantidad de casos= 176; mínimo valor: -2,9; máximo valor: 11,76; suma= 479; media= -2,72; desviación estándar= 2,63. Se observa también que los residuos no siguen la distribución Normal, pero su media está desplazada hacia los valores positivos, por este motivo, se puede afirmar que el Modelo sigue siendo aceptablemente bueno en el nuevo rango de valores, aunque se observe cierta degradación en el ajuste del modelo. Otra prueba, realizada para ambos modelos fue testearlos con los datos experimentales documentados en el trabajo de Davis Johnson [11]. En dicho trabajo, en las tablas I, II,V y VI, se muestran resultados experimentales para grafos aleatorios de 125, 250, 500 y 1000 nodos con p=0,1; 0,5 y 0,9; los resultados obtenidos en aquel experimento, para el algoritmo RLF (Recursive Larges First) son bastante semejantes a los mostrados en el presente trabajo para el grafo de 125 nodos donde la diferencia es de dos colores y se agranda la diferencia para N=500 (67 en el presente trabajo vs. 49). Para otros algoritmos más eficaces usados en [11], la diferencia es aún mayor.
El grado máximo se calculó aproximadamente a partir de los datos del lote experimental. Aplicando el MRLM se obtiene que en todos los casos se cumplen las cotas propuestas, incluso los cuatro casos que corresponden a p=0.9 que está fuera del rango aplicable al modelo por no haber sido tenido en cuenta en el cálculo del Modelo. En todos los casos, se obtuvo una cota superior muy alejada de la cota del Teorema de Brook.

4. Conclusiones

Ambos Modelos de Regresión Lineal presentados, predicen una mejor cota superior del número cromático de grafos aleatorios que la del Teorema de Brooks. En la tabla 2, se muestran los residuos para ambos modelos para un conjunto de grafos sacados del primer lote experimental, se puede observar que los residuos para el MLRM varían entre uno y cero, mientras que la cota de Brooks (el grado máximo) en todos los casos, toma valores muy alejados de los resultados obtenidos con el algoritmo de Powell modificado, esto origina residuos muy grandes y por ese motivo, la cota de Brooks, no es una buena cota ni tiene utilidad práctica. La limitante de ambos modelos, es que su validez está dentro de un rango de valores. El Modelo Lineal de Regresión Simple, es más limitado que el MLRM, tanto en la calidad de la cota, medida con el error estándar de las estimaciones, como en la robustez frente a cambios en los rangos de valores de p y N. En contraparte, el MLS tiene la simpleza de poder predecir la cota superior como la tercera parte de su grado mínimo aproximadamente.

***Tabla 2** : residuos obtenidos para ambos modelos y contraste con la cota de Brooks (el grado máximo)*

cant. de nodos	cant. de arcos	grado maximo	grado minimo	% de ady (100 p)	Powell 10000 corridas	residuos usando la cota de Brooks	residuos del modelo lineal simple	residuos del modelo lineal multiple
110	2382	59	28	39,73	15	44	0	-1
200	7929	97	61	39,84	25	72	0	1
300	17900	147	96	39,91	34	113	-1	0
140	3884	70	39	39,92	19	51	1	0
360	25824	179	112	39,96	40	139	0	0
310	19142	151	99	39,97	35	116	-1	0
450	40380	220	151	39,97	48	172	-4	0
290	16750	140	95	39,97	34	106	-1	1
480	45987	235	154	40,00	51	184	-2	0
280	15630	138	86	40,02	32	106	0	0
390	30361	187	123	40,03	42	145	-1	0
370	27325	176	125	40,03	40	136	-4	0
350	24449	164	117	40,03	39	125	-2	1
340	23075	163	109	40,04	38	125	-1	1

Los Modelos presentados, se hicieron a partir de coloraciones encontradas con un algoritmo simple, aun cuando se hayan corrido miles de veces, es conocido que existen otras soluciones que usen menos colores. Así es de esperar, que las cotas que dan ambos modelos sean holgadas y se podrían mejorar. Esta partida de malas coloraciones puede estar ocultando una naturaleza "más lineal" de la figura 1, pues la diferencias van creciendo con el valor de p. El estudio realizado se puede extender fácilmente a los grafos regulares (grado mínimo = grado máximo) ya que estos tiene un comportamiento parecido a los grafos aleatorios por el motivo que cuando p aumenta, el grafo aleatorio tiende a ser muy regular. El análisis de regresión multivariante puede ser una herramienta útil para buscar cotas para otros algoritmos NP-completos, más en los casos donde no importa los detalles del resultado final sino solamente

una estimación de su resultado óptimo. Queda abierta la tarea de encontrar otro Modelo que abarque rangos de valores mayores a los estudiados y que parta de un conjunto de datos donde la variable objetivo sea el resultado de la aplicación de algoritmos avanzados.

Agradecimientos

A las autoridades de la Universidad Nacional del Oeste y en especial al Mg. Antonio Fotti, por facilitar la tarea de investigación.

Referencias

[1] Diestel M. Graph Theory, Springer ed., Graph Theory GTM, 4th edition 2010

[2] Rapaport A., Contribution to the Theory of Random and Biased Nets, Bulletin of Mathematical Biophysics 19, 257-277 (1957)

[3] Solomono R., Rapoport A., Connectivity of Random Nets, Bulletin of Mathematical Biophysics 13, 107-117 (1951)

[4] Erdös P., Renyi A, On Random Graphs, Publ.Math, (Debrecen) 6,290. (1959).

[5] Erdös P., Renyi A, On the Evolution of Random Graphs, Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5, 17-61 (1960).

[6] Molloy M., Reed B., Graph Colouring and the Probabilistic Method. Springer Science & Business Media, 2001

[7] Montgomery - Peck - Vining, Introducción al análisis de Regresión Lineal. Editado en: México, Compañía Editorial Continental. 2006

[8] Johnson Dallas E., Métodos Multivariados Aplicados al análisis de Datos. Editado en: México, International Thomson Editores, 2000.

[9] Pérez Lopez, César. Minería de Datos, Técnicas y Herramientas. Editorial: Thomson. (cap 12 y 13) 2007.

[10] Ozon Javier. Contribución al coloreo de grafos y a las redes pequeño mundo. Tesis Doctoral - Universitat Politècnica de Catalunya (2001).- http://hdl.handle.net/10803/5844; visitado el 19/08/2014

[11] Johnson David – Aragon C. Optimization by simulated annealing: An experimental evaluation; part II,graph coloring and number partitioning. Operations Research. Vol. 39, No.3, May-June 1991 (1990.).

[12] Douglas West. Introduction to Graph Theory. Prentice Hall, 2000.

[13] Severin, Daniel E. Estudio poliedral y algoritmo branch-and-cut para el problema de coloreo equitativo en grafos. Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Ciencias de la Computación. 2012

[14] Vesel A., Zerovnik J. How Well Can Ants Color Graphs? Journal of Comuting and Information Technology – CIT 8, 2000, 2, 131-136. - http://www.researchgate.net/publication/47397330_How_Well_Can_Ants_Color_Grap

Datos de Contacto: dejean2010@gmail.com

An Extension to Hierarchical Distance-based Conceptual Clustering

A. Funes[1], M. J. Ramírez-Quintana[2], R. Uzal[1]
[1]*Universidad Nacional de San Luis,*
Ejército de los Andes 950,
5700 San Luis, Argentina
{afunes@unsl.edu.ar, ruzal@uolsinectis.com.ar}
[2]*DSIC, Universidad Politécnica de Valencia,*
Camino de Vera s/n,
46022 Valencia, España
{mramirez}@dsic.upv.es

Abstract

Hierarchical Distance-based Conceptual Clustering (HDCC) framework is a general approach to conceptual clustering; it extends the traditional agglomerative algorithm for hierarchical distance based clustering by producing on the fly conceptual descriptions of the discovered clusters. The main contribution of HDCC framework is the set of mathematical tools and theoretical results useful for the analysis of consistency between distances and generalisation operators, which allows beforehand to know how compatible distances and generalisations are when used in the context of HDCC's algorithm. The framework defines three different levels of agreement between the clustering hierarchy induced by the linkage distance and the new hierarchy of concepts and clusters induced by the combined use of generalisation operators and distances. It also defines three properties the generalisation operators must satisfy in order to produce distance-consistent dendrograms according to the three levels of agreement.

Inspired by the concept of distance-based generalisation proposed in Estruch' work [3], *in this paper, we compare the conditions for distance-based generalisation operators vs. the properties defined in HDCC framework for consistent generalisation operators; we also extend the framework by adding a new level of consistency based on the concept of distance-based generalisations.*

1. Introduction

Two different approaches can be found in Machine Learning. On the one hand, we find similarity-based methods; on the other hand, we have symbolic methods based on models.

Methods based on similarity use a similarity function to achieve an inductive learning task. In particular, they can use a distance. Similarity is a broader concept than distance, and it is on the basis of many inductive techniques, where similar elements are expected to behave in similar way. The concept of distance, not only formalizes the concept of similarity between two individuals, but also satisfies additional properties of a metric space that can be used in benefit of the learning task. There are several popular methods based on similarity, both for unsupervised and supervised learning, such as hierarchical clustering [1][12], k-NN (k Nearest Neighbours) algorithm [2], k-means algorithm [14], among others.

In model-based symbolic techniques, the underlying concept is generalisation, which is also an important concept in Machine Learning since any inductive learning involves some kind of generalisation. These techniques, unlike distance and similarity-based techniques rely on the idea that concepts (generalisations or patters) discovered from old data can be employed to describe new data covered or described by them. Model-based techniques receive their name because they produce a symbolic model, which can be interpreted by users. Some well-known supervised symbolic techniques are decision trees and association rules; and a well known unsupervised symbolic learning technique is Michalski's conceptual clustering [15][16]. Other related works are COBWEB [5] and GCF [20].

Although distance-based methods are flexible (they can be applied to any data type provided a suitable distance or similarity function is available for the data), the learned function or model can be inexistent (e.g.

instance-based learning) or can be cryptic for users, offering no explanation on decisions made. In particular, distance-based clustering techniques discover clusters, using a similarity numeric measure between examples, by maximizing intra-cluster similarity and minimizing extra-cluster similarity. In these techniques, the resulting clusters lack of conceptual descriptions, making difficult their interpretation. For example, let us consider a clustering of documents where according to a given similarity function five clusters or document classes with similar characteristics have been found. We know that the elements in each cluster have been grouped because they are close (or similar) with respect to the underling similarity function o distance used; nevertheless, it is more useful to know in addition what the common features to the documents in each cluster are, i.e. to have clusters enhanced with patterns or generalisations that describes the common properties of the elements in the clusters.

Several hybrid methods have been proposed in the literature aimed to overcome precisely this problem. Examples of them are Anaprom system [10], EACH (Exemplar-Aided Constructor of Hyperrectangles) system [19], RISE (Rule Induction from a Set of Examples) system [4], RIONA (Rule Induction with Optimal Neighbourhood Algorithm) [11], among others. However, when combining both paradigms, inconsistency problems between the underling distance and the discovered generalisations can appear. In this category is HDCC, which integrates both learning paradigms under a single framework in the way of a general approach to hierarchical clustering that aggregates distances and generalisations. It provides a hierarchical clustering algorithm that produces a new kind of cluster hierarchies called conceptual dendrograms. Conceptual dendrograms are dendrograms whose clusters have conceptual descriptions extracted from the combined application of generalisation operators and distances.

The main contribution in HDCC framework is the possibility of studying in the context of HDCC algorithm the frequently found inconsistencies between distances and generalisations. That is why this framework provides a set of mathematical tools that allows the analysis, in the context of the new algorithm, of the degree of consistency between pairs of distances and generalisation operators. In consequence, one can verify beforehand whether the hierarchy induced by a given distance and the discovered patterns produced by a given generalisation operator are consistent. In other words, it allows answering the question “Are the elements covered by a pattern produced by a given generalisation operator close with respect to the underlying distance?”.

The main contribution of this paper is an extension to HDCC theoretical framework. To analyse the level of agreement between generalisations and distances, the framework defines three levels of consistency. It also defines, for each level, the corresponding property that generalisation operators must meet in order to satisfy that level of consistency. These levels of consistency guarantee that the resulting conceptual dendrograms will reflect in a higher or lower degree the distribution of the examples in their metric space.

The extension to HDCC framework presented here has been inspired by Estruch’s work [3]. We propose a new level of consistency for conceptual dendrograms based on the concept of distance-based generalisations defined by Estruch.

The rest of this work is organized as follows. In Section 2, some necessary previous concepts are summarized. Main concepts of HDCC framework are given in Section 3. In Section 4, we present the extension to the framework and some examples of distance-based generalisation operators. Finally, Section 5 closes the paper with the conclusions and future work.

2. Preliminaries

A way of defining a generalisation of a finite set of elements E in a metric space (X, d) is extensionally, as a set that contains E. However, this extensional definition does not give any insight into the concept or pattern that the elements in the generalisation share.

A pattern $p \in \mathcal{L}$, where $\mathcal{L}$ is a pattern language, is an intentional way of representing a set of elements in X. For example, given the numbers {10, 3, 8} a pattern describing them could be the closed interval [3, 10].

Additionally, the elements described by a patter p are denoted by $Set(p)$, which is referred as the *coverage* of the pattern, and we say that an element $x \in X$ *is covered by a pattern* p if $x \in Set(p)$. For example, $Set([3, 10]) = \{3, 4, 5, 6, 7, 8, 9, 10\}$ and 6 is covered by [3, 10].

In HDCC framework, a pattern is the result of a function that transforms two elements of a metric space and whose coverage includes at least the transformed elements, or it is the result of a function that transforms two patterns and whose coverage includes at least the elements in the union of both pattern coverages. These functions receive the name of *binary generalisation operators* and *pattern binary generalisation operators*, respectively.

Next definitions introduce formally these concepts. Definition 1 formalise the concept of binary generalisation operator over a metric space, while the concept of binary generalisation operator of patterns is given in Definition 2.

Definition 1. *Let* (X, d) *be a metric space and* $\mathcal{L}$ *a pattern language. A* binary generalisation operator *is a function* $\Delta: X \times X \rightarrow \mathcal{L}$ *such that given* $x_1 \in X$, $x_2 \in X$, $\Delta(x_1, x_2) = p$, *where* $p \in \mathcal{L}$ *and* $x_1 \in Set(p)$ *and* $x_2 \in Set(p)$.

Fig. 1(a) shows five possible generalisations (patterns) of two points *a* and *b* in the metric space $(\Re^2, d)$, where *d* is the Euclidean distance.

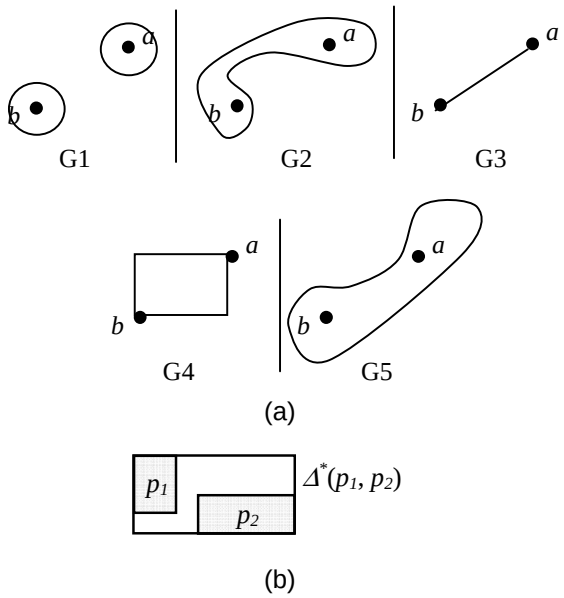


Fig. 1. (a) Five possible generalisations of two points in $\Re^2$. (b) A generalisation of two patterns p_1 and p_2 in $\Re^2$

Definition 2. *Let* (X, d) *be a metric space and* $\mathcal{L}$ *a pattern language. A* pattern binary generalisation operator *is a function* $\Delta^*: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ *such that given* $p_1 \in \mathcal{L}$ *and* $p_2 \in \mathcal{L}$, $\Delta^*(p_1, p_2) = p$, *where* $p \in \mathcal{L}$ *and* $Set(p_i) \subseteq Set(p)$ $(i \in \{1,2\})$.

In Fig. 1(b), we show a possible generalisation for two patterns p_1 and p_2 in $\mathcal{L}$, where $\mathcal{L}$ is the set of all axis-parallel rectangles.

Let us note that when $\mathcal{L} = X$, the operators Δ^* and Δ can be the same. For example, $\mathcal{L} = X$ = first order atoms, Δ^* and Δ can be instantiated in Plotkin´s least general generalization operator (*lgg*) [17].

3. Hierarchical Distance-based Conceptual Clustering Framework

3.1. The Algorithm.

HDCC is based on the traditional agglomerative hierarchical clustering [9][1] that builds a hierarchy of clusters from individual elements by progressively merging clusters. Merging of clusters is driven by a distance between them, referred as *linkage distance.* Usually, the linkage distance is determined either by (a) the maximum distance between elements of each cluster, referred as *complete linkage distance* d^c_L*;* (b) the minimum distance between elements of each cluster, referred as *single linkage distance* d^s_L; (c) the mean distance between elements of each cluster, referred as *average linkage distance* d^a_L; (d) the minimum distance between the cluster prototypes, referred as *prototype linkage distance* d^p_L, among others. In the rest of the paper, we only consider d^c_L, d^s_L, d^a_L, and d^p_L; we use d_L to refer to any of them.

HDCC adapts the hierarchical base algorithm by making work together a generalisation operator and two distances (a linkage distance d_L and a distance *d* between elements in a metric space (X, d)). In each iteration, it merges the last discovered cluster with those clusters covered by its generalisation (or pattern). Each cluster pattern provide a description for the cluster's elements, which are close according to the underlying distance, but also for those elements that –although not close enough to be linked by the distance– are covered by the pattern. To represent the resulting hierarchy of clusters and concepts, HDCC produces an extended dendrogram named *conceptual dendrogram*. A conceptual dendrogram shows not only a hierarchy of clusters, whose elements have been grouped driven by distances d_L and *d*, but it also provides a hierarchy of generalisations or patterns, that give intensional descriptions of the elements of each cluster in the hierarchy. Solid lines in conceptual dendrograms correspond to clusters linked by similarity or proximity in the metric space, while dashed lines corresponds to those clusters merged by generalisation.

Fig. 2 (a) and (b) shows an example of a traditional and the corresponding conceptual dendrogram for the evidence {*aa*, *aab*, *abb*, *aabbbbbb*} using single linkage distance. The elements belong to the metric space (X, d), where X is the set of finite lists of symbols on the alphabet $\Sigma = \{a, b\}$ and *d* is the edit distance or Levenshtein distance [13]. The pattern $p = aa*$ covers the cluster {*aa*, *aab*, *aabbbbbb*}, which has been formed considering, in first place, the distance between clusters and, in second place, the coverage of the resulting pattern $aa*$ under the generalisation operator employed.

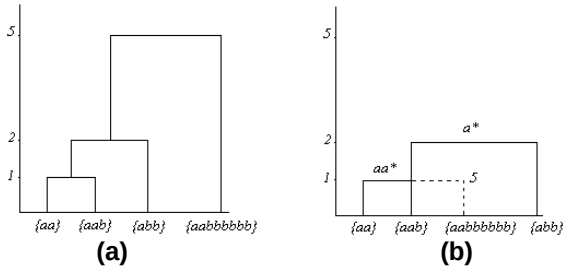


Fig. 2. (a) Traditional and (b) conceptual dendrograms.

3.2. Consistency between Generalisation Operators and Distances.

Usually, inconsistency problems appear when we integrate distance-based and conceptual learning techniques. Distances between elements of a metric space and generalizations of those elements are not always consistent.

To illustrate the problem let us consider again the evidence formed by finite lists of symbols on the alphabet $\Sigma = \{a, b\}$ and the edit distance shown in Fig. 2.

According to the hierarchical clustering algorithm under d^s_L and taking into account the edit distances d between the examples, the first clusters linked by d^s_L are $\{aa\}$ and $\{aab\}$, while the generalisation operator chosen has produced the pattern $aa*$ for cluster $\{aa, aab\}$ (see Fig. 2 (b)). Clearly, there is a metric inconsistency between the elements described by $aa*$ (*aa*, *aab*, *aaa*, *aaba*, *aabb*,…) and the clusters induced by d^s_L and d. Let us note that $aa*$ covers the example *aabbbbbb* but it does not cover the example *abb*, which is closer to the elements in $\{aa, aab\}$ according to the edit distance, as Fig. 3 shows.

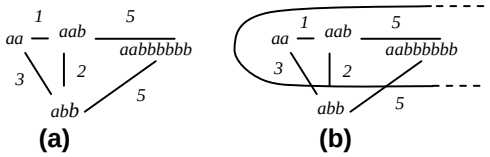


Fig. 3 (a) Edit distances between lists of symbols on alphabet $\Sigma = \{a, b\}$. (b) Coverage of pattern $aa*$.

This inconsistency between distances and generalisations makes differ significantly the shapes that conceptual and traditional dendrograms have. The exact shape of a conceptual dendrogram depends not only on the distance d between the evidence elements and the generalisation operators used but also on the linkage distance d_L employed.

Based on the similarity between conceptual and traditional dendrograms, three different levels of consistency between distances and generalisations have been defined [6][7]. The more similar the dendrograms are the more consistent the distance is with respect to the generalisation operator employed. In the next subsections, we recall these levels and, in Section 4, we present an extension to HDCC framework, in the form of a new property for generalisations operators, which guaranties a minimal level of consistency between distances and generalisations.

3.2.1. Equivalent Dendrograms. In some cases, the conceptual dendrogram is isomorphic to the traditional dendrogram. This situation appears when the discovered patterns cover exclusively clusters that are also linked by the distance. Therefore, a conceptual dendrogram is *equivalent to a traditional dendrogram* if and only if for each cluster C its children are linked at the same distance l. This concept is formalised in Definition 3.

Definition 3. *Let T be the tree resulting from* HDCC. *T is* equivalent to a traditional dendrogram *iff* $\forall$ $(C, p, l) \in T$, $\forall$ (C_1, p_1, l_1),..., (C_n, p_n, l_n) *childs of* (C, p, l): $d_L(C_i, C_j, d) = l, i \neq j$ *and* $i, j = 1, \ldots, n$

To achieve the maximum degree of consistency, i.e. equivalent dendrograms, the generalisation operators must produce patterns such that each time HDCC links two clusters C_1 and C_2 at a minimum linkage distance l, the associated pattern p must not cover any other cluster C whose distances l_1 and l_2 to C_1 and C_2, respectively, are greater than l. Generalisation operators that satisfy this property are *strongly bounded by* d_L.

Intuitively, a pattern binary generalisation operator is strongly bounded by d_L iff for any pair of patterns p_1, p_2 and for any pair of sets C_1 and C_2 covered by *p1* and *p2*, respectively, the linkage distances to C_1 and C_2 from the new elements covered by the generalisation of p_1 and p_2 are equal or lower than the linkage distance between C_1 and C_2, i.e. the new elements covered by the generalisation of p_1 and p_2 fall into the balls of radius $d_L(C_1, C_2, d)$ and centre in the linkage points of C_1 and C_2. The *linkage points* are, in the case of d^s_L, the two closest elements in C_1 and C_2; the two most distant elements in d^c_L; the prototypes in the case d^p_L and the centroids in the case of d^a_L (assuming the metric space is continuous). Strong boundedness property is formalised in Definition 4.

Definition 4. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language and* d_L *a linkage distance. A pattern binary generalisation operator* Δ^* *is* strongly bounded by d_L *iff* $\forall$ $p_1, p_2 \in \mathcal{L}$, $C_1 \subseteq Set(p_1)$, $C_2 \subseteq Set(p_2)$, $C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2))$: $d_L(C, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C, C_2, d) \leq d_L(C_1, C_2, d)$.

Definition 5 formalises the same property for binary generalisation operators, which are used by HDCC to generalise unitary clusters. Given that the linkage distance d_L between unitary sets reduces to the distance d between elements in the metric space, a binary generalisation operator is said to be *strongly bounded by* d (and not by d_L).

Definition 5. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language. A binary generalisation operator* Δ *is* strongly bounded by d *iff* $\forall\ e, e_1, e_2 \in X$*: if* $e \in Set(\Delta(e_1, e_2))$ *then* $d(e, e_1) \leq d(e_1, e_2) \vee d(e, e_2) \leq d(e_1, e_2)$.

The linkage distance d_L employed affects the boundedness property of generalisation operators. Therefore, a generalisation operator could be strongly bounded under a given d_L but not under a different one for the same distance d.

In [6], the authors have proved that strong boundedness property is a sufficient condition for equivalent dendrograms, which is established by Proposition 1.

Proposition 1. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language for* X*,* Δ *a binary generalisation operator,* Δ^* *a pattern binary generalisation operator and* d_L *a linkage distance. For any evidence* $E \subseteq X$*, the conceptual dendrogram* T *resulting from* HDCC$(E, X, d, \Delta^*, \Delta, d_L)$ *is equivalent to the traditional dendrogram if the generalisation operators* Δ *and* Δ^* *are strongly bounded by* d_L.

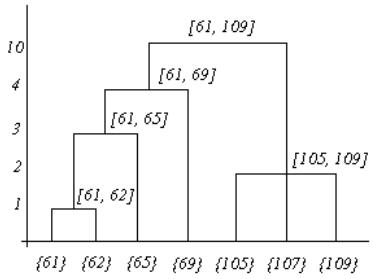


Fig. 4. An equivalent conceptual dendrogram

Fig. 4 shows a conceptual dendrogram for a set of real numbers, which under d^s_L is equivalent to the corresponding traditional dendrogram. $\mathcal{L}$ is the set of the finite closed intervals in $\Re$, and d the absolute difference. $\Delta^*(p_1, p_2)$ is given by the interval $[min, max]$, where min is the minimum value of the lower bounds of p_1 and p_2, and max is the maximum of the upper bounds. $\Delta(e_1, e_2)$ is defined as $[min(e_1, e_2), max(e_1, e_2)]$.

3.2.2. Order-preserving Dendrograms. In some cases, conceptual dendrograms –although not equivalent to the traditional ones– can just preserve the order in which clusters are merged by d_L; that is, the discovered patterns must not cover a farther cluster leaving out a closer one. These conceptual dendrogram are referred as *order-preserving* dendrograms.

Intuitively, an order-preserving conceptual dendrogram is one where for any node (C, p, l) in the hierarchy, its children are linked at the same distance l or they are attracted by p with a linkage distance that is lower than any of the linkage distances to the clusters not covered by p in the hierarchy. This concept is formalised by Definition 6.

Definition 6. *Let* (X, d) *be a metric space and* T *the tree resulting from* HDCC. T *is* order-preserving *iff* $\forall\ (C, p, l), (C_i, p_i, l_i) \in T, \exists\ (C_j, p_j, l_j) \in T$ *with* (C_i, p_i, l_i) *and* (C_j, p_j, l_j) *children of* (C, p, l) *such that* $d_L(C_i, C_j, d) = l \vee (d_L(C_i, C_j, d) < d_L(C', C_i, d) \wedge d_L(C_i, C_j, d) < d_L(C', C_j, d))$*, for all* $(C', p', l') \in T$, $C' \not\subset Set(p)$.

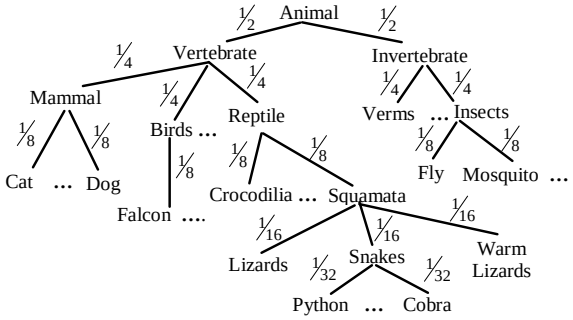
To achieve the order-preserving level of consistency, generalisation operators must produce patterns such that any time HDCC merges two clusters C_1 and C_2, with patterns p_1 and p_2, any other cluster C covered by the generalisation of p_1 and p_2 but not linked by the distance d_L must have lower linkage distances to C_1 and C_2 than the linkage distances to C_1 and C_2 from any other cluster C' not covered by the pattern. This is formalized by the *weak boundedness* property for pattern binary generalisation operators given by Definition 7. Analogously, Definition 8 establishes the same property for binary generalisation operators, while Proposition 2 establishes that weakly boundedness is a sufficient condition for order-preserving dendrograms (see [6] for the proof of Proposition 2).

Definition 7. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language and* d_L *a linkage distance. A pattern binary generalisation operator* Δ^* *is* weakly bounded by d_L *iff* $\forall\ p_1, p_2 \in \mathcal{L}, C_1 \subseteq Set(p_1), C_2 \subseteq Set(p_2), C \subseteq Set(\Delta^*(p_1, p_2)) - (Set(p_1) \cup Set(p_2)), C' \not\subset Set(\Delta^*(p_1, p_2)) : (d_L(C, C_1, d) \leq d_L(C_1, C_2, d) \vee d_L(C, C_2, d) \leq d_L(C_1, C_2, d)) \vee (d_L(C, C_1, d) < d_L(C', C_1, d) \wedge d_L(C, C_2, d) < d_L(C', C_2, d))$.

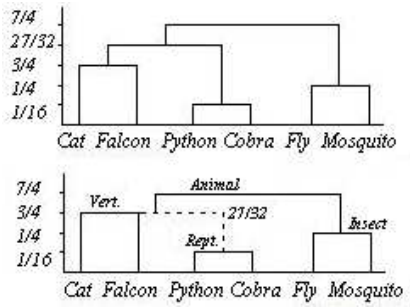
Definition 8. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language. A binary generalisation operator* Δ *is* weakly bounded by d_L *iff* $\forall\ e, e', e_1, e_2 \in X$*: if* $e \in Set(\Delta(e_1, e_2))$ *and* $e' \notin Set(\Delta(e_1, e_2))$ *then* $d(e, e_1) \leq d(e_1, e_2) \vee d(e, e_2) \leq d(e_1, e_2) \vee ((d(e, e_1) < d(e', e_1) \wedge d(e, e_2) < d(e', e_2))$.

Proposition 2. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language,* Δ *a binary generalisation operator,* Δ^* *a pattern binary generalisation operator and* d_L *a linkage distance. For any evidence* $E \subseteq X$*, the conceptual dendrogram* T *resulting from HDCC*$(E, X, d, \Delta^*, \Delta, d_L)$ *is order-preserving if the generalisation operators* Δ *and* Δ^* *are weakly bounded by* d_L.

Fig. 5 (b) shows an example of an order-preserving dendrogram for nominal data using d^S_L. The example applies a distance induced by a relationship R, where R is a partial order. R is defined as xRy if x is a y. Fig. 5 (a) shows part of a relationship R as a tree hierarchy. The distance between two elements is the sum of costs associated to each edge of the shortest path connecting them. The cost of an edge of level i is given by $w_i = 1/2^i$. $\Delta(e_1, e_2)$ is defined as the direct ancestor of e_1 and e_2 if $e_1 \neq e_2$, otherwise is equal to e_1, and $\Delta^*(p_1, p_2)$ is defined analogously. In Fig. 5 (b) we can see the traditional dendrogram and the corresponding conceptual dendrogram. The evidence is formed only by elements in the leaves of R. The internal nodes of R define the language of patterns.



(a)



(b)

Fig. 5. (a) Relationship R as a tree. (b) Traditional dendrogram and the corresponding order-preserving conceptual dendrogram

3.2.3. Acceptable Dendrograms. Some generalisation operators –although not (weakly) bounded by a distance– lead to dendrograms which are consistent with the distance in a broader sense. The rational behind this is that a pattern must not cover new elements whose distance to the old elements is greater than the greatest distance between the old elements. Operators that produce this kind of patterns as well as dendrograms resulting of their application are referred as *acceptable*. Definition 9 formalise this concept for Δ^*.

Definition 9. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language and* d^c_L *the complete linkage distance. A pattern binary generalisation operator* Δ^* *is* acceptable *iff* $\forall\ p_1, p_2 \in \mathcal{L}, e \in Set(\Delta^*(p_1, p_2)), \exists\ e' \in Set(p_1) \cup Set(p_2) : d(e, e') \leq d^c_L(Set(p_1), Set(p_2), d)$.

Acceptability property is independent of d_L. It only depends on the distance d, specifically on $max\{\ d(x, y) : x \in Set(p1),\ y \in Set(p2)\ \}$. To simplify the notation $d^c_L(Set(p_1),Set(p_2),\ d)$ has been used in Definition 9 instead of $max\{\ d(x, y) : x \in Set(p1), y \in Set(p2)\ \}$.

Fig. 6 shows the union of the circles, whit centres in the complete linkage points taken from $Set(p_1)$ and $Set(p_2)$, respectively, and radius $d^c_L(Set(p_1),\ Set(p_2),\ d)$, that determines the maximum coverage for the generalisation of *p1* and *p2* produced by an acceptable generalisation operator. In this example, the complete linkage points are given by a and d taken from the coverage of the two patterns (minimum parallel axis rectangles) shown in the figure for the evidence $\{a, b, c, d\}$ in $\Re^2$.

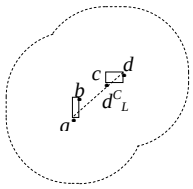


Fig. 6. Maximum coverage for an acceptable pattern

The acceptability property for binary generalisation operators leads to the same condition for strongly bounded binary generalisation operators (see Definition 5). Therefore, *acceptable dendrograms* are those dendrograms resulting from the application of a pair of generalisation operators (Δ, Δ^*) that are strongly bounded and acceptable, respectively.

In this level of consistency, conceptual and traditional dendrograms can significantly differ.

4. A New Level of Consistency: Distance-based Dendrograms

An interesting concept is introduced by Estruch in [3]. In that work, the author discusses the problem of inconsistency between generalisation operators and distances, defines a framework where distance-based and symbolic paradigms in Machine Learning can be integrated in a consistent way, and formalise the concept of distance-based generalisation, which can be intuitively seen as a pattern that connects all the generalised examples in such a way that any of the generalised examples can be reached from any other through the distance by moving inside the generalisation.

In order to build up the concept, the author introduce the notion of *connection*, meaning that when a pattern generalises a pair of elements *e1* and *e2*, we can go from one element to another through similar (close) elements without leaving out of the generalisation. For instance, if we generalise the set of integers number {2, 10}, we expect to go from 2 to 10 through the path $2 \rightarrow 3 \rightarrow ... \rightarrow 9 \rightarrow 10$ without leaving the generalisation. This means that, inside the metric space, we can gradually move from one element of the metric space to another passing through elements that are also covered by the generalisation.

Intuitively, the idea behind Estruch's concept of distance-based generalisation is that the elements in between (in terms of the distance) should be included in the generalisation. However, not any path should be considered, but a generalisation should include short-stepped paths which connect the elements that lead to the generalisation. This means that all the elements in the shortest path between two generalised elements (in many metric spaces, the straight segment which connects them) must be included in the generalisation.

The concept of *intermediate element* in a metric space is obtained by turning the inequality into equality in the triangle inequality property of metric space: given a metric space (X, d) and two elements $x, z \in X$, an element $y \in X$ is *between* x and z, or is an *intermediate element* w.r.t. d, if $d(x, z) = d(x, y) + d(y, z)$.

The author also extends this idea to sets of examples, leading to the notion of *nerve* and *skeleton*, which are used to define *distance-based generalisation operators*.

Intuitively, given a set E of evidence, the *nerve* of E is a connected graph η where each element of E is a vertex in η and vice versa.

The *skeleton* of E with respect to a nerve η of E is the set of elements in X such that are in between of any pair of directly linked vertex in η.

Finally, a generalisation operator is *distance-based* if for any set of evidence E, there is a nerve η connecting the points in E such that the skeleton of η is included in the generalisation of E. In other words, the generalisation must cover all the points in the evidence and the elements in between.

The question is why we want to have the intermediate elements in the generalisation. To answer this question let us consider a pair of sequences *abb* and *acc* and the edit distance. Two possible generalisations of these sequences are patterns **bb+*cc* and **a**. However, we prefer **a** since we can go from *abb* to *acc* through the path $abb \rightarrow_1 ab \rightarrow_1 a \rightarrow_1 ca \rightarrow_1 cca$ which is smother or more direct that the path $abb \rightarrow_1 acbb \rightarrow_1 accbb \rightarrow_2 acc$. As we can see in the second path, the intermediate element *accb* between *accbb* and *acc* is not included in the generalisation **bb+*cc*, leading to a step with a greater distance or forcing us to go out of the generalisation.

If we analyse the lower level of consistency in HDCC framework, we can see that acceptable binary generalisation operators give place to generalisations whose elements in the coverage must be at a distance from the generalised elements that is lesser or equal to the distance between the generalised elements. However, if we take a closer look to the definition of acceptable generalisation operators, we can see that it also permits generalisations that do not cover elements that are "in between" and also close to the generalised elements. To illustrate this problem, let us consider for instance the pattern $[1, 3] \cup [9, 11]$ resulting from the generalisation of the evidence {2, 10} in the metric space of integers and with the absolute difference as distance. Let us note that this generalisation satisfies the acceptability condition: any elements in the coverage of $\Delta(2, 10) = Set([1, 3] \cup [9, 11]) = \{1, 2, 3, 9, 10, 11\}$ has a distance to at least one of the generalised elements that is lesser than the distance between them ($d(2, 10) = 8$). However, we can see that some elements between 2 and 10 are outside the coverage of the pattern. It does not seem to be reasonable consider as consistent with the distance patterns that excludes elements in between of the generalised evidence given that these elements are precisely which make up the distance between the generalised evidence.

Applying Estruch's ideas to HDCC generalisation operators Δ^* and Δ is quite direct. We have named this new level of consistency as the level of *distance-based dendrograms*. Next we give the corresponding definitions.

Definition 10 establishes the sufficient and necessary condition for distance-based binary generalisation operators. Informally, a binary generalisation operator Δ is distance-based if and only if for any pair of points in a metric space, any point in between must be included in the generalisation of the two points.

Definition 10. *Let* (X, d) *be a metric space. A binary generalisation operator* Δ *is* distance-based *iff* $\forall\ e_1, e_2, e \in X$: *if* $d(e_1, e) + d(e, e_2) = d(e_1, e_2)$ *then* $e \in Set(\Delta(e_1, e_2))$.

Let us consider the generalisations shown in Fig. 1 (a) for points a and b in the metric space $(\Re^2, d)$, with d the Euclidean distance. Only G3, G4 and G5 are distance-based with respect to the distance d. In G1 and G2 some elements in between a and b are excluded from the generalisations. Given that we are using the Euclidean distance, the elements between a and b correspond to the shortest segment in the metric space that connects a with b. However, if we consider a different metric space, for

example the one defined in $\Re^2$ by the Manhattan distance[1], the elements between *a* and *b* are placed in the rectangle delimited by *a* and *b*. Therefore, the only generalisation in Fig. 1 (a) that is distance-based with respect to the Manhattan distance is G4, which corresponds precisely to the rectangle delimited by *a* and *b*.

We have also extended this concept to pattern binary generalisation operators. Informally, we could say that a pattern binary generalisation operator Δ^* is distance-based if given a pair of patterns p_1, p_2 for two clusters C_1 and C_2 with nerves η_1 and η_2, the skeleton of the nerve resulting from the union of η_1 and η_2 and the addition of an edge connecting the linkage points of C_1 and C_2 is covered by the generalisation of p_1 and p_2. However, if we take a closer look to this concept we can see that it is enough to ask only for the coverage of the elements in between of the linkage points. The coverage of the skeleton associated to the union of the nerves η_1 and η_2 is given by the definition of pattern binary generalisation operator (a pattern binary generalisation must cover at least the elements in the coverage of the patterns that it generalises). Based on this idea, Definition 11 formally defines distance-based pattern binary generalisation operators.

Definition 11. *Let* (X, d) *be a metric space,* $\mathcal{L}$ *a pattern language and* d_L *a linkage distance. A pattern binary generalisation operator* Δ^* *is* distance-based *iff* $\forall\ p_1, p_2 \in \mathcal{L}, C_1 \subseteq Set(p_1), C_2 \subseteq Set(p_2), e \in X$: *if* $d(e_1, e) + d(e, e_2) = d(e_1, e_2)$ *then* $e \in Set(\Delta^*(p_1, p_2))$, with e_1 and $e2$ *the linkage points in* $d_L(C_1, C_2)$.

Fig. 7 shows different generalisations of patterns on the same evidence {a, b, c, d, e, f} in $\Re^2$ and using the Euclidean distance. In Fig. 7 (a), the pattern language we have adopted is the set of axis-parallel rectangles in $\Re^2$. The figure shows two patterns p_1 and p_2 and the generalisation of p_1 and p_2 given by pattern generalisation operator $\Delta^*(p_1, p_2)$, which corresponds to the biggest rectangle shown in the figure. Graphically, we can see that Δ^* is distance-based given that all the points on the straight line connecting the linkage points (whatever linkage distance used) are included in the region confined by the two dotted lines and hence in the generalisation of $p_1$and p_2, i.e. the biggest rectangle.

In Fig. 7 (b), we have used a different pattern language. In this case, patterns are unions of axis-parallel rectangles. The figure shows two patterns p_1 and p_2, both consisting of the union of two rectangles. The generalisation of p_1 and p_2, $\Delta^*(p_1, p_2)$, is given by the union of all the rectangles in the figure (included the dotted-line rectangle). In this example, we have considered the single linkage points *e* and *a*. As we can see in the figure, the points in between *a* and *e*, described by the dotted straight segment connecting *a* and *e* is included in the dotted-line rectangle, which is part of the resulting generalisation $\Delta^*(p_1, p_2)$.

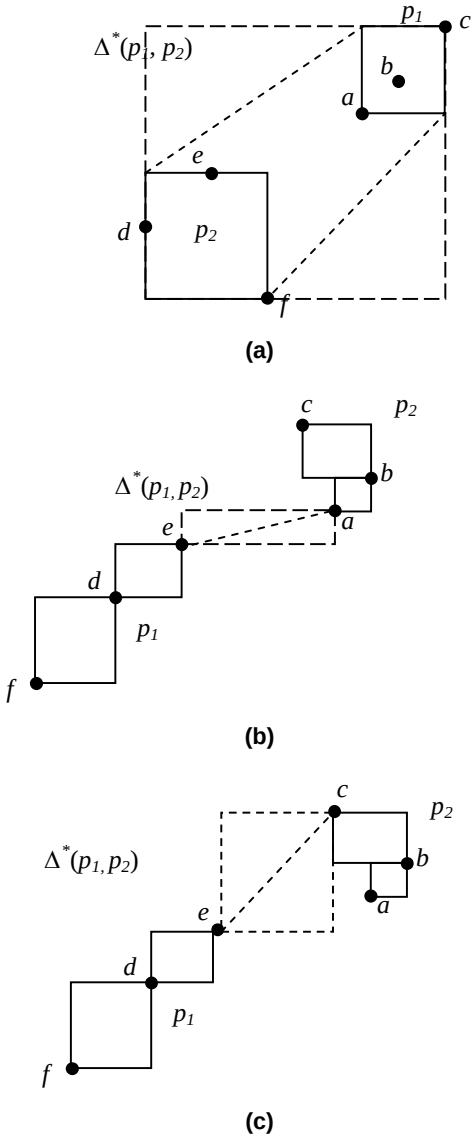


Fig. 7. Examples of distance-based generalisations of patterns in $\Re^2$

In Fig. 7 (c) we have changed to the complete linkage distance. The complete linkage points in this case are given by *e* and *c*. The points in between them are placed on the dotted-line rectangle, which is included in the coverage of $\Delta^*(p_1, p_2)$.

If we consider the Manhattan distance for the three examples shown in Fig. 7, the pattern binary

[1] Given two tuples of real numbers $(a_1, ..., a_n)$ and $(b_1, ...,b_n)$ the Manhattan distance $d = \sum |a_i - b_i|$ $i= 1,...,n$

generalisation operators Δ^* used are also distance-based given that all the elements between the linkage points describes rectangles that are included in the respective generalisations.

Finally, we say that a *conceptual dendrogram* is *distance-based* if it is the result of the application of two distance-based generalisation operators Δ^* and Δ with respect to the distances d and d_L.

An important consequence of using distance-based generalisation operators in HDCC is the existence of pairs of operators and distances that, working together, produce patterns that are distance-based.

In effect, several data types (both structured and non-structured data) have been analysed [3], leaving open the possibility of using such pairs of operators and distances. In that work, we can found consistent combinations for numerical data, nominal data, sets, lists, tuples, graphs, atoms and clauses.

5. Conclusions

In this work we have presented a contribution to HDCC theoretical framework in the way of a new consistency level between distances and generalisations.

HDCC framework allows us to instantiate HDCC algorithm for different data types in a consistent way, meaning that we can transform the output of a traditional distance-based hierarchical clustering algorithm into symbolic based, preserving the consistency between patterns and distances, as long as we have a compatible pair of distance and generalisation operators for the data type involved.

HDCC framework provides us with the theoretical tools necessary to design consistent instances of HDCC algorithm. It means that we can a priori make a theoretical analyse of the level of consistency reached by any pair of generalisation operators and distances.

The new level of consistency we have presented in this work and that we have called the level of distance-based dendrograms is based in the proposal of Estruch's work [3]. In such work, the author analyses the relationship between distances and generalisations and proposes a framework where symbolic and distance-based learning methods can be integrated in a consistent way.

The rational behind distance-based generalisations is the intrisicability property, which guaranties the existence of a straight path of intermediate elements in the metric space connecting the evidence and that are also included in the generalisation. Therefore, we can go, driven by the underling distance, from one element of the evidence into another by moving through a path of neighbour elements without going out of the generalisation.

Although dendrograms resulting from the application of distance-based generalisation operators can differ with respect to traditional dendrograms, as also acceptable dendrograms do, they satisfy an important property (the intrisicability) that is not necessarily meet by acceptable generalisations.

Finally, another contribution of this work is the addition of new consistent combinations of operators and distances for numerical data, nominal data, sets, lists, tuples, graphs, atoms and clauses, from the theoretical results given in [3], to the already available in HDCC framework: strongly bounded operator for nominal data and numerical data with the discrete and difference distances, respectively and for any linkage distance used; the composability of tuples of both nominal and numerical data [7][8], and a pair of distance and generalisation operators that works consistently for sequences [9].

As part of our future work, we plan to study the existing relationships between the three levels of consistency given in HDCC framework and the new level here proposed for distance-based dendrograms. We also plan to work into the establishment of new operative combinations of distances and generalisation operators.

6. References

[1] Berkhin, P. "A Survey of Clustering Data Mining Techniques", *Grouping Multidimensional Data*, pp. 25-71, Springer (2006).

[2] Cover, T and Hart, P. Nearest neighbour pattern classification, IEEE Transactions on Information Theory, pp. 13-27. (1967).

[3] Estruch, V. Bridging the gap between distance and generalisation:Symbolic learning in metric spaces. PhD thesis, DSIC-UPV, http://www.dsic.upv.es/~vestruch/thesis.pdf, (2008).

[4] P. Domingos. Unifying instance-based and rule-based induction. Machine Learning, 24(2):141{168, (1996).

[5] Fisher, D. "Knowledge Acquisition Via Incremental Conceptual Clustering", *Machine Learning* 2: 139-172, Kluwer Acadenfic Publishers. (1987).

[6] Funes, A., Ferri, C., Hernández-Orallo, J., Ramirez-Quintana, M.J.: Hierarchical Distancebased Conceptual Clustering. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 349–364. Springer, Heidelberg (2008).

[7] Funes, A.: Agrupamiento Conceptual Jerárquico Basado en Distancias, Definición e Instanciación para el Caso Proposicional. Master Thesis, DSIC-UPV (2008).

[8] Funes, A., Ferri, C. , Hernández-Orallo, J., Ramírez-Quintana, M. J.: An Instantiation of Hierarchical Distance-based Conceptual Clustering for Propositional Learning.

LNAI 5476, pp. 637–646. Springer-Verlag Berlin Heidelberg (2009).

[9] Funes, A., Ferri, C. , Hernández-Orallo, J., Ramírez-Quintana, M. J.: An Instantiation for Sequences of Hierarchical Distance-based Conceptual Clustering, Proceedings of JAIIO 2011, Córdoba, Argentina. (2011).

[10] A.R. Golding and P.S. Rosenbloom. Improving rule-based systems through case-based reasoning. In *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*, pages 759-764. Morgan Kaufmann, 1993.

[11] G. Góra and A. Wojna. Riona: A classifier combining rule induction and k-NN method with automated selection of optimal neighbourhood. In Proc. of the 13th European Connference on Machine Learning (ECML'02), LNCS, pages 111{123. Springer, (2002).

[12] Jain, A. K., Murty, M. N. and Flynn, P. J., "Data clustering: a review", *ACM Comput. Survey*, Vol. 31, Nº 3, pp. 264-323, (1999).

[13] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*. 10:707–710. (1966).

[14] MacQueen, J. B. Some methods for classification and analysis of multivariate observations, in Proc. of the 5th Berkeley Symposium on Math. Statistics and Probability, pp. 281–297, University of California Press. (1967).

[15] Michalski, R. S. "Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts", *Policy Analysis and Information Systems*, Vol. 4, Nº 3, pp. 219-244. (1980)

[16] Michalski, R. S. and Stepp, R. E. "Learning from Observation: Conceptual Clustering", in Michalski et al (eds.) *Machine Learning: An Artificial Intelligence Approach*, pp. 331-363. TIOGA Publishing Co. (1983)

[17] G. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.

[18] Ramon, J., Bruynooghe, M. and Van Laer, W. Distance measures between atoms. *CompulogNet Meeting ComputingLogic & Machine Learning*, pp. 35–41. (1998).

[19] S. Salzberg. A nearest hyperrectangle learning method. Machine Learning, 6:251-276,(1991).

[20] Talavera, L., Béjar, J.: Generality-Based Conceptual Clustering with Probabilistic Concepts. IEEE Transactions on Pattern Analysis & Machine Inteligence 23(2) (2001)

Aplicación de técnicas y estrategias de Inteligencia de Negocio (BI) Sobre datos académicos integrados de la Facultad de Informática de la UNLP. Análisis la deserción.

Díaz, Javier; Osorio, Alejandra; Amadeo, Paola; Kruzylko Ostojic, Claudia
Universidad Nacional de La Plata, Facultad de Informática

Abstract

El objetivo de este trabajo es la aplicación de tareas y técnicas de minería de datos, sobre los datos de los alumnos y egresados de la Facultad de Informática – UNLP utilizando la herramienta de análisis de datos Weka; para poder determinar si existen patrones de comportamiento en los mismos. Se analizan estos datos desde dos puntos de vista diferentes, datos socio-demográficos obtenidos de un solo sistema y las actividades que los estudiantes realizan con el objetivo de aprobar una materia, correlacionando datos obtenidos de diferentes sistemas que se utilizan en la facultad: Moodle, SIU Guaraní y Meran, con el fin de obtener un análisis más integrado. Se puede observar que existe una relación entre el nivel de estudios de los padres de los estudiantes con la situación que los mismos tienen dentro de la Facultad, siendo los estudiantes más activos, aquellos cuyos padres tienen un nivel de estudios secundario o superior. También se puedo observar que existe una relación entre el uso de los foros del sistema Moodle y el resultado que los estudiantes obtienen en las materias, cuanto más es el uso de los foros del sistema, más es la cantidad de estudiante aprobados que se tiene. Otra información obtenida de los análisis es que aquellos alumnos que utilizan más la biblioteca generalmente obtienen un resultado satisfactorio. Poder contar con datos correlacionados ayuda a realizar un seguimiento del comportamiento de los estudiantes en cada una de las etapas de la carrera, así como también plantear diferentes tipos de análisis e integración con otros sistemas, por ejemplo las redes sociales.

1. Introducción

En la Facultad de Informática[1] de la Universidad Nacional de La Plata[2] los sistemas informáticos que los alumnos pueden utilizar desde que ingresan a la Facultad hasta que obtienen el diploma, son entre otros, sistemas de ingreso, sistema de alumnos, becas, libretas sanitarias, sistemas de biblioteca, entornos virtuales. El uso de estos sistemas genera numerosos registros, que contienen datos de los estudiantes, lo que permitiría el "tracking de los alumnos", un seguimiento del comportamiento de los estudiantes a lo largo de la carrera. No contar con un proceso para el análisis de estos datos, dificulta muchas veces la toma de decisiones. Poder identificar los diferentes perfiles de los estudiantes y egresados, ayudaría a comprender mejor su comportamiento e implementar propuestas educativas que los ayuden a completar sus estudios, evitando así la deserción de la carrera, así como también poder sugerir distintos tramos para el aprendizaje, entre otras posibilidades.

La minería de datos educativa, EDM por sus siglas en inglés, es una disciplina dedicada a la extracción de conocimiento de tipos particulares de datos de provienen de los centros educativos. Calders y Pechenizkiy [3], la definen como "*un área de investigación multidisciplinaria emergente, en la cual se han desarrollado métodos y técnicas para la exploración de datos originados de varios sistemas de información educativos. La minería de datos educativa es tanto una ciencia de aprendizaje como un área de aplicación rica para la minería de datos, debido a la creciente disponibilidad de datos provenientes del sector educativo. Contribuye al estudio de cómo aprenden los estudiantes y los contextos en que lo hacen. Permite la toma de decisiones para la mejora de las prácticas educativas actuales y el material de estudio*".

Propone metodologías que puedan ayudar tanto a administrativos, docentes y alumnos a determinar aquellos casos de bajo nivel para poder tomar medidas apropiadas a tiempo y evitar así, la deserción a nivel universitario. Vemos, en los trabajos de investigaciones consultados [4] [5] [6] [7], que la tarea de minería de datos más frecuente, es la clasificación, siendo árboles de decisión el método más usado, aunque también se usan reglas de clasificación, clustering y redes bayesianas. Otra tarea aplicada con frecuencia, es el agrupamiento y una práctica muy usada es la aplicación de más de una técnica de minería de datos sobre el mismo conjunto de datos, para poder luego comparar la eficiencia de cada uno.

En el trabajo realizado que se describe en el presente artículo, se aplicaron las técnicas de minería de datos arboles de decisión y agrupamiento sobre el mismo conjunto de datos obtenidos de distintas fuentes sobre el mismo sujeto estudiante. Ambas técnicas se aplican sobre

datos concretos y se analizan y comparan los resultados obtenidos para producir así conocimiento basado en hechos concretos. El objetivo principal del trabajo fue la realización del estudio de los datos de los estudiantes mediante la aplicación de estrategias de minería de datos, no la innovación en el uso de alguna metodología nueva o combinación de éstas; es por eso que se utilizaron los métodos y estrategias ya conocidas de investigaciones anteriores.

Respecto a los sistemas y herramientas de minería de datos que se encuentran actualmente en el mercado, se utiliza WEKA –Waikato Environment for Knowledge Analysis – [9]. Es una herramienta visual de libre distribución (licencia GNU) desarrollada en JAVA, por un equipo de investigadores de la universidad de Waikato (Nueva Zelanda). En un inicio era sólo una librería, pero hoy en día es un paquete integrado (suite). Al ser software libre, es actualmente un producto maduro con una rica comunidad de desarrolladores que la promueven como uno de los productos más utilizados para realizar minería o integrar sus librerías con otros sistemas para realizar desarrollos ad-hoc.

2. Elementos del Trabajo y Metodología

2.1. Fuentes de Datos

Los datos que se analizarán en este trabajo serán obtenidos de las bases de datos de los sistemas basados en software libre que se utilizan en la Facultad de Informática y registran los datos de sus estudiantes y egresados: SIU Guaraní [10], Meran [11] y Moodle [12].

3. Resultados y Discusión

La selección de productos de código abierto facilita mediante bibliografía publicada o la facilidad de acceso, el estudio de la estructura de su base de datos. El sistema SIU-Guaraní es el único de los utilizados cuyo código no está distribuido bajo una licencia de software libre; sin embargo, se distribuye en forma libre en todas las instituciones educativas de nivel superior de Argentina, que tengan la intención de usarlo. La integración de los datos entre estos tres sistemas es posible debido a que todos ellos dialogan a través de una integración llevada a cabo por el CeSPI a partir de las necesidades de la institución. De esta manera se pueden identificar unívocamente al estudiante a través de su DNI y el legajo del mismo. Si bien esta información se mantiene oculta en los análisis realizados, son de gran ayuda en la correlación de los datos. De la misma manera que se identifica al estudiante, es posible identificar en los tres sistemas a las materias mediante un código o el nombre de la misma.

3.1. Análisis socio demográfico

En esta sección, se presenta el análisis de los datos desde el punto de vista de factores sociales y demográficos, para intentar establecer si existe una relación entre éstos y la situación académica de los estudiantes.

En el histograma de la figura 1 que se presenta a continuación, se puede observar la distribución de los atributos respecto al atributo Situación del Estudiante.

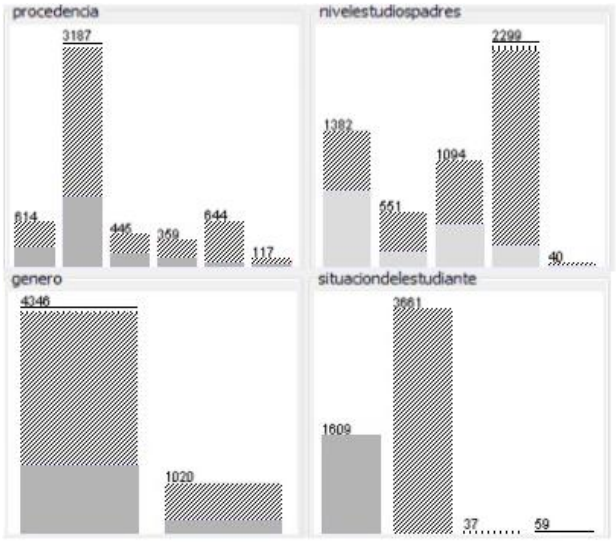


Figura 1. "Distribución de los Datos Situación del Estudiante"

El atributo situación del aspirante será la clase a predecir en la tarea de clasificación.

En la siguiente tabla se indican los valores de los atributos, de izquierda a derecha.

Tabla 1 – Análisis socio demográfico

Atributos	Posibles valores (de izquierda a derecha según la ubicación en los histogramas)
Procedencia	Interior de la provincia de Buenos Aires; La Plata y alrededores; Interior del país; Gran La Plata; Sin Datos; Exterior.
Nivel estudio de los padres	Universitarios; Primarios; Secundarios; Sin estudios; Sin datos.
Género	Masculino; Femenino.
Situación del estudiante	Activo; Pasivo; Egresado en término; Egresado fuera de término.

En la distribución de los estudiantes según las tramas, en cada una de las barras de los atributos procedencia,

nivel estudio de los padres y género es, desde la base de la barra hacia arriba: Activo (liso), Pasivo (rayado), Egresado en término (línea punteada) y Egresado fuera de término (línea completa).

Las clases no están balanceadas, siendo P-Pasivo la clase mayoritaria. Ningún atributo separa bien las clases. Los histogramas muestran, una distribución más equilibrada de estudiantes activos y pasivos, en los procedentes del interior de la provincia de Buenos Aires y del interior de país. En las demás procedencias, se observa una clara diferencia con más estudiantes pasivos que activos. También la distribución entre estudiantes activos y pasivos es más pareja para los niveles de estudio de los padres más altos –universitarios y secundarios –, habiendo en los demás valores una clara diferencia de pasivos frente a activos.

La clasificación a través de árboles de decisión se realiza utilizando el algoritmo J48 con diferentes valores en el parámetro de factor de confianza de la poda. Utilizando el valor por defecto en el factor de confianza el clasificador obtenido tiene un 71.52% de precisión. La razón de aciertos de la clase P – Pasivo, es de 0.843, y la razón de aciertos de la clase A – Activo es de 0.462. Modificando el valor del factor de confianza se obtienen distintos modelos, todos ellos permiten observar una relación entre el nivel de estudios de los padres y la situación del estudiante, siendo más activos aquellos estudiantes cuyos padres poseen estudios secundarios o universitarios.

La clasificación a través de la tarea de agrupamiento utilizando el algoritmo de SimpleKMeans utilizando 5 grupos permitió obtener clusters que contienen, en su mayoría, a estudiantes procedentes de la provincia de Buenos Aires, más o menos cercanos a la Facultad. Podemos concluir, con esta distribución que es muy alto el porcentaje de estudiantes que cursan en la Facultad de Informática procedentes de estas localidades, con distinto nivel de estudio de los padres. Los estudiantes pasivos y los que tienen padres sin estudios realizados, se agrupan en su mayoría juntos, en éste caso en el cluster 0. También se agrupan juntos los estudiantes egresados y activos cuyos padres tienen estudios universitarios o superiores. Por último, podemos ver, cómo en un mismo cluster se destacan los estudiantes activos cuyos padres tienen estudios secundarios –cluster 3 –.

Además, podemos señalar que los estudiantes que no son activos, en su mayoría se encuentran en un mismo cluster sin importar la procedencia de los mismos, en este caso el cluster 0; pero los estudiantes activos están más divididos entre los demás clusters.

3.2. Análisis de la participación de las materias

En esta sección se presenta el análisis de los datos referentes al comportamiento del estudiante en las materias que cursa, que lo lleve a aprobar la misma. Los datos obtenidos para éste análisis fueron correlacionados de diferentes sistemas utilizados por la Facultad de Informática para almacenar los datos de sus estudiantes.

En los histogramas de la Figura 2 – Distribución de los datos, podemos ver la distribución de los atributos con respecto al atributo clase. El atributo m_aprobado, será la clase a predecir en la tarea de clasificación.



Figura 2. "Distribución del atributo aprobado sobre los demás atributos. Materia Aprobada"

En la siguiente tabla se indican, de izquierda a derecha según la imagen, los posibles valores que pueden tomar los atributos.

Tabla 2 – Análisis por materia aprobada

Atributos	Posibles valores (de izquierda a derecha según la ubicación en los histogramas)
participacion_foro	Medio, Alto, Bajo
rta_compañeros	Bajo, Medio, Alto
rta_docentes	Bajo, Medio, Alto
tiempo_foro	Medio, Alto, Bajo
tiempo_materia	Medio, Alto, Bajo
conversaciones_iniciadas	Medio, Bajo, Alto
uso_biblioteca	Sí, No
m_cursada	Aprobado(1) Desaprobado(0)

En la distribución de los estudiantes según la gama de colores, en cada una de las barras de los atributos participación_foro, rta_compañeros, rtas_docentes,

tiempo_foro, tiempo_materia, conversaciones_iniciadas, uso_biblioteca es, desde la base de la barra hacia arriba: Aprobado (gris claro), Desaprobado (gris oscuro). Las clases no están balanceadas, siendo Desaprobado la clase mayoritaria. Ningún atributo separa bien las clases. En los histogramas vemos, cuanta más participación tiene el estudiante en los foros – participación_foro –, cuanto más tiempo emplea en la navegación y uso de los foros y mayor tiempo emplea en las actividades virtuales de las materias en general, – tiempo_foro y tiempo_materia respectivamente –, mayor es la cantidad de estudiantes aprobados. Se observa también, que una participación activa en los foros, como por ejemplo, la cantidad de conversaciones iniciadas o posts realizados, no influye en el resultado de la materia. En cuanto al uso de bibliotecas, la cantidad de estudiantes que hicieron uso de la biblioteca es muy inferior en comparación a aquellos que no hicieron uso de la misma.

La clasificación a través de árboles de decisión utilizando el algoritmo J48 con diferentes valores en el parámetro de factor de confianza de la poda generan árboles que numerosos nodos, alcanzando una precisión máxima de 68.06% con el factor por defecto. La razón de aciertos de la clase Desaprobado es de 0.76, y de aciertos de la clase Aprobado es de 0.576. El modelo generado, indica que el atributo que mejor clasifica a los estudiantes es el tiempo total de uso del sistema Moodle relacionado con la materia. Clasifica con la clase Aprobado a aquellos estudiantes que tienen un tiempo de uso del sistema Moodle alto. La rama del árbol que hace referencia al uso Medio del sistema Moodle, es la más compleja que se formó en este clasificador; ya que se tienen en cuenta luego, el uso activo y pasivo de los foros, y el uso de bibliotecas del estudiante. Los estudiantes con tiempo de uso Bajo del sistema Moodle, son clasificados dependiendo de la participación activa que tienen en los foros; se clasifican con la clase Aprobado a los estudiantes que tienen una participación activa alta o media y se clasifican con la clase desaprobados los que tienen una participación activa baja. Se observa también que en la mayoría de las ramificaciones del árbol generado, en que se evalúa el uso de bibliotecas, si el estudiante hace uso de bibliografía de la materia proveniente de la biblioteca, es clasificado con la clase Aprobado. La modificación del parámetro de confianza de la poda sacrifica precisión o genera árboles más complejos sin lograr una mejora significativa en la precisión.

Luego, se realizarán diferentes ejecuciones del algoritmo SimpleKMeans para la tarea de agrupamiento, formando diferentes cantidades de grupos para comparar los resultados entre ellos.

La ejecución del algoritmo para generar cinco permite observar que los estudiantes con nivel de actividad media en el foro se encuentran, en su mayoría en un mismo cluster –cluster 1. Los estudiantes que tienen participación alta en el foro también se agrupan en un mismo cluster –cluster 3. Los estudiantes con participación baja en el foro, se dividen en dos grupos, en uno se encuentran los aprobados –cluster 2 y en otro se encuentran los estudiantes desaprobados, –cluster 0.

En una siguiente iteración, no se observa una separación de estudiantes entre los diferentes clusters tan clara teniendo una división un poco más compleja de analizar. La mayoría de los estudiantes aprobados se encuentran en el cluster 3, pero el uso de biblioteca parece no afectar la nota final obtenida en la materia. La mayoría de los estudiantes desaprobados se encuentran en el cluster 0 y el uso de biblioteca tampoco parece afectar la nota obtenida en la materia cursada.

4. Análisis de las Redes Sociales

Actualmente se pueden obtener datos referentes a los alumnos y egresados, de sistemas externos a la Facultad, como pueden ser las redes sociales tan, ampliamente usadas en la actualidad. Aprovechando la difusión de información que permiten y el gran uso de las mismas; algunas cátedras de la Facultad, comenzaron a hacer uso de las mismas para comunicarse con sus estudiantes. En particular, los docentes de la materia Tecnologías aplicadas para Business Intelligence utilizan tanto la red social Facebook como Twitter como medio de comunicación secundario con sus estudiantes.

A la vista minable creada para el análisis de la participación de los estudiantes en las materias, se filtraron aquellos registros pertenecientes a la materia TABI y se agregó información que indica si el estudiante es seguidor o no de la cuenta de Twitter de la cátedra y si es amigo de la página de Facebook que la materia posee.

La información de los seguidores de Twitter de la cuenta de la cátedra, se consiguió mediante el sistema simplymeasured[13]. La información de los amigos de la página de Facebook se consiguió mediante el sistema Netvizz[14] junto con el sistema Ghephi[15].

Una vez armada la vista minable con esta información, se realizó el análisis de estos datos de manera similar a los dos análisis descriptos en los apartados anteriores, agregándole los atributos Twitter y Facebook que indican si los estudiantes usan estas redes sociales para comunicación con los integrantes de la materia. En la figura 3 se presenta la distribución de los atributos con respecto a la clase. El atributo m_cursada situación del estudiante será la clase a predecir en la tarea de clasificación.

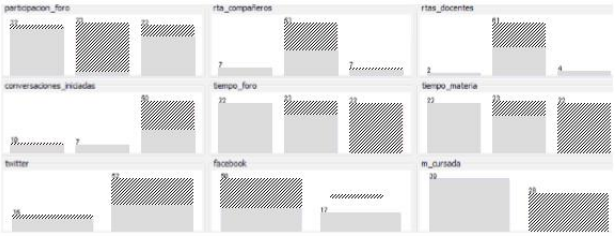


Figura 3. "Distribución del atributo aprobado sobre los demás atributos. Redes Sociales"

A continuación se listan los valores posibles, de izquierda a derecha, de cada uno de los atributos.

Tabla 3 – Análisis por uso de las Redes Sociales

Atributos	Posibles valores (de izquierda a derecha según la ubicación en los histogramas)
participacion_foro	Alto - Bajo - Medio
rta_compañeros	Alto – Bajo - Medio
rtas_docentes	Alto – Bajo – Medio
conversaciones_iniciadas	Medio – Alto – Bajo
tiempo_foro	Alto - Medio - Bajo
tiempo_materia	Alto – Medio – Bajo
Twitter	Sí - No
Facebook	No – Si
m_cursada	Si – No

En la distribución de los estudiantes según la gama de colores, en cada una de las barras de los atributos participación_foro, rta_compañeros, rtas_docentes, tiempo_foro, tiempo_materia, conversaciones_iniciadas, Twitter, Facebook es, desde la base de la barra hacia arriba: Aprobado (gris claro), Desaprobado (gris oscuro) .

Haciendo un análisis de esta distribución de los datos, podemos ver que el uso de las redes sociales no es una elección de comunicación masiva por parte de los estudiantes; ya que más del 50% de los mismos elige no seguir a la cuenta de Twitter de la materia al igual que ser amigos por medio de Facebook. También podemos ver que la mayoría de los estudiantes que sí eligen este medio de comunicación, tienen un resultado exitoso en la materia. Los atributos en este análisis hacen una mejor separación de las clases en comparación con los análisis descriptos en los apartados anteriores. Por ejemplo, tiempo_foro y tiempo_materia separan casi perfectamente las clases; ya que si los valores de estos atributos son Alto o Medio, es muy probable que la clase sea Aprobado, mientras que si el valor es Bajo, la clase es Desaprobado.

El análisis con los árboles de decisión utilizando el algoritmo J48 con diferentes valores en el parámetro de factor de confianza de la poda, con su valor por defecto y una precisión del 86.56%. El modelo generado, indica que el atributo que mejor clasifica a los estudiantes es el tiempo empleado en el foro; clasificando con la clase Aprobado a aquellos que tienen un tiempo de participación Alto y con la clase Desaprobado a los que tienen un tiempo de participación Bajo. El segundo atributo más importante es el tiempo empleado en el sistema Moodle, relacionado con la materia, clasificando con la clase Aprobado a los que tienen tiempo de uso Medio y Alto y clasificando con la clase Desaprobado a los que tienen tiempo de uso Bajo.

La variación en el factor de confianza aumenta la precisión, pero en ninguna de las ejecuciones del algoritmo J48 se utilizaron los atributos Twitter y Facebook para la clasificación de los estudiantes. Esto se puede deber a la poca cantidad de estudiantes que por el momento optaron por estas herramientas como medio de comunicación con los responsables de la cátedra y sus compañeros.

Finalmente, la ejecución de la tarea de agrupamiento arrojo un agrupamiento bastante claro entre los estudiantes que siguen por Twitter la cuenta de la cátedra y los que no, con el resultado obtenido en la materia. Los estudiantes que no siguen por Twitter a la cátedra y los que desaprueban, se encuentran mayoritariamente juntos, agrupados en este caso en el cluster 1. De igual manera, observamos que se encuentran agrupados en su mayoría juntos, los estudiantes que sí siguen la cuenta de la cátedra por Twitter y aprueban la materia –cluster 0, y los estudiantes que siguen a la cátedra por Twitter y desaprueban –cluster 2. La separación de los clusters no es tan precisa. En otra ejecución, el cluster que mejor divide a los estudiantes agrupa mayoritariamente a los estudiantes desaprobados que no tienen relación con la cátedra mediante la red social Twitter. Se observa también que no hay en ningún cluster, estudiantes que se relacionen con la cátedra mediante Facebook y tengan un resultado malo en la materia –parte superior derecha de la imagen.

5. Conclusiones

La minería de datos educativa es una disciplina que se consolida como una herramienta muy útil para extraer conocimiento de los sistemas que se utilizan en los distintos centros educativos, en particular las instituciones educativas. El análisis integrado de distintas fuentes de datos relacionadas al mismo sujeto estudiante

permite obtener una visión más completa de la participación de los estudiantes en los diferentes servicios que ofrece la Facultad para que puedan completar sus estudios; en homogénea, concisa y confiable. En el caso presentado en este artículo, los sistemas utilizados en la Facultad de Informática de la UNLP, Moodle, SIU Guaraní y Meran, permiten observar que existe una relación entre el nivel de estudios de los padres de los estudiantes con la situación que los mismos tienen dentro de la Facultad, siendo los estudiantes más activos, aquellos cuyos padres tienen un nivel de estudios secundario o superior. También se puedo observar que existe una relación entre el uso pasivo de los foros del sistema Moodle y el resultado que los estudiantes obtienen en las materias, cuanto más es el uso de los foros del sistema, más es la cantidad de estudiante aprobados que se tiene. En el análisis de la participación de los estudiantes en la materia, se puede observar una clara división en los clusters, según la participación que los estudiantes tengan en los foros. Se ve también una subdivisión marcada en los estudiantes aprobados y desaprobados con participación baja en los foros. También aquellos alumnos que utilizan más la biblioteca generalmente aprueban.

Poder contar con datos correlacionados y completos ayudan a realizar un seguimiento del comportamiento de los estudiantes en cada una de las etapas de la carrera, así como también plantear diferentes tipos de análisis e integración con otros sistemas, que pueden ser externos a los usados por la Facultad para el almacenamiento de sus datos, por ejemplo las redes sociales tan ampliamente utilizadas en la sociedad actual.

Se pudo completar el trabajo presentado, utilizando exclusivamente herramientas de software libre, lo que permitió tener un acceso completo a la documentación, sobre todo a lo relacionado a los esquemas de base de datos utilizadas por cada una.

Visualizar la información obtenida en forma intuitiva y usable es un paso más, fundamental para que el usuario final destinatario de esta información lo transforme en acciones concretas.

6. Referencias

[1] Facultad de Informática. http://info.unlp.edu.ar
[2] Universidad Nacional de La Plata http://unlp.edu.ar
[3] Calders, Pechenizkiy 2011. Calders Toon, Pechenizkiy Mykola. “Introduction to The Special Section on Educational Data Mining”. Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), Diciembre de 2011. Volume 13, Issue 2.
[4] Romero & Ventura, 2007. Romero Cristóbal, Ventura Sebastián. “Educational data mining: A survey from 1995 to 2005”. Expert Systems with Applications. Volume 33, Issue 1, July 2007, Pages 135-146.
[5] López, et al. 2012. López Manuel Ignacio, Romero Cristóbal, Ventura Sebastián y Luna J.M. “Classification via clustering for predicting final marks starting from the student participation in Forums”. In Proceedings of the Fifth International Conference on Educational Data Mining . Panorama Hotel, Chania, Greece, Junio 19 – 21, 2012, pp 148- 151.
[6] Romero, et al. 2008. Romero Cristóbal, Ventura Sebastián, Espejo Pedro G., Hervás César. “Data Mining Algorithms to Classify Students”. In Proceedings of the First International Conference on Educational Data Mining . Montreal, Québec, Canadá. Junio 20 – 21, 2008, pp. 8-17.
[7] Dimic, et al. 2010. Dimic Gabrijela, Prokin Gragana, Kuk Kristijan, Spalevic Petar. “The Use of Data Mining Methods for Analyzing and Evaluating Course Quality in the Moodle System”. In UNITECH – International Scientific Conference, Gabrovo, Bulgaria . November 19 – 20 2010.
[8] Falakmasir & Habibi 2010. Falakmasir Mohammad Hassan, Habibi Jafar. “Using Educational Data Mining Methods to Study the Impact of Virtual Classroom in E-Learning”. In Proceedings of the third International Conference on Educational Data Mining . Pittsburgh, PA, USA, Junio 11–13, 2010, pp. 241-248
[9] WEKA – The University of Waikato. WEKA – The University of Waikato, Machine Learning Group at University of Waikato. Fecha de consulta Octubre de 2012. Disponible en http://www.cs.waikato.ac.nz/ml/weka/
[10] Consorcio SIU http://siu.edu.ar
[11] Meran http://meran.cespi.unlp.edu.ar
[12] Moodle http://moodle.org
[13] Simply Measured: Easy Social Media Analytics. http://simplymeasured.com
[14] https://apps.facebook.com/netvizz/
[15] gephi. The Open Graph Viz Platform https://gephi.github.io/

Diseño de un taller de introducción a la programación de robots

M. Lorena Talamé, Norberto Aramayo, Ana E. Gardel, M. Alicia Pérez Abelleira
Facultad de Ingeniería e IESIING
Universidad Católica de Salta,
Campo Castañares, Salta A4400
aperez@ucasal.net

Abstract

El interés por la robótica entre alumnos de la carrera de Ingeniería en Informática y la ausencia de temas directamente relacionados con la misma en el currículum, junto con la disponibilidad cada vez mayor de robots a costos accesibles, ha motivado el diseño de un taller de introducción a la programación de robots. Dicho taller permite a los alumnos incursionar en la disciplina usando tanto un robot real como un simulador, y de paso integrar algunas habilidades ya adquiridas a lo largo de la carrera, con la motivación añadida de ver resultados al aplicarlas al control del robot. El presente trabajo describe las herramientas utilizadas y los contenidos, eminentemente prácticos del taller.

1. Introducción y motivación

En nuestra experiencia docente en la carrera de Ingeniería en Informática de la Universidad Católica de Salta hemos venido observando el interés de muchos alumnos con la robótica, algo atractivo pero prácticamente inaccesible en nuestro contexto. Es por eso que comenzamos a explorar la posibilidad de introducir el tema de manera extracurricular para los alumnos de ingeniería y hasta para alumnos de colegios secundarios como elemento motivador. Dado el perfil de nuestra carrera, el énfasis estaría en la programación de robots, en lo posible de robots reales pero también aprovechando la disponibilidad de simuladores de los mismos.

No es casualidad este interés por la robótica. Esta disciplina, que ciertamente no es nueva, se va haciendo cada vez más cercana por la extensión de las aplicaciones robóticas al ámbito del hogar y otras tareas de la vida diaria. Ya en 2007 Bill Gates predijo en un artículo en *Scientific American* titulado *Un robot en cada hogar* [1] que la robótica sería la próxima área destacada de la informática. Por su interés, reproducimos aquí el primer párrafo de este artículo: "Imagínense que están presentes en el nacimiento de una nueva industria. Es una industria basada en tecnologías innovadoras, en la que un puñado de corporaciones bien establecidas vende dispositivos altamente especializados para usos industriales, y un número cada vez mayor de *startups* producen juguetes innovadores, gadgets para aficionados y otros productos interesantes. Pero también es una industria muy fragmentada, con pocos estándares o plataformas. Los proyectos son complejos, el progreso es lento, y las aplicaciones prácticas son relativamente raras. De hecho, a pesar de toda la excitación y la promesa, nadie puede decir con certeza cuándo – o si – esta industria llegará a tener masa crítica. Si lo consigue, seguramente podrá cambiar el mundo." Gates sigue diciendo cómo este párrafo bien podría haberse referido a la industria de los PCs hace ya unas décadas, pero sin embargo está hablando de la industria robótica.

Según la *International Robotics Federation* [2], solo en 2012 se vendieron unos 3 millones de robots para uso personal y doméstico. La mayoría de ellos, 1,96 millones, se utilizan en tareas domésticas (limpieza del suelo, corte del pasto, etc.), con un incremento del 53% sobre 2011 y ventas de US$697 millones. Los restantes son principalmente para entretenimiento (robots que son juguetes o para aficionados), educación e investigación. Las previsiones para el período 2013-2016 indicaban que se venderán unos 22 millones de unidades para uso personal, de los cuales 15 serán para tareas domésticas con un valor estimado de US$5600 millones; unos 3,5 millones de robots para entretenimiento, unos 3 millones para educación e investigación, y unas 6400 unidades para asistencia a personas ancianas y discapacitadas, un mercado éste último que aumentará sustancialmente en los próximos 20 años.

Así cada vez está más cerca un futuro en que los robots y dispositivos relacionados formen parte de la vida diaria, gracias a tecnologías como la computación distribuida, el reconocimiento y síntesis de voz, el procesamiento de imágenes, y la conectividad inalámbrica de banda cada vez más ancha. Este futuro ciertamente motiva a muchos de nuestros alumnos,

incluso aun cuando no está exento de preocupaciones éticas.

Una empresa, iRobot, iniciada por investigadores del MIT, ha fabricado y vendido desde su comienzo más de 10 millones de robots para uso doméstico, entre ellos su conocido modelo de robot aspiradora, Roomba [3]. Gracias a su éxito comercial, iRobot ha podido poner en el mercado un robot con la misma arquitectura que el Roomba para tareas de investigación y educación, el Create (Figura 1). Se trata de una variante del Roomba sin los accesorios de aspirar y cepillar. El precio del mismo de US$150 (más US$100 para una batería recargable), es un costo por fin al alcance de las posibilidades de muchos interesados en la robótica, específicamente en la programación de robots.



Figura 1 Robot Create de iRobot con cámara inalámbrica utilizado en el taller

Es importante destacar que la programación de los robots se está convirtiendo en el principal desafío, más que el hardware. Algunos investigadores de prestigio, como Red Whittaker y Sebastian Thrun, han afirmado que a pesar de las dificultades para que el hardware funcione adecuadamente, las posibilidades desde el punto de vista del hardware ya casi están; ahora la dificultad está en el software que resuelva el problema [1] [4].

La existencia del Create, por su precio y calidad, nos motivó a poner en práctica el deseo de ayudar a nuestros alumnos a incursionar en el mundo de la robótica. Tras explorar el estado de la enseñanza de la robótica y las fortalezas y debilidades de nuestro equipo y contexto, y tras conseguir una unidad de este robot, avanzamos en la posibilidad de dictar talleres de introducción a la robótica para nuestros alumnos, dejando la posibilidad abierta para conseguir en el futuro un número mayor de robots, en particular del tipo Lego Mindstorms, con mayor versatilidad desde su arquitectura física y variedad de sensores y actuadores. El Create viene listo para usar con una variedad de sensores, lo cual nos ha permitido centrarnos en la programación más que en el hardware. No obstante como plataforma el robot permite muchas posibilidades de extensión también desde el hardware y la electrónica.

Parte de nuestra exploración fue la búsqueda de alternativas para desarrollar software para robots, y específicamente para el Create y el Mindstorms. En base a ella decidimos inclinarnos por el Microsoft Robotics Development Studio (MRDS). Los aspectos en que basamos la decisión fueron la disponibilidad de un simulador de robots en diversos contextos; la posibilidad de controlar diversos robots usando la misma interfaz junto con la existencia de librerías para los dos robots mencionados, que nos permitieron comenzar a usar el Create de inmediato; la posibilidad de programar tanto con un lenguaje visual como con un lenguaje más tradicional, C#, conocido por buena parte de nuestros alumnos; y por último la disponibilidad del entorno en forma gratuita.

La introducción a la robótica, además de ofrecer a nuestros alumnos un acercamiento a la robótica, les permite aplicar conceptos y técnicas de programación que han ido adquiriendo durante la carrera, y ampliar sus conocimientos en otras que tal vez solo hayan visto desde la teoría. En particular, uno de los problemas más complejos de la programación en robótica es gestionar en simultáneo todos los datos provenientes de una diversidad de sensores, y emitir los comandos apropiados a los actuadores. En un enfoque de programación convencional un programa con un solo hilo (*thread*) iría sucesivamente leyendo datos de los sensores, después los procesaría y finalmente decidiría el comportamiento del robot, para volver a comenzar el ciclo de nuevo. Este enfoque tiene obvias limitaciones para la capacidad de respuesta ante imprevistos. Así el alumno se enfrenta con el problema y gestión de la concurrencia, un desafío que se extiende más allá de la robótica a aplicaciones que precisen orquestar de manera eficiente código que se ejecute, por ejemplo, en diferentes servidores o procesadores al mismo tiempo.

En resumen, hemos desarrollado el taller de introducción a la robótica, y más concretamente a la programación de robots, que describimos en este trabajo, con la intención de ayudar a los alumnos a acercarse al gran potencial, presente y futuro, de la robótica, combinando técnicas ya aprendidas por ellos a lo largo de su carrera y exponiéndolos a elementos teóricos y prácticos novedosos en esta interesante y excitante tecnología emergente. En las dos próximas secciones describimos el entorno de desarrollo elegido y el enfoque y elementos del taller propuesto, para terminar este trabajo con un análisis de las ventajas e inconvenientes de la propuesta, líneas de trabajo futuro y algunas conclusiones.

2. MRDS

Microsoft Robotics Developer Studio (MRDS) es un entorno de programación basado en .NET para construir aplicaciones robóticas. En esta sección se describen los cuatro componentes principales de MRDS: CCR (*Concurrency and Coordination Runtime*), DSS (*Decentralized Software Services*), el lenguaje de programación visual VPL, y el entorno de simulación.

Como se ha mencionado, la concurrencia es un aspecto esencial en la programación robótica. Un enfoque para gestionar la concurrencia es escribir programas multihilo, siendo ésta una de las tareas más complejas en programación. MRDS esconde esta complejidad tras una librería llamada *Concurrency and Coordination Runtime* (CCR) para gestionar tareas asíncronas ejecutadas en paralelo, diseñada inicialmente para ayudar a los programadores a aprovechar las ventajas de sistemas multinúcleo y multiprocesador [5]. Está escrita en C# y basada en .NET. CCR incluye una clase *Dispatcher* que implementa un pool de hilos, que pueden ejecutarse simultáneamente. Cada despachador (*dispatcher*) consta de una cola de *delegados*, cada uno de los cuales representa el punto de entrada de una tarea que puede ejecutarse asíncronamente. El despachador distribuye esas tareas entre los hilos. Cada elemento de la cola del despachador tiene un puerto, que es a su vez una cola en que se coloca la ejecución de la tarea. La tarea puede tener asociada un objeto *ReceiverTask* que consume el resultado para procesamiento posterior. Un árbitro gestiona este consumidor y lo invoca cuando el resultado esperado está disponible en el puerto.

Además de CCR, MRDS añade otro nivel denominado *Decentralized Software Services* (DSS) para combinar aplicaciones concurrentes en CCR que se denominan servicios, permitiendo además que los servicios corran en diferentes computadoras en una red. Se trata de un modelo de servicios *loosely coupled*. DSS organiza, crea y manipula servicios facilitando la creación de aplicaciones concurrentes y distribuidas. Los servicios – por ejemplo el código que lee un sensor o controla un motor – operan como procesos separados que pueden ser orquestados por otro servicio. El diseño de DSS sigue una arquitectura orientada a servicios web, en analogía con la forma en que texto, imágenes o información de diversos servidores se agrega en una sola página web. DSS utiliza el protocolo DSSP, que es similar a SOAP (Simple Object Access Protocol), utilizado para servicios web. La diferencia entre ambos es que mientras que el estado de un servicio web está oculto, el estado de un robot debe ser visible (el estado, por ejemplo, incluye la lectura de los sensores del robot, o la imagen capturada por su cámara). Así el estado de un servicio puede visualizarse u modificarse mediante un navegador web. DSSP separa el estado del servicio de su comportamiento u operaciones.

DSS permite que los servicios corran aisladamente, por lo que si un componente del robot falla, puede ser apagado o reiniciado desde la misma aplicación. Los servicios de un robot pueden también monitorearse o puede cambiarse su estado desde una ubicación remota usando la misma interfaz web.

Cuando se programa en MRDS la aplicación no reside en el robot, sino en una computadora (o distribuida entre varias). Así el robot puede ser un dispositivo relativamente sencillo que delega las tareas de procesamiento complejo (ej. visión, navegación y planificación de caminos con algoritmos sofisticados, orquestación de las tareas al nivel más alto) al hardware mucho más potente de la PC. Es posible incluso manejar grupos de robots que colaboren para realizar tareas complejas.

Las aplicaciones en C# pueden crearse utilizando la IDE Visual Studio, que tiene plantillas para crear un servicio que incluyen muchas de las directivas y elementos necesarios para el servicio. Hemos de decir que a pesar de estas ayudas la programación en este entorno es bastante compleja. Esta complejidad queda oculta en buena parte cuando se usa el lenguaje visual VPL, que se describe más adelante.

La vinculación entre el software y el hardware se define mediante archivos *manifest*; una misma aplicación puede correr en diferentes plataformas hardware cambiando simplemente el archivo *manifest*. Por ejemplo, un usuario podría programar usando comandos genéricos el movimiento de un dispositivo con tracción diferencial, luego y reemplazar el archivo *manifest* con el de cualquier robot, real o simulado, que utilice tracción diferencial. El servicio convierte automáticamente los comandos de alto nivel en instrucciones de bajo nivel específicas para ese tipo de robot. O también un servicio que precise una cámara, puede utilizar indistintamente una cámara web, una cámara IP o la cámara del simulador simplemente cambiando el *manifest* y no el código del servicio. Esta es una ventaja de utilizar MRDS: la posibilidad de cambiar el hardware sin cambiar el código [6].

El tercer componente a destacar es un lenguaje de programación visual (VPL) y su correspondiente entorno de desarrollo. Está basado en un modelo de programación gráfica para el paradigma de flujo de datos, que modela un programa como un grafo dirigido de datos que fluyen entre operaciones. Un programa, más que una serie de comandos imperativos que se ejecutan en secuencia, es como un conjunto de trabajadores en una línea de ensamblado, que hacen su tarea tan pronto como reciben los componentes, en este caso los datos. Por ello VPL se adapta para escenarios de procesamiento concurrente y distribuido. Está dirigido a programadores principiantes,

aunque otros más veteranos lo pueden aprovechar para el desarrollo rápido y prueba de prototipos de servicios. El código generado puede convertirse automáticamente en código en C# [7].

Por último, MRDS incluye un simulador, en que los usuarios pueden recrear entornos complejos y colocar en ellas robots. Hay una serie de escenas y de robots disponibles de antemano y un lenguaje para crear nuevas escenas y dispositivos simulados. El motor del simulador es capaz de simular colisiones, gravedad y otras fuerzas de manera muy aproximada a como los objetos se comportan en el mundo real. El uso del simulador facilita el acceso a la robótica a estudiantes aunque no se disponga de robots físicos suficientes.

La versión de MRDS que utilizamos es la R4 (2012) que precisa del sistema operativo Windows 7 u 8 y Visual Studio Express 2010 o superior. Con Windows XP solo puede utilizarse la versión R3 (2008). No existen versiones de MRDS para otros sistemas operativos. Todo el software está disponible de manera gratuita en la web o a través de la iniciativa académica de Microsoft, lo cual es una innegable ventaja en nuestro contexto. Como se indicó anteriormente, MRDS incluye librerías para utilizar el Create y el Lego Mindstorms, tanto con el robot real como en simulación.

3. Enfoque del taller

El taller que hemos diseñado tiene dos componentes: teórico y práctico, con un énfasis mayor en el segundo. El taller surge como investigación de cátedra en el área de la inteligencia artificial (IA) y esto marca la orientación de la parte teórica hacia la IA y los robots inteligentes. El material introductorio de las clases teóricas está basado en el de [8] en cuanto a los elementos básicos de un robot, y en [9] acerca de los diversos paradigmas de programación robótica (jerárquico, reactivo, e híbrido) y el proceso de diseño para construir robots inteligentes. Aunque no son obras muy recientes, son adecuadas para el nivel introductorio del taller. Para el trabajo de programación del Create con MRDS la referencia que más hemos utilizado es [10].

El núcleo del taller son los trabajos prácticos que los alumnos deben realizar, y que se irán describiendo en las secciones que siguen. En cada sesión se propone una actividad de programación facilitando a los alumnos un tutorial que indica los pasos a seguir; previo a ello se desarrolla un tema teórico vinculado con el práctico. En la elección de temas para los prácticos hemos adaptado parte de los propuestos en [6]. Los temas de los prácticos están relacionados con los teóricos, y así los alumnos pueden aplicar el conocimiento de manera concreta.

La mayor parte de los prácticos están diseñados para utilizar tanto el simulador como el robot real. MRDS permite esto con pequeños cambios en los programas creados por los alumnos, en particular con el intercambio de archivos *manifest*. Así se puede aprovechar mejor el recurso escaso que es el único robot Create disponible y los alumnos pueden continuar las prácticas en otro horario y lugar. En general se sugiere a los alumnos que trabajen en parejas, tanto por la disponibilidad del robot como por la ayuda que se pueden prestar en el aprendizaje, ya que algunas de las tareas son difíciles de implementar y de hacer funcionar en VPL, especialmente para principiantes; esto hace que el trabajo sea en ocasiones muy satisfactorio y en otras bastante frustrante.

Aunque nuestros alumnos tienen experiencia en programar en diversos lenguajes de .NET, incluyendo C#, la curva de aprendizaje de CCR y DSS es demasiado pronunciada para un taller de las características que nos proponemos. Por ello en el taller se utiliza el lenguaje VPL, aunque al final se propone a los alumnos el desafío de programar servicios en C# y se les proporcionan algunos materiales a modo de tutorial.

El hardware disponible para los prácticos incluye además del robot Create con su batería recargable, una cámara web inalámbrica, y un módulo BAM para conectar el robot mediante Bluetooth a las computadoras en que trabajan los alumnos (éstas deben disponer de conexión Bluetooth o utilizar el *dongle* USB y software del que dispone el taller). También se dispone de una pared virtual diseñada específicamente para el robot Create.

3.1. Práctico I: Introducción a la robótica, al MRDS y al robot Create

El primer práctico pide a los alumnos que realicen los tres primeros tutoriales proporcionados por los creadores del MRDS y disponibles en la web (traducidos por nosotros). El primer tutorial introduce el concepto de servicios usando VPL. Los servicios son el núcleo básico de la programación robótica en nuestro entorno. Un servicio es una entidad (implementada con orientación a objetos en C#) que controla un componente hardware o software particular. Como se indicó, una aplicación robótica consiste en orquestar una serie de servicios, y combinando servicios muy sencillos se puede obtener un comportamiento complejo. En el segundo tutorial se combinan varios servicios, en particular un sensor de impacto y la tracción diferencial del robot. Los alumnos aprenden a encender y apagar los motores del robot mediante un botón, y pueden extender ese comportamiento para controlar la velocidad del robot, por ejemplo.

El tercer tutorial añade algo más de funcionalidad con los mismos elementos, haciendo que el robot deambule y, si choca con un obstáculo retroceda una pequeña distancia, gire aleatoriamente por un corto periodo de tiempo, y luego avance en la nueva dirección. Además se

propone construir una nueva actividad en VPL (análoga a una subrutina) que nuclea este comportamiento. Como ya se indicó, el mismo código puede ser utilizado tanto en el simulador (para lo cual proporcionamos un escenario con un robot y diversos obstáculos, mostrado en la Figura 2) como con el robot real, simplemente modificando el archivo *manifest* utilizado. El objetivo de este práctico es familiarizarse con el funcionamiento de MRDS y VPL, en particular crear un proyecto, acceder a los servicios, y conectarlos entre sí en el paradigma de flujo de datos que sigue VPL. En la introducción a los tutoriales se explica brevemente la diferencia entre la programación basada en flujo de datos y la basada en flujo de control, que es a la cual los alumnos están acostumbrados. Este es un aspecto que suele causar dificultades conceptuales en cuanto la complejidad de los programas va aumentando.

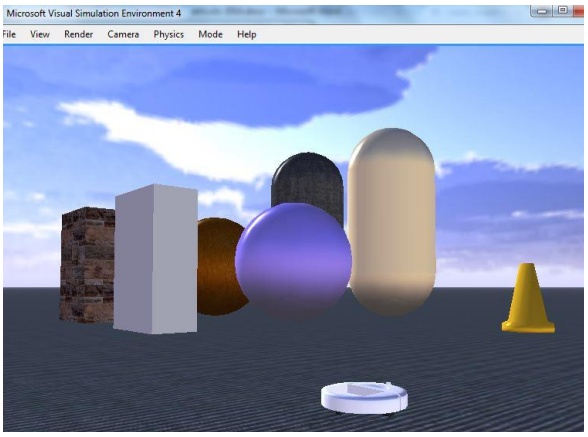


Figura 2. Escenario de simulación por defecto para el robot Create

El práctico en sí no tiene mayor dificultad ya que los alumnos van siguiendo los pasos especificados en los tutoriales. No obstante, en base a nuestra experiencia con estas herramientas, este trabajo puede llevar tiempo considerable y encontrar dificultades complejas y sutiles para hacer que el código funcione.

3.2. Práctico II: Teleoperación

El segundo práctico toma como punto de partida un tutorial sobre teleoperación también disponible en MRDS, en el que además se debe aplicar lo aprendido en el primer práctico. El material teórico cubre el tema de teleoperación. Aunque nuestro énfasis es en robots autónomos, éste es un tema importante, un punto intermedio necesario entre la mayoría de los robots industriales, programados para realizar tareas de precisión en entornos predecibles y no cambiantes, y los robots inteligentes que deben desenvolverse en entornos desconocidos y cambiantes y responder adecuadamente, sin poder ser programados de antemano para responder y adaptarse a todas las posibles contingencias [9]. El puente entre ambos enfoques tradicionalmente han sido mecanismos que permiten que una persona controle de manera remota el robot, o partes del mismo. A este conjunto de mecanismos, locales o remotos, suele llamarse teleoperación. Además la teleoperación forma parte de muchas de las nuevas aplicaciones de robots para la telepresencia, un conjunto de tecnologías que permiten a una persona sentir como si estuviera presente en un lugar remoto, o dar la impresión a otros de que lo está [11].

En este práctico los alumnos crean una interfaz para controlar el robot remotamente y visualizan la imagen devuelta por la cámara ubicada en el robot. Así navegan un recorrido con obstáculos, pero no viendo el robot sino la imagen obtenida por la cámara montada sobre el mismo, como suele ocurrir en la mayoría de las aplicaciones de teleoperación. El práctico incluye una competencia para ver qué equipo de alumnos es más hábil en teledirigir el robot y evitar los obstáculos, algo más difícil de lo que parece.

El primer práctico constaba de sencillos ejemplos. En este se construye en base a ellos una aplicación práctica. El carácter competitivo añade motivación para que los alumnos vayan un poco más allá de lo propuesto en el tutorial, especialmente en cuanto a la interfaz de teleoperación.

2.3. Práctico III: Deambular y cubrir un recinto

Una de las tareas frecuentes de un robot es deambular recorriendo su entorno. En este práctico desde la teoría se presenta la importancia de este comportamiento y ciertos enfoques al mismo [9]. Los alumnos en el práctico pueden implementar sus propios algoritmos, en particular para conseguir que el Create cubra el suelo de la habitación de la manera más eficiente posible. Después deben añadir comportamiento que haga que el robot evite obstáculos usando los sensores disponibles en el Create (sensor de impacto y sensor de infrarrojo que detecta una pared cercana). Así pueden comprobar la efectividad, ventajas e inconvenientes de diversos tipos de sensores para una tarea específica. No existe un sensor perfecto, sino que deben aprovecharse los sensores disponibles en el robot. Tampoco puede decirse que los enfoques más complejos siempre realicen la tarea mejor que comportamientos muy sencillos.

Hay una relación entre este práctico y una aplicación real, ya que la tarea propuesta es la tarea básica del robot aspirador Roomba, mellizo de nuestro robot Create con los mismos sensores y actuadores. El éxito del Roomba está en gran medida basado en la efectividad de sus algoritmos para deambular y recorrer un recinto.

2.4. Práctico IV: Ejemplo del paradigma reactivo

La tarea de este práctico está inspirada en los comportamientos biológicos de buscar comida y evitar peligros, tales como depredadores. Con esta idea se pretende mostrar los fundamentos del paradigma reactivo en robótica. Los alumnos tienen libertad para elegir cómo implementar esos comportamientos. Para ello deben utilizar la cámara web inalámbrica disponible o el simulador junto con un servicio que realiza un procesado de imágenes sencillo para detectar "comida" - pelotas de color azul - y "depredadores" - pelotas de color rojo (ver Figura 3). Este servicio ha sido desarrollado por nosotros usando AForge.NET, un framework en C# para el desarrollo de aplicaciones de procesamiento de imágenes, redes neuronales, algoritmos genéticos, robótica, etc (*www.aforgenet.com*).

Dado que nuestro robot no tiene un brazo, lo que los alumnos pueden realizar con comida y depredadores es bastante limitado. No obstante hay una variedad de respuestas que pueden implementar, por ejemplo alejarse inmediatamente del peligro o tratar de rodearlo como si fuera un obstáculo.

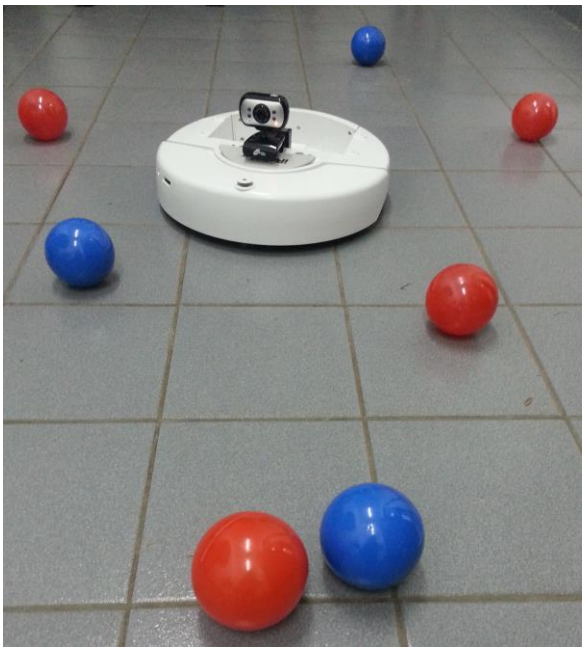


Figura 3 Entorno para el práctico IV

2.6. Práctico V: Seguidor de línea

En este práctico los alumnos enfrentan un ejemplo típico de robótica: seguir una línea marcada en el suelo. En nuestro caso se utilizan los sensores de abismo del Create, cuyo uso original es detectar desniveles, tales como escaleras. Estos sensores utilizan infrarrojos, por lo que pueden aprovechar las diferencias en las propiedades de reflexión de distintas superficies para detectar la ausencia o presencia de la línea. Así los alumnos aprecian la versatilidad, ventajas e inconvenientes de diversos sensores. Este práctico permite el trabajo tanto con el robot real como con el simulado; en el segundo caso se aprovechan los servicios y los escenarios del simulador proporcionados en [10].

2.7. Práctico VI: Contar objetos

Este último práctico es opcional y depende de cuánto cada alumno o equipo avance con los prácticos anteriores. El objetivo es contar los objetos existentes en un entorno. El uso de los temas de los prácticos anteriores facilita esta tarea, que por otro lado no está exenta de dificultades, como saber qué objetos ya han sido contados.

Una alternativa a este práctico es dejar libertad a los alumnos para atacar un problema que sea de su interés después de todas las experiencias anteriores. Aunque el hardware disponible, especialmente la variedad de sensores, es limitado, hay muchas posibilidades de temas que no hayan sido cubiertos ya. Así los alumnos pueden integrar los conceptos anteriores y explorar algo de su propio interés dentro del tema del taller.

3. Ventajas e inconvenientes del diseño del taller

En el diseño del taller se intentó hacer corresponder temas básicos teóricos sobre robótica con los trabajos prácticos. Así los alumnos pueden comprender algunos de los problemas principales de la robótica, especialmente desde el punto de vista de la programación, y aplicar en el laboratorio alguno de los principios básicos para resolverlos.

Por tratarse de un taller opcional para alumnos interesados, y no un curso formal del currículum, no se puso el énfasis en diseñar actividades de evaluación. Es difícil cuantificar la creatividad y el ingenio a la hora de resolver alguno de los prácticos. Además no siempre los esfuerzos se traducen en comportamiento adecuado o hasta deseado del robot, lo cual pude ser resultado de las limitaciones del hardware (ej. precisión de los sensores o actuadores) y del software disponible (ej. para procesado de imágenes, o las características del lenguaje y entorno VPL).

Desde el comienzo MRDS pareció una alternativa adecuada para un taller de estas características. En cierto modo proporciona un puente entre lo que los alumnos ya conocen (sintaxis de lenguajes de programación, o hasta el entorno Visual Studio y .NET) y los elementos semánticos nuevos propios de la robótica. Otra ventaja clara fue la posibilidad de comenzar a trabajar inmediatamente con el robot disponible y con un

simulador gracias al soporte y librerías incluidas en la herramienta. Esta es una ventaja importante dado que nuestro énfasis, como ya se indicó, es en la programación robótica y no tanto en la electrónica u otros aspectos hardware de bajo nivel. La calidad y realismo del simulador, la variedad de escenarios proporcionados y la posibilidad de diseñar nuevos escenarios con relativa facilidad son aspectos muy atractivos. La disponibilidad de tutoriales y otros materiales para iniciarse en el entorno MRDS es otra ventaja adicional. La disponibilidad gratuita de las herramientas necesarias fue también un factor determinante.

Sin embargo la curva de aprendizaje para nuestro equipo ha sido bastante pronunciada. Una de las tareas arduas de la preparación del taller ha sido familiarizarnos lo suficiente con el software para poder transmitir conceptos y tecnología de manera adecuada. En el proceso hemos encontrado muchas dificultades debidas a que algunos aspectos del mismo no funcionaban como se esperaba, incluyendo algunos errores del software, y también debidas a la falta de material en castellano, y a la necesidad de aprender en muchos frentes al mismo tiempo, desde la programación orientada a servicios con CCR y DSS, hasta las dificultades propias de los servicios concretos para controlar el robot.

Microsoft lanzó la iniciativa de MRDS como una propuesta de una plataforma estándar para programar robots; la versión más reciente es de marzo de 2012 e incluye amplio soporte para el dispositivo Kinect. Sin embargo no está claro el futuro de esta herramienta, el sitio web no ha variado sustancialmente desde entonces, y el interés y soporte en los foros disponibles parece ir disminuyendo (aunque las preguntas que realizamos en los mismos fueron contestadas relativamente pronto por los desarrolladores de la herramienta). La complejidad de la misma es uno de los factores que parecen causar el desinterés en algunos usuarios. Esta complejidad es notable cuando se trata de extender la herramienta a otros sensores o robots diferentes de los muchos disponibles inicialmente. En teoría parece una tarea sencilla dada la cantidad de ejemplos y de implementaciones que se pueden adaptar a nuevos robots; en la práctica la complejidad es grande.

No obstante, el escenario de trabajo hardware y software que hemos descripto nos ha dado ya la posibilidad de introducir la robótica de manera accesible en nuestro contexto y con capacidad de entusiasmar a nuestros alumnos y aumentar su interés y curiosidad.

4. Trabajo futuro

A la hora de redactar este trabajo estamos a punto de ofrecer este taller por primera vez a un grupo de alumnos. La evaluación de esta experiencia, que esperamos repetir regularmente, nos dará información para trazar el camino a seguir en el futuro. Idealmente podremos aumentar el número y variedad de robots disponibles. Como se mencionó, una de las posibilidades es utilizar el Lego Mindstorms. El taller no precisaría grandes modificaciones, pero los sensores disponibles (ej. sónar) y las posibilidades de configuración del robot (por ej. como un brazo robótico) amplían el rango de posibilidades de lo que los alumnos podrían llegar a diseñar y construir a lo largo del taller.

Varios alumnos están comenzado proyectos de grado con la tecnología presentada y aumentándola en diversas direcciones. Uno de los aspectos a explorar es el uso de otras herramientas software que permitan el desarrollo de proyectos más avanzados, incluso en el ámbito de la investigación, y que puedan reemplazar el uso de MRDS, especialmente si éste cae en desuso o se torna obsoleto.

5. Conclusiones

La disponibilidad de un robot y especialmente de un equipo de personas interesadas en la robótica ha generado muchas expectativas ente los alumnos y profesores de la Facultad. Aunque el taller propuesto puede mejorarse en diversos aspectos, cumple las expectativas de motivar a nuestros alumnos a incursionar en la robótica y de exponerlos a esta tecnología tan innovadora en nuestro contexto pero de tanta importancia para el futuro de sus carreras y hasta de su estilo de vida.

La disponibilidad de robots de bajo costo, como el Create de iRobot, y de software de acceso gratuito para la programación de robots han hecho posible el acceso de nuestros alumnos a esta tecnología. El campo está abierto además para que los alumnos puedan llevar a cabo trabajos finales e investigaciones a nivel de cátedra, lo que genera entre nosotros grandes expectativas.

Referencias

[1] B. Gates, «A robot in every home,» *Scientific American*, pp. 58-65, Enero 2007.

[2] IRF, «World Robotics - Service Robots 2013. Executive Summary,» Disponible en http://www.worldrobotics.org/uploads/tx_zeifr/Executive_Summary_WR_2013_01.pdf, 2013.

[3] iRobot, «Our History,» 2014. Disponible en http://www.irobot.com/us/Company/About/Our_History.aspx. [Último acceso: 11 07 2014].

[4] S. Thrun, «The great robot race,» *NOVA - PBS*, p. Disponible en http://www.pbs.org/wgbh/nova/transcripts/3308_darpa.html, 28 Marzo 2006.

[5] Microsoft, «Microsoft Robotics User Guide,» 2010. [En línea]. Disponible en http://msdn.microsoft.com/en-us/library/dd936006.aspx.

[6] K. Workman y S. Elzer , «Creating An Upper-Level

Undergraduate Robotics Elective in Computer Science,» de *Proceedings of the 23rd Annual Conference of the Pennsylvania Computer and Information Science Educators (PACISE)*, Kutztown, PA, 2008.

[7] Microsoft, «Visual Programming Language,» 2010. Disponible en http://msdn.microsoft.com/en-us/library/bb964572.aspx..

[8] D. Housten, Learning Roomba, MSc Thesis, Drexel University, 2008.

[9] R. Murphy, Introduction to AI robotics, Cambridge, Massachusetts: The MIT Press, 2000.

[10] K. Johns y T. Taylor, Professional Microsoft Robotics Developer Studio, Wiley, 2008.

[11] R. Lehrbaum, «Attack of the Telepresence Robots!,» *Information Week*, pp. Disponible en http://www.informationweek.com/applications/attack-of-the-telepresence-robots!/d/d-id/1108137, 1 11 2013.

Estudio de la Modalidad Atencional en Educación a Distancia en una Plataforma Learning Management System

Patricia Etcheverry, Calixto Maldonado
Universidad Empresarial Siglo 21

Resumen - *El artículo presenta los resultados preliminares del Estudio llevado adelante por un grupo de investigadores de la UESiglo21 de Córdoba, Argentina. La investigación refiere a la atención como proceso cognitivo de renovado formato vinculado al aprendizaje de dos materias correspondientes a la Licenciatura en Informática e Ingeniería en Software de la Universidad, mediante herramientas de la Plataforma Learning Management System (LMS). El diseño metodológico utilizado es mixto, denominado DexPlis, de fase cuantitativa y cualitativa; los resultados cuantitativos preliminares arrojaron un desempeño diferencial de la atención según soporte, texto o video. Se definen lineamientos para la etapa cualitativa a fin de descifrar las motivaciones profundas que justifiquen el renovado formato atencional, en el contexto de un imaginario social efecto del entorno telemático en donde se desenvuelve el aprendizaje con Tecnologías de la Información y la Comunicación (TICs).*

Palabras clave- *atención, proceso cognitivo, Learning Management System, DexPlis, imaginario social, Tecnologías de la Información y la Comunicación.*

Introducción

El antecedente a la investigación a la cual nos referiremos, ha sido un trabajo publicado en la Revista TE&-ET [15] que hace foco en la influencia de las Tecnologías de la Información y la Comunicación (TICs) en el imaginario social [14] de los estudiantes de nivel medio del ciclo superior de colegios de la Ciudad de Buenos Aires.

Este trabajo, de índole básicamente cualitativa, arrojó interesantes conclusiones. Los estudiantes vincularon como área de significación espontánea a las TICs con las Redes Sociales, especialmente Facebook, en términos de ocio y espacio social, es decir, las TICs generan atención e interés para esta población sólo en estrecha vinculación con el ocio social.

En este contexto el equipo de investigación se interrogó acerca de cómo sería posible, desde este entorno digital mediado por plataformas de enseñanza, generar interés y atención pero ya no vinculado al ocio sino al aprendizaje.

Autores actuales de la corriente de la Psicología Cognitiva han analizado el impacto de las TICs en los procesos cognitivos [1].

A los fines de nuestra investigación se han rescatado, entre otros temas, cuestiones planteadas en relación a la singularidad que implica la dualidad computadora-estudiante durante el aprendizaje. Es decir, se plantea si la computadora, junto con sus programas y poder de conectividad, a diferencia de otras herramientas provistas por la cultura, son efectivamente potenciadoras de la inteligencia humana, si esta vinculación nos hace efectivamente más inteligentes [1].

Desde esta línea de análisis los autores [1] [3] afirman que el individuo, a partir de esta particular relación, se halla influenciado para desarrollar capacidades, pero sólo serían capacidades válidas si en la díada computadora-estudiante se mantiene una vinculación de tipo analítica y no sistémica, dado que esta última resolvería todo por el estudiante. En este sentido sólo la capacidad analítica permite al estudiante la ubicación en situación de esfuerzo y reflexión consciente, facilitando de ese modo el logro del residuo cognitivo y por ello mismo la posibilidad de transferir estrategias y habilidades y su posible aplicación a contextos diferentes al que dio origen su aprendizaje [1].

En definitiva, esta relación particular establecida con la computadora puede significar contar con residuo cognitivo y verdadero aprendizaje. Más aún, la teoría socio cultural realiza aportes en este sentido a partir de las cogniciones distribuidas. [2] [4] [5] [7]. La cognición distribuida se conceptualiza desde el marco teórico de la Teoría de la Actividad Histórico-Cultural de Vigotzky [6]. La misma refiere a que un proceso cognitivo como el aprendizaje no es considerado sólo una propiedad al

interior de un individuo que aprende, sino que justamente se trata de un aprendizaje siempre mediado por la cultura y por eso mismo, distribuido en individuos, colegas, materiales, herramientas y recursos semióticos. Es decir, desde esta concepción es tan importante el individuo como el contexto y ambos debieran ser considerados en proceso e interacción en función de un conocimiento construido de modo estructural o sistémico. Incluso Roy Pea confirma esta posición [2] señalando que un posible diseño de la inteligencia en términos distribuidos podría plantearse como meta educativa, a partir del desarrollo de líneas de investigación orientadas hacia el logro de un aprendizaje más ajustado a los tiempos actuales.

Dada la importancia del residuo cognitivo vinculado al "real aprendizaje" de índole meta cognitivo, se hace indispensable acudir a los aportes de Engestrom [8] y Salomon [3] en relación a los supuestos de aprendizaje y los subtipos de transferencias posibles. Desde este punto de vista el aprendizaje es considerado un fenómeno de muchas capas, distinguiéndose tres niveles de proceso, el del aprendizaje por condicionamiento, el de la comprensión de las reglas de juego y ajuste al entorno y el aprendizaje expansivo. Este último caracterizado por su innovación y rareza [8].
A los aprendizajes conductuales por condicionamiento y de ajuste al contexto les corresponderían transferencias de habilidades desde dos vías posibles. Al primero la transferencia por vía baja, de carácter automático con escaso compromiso mental, y al segundo por vía alta, caracterizado por su compromiso mental empleando procesos psicológicos superiores no de forma automática sino consciente y reflexiva, meta cognitivamente guiados [3].

En consecuencia se rescata el aprendizaje que se acompañe de esfuerzo consciente, por vía alta y de ese modo requiera de atención para ser internalizado y permitir el cultivo de habilidades cognitivas y meta cognitivas con posibilidad de transferencia.

Ahora bien, si la condición para adquirir e internalizar junto con la suma de valor del residuo cognitivo transferible implica contar con un sujeto activo con voluntad y reflexión consciente, se hace indispensable contar con un estudiante dispuesto a una abstracción atenta a los procedimientos, a la autorregulación y a las estrategias [1].
Teniendo en cuenta esta finalidad, es decir, sostener perceptivamente y atencionalmente el aprendizaje en un entorno digital, quedan establecidos los interrogantes que orientan al equipo de investigación: los estudiantes influenciados por el entorno conectivo digital y sus diferentes soportes, ¿Llevan adelante procesos cognitivos singulares a fin de focalizar la atención? Los nativos digitales [9] por influencia de este nuevo entorno, ¿Han aprehendido otras capacidades atencionales? En tal caso ¿Sabemos de ellas? Y si la capacidad de "prestar atención" voluntariamente ha cambiado ¿Sabemos la forma que ha adquirido?

Es de interés para el trabajo de investigación, brindar respuestas en base a la información recabada en la plataforma Learning Management System (LMS) a partir de la actividad de estudiantes actuales que cursan dos materias correspondientes a la Licenciatura en Informática e Ingeniería en Software, utilizando la plataforma denominada EPIC.

El insumo que se utiliza es el análisis de los reportes de actividad y uso de los materiales que brinda la herramienta Plataforma de aprendizaje EPIC, a través del cuaderno de calificaciones. Cabe destacar que la Universidad está avanzando fuertemente sobre el estudio de competencias para la enseñanza en los entornos digitales [16].

Recientemente, un grupo de científicos del MIT – Instituto Tecnológico de Massachusetts- que trabaja en el Laboratorio de Inteligencia Artificial y Ciencias de la Computación de la Universidad de Harvard, ha realizado un estudio sobre el aprendizaje en MOOCs –cursos on line-, a través del análisis de estudiantes a lo largo de 6.9 millones de sesiones de video a fin de estandarizar patrones de conducta y construir perfiles [18].
La investigación se basa en los hábitos de visualización de más de 100.000 estudiantes a lo largo de esas 6.9 millones de sesiones de video con

el único objetivo de medir el nivel de implicación. Pero lo realmente interesante en función de nuestro estudio, es que esta implicación es analizada en términos de tiempo de visualización, es decir "prestar atención" y comprobar si luego soluciona las evaluaciones propuestas, es decir, si logra la transferencia del residuo cognitivo con la consecuente retención y evocación de los aprendizajes.

Elementos del Trabajo y Metodología

En función del esclarecimiento del fenómeno, es decir, cómo juega la variable atención vinculada a los diferentes soportes en el contexto de la interfaz de aprendizaje, se efectúa un recorte de la variable conceptual atención.

En este sentido el término "prestar atención" se vincula con tres subtipos, la atención selectiva, la atención sostenida y la atención dividida [10]. La primera se asocia a la capacidad de filtrar estímulos del ambiente y focalizar sobre los estímulos que volitivamente se han seleccionado, es la atención selectiva. La segunda se vincula a prestar atención en términos de tiempo de duración sobre un mismo estímulo, es la atención sostenida.

Ambos procesos atencionales están íntimamente ligados a intereses y motivaciones que otorgan sintonía y dirección atencional [11].

La atención dividida es considerada un subtipo de la atención selectiva en cuanto se vincula a la cantidad de estímulos a los que podemos prestar atención simultáneamente. Es definida como una tarea compleja teniendo en cuenta los límites de la memoria de trabajo o atención selectiva medida en chunks –límite en porciones de información- [10]. Sin embargo, resulta especialmente interesante en cuanto podría ser sustento para el análisis de las multitareas en paralelo y la inteligencia lateral, conceptos asociados a los hábitos y costumbres de los Nativos Digitales [9].

La variable conceptual "prestar atención" se define entonces como la capacidad de seleccionar y sostener estímulos de modo volitivo y consciente facilitando así el producto del residuo cognitivo transferible.

La variable operacional "prestar atención" se mide desde la plataforma por cantidad de accesos a los materiales del LMS, tiempo asignado a cada uno y el nivel del residuo cognitivo transferido logrado a partir de la auto evaluación de cada tipo de material.

El diseño de investigación seleccionado es mixto, integrando de este modo las fortalezas de ambos enfoques, cuantitativos y cualitativos, partiendo de la idea acerca de que en la práctica no existe la total objetividad ni la total subjetividad, es inter subjetiva, en correspondencia con los aportes de los paradigmas pos positivista y constructivista [12].

El diseño metodológico aplicado al trabajo de investigación es el Secuencial DexPlis [13] abordado desde la perspectiva explicativa y exploratoria. La metodología consta de dos etapas consecutivas, se parte de la recolección de datos cuantitativos y su análisis a fin de obtener una primera aproximación al fenómeno, es decir cómo juega la variable atención vinculada a los diferentes soportes en el contexto de la interfaz de aprendizaje, y se continúa con una segunda etapa de recolección de datos cualitativos en donde se indaga acerca de los porqué de los cuantos hallados y al mismo tiempo las motivaciones profundas vinculadas al fenómeno de estudio. Finalmente se analizan los resultados cualitativos obtenidos y se integran en un análisis final de los resultados.

En definitiva, al integrar ambos enfoques se persigue el logro de una perspectiva más amplia y profunda de conocimiento del fenómeno – la modalidad que adquiere el proceso de atención en la interfaz digital de aprendizaje- y al mismo tiempo apoyar con mayor solidez las inferencias científicas.

El universo de la fase cuantitativa es de 332 casos, todos ellos estudiantes de las materias Base de Datos1 y Recursos Informáticos correspondientes a las Carreras de Licenciatura en Informática e Ingeniería en Software.

En esta primera etapa se utiliza la Plataforma de aprendizaje EPIC como instrumento de medición a partir de los reportes de actividad y uso de los materiales que brinda esta herramienta a través del cuaderno de calificaciones. Se consideran los reportes de accesos medidos en tiempo de apertura de cada material de la LMS y su auto evaluación,

realizada por cada uno de los estudiantes tanto para el formato texto como para video.

En la etapa cualitativa se prevé aplicar la técnica de Focus Group como instrumento de recolección de información, según diseño de una muestra que contemple una segmentación por nivel etario considerando edades según hábitos de nativos e inmigrantes digitales [9] y status de género según representación general de participación en los Centros de Aprendizaje Universitario (CAUs) de las Ciudades de Córdoba y Buenos Aires.

Para la salida cualitativa se confecciona una Guía de Pautas tomando en cuenta puntos de interés derivados de la primera etapa pero también lineamientos de corte netamente exploratorios. A modo de ejemplo puede mencionarse que desde los registros informáticos de la Plataforma EPIC detectamos casos en que los estudiantes aparentemente no ven los videos o no leen los PDF pero de todos modos sacan muy buenas notas en las auto evaluaciones, presuponemos que los deben bajar o tal vez tomar contacto con el material a través de otros o en grupo, es decir, no estamos seguros acerca de los procedimientos habituales que llevan adelante para acercarse al material de estudio. Datos de suma importancia para el esclarecimiento de la investigación en su totalidad.

La Guía de Pautas considera entonces, las siguientes líneas de sentido pertinentes a este Estudio:

- Significados asociados a la Categoría Educación a Distancia con e-learning con identificación de sus elementos y contexto (foro, wiki, e-mail, autoevaluaciones, entre otros). Grado de impacto e interés. Recordación. Vínculos e identidad.
- Criterios de Clasificación espontáneos de los materiales de estudio de la LMS en correlación con texto y video (ppt relatados, animaciones flash, entrevistas con líderes de opinión, entre otros). Asignación de ventajas y desventajas.
- Atributos asignados a los soportes texto y video. Identificación del "ideal" por tipo de soporte. Distancia entre el soporte ofrecido y el ideal.
- Modalidad de uso y procedimental de los soportes texto y video en cuanto a: tratamiento del material de estudio (¿Individual? ¿Grupal? ¿Se imprime? ¿Se baja?). Secuencia espontánea asociada a la interfaz, razones y preferencias.
- Expectativas de innovación y oportunidad asignada a los Mundos Virtuales como material de estudio en la plataforma LMS [17].

A partir de los ítems citados, esta metodología permite dilucidar novedosos significados vinculados a las modalidades atencionales que llevan adelante los estudiantes en el contexto del entorno digital y el aprendizaje vía plataforma LMS [14]. En este sentido, desde los Focus Group se obtendrán significados asociados vinculados a la experiencia educativa de los participantes.

En cuanto al criterio de sistematización de los emergentes obtenidos, los datos se conceptualizan y reducen, permitiendo la construcción de categorías y la definición de sus propiedades. Al mismo tiempo se analizan las relaciones entre las categorías -jerarquía, dependencia e implicancia, entre otras- obteniéndose así el material precursor de la interpretación, cuyo criterio de confiabilidad radica justamente en la coherencia interna, conceptual y metodológica, entre los datos y la interpretación de los mismos.

Resultados Preliminares

Etapa Cuantitativa

La Plataforma LMS incluye, entre otros materiales, los correspondientes a soporte texto y video. El soporte texto contiene básicamente archivos PDF, mientras en el soporte video se incluyen video-presentaciones ppt con explicaciones a través de imagen y audio y videos como los disponibles en Youtube. En cuanto la auto evaluación, se dividen en soporte texto y video y en todos los casos es de 20 preguntas cerradas, una alternativa es con opciones verdadero/falso y una respuesta correcta, otra con cinco opciones y más de una opción correcta y cinco opciones con sólo una respuesta verdadera.

Frente a la actividad de los estudiantes, la totalidad de los datos quedan registrados en la Plataforma LMS automáticamente, son exportados mediante planilla Excel donde se realizan los cálculos reduciendo al mínimo la posibilidad de errores de transcripción.

Se obtuvo del análisis de la Materia Base de Datos I con 215 casos, los siguientes cuadros:

TABLA 1. RESULTADOS DE LECTURA DE TEXTO

Archivo de Texto			
Categorías de Análisis	***Cantidad de accesos/ minutos***	***Cantidad de material***	***Promedio por material***
Accesos	***1893***	***22***	***86,5***
Tiempo de uso	*358,9*	*22*	*16,31*

TABLA 2. RESULTADOS DE VIDEO

Archivo de Video			
Categorías de Análisis	***Cantidad de accesos/minutos***	***Cantidad de material***	***Promedio por material***
Accesos	***155***	***10***	***65,5***
Tiempo de uso	*123,5*	*10*	*12,35*

TABLA 3. ANÁLISIS DEL RESIDUO COGNITIVO

Auto evaluación			
Categorías de Análisis	***Suma de % de aciertos***	***Cantidad de material***	***Promedio de autoevaluación***
Texto	***276,14***	***4***	***69,035***
Video	*282,5*	*4*	*70,512*

Los datos muestran que el soporte texto tiene en promedio mayor cantidad de accesos y necesita de más tiempo de lectura, mientras la autoevaluación promedio muestra una calificación ligeramente menor que la del soporte video.
El soporte video muestra en su desempeño promedio menor cantidad de accesos y la necesidad de menos tiempo de visualización, sin embargo la autoevaluación muestra una mejor calificación.

El análisis preliminar indica que para los videos no es necesario ni tantos accesos ni tanto tiempo dedicado al material para obtener notas superiores en la autoevaluación.

Se coteja con el análisis de la Materia Recursos Informáticos, sumando 117 casos.

	Archivo de Texto	**Archivo de Video**
Categorías de Análisis	**Cantidad de minutos abierto**	
Tiempo de Uso	***8199***	***2253***
Coeficiente (lectura/ video)	***3,64***	

	Archivo de Texto	**Archivo de Video**
Auto evaluación/ Promedio por alumno	***73,97***	***77,12***

En donde se observa, frente a similar estímulo en cuanto a complejidad del material, menos tiempo dedicado al soporte video, un poco más de tres veces menos, y sin embargo un resultado y puntaje sensiblemente mayor en la auto evaluación que la lograda con el soporte texto.

Discusión

Como se ha señalado, desde la visión cuantitativa la atención selectiva –"prestar atención"- promovida por el soporte video es relativamente mayor que la generada por el soporte texto. En este contexto es de interés esclarecer la "modalidad atencional" desde las motivaciones más profundas que pudieren sostener esa preferencia; con lo cual a partir del análisis y hallazgos cognitivos sería posible optimizar la construcción de la interfaz LMS ajustada desde ya a ciertos requerimientos a fin de mejorar significativamente el aprendizaje a distancia.

Queda pendiente la posibilidad de contrastar las autoevaluaciones con el examen final de la materia, pues el sistema no cuenta con la información de las notas finales, para lo que se previsto un procedimiento de carga manual de esta nota. En ese caso se ratificaría y/ o rectificaría el análisis del

presente estudio en la relación específica entre residuo cognitivo y el desempeño final de la materia en la interfaz LMS.

Conclusión

Reflexionando en términos de objetivos de enseñanza y considerando que efectivamente existe un efecto cognitivo singular a partir de las tecnologías, la pregunta más adecuada se vincula con la posibilidad de "dar forma" a un efecto cognitivo derivado de las TICs a través del diseño, la actividad y el entorno justamente a los fines de fomentar la abstracción consciente de habilidades y estrategias.

Si se toma en cuenta que el rendimiento en la autoevaluación es equivalente al residuo cognitivo transferido por vía alta a partir de la reflexión consciente, volitiva y meta cognitivamente guiada, se puede inferir que la atención selectiva promovida por el soporte video es mayor que la generada por el soporte texto.

Aunque esta diferencia es mínima, se toma en consideración la preferencia atencional del soporte video en términos de hipótesis introductoria para la salida cualitativa.

En este sentido y mediante el abordaje exploratorio, se impone rastrear las motivaciones que sostienen el interés por el soporte, así como los novedosos procesos cognitivos dinamizados a partir del nuevo entorno digital en el contexto de la Plataforma LMS.

Otro de los aspectos a tomar en cuenta desde el abordaje exploratorio cualitativo es la posibilidad de construir ambientes de aprendizaje con TICs –chequear los existentes y crear los novedosos- que contemplen las cogniciones distribuidas en los entornos culturales situados.

Agradecimientos

Agradecimiento a la Universidad Empresarial Siglo 21 de Córdoba, Argentina, ámbito donde se realiza el Proyecto de Investigación 2013 "El impacto del entorno conectivo digital en los procesos cognitivos, en especial la atención; hacia el diseño de ambientes de aprendizaje con Tecnologías de la Información y la Comunicación (TICs) que generen atención"

Referencias

[1] Perkins, D. N., Bloberson, T., & Salomon, G. (1992). Coparticipando en el conocimiento: la ampliación de la inteligencia humana con las tecnologías inteligentes. CL & E: Comunicación, lenguaje y educación, (13),6-22.

[2] Pea, R. D. (1993). Practices of distributed intelligence and designs for education. Distributed cognitions: Psychological and educational considerations, 47-87

[3] Salomon, G. (1992). Las diversas influencias de la tecnología en desarrollo de la mente. Infancia y Aprendizaje: Journal for the Study of Education and Development, (58), 143-159.

[4] Cole, Michael and Engestrom Yrjo (2001): Distributed cognitions. Psychological and educational considerations. Edite by Gavriel Salomon. Cambridge University Press. USA.

[5] Salomon, G. (2001). Cogniciones distribuidas. Consideraciones psicológicas y educativas. Buenos Aires: Amorrortu.

[6] Perinat, A. (2011). Vygotsky's socio-cultural approach as an alternative to the biological-maturationist and idealistic explanations of human development.Psychologia. Avances de la Disciplina, 5(2), 137-145.

[7] Perkins, David N. (1986). "Knowledge as design: Teaching thinking through content. And Thinking frames: An Integrative Perspective on Teaching Cognitive Skills" en J. B. Baron y R. J. Stemberg (eds.). New York: W. H. Freeman, 1986.

[8] Engeström, Y. (1999). Activity theory and individual and social transformation.Perspectives on activity theory, 19-38.

[9] Prensky, M. (2001). Digital natives, digital immigrants part 1. On the horizon,9(5), 1-6.

[10] Téllez, J. A., & Muñoz, J. A. T. (2005). La comprensión de los textos escritos y la psicología cognitiva: más allá del procesamiento de la información. Librería-Editorial Dykinson.l

[11] Lozano Fernández, L. M., & García-Cueto, E. (2000). Relación entre motivación y aprendizaje. Psicothema, 12(Suplemento), 344-347.

[12] Collado, C. F., Sampieri, R. H., & Lucio, P. B. (1998). Metodología de la investigación. McGraw-Hill Interamericana.

[13] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2007). Fundamentos de metodología de la investigación. Madrid [etc.]: McGraw-Hill.

[14] Castoriadis, C. (1989). La institución imaginaria de la sociedad (Vol. 2). Barcelona: Tusquets.

[15] Etcheverry, P. (2013). "Imaginario social y construcción de subjetividad desde el entorno telemático: hacia una investigación posible". TE & ET. Revista iberoamericana de Tecnologia en educación y Educacion de la tecnologica Edicion Especial Nro 9 abril 2014 Pag 69-74. ISSN 1850-9959- RedUNCI - UNLP

[16] UESiglo21. (2013). Modelo de Aprendizaje basado en Competencias http://www.youtube.com/watch?v=VDCf2DWJJQ0. Consultado el 12/02/2014.

[17] Maldonado, C.; Etcheverry P (2013)." Hacia el uso de Mundos Virtuales en "Blended Learning 2.0" Conaiisi 2013 Congreso Nacional de Ingeniería Informática/Sistemas de Información (En línea) ISSN : 2347-0372 Córdoba, Argentina (2013).

[18] Guo, P. J., Kim, J., & Rubin, R. (2014) How video production affects student engagement, an empirical study of moc's videos. Proceedings of the first ACM Conference on Learning@ scale conference (PP. 41-50). ACM. ISBN 978-1-4503-2669-8

Datos de Contacto:
Prof. Lic. Patricia Etcheverry patriciaetcheverry@yahoo.com.ar
DEA Ing. Calixto Maldonado calixtomaldonado@hotmail.com

Avances en el uso del Aula Virtual con nuevas propuestas de mejora

Verónica L. Vanoli, M. Lucrecia Lavirgen
Facultad Regional Bahía Blanca
Universidad Tecnológica Nacional
11 de Abril 461
vvanoli@frbb.utn.edu.ar, lucrecialavirgen@gmail.com

Resumen

El advenimiento de la tecnología ha posibilitado grandes cambios dentro de la sociedad en general y en el ámbito educativo en particular. Mediante la realización de diversas actividades tales como talleres, se incentiva al docente a desarrollar nuevas competencias digitales haciendo uso de las herramientas que provee el Entorno Virtual de Enseñanza y Aprendizaje (EVEA) de la Facultad Regional Bahía Blanca (FRBB) de la Universidad Tecnológica Nacional (UTN).

En este trabajo se comienza analizando las conclusiones de una encuesta realizada en el año 2013 a los docentes sobre el uso del entorno, y cómo los docentes administradores del mismo han seguido colaborando mediante la enseñanza de nuevas herramientas tales como el desarrollo de la Evaluación Diagnóstica mediante las actividades Encuesta y/o Cuestionario, la incorporación de material en formato video a sus clases, entre otros. Además de brindar la posibilidad que cada departamento cuente con un espacio común (dentro del Aula Virtual) para sus respectivas materias.

Se presentan nuevos recursos/actividades que ha venido desarrollando el servicio de la administración del área TIC encargada del EVEA, para que los docentes puedan hacer un uso apropiado de la tecnología en el proceso de enseñanza-aprendizaje.

1. Introducción

Las nuevas tecnologías aplicadas al ámbito de la educación han facilitado la aparición de Entornos Virtuales de Enseñanza y Aprendizaje (EVEA) que desarrollan procesos de enseñanza-aprendizaje en los distintos niveles del sistema educativo como oportunidad para crear un nuevo ecosistema de formación basado en elementos como [1]:

- conectividad,
- colaboración,
- convergencia [informática+información+comunicación],
- comunidad,
- creatividad,
- etc.

Un modelo pedagógico en la educación tecnológica tiene por finalidad dar sentido y sostener la tarea de promover y acompañar el aprendizaje del mundo de los productos, de sus procesos, de los medios de comunicación, de las tecnologías de la información y de la lectura en profundidad de fenómenos y productos tecnológicos. La tecnología no se enseña, sino que se aprende a partir de experiencias [2].

En la FRBB contamos con un EVEA denominado Aula Virtual (AV) cuyo programa informático que permite diseñar, estructurar y realizar diversos procesos formativos a través de Internet [3] es la herramienta Moodle. La modalidad implementada es una hibridación desde la optimización de la mediación de contenidos y estrategias didácticas entre: Aula Extendida, Aula Invertida y Aprendizaje combinado o mixto (B-Learning) [4].

Se comenzó a interactuar con la tecnología en la educación en el año 2007, subiendo el material educativo de algunas pocas cátedras en la página web de la FRBB. Actualmente el AV se encuentra implementado desde el año 2009 y en constante crecimiento respecto a los espacios-materias. La figura 1 muestra la evolución hasta la fecha, y se puede apreciar el salto de los 2 últimos años en un 20%. Con respecto a la cantidad de usuarios el aumento es proporcional entre alumnos y docentes, y se acerca a un 18% respecto al año anterior (2012).



Figura 1: Evolución del AV desde sus comienzos a la fecha

El siguiente trabajo toma como punto de partida las conclusiones expuestas en un trabajo anterior [5]. Cada conclusión (remarcada en un recuadro) presentará una explicación de lo que se ha hecho al respecto, luego la pregunta asociada presentada a los docentes en una encuesta[1] y por último, la respuesta a la misma, con las conclusiones correspondientes. La muestra en este caso, se constituye con un total de 82 docentes que han participado de la encuesta, respondiendo a 10 preguntas relacionadas al uso que cada uno le ha dado al AV. El trabajo se complementa con nuevas propuestas de mejora y crecimiento.

2. Fundamentos sobre Conclusiones

Realizar jornadas de capacitación sobre cómo realizar la evaluación diagnóstica y su procesamiento automático.

En los talleres optativos que se realizan anualmente en el marco del área TIC[2], con el objetivo de capacitar al personal docente, se incorporó este nuevo tema referente a la evaluación diagnóstica. Para ello, se les impartió a un grupo de 20 docentes los conocimientos necesarios a partir de las actividades Cuestionario y Encuesta. Iniciando en las características generales de edición y el armado de preguntas. Se ha hecho un énfasis en el buen uso de los tipos de preguntas para esta clase de evaluaciones y se remarcó cómo obtener los resultados y algunas sugerencias para el procesamiento posterior adecuado.

Pregunta 1. ¿Pudo implementar la Evaluación Diagnóstica mediante las actividades Encuesta y/o Cuestionario? (Si/No)

El resultado de la primera pregunta (figura 2) arrojó un resultado bajo en cuanto a la implementación, sólo el 23% logró poner en práctica lo aprendido durante el taller (situación recurrente en los docentes que asisten al mismo).

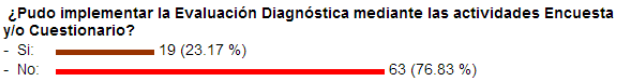


Figura 2: Primera pregunta de la encuesta

Aquellos docentes que hayan respondido en forma afirmativa, se les permitió explayarse un poco más con la siguiente pregunta:

Pregunta 2. En caso afirmativo de la pregunta anterior, por favor, comente su experiencia.

Si bien, la pregunta apuntaba a los casos afirmativos, las opiniones fueron diversas en cuando a si pudieron o no implementar la Evaluación Diagnóstica. En varios casos reconocieron la falta de tiempo para el armado de la encuesta/cuestionario, asegurando su implementación para un próximo año.

En cuanto a los comentarios según su experiencia, se han recolectado expresiones textuales como:

- "Muy positivo"
- "Fue muy buena ya que permitió que la contestaran todos los alumnos al tenerla disponible mas tiempo que la duración de una clase y facilitó también el procesamiento de los datos obtenidos"
- "Fue muy positiva para los alumnos (que pudieron hacerla en forma más rápida) y para los docentes (es una ventaja no tener que corregir y que los datos ya estén tabulados)"
- "Fue muy buena, pues la corrección y obtención de resultados fue inmediata. Además del ahorro de papel"
- "Muy buena experiencia"
- "Muy buena, ya que ahorra mucho tiempo de procesamiento de datos"
- "Me ayudó a evaluar la situación personal de cada alumno haciendo rápidamente una conclusión al respecto"
- "Brinda en forma automática información muy útil"
- "Altamente positiva"
- "Se obtuvieron buenos resultados puesto que agilizó la estadística y obligó a los alumnos a usar el aula virtual. La misma tuvo carácter obligatorio y le pusimos un plazo de unos 10 días para completarla"
- "Lo interesante de usar el aula virtual es que la corrección y los resultados los realiza el sistema, con el consecuente ahorro de tiempo. Además, las encuestas pueden quedar habilitadas mas tiempo,

[1] Se les presentó a los docentes una encuesta iniciada al cierre del ciclo 2013 y finalizada en el mes de mayo de 2014.
[2] Los talleres están a cargo de la administración del AV, que cuenta con personal docente capacitado en el tema para su correspondiente dictado.

para que aquel alumno que no asistió a completar el cuestionario en clase, pueda llenarlo igual y tener un perfil completo del curso"

Ofrecerles a los docentes, como complemento en su materia, la incorporación de videos grabados en sus clases.

En el transcurso del año, una asignatura relacionada a informática de la FRBB hizo una prueba al respecto, que resultó satisfactoria. Contó con personal capacitado y equipamiento acorde correspondiente a la misma institución. Los videos grabados fueron subidos al AV y presentados a los alumnos para su correspondiente visualización. La experiencia motivó a invitar al resto de las asignaturas, que cuentan con un espacio en el AV, puedan contar con este recurso. En busca del acuerdo, se presentó la siguiente pregunta:

Pregunta 3. ¿Le interesaría incorporar a su materia material en formato video referente a sus clases? (Si/No por el momento/No)

Y la respuesta fue la esperada, casi un 61% de los docentes encuestados demostró su interés en el tema. La figura 3 muestra este resultado.

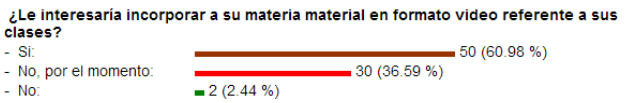


Figura 3: Tercera pregunta de la encuesta

Permitir la interrelación entre materias, que favorezca las correlatividades o la necesidad de contenidos mínimos requeridos.

Asesorar a los departamentos de la FRBB para su máximo aprovechamiento en forma académica y administrativa.

Para comenzar con una idea clara de cómo focalizar el tema, fue necesario dar participación a los docentes, para que opinen al respecto, a partir de la siguiente pregunta:

Pregunta 10. Si cada departamento contara con un espacio común (dentro del Aula Virtual) para sus respectivas materias, ¿qué tipo de actividades incluirías?

De las diversas opiniones (en un 41%) se pueden apreciar dos enfoques bien marcados, aprovechar el espacio tanto para los alumnos como para los docentes que pertenecen a cada departamento. Una de las propuestas mas destacadas recae en la utilización de foros para compartir preocupaciones, inquietudes, apreciaciones, experiencias y problemáticas. Las carteleras de noticias o anuncios también fueron nombradas en mayor medida. Sumado a esto la alternativa de comunicación asincrónica [6] a través de los correos electrónicos. En términos generales sugirieron actividades intercátedra, interdisciplinarias y de integración (tanto horizontal como transversal). Se planteó el intercambio de materiales, bibliografías, trabajos prácticos, exámenes y evaluaciones diagnósticas. La inclusión de notas de pedido, para todo lo que refiera a la compra de material, herramientas, etc. Y por último, 3 casos aislados, pero que no restan importancia, como lo es la posibilidad de espacios abiertos y colaborativos [7] donde los docentes puedan formular descripciones de proyectos, laboratorios, etc. Realizar encuestas por carreras. Y la presentación del desarrollo de habilidades profesionales destinado a los egresados.

El departamento de Ingeniería Electrónica, previo a la encuesta, ya había solicitado y se le había creado (ese mismo año) un espacio en común denominado Comisión Curricular, el mismo sólo incluye encuestas destinadas a los alumnos del 4to año de la carrera Ingeniería Electrónica con el objetivo de mejorar el cursado de los mismos.

3. Refuerzos a consultas previas

A partir de los porcentajes bajos obtenidos en relación al uso de actividades en los respectivos espacios, dentro del AV, se reformuló la pregunta y se volvió a insistir en cuáles son las necesidades presentadas por los docentes respecto a las actividades didácticas medidas a través de un entorno virtual, también conocidas como e-actividades [8].

Pregunta 7. ¿Cuál de las siguientes actividades desea o necesita aprender? (Cuestionario/Encuesta/Foro/Glosario/Tareas - Subida de archivos/Wiki)

Los resultados obtenidos (figura 4) en la necesidad de aprender el uso de las actividades, se los describir ordenados de mayor a menor respecto al porcentaje. Actividades de comunicación: foro y encuesta (87%), actividades de evaluación: tarea y cuestionario (77%) y actividades de trabajo en grupo: glosario y wiki (71%).

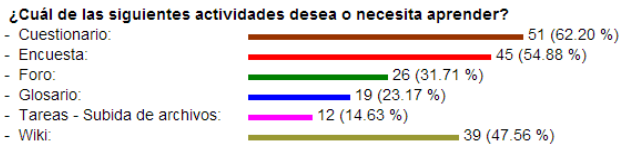


Figura 4: Séptima pregunta de la encuesta

4. Generalidades del Aula Virtual

Dado que el espacio destinado a la biblioteca, donde la bibliotecaria de la FRBB sube material digitalizado, ha mejorado su presentación y ha incorporado nuevos materiales de consulta, y dado que el espacio de ayuda virtual a los docentes ha crecido, se han formulado las siguientes preguntas:

Pregunta 4. ¿Le resulta de utilidad el espacio de la Biblioteca (Menú)? (Si/No lo he utilizado/No)

Del 29% de docentes que ingresan a este espacio bibliotecario, un 27% reconoce que le resulta útil. El porcentaje más alto (71%) se lo lleva la no utilización de dicho espacio. Los motivos pueden deberse: a la falta de necesidad, a la concurrencia personal al espacio físico que la biblioteca posee en la institución, al desconocimiento de este espacio en el AV, entre otros. La figura 5 muestra lo expuesto anteriormente.



Figura 5: Cuarta pregunta de la encuesta

Pregunta 5. ¿Le resulta de utilidad el espacio de Ayuda (Menú)? (Si/No lo he utilizado/No)

Del 46% que utilizan el espacio, a modo de consulta, un 44% remarca que le resulta de utilidad. Los resultados se grafican en la figura 6.

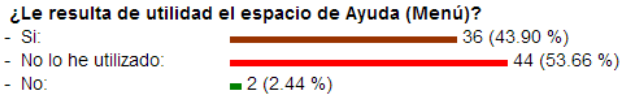


Figura 6: Quinta pregunta de la encuesta

Para comprender mejor las respuestas presentadas en el gráfico anterior (figura 6), se le acopló la posibilidad de una respuesta abierta con la siguiente pregunta:

Pregunta 6. En relación a la pregunta anterior, ¿qué le agregaría o modificaría al espacio de ayuda?

Dado que esta pregunta no requería obligatoriedad en su respuesta, se deduce que un total de 65 docentes han optado por no responder. La no respuesta o el alto porcentaje (53%) en la no utilización de dicho espacio (figura 5) marcan la falta de conocimiento en la existencia de dicho espacio o por disolver sus dudas a través del servicio de correo electrónico o por recurrir personalmente a la oficina que se encarga de la administración del AV. El 44% que ha respondido afirmativamente a la utilidad de la ayuda, agregó que hasta el momento está conforme con las secciones que presenta dicho espacio. Sólo un docente sugirió incorporar un servicio de preguntas y respuestas frecuentes. El espacio cuenta con servicio de foros y hasta el momento, no hemos tenido un caudal de dudas generales que requiera la necesidad de creación de lo solicitado por el docente.

Es muy común escuchar la frase "quiero hacer algo y no sé cómo hacerlo..." Situación que genera adaptar la tecnología a las necesidades del docente. Para ello, y a modo de incentivo (alentarlos para que no renuncien a la tecnología y cuenten con una herramienta que complementa sus clases presenciales) se les dejó abierta una pregunta referida a este tema.

Pregunta 8. Comente si está necesitando implementar alguna actividad y no encuentra cómo hacerlo.

Un 74% de docentes expresó no necesitar, por el momento, actividad alguna. El resto (un 26%) han aportado diferentes necesidades, de las cuales, algunas referentes a las actividades que forman parte del AV pero que no han podido implementarlas aún. Sólo un 9% de los docentes propuso nuevas actividades como la incorporación de corchos virtuales [9], simuladores, templates o plantillas específicos, actividades virtuales extras, visores de archivos tipo Power Point, etc.

En el año se incorporó una escala de calificación nueva para que los docentes puedan ampliar las posibilidades de resultados de evaluaciones, que no sea en formato numérico. La nueva escala se la denominó Resultado con dos alternativas posibles: Aprobado o Desaprobado. Parte del aviso enviado a los docentes expresaba lo siguiente: *"Aquellos docentes que hacen uso de la actividad Tareas, podrán evaluar de esta forma, en caso de requerirlo. Al crear una tarea (por ejemplo, una entrega de trabajos prácticos) dentro de Ajustes generales/Calificación podrán optar por esta escala. Tengan en cuenta, que al ser una Escala, la misma quedará reflejada en la sección Calificaciones, junto con el resto de las calificaciones numéricas. Si alguno de Uds. califica de otro forma y necesita crear una escala nueva que se adapte a dicho formato, podrán solicitarla sin ningún compromiso".*

Para comprobar si quedó claro tanto el aviso como la nueva incorporación, la siguiente pregunta se basó en la opción que la incluye y manifiesta.

Pregunta 9. ¿Hace uso del libro de calificaciones? (Con todas las actividades/Con algunas actividades/No lo utilizo)

El resultado (figura 7) develó que el 85% de los docentes no utilizan el libro de calificaciones. Y sólo el 15% sí lo hacen. Esto puede deberse al desconocimiento de las actividades que permiten la calificación dentro del AV.

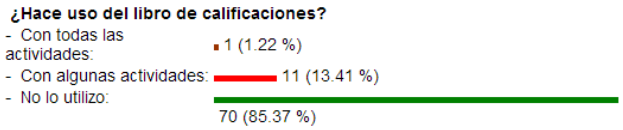


Figura 7: Novena pregunta de la encuesta

Debido al incremento del uso de actividades que establecen fecha y hora de apertura y cierre, y a algunos inconvenientes surgidos en las computadoras del centro de cómputos de la FRBB, se incorporó un bloque denominado *Relojes* (Figura 8) (en la portada del AV, con posibilidad de inclusión en los respectivos espacios de cátedras) que muestra y diferencia la hora del servidor (donde se encuentra instalado el AV) y la hora de la computadora del usuario registrado. Luego se les aclaró que este entorno tiene en cuenta siempre la hora del servidor, por lo tanto, es el tiempo a visualizar correspondiente a las actividades. Para mayor tranquilidad se les comentó que quien administra el servidor mantiene actualizado dicho horario.

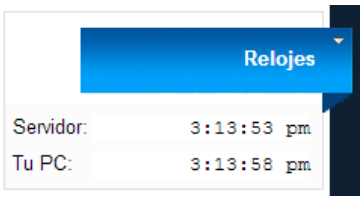


Figura 8: Bloque Relojes

Cabe destacar que el bloque (Relojes) tuvo mucha aceptación, porque varios docentes lo agregaron a sus respectivas materias.

5. Conclusiones

Este tipo de encuestas permiten conocer el estado del arte reflejando la situación actual, ayudando al crecimiento de los servicios prestados. También permite que los docentes se sientan acompañados y descubran los nuevos y existentes recursos tecnológicos en favor de la educación.

Las opiniones referidas a la incorporación de nuevos recursos/actividades demuestran un interés en el AV y alientan la continuidad de este servicio por parte de la administración en el área TIC. Otra forma de demostrar interés se ve reflejada en el aumento de solicitudes de apertura de espacios destinados a materias que año a año se manifiestan (ver nuevamente la figura 1).

En cuanto a los resultados puntuales de la encuesta queda claro que:

- Continuaremos con las capacitaciones (talleres) haciendo foco en los resultados de la pregunta 7.
- Comenzaremos a implementar las grabaciones de videos para luego subirlos al AV. Seguramente previo a ello, se hará una demostración de la experiencia ya hecha con éxito.
- Crearemos los espacios por departamento según los pedidos, clasificándolos por los dos enfoques detectados (alumnos/docentes) y las propuestas presentadas.
- Continuaremos con el crecimiento del espacio de la biblioteca y el espacio de ayuda.
- Investigaremos y de acuerdo a su viabilidad, incorporaremos las nuevas actividades solicitadas en la pregunta 8.

Las tareas pendientes en el porvenir son:

- Revisar la encuesta y responder a cada encuestado (en forma personalizada) sus respectivas inquietudes, sugerencias o reclamos.
- Solucionar problemas de dirección de correo errónea correspondiente a los alumnos, para reforzar el medio de comunicación. Haciendo partícipe al departamento de Alumnos de la FRBB a través del Sysacad (Sistema Académico de Alumnos).
- Ampliando los criterios que presenta el libro de calificaciones, incorporar alguna actividad que permita el seguimiento de los alumnos, desde el inicio de su cursado hasta el cierre de la materia en el AV.
- Actualizar la versión del AV (a un Moodle 2.0+), dadas las limitaciones de la versión actual (Moodle 1.9), manteniendo toda la información que el entorno posee al momento de la actualización.

6. Agradecimientos

Este trabajo no sería posible sin la participación no obligatoria de los docentes con sus respuestas aportadas en la encuesta, bajo un recordatorio constante por parte de la Administración del AV.

7. Referencias

[1] Lorenzo García Aretio. “Sociedad del conocimiento y educación”. Editorial UNED. ISBN: 978-84-362-6573-6. Madrid – España 2012.

[2] Daniel Prieto Castillo. “La comunicación en la educación”. Editorial Stella. Ediciones La Crujía. Buenos Aires – Argentina 2004.

[3] Linda Castañeda Quintero y Patricia López Vicent. “Entornos Virtuales de Enseñanza Aprendizaje Libres: Moodle Para Profesores”. En María Paz Prendes Espinosa “Herramientas Telemáticas para la Enseñanza Universitaria en

el Marco del Espacio Europeo De Educación Superior". Grupo de Investigación de Tecnología Educativa. Universidad de Murcia. CD – ROM. ISBN: 978-84-611-7947-3. 2007

[4] Alejandra Zangara. "Apostillas sobre los conceptos básicos de educación a distancia o… una brújula en el mundo de la virtualidad". Universidad Nacional de La Plata, Facultad de Informática. Doctorado en Tecnología Informática. Enero, 2014.

[5] M. Lucrecia Lavirgen y Verónica L. Vanoli. "Incidencia del Uso del Aula Virtual en los Distintos Departamentos de la Facultad". III Jornadas de Enseñanza de la Ingeniería. UTN - FRBB. 5 y 6 de septiembre de 2013. Volumen 1. ISSN Nº: 2313-9056. Pág. 92-95.

[6] Lorenzo García Aretio. "Fundamento y Componentes de la Educación a Distancia". Revista Iberoamericana de Educación a Distancia. Pág. 43 a 61. Vol. 2. Nro. 2. Diciembre de 1999.

[7] Luz María Zañartu Correa. "Aprendizaje colaborativo: una nueva forma de Diálogo Interpersonal y en Red". Revista Digital de Educación y Nuevas Tecnologías - Contexto Educativo. Nro. 28. Año V. 2000

[8] Cecilia Sanz y Alejandra Zangara. "Las e-actividades como elemento central en el diseño de propuestas de educación mediada. Una posible definición y clasificación". I Jornadas Nacionales de TICs e Innovación en el Aula. III Jornadas de Experiencias en Educación a Distancia de la Universidad Nacional de La Plata. Dirección de Educación a Distancia. 1 y 2 de diciembre del 2011.

[9] Alberto García Pérez. "Las TIC en la promoción de la lectura y la escritura". X Jornadas de Intercambio de Experiencias Educativas. Avilés. 25 de junio de 2012.

Software Educativo y Estrategia Didáctica para la Enseñanza de Conceptos y Métodos en Simulación de Eventos Discretos

Darío Weitz
Departamento Ingeniería en Sistemas de Información
Facultad Regional Rosario, Universidad Tecnológica Nacional
Rosario
dar.wtz@gmail.com

Abstract

Simulación es una asignatura obligatoria en el plan de estudio de Ingeniería en Sistemas de Información en la Universidad Tecnológica Nacional. Sus contenidos incluyen definiciones teóricas y metodológicas, alto nivel de abstracción, procedimientos de cálculos intensivos y una sólida base matemática y estadística. En este trabajo, exploramos como la utilización de un software educativo orientado a sistemas de espera y sistemas de inventario puede ayudar a la comprensión y el aprendizaje de conceptos básicos y elementos metodológicos presentes en un curso de Simulación. El software educativo incorpora elementos de animación como estrategia didáctica. La técnica de animación facilita la enseñanza de los contenidos correspondientes a la asignatura. El software demostró ser una herramienta apropiada para propósitos educativos a nivel universitario de grado y puede ser utilizado en cursos introductorios de Simulación en cualquier carrera de ingeniería.

1. Introducción

Simulación es una asignatura obligatoria en numerosas carreras universitarias de ingeniería. Está presente en los planes de estudios de Ingeniería en Sistemas, Industrial y de otras ingenierías *no tradicionales* que se han sumado a la oferta educativa de grado en los últimos años. La asignatura conforma el nexo de vinculación entre la línea curricular de Sistemas, Investigación Operativa y Matemática. Los contenidos del curso suelen poseer definiciones teóricas y metodológicas, alto nivel de abstracción, procedimientos de cálculos intensivos y requerir de los estudiantes una sólida base matemática y estadística.

Un desafío que se le plantea a docentes de estos cursos es que deben enseñar las definiciones teóricas y metodológicas, los procedimientos de cálculos y las herramientas informáticas sin perder de vista que las mismas van a evolucionar durante la vida laboral del futuro profesional. En tal sentido, es conveniente que los docentes de los cursos de Simulación hagan uso de nuevas herramientas didácticas para poder superar el citado desafío.

SIMUL8 y Arena son dos entornos de simulación habitualmente utilizados en un curso introductorio de simulación. Sin embargo, los mismos tienen importantes limitaciones al momento en que el docente, al comienzo del curso, debe explicar las metodologías subyacentes en la construcción de diagramas y en los procedimientos de cálculos de la simulación de eventos discretos. Desde un punto de vista pedagógico, los entornos de simulación deberían utilizarse para complementar la tarea docente una vez que los alumnos tienen incorporados los conceptos básicos de la asignatura.

Los métodos tradicionales de enseñanza en ingeniería se basan en gráficos y dibujos estáticos. Por su parte, las computadoras poseen la capacidad de integrar elementos de animación como herramientas de enseñanza – aprendizaje. La técnica de animación tiene un gran potencial para cumplir una función pedagógica en la enseñanza de conceptos complejos, particularmente aquellos que no son sencillos de describir verbalmente. También ayuda a crear imágenes mentales de sistemas que poseen elementos que cambian en tiempo y espacio. Sirve además para mostrar como la manipulación de una

variable afecta a otras o la existencia de influencias simultáneas en sistemas cerrados.

La sencillez con que se pueden implementar elementos de animación ha resultado en numerosos casos en *clases muy vistosas* pero deficientes en lo que respecta a la posibilidad de los alumnos para procesar información suministrada en forma no apropiada y poco efectiva. Por lo tanto, es importante evaluar si las características y los contenidos del curso justifican la inclusión de elementos de animación, utilizando para tal fin las teorías vigentes sobre aprendizaje multimedia.

El objetivo del presente trabajo es mostrar como se puede facilitar la comprensión y el aprendizaje de conceptos básicos y elementos metodológicos presentes en un curso introductorio de simulación, mediante el uso de un software educativo orientado a sistemas de espera y sistemas de inventario que incluye elementos de animación como estrategia didáctica.

2. Análisis de la Estrategia Didáctica

La carrera de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional tiene a la asignatura Simulación en el cuarto año del Plan de Estudios vigente. Entre los objetivos específicos del curso se encuentran: i) reconocer la clase de problemas que pueden ser resueltos mediante la técnica de simulación; ii) aplicar la técnica al estudio de sistemas de espera, sistemas de inventario y sistemas de producción.

Para satisfacer en tiempo y contenido a estos objetivos específicos, se decidió desarrollar un software educativo que pudiera ser utilizado como apoyo a la tarea docente en el aula. La primera etapa del citado software tiene como propósito presentar a los alumnos un tratamiento comprensible y rápido de un sistema de espera y de un sistema de inventario.

Puesto que la técnica de animación ha demostrado ser una herramienta valiosa para enseñar conceptos complejos que involucran cambios en el tiempo, sistemas afectados por influencias simultáneas o la sensibilidad de un sistema al cambio en una variable [1], se procedió a analizar si tanto los objetivos generales de enseñanza como los conceptos básicos impartidos durante el dictado de un curso introductorio de Simulación justifican el uso de elementos de animación.

3. Animación y Procesos Cognitivos

Se define a la animación como una visualización gráfica que cambia su estructura u otra propiedad a lo largo del tiempo, y que dispara la percepción de un cambio continuo [2]. En ambientes educativos, se utilizan a menudo animaciones para mejorar la comprensión por parte de los estudiantes de ciertos procesos complejos o de conceptos abstractos que cambian en tiempo y espacio. La animación ha sido utilizada en la enseñanza de sistemas complejos tales como mecánicos, biológicos, físicos, operacionales y computacionales. Se han utilizado animaciones en ciencias naturales tales como física, química, biología, etc., para visualizar conceptos científicos y sus relaciones a través de representaciones gráficas abstractas. También se la ha utilizado en la enseñanza de software para lograr una mejor comprensión de los algoritmos y las estructuras de datos [3].

La literatura reporta casos de éxito en el uso de animaciones respecto a gráficas estáticas [4], como así también situaciones donde no solo la animación sino tampoco el uso de recursos gráficos resultaron ventajosos frente a las técnicas de enseñanza tradicionales [5]. Aún en los casos exitosos, se ha observado que la animación conlleva demandas muy específicas en el proceso de aprendizaje de los alumnos: deben procesar información que está cambiando en forma continua y tienen que identificar y relacionar elementos que van tomando diferentes ubicaciones espaciales a lo largo del tiempo. Como resultado, algunos alumnos se ven abrumados por la intensidad de la información que reciben y no son capaces de procesar en forma sistemática el material que se les presenta. Por lo tanto, es recomendable evaluar *a priori* si las características y los contenidos del curso son apropiados para utilizar técnicas de animación, utilizando para tal fin las teorías vigentes sobre aprendizaje multimedia.

La teoría cognitiva del aprendizaje multimedia es una de las teorías más abarcativas relacionadas con el aprendizaje basado en dispositivos multimedios en general y con animaciones en particular [6]. Entre los supuestos básicos de esta teoría se incluye la hipótesis de la codificación dual integrada.

La hipótesis de la codificación dual integrada [6] asume que el proceso de conocimiento consiste principalmente en la actividad de dos partes interconectadas pero funcionalmente independientes y con distintos sistemas simbólicos. Una parte codifica eventos verbales (palabras) mientras que la otra codifica eventos no verbales (dibujos). Cuando el conocimiento se codifica en ambos modos, verbal y no verbal, los observadores construyen en el cerebro una representación dual y establecen conexiones referenciales entre esas dos representaciones. Como resultado, la transferencia de conocimiento se da en forma más eficiente y robusta que cuando se utiliza alguna otra técnica que no involucra codificación dual.

Otra teoría de aprendizaje relacionada con animaciones es la teoría de la fidelidad epistémica [7].

Parte de la premisa de la dificultad de representar un sistema de conceptos a estudiantes que están alejados de la cosmovisión del profesor. La teoría asume que se puede mejorar la transferencia de conocimientos cuando la representación externa describe al modelo mental del experto con la mayor fidelidad posible. A mayor fidelidad entre las imágenes visualizadas por los alumnos y el modelo mental del profesor, más eficiente y robusta es la transferencia de ese modelo mental al receptor de la visualización, quien la decodifica e internaliza con mayor claridad y menor ambigüedad.

Otros estudios [5] han mostrado que gráficos, y particularmente animaciones, pueden facilitar la comprensión y el aprendizaje solo si son apropiados y están cuidadosamente diseñados. En este sentido, las representaciones gráficas deben conformar tanto al principio de congruencia como al principio de aprehensión.

El principio de congruencia establece que para lograr una representación gráfica efectiva, la estructura y el contenido de la representación externa deben corresponderse con la estructura y el contenido deseado de la representación interna. El contenido y el formato de la imagen deben corresponderse con el contenido y el formato de los conceptos que son presentados; según lo anterior, la animación debería ser una herramienta eficiente para mostrar cambios en un sistema a lo largo del tiempo. Además, la animación debería ser efectiva para expresar procesos o una forma natural de expresar un cambio metafórico.

La estrategia de animación podría ser ineficiente si no cumple con el segundo principio para una representación gráfica efectiva, el principio de aprehensión, el cual establece que las imágenes deben ser percibidas con exactitud y concebidas apropiadamente [5]. La percepción se dificulta si las imágenes son muy complejas o pasan muy rápidas.

Weiss et al. [8] presentan un conjunto de heurísticas y consejos prácticos relacionados con el uso de animación en software educativo. En primer lugar describen la naturaleza específica de la animación examinando sus características y propósitos. Afirman que la animación puede cumplir con uno o varios de los siguientes propósitos o funciones: cosmética o decorativa; para conseguir la atención del alumno; motivadora; como estrategia de presentación; para clarificación. También analizan la naturaleza física de la animación a través de la estructura física (textura, color) y el nivel de fidelidad. Distinguen entre fidelidad física –cuan parecida es la animación al objeto o proceso del mundo real–, y la fidelidad funcional –cuan cercana se comporta la animación al compararla con el objeto o proceso del mundo real–.

El movimiento y la trayectoria son características inherentes en una animación. Por lo tanto, no sería imprescindible el uso de elementos de animación si tales características no se encuentran presentes en los contenidos a enseñar. Si el movimiento y la trayectoria están presentes, la animación se justifica si cumple con funciones de clarificación y/o como estrategia de presentación. La animación como estrategia de presentación implica dar un contexto visual a las ideas, ayudar a la codificación dual y es útil para mostrar ejemplos de sistemas dinámicos o con alto nivel de abstracción. La animación como función de clarificación implica promover el entendimiento de un nuevo concepto o de una nueva relación sin necesidad de agregar información textual adicional.

Si los objetivos generales de enseñanza y los contenidos básicos justifican el uso de animaciones, se debe analizar el nivel de fidelidad (física o funcional) y la estructura física. Pueden ser diferentes según se trate de enseñar conceptos o procedimientos. La animación es muy valiosa cuando se enseñan conceptos complejos que involucran cambios en el tiempo, sistemas afectados por influencias simultáneas o la sensibilidad de un sistema al cambio de una variable. En el caso de influencias simultáneas, la animación puede ayudar a superar la tendencia natural que tienen los seres humanos a procesar la información de manera secuencial. Rieber y Kini [1] afirman que la animación debería estar presente cuando en los contenidos básicos es muy significativa la presencia de “cambios en el tiempo” y “características direccionales”. En el caso de conceptos, el nivel de realismo y fidelidad se debe ajustar a las necesidades y posibilidades del estudiante. En el caso de procedimientos, el nivel de realismo y fidelidad debe ser alto para que la animación sea la mejor representación posible del procedimiento real.

4. Software Educativo

El software educativo fue desarrollado utilizando el lenguaje de programación C#. Es un lenguaje de programación de propósito general orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Además, es un lenguaje diseñado para la infraestructura de lenguaje común (Common Language Infrastructure, CLI), una especificación estandarizada que describe un entorno virtual para la ejecución de aplicaciones; tales aplicaciones, aunque estén escritas en diferentes lenguajes de alto nivel, pueden ejecutarse en diversas plataformas tanto de software como de hardware sin la necesidad de volver a compilar el código fuente.

Para la construcción de gráficos y curvas se utilizó la librería ZedGraph. Se trata de una librería de clases,

control de usuario y control web escrita en C# que permite crear gráficas en proyectos .NET (Licencia: GNU Library o Lesser General Public License).

Los contenidos del software educativo se proyectan en el aula en formato gigante mediante una computadora portátil y un proyector apropiado. La interfaz gráfica es amigable e intuitiva. La mayoría de los datos que se ingresan son numéricos y están validados en lo que respecta al tipo de dato y al rango de valores que se pueden ingresar. Cada pantalla tiene un texto de identificación que la distingue de las demás para evitar confusiones. Todas las curvas se grafican en colores diferentes y tienen su leyenda correspondiente para una apropiada identificación.

5. Descripción y Justificación del Software Educativo

El software educativo está conformado por dos módulos correspondientes a áreas de aplicación de la técnica de simulación de eventos discretos: i) sistemas de espera; ii) sistemas de inventario.

El módulo sistema de espera muestra la animación de un centro de servicio en el cual los clientes llegan según una determinada distribución de probabilidad de los tiempos entre arribos y son atendidos por un servidor, también con una determinada distribución de probabilidad de los tiempos de servicio. Tanto para el proceso de arribo como para el de servicio se puede seleccionar entre una distribución de probabilidad exponencial, uniforme o normal. Una vez elegida la distribución, se deben seleccionar valores para los parámetros de la misma.

Haciendo click en el botón *Simular* comienza una animación de la llegada de clientes al centro de servicio (Figura 1). En la pantalla se observa el valor del reloj de la simulación, el status del servidor, el número de clientes en cola y el tiempo de partida del cliente que está siendo atendido. Cuando el número de clientes en cola supera un valor preestablecido se produce la negación del servicio. Pulsando el botón *Pausa*, se detiene la animación y se puede acceder a la visualización de dos gráficos: i) gráfico de la función número de clientes en cola Q(t); ii) gráfico de la función ocupación del servidor B(t) (Figura 2). Estos gráficos se utilizan para la explicación de dos medidas de rendimiento habitualmente calculadas en los sistemas de espera: i) valor esperado del número promedio de clientes en cola q(n); ii) valor esperado de la utilización del servidor u(n).

Law y Kelton [9] en su libro *Simulation Modeling and Analysis* proponen realizar un ejercicio de simulación evento a evento del sistema de espera, para así mostrar los cambios en las variables y las estructuras de datos involucradas en el enfoque orientado a los eventos. El módulo del software educativo va desarrollando cálculos equivalentes y permite así que el docente tenga en simultáneo la representación del sistema físico, las gráficas de Q(t) y B(t) y las instantáneas de los valores que adoptan las variables de estado, la lista de eventos y los contadores estadísticos durante cada evento (Figura 3).

El módulo sistema de espera incorpora un botón que permite realizar una simulación completa –sin animación– donde el parámetro a variar es la duración de la corrida. Además de indicar los valores finales de los estimadores de las medidas de rendimiento, se muestran gráficos de la evolución de tales estimadores a lo largo de la corrida. Lo anterior permite al docente explicar la conducta transiente y de estado estacionario del proceso estocástico que caracteriza a los sistemas de espera previamente descriptos. También se incluye un botón denominado *Pronóstico* que permite realizar 500 corridas independientes de los datos ingresados, con el objetivo de mostrar gráficos de frecuencia de las diferentes medidas de rendimiento como introducción al tema de Análisis de Resultados.

El módulo sistema de inventario muestra la animación de un depósito al cual llegan clientes según una determinada distribución de probabilidad de los tiempos entre arribos y demandan artículos según valores asignados a una distribución empírica. El módulo permite el ingreso de otros datos de entrada (inventario inicial, fin de la simulación, distribución y parámetros del retardo de envío) y valores para el cálculo de los costos de pedido, mantenimiento y faltantes (Figura 4). La política de reposición es estacionaria con parámetros nivel inferior de inventario (s) y nivel superior de inventario (S).

Haciendo click en el botón *Animación* comienza una representación de la llegada de clientes al depósito y la satisfacción de la demanda mientras el nivel de inventario pueda realizarla. Pulsando el botón *Pausa* se detiene la animación y se puede acceder a la visualización de gráficos que sirven para el cálculo de medidas de rendimiento tales como costo de mantenimiento, costo por faltante y costo de pedido. Se visualizan las gráficas de I(t) –nivel de inventario al tiempo t–, $I^+(t)$ –nivel real de inventario al tiempo t– e $I^-(t)$ –nivel de faltantes al tiempo t– y sus correspondientes área bajo la curva (Figura 5). El módulo permite realizar una simulación evento a evento como explicación intuitiva de la técnica. De esta manera, el docente puede tener en simultáneo la representación del sistema físico, las gráficas de I(t), $I^+(t)$ e $I^-(t)$ y las instantáneas de los valores que adoptan las variables de estado, la lista de eventos y los contadores estadísticos durante cada evento.

El módulo incluye un botón que permite realizar una corrida rápida sin animación para mostrar los valores finales de los estimadores de las medidas de rendimiento y gráficos de la evolución de tales estimadores a lo largo de la corrida. Lo anterior permite explicar la conducta transiente y de estado estacionario del proceso estocástico que caracteriza a los sistemas de inventario previamente descriptos. También se incluye un botón denominado *Pronóstico* que permite realizar 500 corridas independientes de los datos ingresados, con el objetivo de mostrar gráficos de frecuencia del costo total promedio mensual para su posterior análisis durante el desarrollo del tema Análisis de Resultados (Figura 6).

Se considera que está justificado el uso del software educativo para la enseñanza de los conceptos básicos de la simulación de eventos discretos. Las premisas de la teoría de fidelidad epistémica apuntalan la propuesta de usar el software educativo ya que el mismo permite representar el modelo mental del docente con la mayor fidelidad posible. El software educativo conforma al principio de congruencia puesto que el contenido y el formato de las imágenes se corresponden con el contenido y el formato de los conceptos que son presentados a los alumnos. Se diseñó al software para que las imágenes fueran simples y no pasaran excesivamente rápidas por la pantalla para conformar al principio de aprehensión.

El uso de animación está *a priori* justificado porque el movimiento y la trayectoria están presentes en los contenidos a enseñar. Además, su utilización se basa en la hipótesis de la codificación dual integrada, ya que mientras el docente explica verbalmente las características del sistema y los aspectos metodológicos de la técnica, la animación ayuda a establecer conexiones referenciales mediante eventos no verbales.

En ambos módulos se representa al sistema físico (fidelidad física), como así también gráficas de funciones continuas que muestran la evolución de algunas variables o medidas de rendimiento a medida que transcurren los eventos (fidelidad funcional). En una primera fase, la animación actúa como una estrategia de presentación, al darle un contexto visual a una idea compleja, difícil de presentar verbalmente. *A posteriori*, actúa como función de clarificación al proveer el entendimiento del método y de los procedimientos de cálculos sin la necesidad de agregar información textual adicional. El nivel de realismo y fidelidad se ajustó a las necesidades de los alumnos.

Un número significativo de alumnos consultados afirmó haber incorporado los contenidos introducidos de manera satisfactoria. A partir de una encuesta informal, podemos afirmar que los estudiantes comprendieron mejor los conceptos y métodos que se dictan en el curso de Simulación y se sintieron más comprometidos con la asignatura al compararlos con alumnos de cursos anteriores a la utilización del software educativo.

6. Conclusiones

Simulación es una disciplina consolidada de la Investigación Operativa y de las Ciencias de la Administración. Los planes de estudio de numerosas carreras de de ingeniería la incluyen como asignatura obligatoria. Algunos estudiantes de Ingeniería en Sistemas de Información tienen dificultades para comprender los conceptos básicos, las cuestiones metodológicas, como así también los procedimientos y métodos computacionalmente intensivos incluidos en los contenidos de un curso de Simulación.

Los métodos tradicionales de enseñanza en ingeniería se basan en gráficos y dibujos estáticos. Fundamentado en teorías vigentes en enseñanza multimedia (teoría cognitiva de enseñanza multimedial, teoría de la fidelidad epistémica), en principios para la aplicación exitosa de gráficos animados (principio de congruencia, principio de aprehensión) y en heurísticas prácticas, se diseñó un software educativo que incluye elementos de animación asociados a contenidos habitualmente enseñados en un curso de simulación de eventos discretos.

La técnica de animación facilita la enseñanza de conceptos básicos, métodos de cálculo y cuestiones metodológicas correspondientes a un curso de Simulación. El software demostró ser una herramienta apropiada para propósitos educativos a nivel universitario de grado y puede ser utilizado en cursos introductorios de Simulación en cualquier carrera de ingeniería.

7. Agradecimientos

Los siguientes alumnos, pertenecientes a la carrera de Ingeniería en Sistemas de Información de la Facultad Regional Rosario de la Universidad Tecnológica Nacional (UTN), colaboraron en la programación del software educativo descripto en este trabajo: Adrián Borella, Matías Gentile, María Cecilia Kraiselburd, Bruno Varani.

8. Referencias

[1] Rieber, L.P., Kini, A., "Theoretical foundations of instructional applications of computer-generated animated visuals", *Journal of Computer-Based Instruction*, 18(3), 1991, pp. 83-88.

[2] Schnotz, W., Lowe, R.K., "A unified view of learning from animated and static graphics", in: Lowe, R.K., Schnotz, W. (eds.) *Learning with animation: Research implications for design*, New York: Cambridge University Press, 2008, pp. 304-356.

[3] Esponda-Argüero, M., "Techniques for visualizing data structures in algorithmic animations", *Journal of Information Visualization*, 9, 2010, pp. 31-46.

[4] Höffler, T.N., Leutner, D., "Instructional animation versus static pictures: a meta-analysis", *Learning and Instruction*, 17, 2007, pp. 722-738.

[5] Tversky, B., Morrison, J.B., Betrancourt, M., "Animation: can it facilitate?", *Int. J. Human-Computer Studies*, 57, 2002, pp. 247-262.

[6] Zoabi, W., Sabag, N., Gero, A., "Using Animation to Improve the Student's Academic Achievement on Bipolar Juntion Transistor", *American Journal of Engineering Education*, Fall 2012, 3, 2, 2012.

[7] O'Donnell, F., *Simulation frameworks for the teaching and learning of distributed algorithms*, University of Dublin College, Ph. D. Thesis, 2006. http://www.tara.tcd.ie/jspui/bitstream/2262/1277/1/TCD-CS-2006-20.pdf

[8] Weiss, R.E., Knowlton, D.S., Morrison, G.R., "Principles for using animation in computer-based instruction: theoretical heuristics for effective design", *Computers in Human Behavior*, 18, 2002, pp. 465-477.

[9] Law, A.M., Kelton, W.D., *Simulation Modeling and Analysis*, Mc Graw-Hill, 2000.

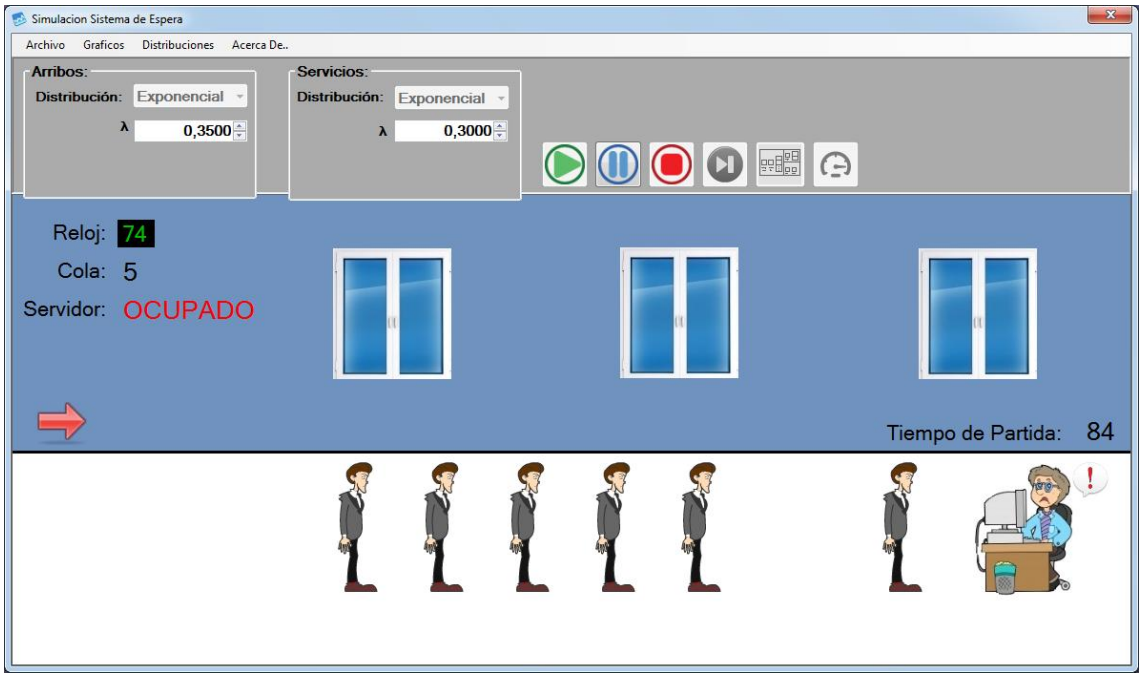


Figura 1. Animación del sistema de espera

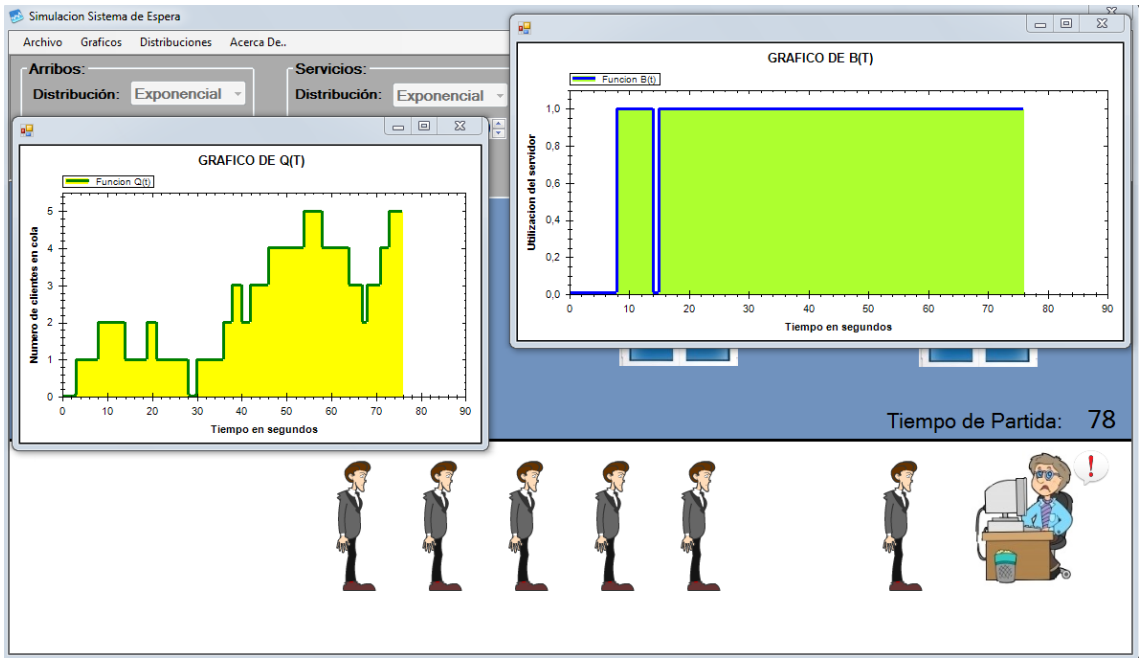


Figura 2. Gráficos de la funciones Q(t) y B(t)

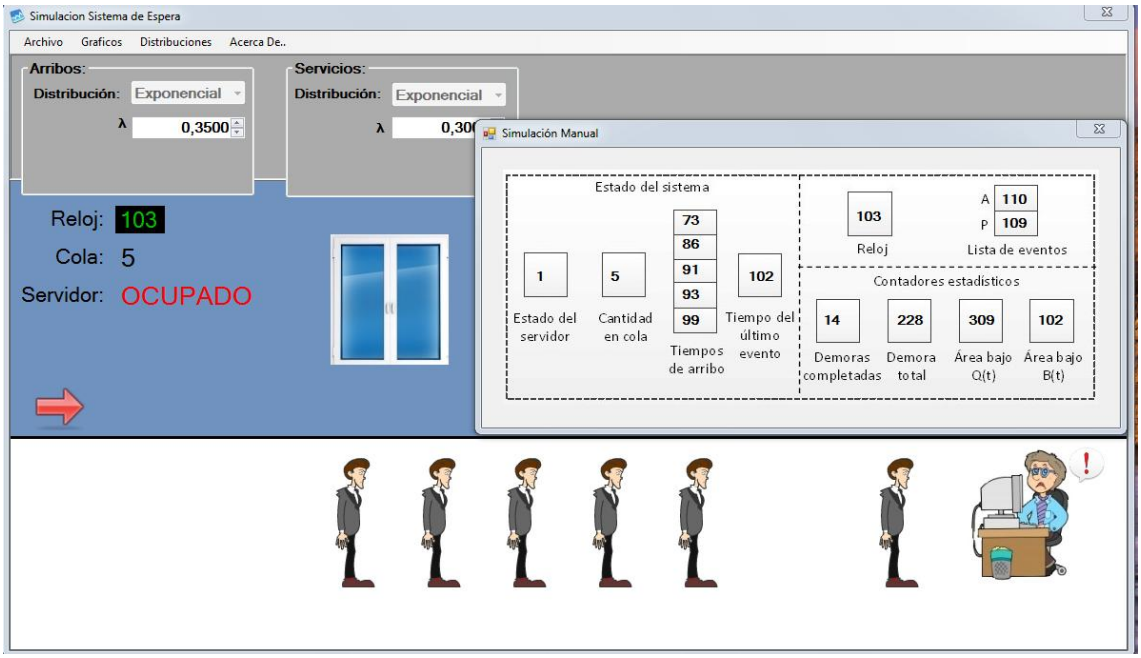


Figura 3. Instantánea de un evento del sistema de espera

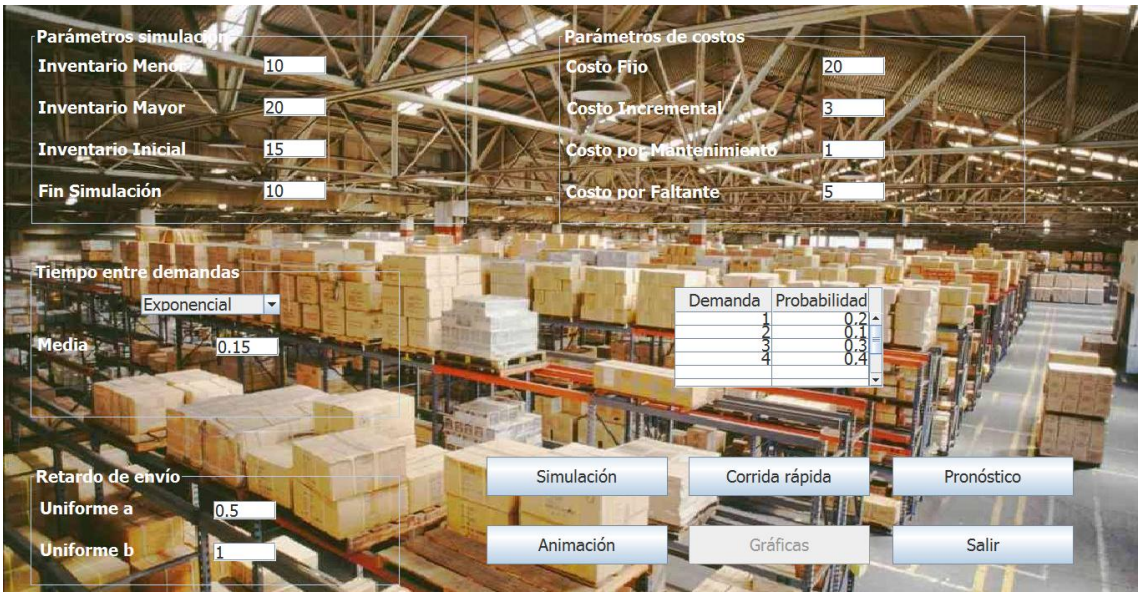


Figura 4. Pantalla de ingreso de datos al sistema de inventario

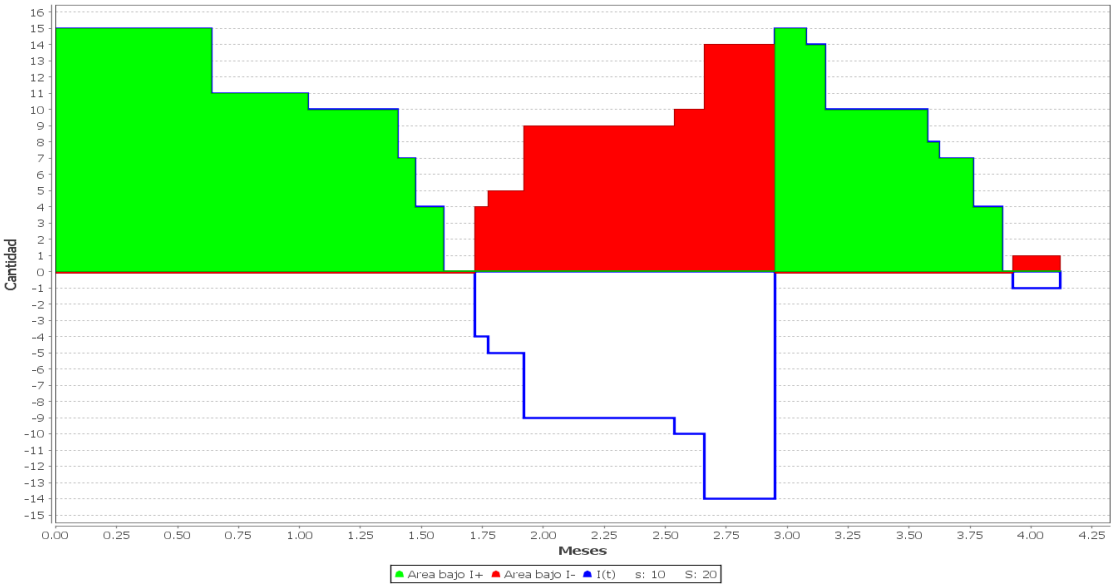


Figura 5. Gráficos de las funciones I(t), $I^{+}(t)$ e $I^{-}(t)$

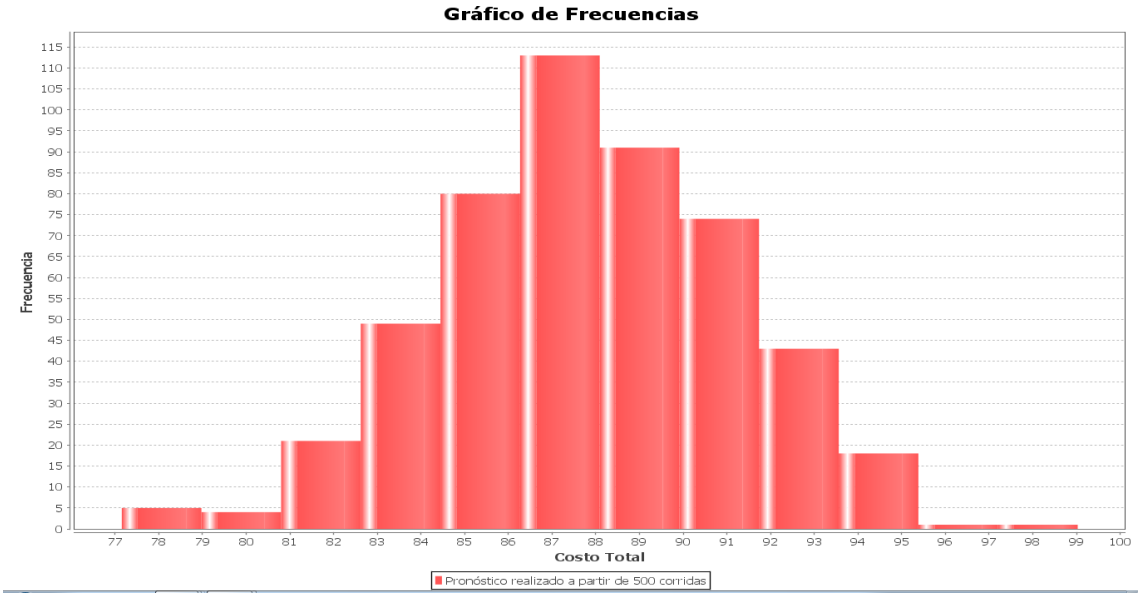


Figura 6. Gráfico de frecuencias del costo total para una simulación del sistema de inventario

La educación con TICs en un trabajo colaborativo sobre calorimetría entre docentes y alumnos de los niveles secundario y universitario

Velazque, Mirta Susana; Martínez, Horacio José

Facultad de Ciencias de la Alimentación
Universidad Nacional de Entre Ríos
Mons. Tavella 1450, Concordia
velazquem@fcal.uner.edu.ar; martinezh@fcal.uner.edu.ar

Abstract

ICT in education implies the teaching of material as well as symbolical contents. In addition to ICT based activities, it is necessary to focus on other aspects, such as innovative educational projects and the implicit digital training and competence in order to carry out these projects. University Extension projects aimed at articulating secondary and university contents are a good option to implement quality collaborative works, that can substantially contribute both to teachers and students education. Since the beginning of the 21st century, many authors have devoted themselves to the study of conceptual error detection in Heat and Temperature. They believe error detection should be a priority in the education program of every level. The objective of the present work was to design and implement a pedagogical strategy for the teaching of calorimetry by means of a collaborative work between secondary and university students and teachers. The idea was to combine traditional calorimetry teaching strategies such as laboratory experiments, and digital tools such as netbooks and mobile phones. The results of the experience were very positive. Although the sample was small, findings support the idea that this type of activities could be further exploited and that more schools could be incorporated.

Keywords: calorimetry, ICT based activities, collaborative work.

Introducción

La educación con TICs presupone la enseñanza de aspectos tanto materiales como simbólicos. Además del acceso a la tecnología a través del uso de computadoras o terminales y de la Web es necesario contemplar otros aspectos como proyectos educativos innovadores y la capacitación implícita para llevar a cabo dichos planes. Los Proyectos de Extensión Universitaria enfocados a la articulación entre diferentes niveles educativos, tales como el nivel secundario y el universitario se presentan como una buena opción para implementar trabajos colaborativos con contenidos de calidad que enriquezcan tanto a los alumnos como a los docentes que participan en ello. El término colaborativo se emplea aquí por el hecho de combinar actividades tradicionales con el empleo de TICs, ya que en este último contexto el trabajo compartido es una condición primordial, pero no ha sido así históricamente en el diseño y puesta a punto de actividades experimentales. En este sentido este tipo de experiencias son novedosas en el ámbito universitario.

El estudio de la calorimetría se aborda en los Programas de Física Clásica en la escuela secundaria y también en la rama específica de la Física denominada Termodinámica. Esta

última aparece en forma específica, generalmente en los cursos universitarios. El hecho de que los conceptos fundamentales de calor y temperatura, que son objeto del estudio de la termodinámica se hayan estudiado con muchos otros conceptos de la Física hace que los estudiantes manifiesten problemas al tener que utilizarlos en la Universidad, teniendo en cuenta que además se enseñan en los ciclos medios de las carreras en los niveles superiores, quedando muy lejos de su aprehensión inicial realizada en la escuela secundaria. Además, cabe agregar que calor y temperatura en la vida cotidiana tienen significados muy diferentes a los definidos por la Física, lo cual produce confusion y conflicto. Muchos autores desde los comienzos del Siglo XXI hasta ahora se han dedicado al estudio de la detección de los errores conceptuales en Calor y Temperatura (Cotignola y col., 2002[1]; Flores y col., 2003 [2]; González, A., 2003[3], Furió y col., 2005, 2007[4][5]; Rodríguez y col., 2006[6]; Chávez y col., 2007[7]; Doménech y col. 2010[8]; Barragán Gómez y col., 2010[9]; Mahmud y col., 2010[10]), quienes además opinan que el caso de los conceptos de calor y temperatura han de ocupar un lugar prioritario en el quehacer educativo de todos los niveles, continuando estudios anteriores, relacionados con el tema sobre todo de la década de los 90´s.

De acuerdo con Cárdenas y col. 1197[11], es necesario destacar y valorar más las actividades que conducen a la construcción del concepto de equilibrio térmico como punto de partida para logar la diferenciación de los conceptos de calor y temperatura. Según estos mismos autores, el Principio Cero de la Termodinámica es considerado un enunciado trivial sin que se capte su relevancia espistemológica.

También García García y col., 2013[12], proponen un análisis de 9 conceptos de Termodinámica, entre los cuales se encuentran el calor, la temperatura y el equilibrio térmico, a través de problemas abiertos de predicción.

Durán Aponte y col., 2013[13], y Durán-García y col., 2013[14], señalan que entre las metodologías innovadoras para la enseñanza de la Termodinámica, las metodologías de aprendizaje cooperativo favorecen no sólo la adquisición de contenidos sino también el desarrollo de habilidades sociales necesarias en la formación de todo profesional.

Delgado y col., 2008[15], respecto del uso de Softwares educativos para el aprendizaje de la Termodinámica que las tecnologías informáticas han impactado en la modalidad educativa modificando la educación presencial, este fenómeno conocido como la virtualización de la enseñanza introduce una situación nueva: la tradicional distinción entre educación presencial y a distancia está absoleta, y ambas modalidades educativas se tienden a acercar en torno a dos hechos: virtualización de textos y otros materials de enseñanza y búsqueda de la autonomía del estudiante. Estas autoras recomiendan, entre otras cosas: emplear más herramientas de este tipo en la enseñanza de la Termodinámica, dotar a las instalaciones educativas de herramientas de tecnologías de información y comunicación (TICs) y fomentar en el estudiante el proceso de investigación bibliográfica especializada.

De acuerdo con Barragán-Aroche y col., 2004[16], es posible orientar el desarrollo de los cursos propios de termodinámica hacia el planteamiento y análisis de los diferentes problemas que esta ciencia aborda, proveyéndole al alumno de un conjunto de elementos de cómputo, tan simples como pueden ser las hojas de cálculo. Ramírez-Correa y col., 2010[17], también indican que el uso de la tecnología de la información queda determinada por la utilidad percibida y la facilidad de uso percibida. En el caso del Proyecto que nos ocupa los alumnos voluntarios universitarios ya han tenido oportunidad de percibir la utilidad del uso de Páginas Web en la búsqueda de información y tienen una rutina de uso, dado que además de

sus computadoras personales en la Facultad de Ciencias de la Alimentación existe una sala de computación, al lado de la biblioteca, de uso exclusivo para los alumnos, lo cual les permite comparar los contenidos de las asignaturas que están cursando entre la información impresa en libros y revistas especializados y la de Internet. En cambio los alumnos secundarios recién han recibido sus notebooks y todavía no tienen rutina de uso y tampoco aprecian la utilidad de las TICs en la adquisición de conocimientos de las ciencias en general y de la termodinámica en particular.

Cabe agregar que otro tema que inquieta a la comunidad educativa de todos los niveles es la irrupción del teléfono móvil en la vida cotidiana de los estudiantes y por lo tanto se está planteando la posibilidad de utilizarlo como un elemento que colabora en el proceso de enseñanza y aprendizaje y no como un distractor (Moreno Guerrero, 2011[18]; Brener, 2011[19], Tirado Medina, 2012[20]).

El objetivo del presente trabajo fue diseñar e implementar una estrategia pedagógica para la enseñanza de calorimetría a través de un trabajo colaborativo con alumnos y docentes de los niveles secundarios y universitario que combine estrategias tradicionales de enseñanza y herramientas informáticas.

Metodología

El Proyecto de Extensión Universitaria contó con una becaria y un grupo de tres alumnos colaboradores que cursaban las carreras de Ingeniería en Alimentos. La Escuela Secundaria que participó de la experiencia fue la Nº 31 Benito Juárez, de una zona semi-rural de Concordia, Paraje Estación Yuquerí, Provincia de Entre Ríos, cuya modalidad adoptada es la de Bachiller con orientación en Ciencias Naturales, que recientemente recibió netbooks a través del Programa Nacional de la República Argentina "Conectar igualdad". Participaron dos docentes universitarios y un docente de escuela secundaria.

Se comenzó a trabajar en forma grupal entre los directores del proyecto, la alumna becaria y los alumnos voluntarios universitarios, aunando criterios para las diferentes tareas y estableciendo un cronograma que no afectara el cursado normal y los turnos de exámenes de los alumnos.

La primera tarea consistió en mejorar y ensayar el Trabajo Práctico "Introducción a calor y temperatura", de acuerdo a experiencias que figuran en el Manual de Entrenamiento del Cuarto Taller Regional del Cono Sur sobre Aprendizaje Activo de la Física, publicado en 2011 por la Universidad Nacional de San Luis y el Ministerio de Educación de la Nación Argentina. Para diseñar el trabajo práctico los alumnos voluntarios efectuaron un repaso de los conocimientos teóricos sobre calor y temperatura de la bibliografía utilizada en las asignaturas Química General y Termodinámica, biblioteca de la Facultad de Ciencias de la Alimentación y a través de Internet. Con este material prepararon una introducción en Power-Point para presentar a los alumnos de secundaria. Los directores del proyecto, de acuerdo con el cronograma establecido realizaron las reuniones pertinentes para aclarar las dudas tanto teóricas como prácticas que surgían de las actividades llevadas a cabo por los alumnos voluntarios universitarios. Se diseñaron tres actividades experimentales para trabajar el concepto de equilibrio térmico y la Ley Cero de la Termodinámica, de manera de combinar enseñanza tradicional con el análisis de datos asistido por computadora.

La segunda tarea llevada a cabo por los alumnos mencionados al final del párrafo precedente consistió en poner en práctica las experiencias frente a los alumnos de la escuela secundaria acompañados por los directores del proyecto y los docentes. Las dinámicas adoptadas mostraron dominio, tanto teórico como práctico de las actividades, dividiéndose la presentación en Power-Point, a fin de que todos los alumnos voluntarios enriquecieran la

explicación teórica. Luego dividieron los cursos de la escuela participante en tres comisiones. Cada comisión era atendida por la becaria o un alumno voluntario, pero periódicamente se consultaban sobre la marcha de la actividad.

La tercera tarea consistió en llevar a cabo las actividades que se reseñan seguidamente utilizando el Programa Excel.

Finalmente se realizó un Plenario con participación de todos los alumnos y docentes para aclarar las dudas, completar las preguntas de la guía y realizar una encuesta.

En la Tabla 1 se muestra un resumen de las actividades reseñadas.

Tabla 1. Resumen de las actividades realizadas

Grupo	Actividad	TICs
A	Introducción sobre calorimetría para presentar a los alumnos de secundaria	•Trabajo Práctico tradicional •Internet •Power-Point
B	Confección del gráfico temperatura-tiempo	•Trabajo Práctico tradicional •Excel •Uso teléfono móvil
C	Conclusiones	•Power-Point •Excel
D	Plenario	Análisis de resultados

Grupo: A = alumnos voluntarios universitarios y becaria; B = alumnos secundarios (a su vez divididos en B-1, B-2 y B-3); C = alumnos voluntarios universitarios y becaria + alumnos secundarios; D = alumnos voluntarios universitarios + becaria + docentes.

Cabe agregar que las conclusiones de esta actividad fueron presentadas por la becaria y los alumnos voluntarios, mediante exposición oral en Power-Point, en las *II Jornadas Puertas abiertas a la Educación,* realizadas los días 07 y 08 de Noviembre de 2013 en la Escuela Benito Juárez, actividad declarada de interés educativo por la Dirección Departamental de Escuelas de Concordia, según Disposición Nº 127 D.D.E. del 10/10/13.

Con la misma modalidad señalada en el párrafo anterior se presentaron las conclusiones en el Taller "*La Enseñanza de conceptos termodinámicos a través de la Física",* evento realizado a través del Programa "Museo Interativo de Ciencias", organizados por el Área de Desarrollo de la Comisión Administradora para el Fondo Especial de Salto Grande (CAFESG), la Subsecretaría de Ciencia y Tecnología de la Provincia de Entre Ríos, la Universidad Nacional de Entre Ríos, la Dirección Departamental de Escuelas y con el auspicio del Ministerio de Ciencia y Tecnología de la Nación. El auditorio estuvo representado por seis escuelas secundarias con un total de ciento treinta y nueve alumnos secundarios acompañados por doce docentes.

A continuación se describen las Guías de las actividades desarrolladas.

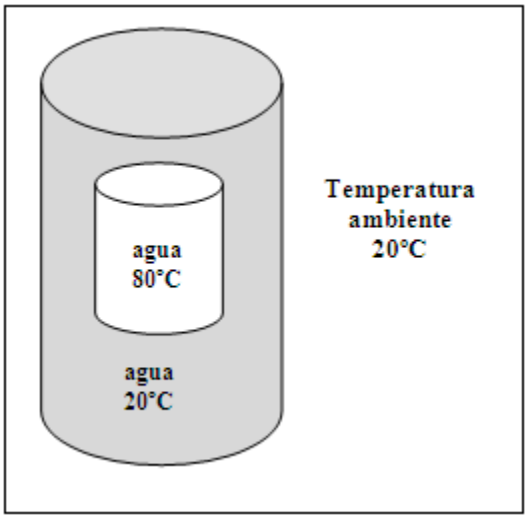
Actividad 1. Cada grupo utilice un termómetro de mercurio y mida la temperatura en distintos puntos del laboratorio una vez por minuto durante 10 minutos y llene el siguiente cuadro:

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

Hora de comienzo de la actividad:……..
Hora de finalización de la actividad:……..
¿Qué conclusiones se pueden hacer?.....
¿Hay una Ley de la Termodinámica que explique este fenómeno?.....

Confeccione un gráfico temperatura-tiempo utilizando el Programa Excel. Utilice el celular para medir el tiempo y para tomar fotografías o generar un video de la experiencia.

Actividad 2. *Predicción.* Supongan que se coloca una cantidad de agua a temperatura ambiente, unos 20ºC en un recipiente grande y una cantidad pequeña de agua caliente, supongamos 80ºC, en un recipiente menor y no aislado. El recipiente pequeño se sumerge en el recipiente grande y se deja allí por un tiempo largo.



a. ¿Qué temperatura final tendrá el agua del recipiente pequeño? Podemos decir que:

- La temperatura será igual a la suma de las dos temperaturas dividido por dos.
- La temperatura estará más cerca de 80ºC.
- La temperatura estará más cerca de 20ºC.

b. ¿Qué temperatura final tendrá el agua del recipiente grande? Podemos decir que:

- La temperatura será igual a la suma de las dos temperaturas dividido por dos.
- La temperatura estará más cerca de 80ºC.
- La temperatura estará más cerca de 20ºC.

Comprobación. Supongan que se coloca una cantidad de agua a temperatura ambiente, la que será registrada en un termómetro en un recipiente grande, como por ejemplo un vaso de precipitado.

En un tubo de ensayo se coloca una cantidad pequeña de agua caliente, cuya temperatura también será registrada.

Posteriormente, el tubo de ensayo, conteniendo el termómetro, se sumerge en el recipiente grande, el cual también contiene un termómetro. A partir de dicho momento se procederá a registrar en intervalos regulares las lecturas de ambos termómetros hasta llegar al equilibrio térmico.

Hora de comienzo de la actividad:……..

Hora de finalización de la actividad:……...

Termómetro 1

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

Termómetro 2

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

Preguntas

2.1. Compare las temperaturas finales del agua del vaso de precipitado y del tubo de ensayo con su predicción de la Actividad 3……

2.2. ¿A medida qué el agua caliente se enfría, qué sucede con el agua que estaba a temperatura ambiente?.....

2.3. ¿Se pueden explicar los cambios de temperatura por un intercambio de agua entre los recipientes? Si la respuesta es no, ¿cuál es la explicación?.....

2.4. ¿Cuándo se ha alcanzado el equilibrio térmico?.....

Actividad 3. Tomar tres vasos de precipitado de 500 ml. Colocar 150 ml, 250 ml y 500 ml de agua en cada uno de ellos. Encender el mechero y exponer durante 20 minutos cada uno de los recipientes, registrando a lapsos de tiempo constantes la temperatura de las masas de agua mediante un termómetro de mercurio. Registrar dichos valores en las siguientes tablas. Mida la presión del laboratorio.

Cantidad de agua: 150ml

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

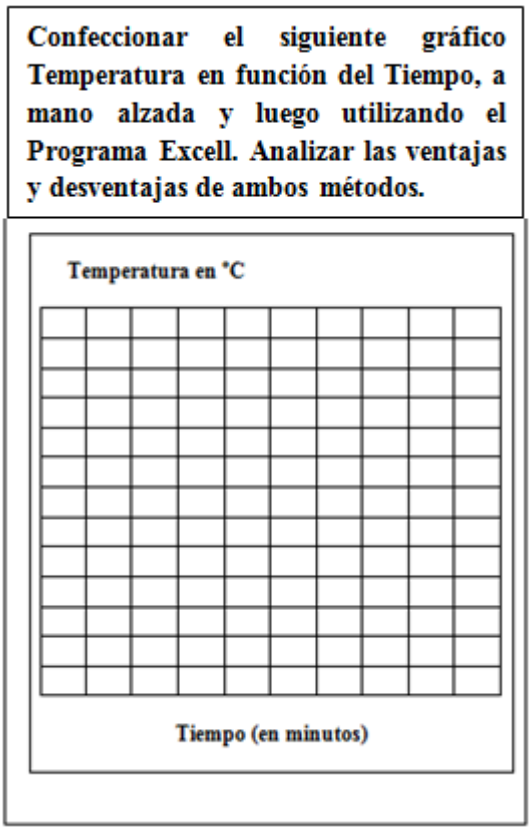
Cantidad de agua: 250ml

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

Cantidad de agua: 500ml

Nº medición	Tiempo	Temperatura	Nº medición	Tiempo	Temperatura
1			6		
2			7		
3			8		
4			9		
5			10		

Una vez finalizada la actividad 3 confeccionar el gráfico Temperatura-Tiempo, a mano alzada y utilizando el Programa Excel. Utilice el celular para medir el tiempo y para tomar fotografías o generar un video de la experiencia.



Resultados

Primero se realizará un análisis de los resultados en cuanto a los contenidos específicos de calorimetría.

Actividad 1. Los alumnos de los tres grupos: B-1, B-2 y B-3 observaron una diferencia en ±1ºC en un rango de 32-33ºC (Fig.1). El grupo B-1 dijo que la explicación era porque "la ley de equilibrio térmico establece que los cuerpos siempre tienden a tener la misma temperatura". El grupo B-2 dijo "la explicación la da la Ley Cero de la Termodinámica" (pero no la enunciaron) y el grupo B-3 reportó "hay

variación de temperaturas cerca de las ventanas por la energía solar".

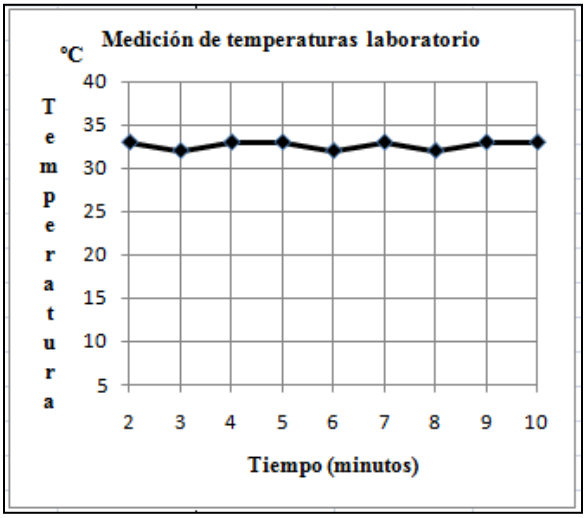


Figura 1. Gráfico temperatura-tiempo de las condiciones térmicas del laboratorio

Actividad 2. El grupo B-1 y B-3 contestaron en la predicción que la temperatura final del recipiente grande y del recipiente pequeño debían estar más cerca de 20ºC porque los dos sistemas son abiertos y deben equilibrarse con la temperatura del medio ambiente. En cambio el grupo B-2 predijo que la temperatura final de los dos recipientes sería igual a la suma de las dos temperaturas dividido por dos. Una vez realizada la experiencia y confeccionado el gráfico temperatura-tiempo pudieron observar fácilmente que las dos temperaturas tienden a igualarse con la del medio ambiente (Fig.2).

Actividad 3. Los Grupos B-1 y B-2 llegaron a las mismas conclusiones, partieron de una temperatura de 25ºC y llegaron a 100ºC a los 8 min calentando 150 ml de agua, a los 14 min con 250 ml de agua y a los 20 min con 500 ml de agua (Fig. 3 y 4). El grupo B-3, en cambio tuvo resultados diferentes porque partieron de 26ºC y la temperatura final de ebullición para las distintas cantidades de agua les dio 101ºC. La presión resultó de 1 atm según un barómetro de mesada Luft Germany, calibrado en la estación metereológica de la ciudad de Concordia.

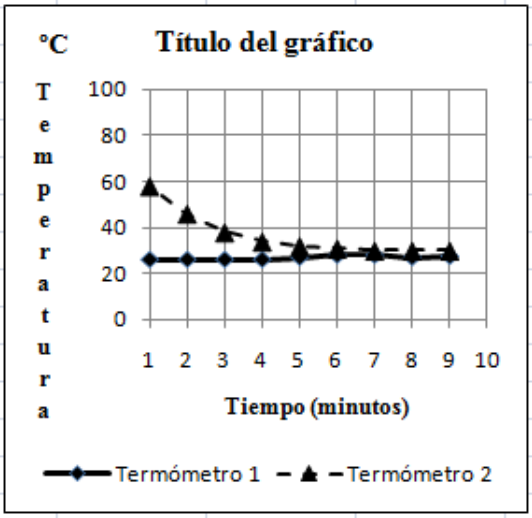


Figura 2. Resultados de la experiencia de la actividad 2.

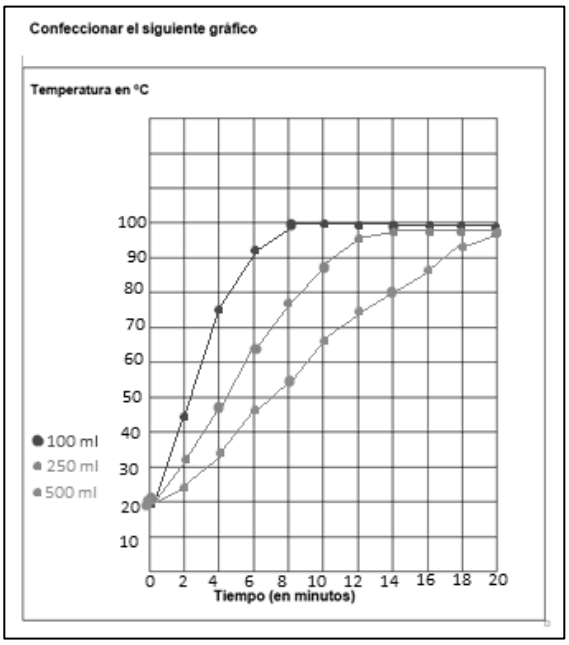


Figura 3. Gráfico Temperatura-Tiempo confeccionado a mano alzada.

Durante el Plenario de cierre se observó que si bien la temperatura de equilibrio final es de 100ºC, las masas tienen distintas velocidades, dadas por las pendientes de las curvas, por lo

que las masas de agua más pequeñas llegan más rápido a la temperatura de 100ºC que las masas más grandes.

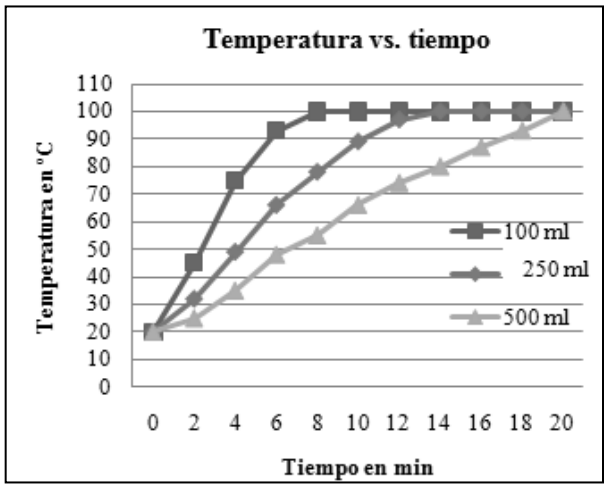


Figura 4. Gráfico Temperatura-Tiempo confeccionado en Excel.

En lo que respecta al trabajo colaborativo los alumnos voluntarios universitarios presentaron un Informe de Actividades en el que destacaron la importancia de participar de esta experiencia, ya que consolidaron conocimientos específicos adquiridos en la asignatura de Termodinámica durante el cursado de la Carrera de Ingeniería en Alimentos, pero además fortalecieron competencias como la oralidad, organización de grupos, gestión de recursos, solidaridad. En cuanto al uso de las TICs concordaron que es conveniente que los alumnos secundarios se habitúen a su uso, ya que es una herramienta indispensable en la Universidad. Los alumnos secundarios, por su parte, contestaron una encuesta en la que se les preguntó si consideraban positiva la experiencia, a lo cual el total de los participantes contestó que sí y que había sido más agradable que las actividades fueran conducidas por alumnos y no por docentes porque tenían más confianza a la hora de plantear las dudas.

Discusión

En la actividad 1 la respuesta esperada era que pudieran asociar el laboratorio con el concepto de sistema termodinámica en estudio ya que previamente se habían explicado los conceptos de sistema, entorno y frontera en el Power-Point preparado por los alumnos voluntarios universitarios y la becaria. También se había presentado el enunciado de la Ley Cero de la Termodinámica "establece que dos sistemas a la misma temperatura que un tercer sistema tienen la misma temperatura entre sí" (Resnick y col., 2000[21]; Morán y col., 2005[22], Çengel y col., 2007[23]). La respuesta de que los cuerpos siempre tienden a tener la misma temperatura mezcla el proceso de transferencia de calor con el de temperatura. En cuanto a la respuesta de que hay variación cerca de las ventanas por la energía solar implica una aplicación intuitiva más compleja del concepto de frontera, ya que efectivamente la relación entre un sistema y su entorno se da a través de una superficie de separación que puede tener diferentes puntos de interacción. En el plenario de cierre se vuelve a discutir el enunciado de la Ley Cero de la Termodinámica.

En cuanto al uso de las TICs se realizó una encuesta a los alumnos secundarios sobre si los gráficos en Excel ayudaban a la comprensión del tema. Las respuestas fueron que sí, efectivamente comparando los gráficos de las distintas experiencias se observa claramente en la primera que la temperatura del laboratorio tiende a una línea recta horizontal, indicando que la temperatura ambiente es más o menos homogénea. En la segunda experiencia se observa que las temperaturas de las dos cantidades de agua que tienen temperaturas diferentes al comienzo de la experiencia tienden a acercarse a la temperatura ambiente por tratarse de sistemas abiertos. En cuanto a la tercera experiencia se aprecian las distintas velocidades para llegar a la temperatura de equilibrio que es la de 100ºC, debido a las diferencias en las masas que se someten al

proceso de calentamiento. En esta última experiencia se solicitó también la realización de un gráfico a mano porque se considera que el trazado manual es una competencia que se debe continuar enseñando. La comparación del trazado mediante un Software y a en forma manual indica que el primero tiene muchas ventajas sobre el segundo: es más rápido, permite cambiar tamaño, color, forma e imprimir las diversas versiones.

En cuanto a las dificultades que se presentaron, la más importante fue que los alumnos voluntarios universitarios tuvieron que explicar previamente el uso del Excel, ya que los alumnos secundarios manifestaron desconocer su aplicación. Esto determinó una readecuación del tiempo asignado a las actividades para cumplir con las prácticas de laboratorio y la discusión final, a fin de respetar los tiempos pactados con la escuela secundaria, que incluye su vuelta a la escuela a un cierto horario. Este inconveniente se subsanó sin mayores problemas, dada la solvencia en el uso del Programa Excel que poseen los alumnos universitarios.

En cuanto al uso de los celulares la mayoría responde que lo utilizan como reloj, reemplazando al tradicional reloj de muñeca. También reconocen que la toma de fotografías con los teléfonos móviles es una actividad que realizan a menudo, y que les permite difundir sus vivencias diarias a través de Facebook y de la Página Web de la escuela, donde repitieron los comentarios expresados en la encuesta realizada al final de la experiencia. Si bien estas son actividades básicas, en futuras prácticas se espera que estos aportes, una vez explicitados, puedan enriquecerse y potenciarse mediante el uso de nuevas aplicaciones.

Conclusión

Se logró una buena complementariedad entre actividades tradicionales en la enseñanza de la calorimetría como son las actividades experimentales de laboratorio y problemas abiertos de predicción con la utilización de TICs como uso de páginas Web, computadoras y cañón de proyección, netbooks y teléfonos móviles.

Si bien la muestra de alumnos secundarios y universitarios es pequeña, los resultados obtenidos permitirían optimizar las actividades, a fin de continuar con esta línea de trabajo sumando mayor cantidad de escuelas. Además la participación en eventos científicos locales permitió transferir la experiencia a más escuelas secundarias, a través de la exposición de los alumnos voluntarios universitarios, recalcando la utilidad de uso percibida de las TICs.

Referencias

[1] Cotignola, M., Bordogna, C., Punte, G. y Cappannini, O., "Difficulties in Learning Thermodynamic Concepts Are They Linked to the Historical Development of this Field?", *Science & Education*, 11(3), 2002, pp. 279-291.

[2] Flores, S., Trejo, A. y Trejo, L., "¿Cómo Mejorar el Proceso Enseñanza-Aprendizaje Mediante la Evaluación-Regulación? El Caso de la Termodinámica", *Memorias de las Terceras Jornadas Internacionales de la Enseñanza Universitaria de la Química*, Argentina, 1-8 de octubre 2003.

[3] González, A., "Calor y trabajo en la enseñanza de la termodinámica", *Revista Cubana de Física*, 20, 2013, pp. 129-134.

[4] Furió, C., Solbes, J. y Furió, C., "¿Cómo se presentan los conceptos y modelos en la enseñanza de la termoquímica? Visiones distorsionadas de la ciencia en libros de texto", *Enseñanza de las Ciencias*, número extra. VII Congreso 2005, pp. 1-6.

[5] Furió, C., Solbes, J. y Furió, C., "La historia del primer principio de termodinámica y sus implicaciones didácticas", *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 4(3), 2007, pp. 461-475.

[6] Rodríguez, V.; Díaz Gigson, S., "Concepciones alternativas sobre los conceptos de energía, calor y temperatura de los docentes en formación del Instituto Pedagógico en Santiago, Panamá", *Actualidades Investigativas en Educación*, Sep. 2012, 12, 3, pp.1-27.

[7] Chávez, E. A.; Malaver, M., Los conceptos de calor, trabajo, energía y teorema de Carnot en textos universitarios de Termodinámica, *EDUCERE*, ISSN: 1316-4910, Nº 38, Jul.-Ag.-Sep. 2007, pp.477 - 487.

[8] Doménech, J. L.; Martínez-Torregosa, J., ¿Disponen los estudiantes secundarios de una comprensión adecuada de los conceptos de calor y trabajo y su relación con la energía?, *Revista Brasileira de Ensino de Física*, 32, 1, 2010, pp.1308-1, 1308-10.

[9] Barragán Gómez, A.L.; Hernández, A.S., "Detección y Clasificación de Errores Conceptuales en Calor y Temperatura", *Latin-American Journal Of Physics Education*, 25, 1, May 2010, pp.399-407.

[10] Mahmud, M. C.; Gutiérrez, O. A., "Estrategia de Enseñanza Basada en el Cambio Conceptual para la Transformación de Ideas Previas en el Aprendizaje de las Ciencias", *Formación Universitaria*, 2010, 3, 1, pp. 11-20.
[11] Cárdenas, M.; Ragout de Lozano, S., "Análisis de una experiencia didáctica realizada para construir conceptos fundamentales de termodinámica", Cad.Cat.Ens.Fis., 14, 2, ago.1997, p.170-178.
[12] García García, J. J.; Rodríguez, E. R., "Resolver problemas: una estrategia para el aprendizaje de la termodinámica", *Revista Científica Guillermo de Ockham*, Jul.-Dic. 2013. 11, 2- ISSN: 1794-192X, pp. 117-134.
[13] Durán-Aponte, E.; Durán-García, M.; "Aprendizaje cooperativa en la Enseñanza de Termodinámica: Estilos de Aprendizaje y Atribuciones Causales, *Revista Estilos de Aprendizaje*, Abril 2013, 6, pp.256-275.
[14] Durán-García, M. E.; Durán-Aponte, E. E., "La Termodinámica en los estudiantes de Tecnología: una experiencia de Aprendizaje Cooperativo, *Enseñanza de las Ciencias*, Nº 31, 2013, pp. 45-59.
[15] Delgado, R. I.; Cambil, M. M., "Uso de Software Educativo para el Aprendizaje de la Termodinámica", *Memorias de las IV Jornadas de Investigación e Innovación Educativa*, 30-31 de Octubre de 2008, organizadas por la Universidad Centroccidental Lisandro Alvarado, Venezuela.
[16] Barragán-Aroche, J. F.; Bazúa-Rueda, E., "Herramientas para la enseñanza de la termodinámica en ingeniería química", *Tecnol. Ciencia Ed. (IMIQ)*, 2010, 19, 2, pp.83-91.
[17] Ramírez-Correa, P.; Rondán-Cataluña, F.; Arenas-Gaitán, J., "Influencia del Género en la Percepción y Adopción de e-Learning, *Journal of Technology Management Innovation*, 2010, 5, 3, pp.129-141.
[18] Moreno Guerrero, A. A., "Móvil-learning", *Observatorio Tecnológico, Gobierno de España, Ministerio de Educación, Cultura y Deporte*, 2011, disponible en www.recursostic.educacion.es/observatorio/web/es/cajon-de-sastre/38-cajon-de-sastre/1016-movil-learning.
[19] Brener, G., "El celular en la escuela", *ALAI, América Latina en Movimiento*, 2011, disponible en: http://alainet.org/active/47667&lang=es.
[20] Tirado Medina, D. M., "El celular en el aprendizaje pedagógico en el tercer año del centro de estudios de bachillerato", Proyecto de Investigación, Innovación y Creatividad de la Licenciatura en Ciencias de la Educaci´n de la Universidad Autónoma de Sinaloa, 2012, disponible en: http://es.slideshare.net/MarilyTirado/trabajo-francisco11.

Análisis de los modelos de datos de Moodle y LAMS

Brachetta, M., Monetti, J., Cortez, A., Navas, C., León, O.
UTN, Facultad Regional Mendoza
mariana.brachetta;jmonetti;oleon@frm.utn.edu.ar

Resumen

El artículo describe los resultados obtenidos en una etapa de un proyecto de investigación y desarrollo, cuya meta es obtener información que ayude a comprender el proceso de aprendizaje realizado por los alumnos. Para esto, se realiza la transposición de un contenido disciplinar de álgebra, en términos de un modelo didáctico representado mediante una red semántica. Esta última será implementada en la plataforma Moodle con la herramienta LAMS, tomando como parámetros adaptativos, el rendimiento de los alumnos y sus estilos de aprendizaje. El desarrollo requiere extraer los datos acerca de las actividades realizadas por los alumnos en el entorno virtual, los cuales serán objeto de un análisis posterior. Para cumplir con este propósito, se necesita comprender el modelo de datos en el que se basan las herramientas virtuales aplicadas. En el presente trabajo, se describen los principales aspectos del modelo de datos que subyace en las entornos utilizados y se comentan algunas conclusiones de realizado.

Introducción

Lo presentado en el artículo, es parte de los resultados alcanzados en el marco de un proyecto de desarrollo tecnológico e investigación experimental, en el cual se realizó la transposición de un contenido disciplinar de Álgebra y Geometría Analítica, en términos de la teoría didáctica para la enseñanza de la matemática denominada APOE (Acciones, Procesos, Objetos y Esquemas). A partir del modelo obtenido [7], se planteó el esquema de una red semántica adaptada en base al estilo de aprendizaje y el rendimiento de los estudiantes, la cual será implementada con Moodle y LAMS.

En el trabajo se describe el modelo de datos que soporta el funcionamiento de la herramienta LAMS Lesson integrada en la plataforma Moodle. No se realiza un abordaje exhaustivo de todas las bases de datos (BD), tablas, campos de datos y relaciones que conforman el modelo, sino del conjunto que resulta significativo en el marco del proyecto. En particular, se pretende explicar cómo se integran ambas herramientas, cómo se persisten los diseños de aprendizaje (LD), las secuencias de actividades (lecciones); y cómo se gestionan los usuarios, grupos y agrupamientos, de manera integrada entre Moodle y LAMS.

El desarrollo del artículo se divide en cinco secciones. La primera aborda consideraciones generales sobre el modelo de datos de Moodle (su arquitectura, tecnología utilizada, estándares a los que se ajusta, etc.). La segunda hace lo propio sobre el modelo de datos de LAMS. La tercera sección detalla aspectos vinculados a la integración de LAMS en Moodle desde el punto de vista de sus modelos de datos, mostrando cómo se registra en la BD de Moodle la creación de una lección en LAMS. La cuarta sección analiza el modelo de persistencia de LD y lecciones en LAMS. Por último se describe el conjunto de tablas y relaciones que gestionan de manera integrada entre Moodle y LAMS los usuarios, grupos, agrupamientos y organizaciones.

Método de trabajo

La tarea de análisis se realizó sobre un servidor Ubuntu 12.04, con Moodle v2.5.3, Apache 2.2.22, extensiones PHP y LAMS v2.4., corriendo sobre JBoss 5.1. Se utiliza LAMS incrustado en Moodle mediante el correspondiente módulo de integración. El motor de base de datos fueMySQL 5.5.34.

Para estudiar el modelo se construyó un conjunto de secuencias de aprendizaje en LAMS, las que fueron creadas y trabajadas como actividades de cursos Moodle. Luego se analizaron los datos persistentes en el modelo mediante el uso de phpMyAdmin 3.4.10 para la observación individual de estructuras de tablas, campos de datos, valores y relaciones.

Complementariamente se realizó la ingeniería inversa sobre las bases de datos de LAMS y Moodle, utilizando MySQLWorkbench 6.0 para extraer los diagramas de Entidad-Relación y estudiar características generales del modelo de persistencia [8].

Acerca del modelo de datos Moodle

El modelo de base de datos de Moodlecontiene aproximadamente 200 (doscientas) tablas, de las cuales al núcleo(core) central pertenecen unas 50 (cincuenta). Cada módulo tiene un grupo de tablas independientes,

vinculadas entre sí y al núcleo, pero no en general con otros módulos.

La estructura de BD se define, gestiona y actualiza usando XMLDB [4].

El núcleo de Moodle usa ADODB como driver para interactuar con distintas BD. Basándose en esto, se desarrolló esta capa de persistencia con el objetivo de que Moodle pudiera interactuar con más RDBMs, y no sólo con MySQL y PosgreSQL. Así los desarrolladores pueden crear y gestionar las estructuras de tablas para sus módulos de manera independiente de un motor específico.

Básicamente la arquitectura del paquete XMLDB incluye: una capa para la creación y actualización de BD (DDL) y una capa de administración (altas, bajas, modificaciones y consultas sobre la base de datos y querys) (DML). Esto se materializa en dos librerías:

1. DDL (ddllib.php): incluye todas las funciones necesarias para manejar objetos de BD, en un lenguaje con alto nivel de abstracción.Son descripciones bien formadas en XML que permiten a los desarrolladores crear, modificar y borrar tablas, campos, índices, restricciones y otros objetos de BD con independencia del RDBMS sobre el que se va a trabajar en una implementación puntual.
2. DML (dmllib.php): incluye un conjunto de funciones para realizar altas, bajas, modificaciones y consultas sobre el contenido de la BD.

Ambas librerías ddllib.php y dmllib.php siguen usando ADODB para conectarse con un motor de BD específico.Además, los archivos lib.php incluidos en cada módulo (Por ejemplo: course/lib.php, user/lib.php, etc.), incluyen funciones específicas para administrar el conjunto de tablas de ese módulo.

Acerca del modelo de datos de LAMSLesson

LAMS usa hibernate como motor de persistencia contenido en JBoss. Los scripts de administración y gestión del modelo de datos están programados en Java (la capa de negocios está implementada mediante el framework Spring) y utilizan el driver JDBC para accesar a la BD. Se usa JNDI para referenciar a las bases de datos. En la actualizad LAMS opera con una única base de datos que se crea al instalar la herramienta.

Todas las tablas vinculadas a herramientas comienzan con el prefijo tl_ y su nombre se ajusta al formato tl_<signature>_<rest_of_name> donde <signature> es la firma o nombre asignado a cada herramienta.Por ejemplo: tl_latchatl1_conditions.

Las tablas que soportan la persistencia del núcleo (de acuerdo a la arquitectura de LAMS), comienzan con el prefijo lams_seguido del nombre de la tabla. Por ejemplo: lams_learning_design.

Integración de ambos modelos de datos

La instalación de LAMS Lessons como módulo de Moodleimplica dos etapas [5], lo que genera el esquema que se muestra en la Figura I:

1. se crea la base de datos propia de LAMS, de igual forma que si se instala la herramienta de forma independiente.
2. se crean las tablas mdl_lamslesson y mdl_lamslessongrades en la base de datos de Moodle, para permitir luego la integración entre ambos modelos de datos.

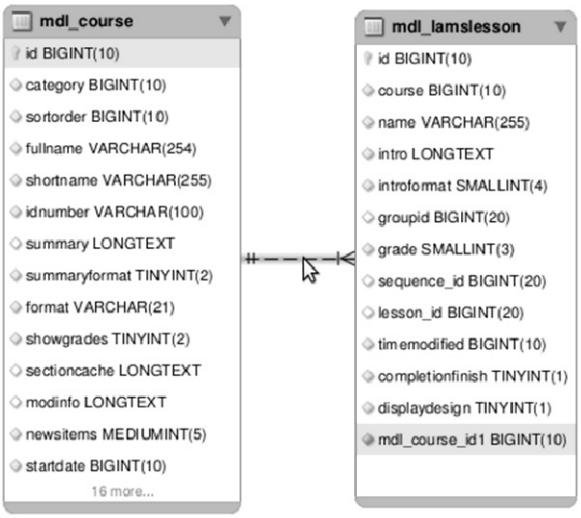


Figura 1. Tablas agregadas en la BD Moodle al instalar el módulo de integración con LAMS

La tabla mdl_lamslesson vincula la secuencia de aprendizaje en la base de datos de LAMS a un curso específico en la BD de Moodle. El campo course en esta tabla vincula a la tabla mdl_course en la BD de Moodle que persiste los datos elementales de un curso. El campo lesson_id vincula a la tabla lams_lesson en la BD de LAMS donde se persisten los datos elementales de una lección. El campo name coincide en ambas tablas (de ambas BD) para referenciar al nombre de una lección concreta.

Learnings Design (LD) Y LAMS Lessons

Se describen en esta sección aquellos aspectos del modelo de datos de LAMS vinculados a la persistencia de los diseños de aprendizaje (LearningDesings – LD) y las secuencias de actividades (o lecciones).

LD (LearningsDesign)

Un diseño de aprendizaje es la descripción de una "unidad de aprendizaje" (Unit of Learning) o una

"unidad de estudio" (Unit of Study), lo cual representa la unidad de contenido más pequeña que satisface uno o más objetivos de aprendizaje. En la práctica puede ser un curso, un módulo, una lección o simplemente una actividad como por ejemplo participar en un foro de discusión [3].

En términos concretos, un LD supone un flujo de actividades que impliquen una participación activa del estudiante en sus aprendizajes. Así mismo, un LD es un modelo que está pensado para ser compartido y reutilizado. En ese contexto los LD incluyen en sus secuencias a objetos de aprendizaje (OA's).
La siguiente Figura 2 ilustra las principales actividades que pueden incluirse en un LD [6].

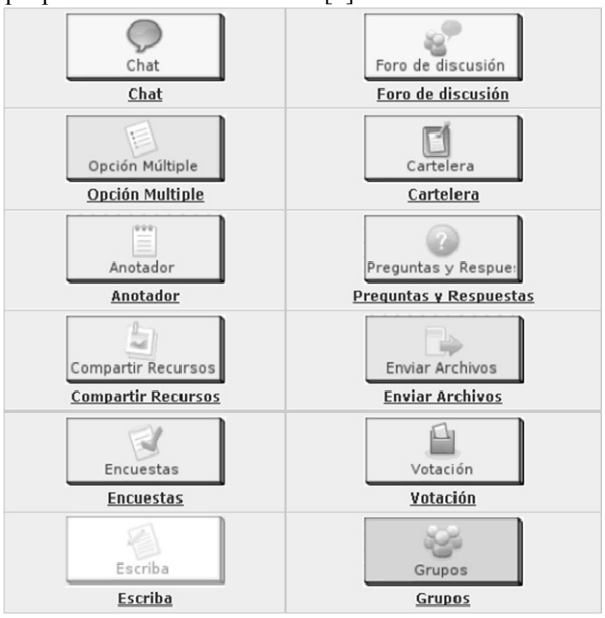


Figura 2. Principales tipos de actividades de LAMS

LAMS es una herramienta basada en LD. Su diferencia con otras basadas en otros estándares, como por ejemplo SCORM (Sharable Content Object Reference Model), radica en que los LD están pensados para crear secuencias de actividades de aprendizaje que impliquen la colaboración entre grupos de estudiantes que interactúan entre sí y con los profesores; y sobre cómo se pueden hacer estas secuencias fácilmente reutilizables.

Aunque se encuentran diferentes definiciones de diseño de aprendizaje, en todas subyace el concepto de personas realizando actividades, utilizando recursos en un entorno determinado.

Es importante destacar que si bien LAMS respeta el concepto de LD [1, 2], no implementa la especificación IMS-LD.

LAMS Lessons (Lecciones)

Una lección en LAMS está compuesta de una secuencia de actividades, un diseño de aprendizaje también. La diferencia entre ambos radica en que los diseños de aprendizaje son la versión estática de las actividades que después, en tiempo de ejecución, se instancian para cada lección. En la estructura de LAMS, un diseño de aprendizaje es independiente de la instancia que se use de él en un curso lección específica.

La Figura3, corresponde al recorte de tablas y relaciones en el modelo de datos de LAMS cuyo análisis permite comprender, en términos generales, cómo se persisten los LD y las lecciones en LAMS.



Figura 3.Modelo de persistencia de LD y lecciones en LAMS

Cada vez que se crea una lección, una entrada es agregada a la tabla lams_lesson y mediante el campo learning_design_idse vincula al diseño de aprendizaje que le corresponde en la tabla lams_learning_design. Múltiples lecciones pueden vincularse a un mismo LD, lográndose de este modo la reusabilidad de los LD pretendida en el diseño de la herramienta.

Cada lección está descripta por un identificador único, un nombre, una descripción, y campos del tipo datetime que determinan su fecha de creación, su fecha programada de inicio y de finalización.

La tabla lams_lesson, se vincula a la tabla lams_lesson_state, la cual es una tabla de parametrización en el modelo que describe los posibles estados de ejecución en los que puede ubicarse una lección (Created, NotStarted, Started, Suspendes, Finished, Archived, Moved). Cada lección está dirigida a un grupo de estudiantes, el que se determina mediante su vínculo a la tabla lams_grouping.

Pueden establecerse relaciones de precedencia entre lecciones. La tabla lams_lesson_dependency es una tabla de relación que permite ligar una lección con una o más lecciones precedentes.

Cada diseño de aprendizaje está descripto por un identificador único, un nombre o título y su descripción. Se mantienen también los datos de fecha de creación, usuario que lo creó y versión. Una bandera (valid_design_flag) indica si el diseño ha sido validado por la herramienta durante su creación. Otra (read_only_flag) determina si el LD puede ser modificado. El campo first_activity_id identifica la actividad inicial en la secuencia de actividades que involucra el LD, ligando a una entrada en la tabla lams_learning_activity.

Cada LD tiene una carpeta de trabajo asociada donde se almacena su contenido. Esta asociación se establece mediante el campo workspace_folder_idque vincula a la tabla lams_learning_design con lams_workspace_folder. Así mismo, muchos LD pueden utilizar la misma carpeta de trabajo, permitiéndose de esta manera la reutilización de contenidos entre distintos diseños.

La tabla lams_copy_type es una tabla de parametrización en el modelo que permite determinar si un LD está vinculado a una lección, a una vista preliminar o a ninguna de ellas. Los valores posibles en esta tabla son LESSON, PREVIEW, NONE.

Cada vez que se modifica un LD, un nuevo registro es agregado en la tablalams_learning_design. Esto evita que el cambio impacte indebidamente sobre múltiples lecciones que utilizaban una versión anterior del LD. No obstante, se incluyen en esta tabla campos que permiten determinar cuál fue la versión original del LD (original_learning_design_id, vincula a la primera versión que se creó del LD y original_user_id: identifica al usuario que creo el diseño original).

En la tabla lams_learning_activity, se persisten las actividades concretas incluidas en cada diseño de aprendizaje. De este modo, cada entrada en la tabla lams_learning_design se vincula a una o más entradas en la tabla lams_learning_activity. En la Figura 3 puede observarse cómo se establece la relación entre una lección, el diseño de aprendizaje que le corresponde y el grupo de actividades incluidas en dicho diseño de aprendizaje.

Los campos de datos que describen una actividad (para cada entrada en la tabla lams_learning_activity) son: un identificador único, su nombre o título y su descripción. También se mantiene para cada actividad las coordenadas de su ubicación en el gráfico que representa la secuencia de actividades y la imagen (archivo .swf) que la representa. Se persiste también la fecha en que la actividad se agregó al LD a la cual está vinculada.

Dado que una actividad puede estar compuesta de otras, por ejemplo en el caso de las actividades opcionales (optionalactivity) o las bifurcaciones (branching), el campo parent_activity_idpermiteestablecer este tipo de relaciones de composición entre actividades (con auto referencias a la misma tabla).

La tabla lams_learning_activity_type, contiene una clasificación de los tipos de actividades de aprendizaje que propone la plataforma, como por ejemplo TOOL , GROUPING, GATE_SYNCH, GATE_SCHEDULE, GATE_PERMISSION, entre otros.

Cada actividad en la tablalams_learning_activityse vincula a uno de estos tipos.

Para cada actividad posible el campo grouping_support_type_iddetermina si dicha tarea puede realizarse en grupo o no. Complementariamente la bandera apply_grouping_flag determina si la actividad debe realizarla algún grupo o toda la clase. En caso que corresponda, mediante el campo grouping_id, se vincula cada actividad a la tabla lams_grouping la cual persiste los grupos de estudiantes a los que puede estar dirigida una lección, una actividad en particular o un grupo de actividades.

Como antes se mencionó, un LD está compuesto de una secuencia de actividades (incluso puede consistir en una única actividad), pero también una actividad puede ser o contener un LD. Si este es el caso, la actividad se vincula (mediante el campo learning_library_id) a la tabla lams_learning_library.Esta tabla mantiene un conjunto de LD prediseñados que pueden utilizarse en la actividad. Algunos de los LD incluidos en esta tabla, consisten sólo de una actividad individual (noticeboard, fórum, question&answer, etc.).

La tabla lams_activity_category, es una tabla de parametrización del sistema, que se utiliza para clasificar a cada actividad dentro de una de las siguientes categorías: SYSTEM, COLLABORATION, ASSESSMENT, CONTENT, SPLIT o RESPONSE.

La tabla lams_tools, mantiene los tipos de actividades predefinidos que la plataforma ofrece (noticeboard, forum, question&answer, etc.). La tabla lams_tools_contentes una tabla de relación que vincula un tool_id en una actividad con su tool_content_id; este último identificador constituye una entrada en las tablas que gestionan el contenido de cada tipo de herramienta (por ejemplo las preguntas y respuestas en un Q&A).

Por último, la tabla lams_learning_transitionalmacena toda transición entre dos actividades secuenciales incluidas un LD. Así cada entrada en la tabla lams_learning_design(cada LD), se ligará a una o más entradas en la tabla lams_learning_transition(a una o más transiciones entre actividades). Cada transición se describe por: un identificador único, su descripción, su fecha de creación y el LD al que está ligada. Además, cada transición vincula dos actividades en la tabla

lams_learning_activity, mediante los campos to_activity_idyfrom_activity_id.

Persistencia, gestión de usuarios y grupos

Al crear un usuario en Moodlela primera tabla que se modifica esmdl_user. Luego de crear cada usuario debe asignarse su rol. En Moodle, cada usuario puede cumplir el rol de profesor o alumno, el cual se registra en la tabla mdl_role_assignments. Dicha tabla extrae los roles de la tabla mdl_role. El profesor podrá crear cursos y los alumnos enrolarse en algún curso. Cuando un alumno se matricula en un curso, el dato se persiste en la tabla mdl_user_enrolments. En el momento de crear un curso se modifica tanto la tabla mdl_course como la tabla mdl_enrol, de manera de crear un identificador para el enrolamiento de los estudiantes en esta última tabla.

Si se quiere realizar una actividad (como por ejemplo una asignación de tareas o un quiz) visible sólo a un conjunto de usuarios dentro de un curso, se necesita primero colocar los usuarios en un grupo y luego ponerlos en un agrupamiento. Un agrupamiento simple puede tener uno o varios grupos. Si se asigna una actividad a un agrupamiento sólo el grupo/agrupamiento que es seleccionado será capaz de ver y hacer la actividad. Será invisible a otros grupos o agrupamientos. En el modelo de datos se plasma la creación de un grupo a través de las tablas:mdl_groups y mdl_groups_members. Al crear un agrupamiento se modifica la tabla mdl_groupings y al asociar un grupo a un agrupamiento se modifica la tablamdl_groupings_groups. Este modelo se muestra en la Figura 4.

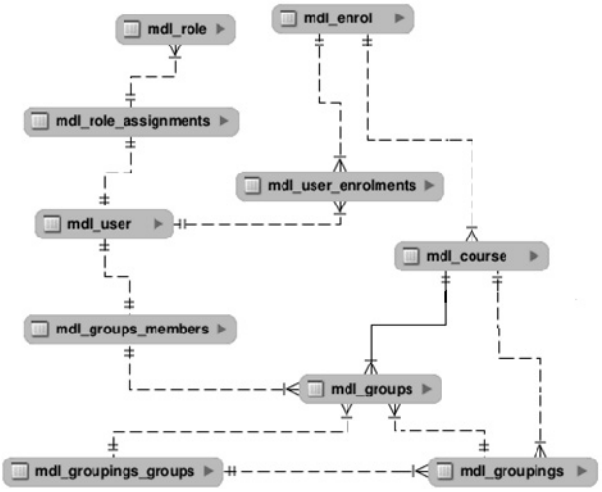


Figura4. Modelo de persistencia de usuarios y grupos en Moodle

Hasta ahora se han explicado los aspectos definidos en el modelo de usuarios de Moodle, en los próximos párrafos se relacionará este modelo con el modelo LAMS.

Como anteriormente se mencionó, al crearse dentro de un curso Moodle una lección LAMS se genera un registro en la tabla mdl_lamslesson. Este registro contiene un identificador de lección, lesson_id, que vincula a la tabla lams_lesson en la base de datos de LAMS. Al mismo tiempo se crea el usuario en la tabla lams_user.

Una vez creado el usuario se le asigna una organización. Esta relación se materializa en la tabla lams_user_organization. Para cada curso en Moodle se crea una organización en la tablalams_organizationen LAMS.Los roles de cada usuario se definen en la tabla lams_user_organization_role, referenciando el user_organization_id y tomando los roles de la tablalams_role.

Los privilegios de usuario para ese rol se persisten en las tablas lams_role_privilegeylams_privilege.

La conexión con el modelo Moodle se establece en la tabla lams_ext_user_org_map_id.

Respecto de grupos y agrupamientos se crean también nuevos registros en las tablas: lams_user_group, lams_groupylams_grouping. El modelo completo de tablas de usuario se muestra en la Figura 5.

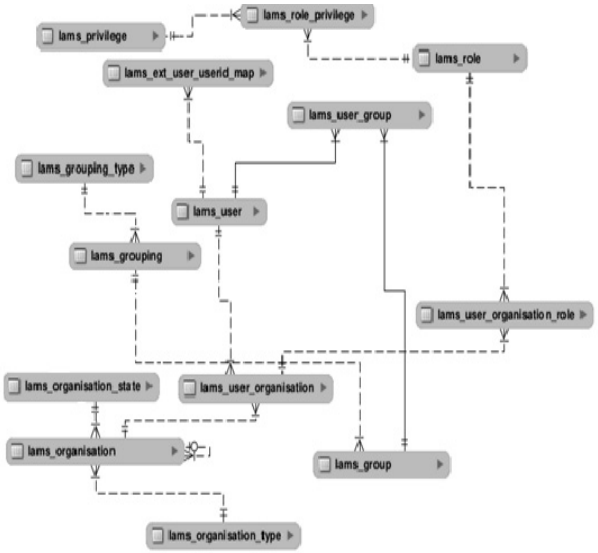


Figura 9. Modelo de usuarios en LAMS

Conclusiones

Como resultado del análisis realizado, se obtuvo un esquema que ha permitido comprender el modelo de

datos subyacente en las plataformas Moodle y LAMS y así también, el modo en que se integran ambos entornos.

Se ha abordado en particular el estudio y análisis de aquel conjunto de tablas, datos y relaciones, que ha posibilitado comprender y explicar desde el modelo de persistencia los siguientes aspectos:

- Cómo se persisten los diseños de aprendizaje y lecciones en LAMS.
- Cómo estas lecciones LAMS, se convierten en una actividad dentro de un curso Moodle.
- Cómo los participantes de un curso Moodle, se convierten en usuarios LAMS.
- Cómo se administran los perfiles, roles y agrupamientos de estos usuarios en una lección LAMS (docentes, alumnos, monitores).
- Cómo se persiste el conjunto de acciones que cada usuario realizó sobre una secuencia de actividades LAMS. En qué momento realizó cada acción y qué opciones tomó en su camino.

Con la comprensión de aquellos aspectos del modelo que responden a estos interrogantes, se espera poder orientar el desarrollo de un aplicativo que, conectándose de forma externa a ambas bases de datos (BD de Moodle y BD de LAMS), extraiga y procese datos almacenados con el fin de reconstruirlas actividades realizadas por cada alumno en el entorno virtual, durante la ejecución de una secuencia de actividades concreta.

El aplicativo permitirá reunir datos acerca de las acciones y decisiones de los alumnos durante la ejecución de las secuencias de actividades LAMS y posibilitará, correlacionar los datos extraídos sobre cada alumno a fin de analizar su desempeño en base a sus estilos de aprendizaje y el rendimiento alcanzado en cada etapa. Esta información se espera que permita estudiar algunos aspectos de cómo es modelado cognitivamente por cada alumno, un concepto matemático.

Agradecimientos

Este trabajo es parte del proyecto “Sistema Adaptativo Aplicado a la Enseñanza del Álgebra” (PID UTN1733), el cual es financiado por la Universidad Tecnológica Nacional – Facultad Regional Mendoza.

Referencias

[1] Britain, S. “A review of learning design: concepts, specifications and tools”. *JISC Report: E-learning Pedagogy Programming*, 2004.

[2] IMS Global Learning Consortium, Inc: “IMS Interoperability Conformance Certification Status”[online].Disponible en: http://www.imsglobal.org/cc/statuschart.cfm

[3] Koper, R, Tattersall, C. “Learning design: a handbook on modelling and delivering networked education and training”. *Springer, Berlin Heidelberg.* New York,2005

[4] Moodle Org. “XMLDB Documentation”. [online]. Disponible en: http://docs.moodle.org/dev/XMLDB_Documentation

[5] Page, J. Ghiglione, E. (2012). “Install Moodle 2 LAMS Plugin” [online].Disponible en: http://wiki.lamsfoundation.org/display/lamsdocs/Install+Moodle2+LAMS+Plugin

[6] Page, J. Ghiglione, E. (2011). “LAMS 2 UserDocumentacion”.[online].Disponible en: http://wiki.lamsfoundation.org/display/lamsdocs/Home

[7] Segura, S., Schilardi, A., León, O., Andía, S., Cívico, C., &Repetto, L. “Análisis de un Concepto de la Geometría Analítica Según la Teoría APOE". *Actas del XVIII EMCI Nacional y X Internacional*. Facultad de Ingeniería de la Universidad de Mar del Plata, 2014. ISBN: 978-987-544-564-2.

[8] Ventura, P. “Crear un esquema EER desde el gesto de base de datos MYSQL Workbench”.Blog,2011. [online]. Disponible en: http://www.pedroventura.com/gestion-de-proyecto/crear-un-esquema-eer-desde-el-gestor-de-base-de-datos-mysql-workbench/

Las nuevas realidades educativas a través de las TICS

Antonieta Kuz[1], Mariana Falco[1], Roxana Giandini[1,2]
[1] *LINSI, Universidad Tecnológica Nacional.*
Av.60 esq. 124 s/n (1900),
La Plata, Buenos Aires, Argentina.
{akuz,mfalco}@linsi.edu.ar
[2] *LIFIA, Facultad de Informática, UNLP.*
Calle 50 y 120, Primer piso (1900),
La Plata, Buenos Aires, Argentina
giandini@info.unlp.edu.ar

Resumen

Las TICS se constituyen en un gran aliado de la docencia, ya que ayudan a hacer más atractiva las asignaturas y el dictado de cursos. El aprendizaje es una actividad constructiva del conocimiento, no reproductiva. El papel del docente no es transmitir lo que sabe, sino ayudar a que el alumno aprenda cómo combinar métodos y técnicas diversas de enseñanza. Los Nuevos entornos virtuales (on-line) de aprendizaje (EVEA) al aprovechar las funcionalidades de las TIC, se multiplican. El presente artículo propone aunar la combinación de los beneficios de las TICs con las herramientas existentes para Redes Sociales, a través de la investigación, el análisis y la aplicación de las mismas. Se ha llevado a cabo un caso de estudio real en uno de los EVEAs, para evidenciar que a través de ellas es factible dar lugar al impulso y el desarrollo de Redes Sociales educativas.

1. Introducción

El concepto de Tecnologías de la Información y la Comunicación (TICs) surge como convergencia de la electrónica, el software y las infraestructuras de telecomunicaciones. La asociación de estas tres tecnologías da lugar a una concepción del proceso de la información, en el que las comunicaciones abren nuevos horizontes y paradigmas. Las TICs atraviesan la vida de las instituciones, organizaciones e individuos de modo tal que su impulso en el sistema productivo como en el educativo y en la sociedad en general, se torna imprescindible.

La realidad socioeducativa nos permite apreciar la diversidad de las diversas relaciones existentes entre profesores y alumnos, como así también valorar una serie de estrategias e instrumentos de medición para determinar las relaciones entre los alumnos, y la detección temprana de situaciones que ocasionan la toma de medidas, como los alumnos aislados.

Por lo tanto, si ajustamos el concepto de comunidad a un entorno educativo y lo delimitamos a aquel que surge de un acto pedagógico, como podría ser una tradicional clase en el aula, la comunidad es el espacio en el que profesores y alumnos, encarnan el ámbito viable para el desarrollo de una fructífera convivencia estrechamente relacionada con las diversas relaciones sociales.

Este nuevo espacio de interacción educativo en conjunción con las TICs, implican la necesidad de definir una estrategia didáctica, tecnológica e innovadora, mediante la combinación de una variedad de herramientas cuyo fin es brindar soporte a profesores y estudiantes. Este artículo propone aunar e interrelacionar los beneficios de las TICs con las herramientas existentes para Redes Sociales a través de la investigación, el análisis y la aplicación de las mismas.

Asimismo, se analizarán los EVEAs mediante una descripción pormenorizada de las características que presentan, adicionando una aplicación de los mismos en el ámbito áulico, vislumbrando así la practicidad y viabilidad de su implementación.

Este trabajo consta de 11 secciones, incluyendo la presente introducción, los agradecimientos y las referencias. El artículo se estructura como sigue: la sección 2 define y contextualiza las TICs. En la sección 3, se detalla el cambio de paradigma que trajo consigo el auge y evolución de las TICs. En la sección 4, se vislumbra la importancia de las TICs al ser representantes un nuevo paradigma tecnológico. En la sección 5, se presentan las TICs como herramientas de

apoyo, aplicadas al ámbito educativo. En la sección 6, definimos las redes sociales educativas junto a los EVEAs, llevando a cabo una evaluación de los roles del alumno y del docente en las redes sociales educativas. En la sección 7, se evidencia un relevamiento de los nuevos entornos educativos, denotando sus características y particularidades. En la sección 8, se evidencian los matices de la realización de un caso de estudio real llevado a cabo con una de las EVEAs. Finalmente, en la sección 9 se presentan las conclusiones y el trabajo a futuro.

2. TICs: Definición e Importancia

En la vida diaria, ya a nadie le sorprende estar informado minuto a minuto, utilizar Skype para comunicarse con personas que se encuentran del otro lado del planeta, visualizar en YouTube el nuevo video de un artista o el *trailer* de una película a estrenar, o incluso trabajar en equipo mediante un documento en el GDrive sin encontrarse en un mismo sitio geográfico.

Las TICs se han convertido a gran velocidad, en una parte primordial de nuestra cotidianeidad. Este concepto es también conocido como Sociedad de la Información. [1] [2] De acuerdo con la declaración de principios de la Cumbre Mundial sobre la Sociedad de la Información , la sociedad de la información debe estar centrada en la persona, integradora y orientada al desarrollo en que todos puedan crear, consultar, utilizar y compartir la información y el conocimiento, para que las personas, las comunidades y los pueblos puedan emplear plenamente sus posibilidades en la promoción de su desarrollo sostenible y en la mejora de su calidad de vida, sobre la base de los propósitos y principios de la Carta de las Naciones Unidas .

Las Tecnologías de la Información y la Comunicación son un concepto muy asociado al de Informática. Si se entiende esta última como el conjunto de recursos, procedimientos y técnicas usadas en el procesamiento, almacenamiento y transmisión de información, esta definición se ha matizado de la mano de las TICs, pues en la actualidad no basta con hablar de una computadora cuando se hace referencia al procesamiento de la información.

Internet puede formar parte de ese procesamiento que, quizás, se realice de manera distribuida y remota. Y al hablar de procesamiento remoto [3] se hace referencia a un dispositivo muy distinto a lo que tradicionalmente se entiende por computadora pues podría llevarse a cabo, por ejemplo, con un teléfono móvil, con capacidad de operar en red mediante comunicación inalámbrica.

Así mismo, puede definirse a las TICs como el conjunto de herramientas relacionadas con la transmisión, el procesamiento y el almacenamiento digitalizado de información (centrado en medios como la informática y las telecomunicaciones), como al conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), en su utilización en la enseñanza [4]. Enfocándonos dentro de las TICs en la informática, destacamos herramientas de software tales como las redes sociales educativas debido a que reducen las barreras del espacio y del tiempo [5].

3. Evolución de las TICs

El concepto de TIC se debe principalmente a un invento que surgió en el año 1969. Internet se gestó como parte de la Red de la Agencia de Proyectos de Investigación Avanzada (ARPANET), creada por el Departamento de Defensa de los Estados Unidos y se diseñó para comunicar los diferentes organismos del país.

Sus principios básicos eran: ser una red descentralizada con múltiples caminos entre dos puntos y que los mensajes estuvieran divididos en partes que serían enviadas por caminos diferentes. La presencia de diversas universidades e institutos en el desarrollo del proyecto hizo que se fueran encontrando más posibilidades de intercambiar información.

Posteriormente se crearon los correos electrónicos, los servicios de mensajería y las páginas web. Pero no fue sino hasta mediados de la década de los 90, en que tuvo lugar la verdadera explosión de Internet. Y a su alrededor todo lo que conocemos como Tecnologías de la Información y Comunicación [6].

4. Nuevo Paradigma de las TICs

Las nuevas tecnologías son inmateriales ya que la materia principal es la información; permiten la interconexión y la interactividad; son instantáneas, y presentan elevados parámetros de imagen y sonido. Al mismo tiempo, suponen la aparición de nuevos códigos y lenguajes, la especialización progresiva de los contenidos sobre la base de la cuota de pantalla (diferenciándose de la cultura de masas) y dando lugar a la realización de múltiples actividades en poco tiempo [7].

El concepto presenta dos características típicas de las nuevas nociones, por un lado es frecuentemente evocado en los debates contemporáneos. Por otro lado, [8] su definición semántica queda borrosa y se acerca a la de la sociedad de la información.

El advenimiento de Internet y principalmente de la *World Wide Web* como medio de comunicación de masas y el éxito de los blogs , las wikis o las tecnologías peer-to-peer confieren a las TICs una dimensión social. El rol de las Nuevas Tecnologías de la Información en los procesos de cambio social y cultural refleja particular relevancia en el ámbito educativo. En este sentido, Edith Litwin [9] sostiene que ciertas concepciones sobre las reformas de los sistemas educativos en distintos países,

atribuyen a la incorporación de estos recursos un efecto determinante en la mejora de la calidad del proceso enseñanza-aprendizaje.

Las tecnologías de la información se aplican al campo pedagógico con el objeto de racionalizar los procesos educativos, mejorar los resultados del sistema escolar y asegurar el acceso al mismo, de grupos convencionalmente excluidos. Por lo tanto, las Nuevas Tecnologías y su incorporación al ámbito educativo promueven la creación de nuevos entornos didácticos que afectan de manera directa tanto a los actores del proceso de enseñanza-aprendizaje como al escenario donde se lleva a cabo el mismo.

5. Las TICs en la educación y el software educativo

La aplicación del software educativo en el proceso de enseñanza-aprendizaje es uno de los grandes aportes tecnológicos al área educativa de las TICs. Esta relación simbiótica entre tecnología y educación surge en un momento clave en la historia del software, especialmente como una herramienta de apoyo a la educación a distancia, de forma que el software educativo evoluciona al mismo tiempo que la tecnología que lo soporta, abre caminos cada día más flexibles y potentes que enriquecen enormemente el proceso educativo.

Es necesario conocer los grandes hitos que han guiado la evolución del software educativo, haciendo especial hincapié en las características inteligentes y, sobre todo, en la influencia de la web, para comprender el porqué de su crecimiento notable y progresivo. Marqués Graells [10] define el software educativo como programas de computadoras creados con la finalidad específica de ser utilizados como medio didáctico para facilitar el proceso de aprendizaje, ofreciendo al usuario un ambiente propicio para la construcción del conocimiento.

Igualmente Galvis [11], entiende al software educativo como programas que desempeñan funciones educativas, sean éstos los que apoyan la administración de procesos educacionales o de investigación, así como los sustentadores del proceso de enseñanza- aprendizaje. Partiendo de estas concepciones, se puede precisar que se denomina software educativo al software destinado a la enseñanza y el auto aprendizaje, y que permite además el desarrollo de ciertas habilidades cognitivas.

Así como tienen lugar profundas diferencias entre las filosofías pedagógicas, también existe una amplia gama de enfoques para la creación de software educativo, haciendo referencia a los diferentes tipos de interacción que debería existir entre los actores del proceso de enseñanza-aprendizaje: el educador, el educando, el conocimiento y los sistemas.

Dentro de la gama que atañe al software educativo, es factible encontrar desde programas orientados al aprendizaje hasta sistemas operativos completos destinados a la educación. Marqués Graells [12] ha estipulado un conjunto de características que permiten brindar una respuesta concreta a este interrogante y es que han sido elaborados con una finalidad didáctica. Además utilizan las computadoras como soporte, son interactivos, se adaptan al ritmo de trabajo del alumno y finalmente son fáciles de utilizar.

5.1. Clasificación de software educativo

Para identificar el tipo de software educativo a desarrollar, generalmente se considera como base la clasificación que realiza Salvat [13], con el objeto de determinar posteriormente cómo se clasificará el tipo de software que se desea obtener (ver Tabla 1).

Tabla 1. Tipos de Software Educativos.

Tipología	Objeto	Decisión sobre el diseño e instructivo
Tutorial	Programas de enseñanza	Conteniendo en función del nivel de los usuarios. Estructuración del contenido. Estrategia didáctica.
Práctica y ejercitación	Programas de ejercicios. Ayuda a la adquisición de una destreza.	Nivel, contenido y estructura de los ejercicios. Tipos de feedback. Tipos de refuerzo. Control del progreso.
Simulación	Proporciona entornos de aprendizaje basados en situaciones reales.	Modelo de simulación. Tipos de feedback.
Hipertextos / Hipermedios	Proporciona un entorno de aprendizaje no lineal.	Organización del contenido. Determinación de los enlaces. Selección de los medios.

6. Redes Sociales Educativas

En aquellas situaciones en las que una red social se utiliza con fines educativos en entornos de aprendizaje se denomina "Edured social". Las redes sociales tienen una innegable utilidad para la educación formal, aunque su mayor logro consiste en el establecimiento de un vínculo que las une con la informalidad, lo que favorece el proceso educativo de los educandos.

Esta unión de lo formal e informal procede de la necesidad de interacción que surge entre los contactos. El aspecto social adquiere un valor importante junto con el aspecto docente, lo que favorecerá un entorno de trabajo amistoso y rápidamente aceptado entre los alumnos. De esta manera, las redes sociales serán ideales para organizar equipos de trabajo mediante los cuales se pueda fomentar la participación cooperativa, la motivación y el interés de los alumnos por su propio aprendizaje [14].

A través de redes sociales como Twiducate, los educandos amplían el apoyo que puedan recibir, tanto dentro como fuera de las aulas. Las formas de enseñar y de aprender han cambiado radicalmente. Especificaremos las diversas herramientas web de asistencia técnica, administrativa y pedagógica, basadas en redes sociales y que generan un impacto en el universo educativo, con las que cuentan los docentes y alumnos, al momento de interactuar, así como para usar y generar contenidos [15].

6.1. Comparación y evaluación de roles alumno y docente en las redes sociales educativas

El Análisis de Redes Sociales (ARS) es una herramienta metodológica de investigación social [16] cuyo objeto de estudio son las relaciones que se dan entre ellos a través de diferentes vínculos, como amistad o a través de los roles que ocupan en una organización [17]. En la Tabla 2, es viable visualizar una comparación que permite determinar de qué manera las redes sociales presentan diferentes funciones según sea el rol del docente o del alumno.

Tabla 2. Comparativa de roles.

Rol Docente

- Ejercer el rol de mediador, ya que su participación es relevante en la creación de la autonomía de los alumnos para construir los aprendizajes.
- Trabajar en torno a la información y sus medios tecnológicos.
- Construir con los alumnos esquemas para percibir, expresar y reaccionar ante diversos hechos.
- Intervención desde el punto de vista de la orientación y la guía del trabajo personal y progresivo de los alumnos.
- Planificar el proceso de enseñanza– aprendizaje.
- Ofrecer informaciones y explicaciones comprensibles.
- Relacionarse con los alumnos.
- Reflexionar e investigar sobre la enseñanza con empleo de las TICs.
- Seleccionar y presentar el contenido de las materias. Enseñar a sus alumnos a buscar, analizar y seleccionar la información.

Rol Alumno

- Protagonista ante las TICS y conlleva un esfuerzo por aprender (no por aprobar un curso solamente o por obtener un título).
- Adquirir los conocimientos, habilidades y destrezas necesarias para el desarrollo profesional.
- Buscar información, seleccionarla (evaluarla, analizarla, y juzgar lo que es útil, recordando que el aprendizaje es un fenómeno social que acontece en el ámbito y como tal es intransferible.
- Familiarizarse con el uso de herramientas de comunicación.
- Aprender y desaprender lo que se aprendido inadecuadamente.

7. Entornos Virtuales de Enseñanza y Aprendizaje (EVEAs)

Dentro de estos Nuevos Entornos Educativos se encuentran las redes, donde se van concibiendo entornos colaborativos entre pares, estilos contextuales y flujos de trabajo, como herramientas para la educación inclusiva. Los Entornos Virtuales de Enseñanza y Aprendizaje (EVEAs) [18] son tecnologías de la información y la comunicación que surgen como producto del trabajo interdisciplinario entre las ciencias informáticas y las ciencias de la educación. Se ha llevado a cabo un estudio y análisis pormenorizado de determinados EVEAs existentes en el mercado. A continuación se esbozan los mismos.

7.1. Edmodo

Es una plataforma web de redes sociales (ver Figura 1) que nació en el año 2008, siendo una red que se ha mantenido en forma gratuita para todos los estudiantes y docentes del mundo. Es posible acceder al portal desde tres perfiles: profesores, estudiantes o padres de familia. Actualmente tiene 3 (tres) millones de usuarios. A través del sistema, los profesores son los administradores de la red de alumnos que forman parte de los grupos, ofrecen evacuar dudas referidas a distintas materias, contestar mensajes, realizar evaluaciones, utilizar la biblioteca virtual y libros de evaluación.

Los estudiantes en cambio, reciben los mensajes de los docentes y pueden responder a los mismos, como así también comunicarse con otros compañeros; también pueden recibir y entregar evaluaciones, esperar a ser calificados y ver los resultados de las evaluaciones.

Finalmente, los padres de familia pueden observar el resultado académico obtenido por su hijo. Su acceso es totalmente consultivo.



Figura 1 . Ingreso a Edmodo.

7.2. redAlumnos

redAlumnos es una red social gratuita para mantener en contacto a profesores, alumnos y padres. Es accesible a instituciones públicas y privadas (ver Figura 2). Permite a los profesores impartir cursos a través de la web. Las opciones de las que dispone son: gestionar plazos de trabajo, enviar mensajes, importar y exportar listados de calificaciones y alumnos, compartir recursos, crear documentos, informar en tiempo real, publicar notas de exámenes, crear y compartir exámenes con autocorrección.



Figura 2 . Ingreso a redAlumnos.

7.3. educaNetwork

educaNetwork es una red social que se basa en la consigna de aprender y enseñar en grupo (ver Figura 3), en la misma pueden crearse grupos que permiten crear cursos; compartir contenidos, como materiales, documentos, vídeos o apuntes, chatear, incluir test multimedia y organizar eventos.



Figura 3 . Ingreso a educaNetwork.

7.4. Diipo

Diipo es una red social didáctica y colaborativa para profesores y alumnos, que también permite la creación de blogs y proyectos. Brinda la posibilidad de conectar con profesores de otras materias y relacionar las clases con otras dentro de su red (ver Figura 4). La interfaz de usuario es presentada como accesible y familiar, con un modelo parecido al de Facebook y sólo está disponible en inglés.



Figura 4 . Ingreso a Diipo.

7.5. Edu2.0

Edu2.0 es una plataforma en español, donde cada organización que la utiliza recibe un portal propio para personalizar. Incluye creación de grupos, registro de notas, chat, foros , noticias, wikis, blogs, mensajería, controles de asistencia, encuestas, portafolio, entre otros (ver Figura 5). Es una red muy completa y segura pensada para trabajar con niños y colegios. Esta plataforma cuenta con más de noventa mil usuarios y dos mil quinientas escuelas.



Figura 5 . Ingreso a Edu2.0.

7.6. Moodle

La herramienta cuenta con una interfaz que funciona en bloques, con actividades o funciones concretas y configurables a elección del docente (ver Figura 6). Facilita la edición y el movimiento en pantalla, así como la posibilidad de agregar recursos y actividades de manera sencilla (*drag and drop*).



Figura 6 . Ingreso a Moodle.



Figura 7 . Ingreso a webUNLP.

Otras características adicionales son: brinda la posibilidad de crear módulos a medida y compartirlos con la comunidad, permite organizar con una estructura jerárquica y barra de navegación siempre visible, tiene diversos roles de usuario (administrador, gestor, creador de cursos, profesor, estudiante, invitado), así como el manejo de grupos y permisos, permite estructuración del curso por diversos criterios (semana, temas, etc.), permite trabajar con definición de grupo, es decir un mismo curso utilizado por comisiones diferentes.

Da lugar al manejo de visibilidad de cursos, recursos y actividades por fecha pero no de unidades completas, permite la incorporación de paquetes SCORM de forma sencilla y rápida. Por otro lado, el paradigma basal del Moodle es el constructivismo social, de ahí que el uso de los foros es casi inherente a la herramienta.

La creación de foros es una de las tareas que un usuario inexperto podría hacer sin mayores problemas. La única dificultad que podría surgir está relacionada con los tipos de foros, ya que actualmente se presentan cinco tipos. En estos casos debe recurrirse a la ayuda o bien al manual de usuario.

Finalmente, en Moodle se presentan distintos tipos de foros y en cada uno se establecen las condiciones de participación. Es el docente del curso el encargado, de acuerdo a algún fin pedagógico, de determinar qué tipo utilizar.

7.7. webUNLP

Se encuentra organizado en áreas, las cuales son unidades pedagógicas que permiten orientar a alumnos y docentes tanto en el desarrollo del curso como en la creación del mismo (ver Figura 7). Cada área está compuesta por diversas herramientas, varias de las cuales se encuentran habilitadas por defecto, y se presentan dentro de una solapa diferente en la pantalla principal del entorno.

Estas herramientas cubren las acciones básicas que los cursos necesitan para ser implementados: contenidos, comunicación (mensajería, foros, cartelera), trabajo grupal, seguimiento de alumnos y docentes y estadísticas de uso. Existen dos formas de que un alumno sea agregado a un curso. Una forma requiere de la acción del alumno el cual debe solicitar la incorporación a un curso en particular. La otra forma la realiza directamente el docente el cual debe buscar al alumno dentro de todos los usuarios de webUNLP y agregarlo al curso deseado.

La mensajería que maneja webUNLP es de tipo interna. Es posible enviar mensaje de manera individual como a grupos creados con anterioridad. Existen diversos criterios para la creación de foros. Es posible crear y gestionar grupos creando un espacio compartido para los mismos donde los integrantes pueden trabajar colaborativamente subiendo y descargando materiales.

8. Caso de Estudio

A partir de los conceptos que constituyen la web 2.0, es factible llevar a cabo la implementación simple y práctica de los EVEAs no sólo como medio de aprendizaje sino también como punto fuerte de incorporación de la tecnología en el aula. Se hace referencia a la Educación 2.0. Ha sido elegida como herramienta para llevar a cabo el presente caso de estudio, la plataforma Moodle debido a que conlleva una gran claridad a la hora de su aplicación en el aula y asimismo, porque su paradigma educacional basal es el constructivismo social. Éste concepto origina que el uso de foros sea un tópico casi inherente a la herramienta.

Asimismo, se ha seleccionado un grupo de 30 (treinta) alumnos con el fin de que interactúen con la herramienta a través de la implementación en Moodle de un curso con un foro asíncrono y una encuesta. Se dice que en un foro no hay un "diálogo" en tiempo real, sino que solamente se publica una opinión que será leída más tarde por alguna persona que tiene la libertad de dejar su comentario.

Ha tenido como objetivo incorporar a los alumnos al trabajo colaborativo en la plataforma. Así, se les entregó un cuestionario teórico y práctico de la materia “Historia” para que resuelvan, pero en vez de ser un trabajo individual se buscó que los alumnos puedan

construir el conocimiento entre todos, al responder en el foro del Moodle dichas preguntas e interactuar entre ellos en el caso que se les presenten interrogantes al resolverlas. Vale mencionar que se decidió probar con "Historia" debido a que los docentes de dicha materia han mencionado la dificultad que se les presenta para que los alumnos completen los trabajos que se les piden, ya que según ellos "es una materia demasiado teórica y hay que leer mucho".

Es conveniente destacar que el tipo de foro utilizado es el de pregunta-respuesta, con el fin de que entre los alumnos puedan hilvanar la respuesta final, al responderse y/o repreguntarse partiendo de la pregunta original. La encuesta tuvo como objetivo primordial que los alumnos puedan determinar cuán satisfactorio fue el trabajo online y qué dificultades encontraron.

El cuestionario que el docente les entregó a los alumnos como trabajo práctico y que debían resolverlo colaborativamente a través del Moodle, fue resuelto exitosamente y en su totalidad. En la evolución del foro, las respuestas fueron de al menos 3 (tres) mensajes por alumno. Los participantes contribuyeron a los foros con un promedio de 15 (quince) mensajes por alumno. Vale destacar que se considera que responder un mensaje ocurre cuando un alumno le responde o comenta el aporte de otro. Cuando el alumno se limita sólo a aportar la acción, ésta acción es interpretada como sólo enviar mensajes.

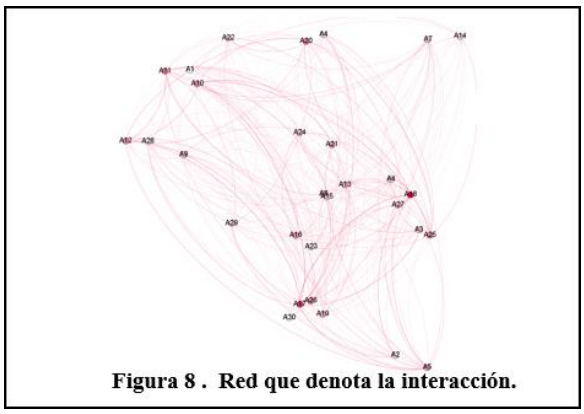


Figura 8 . Red que denota la interacción.

En la Figura 8, puede visualizarse la red que se ha generado por la interacción entre los alumnos y que fue generada mediante la herramienta de software Gephi. En dicha figura es factible denotar que todos los alumnos han participado del foro vislumbrando así un uso exitoso del mismo en el Moodle para llevar a cabo la resolución del trabajo propuesto.

Enfocando ahora la mirada sobre la encuesta, puede dirimirse en la Tabla 3 las preguntas realizadas al alumnado, cuyas respuestas eran del tipo "si/no". En este caso las preguntas formuladas en la Tabla 3 denotan la dos atributos la facilidad de aprendizaje y de utilización del software de la usabilidad. A través de estas preguntas se busco verificar dicha usabilidad de este sistema ya que es considerada hoy en día cómo un atributo que se define como la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso [20].

Tabla 3. Encuesta de utilización del software

Pregunta	Descripción
1	¿Fue de utilidad resolver el trabajo en el Moodle?
2	¿Te generó mucha dificultad el uso del Moodle?
3	¿Fue práctico interactuar con tus compañeros para resolverlo?
4	¿Incorporaste conocimientos nuevos luego de usar el Moodle?
5	¿Usarías el foro como un método para incentivar el trabajo en equipo?
6	¿Volverías a utilizarlo para resolver un trabajo?

La encuesta de la Tabla 3 ha sido procesada para obtener los porcentajes y así poder informar al docente sobre cuáles fueron las valoraciones y perspectivas que tuvieron los alumnos al utilizar el foro. Para la pregunta 1, el 90 % dijo que fue de utilidad. El 16 % mencionó que le generó cierta dificultad el uso del Moodle (pregunta 2) pero que luego de un uso continuado en el tiempo, dejaron las dificultades atrás.

El 95% mencionó que les fue práctico resolver el cuestionario a través del foro y el mismo porcentaje puntualizó que usaría el Moodle para incentivar el trabajo en equipo, debido a que sólo se necesita una pc con conexión a Internet y que les ayuda a evitar la problemática de reunirse cada vez que se les asigna un trabajo en grupo. Para la pregunta 4, hubo una división entre las opiniones. Esto ocurrió debido a que la mitad de los alumnos una vez terminado el trabajo al releer las preguntas pudieron incorporar esos temas y tópicos nuevos; mientras que la otra mitad lo hizo paulatinamente.

Finalmente, el 100% mencionó que volvería a utilizar el Moodle para llevar a cabo la resolución de una encuesta colaborativa. A raíz de los porcentajes obtenidos es factible afirmar que los foros son herramientas de la Web 2.0 que permiten a los usuarios de la comunidad virtual interactuar y colaborar entre sí como creadores de contenido. Los alumnos fueron generadores de contenidos, manifestándose como un fenómeno complejo y de público conocimiento denominado "inteligencia colectiva".

En nuestro análisis nos hemos centrado en el escenario planteado dado que las diferencias de sincronía nos han permitido diferenciar que el participante puede dar un tratamiento más cuidadoso de la discusión y resolver con mayor eficacia problemas complejos. Asimismo se buscó contribuir a la distancia transaccional necesaria para fomentar la reflexión sobre el contenido publicado.

9. Conclusiones y Trabajo Futuro

García Aretio [19] expone en sus investigaciones que "en una comunidad se agrupan personas para interaccionar socialmente, establecer lazos comunes y compartir ciertos intereses, expectativas, creencias, valores y actividades que establecen los límites e identidad diferenciadora del grupo y todo ello, al menos durante un tiempo". Son, en suma, grupos humanos, comunidades de personas que se basan en los intereses, afinidades y valores personales, que discuten, contrastan pareceres y puntos de vistas o intercambian información, a través de Internet, en forma relativamente continuada o lo largo del tiempo y ateniéndose a unas determinadas reglas.

En consecuencia, la era de Internet exige cambios en el mundo educativo y son los profesionales de la educación los que deben tener múltiples razones para aprovechar las nuevas posibilidades que proporcionan las TIC para impulsar este cambio hacia un nuevo paradigma educativo más personalizado y centrado en la actividad de los estudiantes.

Se buscó denotar que los entornos virtuales para la enseñanza y el aprendizaje, se encuentran libres de las restricciones que imponen el tiempo y el espacio en las enseñanzas presenciales y que son capaces de asegurar una continua comunicación (virtual) entre estudiantes y profesores.

Sin lugar a dudas las nuevas tecnologías pueden suministrar medios para la mejora de los procesos de enseñanza y aprendizaje, entrelazado a la gestión de los entornos educativos, debido a que dan lugar y facilitan la colaboración entre los docentes y alumnos. Así, hemos podido relevar las distintas tecnologías de redes sociales y los escenarios sociales que proveen, llevando a cabo un caso de estudio de real en donde fueron los alumnos los que nos han dado su perspectiva sobre el uso de la herramienta elegida.

En futuras investigaciones se buscará desarrollar un soporte metodológico propio que incluya e incorpore técnicas de Inteligencia Artificial, en combinación con técnicas vinculadas al Análisis de Redes Sociales.

10. Agradecimientos

Se agradece al Decano de la Universidad Tecnológica Nacional, Facultad Regional La Plata por el apoyo brindado durante la producción del presente artículo.

11. Referencias

[1] Suárez y Alonso, R. C. (2010) Tecnologías de la Información y la Comunicación: Introducción a los Sistemas de Información y deTelecomunicación. Editorial Ideaspropias S.L.

[2] Camargo, S.R.M. (2012) Globalización Y Tecnologías de la Información Y la Comunicación. Editorial Académica Española.

[3] Malbernat, Lucía Rosario (2010). «Tecnologías educativas e innovación en la Universidad». LaCapitalmdp.com.

[4] G. Sunkel. (2007) Las tecnologías de la información y la comunicación (TIC) en la educación en América Latina: una exploración de indicadores. Number n.125. Naciones Unidas, CEPAL, División de Desarrollo Social.

[5] Azinián, H. (2009) Las tecnologías de la información y la comunicación en las prácticas pedagógicas. Novedades Educativas.

[6] Atiar Rahman (2009). «Conceptos fundamentales y lista» (en inglés). stretdirectory.com.

[7] Demystifying Information Technology and All Things Cyber – Overview, último acceso: 23/03/2014 http://www.counterbalance.org/itdemyst/itdemyst-print.html

[8] Taking it Global, último acceso: 23/03/2014 http://www.tigweb.org/members/book.html?ISBN=014101038X

[9] Litwin, E., Libedinsky, M. (1995) Tecnologia educativa: politica, historias, propuestas. Biblioteca de Cuestiones de Educacion. Editorial Paidós.

[10] Graells, P.M. (1995) Software educativo: guía de uso y metodologia de diseño. Editorial Estel.

[11] Galvis (1997) Ingeniería de software educativo. Colección Biblioclase. Universidad de Los Andes.

[12] Coscollola, M.D., Marques Graells, P. (2011) Aulas 2.0 y uso de las TIC en la práctica docente. Revista Científica de Comunicación y Educación.

[13] Begoña Gros Salvat (1997). Diseños y Programas Educativos: Pautas Pedagógicas para la Elaboración de Software. Ariel Educación. Ariel, Editorial S.A.

[14] Carmona Suárez, E.J., Rodríguez Salinas, E. (2009) Tecnologías de la Información y la Comunicación Ambientes Web para la Calidad Educativa. Editorial Elizcom S.A.S, Colombia

[15] Prato, L.B. (2010) Aplicaciones Web 2.0 - Redes Sociales. Editorial Eduvim. Villa María, Córdoba. ISBN 9871727135, 9789871727131

[16] Hummon, N.P., Doreian, P. (2010) Computational Social Network Analysis. Social Networks, 12(4):2–25.

[17] Jamali, M., Abolhassani, H. (2006) Different Aspects of Social Network Analysis. In Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, pages 66–72.

[18] Muñoz Carril, P.C., González Sanmamed, M. (2009) Plataformas de teleformación y herramientas telemáticas. Barcelona, Editorial UOC.

[19] García Aretio, L. (2001) La educación a distancia; de la teoría a la práctica. Barcelona, Ariel.

[20] International Organization for Standardization ISO 9126: Software Engineering – Product quality, Geneva, Switzerland. ISO 2001.

E-curriculum, Flexibilización Curricular: Un Caso en Matemática

Nadal, Jorgelina.C.
Universidad Tecnológica Nacional
Depto. Materias Básicas
jorgelinanadal@gmail.com

Martinelli, Mercedes B.
Universidad Tecnológica Nacional
Depto. Materias Básicas
mechita13@hotmail.com

Poco, Adriana N.
Universidad Tecnológica Nacional
Depto. Materias Básicas
pocoadriana@gmail.com

Haudemand, Norma Y.
Universidad Tecnológica Nacional
Depto. Materias Básicas
haudemandn@frcu.utn.edu.ar

Constantino, Gustavo D.
CONICET
Depto. TIC - CIAFIC
gustavo.constantino@gmail.com

Brunner, Gisela P.
Universidad Tecnológica Nacional
Depto. Sistemas
giselabrunner.gb@gmail.com

Monetta, Agustín I.
Universidad Tecnológica Nacional
Depto. Sistemas
agustin.monetta@gmail.com

Villanueva Rottini, Juan P.
Universidad Tecnológica Nacional
Depto. Sistemas
juanpablov15@gmail.com

Resumen

Este Este artículo describe una experiencia desarrollada en el marco del Proyecto de investigación: "E-Curriculum: Flexibilización del Curriculum hacia La Sociedad del Conocimiento Complejo en las Carreras de Ingeniería". La que permitió analizar el proceso de aprendizaje y evolución cognitiva de los estudiantes involucrados en la misma.

Esta tiene la particularidad de haber sido aplicada a estudiantes ingresantes al sistema de educación superior y ser un planteo desafiante y novedoso, dadas las condiciones, de desarrollar un tema de forma autónoma, sin explicación previa de los docentes.

Este trabajo describe la estrategia didáctico-pedagógica aplicada por medio de la realización de un Trabajo Práctico que permitiera a los alumnos el desarrollo de diferentes habilidades y competencias, aprendizaje autónomo, búsqueda bibliográfica, selección de la información, integración de los contenidos curriculares al perfil profesional, el aprendizaje colaborativo, trabajo en grupos, la utilización de TIC y otros aspectos.

Palabras claves

e-curriculum; Integración curricular, Educación Superior, Ingeniería, aprendizaje autónomo, aprendizaje colaborativo, matemática discreta.

Abstract

This paper describes an experience developed from a research project called "E-Curriculum: Flexibilization of the curriculum through a society complex knowledge inside engineering careers", which allowed the analysis of the learning process and cognitive improvement of the students involved in it.

A proposal of an innovative challenge in education was implemented with students recently entered into the system of higher education, inviting them to build up a theme without any explanation of the teacher.

This paper describes a didactic - pedagogical strategy applied through the construction of a practical work that makes students improve different skills and competences, autonomous learning, literature search, information selection, integration of curricular

contents to professional profile, collaborative learning, team work, use of ICT and other items.

Key words

e-curriculum,integration of curricular contents, higher education, engineering, autonomous learning, collaborative learning, discrete mathematics.

1. Introducción

La formación de profesionales de la ingeniería requiere de un fuerte compromiso de la Universidad, desde los ámbitos de la gestión hasta las propuestas académicas que compete fuertemente a cada docente a cargo de cátedra. La investigación educativa tendiente a introducir procesos de mejora, necesita de la interacción entre docentes investigadores, alumnos becarios de investigación y docentes de las diferentes materias que comprendan esta problemática, que sean flexibles y que puedan insertarse inicialmente desde un rol de mediadores y facilitadores, creando instrumentos e instancias donde potenciar y promover una formación profesional con competencias actuales y vigentes a la época.(Alberto, M. et. al.; 2013)

Desde lo cotidiano de las experiencias áulicas , estudiantes y docentes, principalmente de las materias de formación básica, refieren dificultades respecto del desempeño de los estudiantes al abordar el estudio de nivel superior. Se requiere establecer nuevos mecanismos didáctico-pedagógicos como un puente entre las disciplinas involucradas, las prácticas y los contenidos, y el propio estudiante; así como para lograr la implementación efectiva de la Ordenanza 643 del Consejo Superior (CS) de la Universidad Tecnológica Nacional (UTN), la cual expresa la necesidad de romper con el esquema tradicional de enseñanza y dotar al estudiante de una mayor autonomía.

Un aspecto que aquí nos interesa es la utilización de las TIC en sus diferentes formas, es así que al comparar y contrastar el método de enseñanza universitaria, en la educación superior, basado en un modelo de curriculum prescripto, definido igual para todos los estudiantes, generalmente de un cuerpo de la información monolítico, donde la evaluación tiene como objetivo medir cuánta información o conocimiento puede ser reproducido y aplicado a tareas simples, testeado con pruebas formales, y el modelo de acceso a la información [1] donde se fortalece la exploración individual, libre y amplia, de bases de datos multimedia, en el cual es el estudiante quién produce una trasformación y síntesis que puede verse favorecida grupalmente para propósitos individuales, sociales y particulares donde los resultados son evaluados como un proceso evolutivo de competencias y otros de acuerdo a una variedad de criterios funcionales.

Al plantear el proyecto damos un paso más y nos atrevemos a introducir una nueva dimensión del conocimiento, la del pensamiento complejo. Ésta corriente cuyo padre es el Filósofo y Pensador Edgar Morin [5] aporta valor al campo de la ingeniería en cualesquiera de sus especialidades, dado la naturaleza de la misma, donde la visión compleja e integral hacen a la calidad de los servicios y/o trabajos. Lejos de interpretar que la complejidad conduce a la eliminación de la simplicidad, el abordaje del pensamiento complejo aspira al conocimiento multidimensional , donde cobra su fuerza el espíritu de los planes de estudio de la UTN, respecto de su modelo de tronco integrador.

Ahora bien, desde el conocimiento podemos marcar diferencias que se plantean entre el curriculum tradicional prescripto y el e-curriculum, el primero se define controlado circunscripto a criterios de relevancia externa, desarrollado por materiales instructivos, masivos producidos por "expertos", existe una mediación y control del contenido curricular, generalmente se utilizan estrategias didácticas expositivas, se produce un aprendizaje predictivo, bajo un modelo top-down de actividades centradas en expertos, hay una enseñanza fuertemente presencial, cara a cara, algunas veces con tutoría on-line. Básicamente el desarrollo es la reproducción del contenido curricular, se plantean objetivos y se esperan resultados estándar, se utilizan pruebas estándar, presentando tareas estándar y textuales. Y el e-curriculum supone algunas ventajas que brinda el acceso a la información básicamente la libertad de indagación, sin límites, potenciada por la posibilidad de acceso a grandes bases de datos multimediales, no sólo textuales y disponibles en la web, a partir de lo cual el individuo o un grupo de individuos son capaces de modificar los criterios que definen el currículo. Los materiales para desarrollar el contenido son desarrollado por docentes y estudiantes. Quizá uno de los puntos más fuertes es que la asistencia de expertos está destinada a dar soporte a la búsqueda de información, se utilizan estrategias dialógicas. [2].

A partir de este proyecto se presenta de la inquietud de participar y fortalecer estos aspectos antes enunciados, los docentes de Matemática Discreta sumaron al esfuerzo en conjunto de generar un trabajo práctico con determinadas características. Como objeto de estudio se trabajó sobre una muestra de 60 estudiantes del 1er. año de Ingeniería en Sistemas de Información, en adelante ISI, de la UTN Facultad

Regional Concepción del Uruguay, en adelante UTN-FRCU, particularmente cursando la materia Matemática Discreta.

La propuesta se ve enmarcada en relación a las definiciones del Plan de estudio [7] que establece lineamientos de desarrollo de prácticos tendientes a favorecer la integración curricular y de relación con el perfil profesional.

La materia desarrollada durante el primer cuatrimestre tuvo la característica especial de ser un curso de ingresantes a la universidad, los cuales pasan por una etapa de adaptación a esta nueva forma de estudiar.

Como estrategia didáctico-pedagógica se elaboró un Trabajo Práctico (Anexo I), en adelante TP que tuviera características particulares, con pautas claras respecto a desarrollo de contenido, fechas de presentaciones parciales, estilo y formato, una propuesta innovadora para la cátedra pues dado un tema de los aún no desarrollados al momento de presentar el TP, los estudiantes podían elegir el sub-tema dentro del mismo abriendo un espacio de aprendizaje autónomo desde la elección de los sub-temas a investigar, desarrollar y vincular con el perfil profesional, sin haber tenido ninguna clase teórica ni introductoria del tema a desarrollar.

Hemos utilizado la plataforma virtual Moodle disponible en UTN-FRCU "Figura 1. Presentación del TP en plataforma virtual Moodle", donde se aloja el aula de Matemática Discreta dentro de la cual se desarrolló un bloque presentando la actividad, acompañado de foros de seguimiento y tutoría además de disponer espacios de encuentros presenciales para fortalecer las tutorías.

2. Detalles de la experiencia

Esta experiencia didáctico-pedagógica tiene algunas característica principales, por un lado el aprendizaje autónomo y colaborativo de los estudiantes y por otro lado el seguimiento continuo por parte del equipo de cátedra. Para ello, se crea un sub-espacio de trabajo dedicado específicamente al acompañamiento de esta actividad, en el cual los estudiantes pudieron, no solo subir los resultados, sino además realizar consultas.

Con la finalidad de darle seguimiento al progreso del TP se diseñó un formato de tres entregas escalonadas respecto del contenido a presentar. Una primer pre-entrega que consistía en la definición de objetivo y alcance; la consigna permitía a cada grupo definir los mismos en relación al trabajo que iban a desarrollar, en este espacio los diferentes grupos presentaron los sub-temas a desarrollar. Una segunda pre-entrega consistió en el desarrollo, aquí debían explayarse en presentar 3 subtemas con aplicación en el perfil profesional y una "entrega final" completa donde desarrollaban una de las 3 aplicaciones respecto del Perfil profesional, en esta instancia también se realizarían observaciones para una posterior corrección.

La evaluación del trabajo, a diferencia de la mayoría de los trabajos prácticos convencionales, no tuvo calificación sobre escala numérica, sino un concepto, podía estar aprobado o desaprobado, el estado de aprobado habilita al alumno a rendir la instancia de "Coloquio", en la cual se evalúa todo el contenido teórico de la materia, incluyendo el tema desarrollado en este TP. Por su parte la instancia de coloquio conlleva una calificación numérica y es condición para promocionar la materia. Por lo que la evaluación está distribuida entre el proceso (con las pre-entregas), el resultado escrito (entrega final) y la defensa oral (coloquio).

Se presentó el TP a los estudiantes a mitad de cuatrimestre, explicando cómo debían expresar objetivo, alcance, se les explicó la forma de evaluación, fechas de entrega y metodología de trabajo. Esta presentación del TP fue presencial en el aula y ese mismo día se les informó la conformación de los grupos, los cuales habían sido establecidos previamente por el equipo docente, siguiendo el criterio de formar grupos equilibrados de personalidades y preferentemente donde los integrantes no fueran compañeros de trabajo habituales. De esta manera contribuir al desarrollo de habilidades de trabajo en grupo/equipo y enriquecer ésta experiencia planteando un escenario de trabajo más realista en donde tuvieran que conocer nuevas personas y adaptarse a las diferentes personalidades teniendo como ejemplo el cumplimiento del TP en tiempo y forma.



Figura 1. Presentación de TP en plataforma virtual Moodle

Con el objetivo de tomar contacto con los estudiantes respecto de la realización del TP se diseño una encuesta que sería completada en forma grupal, con formato de entrevista. Se decidió citar en una fecha y hora determinada para cada grupo, de manera que pudieran asistir de forma ordenada a responderla y expresarse respecto de la experiencia. La misma contó con una serie de preguntas orientadas a indagar sobre las dificultades y los factores que favorecían al trabajo; algunas de ellas con respuesta cerradas con opciones y otras abiertas con campos de texto, de manera de capturar información valiosa a los efectos de la investigación, pretendiendo registrar cuestiones como la reacción y la actitud de los estudiantes con su grupo frente al cumplimiento del TP durante el proceso de aprendizaje.

Utilizando la "Planilla de Análisis y Evaluación de las Entregas" se registró la evaluación de los grupos respecto de: presentación (formato-esquema-ortografía), redacción (lenguaje académico, consistencia, coherencia, cohesión), desarrollo del tema que eligieron, relación con la carrera, teoría de árboles, entrega bibliografía. Para evaluar se definió una escala de valores: 0-No contemplado; 1-Incumplido; 2-Insuficiente 3-Medianamente satisfactorio; 4-Satisfactorio; 5-Muy satisfactorio.

Otro instrumento es una planilla "Seguimiento de grupos" para poder registrar aspectos cualitativos referidos a: situación particular de cada estudiante, observaciones como cumplimiento de (fecha y horario de entrega, objetivo, alcance, formato de entrega), errores frecuentes (redacción, contenido teórico, relación entre conceptos, referencia bibliográfica, citas textuales) y estado (aprobado/ desaprobado).

3. Desarrollo

El trabajar en grupos que ellos mismos no eligieron, les presento varios desafíos, cuestiones de afinidad, cercanía, disponibilidad horaria, intercambio de ideas, datos que se pudieron recabar en la encuesta realizada después de la segunda pre-entrega. Del análisis de la encuesta, surge que la organización grupal fue la cuestión más relevante, lo cual han visto reflejado los tutores debiendo intervenir al comienzo de esta actividad para lograr que los estudiantes establezcan el contacto inicial.

Esto se ha debido a varios motivos, al tratarse de estudiantes ingresantes y por cómo se conformaron los grupos, se separaron los grupos a los que se estaban habituados y no todos los miembros de cada grupo se conocían mutuamente. Por otro lado, las distancias físicas de los lugares de origen y los diferentes horarios de cursada y trabajo resultó un desafío motivador para algunos grupos para emplear los medios web para hacer tanto trabajo on-line como off-line.

En un principio, habiendo previsto estas posibles dificultades, se propuso un horario de tutoría presencial, además de disponer de un foro en la plataforma para consultas sobre el trabajo. Ambos intentos no resultados fructíferos por la poca participación de los estudiantes, fue por ello que no se logró en primera instancia establecer un contacto personal y se decidió formular una encuesta pre-formateada a modo de entrevista obligatoria por grupo, con la finalidad de crear un espacio de diálogo personal y obtener información sobre cómo realizaban el trabajo y la impresión sobre la propuesta.

Llegada la fecha de entrega final, a pesar de los retrasos de algunos grupos, se evidenció los resultados del seguimiento continuo en los desarrollos, ya que con cada entrega los trabajos presentaban correcciones o adicionales según los comentarios de los evaluadores dispuestos sobre los documentos entregados.

Para realizar estos comentarios, se dispuso que las entregas fueran virtuales (ver Figura 1: Presentación de TP en plataforma virtual Moodle) en formato PDF, de esta manera, y empleando el software FoxitReader ® como herramienta de lectura, se dispusieron notas a lo largo del documento. Esta herramienta no solo permitió escribir notas en prosa sino, además, dotarlas de un lenguaje paralingüístico empleando colores distintos según el tipo, hacer recuadros y subrayados con notas, y símbolos (como flechas, cruces, triángulos y círculos) cada uno con un significado para cada caso.

4. Información recogida

Durante el proceso de seguimiento sobre las correcciones, se realizo una etapa de ponderación sobre los puntos abarcados en el TP por cada etapa de entrega y complementando al relevamiento se realizo encuesta directamente a los integrantes de los grupos.

A partir de la Planilla de Análisis y Evaluación de las Entregas que contiene la evaluación de los grupos en las categorías: presentación (formato-esquema-ortografía), redacción (lenguaje académico - consistencia, coherencia, cohesión), desarrollo del tema que eligieron, relación con la carrera, teoría de árboles, entrega bibliografía; se obtuvo el siguiente gráfico que muestra la evaluación de la categoría presentación (verde-formado, naranja-esquema, azul-ortografía) de la "entrega final":

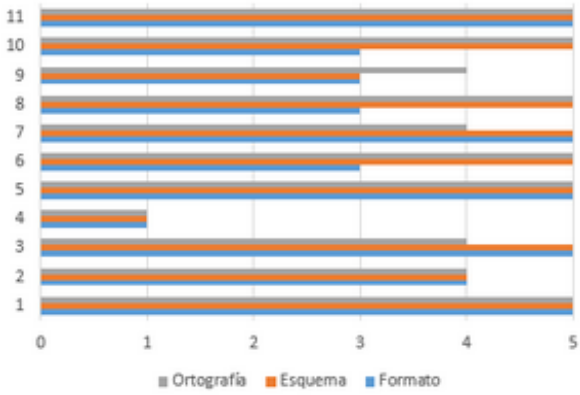


Gráfico. 2. Categoría presentación

En cuanto a la Ortografía:

- Los grupos 1 - 3 - 5 - 6 - 11 refleja una puntuación de 4 considerándose satisfactorio.
- Los grupos 2 - 4 - 8 - 9 - 10 poseen una puntuación de 3 considerándose medianamente satisfactorio.
- Se observa que el grupo 7 tiene una puntuación de 1 unidad indicando incumplido.

En cuanto al Esquema:

- Los grupos 1 - 4 - 10 tienen una puntuación de valor 4 considerándose satisfactorio.
- Los grupos 2 - 3 - 5 - 6 - 8 - 9 poseen un valor de 3 unidades considerándose medianamente satisfactorio.
- El grupo 7 tiene una puntuación de una unidad considerándose incumplido.
- El grupo 11 posee una puntuación de 5 considerándose muy satisfactorio.

En cuanto al Formato:

- Los grupos 1 - 3 - 5 - 6 - 11 tienen una puntuación de valor 4 considerándose satisfactorio.
- Los grupos 2 - 4 - 8 - 9 - 10 obtuvieron un merito de valor 3 considerándose medianamente satisfactorio.
- El grupo numero 7 obtuvo un valor 1 considerándose incumplido.

En general sobre la presentación, los grupos tuvieron un buen nivel de desempeño evaluado como "Satisfactorio" respecto de la subcategoría Ortografía, pero considerando correcciones a realizar, un desempeño "satisfactorio" en la categoría Formato, pero con necesidad de realizar cambios y un menor desempeño evaluado como "medianamente satisfactorio" en la categoría Esquema, lo cual evidencia una corrección adecuada.

Anteriormente detallamos como realizamos en una categoría particular el análisis y la evaluación sobre el seguimiento de los grupos en las distintas etapas. En el siguiente gráfico se tomó la sumatoria de todas las puntuaciones de las categorías incluyendo a todos los grupos. Considerando cada barra una etapa en cuestión (primera pre-entrega, segunda pre-entrega, entrega final).

La experiencia final sobre todas las categorías en cada etapa lanzó el siguiente resultado final en "Gráfico 3. Resultado por categoría".

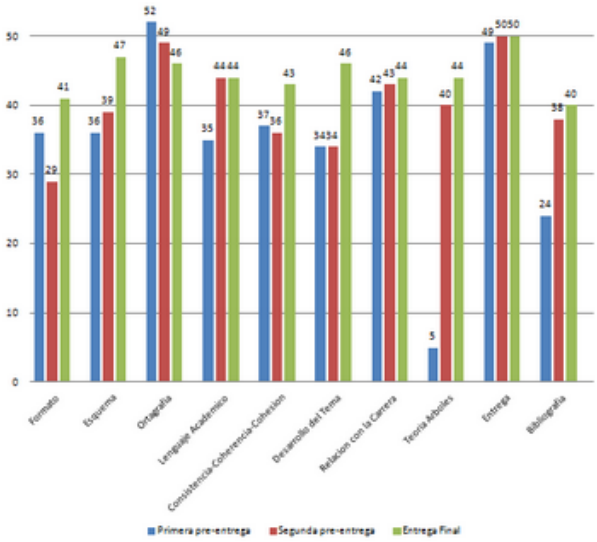


Gráfico 3. Resultados por categoría

Esta evaluación nos permitió observar un nivel satisfactorio sobre la etapa final cumpliendo con todos

los aspecto requeridos ya que todas las categorías superan un puntaje de 40. También se destaca una evolución entre cada etapa siendo gracias a una evaluación detallada y completa de los coordinadores de la materia y en conjunto a los tutores, estos últimos asignados para guiar a los estudiantes al desarrollo de la propuesta.

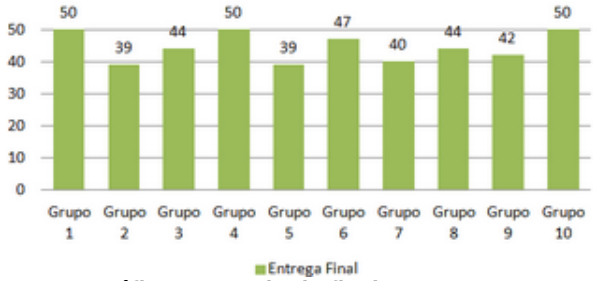


Gráfico 4. Resultado final por grupo

Sobre el gráfico (ver Gráfico 4. Resultado final por grupo) anterior se organizo la exposición de los resultados tomando la sumatoria del puntaje total del grupo sobre todas las categorías de la planilla de análisis y evaluación de las entregas. Se observa que los grupos obtuvieron una aprobación satisfactoria por sobre un puntaje de 40 en relación a la entrega final.

La entrevista, mencionada anteriormente, realizada después de la segunda pre-entrega del TP, llevada a cabo por los tutores y un auxiliar de cátedra, consistió en 10 (diez) preguntas predefinidas realizadas de forma oral y, luego, registradas por escrito, siguiendo un orden preestablecido aunque flexible. Se recibieron en forma paralela de a 2 (dos) grupos, en algunos casos completos y en otros solo el representante, teniendo una duración entre 15 a 20 minutos por cada grupo. Algunas preguntas fueron con respuestas cerradas y otras con respuestas abiertas, permitiendo ésta última que los entrevistados se explayen más en sus respuestas.

Basándose en los resultados de las encuestas, los aspectos positivos más remarcados fueron la aceptación de la modalidad de trabajo respecto al uso de la plataforma virtual y libre elección del tema a desarrollar, y la apreciación del método de corrección como cómodo, ágil, novedoso y orientativo.

Dentro de los inconvenientes que se fueron presentando a en el transcurso del trabajo, los primeros en evidenciarse fueron respecto de la organización grupal, por cuestiones de afinidad, desconocimiento mutuo, disponibilidad horaria (diferencias en horarios de cursada y trabajo), geográfica, y en segundo plano en la búsqueda bibliográfica, al no estar habituados al empleo de fuentes confiables. Para solucionar los problemas de disponibilidad, cuentan haber utilizado medios tecnológicos de comunicación como Skype, Facebook y Google Drive para trabajo de modo tanto on-line como off-line.

No expresaron dificultad alguna en la comprensión de la consigna de trabajo, redacción del TP o uso de la plataforma virtual. Sin embargo no se utilizaron funciones tales como foros, mensajes, calendario, como se pretendía, debido a la falta de costumbre; y los resultados de la evaluación reflejan carencia en estos aspectos.

5. Resultados

La evaluación continua permitió enfocarse al desarrollo de competencias y apreciar la evolución de cada trabajo práctico (TP) acorde al desempeño del grupo, ésta evolución en parte motivada por las correcciones que realizaban los profesores mediante comentarios reflejados en los documentos y por otra la iniciativa de los estudiantes al re-entregar el TP, redactando, corrigiendo, eliminando contenido, agregando bibliografía o mejorando referencias.

Se procuró que las correcciones o comentarios que realizaban los docentes de cátedra y colaboradores fueran del tipo abiertas, es decir no puntuales ni taxativas, esto dejó en evidencia , en varios grupos, la pro-actividad al responder a ellos, en algunos casos, incluso agregando una sección entera y en otros complementando las explicaciones con algoritmos o desarrollando ejemplos.

La mayor evolución se notó en el modo de redacción, el ordenamiento lógico de la exposición de las ideas y la redacción del objetivo. En este último punto en particular se evidenciaron las mayores dificultades al principio, si bien todos los grupos pudieron expresaron claramente los temas a desarrollar, no así el objetivo propio, enunciados que asemejaban a introducciones.

Estos aspectos se vieron reflejados en la "Planilla de Análisis y Evaluación de las Entregas" mediante un aumento significativo del promedio y disminución de la desviación estándar. Los TPs reflejan un gran crecimiento en la primera etapa y un crecimiento menos notable en la segunda, según muestra la "Tabla 1. Promedios y desviación estándar" realizada a partir de los grupos que entregaron en cada instancia.

Tabla 1. Promedios y desviaciones estándar.

Etapa	Promedio	Desviación estándar
Primera Pre-entrega	31,82	5,92
Segunda Pre-entrega	40,2	4,7
Entrega final	41,36	4,30

Sin embargo, se concluyó que esto no era suficiente para evaluar el crecimiento de los grupos, ya que esta evaluación solo considera la adecuación a las expectativas de cada etapa, por lo que se decidió incorporar una devolución extra que reflejará este aspecto, teniendo en cuenta los cambios que había de una versión a otra.

Así como menciona Malva, Alberto et. al., en su trabajo [4] el plagio también estuvo presente en esta circunstancia, sin embargo fue, no solo un inconveniente, sino una oportunidad de crecimiento. Para enfrentarnos a este problema y asumiendo una formación insuficiente en este aspecto, se invito a una investigadora, de nuestra facultad para realizar una charla de concientización sobre la relevancia y las implicancias que podría llegar a tener esta acción desde el punto de vista profesional y estudiantil. Esto dio muy buenos resultados ya que hubo una importante evolución en la formación profesional al destacarle la importancia de originalidad y el respeto por los derechos de autoría. Al realizar las respectivas correcciones, poniéndose en evidencia la poca redacción personal, que fue necesario complementar para otorgar coherencia al práctico, debido a las referencias ahora sí explícitas.

Del análisis de la entrevista se pudo arribar a la conclusión que, a pesar de las dificultades que presentaba la propuesta, resultó motivador el tener la libertad de una consigna mayormente abierta a la creatividad, a la elección del tema a desarrollar y trabajar con compañeros distintos a los habituales.

Una característica destacable de las observaciones y comentarios respecto del contenido realizadas es que no tienden a marcar directivas, sino que son interrogantes con la finalidad que sirvan de guía, pero dejando a criterio de los estudiantes el enfoque de su práctico. Esto, sin embargo, difiere en caso de copia textual en el cual se realiza una indicación directa de corrección.

Al finalizar el plazo de entrega pautado, un 50% de los trabajos entregados se encontraron en condición de aprobación y un 30% en una entrega siguiente, siendo el 20% restante entre deserciones de la cátedra y correcciones no realizadas.

Los resultados del aprendizaje pudieron ser apreciados por los docentes durante el desarrollo de la clase teórica realizada con bastante posterioridad, en la fluidez de diálogo y capacidad para deducir las fórmulas relativas a esta estructura, cambio notable del curso respecto a clases anteriores, pudiendo adquirir un rol más activo, tal como prevé la Ordenanza de Consejo Superior [6]. Los estudiantes pudieron interactuar en diálogo con el profesor y deducir las formulas relacionadas con las propiedades de los árboles, diferenciándose de clases anteriores donde el desarrollo de temas nuevos los estudiantes presentaron una actitud pasiva.

6. Discusión

En próximas experiencias, se espera lograr una comunicación constante con los estudiantes, promover el empleo de los medios virtuales para ello, como la participación en foros. Esto último se considera que no se pudo lograr ya que, como reflejan las encuestas, los estudiantes ingresantes no estaban habituados al uso de estas herramientas.

Con el objetivo de contribuir al desarrollo de habilidades oratorias, y se invitará la participación de un equipo docente especializado en redacción para acompañar el proceso hasta la entrega final del TP.

Por último, lograr la integración con otras cátedras, como prevé el plan curricular de la carrera, en pos de la formación total del ingeniero, esperando sea un primer paso a la propuesta de TP que no caiga en mero aglomeramiento de información, sino un TP interdisciplinario, articulado entre las diferentes materias, en el cual los conocimientos construidos por el estudiante le resulten en un único esfuerzo por un objetivo en común.

7. Conclusión

El factor de éxito más importante para este tipo de trabajo ha sido la motivación lograda en los estudiantes debido a las características de esta propuesta. He aquí unos de los principios, principio de autonomía / dependencia (auto-eco-organización) destacados por Edgar Morín [5] sobre llevar a los estudiantes a una instancia de autonomía para llegar a ser profesionales exitosos pero a su vez creando la necesidad de seguir adquiriendo conocimiento a través de diferentes medios. (revisar lo de auto-eco-organización).

Aunque la libertad de elección del tema a desarrollar, haya sido el aspecto más destacado del TP, no ha sido el único. Se valoró poder reconocer la integración y utilidad de los contenidos desarrollados en la formación del perfil profesional.

Es menester recodar que todo trabajo ha de estar pensado con esta finalidad, formar en aptitudes, habilidades y competencias, y que el estudiante pueda percibirlo como real, de esta manera se convierta en un factor motivador.

Por ende, el compromiso docente, el planteo de un desafío novedoso, atrayente, pero alcanzable, y el seguimiento continuo fueron los aspectos clave para el éxito de esta actividad didáctico-pedagógica que, si

bien ha requerido un esfuerzo considerable para cumplir con los plazos pautados, ha brindado resultados satisfactorios, contribuyendo en la formación de los estudiantes según los objetivos planteados por la cátedra y el trabajo, reconociendo en la web un medio para la ampliación del currículum prescripto.

8. Bibliografía

[1] Alonso Izquierdo, A., Sevilla, A., Isabel, M., Bustos Muñoz, M. T. D., Chaves Tolosa, M., Díaz Len, R., ... & Visus Ruiz, M. I. (2011). Memoria del proyecto: e-MATE. http://hdl.handle.net/10366/111147

[2] Constantino, Gustavo D. (2010) "Del curriculum interdisciplinar al e-curriculum: nuevos modelos para la construcción del conocimiento del profesor". Rivista Formazione & Insegnamento, vol. 2010, Nº1 (en prensa).

[3] Lemke, J. Las Próximas Guerras de Paradigmas en Educación: Curriculum vs.*Acceso a la Información, en revista Did@ xis_online, 1*(1).

[4] Malva, Alberto; Frausin, Adriana; Castellaro, Marta; 2013"Desarrollo y uso de aplicaciones tecnológicas: un desafío interdisciplinario en aulas de ciencias y tecnologías básicas". Universidad Tecnológica Nacional, Facultad Regional Santa Fe http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/213-432-1-DR.pdf

[5] Morin, E., & Pakman, M. (1994). *Introducción al pensamiento complejo*. Barcelona: Gedisa.

[6] Ordenanza del Consejo Superior de UTN Nro. 643 sobre el "Régimen de promoción. directa" http://www.frbb.utn.edu.ar/comun/secretaria_academica/Ordenanza_CSU_643_RPD.pdf

[7] Plan de Estudios, Ingeniería en Sistemas de Información, última modificación año 2013. http://www.frcu.utn.edu.ar/archivos/planes/plan_sistemas2_2013.pdf

LAS COMPETENCIAS EN INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD DE LA MARINA MERCANTE. ANÁLISIS Y APLICACIÓN DE COMPETENCIAS.

Marcovecchio, Osvaldo Donato
Universidad de la Marina Mercante

Conde, Sergio Daniel
Universidad de la Marina Mercante

Resumen:
El objetivo de la investigación es demostrar la aplicación de las competencias en la formación del Ingeniero en Sistemas y su relación con el perfil del egresado de la Universidad de la Marina Mercante.
Se parte de la Hipótesis: La aplicación de competencias del plan de estudios de las asignaturas de Ingeniería en Sistemas tiene relación en la formación del perfil del egresado de la Ingeniería en Sistemas de la Universidad de la Marina Mercante.
Se aplica una metodología detallada donde se integran diferentes elementos que permiten identificar el análisis cualitativo y cuantitativo comparando el plan de estudios con los contenidos que se evalúan en asignaturas de tecnología básica y tecnología aplicada, los elementos que se cumplen del perfil del egresado en cada una de las asignaturas de tecnología básica y tecnología aplicada.
Por ultimo establecer la cantidad de alumnos aprobados y desaprobados en asignaturas de tecnología básica y tecnología aplicada.
El tipo de diseño es cuantitativo / cualitativo.

PALABRAS CLAVE.
Competencias, Tecnología Básica, Tecnología Aplicada.

ABSTRACT

The objective of the research is to demonstrate the application of skills in the formation of Systems Engineering and its relation to the profile of the graduate of the Universidad de la Marina Mercante.
A detailed methodology is applied where different elements are integrated allowing the identification of qualitative and quantitative analysis comparing the curriculum with the content evaluated in basic technology and applied technology subjects, the elements that shall meet the graduate profile in each of them.
Finally, it is set the amount of students that approve and disapprove the subjects of basic technology and applied technology.
The type of design is quantitative / qualitative.

KEY WORDS.
Skills, basic technology, applied technology

1. Introducción

El objeto de la investigación comprende identificar la aplicación de competencias en cada una de las asignaturas que componen plan de estudios Ingeniería en Sistemas de la Universidad de la Marina Mercante en la formación de Ingenieros y su relación en el perfil del egresado.
Comparar el perfil del egresado y su relación con las competencias aplicadas en cada una de las Asignaturas que comprenden el plan de estudios.
Identificar las competencias en asignaturas de Ingeniería en Sistemas.
Obtener los análisis de variables, indicadores y conclusiones pertinentes de las competencias que intervienen en la formación de Ingenieros en la carrera de Ingeniería en Sistemas de la Universidad de la Marina Mercante.

Elementos y metodología

2. Desarrollo

Las Competencias

Spencer y Spencer (1999): entienden la competencia como una "característica subyacente en el individuo que está causalmente relacionada con un estándar de efectividad y/o una performance superior en un trabajo o situación.

En la definición de Spencer y Spencer puede distinguirse tres elementos importantes:
En primer lugar la idea de característica subyacente resaltando que en parte competencia supone una parte profunda de la personalidad y predice comportamiento en una amplia variedad de situaciones y desafíos laborales.
En segundo lugar, se afirma que está causalmente relacionada, lo que quiere expresar que la competencia origina o anticipa el comportamiento y el desempeño. Finalmente se usa el término de estándar de efectividad, que significa que puede determinarse si alguien desempeña la competencia bien o mal al referenciarlo con un criterio estándar.
Perrenoud(2004), entiende a la competencia como capacidades muy amplias, que implicar elegir y movilizar recursos, tanto personales conocimientos, procedimientos, actitudes como de redes (bancos de datos, acceso documental, especialistas y realizar con ellos una atribución contextualizada en especio, tiempo, relación).
Existen tres elementos que caracterizan a las competencias como:
a) Articulan conocimiento conceptual, procedimental y actitudinal. Acumular conocimientos no implica ser competente necesariamente. La sumatoria de saberes y capacidades no nos introduce y lleva a la competencia. El ser competente supone, de todo el acervo de conocimiento que uno posee o al que puede acceder, seleccionar el que resulta pertinente en aquel momento y situación desestimando otros conocimientos que se tienen pero que no nos ayudan en aquel contexto para poder resolver el problema que se enfrenta.
b) Se vinculan a rasgos de personalidad pero se aprenden: el hecho de poseer de forma innata ciertas inteligencias es un buen punto de partida pero no garantiza ser competente.
Las competencias deben desarrollarse con formación inicial, con formación permanente y con experiencia a lo largo de la vida.
Las competencias tienen un carácter recurrente y de crecimiento continuo.
c) Toman sentido en la acción pero con reflexión: para ser competente es imprescindible la reflexión, que nos aleja de la estandarización del comportamiento.
En el año 2007 Perrenoud define: “Competencia es la capacidad de administrar en forma eficaz un conjunto de estructuras y valores, permitiendo disponer de diferentes conocimientos, que por lo general, son disciplinarios, en un contexto determinado con el fin de resolver situaciones profesionales específicas.
El grado de competencia se encuentra relacionado a la eficiencia con que se apliquen un conjunto de estructuras y valores dando el conocimiento origen a distintas competencias en diferentes personas.
El Consejo Federal de Decanos de Ingeniería (CONFEDI) en el año 2007 adopta un esquema con diez competencias genéricas de la ingeniería, desagregadas en niveles simples e integradores de capacidades. Distingue dos clases: las tecnológicas; y las competencias sociales, políticas y actitudinales.
Lo expresado, indica que la competencia es una construcción compleja, en relación con una situación multidisciplinaria que cobra relevancia cuando se relacionan modelos educativos desde los contenidos y objetivos.
Según la OIT(2000) la competencia es la capacidad efectiva para llevar a cabo exitosamente una actividad laboral plenamente identificada. Las competencias son el conjunto de conocimientos, procedimientos y actitudes combinados, coordinados e integrados en la acción adquiridos a través de la experiencia formativa y no formativa que permite al individuo resolver problemas específicos de forma autónoma y flexible en contextos singulares.
Para Le Boterf (2000) las competencias son un saber combinatorio y que no se transmite sino que el centro de la competencia es el sujeto-aprendiz que construye las competencias a partir de la secuencia de las actividades de aprendizaje que movilizan múltiples conocimientos especializados. El profesor sólo crea condiciones favorables para la construcción siempre personal de las competencias.
Tres elementos que caracterizan las competencias pueden aproximar a su comprensión:

a) Articulan conocimiento conceptual, procedimental y actitudinal pero van más allá: El hecho de acumular conocimientos no implica ser competente necesariamente. La sumatoria de saberes y capacidades no nos lleva a la competencia. Ser competente implica un paso más: supone, de todo el acervo de conocimiento que uno posee o al que se puede acceder, seleccionar el que resulta pertinente en aquel momento y situación desestimando otros conocimientos que se tienen pero que no nos ayudan en aquel contexto para resolver el problema o reto que enfrentamos.
b) Se vinculan a rasgos de personalidad que se aprenden. El hecho de poseer de forma innata ciertas inteligencias es un buen punto de partida pero no garantiza ser competente.
Las competencias deben desarrollarse con formación inicial, con formación permanente y con experiencia a lo largo de la vida. Se puede

ser competente hoy y dejarlo de ser mañana o serlo en un contexto y dejarlo de ser en otro contexto que no me resulta conocido. Las competencias tienen un carácter recurrente y de crecimiento continuo donde nunca se "es" competente para siempre.

c) Toman sentido en la acción pero con reflexión. El hecho de tener una dimensión aplicativa en tanto que suponen transferir conocimientos a situaciones prácticas para resolverlas eficientemente no implica que supongan la repetición mecánica e irreflexiva de ciertas pautas de actuación. Al contrario, para ser competente es indispensable la reflexión, que se aleja de la estandarización del comportamiento.

La evaluación

En palabras de Miller, la evaluación orienta el currículum y puede, por lo tanto, generar un verdadero cambio en los procesos de aprendizaje.

McDonald, R,;Boud, D.; Francis, J. y Gonczi, A. (2000) ayudan a entender que la evaluación no puede limitarse a la calificación (sino que ésta es un subconjunto de la evaluación); no puede centrarse en el recuerdo y la repetición de información (sino que se deben de evaluar habilidades cognitivas de orden superior) y que no puede limitarse a pruebas de "lápiz y papel", sino que se requieren instrumentos complejos y variados.

La evaluación debe constituir una oportunidad de aprendizaje y utilizarse no para adivinar o seleccionar a quien posee ciertas competencias, sino para promoverlas en todos los estudiantes. Esta dimensión formativa formulada por Scriven ha sido abordada ampliamente en los últimos años por Hall y Burke(2003) y Kaftan et al. (2006).

La evaluación ha de hacer más conscientes a los estudiantes de cuál es su nivel de competencias, de cómo resuelven las tareas y de qué puntos fuertes deben potenciar y qué puntos débiles deben corregir para enfrentarse a situaciones de aprendizaje futuras. Este proceso de autorregulación, tratado, entre otros, por Boekaerts, Pintrich y Zeidner (2000) va a ser esencial para seguir aprendiendo a lo largo de toda la vida constituyendo en sí mismo, una competencia clave.

En los sistemas de formación profesional vigentes existe un mercado del aprendizaje en el que los jóvenes que salen de los niveles escolares o los que abandonan la dedicación completa a los estudios pueden buscar una plaza de aprendices en una empresa haciendo valer su capital escolar. (cf. Amos 1982, 1984).

Se considera el contexto de la institución, entendido como los propósitos plasmados en su filosofía, misión y visión (Cashin, 1996); las características que definen la gestión académica y administrativa del plan de estudios, tales como: el número de horas asignadas a cada profesor frente a grupo, la cantidad de asignaturas distintas que enseña, el total de alumnos que atiende, la modalidad de enseñanza: presencial, a distancia, taller, prácticas de campo; asimismo, la cultura institucional que permite apreciar el conjunto de normas y valores compartidos por los miembros de la organización, en particular las relacionadas con el trabajo colegiado y colaborativo por ser elementos claves de la cultura institucional (Zabalza, 2002).

Los principios particulares que guían el modelo de evaluación de competencias docentes para educación media y superior y que permiten su desarrollo (García-Cabrero, Loredo y Carranza, en prensa) son:

- Orientación formativa. El modelo permite al profesor reflexionar y retroalimentar sobre su acción en la docencia y plantear acciones para su mejora.
- Orientación participativa. La evaluación/formación de la práctica docente no es elaborada e instrumentada por las autoridades de una institución educativa, o por un grupo de especialistas ajenos a los profesores, es el docente quien se involucra en el proceso y participa en su diseño.
- Orientación humanista. Considera al docente como una persona, como un ser humano, con preocupaciones, intereses, necesidades, emociones; de ahí que busca la preservación de su dignidad, autoestima e individualidad.
- Enfoque multidimensional. Son muchos rasgos, acciones, conductas y actitudes que gravitan en torno a la relación cotidiana de maestro-alumno en el aula.

Se considera el contexto de la institución, entendido como los propósitos plasmados en su filosofía, misión y visión (Cashin, 1996); las características que definen la gestión académica y administrativa del plan de estudios, tales como: el número de horas asignadas a cada profesor frente a grupo, la cantidad de asignaturas distintas que enseña, el total de alumnos que atiende, la modalidad de enseñanza: presencial, a distancia, taller, prácticas de campo; asimismo, la cultura institucional que permite apreciar el conjunto de normas y valores compartidos por los miembros de la organización, en particular las relacionadas con el trabajo colegiado y colaborativo por ser elementos claves de la cultura institucional (Zabalza, 2002).

Los principios particulares que guían el modelo de evaluación de competencias docentes para educación media y superior y que permiten su

desarrollo (García-Cabrero, Loredo y Carranza, en prensa) son:

- Orientación formativa. El modelo permite al profesor reflexionar y retroalimentar sobre su acción en la docencia y plantear acciones para su mejora.
- Orientación participativa. La evaluación/formación de la práctica docente no es elaborada e instrumentada por las autoridades de una institución educativa, o por un grupo de especialistas ajenos a los profesores, es el docente quien se involucra en el proceso y participa en su diseño.
- Orientación humanista. Considera al docente como una persona, como un ser humano, con preocupaciones, intereses, necesidades, emociones; de ahí que busca la preservación de su dignidad, autoestima e individualidad.
- Enfoque multidimensional. Son muchos rasgos, acciones, conductas y actitudes que gravitan en torno a la relación cotidiana de maestro-alumno en el aula.

Las competencias

Las competencias que los estudiantes tendrán que cumplir son:

.

1. La formación se dirige al desarrollo de cada competencia y a una evaluación por cada competencia.
2. La evaluación toma en cuenta el conocimiento, las actitudes y el desempeño de la competencia como principal fuente de evidencia.
3. El progreso de los alumnos en el programa sigue el ritmo que ellos determinan, según las competencias demostradas.
4. La instrucción es personalizada.
5. Las experiencias de aprendizaje son guiadas por una retroalimentación sistemática.
6. La formación se hace con material que refleja situaciones de trabajo reales y experiencias en el trabajo.
7. El programa en su totalidad es cuidadosamente planeado, y la evaluación sistemática es aplicada para mejorar el programa, es flexible en cuanto a materias obligatorias y las opcionales.
8. La enseñanza debe ser menos dirigida a exponer temas y más al proceso de aprendizaje de los individuos.
9. Hechos, conceptos, principios y otros tipos de conocimiento deben ser parte integrante de las tareas y funciones.

Dentro de los aspectos críticos del enfoque por competencia en el diseño curricular tenemos:

- Posibilidad de transferencia de la norma de competencia.
- Exigencia de la norma.
- Formación personalizada.
- Participación de los trabajadores y de sus organizaciones en la formulación de la norma.
- Autonomía del trabajador.
- Relaciones laborales.
- Heterogeneidad de la norma.
- Aspectos institucionales.
- Riesgo de exclusión social.
- Costo.

La Formación Basada en Competencias presenta varias diferencias con la formación que se clasifica como convencional o tradicional, pero no existe duda de que la diferencia más clara y definitiva es que tiene como referente una competencia y esto obliga a que su diseño curricular se ordene desde el comienzo en torno a su desempeño.

Analizando Resultados

Evaluando el total de Asignaturas de Tecnología Básica y Tecnología Aplicada que componen el Plan de Estudios de la Universidad de Marina Mercante se observa en la Tabla 1:

	Asignaturas del Plan de Estudios de Tecnología Básica y Tecnología Aplicada	Contenidos Evaluados	Competencias Evaluadas	Totales
1	Introducción a la Ingeniería	0	8	8
2	Sistemas y Organizaciones	10	0	10
3	Arquitectura de Computadoras	8	0	8
4	Responsablidad Social Profesional	0	8	8
5	Paradigmas de Computación	3	4	7
6	Sistemas Operativos	4	2	6
7	Sitema de Gestión II	0	18	18
8	Sistema de Gestión I	0	6	6
9	Investigación Operativa	0	8	8
10	Gestión Económica de Proyectos	0	3	3
11	Desarrollo Avanzado de Software	0	8	8
12	Inteligencia Artificial	2	2	4
13	Sistemas industriales informáticos y de control	2	4	6
14	Algoritmos y Estructuras de Datos	0	5	5
15	Sintaxis y Semantica del Lenguaje	2	6	8
16	Analisis de Sistemas I	0	7	7
17	Analisis de Sistemas II	0	8	8
18	Comunicaciones	0	5	5
19	Redes de Información	0	20	20
20	Ingeniería del Software I	4	7	11
21	Diseño de Sistemas I	0	2	2
22	Gestión de Datos	0	18	18
23	Diseño de Sistemas II	0	11	11
24	Ingeniería del Software II	2	1	3
25	Simulación	0	2	2
26	Seguridad Informática	0	6	6
27	Proyecto de Sistemas	0	9	9
28	Proyecto Final de Sistemas	0	15	15
29	Base de Datos	0	20	20
30	Robótica	0	7	7
31	Domótica	0	6	6
	Totales:	37	226	263
	Media	1,19	7,29	
	Desviación Estándar	2,43	5,62	
	Desviación Estandar / Media	2,03	0,77	
		2,03	0,77	

Tabla 1: Tabla de Contenidos Evaluados y Competencias Evaluadas en Asignaturas de Tecnología Básica y Tecnología Aplicada.

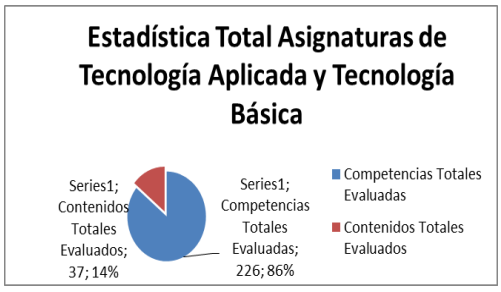


Figura 1: Gráfico de Estadística Total de Asignaturas de Tecnología Aplicada y Tecnología Básica.

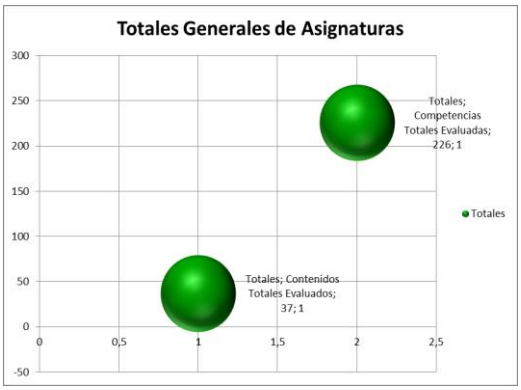


Figura 2: Gráfico de Totales Generales de Asignaturas de Tecnología Básica y Tecnología Aplicada.

Analizando la Figura 1 (Estadística Total de Asignaturas de Tecnología Básica y Tecnología Aplicada) se obtiene como conclusión que sobre un total de 263(doscientos sesenta y tres consignas) se evaluaron competencias en un 86% en 226(doscientas veintiséis consignas).
Se evaluaron contenidos en un 14% en un total de 37(treinta y siete) consignas.
En la tabla 1: (Tabla de Contenidos Evaluados y Competencias Evaluadas en Asignaturas de Tecnología Básica y Tecnología Aplicada) se observa lo siguiente:
Los Contenidos Evaluados obtienen una Media de 1,19.
Los Contenidos Evaluados obtienen una Desviación Estándar de 2,43.
Las Competencias Evaluadas obtienen una Media de 7,29.
Las Competencias Evaluadas obtienen una Desviación Estándar de 5,62.
El Resultado de la división de Desviación Estándar y Media de Contenidos Evaluados es de: 2,03.
El Resultado de la división de Desviación Estándar y Media de Competencias Evaluadas es de: 0,77.
Comparando el Resultado obtenido de Contenidos Evaluados 2,03 y 0,77 de Competencias evaluadas se puede afirmar que 0,77 es menor a 2,03 resultando valida la muestra de los contenidos evaluando competencias.

A continuación se elabora una comparación del Perfil de Egresado teniendo en cuenta las Asignaturas de Tecnología Básica y las Asignaturas de Tecnología Aplicada con el siguiente resultado según la Tabla 2:

	Perfil	Tecnología Básica	Tecnología Aplicada	Total
1	Participar en la toma de decisiones estratégicas de una organización y asesorar, en concordancia con las mismas, acerca de la política de desarrollo de sistemas de información	3		**3**
2	Evaluar, clasificar y seleccionar proyectos de sistemas de información y evaluar y seleccionar alternativas de asistencia externa.	2	1	**3**
3	Planificar, efectuar y evaluar los estudios de factibilidad inherentes a todo proyecto de diseño de sistemas de información y de modificación o reemplazo de los mismos, así como los sistemas de computación asociados.	2	2	**4**
4	Planificar, dirigir, ejecutar y controlar el relevamiento, diseño, desarrollo, implementación y prueba de sistemas de información.	1	9	**10**
5	Analizar, diseñar y aplicar base de datos adecuadas para el control de la información procesada y / o generada por los sistemas de información		2	**2**
6	Evaluar y seleccionar los sistemas de programación disponibles con miras a su utilización en sistemas de información.	1		**1**
7	Evaluar y seleccionar, desde el punto de vista de los sistemas de información, los equipos de procesamiento y comunicación y los sistemas de base.	1	10	**11**
8	Organizar y dirigir el área de sistemas; determinar el perfil de los recursos humanos necesarios y contribuir a su elección y formación.		6	**6**
9	Participar en la elaboración de programas de capacitación para la utilización de sistemas de información.	1		**1**
10	Determinar y controlar el cumplimiento de las pautas técnicas que rigen el funcionamiento y la utilización de recursos informáticos en cada organización.	2	6	**8**
11	Elaborar métodos y normas a seguir en cuestión de seguridad y privacidad de la información procesada y / o generada por los sistemas de información; participar en la determinación de las acciones a seguir en esta materia y evaluar su aplicación.		5	**5**
12	Elaborar métodos y normas a seguir en cuestión de seguridad y privacidad de la información procesada y / o generada por los sistemas de información; participar en la determinación de las acciones a seguir en esta materia y evaluar su aplicación.		5	**5**
13	Elaborar métodos y normas a seguir en el diseño de base de datos en cuestión de seguridad y privacidad de la información procesada y / o generada por los sistemas de información.	1	3	**4**
14	Elaborar métodos y normas a seguir en el diseño de diferentes prototipos aplicados en el ámbito industrial.	4	3	**7**
15	Aplicar lenguajes de programación adecuados para el control de la información procesada y / o generada por los sistemas de información.		2	**2**
16	Aplicar diferentes metodologías en la resolución de casos prácticos en el ámbito profesional.	3	12	**15**

17	Elaborar métodos y normas a seguir en cuestión de calidad de la información procesada y / o generada por los sistemas de información; participar en la determinación de las acciones a seguir en esta materia y evaluar su aplicación.	2	5	**7**
18	Identificar la estructura organizacional de una empresa.	7	3	**10**
19	Elaborar métodos y normas a seguir en cuestión de salvaguarda y control de los recursos físicos y lógicos de un sistema de computación; participar en las acciones a seguir en esta materia y evaluar su aplicación.		5	**5**
20	Desarrollar modelos de simulación, sistemas expertos y otros sistemas informáticos destinados a la resolución de problemas y asesorar en su aplicación.	2	6	**8**
21	Elaborar métodos y normas que permitan resolver situaciones prácticas aplicadas en el ámbito empresarial.	12	19	**31**
22	Realizar estudios e investigaciones conducentes a la creación y mejoramiento de técnicas de desarrollo de sistemas de información y nuevas aplicaciones de la tecnología informática existente.	9	16	25
23	Realizar soluciones integrales aplicadas en el ámbito empresarial.	12	14	26
24	Elaborar la implementación de diferentes tipos de arquitectura en el ámbito profesional.	6	3	9
25	Evaluar, clasificar y seleccionar proyectos de sistemas de información y evaluar y seleccionar alternativas de asistencia externa.		1	1
	Total	**71**	**135**	**206**
	Media	**3,94**	**6,27**	**8,36**
	Desviación Estándar	**3,7**	**5,1**	**8,08**
	Desviación Estándar/Media	**0,94**	**0,81**	
		Muestra Validada		

Tabla 2: Perfil del Egresado que se cumple em Asignaturas de Tecnología Básica y Asignaturas de Tecnología Aplicada.

Realizando el análisis de la Tabla 2 sobre total del perfil de egresado aplicado en todas las Asignaturas de Tecnología Básica y Tecnología Aplicada en las treinta y uno Asignaturas que componen el Plan de Estudios de la Ingeniería en Sistemas de la Universidad de Marina Mercante se obtiene que los diferentes veinticinco perfiles se aplican con un porcentaje del 34% en Asignaturas de Tecnología Básica y el 66% se aplican en Asignaturas de Tecnología Aplicada.
En Tecnología Básica se obtiene una Media de 3,94.
En Tecnología Básica se obtiene una Desviación Estándard 3,70.
En Tecnología Aplicada se obtiene una Media de 6,27
En Tecnología Aplicada se obtiene una Desviación Estándar de 5,10
En el Total de Tecnología Básica y Tecnología Aplicada se obtiene una Desviación Estándar de 8,08.
El Resultado de la división entre Desviación Estándar y Media en Tecnología Básica es 0,94
El Resultado de la división entre Desviación Estándar y Media en Tecnología Aplicada es 0,81
Se puede afirmar que el resultado obtenido en Tecnología Aplicada: 0,81 es menor al resultado obtenido en Tecnología Básica 0,94 quedando validada la muestra.

Análisis Cuantitativo de Finales con Evaluación de Competencias.

	Materia	Año	Alumnos	Desaprobados	%	Aprobados
1	Algoritmos y Estructuras de Datos	1	30	9	**30**	21
2	Introducción a la Ingeniería	1	30	3	10	27
3	Sintaxis y Semantica del Lenguaje	2	25	5	**20**	20
4	Analisis de Sistemas I	2	30	3	10	27
5	Analisis de Sistemas II	2	30	3	10	27
6	Responsablidad Social Profesional	2	30	3	3,43	27
7	Comunicaciones	3	28	4	14,29	24
8	Redes de Información	3	28	4	**14,29**	24
9	Ingeniería del Software I	3	32	2	6,25	30
10	Diseño de Sistemas I	3	30	2	6,67	28
11	Gestión de Datos	3	32	3	9,38	29
12	Diseño de Sistemas II	3	31	3	9,68	28
13	Ingeniería del Software II	4	33	2	6,06	31
14	Simulación	4	27	3	11,11	24
15	Seguridad Informática	4	27	4	**14,81**	23
16	Sitema de Gestión II	4	20	3	5,15	17
17	Sistema de Gestión I	4	20	3	5,15	17
18	Investigación Operativa	4	22	5	4,77	17
19	Gestión Económica de Proyectos	4	28	3	3,68	25
20	Proyecto de Sistemas	5	25	2	8	23
21	Proyecto Final de Sistemas	5	20	0	0	20
22	Base de Datos	5	25	2	8	23
23	Robótica	5	20	2	**10**	18
24	Domótica	5	22	2	9,09	20
	Total		**645**	**75**		**570**
	Media			**3,13**		**23,75**
	Desviación Estándar			**1,65**		**4,28**
	Desviación Estándar/Media			**0,53**		**0,18**
				Muestra Validada		

Tabla 3: Tabla de Análisis Cuantitativo de Finales con Evaluación de Competencias.

Se realiza el análisis de las Asignaturas según el detalle de la Tabla 3 en donde se evalúan en todas las Asignaturas la totalidad de las competencias.
Realizando el análisis de la Evaluación de Competencias en Finales de las Asignaturas de Tecnología Aplicada y Asignaturas de Tecnología Básica se obtiene que fueran aprobados quinientos setenta alumnos y desaprobados sesenta y cinco alumnos en un total seiscientos cuarenta y cinco alumnos de dieciocho materias que componen el plan.
Se observa como materias críticas en Primer Año: Algoritmo y Estructura de Datos con nueve alumnos desaprobados con un porcentaje de 30%.

6. Resultados

Se pueden observar en los gráficos 3 y 4 la dispersión de cada uno de los escenarios:

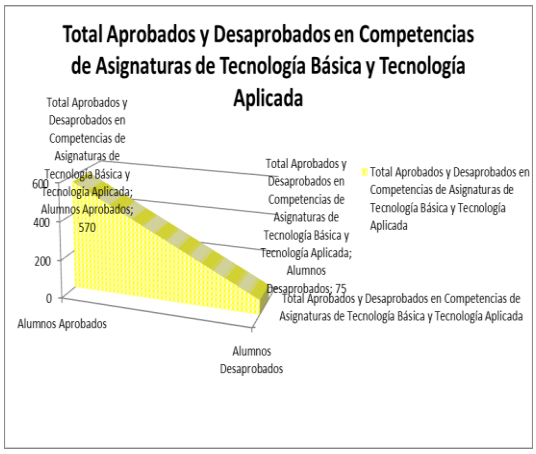


Figura 3: Gráfico de Total de Aprobados y Desaprobados en Competencias en Asignaturas de Tecnología Básica y Tecnología Aplicada.

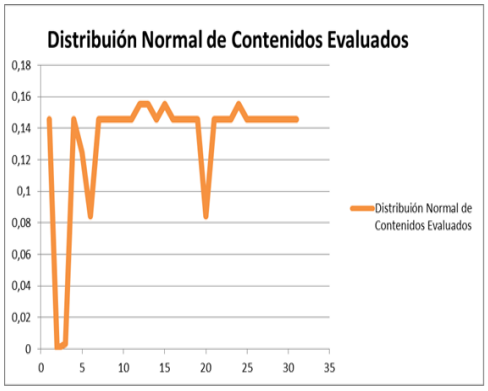


Figura 4: Gráfico de Distribución Normal de Contenidos Evaluados.

Es necesario establecer que para obtener la dispersión se calcula la distribución normal de los contenidos evaluados.
El eje X de la Figura 3 (Gráfico de Total de Aprobados y Desaprobados en Competencias en Asignaturas de Tecnología Básica y Tecnología Aplicada) representa el nombre de las Asignatura y el eje Y representa la dispersión de los contenidos evaluados.
En la Figura 4: (Gráfico de Distribución Normal de Contenidos Evaluados) se observa que en el gráfico se detecta la Asignatura Número 2 (Sistemas y Organizaciones) 0,000228428 con diez contenidos produce una dispersión mayor.
También se observa que la Asignatura 3 (Arquitectura de Computadoras) 0,00322887 con ocho contenidos evaluados produce otra dispersión mayor.
Las dos Asignaturas Sistemas y Organizaciones y Arquitectura de Computadoras son las que tienen mayor dispersión.

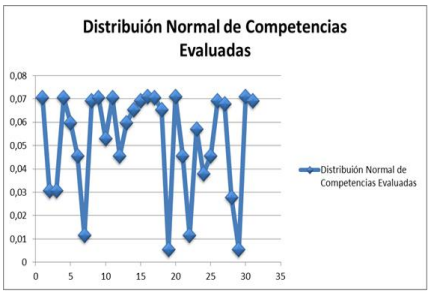


Figura 5: Gráfico de Distribución Normal de Competencias Evaluadas.

En la Figura 5 (Gráfico de Distribución Normal de Competencias Evaluadas) el eje X representa el nombre de las Asignatura y el eje Y representa la dispersión de las competencias evaluadas.
Elaborando un análisis de las competencias evaluadas se observa que en el gráfico se detectan las siguientes Asignaturas donde se evalúan mayor cantidad de competencias:
La Asignatura 29 (Base de Datos) y la Asignatura 19 (Redes de Información) tienen una dispersión de 0,005489218 en veinte competencias evaluadas.
La Asignatura 22 (Gestión de Datos) y la Asignatura 7 (Sistemas de Gestión II) en dieciocho competencias evaluadas tiene una dispersión de 0,011532483.
La Asignatura 28 (Proyecto Final de Sistemas) con quince competencias evaluadas tiene una dispersión de 0,027687865.
La Asignatura 23 (Diseño de Sistemas II) con once competencias evaluadas tiene un dispersión de 0,057109471.
Por último se observa que la Asignatura 4

(Responsabilidad Social Profesional) con ocho competencias evaluadas tienen una dispersión de 0,070464356.

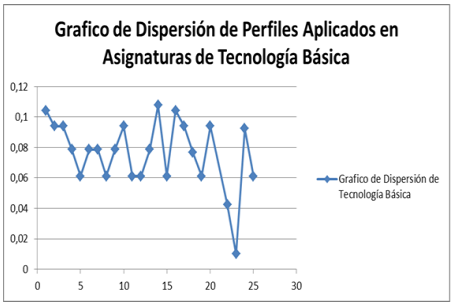


Figura 6: Gráfico de Dispersión de Perfiles Aplicados en Asignaturas de Tecnología Básica.

En la Figura 6: (Gráfico de Dispersión de Perfiles Aplicados en Asignaturas de Tecnología Básica) el eje X representa el nombre de los perfiles y el eje Y representa la dispersión de los perfiles aplicados en Asignaturas de Tecnología Básica.
Se detectan que los siguientes perfiles se contemplan una sola vez con una dispersión de 0,078633172.
4. Planificar, dirigir, ejecutar y controlar el relevamiento, diseño, desarrollo, implementación y prueba de sistemas de información.
6. Evaluar y seleccionar los sistemas de programación disponibles con miras a su utilización en sistemas de información.
7. Evaluar y seleccionar, desde el punto de vista de los sistemas de información, los equipos de procesamiento y comunicación y los sistemas de base.
9. Participar en la elaboración de programas de capacitación para la utilización de sistemas de información.
También se detecta que los perfiles que más de aplican en Asignaturas de Tecnología Básica son:
18. Identificar la estructura organizacional de una empresa en siete Asignaturas.
22. Realizar estudios e investigaciones conducentes a la creación y mejoramiento de técnicas de desarrollo de sistemas de información y nuevas aplicaciones de la tecnología informática existente en nueve Asignaturas.
23. Realizar soluciones integrales aplicadas en el ámbito empresarial en doce Asignaturas.

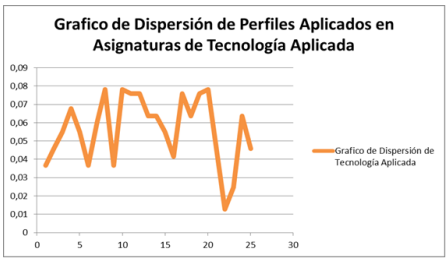


Figura 7: Gráfico de Dispersión de Perfiles Aplicados en Asignaturas de Tecnología Aplicada.

En la Figura 7 (Gráfico de Dispersión de Perfiles Aplicados en Asignaturas de Tecnología Aplicada) el eje X representa el nombre de los perfiles y el eje Y representa la dispersión de los perfiles aplicados en Asignaturas de Tecnología Aplicada.
Se detectan que los siguientes perfiles no se contemplan en ninguna Asignatura de Tecnología Aplicada:
• 1. Participar en la toma de decisiones estratégicas de una organización y asesorar, en concordancia con las mismas, acerca de la política de desarrollo de sistemas de información.
• 9. Participar en la elaboración de programas de capacitación para la utilización de sistemas de información.
• 6. Evaluar y seleccionar los sistemas de programación disponibles con miras a su utilización en sistemas de información.
Los siguientes perfiles se contemplan una sola vez:
• 25. Evaluar, clasificar y seleccionar proyectos de sistemas de información y evaluar y seleccionar alternativas de asistencia externa.
• Evaluar, clasificar y seleccionar proyectos de sistemas de información y evaluar y seleccionar alternativas de asistencia externa.
Los siguientes perfiles se contemplan dos veces:
• Aplicar lenguajes de programación adecuados para el control de la información procesada y / o generada por los sistemas de información.
• Analizar, diseñar y aplicar base de datos adecuadas para el control de la información procesada y / o generada por los sistemas de información.
• Planificar, efectuar y evaluar los estudios de factibilidad inherentes a todo proyecto de diseño de sistemas de información y de modificación o reemplazo de los mismos, así como los sistemas de computación asociados.
Se observa que los perfiles que más se contemplan son:
• 7. Evaluar y seleccionar, desde el punto de vista de los sistemas de información, los equipos

de procesamiento y comunicación y los sistemas de base en diez Asignaturas de Tecnología Aplicada.
• 16. Aplicar diferentes metodologías en la resolución de casos prácticos en el ámbito profesional en doce Asignaturas de Tecnología Aplicada.
• 23. Realizar soluciones integrales aplicadas en el ámbito empresarial en catorce Asignaturas de Tecnología Aplicada.
• 22. Realizar estudios e investigaciones conducentes a la creación y mejoramiento de técnicas de desarrollo de sistemas de información y nuevas aplicaciones de la tecnología informática existente en dieciséis Asignaturas de Tecnología Aplicada.

1. **Conclusiones**

Partiendo de la Hipótesis la aplicación de competencias del plan de estudios de las Asignaturas de Ingeniería en Sistemas tiene relación en la formación del perfil del egresado de la Ingeniería en Sistemas de la Universidad de la Marina Mercante se elaboró un análisis cualitativo y cuantitativo comparando los siguientes conceptos:
- Contenidos Evaluados
- Competencias Evaluadas
- Alcance del Título contemplando los veinticinco items del perfil en las Asignaturas de Tecnología Básica y Tecnología Aplicada que componen el Plan de Estudios de la Ingeniería en Sistemas de la Universidad de la Marina Mercante.
- Cantidad de Alumnos Aprobados en Asignaturas de Tecnología Aplicadas y Tecnología Básica donde se evalúan competencias en su totalidad.

1. Comparando Contenidos Evaluados y Competencias Evaluadas se obtuvo:
Los Contenidos Evaluados obtienen una Media de 1,19.
Los Contenidos Evaluados obtienen una Desviación Estándar de 2,43.
Las Competencias Evaluadas obtienen una Media de 7,29.
Las Competencias Evaluadas obtienen una Desviación Estándar de 5,62.
El Resultado de la división de Desviación Estándar y Media de Contenidos Evaluados es de: 2,03.
El Resultado de la división de Desviación Estándar y Media de Competencias Evaluadas es de: 0,77.
Comparando el Resultado obtenido de Contenidos Evaluados 2,03 y 0,77 de Competencias evaluadas se puede afirmar que 0,77 es menor a 2,03 resultando valida la muestra de los contenidos evaluando competencias.
Analizando el resultado de Contenidos Evaluados y Competencias evaluadas se observa que en treinta y siete consignas de evaluaron contenidos en las Asignaturas de Tecnología Básica y Tecnología Aplicada.
1.1. Análisis de Dispersión de Contenidos Evaluados.
Teniendo en cuenta los tres escenarios analizados con la validación cuantitativa y cualitativa de cada uno de los escenarios con valores demostrados se puede afirmar que en la Ingeniería en Sistemas de la Universidad de la Marina Mercante se aplican Competencias en el Plan de Estudios y tiene estrecha relación con en la formación académica del Perfil del Egresado.
1.2. Análisis de Dispersión de Competencias Evaluadas.
Elaborando un análisis de las competencias evaluadas se observa que en el gráfico se deectan las siguientes Asignaturas donde se evalúan mayor cantidad de competencias:
La Asignatura 29 (Base de Datos) y la Asignatura 19 (Redes de Información) tienen una dispersión de 0,005489218 en veinte competencias evaluadas.
La Asignatura 22 (Gestión de Datos) y la Asignatura 7 (Sistemas de Gestión II) en diecicho competencias evaluadas tiene una dispersión de 0,011532483.
La Asignatura 28 (Proyecto Final de Sistemas) con quince competencias evaluadas tiene una dispersión de 0,027687865.
La Asignatura 23 (Diseño de Sistemas II) con once competencias evaluadas tiene un dispersión de 0,057109471.
Por último se observa que la Asignatura 4 (Responsabilidad Social Profesional) con ocho competencias evaluadas tiene una dispersión de 0,070464356.
El resultado del análisis cualitativo y cuantitativo obtenido en cada una de las Asignaturas de Tecnología Básica, en las Asignaturas de Tecnología Aplicada junto con el Análisis de Dispersión de las Asignaturas de Tecnología Básica y de las Asignaturas de Tecnología Aplicada permite a la Universidad de la Marina Mercante elaborar el control adecuado por intermedio de una matriz F.O.D.A. obteniendo el siguiente Análisis:
Fortalezas:
- Identificación de requerimientos del ámbito industrial.
- Identificación de requerimientos del ámbito comercial.
- Identificación de requerimientos del

ámbito de bienes y servicios.
- Aplicación de competencias en Asignaturas de Tecnología Básica
- Aplicación de competencias en Asignaturas de Tecnología Aplicada.
- Aplicación de competencias en Asignaturas optativas.
- Aplicación de competencias que permiten la inserción en el ámbito laboral.
- Fortalecer estrategias de capacitación para los alumnos de la Universidad de la Marina Mercante para la inserción laboral.

Debilidades:
- Identificación de competencias que se pueden aplicar en Asignaturas de Tecnología Básica.
- Identificación de competencias que se pueden aplicar en Asignaturas de Tecnología Aplicada.
- Optimizar la evaluación de competencias en asignaturas de tecnología aplicada donde se evalúan contenidos.

Oportunidades:
- Aplicación de competencias adquiridas en la formación académica de la Universidad de la Marina Mercante en el ámbito laboral del estudiante y profesional del egresado.

Amenazas:
- Inestabilidad laboral.

El resultado del análisis cualitativo y cuantitativo obtenido en la aplicación del Perfil de las Asignaturas de Tecnología Básica, en las Asignaturas de Tecnología Aplicada junto con el Análisis de Dispersión de las Asignaturas de Tecnología Básica y de las Asignaturas de Tecnología Aplicada se puede observar que existen Perfiles que se aplican en pocas Asignaturas.

Esta debilidad es un instrumento de aporte muy importante para poder corregir y actualizar el perfil del Egresado de los Alumnos de la Universidad de la Marina Mercante, que permite:
- Analizar cuantitativamente los Perfiles de Egresados que se aplican en pocas Asignaturas.
- Analizar cualitativamente los Perfiles de Egresados que se aplican en pocas Asigna-turas.
- Tomar decisiones validadas cuantitativamente y cualitativamente en la futura confección de los Planes de Estudios de la Ingeniería en Sistemas.
- Corregir desviaciones que se presentan en el actual perfil de egresado del alumno de la Universidad de la Marina Mercante.
- Controlar el nuevo Perfil del Egresado del alumno de la Universidad de la Marina Mercante.
- Mejorar continuamente el Perfil del Egresado del alumno de la Universidad de la Marina Mercante.

El resultado del análisis cualitativo y cuantitativo de alumnos Aprobados y Desaprobados obtenido en cada una de las Asignaturas de Tecnología Básica, en las Asignaturas de Tecnología Aplicada junto con el Análisis de Dispersión de Alumnos Aprobados y Desaprobados le permite a la Universidad de la Marina Mercante:

En los alumnos desaprobados
- Identificar cuantitativamente las Asignaturas de Tecnología Básica con índices superiores de desaprobación.
- Identificar a que año pertenecen las Asignaturas donde se encuentra la mayor cantidad de desaprobados.
- Identificar cuantitativamente las Asignaturas de Tecnología Aplicada con índices superiores de desaprobación.
- Elaborar instrumentos pedagógicos para mejorar la adquisición de conocimientos del alumno de la Universidad de la Marina Mercante.
- Elaborar metodologías de aprendizaje que le permitan al alumno el desarrollo de la estructura cognitiva del alumno de la Universidad de la Marina Mercante.
- Elaborar metodologías de aprendizaje que le permitan al alumno la adquisición de un aprendizaje significativo del alumno de la Universidad de la Marina Mercante.
- Establecer Tutorías de seguimiento en las Asignaturas con amplio índice de desaprobación del alumno de la Universidad de la Marina Mercante.
- Mejorar continuamente la aplicación de nuevas metodologías de enseñanza en be-neficio del alumno de la Universidad de la Marina Mercante.

Referencias

Boekaerts, M., Pintrich, P.; Zeidner, M. (Eds.). (2000) Handbook of self –regulation. Nue-va York: AcademicPress.

Cashin, W. E. (1996). "Developing and effectivefacultyevaluationsystem". Idea paper no. 33 Manhattan, Kans.: Center forfacultyEvaluation and facultydevelopment, Kansas, StateUniversity

Consejo Federal de Decanos de Ingeniería (CONFEDI). (2007). Competencias Genéricas. Desarrollo de competencias en la enseñanza de la ingeniería argentina. Universidad Nacional de San Juan. San Juan.

García-Cabrero, B., Loredo, J., Carranza, G. (en prensa). "Pensamiento-interacción-reflexión: propuesta de un modelo para analizar la práctica educativa". Revista Electrónica de

Investigación Educativa.
Hall, K. & Burke, W. (2003). Making formative assessment work - Effective practice in the primary classroom.Maidenhead, UK: Open UniversityPress.
Le Boterf G.(2000) Ingeniería de las Competencias. Barcelona. Ed. Gestión.
McDonald, R,;Boud, D.; Francis, J. y Gonczi, A. (2000). Nuevas perspectivas sobre la evaluaciónBoletín Cinterfor14941-72
Perrenoud, P. (2004). Desarrollar la práctica reflexiva en el oficio de enseñar. Barcelona: Graó.
Perrenoud P.(2007) "Diez Nuevas Competencias para enseñar" Grao, Barcelona.
Spencer, L.M. & Spencer, S.M. (1993). Competence at Work. New York : John Wiley and Sons.
Zabalza, M. (2003). Competencias docentes del profesorado. Calidad y desarrollo profesional. Madrid. Narcea.

La importancia de las TIC's en las instancias de Evaluación: Análisis de una experiencia educativa

Grossi, María Delia
Departamento de Computación
Facultad de Ingeniería, Universidad de Buenos Aires
mdg7501@yahoo.com.ar

Abstract

En este trabajo se presenta una experiencia educativa específica que incluye a las Tecnologías de Información y Comunicación (TIC's) en el proceso de evaluación de una asignatura de educación superior.

Se focaliza la mirada en dos instrumentos de evaluación de la Web 2.0: el portfolio y la rúbrica. Y en la integración de la instancia de evaluación al proceso de enseñanza y aprendizaje, posibilitando al estudiante la reflexión acerca de los contenidos abordados y la reconstrucción del error.

1. Introducción

Las Tecnologías de la Información y la Comunicación (TIC's) se han ido incorporando de manera progresiva en las actividades y en las intervenciones didácticas de modalidad presencial y a distancia. Se las utiliza como herramientas que favorecen la mediación del contenido didáctico y como espacios de comunicación y colaboración entre docentes y alumnos, y entre alumnos. También como entornos de evaluación y en particular, de autoevaluación, modalidad que habitualmente se implementa en el marco de la eficiencia. Bajo esta concepción, la evaluación se concibe desde el planteo tecnológico y no desde el pedagógico.

Adhiriendo a concepciones que consideran que la evaluación debe acompañar las transformaciones de las prácticas de aprendizaje [1], en este trabajo se propone la utilización del portfolio y de la rúbrica como recursos para evaluar y promover el aprendizaje de los estudiantes.

2. Instrumentos de evaluación de la Web 2.0

La Web 2.0 comprende a las herramientas de software que conforman el software social y favorecen a la construcción colectiva del conocimiento [2]. Dentro de los recursos ofrecidos por las nuevas tecnologías, se seleccionaron dos para implementar una instancia de evaluación formativa.

2.1. El portfolio

De acuerdo a la definición de Elena Barberá y otros [3], "el portfolio es un sistema de evaluación integrado en el proceso de enseñanza y aprendizaje".

El estudiante, con un objetivo concreto, selecciona las distintas evidencias elaboradas, abordadas y generadas en la práctica educativa. Estas evidencias pueden consistir tanto en trabajos desarrollados para el curso, en fragmentos de fuentes de información propuestas por el docente y complementarias, como en reflexiones acerca del proceso que atraviesa el alumno para apropiarse del contenido.

El portfolio requiere un tiempo de elaboración considerable (puede iniciarse al comenzar el proceso de enseñanza y aprendizaje), y de la guía del docente para su confección. Siguiendo a Anijovich y otros [4], "el proceso de evaluación de los trabajos que contiene el portfolio deberá facilitar que el alumno por sí mismo o guiado por las intervenciones didácticas del docente, halle evidencias de comprensión de los contenidos que aprende, del tipo de estrategia que utiliza para resolver las diferentes tareas, de sus fortalezas, de sus aspectos a mejorar, del uso de su tiempo, de la organización de su tarea, entre otros".

2.2. La Rúbrica

La rúbrica es una herramienta de evaluación que permite externalizar el conjunto de criterios y de descriptores que utiliza el docente para evaluar.

Anijovich y otros [4], la entienden como "una herramienta de evaluación que selecciona criterios a tener

en cuenta para la elaboración de un trabajo y muestra los niveles de calidad posibles para cada uno de estos criterios".

Se la representa a través de una tabla de doble entrada, en la cual las columnas indican las dimensiones de calidad, y las filas los criterios a evaluar.

Algunas de las características de esta herramienta son:

- Para el estudiante: ayuda a que éste comience a asumir la responsabilidad de su aprendizaje, orientando su actividad y favoreciendo a mejorar su desempeño [4].
- Para el docente: en la etapa de diseño, facilita la reflexión y planificación de las unidades temáticas, en tanto que en la etapa de implementación, evita la corrección basada en la intuición, favoreciendo a la retroalimentación, que se basa en los descriptores incluidos en la rúbrica.

3. Descripción de la experiencia

3.1. Acerca del contexto

Habitualmente se observa que los estudiantes de las asignaturas introductorias a las ciencias de la Computación presentan dificultades para relacionar los conceptos abordados. Una vez que, a través de la ejercitación adquieren habilidades que les permiten utilizar las herramientas de un lenguaje de programación de forma apropiada, muchas veces manifiestan problemas de comprensión.

Acerca de la comprensión, Perkins y Bythe [5], la conciben como "el poder realizar una gama de actividades que requieren pensamiento respecto a un tema, por ejemplo, explicarlo, generalizarlo, presentar analogías y representarlo de una manera nueva".

La experiencia procura que el estudiante aprenda para la comprensión. Con esta finalidad, a partir de los objetivos del curso, se propone a los alumnos una actividad integradora que les permita explorar, relacionar, sintetizar, explicar, generar y aplicar el conocimiento.

3.2. Acerca de las consignas del trabajo

Elaborar un portfolio que integre los conocimientos adquiridos durante el curso partiendo de los objetivos del mismo:

- Procurar que el alumno se compenetre con la computadora como herramienta de trabajo, sabiendo su precisión, capacidad y limitaciones.
- Enseñar el análisis, la sistematización, programación y procesamiento de distintos problemas de tipo técnico-científico.
- Conferir una visión global de la Computación.
- Focalizar la mirada en la algoritmia como paradigma de resolución de problemas.

Las pautas acerca de la organización son:

- Presentación del alumno.
- Índice.

Y los temas que como mínimo se deben abordar:

- La computadora como herramienta de trabajo: descripción de las características del hardware y del software que el alumno utiliza habitualmente. Incluir un detalle sobre otras herramientas tecnológicas que utilice para el estudio y para el ocio.
- Sistematización, programación y procesamiento de distintos problemas de tipo técnico-científico: a partir de los conocimientos sobre paradigmas de programación y compiladores, junto a la experiencia adquirida en el uso de un lenguaje de programación de alto nivel y de otro de bajo nivel que corresponde a una máquina genérica, vincular conceptos y reflexionar sobre los mismos.
- Visión Global de la Computación: reflexión sobre el área de la Computación que más le interesa al alumno, e investigación sobre las aplicaciones de software y usos de la computación en el ejercicio del campo profesional.
- Focalizar la mirada en la algoritmia como paradigma de resolución de problemas: en el curso se plantearon problemas y se resolvieron de manera metodológica. A modo de síntesis, se pide la elaboración del algoritmo que le permitió al alumno retener, comprender y hacer uso del conocimiento desarrollado en el curso. Se lo puede expresar en lenguaje natural o de programación.

Se trata de una actividad individual, el recurso tecnológico elegido es Sites de Google. El tiempo asignado es de un mes. Los alumnos, luego de haber presentado un primer avance del trabajo al docente, lo comparten al grupo.

3.3. Criterio de evaluación del portfolio

Junto con las consignas de la actividad se externalizaron los criterios de evaluación a través de la publicación de la rúbrica. A continuación, en la Tabla 1, se pueden observar los criterios empleados.

Tabla 1: Rúbrica implementada

Categoría	Excelente	Muy Bueno	Bueno	Aprobado	Insuficiente
Conceptos sobre programación	Se manifiesta comprensión y se hace un correcto uso de todas las herramientas de programación abordadas.	Se logra identificar las herramientas de programación abordadas. Se las vincula y se reflexiona sobre las mismas. Se aportan ejemplos pertinentes.	Se logra identificar algunas de las herramientas de programación abordadas. Distingue su implementación en los lenguajes utilizados. Se las ilustra con ejemplos pertinentes.	Se logra identificar unas pocas herramientas de programación abordadas y distingue su implementación en los lenguajes utilizados.	No se denota comprensión de los conceptos de programación abordados.
Conceptos sobre hardware y software	Se denota comprensión vinculando la descripción de las herramientas tecnológicas de trabajo y de ocio con el marco teórico. Se aportan nuevas referencias teóricas pertinentes.	Se denota comprensión vinculando la descripción de las herramientas tecnológicas de trabajo y de ocio con el marco teórico.	Se describen las herramientas tecnológicas de trabajo y de ocio y se vincula algunas de ellas con el marco teórico.	Se enumeran las herramientas tecnológicas de trabajo y de ocio y se vinculan algunas de ellas con el marco teórico.	Se enumeran las herramientas tecnológicas de trabajo y de ocio, pero no se relacionan con el marco teórico.
Paradigmas de programación	Se vinculan los conceptos abordados con los lenguajes con los que se trabajó. Logra caracterizar a los distintos paradigmas aportando ejemplos y nuevas referencias teóricas.	Se vinculan los conceptos abordados con los lenguajes con los que se trabajó. Logra caracterizar a los distintos paradigmas aportando ejemplos.	Se vinculan los conceptos abordados con los lenguajes con los que se trabajó. Se mencionan los distintos paradigmas, distinguiendo algunas características de los mismos.	Se vinculan los conceptos abordados con los lenguajes con los que se trabajó. Se mencionan los distintos paradigmas sin reflexionar sobre éstos y sin caracterizarlos.	No se logra vincular los conceptos abordados con los lenguajes con los que se trabajó.
Concepto de algoritmia	Se realiza una revisión de lo actuado, logrando expresar con claridad a través de un algoritmo, el proceso de aprendizaje seguido. Se utiliza en el algoritmo una variedad de herramientas de programación vistas en el curso, identificando con claridad el objeto de aprendizaje. Se vincula con el marco teórico.	Se expresa a través de un algoritmo el proceso de aprendizaje. identificando los obstáculos conceptuales y procedimentales e indicando los pasos seguidos para superarlos. Se desarrolla el algoritmo haciendo uso de algunas herramientas de programación vistas en el curso.	Se logra expresar a través de un algoritmo el proceso de aprendizaje. Se identifican los obstáculos procedimentales. Se desarrolla el algoritmo haciendo uso de algunas herramientas de programación.	Se expresa a través de un algoritmo el proceso de aprendizaje. Se tiene dificultad para identificar el objeto de aprendizaje. El algoritmo es básicamente secuencial.	No se logra hacer una revisión de lo actuado a través de un algoritmo. No se logra identificar el objeto de aprendizaje.

4. Resultados

Participaron de la actividad un total de veinticinco alumnos. La experiencia en general ha sido muy alentadora porque permitió al alumno analizar sus propios procesos de aprendizaje y relacionar los conocimientos abordados. También le permitió reflexionar e investigar acerca de la aplicación de la tecnología en la carrera elegida.

En las primeras versiones se observaron dificultades para:

- Vincular el marco teórico con el uso que el alumno hace de la tecnología.
- Expresarse en forma escrita. Por ejemplo, un estudiante que en otras instancias de evaluación demostró haber adquirido las habilidades necesarias para diseñar e implementar un programa en un lenguaje de programación afirmó: "El lenguaje es un programa que me permite armar una secuencia de pasos ordenados que se van a llevar a cabo uno luego del otro e ir pidiendo e informando al usuario datos con los cuales se van a realizar ciertas instrucciones especificadas previamente".
- Vincular herramientas de distintos lenguajes de programación, por ejemplo, aquellas que permiten implementar ciclos repetitivos.
- Diseñar el algoritmo que permitió al alumno incorporar el contenido del curso.

Las devoluciones personalizadas del primer avance permitieron al docente señalar los aciertos, corregir malas interpretaciones y orientar a los estudiantes sobre los objetivos de la tarea.

Continuar con la elaboración del portfolio, teniendo como referencia las producciones de los compañeros, les permitió avanzar con nuevas guías.

La actividad brindó un espacio para desestructurar a los alumnos y animarlos a tener una actitud más creativa. La generación de un ámbito donde pudieron explorar e investigar las áreas de aplicación de la informática en la profesión elegida, permitió mejorar la motivación. Algunos alumnos se animaron a implementar el algoritmo de aprendizaje en un lenguaje de programación, desarrollando el enunciado del problema y las fases de análisis, algorítmica, de programación y de evaluación [6].

Los resultados cuantitativos de la evaluación de la actividad basados en la rúbrica se observan en la Figura 1.

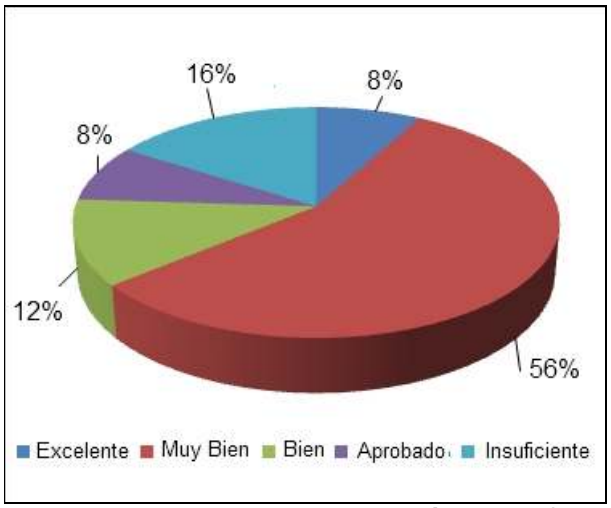


Figura 1: Resultado de la evaluación del portfolio

5. Conclusiones

En este artículo se ha presentado una experiencia de incorporación de las TIC's en el proceso de evaluación de una asignatura de educación superior.

Asimismo, se han caracterizado los instrumentos de evaluación de la Web 2.0 seleccionados.

La generación de un ámbito colectivo donde los estudiantes pudieran socializar el proceso de aprendizaje realizado a lo largo del curso, permitió al docente evaluar las decisiones de la enseñanza efectuadas.

La rúbrica permitió transparentar los criterios de evaluación y definir los aspectos que se consideraron relevantes evaluar con el objetivo final de que el alumno aprenda para la comprensión. Acerca de las rúbricas, Heidi Andrade [7] sostiene que ayudan a los estudiantes a ser jueces más reflexivos sobre la calidad de su propio trabajo, así como sobre el trabajo de otros.

Se pudo observar que la realización del portfolio permitió a algunos alumnos evidenciar errores conceptuales que pudieron subsanar y evitar la repetición de los mismos en la posterior instancia de evaluación.

6. References

[1] Camilloni, A., *Sistemas de calificación y regímenes de promoción*. En Camilloni, A., Celman, S., Litwin, E., Palou de Maté, M.: La evaluación de los aprendizajes en el debate didáctico contemporáneo, Ed. Paidós, 1998.

[2] O'Reilly, T, *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.* International Journal of Digital Economics No. 65, March 2007, pp. 17-37.

[3] Barberá, E., Bautista, G., Espasa, A., Gasch, T., *SPortfolio electrónico: desarrollo de competencias profesionales en la red*, Revista de Universidad y Sociedad del Conocimiento. Vol 3 - N.2, Octubre de 2004, pp. 56-66.

[4] Anijovich, R., Malbergier, Mirta, Sigal, C. Una Introducción a la Enseñanza para la Diversidad. Fondo de Cultura Económica, 2004.

[5] Perkins, D., Blythe, T., *Ante todo la Comprensión*, Educational Leadership 51 (5), 4-7. Traducido al español por Patricia León Agustí y María Ximena Barrera. Junio de 2006. Disponible en: http://www.eduteka.org/AnteTodoComprension.php

[6] Broookshear, G. Introducción a las ciencias de la computación, Addison-Wesley Iberoamericana, 1995.

[7] Andrade, H. *Comprendiendo las rúbricas*, Traducido al español por Ana Atorresi. Enunciación Vol 15, Núm. 1, enero junio de 2010, pp. 157-163.

La educación problemática o la problemática de la educación

Andrés Bursztyn
Universidad Tecnológica Nacional Regional Buenos Aires
andres.bursztyn@gmail.com

Santiago Ferreiros Cabrera
Universidad Tecnológica Nacional Regional Buenos Aires
santiago.ferreiros@gmail.com

Susana Alcira Granado Peralta
Universidad Tecnológica Nacional Regional Buenos Aires
sgperalta@gmail.com

Marcela Guerrero
Universidad Tecnológica Nacional Regional Buenos Aires
marceaguerrero@gmail.com

Alejandra Laura Isola
Universidad Tecnológica Nacional Regional Buenos Aires
ail@mrecic.gov.ar

Marcela Eugenia Portillo
Universidad Tecnológica Nacional Regional Buenos Aires
mportillo@frba.utn.edu.ar

Alejandro Prince
Universidad Tecnológica Nacional Regional Buenos Aires
aprince@princeconsulting.biz

Abstract

Este trabajo tiene por objetivo central reflexionar acerca de la complejidad que presenta la acción de formar a los futuros profesionales en la sociedad contemporánea en la que muchos valores, entre ellos la educación, están en crisis. Su importancia es contribuir a que a la educación, en particular en ingeniería, sea considerada como una variable multidimensional inmersa en un mundo cambiante que debe respetar las formas socio-institucionales y responder a objetivos político-nacionales. Entre los objetivos particulares nos planteamos la necesidad de tomar posición respecto al cómo, el qué y el porqué enseña.

1. Introduction

En el sistema educativo la relación entre educando y educador se da a través del currículo. Alicia de Alba, según palabras de Adriana Puiggrós [1] inició la desconstrucción del discurso curricular vigente hasta los 90, al reflexionar sobre el currículo, la modernidad y la postmodernidad.

Henry Giroux-P. Mc Laren, [2] al analizar las relaciones entre currículo, cultura y poder, consideraron necesario encontrar una pedagogía que inicie a los estudiantes en el conocimiento de sí mismos y que favorezca la educación como formadora de ciudadanos críticos y activos. Acordamos con H. Giraux [2] cuando señala: "Un aspecto importante para empezar a desarrollar un

discurso crítico de la educación sería concebir el currículo en relación con las formas de conocimiento y las prácticas sociales que legitiman y reproducen formas particulares de cultura y vida social. En otras palabras, el currículo debería ser visto como la representación de un proceso de selección y exclusión; y en este sentido, como una expresión de lucha en relación a las formas de autoridad política, de representatividad, de regulación moral, y de las versiones del pasado y el futuro que deben ser legitimadas, transmitidas y debatidas en contextos pedagógicos específicos".

Una cuestión para tener en cuenta es que muchas veces la sociedad se muestra apática respecto de los acontecimientos cotidianos, ignora que forma parte de una red de poder, del poder cotidiano. Este comportamiento colectivo está usufructuado por la esfera de poder (gobierna sin debatir) y es una consecuencia de la indiferencia de los educadores progresistas quienes no cuestionan las implicancias sociales, culturales y éticas de dicha actitud. [3]

El currículo se constituye como el proyecto social, cultural y educativo de un país, representa los intereses de la sociedad. ES dinámico y plasma las políticas educativas de la nación. Es un contrato entre la sociedad y el estado, quien debe proveer los medios para garantizar su cumplimiento.

Mc Laren [3]se pregunta: "Qué formas curriculares entrevemos que nos hagan capaces de articular nuestra visión de un futuro libre de pena y sufrimiento?".....Entendemos que es imposible basar la pedagogía –ni siquiera una forma política- en una noción de verdad conceptual. La pedagogía debe ser desarrollada en una realidad referencial contra la cual podamos mediar nuestra acciones presentes y futuras."

Según C. Cullen [4] "un currículum implica una toma de posición didáctica frente al conocimiento y los valores y una contextualización institucional de la didáctica propuesta.

Diaz Barriga [5], respecto del concepto de currículo, dice "…. es objeto de una infinidad de adjetivos y por supuesto de una enormidad de significados…… el concepto está en proceso de disolución como resultado de una polisemia que lo acompaña……. De esta manera encontramos la permanente necesidad de dotar al término de un adjetivo que ayude a su conceptualización, y no negamos que en algunos casos efectivamente haya logrado enriquecer un matiz significativo y relevante en la construcción de significados; por ejemplo: oculto, formal, vivido, procesual. Todos estos adjetivos de alguna forma permiten comprender un sentido de lo curricular que rebasa a quienes pretenden "definir" lo curricular" .Distingue, además, los conceptos currículo (plan de formación para desempeñarse en la vida) y campo curricular (disciplina en el área de las ciencias de la educación).

Paulo Freire [6] centra su preocupación en la formación de un ciudadano libre, emancipado, reflexivo, critico y político. Ve al docente como un investigador en el aula, preocupado y comprometido con las cuestiones socio-políticas. Sostiene que la educación y el currículo no acontecen en un vacío cultural, religioso, político e ideológico. Por el contrario se interrelacionan en la complejidad social y temporal en la que se desenvuelven. Considera que el currículo es un acto político y el docente el agente de cambio para lograr una sociedad más justa, democrática y participativa.

El campo del currículo es complejo y obliga a una reconceptualización y a una profunda reflexión acerca de que si la complejidad en la que está inmerso facilita la comprensión de los temas que le son inherentes o si se constituye como un obstáculo para su análisis.[1]

Alicia De Alba [1] considera que es necesario asumir la crisis en la que se encuentra el campo del currículo y para comprenderla, desde cada país, se debe hacer una lectura política y no académica.

Observa que la crisis del campo curricular trae aparejada ausencia:

a) del análisis del problema formativo sobre la cuestión curricular
b) del análisis epistemológico en educación
c) del análisis de currículos particulares
d) de reflexión de la dimensión metodológica desde la postura crítica.

Dice que tal situación puede orientarse de dos modos. Por un lado hacia la reproducción mecanicista, que no soluciona los problemas centrales. Por otro a una profunda transformación de las relaciones entre práctica y discurso, que atienda las particularidades de cada país. Considera central la formación de equipos en las instituciones educativas, incorporar en la agenda el análisis de las ciencias de la educación y tener en cuenta a teóricos tales como Giraux, Apple y Mc Lauren. Destaca los pensamientos curriculares liberadores que responden a las necesidades sociales y la construcción del campo curricular que concibe como intervención institucional y al mismo tiempo como un campo de estudio específico.

2. El currículo en la educación universitaria

Con respecto a la educación superior universitaria, De Alba [1] distingue la formación teórica básica de la formación crítico social (que permitirá al graduado comprender el papel de su profesión en el imaginario social). Destaca, también, la necesidad de una fuerte formación tecnológico-práctica. Sostiene que el

[1] El debate del currículo se da tanto en el currículo como práctica social, sino también en la diferencia entre currículo formal, vivido y oculto [1](pág. 57)

antagonismo entre las posturas técnica y crítica ha generado una crisis del currículo y enfatiza la importancia de considerarlo en cada una de sus dimensiones y aspectos relevantes. La postura técnica está centrada en los contenidos. La postura crítica, por el contrario, coloca su eje en el carácter político ideológico y sociocultural de toda propuesta curricular.

Finalmente Alicia De Alba [1] da la siguiente noción de currículo "Por curriculum se entiende a la síntesis de elementos culturales (conocimientos, valores, costumbres, creencias, hábitos) que conforman una propuesta político-educativa pensada e impulsada por diversos grupos y sectores sociales cuyos intereses son diversos y contradictorios, aunque algunos tienden a ser dominantes o hegemónicos, y otros tienden a oponerse y resistirse a tal dominación o hegemonía. Síntesis a la cual se arriba a través de diversos mecanismos de negociación e imposición social. Propuesta conformada por aspectos estructurales formales y procesales prácticos, así como por dimensiones generales y particulares que interactúan en el devenir de los currícula en las instituciones sociales educativas. Devenir curricular cuyo carácter es profundamente histórico y no mecánico y lineal. Estructura y devenir que conforman y expresan a través de distintos niveles de significación"

La síntesis de los elementos culturales a los que refiere (hábitos, costumbres, etc.) son incorporados formalmente o a través de las relaciones sociales. Representan el carácter de lucha que se desarrolla tanto en la conformación como en su desarrollo y evaluación. Algo para tener en cuenta es que el carácter político, al que refiere, no es política partidaria sino la respuesta de la educación a un proyecto político contextualizado en tiempo y espacio. La puesta en marcha de un determinado currículo está consensuada por el grupo hegemónico y el resto de la sociedad.

No se puede ignorar la función social de la educación y, desde ese lugar, la lucha entre los grupos dominantes y los otros es un tema analizado por la sociología de la educación. A la síntesis se arriba a través de imposiciones y negociaciones en el centro del poder. Los aspectos estructurales y el carácter histórico nombrado en el concepto dado por De Alba se refieren al desarrollo del currículo. Con referencia a los niveles de significación o concreción acuerda con Derrida [7] quien sostiene que hay datos que deben ser descritos en términos de estructura, otros en términos de génesis. Hay capas de significaciones que aparecen como sistemas, o como configuraciones estáticas. Sin embargo en su interior son posibles el movimiento y la génesis, condicionada a la legalidad propia y a la significación funcional de la estructura.

Se observa que a pesar de los distintos posicionamientos respecto de los autores citados, coinciden en afirmar que el currículo es un proyecto político que involucra prácticas para alcanzar metas sociales. Que implica la decisión de seleccionar los contenidos a enseñar, teniendo en cuenta aquellas cuestiones que toda la sociedad considera valioso. Valor que, usualmente, en los países democráticos es "interpretado" por el grupo de hegemónico o de poder y , negociado con el resto de la sociedad. De dicha negociación debería surgir, como efecto deseado, la síntesis que constituye el proyecto curricular.

3. El currículo y sus dimensiones

Entendemos por dimensión curricular a la característica fundamental que define el significado de la formación. Que impone alcances y límites, y responde a un proyecto socio-político en un tiempo y espacio determinados. De Alba distingue dimensiones generales y específicas interrelacionadas. [1]

Las dimensiones generales forman parte de todos los currículos y se concretan en la institución y en las prácticas educativas.

Dentro de las dimensiones generales se refiere a:

a) la dimensión política que determina la posibilidad de su implementación,
b) la dimensión cultural con rasgos del grupo hegemónico o dominante y características de los grupos dominados que dejan entrever cierta expresión de resistencia,
c) la dimensión social atada a las concepciones socio políticas imperantes en ese tiempo y espacio,
d) la dimensión económica ligada con la determinación económica.
e) las dimensiones ideológica e institucional que impactan en su implementación. Están tan interrelacionadas que no es posible pensar en una sin referirse a la otra,
f) la dimensión didáctica, como puesta en marcha del proyecto. Claramente en correspondencia con las dimensiones ideológica e institucional. Hay que tener en cuenta que la cultura institucional responde a un proyecto y éste a una ideología que, de alguna forma, condicionan las actividades en el aula.

Las dimensiones particulares o específicas son propias de cada currículo, se nutren de las dimensiones generales pero tienen su impronta en cuanto al nivel de desarrollo, a los objetivos según el tipo de educación, etc.

Algo para tener en cuenta es que las dimensiones, que constituyen las características de los aspectos esenciales y los límites de cada proceso no son compartimientos estancos. Están interrelacionadas y de la comprensión de esa interrelación se logra o no percibir el espíritu del currículo.

4. El currículo universitario: contexto y sujetos

A partir de los noventa las universidades públicas en nuestro país las se vieron afectadas por distintas cuestiones. Entre ellas políticas de corte neoliberal, falta de presupuesto, promulgación de la ley de Educación Superior, creación de la CONEAU[2] y la SPU[3]. Al mismo tiempo el estado dejó de ser "benefactor" y el saber universitario pasó a ser "conocimiento mercantilista" que es medido a través de indicadores cuantitativos. Al respecto M. Mollis [8] dice: "Nuestras universidades tienen alterada su identidad como institución de los saberes hacia la construcción de una nueva identidad que las asemeja al supermercado donde el estudiante es cliente, los saberes una mercancía y el profesor un asalariado enseñante".

Observa, además, desarticulación con el nivel medio. El nivel superior universitario y no universitario se caracterizó por su heterogeneidad y superposición de ofertas educativas, muchas veces con dudoso nivel académico. Esa situación no surge por la diversidad ni por la masividad sino por la falta de estructura de planificación y de articulación entre los niveles educativos involucrados. [9][4].

En esa época en algunas universidades públicas y estatales como UTN [5]y UB[6] se produjeron cambios en los planes de estudio, en particular en las carreras de ingeniería presionados, tal vez, por la CONEAU.

Se observa, además, que hay cosas que se han perdido en la relación sociedad-universidad. En efecto, tradicionalmente la universidad transmitió la cultura, hacía una crítica social constructiva y promovía el conocimiento. La profesionalización actual de la universidad impactó negativamente en esa dirección y creemos se deben tomar las medidas pertinentes para revertir la situación.

Por lo expresado se constituye como tema central, en el campo del currículo, la formación que recibirán nuestros universitarios que puede priorizar las demandas del mercado ó favorecer una profunda formación teórica. Es ésta una decisión difícil que requiere un importante debate, entre todos los actores interesados en el análisis de la problemática curricular, en el contexto socio-económico actual.

De Alba [1] distingue como actores involucrados:

a) a los sujetos de la determinación curricular que dan los lineamientos. Entre ellos podemos el estado, los partidos políticos, la iglesia, las instituciones,
b) a los sujetos del proceso de estructuración formal del currículo que son quienes les dan forma y estructura teniendo en cuenta los lineamientos generales,
c) a los sujetos del desarrollo curricular que son quienes lo ponen en práctica poniendo cada día en juego su impronta, proyectos y creencias.

Pensamos que la dinámica de la sociedad va produciendo un devenir de reestructuración social que no puede ni debe ser ignorado. Esta cuestión condiciona la conformación del currículo que tiene que ser lo suficientemente flexible para acompañar a la sociedad en acomodamientos y al mismo tiempo fuerte para no abandonar sus aspectos centrales.

De acuerdo a lo considerado hasta ahora hay dos temas que creemos que no pueden faltar en la agenda de discusión del currículo universitario: los saberes seleccionados y la función social de la universidad.

Respecto a los contenidos y siguiendo a Alicia De Alba [1] los campos del currículo deberían ser los siguientes:

a) Epistemológico –teórico que permite la reflexión epistemológica en cada disciplina.
b) Crítico –Social para comprender tanto el papel social de la profesión como de las disciplinas que la involucra.
c) Científico-Tecnológico que favorece la adquisición temprana de los nuevos conocimientos y tecnología.
d) Incorporación de prácticas profesionales para poner al estudiante en contacto con los problemas de su profesión.

Con referencia al papel de la universidad en la formación de profesionales críticos y reflexivos, sin duda es el campo social el que da cuenta de esa formación. Esa cuestión condiciona a que la problemática ambiental, los derechos humanos, la pobreza y la vida en democracia son temas que no deberían estar ausentes un currículo. Se constituyen como los problemas que están en la agenda cotidiana en la actualidad.

Creemos que la capacidad crítica es un aporte fundamental de la educación universitaria que impulsa el cambio de dirección cuando se estima necesario. Por lo tanto es necesario definir una política universitaria referida al saber, la pregunta sería ¿qué necesitamos saber para intentar solucionar los miles de problemas que, cómo sociedad tenemos?

[2] Comisión Nacional de Evaluación y Acreditación Universitaria
[3] Secretaría de Políticas Universitarias
[4] Para profundizar el tema ver M. Mollis, Memorias de la Universidad, 2009,capítulo 8
[5] Nos referiremos luego a ese cambio en UTN(Universidad Tecnológica Nacional)
[6] Universidad de Belgrano

5. Diseño curricular vigente en la Universidad Tecnológica Nacional

En este apartado haremos una breve referencia al Diseño Curricular implementado en la Universidad Tecnológica Nacional en la carrera Ingeniería en Sistemas de Información, en el año 1995 modificado en 2008 y actualmente vigente.

Entendemos que diseño curricular es metodología, acción y resultado. Metodología porque explica cómo elaborar la concepción curricular. Acción por cuanto constituye un proceso de elaboración. Resultado porque de dicho proceso quedan plasmados, en documentos curriculares, la concepción y las acciones para ponerla en práctica y evaluarla.

Hasta el año 1994 en UTN, para cada especialidad de ingeniería había un plan de estudios, que fue modificándose en algunas carreras, sin presentar novedades respecto de los anteriores salvo por alguna modificación de programas o de correlativas.

Entre los años 1994 y 1995 se pone en marcha un nuevo plan de estudios para todas las carreras de ingeniería de la oferta educativa que posee.

La novedad respecto de los anteriores es que presenta y desarrolla cada uno de los siguientes ítems.

a) Fundamentación
b) Perfil profesional
c) Incumbencias profesionales
d) Objetivos generales
e) Estructura general
    1. Diseño Curricular
    2. Tronco Integrador
    3. Asignaturas Electivas
    4. Salida Intermedia
    5. Idioma
f) Metodología de la enseñanza
    1. Metodología pedagógica
    2. Evaluación
g) Organización de la carrera
    1. Duración
    2. Áreas
h) Plan de estudio
i) Régimen de correlatividades
j) Programas sintéticos
k) Régimen de equivalencias
l) Régimen de homologación

Se destaca en la fundamentación, la necesidad de actualizar los criterios para la formación del ingeniero, de aumentar la motivación de la comunidad educativa, de facilitar la inserción laboral del egresado y de formar un ingeniero creativo capaz de generar cambios.

Con respecto al perfil profesional: "Resumiendo, la preparación integral recibida en materias técnicas y humanísticas lo ubican en una posición relevante en un medio donde la sociedad demandará cada vez más al ingeniero un gran compromiso con la preservación del medio ambiente, el mejoramiento de la calidad de vida en general y una gran responsabilidad social en el quehacer profesional". [10]

Dentro de los objetivos generales subrayamos "….formar un profesional capaz de analizar y evaluar requerimientos de procesamiento de información, y sobre esa base, diseñar, desarrollar, organizar, implementar y controlar sistemas informáticos, al servicio de múltiples necesidades de información, de las organizaciones y de todas las profesiones con las que deberá interactuar con versatilidad y vocación de servicio interdisciplinario." ". [10]

Con respecto al diseño curricular se observa que abarca los contenidos, y los aspectos metodológicos del trabajo profesional. Los contenidos básicos están fijados en función de las incumbencias y el perfil profesional. Permiten la profundización, de acuerdo con los requerimientos de la región, de los proyectos de cada facultad Regional y de las necesidades de actualización. Es por lo tanto flexible y orientado a las necesidades de cada región.

El plan de estudios consta de:

a) un tronco integrador constituido por un conjunto de materias cuya finalidad es crear, a lo largo de la carrera, un espacio de estudio multidisciplinario de síntesis. Se pretende que el estudiante se familiarice con trabajo ingenieril, partiendo de los problemas básicos de la profesión
b) de asignaturas electivas: que cada facultad regional oferta en función de sus posibilidades y necesidad regional; las áreas[7] a las que deben pertenecer las asignaturas son Científico- Técnica, Ciencias Sociales o Gestión Ingenieril, de las cuales al menos una deberá contribuir a la formación del alumno en la Administración de Recursos Humanos. La formación del ingeniero debe ser integral. Un problema específico debe ser considerado en forma global, no sólo técnicamente. Solucionarlo implica tener en cuenta su complejidad técnica y social. Los contenidos curriculares deben tener en cuenta esa cuestión. La inclusión de las asignaturas electivas en el diseño curricular de la oferta educativa de UTN, es un avance muy importante para la formación integral del profesional ingeniero. Se constituyen en un valioso aporte para la formación individual de cada quien. Son el eje que permita a los jóvenes emprender su propio recorrido. Su objetivo central es ofrecer a los cursantes la posibilidad de delinear su formación en sus áreas de interés. Cada facultad

[7] Áreas: campos epistemológicos (ORD;764/95)

tiene la libertad de la elección, no sólo de las asignaturas sino del mecanismo para seleccionarlas.

En un trabajo anterior [11] se mostró y trabajó sobre un relevamiento documental de asignaturas electivas de las 12 facultades regionales donde se oferta la carrera ingeniería en Sistemas de Información. En todos los casos se observó que se tiene en cuenta los avances tecnológicos, los requerimientos regionales y el perfil que la UTN impone a sus carreras. De esa forma se facilita el estar al tanto de las nuevas técnicas y herramientas y al mismo tiempo gestionar las habilidades blandas adquiridas.
El futuro ingeniero, por un lado puede aplicar su conocimiento técnico. Por otro está capacitado para desempeñarse en una organización gestionando los recursos desde una postura ética, eficaz, eficiente y creativa.
En e trabajo de referencia también se administró un cuestionario. Su objetivo fue conocer el grado de satisfacción de los graduados con respecto a la formación y en particular a la posibilidad de elección de dichas asignaturas
La metodología de enseñanza, al considerar los problemas básicos de ingeniería como punto de partida del proceso enseñanza-aprendizaje, favorece la integración de conceptos y conduce a la formación de un cuerpo de conocimientos facilitando la autogestión del aprendizaje.
Al estar organizadas las asignaturas por área de conocimiento, se favorece la integración. No nos detendremos en las áreas, sólo las nombraremos[8]:

a) área de formación homogénea básica(homogéneas por ser común a todas las especialidades de ingeniería de la oferta educativa)
b) área programación
c) área computación
d) área sistemas de información
e) área gestión ingenieril
f) área modelos

Contando con el tronco integrador y la organización de las asignaturas por campos epistemológicos es coherente y necesario no separar la teoría de la práctica. La dupla teoría-práctica debería se considerada cómo un campo de generación de conocimiento y no como campo de aplicación. Se observa que los campos epistemológicos de este currículo se corresponden con los propuestos por Alicia de Alba y explicitados
La UTN desde el año 1959, cuando comenzó a funcionar como universidad hasta la actualidad, se ha consolidado y participado muy activamente en el desarrollo, a través de la inserción de sus graduados, en el mundo del trabajo.
La posibilidad de que cada Facultad Regional oferte las asignaturas electivas atiende no sólo las necesidades globales sino que facilita respuestas a las demandas regionales. Esto, de alguna forma, favorece su crecimiento constante.
La flexibilidad del plan de estudios representada por la elección de asignaturas, según necesidades de la región, favorece la respuesta a demandas locales. Sin embargo es posible observar que las electivas ofertadas en las regionales consideradas, son similares o versan sobre contenidos análogos. Esa cuestión no contradice la regionalización ni ignora las demandas locales. Por el contrario, da respuesta a la globalización del mundo actual. Contextos diversos con múltiples demandas, coincidentes o no.
La filosofía subyacente en la conformación de este diseño curricular favorece: la planificación de actividades, la investigación, la realización de informes y el planteo de situaciones problemáticas. En síntesis se centra en la significación de cada contenido.
Con respecto a la evaluación, coherente con la filosofía subyacente, es continua. Es parte del proceso de educativo y está al servicio de la dupla enseñanza-aprendizaje.
Los cursantes, además, deben cumplir con 400 horas de Práctica Profesional Supervisada, por cuanto la estructura curricular cuenta con un espacio que da cuenta de las posibles prácticas que llevará a cabo el estudiante cuando se reciba
Este diseño curricular, nacido como primera versión en los noventa, tiene un corte netamente posmoderno. Favorece la toma de conciencia de problemas compartidos en todas las sociedades actuales. Entre ellos la problemática de las armas nucleares, el cuidado del medio ambiente y la atención a los problemas regionales.

6. A modo de cierre

Sin intentar una definición de educación la reconocemos como un complejo proceso social que debe ser analizado desde la historia, la política, el poder, la cultura y la ética. Se desarrolla en un tiempo y espacio determinados.
El medio del que disponemos los educadores es el conocimiento. Suele decirse que la diferencia entre la educación y otras prácticas sociales es "socializar mediante el conocimiento". [12]
A lo largo de la historia hubo distintas interpretaciones del concepto de conocimiento. Cada una ligada a la época y cultura. Todas coincidentes en que la dupla realidad-conocimiento forma parte de sistemas complejos y multidimensionales.

[8] Para profundizar ver Ord, 764/95

Con referencia a las relaciones de la educación con el conocimiento. C. Cullen [12] sostiene que la escuela socializa mediante la enseñanza de conocimientos legitimados públicamente. No inventa sino resignifica el saber. No excluye, por el contrario, lo enseña a todos. Sostiene que es central la relación con el conocimiento, tanto del que aprende, como del que enseña. Se condiciona la selección de contenidos en concordancia con las competencias a lograr.

Según A. Perez Lindo [13] a partir de los setenta, la epistemología comenzó a considerar que toda teoría científica tiene que ser refutable. Esta cuestión cambió el paradigma vigente: se debe enseñar a pensar desde la incertidumbre, la pluralidad y la complejidad. La práctica educativa se complicó, los docentes deben justificar sus saberes. Hay más exigencia epistemológica. Deben poseer una conciencia de lo que enseñan. Paulo Freyre [6] denominaba "concienciación", la ciencia cognitiva "meta-cognición" y Edgar Morin [14] "conocimiento del conocimiento.

El docente debe gestionar el proceso del conocimiento. No es una cuestión trivial ya que se mueve, permanentemente, entre lo objetivo (el hecho, el concepto, etc.) y lo subjetivo (interpretación, creencias, etc.) mediado por el lenguaje y el entorno social.

La teoría del conocimiento ofrece a la educación el descubrimiento de que es posible aprender sólo. De que es bueno interpretar, discutir y validar. De la necesidad de justificar los saberes. De que para enseñar y aprender hay que participar.

Es imprescindible formar a los futuros profesionales teniendo en cuenta el contexto en el que se desarrolla esa formación. Mostrar las características del entorno primario y el macro entorno. No es posible ignorar lo que pasa en el mundo. Es central insistir en la responsabilidad que, como individuos integrantes de una sociedad, nos toca vivir. Hay que alertar sobre la necesidad de trabajar para lograr la solución de conflictos existentes.

Al respecto K. Apel [15] dice: "que sea hoy de hecho problemática la relevancia racional de la ética para solucionar el problema de la regulación de los conflictos en la era atómica".

Creemos que en la formación de los hombres del siglo XXI es central la necesidad de reconocer y respetar la sociedad multicultural [9]en la que nos toca vivir. Al mismo tiempo se debe insistir en la necesidad de reconocer nuestra historia y cultura. De esta forma podremos comprender las diferencias y establecer el contacto cultural.

Touraine [16] al respecto afirma :…"será vigorosa una sociedad capaz de reconocer la diversidad de los individuos, los grupos sociales y las culturas, y que sabrá hacer a la vez que se comuniquen entre ellos, suscitando en cada uno el deseo de reconocer en el otro el mismo trabajo de construcción que efectúa en sí mismo"

Para caracterizar las luchas que se llevan a cabo en la conformación de un currículo, acordamos con el siguiente párrafo de Foucault [17]: "Por poder hay que comprender primero la mutiplicidad de las relaciones de fuerza inmanentes y propias del dominio en que se ejerce, y que son constitutivas de su organización; el juego que por medio de luchas y enfrentamientos incesantes las transforma, las refuerza, las invierte; los apoyos que dichas relaciones de fuerza encuentran las unas en las otras , de modo que formen cadena o subsistema, o, al contrario, los corrimientos, las contradicciones que aíslan unas de otras, las estrategias , por último, que las tornan efectivas, y cuyo dibujo general o cristalización institucional toma forma en los aparatos estatales en la formulación de la ley, en las hegemonías sociales"

Granese, A. [18] conjetura : " El dilema recurrente , a saber, si la educación es un "condicionamiento" o un factor de crecimiento y de libertad en el desarrollo , nos acerca la cuestión……..Una educación inspirada en la interpretación crítica debería favorecer el crecimiento, tanto en las intenciones(en los deberes) de quien educa, como en los resultados que logra el educador en la persona del que es educado(con y para esta última)"

De Alba [2] sostiene que la educación es un proceso complejo de mutua afectación, en el que se reproducen y producen elementos culturales (símbolos, o textos), a través del contacto cultural. No acuerda con Durkheim, y Bourdieu para quienes la educación se reduce a la transmisión o reproducción de saberes.

Concretando creemos necesario formar a nuestros alumnos teniendo en cuenta que en un futuro serán dirigentes.

Es central el desarrollo de aptitudes para encarar problemas globales. Insistir en la importancia de conocer las fortalezas y las debilidades del contexto en el que se desempeñarán. Favorecer el proceso de reflexión y el logro de alcanzar la madurez de la observación crítica. Entendemos que la capacidad crítica es un aporte fundamental de la educación universitaria que impulsa el cambio de dirección cuando se estima necesario.

Finalmente opinamos que es necesario definir una política universitaria referida al saber, la pregunta sería ¿qué necesitamos saber para intentar solucionar los miles de problemas que, cómo sociedad tenemos?

Referencias

[9] Encuentro de diferentes culturas, ver Tourreine, A. 196, p.198

[1]. Puiggrós, A., en *Currículum: Crisis, Mito y Perspectivas* De Alba, Alicia, Buenos Aires, Argentina, Miño y Ávila, 1995, pp.7,pp.59,pp.6 3

[2]. Giroux H. – Mc Laren, P., *Sociedad, cultura y educación*, Madrid, Miño y Dávila, 1998, pp.82
[3]. McLaren, Peter, *Pedagogía Crítica, resistencia cultural y la producción del deseo*, Buenos Aires, Argentina, Aique, 1994, pp.30, pp.237

[4].Cullen, Carlos, *Autonomía moral, participación democrática y cuidado del otro: bases para un curriculum de formación ética y ciudadana*, Buenos Aires, Argentina, Ediciones Novedades Educativas, 3ra.edición, 2004, pp.24

[5].Díaz Barriga, Á., *Currículum. Tensiones conceptuales y prácticas. Revista Electrónica de Investigación Educativa, 5 (2*), 1993 Consultado el día 10 de junio de 2014 en: http://redie.uabc.mx/vol5no2/contenido-diazbarriga.html

[6]. Freire, Paulo, *Pedagogía del oprimido*, Montevideo, Uruguay, Tierra Nueva, 1970

[7]. Derrida J., *Políticas de la amistad*, Madrid, *España* , Trotta, 1998

[8] Mollis, M., *Las Universidades en América Latina: Reformadas o Alteradas?*, Buenos Aires, Argentina CLACSO, 2003,pp. 204

[9] Mollis, M., *Memorias de la Universidad: otras perspectivas para una nueva ley de Educación Superior*, CLACSO, Buenos Aires, Argentina, 2009

[10] UTN, Argentina, Ord. 764, 1995, Anexos B2, B4

[11] A. Ísola, M. Guerrero, M.Portillo; S. Ferreiros, A. Bursztyn, S. Granado Peralta, *Las asignaturas electivas como eje de la formación del ingeniero ante los desafíos del desarrollo tecnológico* Conassi, Córdoba, Argentina, 2013

[12] C. Cullen, *Crítica de las razones de educar. Temas de Filosofía de la Educación*, Bs.As., Argentina, Paidos, 2da reimpresión 2000 p p.27).

[13] Augusto, Perez Lindo,*Teoría del conocimiento y educación*, Buenos Aires, Argentina UNTREF, 2010

[14]Morín, Edgar, *Los siete saberes necesarios para la educación del futuro*, París, Francia, UNESCO, 1999

[15] K. Apel *La solución de conflictos en la era atómica como problema de una ética de la responsabilidad*, Cuadernos de ética nº5, Bs. As. , Argentina 1988, pp.10

[16] Touraine, A.; *¿Podremos vivir juntos?*; FCE, Buenos Aires, Argentina 1996, pp. 189

[17] M. Foucault en *Currículum: Crisis, Mito y Perspectivas* De Alba, Alicia, Buenos Aires, Argentina, Miño y Ávila, 1995, pp. 66

[18] A. Granese, A., Pensamiento crítico y educación, Buenos Aires, Argentina, Eudeba, 2003, pp. 160

Bases para el Diseño Curricular de Carreras de Ingeniería en Informática / Sistemas de Información

Gustavo López[(1)], Arturo Servetto [(1)], Adriana Echeverría[(1)], José Luis Cabrera[(1)], Patricia Calvo[(1)], Ismael Jeder[(1)]
[(1)]Facultad de Ingeniería
Universidad de Buenos Aires
Paseo Colón 850, C.A.B.A.
glopez@fi.uba.ar,aserve@gmail.com, adelacrena@yahoo.com.ar, jlc0044@yahoo.com, pat_calvo@yahoo.com, jeder@fi.uba.ar

Abstract

A partir de las áreas de conocimiento consensuadas por las redes de Ingenierías en Computación (RUNIC) y de Ingenierías en Informática / Sistemas de Información (RIISIC) del CONFEDI, y de las intensidades de enseñanza y aprendizaje elaboradas por la RIISIC para establecer los contenidos curriculares básicos de las mismas, se propone cargas horarias mínimas y máximas para los planes de estudio, y una caracterización de los niveles de intensidad con pautas de diseño curricular asociadas.

1. Introducción

En abril de 2012, en la ciudad de Córdoba se desarrolló un taller conjunto de las redes RUNIC y RIISIC del CONFEDI con el objeto de:

- Consensuar el núcleo común para dichas terminales, tomando como punto de partida la resolución ministerial 786/09 y los documentos de IEEE y ACM.
- Establecer lineamientos que garanticen un conjunto de conocimientos mínimos que permitan orientar la implementación de los diseños curriculares de ambas ingenierías, dando respuesta a los intereses expresados en las prospectivas nacionales atendiendo el desarrollo de cada Región.

En dicho taller se analizaron las áreas de conocimiento del documento *"Computing Curricula 2005"* a efectos de establecer un glosario común de referencia y se elaboró un listado de áreas y contenidos que pueden estar presentes en cada una de las terminales de Ingeniería y que además, permitan la internacionalización de sus profesionales. El listado de áreas de conocimiento incluye principalmente Tecnologías Básicas y Aplicadas, debiendo complementarse con las de Ciencias Básicas y Complementarias, para cumplimentar una adecuada formación ingenieril.

En el mismo taller, la RIISIC consensuó intensidades mínimas y máximas de participación de las áreas en los contenidos curriculares básicos involucrados en la terminal de Ingeniería en Sistemas de Información / Informática. Las intensidades definen la importancia relativa con que cada área de conocimiento debe estar presente, y pueden variar entre cero (0) y cinco (5), donde cero (0) implica no participación y cinco (5) máxima prominencia.

2. Fundamentos

El curriculum puede considerarse como el marco académico, político y social que establece las pautas para la formación de los profesionales del país. Debería propiciar la formación integral y el desarrollo de capacidades de los futuros profesionales.
Desafíos en la actualización del curriculum incluyen, entre otros, la atracción y retención de los programas de grado; el grado de integración que tenga la curricula con áreas de aplicación en el medio social y laboral adyacente; la manera en que se enseñe la abstracción a lo largo de toda la curricula; reducir la redundancia en el curriculum y reorganizarlo eficientemente [1].
El diseño curricular puede verse como un instrumento de mediación entre el curriculum ideal y el proceso de aprendizaje-enseñanza, actúa como elemento guía en la concreción de los objetivos educativos deseados, con una orientación en la formación integral y pertinente del estudiante, que permite mejorar y asegurar la calidad de los procesos educativos, en los resultados del sistema de seguimiento y evaluación curricular [2].

Mediante el plan de estudios se sistematiza el proceso de aprendizaje vía una estructura lógica en contenidos y tiempo.

Considerando el diseño curricular, tanto de carreras como de asignaturas, problemas de ingeniería, se impone una solución sistemática, disciplinada y cuantificable como corresponde a la disciplina.

Si el Ministerio de Educación adoptara, para futuras etapas de acreditación de carreras de Ingeniería en Informática / Sistemas de Información, las recomendaciones de contenidos curriculares básicos y los rangos de intensidad para su enseñanza y aprendizaje clasificados en las áreas asimiladas de las recomendaciones del IEEE y la ACM, por cuanto dichas áreas representan tanto una reestructuración de las subáreas establecidas en la resolución ministerial 786/09 cuanto una innovación por el establecimiento de rangos admisibles de intensidad, las universidades con carreras en esta terminal deberán adaptar y evaluar el diseño curricular de sus carreras y asignaturas para verificar su ajuste a estos nuevos estándares.

Los rangos de intensidad para la enseñanza y aprendizaje de contenidos en las nuevas áreas no sólo aún requieren la determinación de cargas horarias mínimas y máximas que pueden tener en los planes de estudio, sino también el establecimiento de estándares ad-hoc para especificar y verificar los tipos de saberes y habilidades de cada nivel de intensidad.

3. Propuestas

En la sección 4 se propone una taxonomía de las áreas de conocimiento, con determinación de cargas horarias mínimas y máximas para cada taxón, en función de los rangos de intensidad recomendados para cada área incluida.

En la sección 5 se propone una especificación de cada nivel de intensidad y pautas de diseño curricular de asignaturas que incluyan contenidos de áreas con ese nivel.

4. Cargas Horarias

En la Tabla 1 se propone los taxones para clasificar las áreas de conocimiento establecidas por las redes del CONFEDI, que excluyen las subáreas de Ciencias Básicas, excepto por algunos descriptores de contenidos de Matemática, e incluyen a las Complementarias establecidas en la resolución 786/09 del Ministerio de Educación.

En la Tabla 2 se propone la taxonomía de las áreas, con el rango de intensidades admisibles de cada una.

En la Tabla 3 se suman las intensidades mínimas y máximas de las áreas de cada taxón, para luego fundamentar la determinación de cargas horarias, que se indican en la Tabla 4.

Tabla 1. Taxones de Áreas de Conocimiento

Taxón	Nombre
AP	Algoritmos y Programación
EM	Elementos de Matemática
EP	Ética Profesional
FT	Fundamentos Tecnológicos
HE	Hardware y Electrónica
IS	Ingeniería de Software
SI	Sistemas de Información
TIC	Tecnologías de Información y Comunicación

Tabla 2. Taxonomía de Áreas de Conocimiento

#	Área de Conocimiento	Mín	Máx	Taxón
1	Elementos de Programación	3	4	AP
2	Programación Integrativa	2	4	AP
3	Algoritmos y Complejidad	1	3	AP
4	Arquitectura y Organización de Computadoras	1	2	HE
5	Principios y diseño de Sistemas Operativos	1	2	FT
6	Uso y Configuración de Sistemas Operativos	2	3	FT
7	Principios y Diseño enfocado a redes	1	3	FT
8	Uso y Configuración enfocado a redes	2	4	FT
9	Tecnologías de plataforma	1	2	FT
10	Teoría de la Computación y Lenguajes de Programación	1	3	AP
11	Interacción Humano-Computadora	2	4	TIC
12	Gráficos y Visualización	1	2	TIC
13	Sistemas Inteligentes (IA)	1	1	AP
14	Teoría de Gestión de Información (BD)	1	3	TIC
15	Práctica de Gestión de Información (BD)	2	4	TIC
16	Computación científica (métodos numéricos)	0	0	EM
17	Legal / Profesional / Ética / Sociedad	0	3	EP
18	Desarrollo de Sistemas de Información	3	5	IS
19	Análisis de Requerimientos de Negocios	3	4	IS
20	Negocios Electrónicos (E-Business)	2	4	TIC
21	Análisis de Requerimientos Técnicos	2	4	IS
22	Fundamentos Ingenieriles para software	1	3	IS
23	Economía Ingenieril para software	1	2	IS

#	Área de Conocimiento	Mín	Máx	Taxón
24	Modelado y Análisis de Software	2	4	IS
25	Diseño de Software	2	4	IS
26	Verificación y Validación de Software	1	3	IS
27	Evolución del Software (mantenimiento)	1	3	IS
28	Proceso de Software	2	4	FT
29	Calidad de Software	2	3	IS
30	Ingeniería de Sistemas Computacionales	0	1	IS
31	Lógica digital	1	1	HE
32	Sistemas Embebidos	0	1	FT
33	Sistemas Distribuidos	1	3	FT
34	Seguridad: problemas y principios	1	3	TIC
35	Seguridad: implementación y gestión	1	3	TIC
36	Administración de sistemas	1	3	FT
37	Gestión de Organizaciones de Sistemas de Información	1	3	SI
38	Integración de Sistemas	2	4	IS
39	Desarrollo de medios digitales	0	1	HE
40	Soporte técnico	1	1	IS
41	Teoría Organizacional	1	3	SI
42	Teoría de Decisión	1	2	SI
43	Comportamiento Organizacional	1	3	SI
44	Gestión de Cambio Organizacional	1	2	SI
45	Teoría General de Sistemas	1	2	SI
46	Gestión del Riesgo (Proyecto, riesgo de seguridad)	1	3	SI
47	Gestión de Proyecto	3	5	SI
48	Modelos de Negocios	1	4	SI
49	Áreas Funcionales de Negocios	1	4	SI
50	Evaluación del Desempeño de un Negocio	1	4	SI
51	Circuitos y Sistemas	0	1	HE
52	Electrónica	0	1	HE
53	Procesamiento de Señal Digital	0	1	HE
54	Diseño VLSI	0	1	HE
55	Tolerancia de Errores y Testeo de Hardware	0	0	HE
56	Bases Matemáticas	2	4	EM
57	Comunicación Interpersonal	3	4	TIC
	Intensidades Totales	**70**	**158**	

Tabla 3. Intensidades de Taxones

Taxón	Nombre	Mín	Máx
AP	Algoritmos y Programación	8	15
EM	Elementos de Matemática	2	4
EP	Ejercicio Profesional	0	3
FT	Fundamentos Tecnológicos	11	25
HE	Hardware y Electrónica	2	8
IS	Ingeniería de Software	21	41
SI	Sistemas de Información	13	35
TIC	Tecnologías de Información y Comunicación	13	27
	Intensidades Totales	**70**	**158**

En los estándares actuales, los contenidos curriculares básicos deben tener una carga horaria mínima de dos mil setenta y cinco (2075) horas, siendo la carga total para una carrera de cinco años de tres mil setecientos cincuenta (3750). Si se respeta la misma cantidad de horas mínimas correspondientes a las subáreas de Tecnologías Básicas (575), Tecnologías Aplicadas (575) y Complementarias (175), que totalizan mil trescientas veinticinco (1325), se infiere que cada unidad de intensidad equivale a 1325/70=18,93 horas. Se redondea la relación a diecinueve (19) horas por unidad de intensidad para trabajar con valores enteros.

Tabla 4. Cargas Horarias

Taxón	Nombre	Mín	Máx
AP	Algoritmos y Programación	152	285
EM	Elementos de Matemática	38	76
EP	Ejercicio Profesional	0	57
FT	Fundamentos Tecnológicos	209	475
HE	Hardware y Electrónica	38	152
IS	Ingeniería de Software	399	779
SI	Sistemas de Información	247	665
TIC	Tecnologías de Información y Comunicación	247	513
	Horas Totales	**1330**	**3002**

Con estos presupuestos, se deduce entonces que la carga horaria que cada Universidad puede asignarle a las Ciencias Básicas más áreas Complementarias, tiene un mínimo de 3750-3002=748 y un máximo de 3750-1330=2420. Cabe destacar que el mínimo coincide con el de los estándares actuales.

Dentro de áreas Complementarias se puede encuadrar actividades que impliquen la integración de áreas de conocimiento, el desarrollo de habilidades o aptitudes profesionales o asignaturas de áreas de conocimiento no incluidas, p. ej., tesis o trabajos de graduación, prácticas profesionales supervisadas, asignaturas de práctica tipo taller (integradoras de áreas de conocimiento), participación en proyectos de investigación o de extensión, asignaturas humanísticas (filosofía, letras, liderazgo, relaciones públicas), prácticas deportivas obligatorias, videojuegos, informática forense, etc.

5. Niveles de Intensidad

Cada una de los taxones descriptos en las secciones anteriores se asocia a un grupo de materias específicas de la carrera, correspondiéndole una determinada intensidad recomendada, tal como se ha indicado. Cada materia, acorde al trayecto al que corresponda, englobará un conjunto de temas, los cuales se detallarán en el programa correspondiente.

Se plantea entonces la cuestión de cómo expresar de la forma más precisa el grado de profundidad de cada tema, en función de la intensidad del taxón correspondiente. Una forma de estandarización de este aspecto del vocabulario facilitaría el armado de programas.

Para especificar los saberes y habilidades asociados y su grado, existen diversas taxonomías, siendo las más conspicuas las de Bloom, y la modificación posterior de Kratwohl y Anderson, y menos conocidas las más recientes, que incorporan orientación en función de las innovaciones tecnológicas actuales.

La de Bloom, presentada en 1956 (que en la propuesta inicial considera los dominios cognitivo, afectivo y psicomotor) propone para el aspecto cognitivo seis niveles a saber: conocimiento, comprensión, aplicación, análisis, síntesis y evaluación [3].

Varias revisones de esta clasificación fueron realizadas posteriormente [4]. La modificación hecha por Kratwohl y Anderson en 2000 [5], amplia el espectro de Bloom y expresa los niveles por verbos, los cuales son, en orden creciente: recordar, aprender, aplicar, analizar, evaluar y crear. En este planteo, una de las modificaciones fue asociar la capacidad de sintetizar con la posibilidad de crear, ubicando en consecuencia dicha competencia en un nivel superior. Si bien esta revisión incorpora elementos eficaces para el análisis y diseño de programas, cabe señalar que no cubre aspectos contemporáneos asociados a los procesos propios de la aplicación de las nuevas tecnologías de información y comunicación. Es por eso que se han ido desarrollando otras taxonomías, como la de Churches (2008), la cual considerando las características propias de la era digital, incorpora no sólo elementos del ámbito cognitivo (como verbos apropiados para el contexto tecnológico) sino también diversas herramientas (como actividades digitales) [6] [7].

Una consideración aparte merecen las formas colaborativas de aprendizaje. Se ha señalado que actualmente las habilidades para el trabajo colaborativo son esenciales, constituyendo cualidades clave el manejo de las formas apropiadas para la comunicación y el trabajo en equipo. En este sentido Churches señala que la colaboración puede asociarse en algunas ocasiones a la taxonomía de Bloom y en otras ser un mecanismo para facilitar alcanzar formas de pensamiento de orden superior.

Sobre la base de los aportes de estas clasificaciones actualizadas se propone abrir la discusión sobre los modos de expresar adecuadamente los objetivos en función de la graduación estandarizada de intensidades para ofrecer a los docentes responsables del diseño de programas herramientas para expresar adecuadamente el nivel a alcanzar en cada tema y facilitar el análisis comparado de programas.

6. Referencias

[1] http://www.acm.org/education: Future of Computing Education Summit, June 25- 26, 2009.

[2] Manual para el Diseño Curricular de Programas Educativos de Licenciatura, Profesional Asociado (Técnico Superior Universitario) y Técnico de la BUAP en el Marco del Modelo Universitario Minerva, 2007

[3] Bloom, B., (1956) Taxonomy of Educational Objectives Book 1: Cognitive Domain, Longman

[4] Owen Wilson (2006) Beyond Bloom - A new Version of the Cognitive Taxonomy. http://www.uwsp.edu/education/lwilson/curric/newtaxonomy.htm

[5] Andersen Kratwohl Anderson, L.W., and D. Krathwohl (Eds.) (2001). A Taxonomy for Learning, Teaching and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York.

[6] Churches, A. (2007), Educational Origami, Bloom's and ICT Tools http://edorigami.wikispaces.com/Bloom's+and+ICT+tools

[7] Churches, A. (2008): Welcome to the 21st Century http://edorigami.wikispaces.com/21st+Century+Learners

Metodologías activas y corrección por rúbricas en el proceso de enseñanza-aprendizaje de programación

Aubin, Verónica Blautzik, Leonardo Guatelli, Renata
Pafundi, Federico Salica, Matías Nisi, Zulema Zanga, Mabel
Departamento de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de la Matanza
Florencio Varela 1903 (B1754JEC)- San Justo, Buenos Aires, Argentina
vaubin@ing.edu.ar

Abstract

El presente trabajo cuenta la experiencia de la incorporación de metodologías activas y metodologías utilizadas en la industria del software con el objetivo de incrementar las habilidades del estudiante en su desempeño académico y acercarlo en forma temprana a la actividad profesional. Se enfoca en la creación y utilización de rúbricas, que contemplen la metodología aplicada y que colaboren con la evaluación de la adquisición de competencias en el área de programación, campo en el que su uso no está ampliamente difundido. La implementación de cuestionarios, encuestas de opinión, la observación en el aula y el rendimiento académico obtenido, permitieron la evaluación de las metodologías aplicadas.

Palabras clave

Metodología de enseñanza, enseñanza-aprendizaje programación, Enseñanza de la Ingeniería, Competencias, ABP, Aprendizaje basado en problemas, Metodologías activas, Rúbricas.

1. Introducción

La Educación y especialmente la Educación Superior deben agiornarse para formar profesionales que cuenten con las competencias requeridas en el ámbito laboral [1]. Esto implica revisar los conceptos y técnicas pedagógicas que se están utilizando. Requiere cambiar de la educación tradicional, donde el educador es el que sabe y el educando recibe todos los conocimientos del educador sin participar en el proceso, a una educación centrada en el discente y su aprendizaje, donde el estudiante debe asumir un papel más activo y responsable de su proceso formativo. El docente debe ser más un guía que facilite el aprendizaje apoyándose en diferentes recursos [2], definiendo las actividades y experiencias que se deben realizar para alcanzar los objetivos, debiendo precisar los métodos de enseñanza y de evaluación que sean útiles para lograr los aprendizajes propuestos.

Hay consenso en cuanto que el Ingeniero no solo debe saber, sino también saber hacer. El saber hacer no surge de la mera adquisición de conocimientos sino que es el resultado de puesta en funciones de una compleja estructura de conocimientos, habilidades, destrezas, etc., que requiere ser reconocida expresamente en el proceso de aprendizaje para que la propuesta pedagógica incluya las actividades que permitan su desarrollo [3]. El desarrollo de dichas competencias implica un desafío desde el punto de vista de las estrategias didácticas yendo más allá de las prácticas tradicionales de transmitir información, sino identificar estrategias que quizás no sean frecuentemente utilizadas.

Por metodologías activas se entiende hoy en día a aquellos métodos, técnicas y estrategias que utiliza el docente para convertir el proceso de enseñanza en actividades que fomenten la participación activa del estudiante generando aprendizajes más profundos, significativos y duraderos, ya que facilitan la transferencia a contextos más heterogéneos.

Se expone en este trabajo la experiencia realizada en la cátedra de Programación Avanzada de la Universidad Nacional de la Matanza (UNLaM), dentro del marco del Proyecto Estratégico en Ingeniería para Ciencias Básicas (PEICB), con el objetivo de buscar las metodologías más efectivas, que permitan aportar valores didácticos que incrementen las habilidades del estudiante en su desempeño académico y profesional. En un trabajo reciente de los autores [4], se presentan resultados de la aplicación de metodologías de la industria del desarrollo de software a la enseñanza de la programación. Se muestran las ventajas de la incorporación de la programación de a pares y la aplicación de la prueba primero, pertenecientes a Extreme Programming (XP) [5, 6, 7] y la utilización del Proceso Personal Software (PSP) [8].

En el presente trabajo se muestra la necesidad de elaborar y utilizar rubricas que permitan detallar criterios

específicos, que consideren la parte metodológica a la hora de evaluar al alumno. De este modo se pretende favorecer el proceso de enseñanza-aprendizaje también en el momento de la evaluación. El uso de rubricas trae aparejadas ventajas tanto para el alumno como para el docente, las que se detallarán más adelante. La rúbrica se presenta entonces, como un recurso para la evaluación integral y formativa, como un instrumento de orientación y como herramienta pedagógica que establece criterios con los que valorar y evaluar distintos niveles de desempeño y dominio de competencias.

La rúbrica es una herramienta que provee al profesor la posibilidad de manifestar sus expectativas sobre los objetivos de aprendizaje prefijados y al alumno de un referente que proporciona una retroalimentación de cómo mejorar su trabajo [9, 10]. El profesor puede especificar claramente qué espera del estudiante y cuáles son los criterios con los que se va a calificar un objetivo que se ha establecido previamente.

El artículo está organizado de la siguiente manera: en la sección 2 se presenta el marco teórico y se describen los cambios culturales necesarios para el éxito de la metodología. En la sección 3 se describe la experiencia realizada por el grupo de investigación y los resultados obtenidos. En las secciones 4 y 5 se exponen discusiones y conclusiones respectivamente.

2. Marco teórico

En esta sección se describen las metodologías activas y las provenientes de la industria de software que se utilizan en la cátedra de Programación Avanzada de la UNLaM.

2.1 Metodologías activas

Las metodologías activas promueven habilidades que permiten al estudiante juzgar la dificultad de los problemas, detectar si entendieron un texto, saber cuándo utilizar estrategias alternativas para comprender la documentación y saber evaluar su progresión en la adquisición de conocimientos [11]. Durante un aprendizaje auto-dirigido, los estudiantes trabajan en equipo, discuten, argumentan y evalúan constantemente lo que aprenden.

El aprendizaje en grupo hace que los estudiantes funcionen como un equipo, donde es necesaria la participación directa y activa de todos los integrantes, que al igual que los alpinistas, citados por Johnson, los estudiantes escalan más fácilmente las cimas del aprendizaje cuando forman parte de un equipo cooperativo [12]. En nuestro caso, más que un trabajo cooperativo planteamos un trabajo colaborativo, ya que el primero implica un ambiente cerrado y controlado, con reglas rígidas, definidas previamente donde el desarrollo personal está supeditado a los objetivos organizacionales, mientras que el segundo tiene un ambiente abierto, libre, que estimula la creatividad y la motivación está supeditada al compromiso personal [13].

Dentro de las metodologías que requieren un aprendizaje activo, el Aprendizaje Basado en Problemas (ABP) despierta en el estudiante la curiosidad, creatividad y razonamiento crítico a la vez que permite la integración de la teoría con la práctica, la habilidad para buscar información, y la capacidad de autoevaluar lo aprendido [14].

Actualmente, la tecnología ubicua permite la interacción con pares y el acceso en todo momento y en todo lugar a la información [15], ya no hace falta recurrir a un experto para conocer un tema, sólo hay que saber dónde y cómo buscar esa información

2.2 Metodologías de la industria del software

La industria necesita profesionales con una visión total del proceso de desarrollo de software y con la disciplina para enfrentarlo. Conocimientos y disciplina deben adquirirse desde la universidad, de forma que permitan incrementan las habilidades del estudiante en su desempeño académico y profesional.

2.2.1 Programación Extrema (XP)

Las metodologías ágiles de desarrollo de software, y en particular XP, constituyen una de las tendencias de mayor impacto en la industria del desarrollo de software en la última década, gracias a su enfoque centrado en la generación temprana de valor y en su acento en el aspecto humano del desarrollo de software [16].

En la materia se aplica, en especial, dos aspectos de las metodologías ágiles:

Programación en Pares: XP propone que el trabajo de desarrollo se lleve a cabo por grupos de trabajo compuestos por dos programadores, ambos trabajan juntos en un mismo ordenador, se complementan y ayudan para generar la solución a cada problema, escribir el código y/o las pruebas.

Los estudios demuestran [17, 18] que, contra lo que pudiera pensarse, dos programadores son más eficientes que uno solo para una tarea determinada, el resultado es mayor que la suma de las partes.

En el trabajo en el aula la selección de los integrantes de cada par queda a cargo de los mismos estudiantes. Esta selección se hace de manera natural y espontánea. Una acertada elección del compañero es clave para el buen desempeño del par, Faria plantea en [19], que la conformación de los grupos de estudiantes con las mismas capacidades, hacen que su aprendizaje y producción sea más efectiva, que si fuera con capacidades diferentes.

La preparación de la prueba primero: Las metodologías tradicionales, ubican sobre el final del proyecto, o sobre el final del desarrollo de cada módulo la fase de pruebas, incluyendo la definición de los tests. Por el contrario, la metodología XP propone que, lo primero que se escriba sean los test que el sistema debe pasar [20]; luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.
Luego de la etapa de análisis, antes de encarar cualquier solución de un problema o consigna, por lo tanto, antes de la codificación, los estudiantes deben tener preparado el lote de pruebas y/o el programa probador o el método test donde se realizará la prueba unitaria en el caso de un método de una clase. Todo por supuesto dentro de un ciclo de vida iterativo e incremental. Esto implica un cambio cultural importante sobre las prácticas habituales de los estudiantes.
La preparación temprana de la prueba mejora el análisis, hace que se comprenda mejor el problema pensando en los casos particulares y de fatiga. La prueba se hace inmediatamente después de la codificación de cada incremento o iteración. Al hacer la prueba de ésta manera se descubren errores que se corrigen con el código más "fresco" [4].

2.2.2 Estimaciones y métricas - PSP

El PSP muestra cómo producir de forma habitual software de alta calidad, entendiendo que con un proceso de calidad los productos derivados de éste serán también de calidad [21]. Ayuda a los desarrolladores a organizar y planificar su trabajo, realizar el seguimiento de su rendimiento, gestionar defectos de software, y analizar y mejorar su proceso de desarrollo personal [22]. Les da a los desarrolladores un marco de referencia de su disciplina personal para mejorar su trabajo y realizarlo con alta calidad, los ayuda a aprender y practicar aquellos métodos para producir software que son más efectivos para ellos, ya que reconoce que cada persona es esencialmente diferente [23].

El PSP consiste en un conjunto de métodos, formularios y guías que muestran a los desarrolladores, la forma de planificar, medir y administrar su trabajo [21]. El PSP requiere una recopilación y análisis de métricas con un elevado nivel de detalle, lo cual no es algo trivial. Cuando el desarrollador tiene suficiente cantidad de mediciones, será capaz de comprender en qué gasta su tiempo, dónde y por qué introduce defectos y cuánto le toma encontrarlos, corregirlos y prevenirlos [23].

Se debe tener presente que el PSP no resuelve los problemas que aparezcan durante el desarrollo de software, pero ayuda en el proceso de crear una práctica disciplinada que puede ser analizada y mejorada [24].

La incorporación del PSP en la formación del profesional en la universidad, tiene como objetivo crear las bases para introducir métodos disciplinados de trabajo cumpliendo con los requerimientos de la industria. La toma permanente de métricas, fue quizás el cambio de más difícil aceptación, por parte de los estudiantes. Los pares deben estimar y medir cada incremento del ciclo de vida [25]. PSP le permite al estudiante conocer y establecer su ritmo de trabajo para alcanzar la meta fijada.

2.3 Rubricas

Una rúbrica o matriz de valoración se define como "un descriptor cualitativo que establece la naturaleza de un desempeño" [26]. Son instrumentos de medición en los cuales se definen escalas de estándares, para los diferentes criterios a evaluar, de forma que permitan determinar la calidad con la cual los estudiantes resuelven una amplia variedad de tareas específicas [27]. Según Cebrián et al. [28] la rúbrica "ayuda a definir y explicar a los estudiantes lo que espera el profesor que aprendan, y dispone de criterios sobre cómo va a ser valorado su trabajo con ejemplos claros y concretos. (...) El alumno puede observar sus avances en términos de competencias, saber en cualquier momento qué le queda por superar y qué ha superado y cómo", por lo cual, son una herramienta de gran utilidad para dar soporte al proceso de enseñanza-aprendizaje.

Podemos ver a la rúbrica como un recurso para la evaluación integral y formativa, como un instrumento de orientación o como herramienta pedagógica [29].

Las rúbricas generalmente están compuestas por:

- Dimensiones, Criterios de evaluación, Criterios de rendimiento o Indicadores: detallan los procesos y contenidos que se consideran importantes
- Escalas: niveles de rendimiento esperados
- Descriptores: Narraciones detalladas, que describen la habilidad que debe demostrar el estudiante para alcanzar una dimensión y un nivel de rendimiento determinado.

Se distinguen dos tipos de rúbricas [30]:

Holística, Comprehensiva o Global: En este tipo de rúbrica se evalúa el todo, sin juzgar cada una de las partes por separado, los diferentes criterios son considerados de manera general, sobre una única escala descriptiva resultando un juicio más global [31]. Son más fáciles de calificar, pero proporcionan menos retroalimentación al estudiante. Promueven una valoración sumativa.

Analítica: inicialmente, los aprendizajes se desglosan en tareas específicas y se definen criterios cuantitativos para cada uno de ellos. El docente evalúa por separado, las diferentes partes del producto o desempeño y posteriormente suma la puntuación de éstas para obtener una calificación total [31]. El proceso de calificación es más lento, ya que se requiere evaluar varias veces el

producto, analizándolo individualmente para cada habilidad o característica. Por eso, tanto su elaboración como su utilización requieren más tiempo. Los estudiantes reciben retroalimentación de cada uno de los aspectos o características considerados, además, al conocerla con anticipación a la evaluación, les da los elementos necesarios para autoevaluarse. Permiten que el docente tenga un perfil de las fortalezas y debilidades específicas de cada estudiante, lo cual permite rever las prácticas implementadas y analizarlas para implementar mejoras. Por lo tanto pueden utilizarse para determinar el estado alcanzado por los estudiantes, identificar sus fortalezas, debilidades, y para permitir que los estudiantes conozcan lo que requieren para mejorar [32]. Promueven una valoración formativa.

Trabajar con rúbricas permite darle un valor más auténtico o real, a las calificaciones tradicionales expresadas en números o letras, sirve para averiguar cómo está aprendiendo el estudiante, permite evaluar el proceso y el producto, nos da una forma más objetiva y consistente para evaluar, ya que le permite al docente especificar claramente qué espera del estudiante y cuáles son los criterios con los que se va a calificar un objetivo que se ha establecido previamente. Al trabajar con un equipo de docentes, la evaluación se hace independiente de quien lleve a cabo la tarea, ya que el criterio fue establecido con anterioridad y es conocido por todo el equipo docente y por los alumnos.

Al evaluar con rúbricas, el estudiante entiende porqué razón obtiene una nota determinada, sabe qué es capaz de hacer y qué le falta para llegar a los distintos niveles de rendimiento esperados. Proporciona la información suficiente para que el estudiante sepa qué puede hacer para avanzar en su proceso personal de aprendizaje.

Debemos tener en cuenta que una rúbrica es un documento perfectible que se encuentra en constante modificación, atendiendo a las necesidades de cada curso y a las propias prácticas aprendidas para mejorar el proceso de enseñanza – aprendizaje.

3. Experiencia y resultados

En esta sección mostramos los resultados obtenidos durante los años (2013, 2014) de aplicación de la metodología descripta en el presente trabajo, en el curso Programación Avanzada de la UNLaM. Se explica también el método de evaluación de los alumnos.

3.1 Modalidad de trabajo

La materia donde se aplican las prácticas metodológicas expuestas, tiene una carga horaria de 10 horas semanales con una modalidad cuatrimestral. Se dicta un curso por turno, mañana, tarde y noche, con aproximadamente 40 alumnos cada uno y 2 o 3 docentes por curso.

Durante la cursada los alumnos deben resolver una guía de ejercicios de 1600 líneas de código (LOC). Resolver entre 4 o 5 trabajos prácticos especiales de entrega obligatoria, que totalizan aproximadamente 1000 LOC, más una aplicación Cliente - Servidor de aproximadamente 1200, 1500 LOC, dependiendo de las funcionalidades solicitadas. Además, para completar la aprobación de la materia deben rendir dos evaluaciones parciales.

Los alumnos trabajan en parejas que se forman de manera espontánea. El grupo de trabajo debe seguir la siguiente metodología para la resolución de todas las actividades planteadas: debe completar la planilla de PSP con las estimaciones, confeccionar el lote de pruebas o programa probador de la solución general, codifica la solución correspondiente a la iteración actual, completa la planilla de PSP con los valores reales, realiza el testeo unitario de esa iteración, continua con el mismo proceso para las restantes iteraciones hasta lograr la solución final.

Todas las parejas aplican ésta metodología, tanto durante el curso como en las evaluaciones.

La evaluación se lleva a cabo en alguno de los laboratorios de la universidad, donde cada PC cuenta con todo el software necesario, entorno de programación, lenguaje Java y demás herramientas adecuadas para cumplir con la metodología, procesador de texto y planilla de cálculo. Los estudiantes se ubican de a dos por PC, se les entrega un problema nivel II o III seleccionado del banco de problemas de la O.I.A (Olimpiada Informática Argentina) los cuales se encuentran clasificados en tres niveles, según su dificultad, o un enunciado de nivel similar generado por los docentes de la cátedra.

Al finalizar el tiempo de la evaluación, generalmente de 2 a 3 horas, se le entrega a cada par de estudiantes, un lote de prueba confeccionado especialmente por el equipo docente. Se evalúa la salida obtenida para cada caso de prueba del lote entregado por el docente, la complejidad computacional del algoritmo que resuelve el problema, las estructuras de datos utilizadas, la eficiencia y la economía de recursos. Otro ítem evaluado es el la calidad del lote de prueba generado por los alumnos durante la evaluación, que contenga casos de fatiga y casos particulares e ingeniosos que muestran que el problema ha sido comprendido en su totalidad y ha sido resuelto cubriendo todos los requerimientos del mismo. El lote de pruebas que generan los alumnos durante la evaluación es un lote reducido, en comparación con el que generan habitualmente en las otras actividades del curso.

La evaluación se efectúa de manera presencial y cada alumno tiene la posibilidad de defender su examen, transformando la evaluación en un acto más del proceso de enseñanza-aprendizaje.

La diversidad de las respuestas planteadas por los alumnos frente a un mismo problema, lejos de ser un inconveniente, es fruto de la creatividad, de la falta de pre-conceptos y de una modalidad de trabajo activa.

¿Qué ocurre al momento de evaluar frente a esta diversidad de respuestas, algunas más eficientes que otras, algunas más creativas, otras triviales?

¿Cómo lograr unificar criterios de corrección y despojar a ésta de la mayor subjetividad posible?

¿Cómo contemplar en la evaluación el uso correcto de la metodología de trabajo y determinar si este buen uso conduce a los resultados esperados?

Tratándose de la enseñanza de programación y sobre todo en las carreras de ingeniería informática, la respuesta que se propone en este trabajo a estas preguntas, es la aplicación de rúbricas, diseñadas especialmente para contemplar la totalidad de la metodología implementada y que aseguren la completa revisión de todos los objetivos buscados al momento de la evaluación, permitiendo que todas las estrategias de resolución válidas sean analizadas y ponderadas, siguiendo el mismo criterio objetivo. Estas rúbricas ponen de manifiesto para los alumnos la importancia del cumplimiento de la metodología para la aprobación de la materia. Las rúbricas están implementadas en planillas de cálculo, se encuentran on-line y son de acceso libre para todos los alumnos.

En las rúbricas diseñadas se han definido dos tipos de objetivos:

- Objetivos esenciales: Serán aquellos que requieren un mínimo de cumplimiento para que la evaluación sea aprobada. En caso de no alcanzar este mínimo, la evaluación no podrá ser aprobada.
- Objetivos no esenciales: serán aquellos que pueden o no ser alcanzados hasta un mínimo. Influyen en la nota final, pero en caso de no ser alcanzados, no implica la desaprobación de la evaluación.

La figura 1 muestra a modo de ejemplo, uno de los objetivos esenciales, “La Prueba Primero”, de la rúbrica correspondiente a la segunda evaluación parcial. Dicho objetivo tiene asignado un puntaje máximo de 13 puntos, siendo indispensable alcanzar al menos 7 puntos (que representan más del 50% del puntaje máximo posible) para lograr la aprobación.

Objetivo	Carácter	Puntaje mínimo	Puntaje máximo
La Prueba Primero	Esencial	7	13

La Prueba Primero	Puntaje / Objetivos evaluados	0	1	2	3	4	5
	Cantidad y calidad de los casos de prueba	No realizado	Solo contempla casos incompletos o con muy pocos datos (uno o dos)	Los casos son variantes del mismo, sin creatividad ni ingenio	Hay poca diversidad pero las principales variantes son contempladas	Casos ingeniosos donde se tienen en cuenta la mayoría de las variantes del problema.	Ídem el anterior, pero agrega y genera los casos de fatiga.
	Archivos generados	No fueron generados	Fueron generados solo los archivos de entrada. No se corresponde con la documentación.	Fueron generados tanto entradas como salidas esperadas, pero los nombres no son descriptivos	Todos los casos fueron generados (IN/OUT). Los nombres son descriptivos y se corresponde con la documentación		
	Documentación del Lote de Pruebas	No realizada	Solo es una enumeración de los casos de prueba	Solo presenta los datos de entrada y salida esperada, sin descripcoón	Datos de entrada, salidas esperadas pero las descripciones son muy deficientes.	Datos de entrada, salidas esperadas, buenas descripciones.	Buenas descripciones, claras y concretas. Muestra los datos de entrada y salida esperados. Indica ubicación de archivos

Figura 1

En las figuras 2 y 3 se muestra una rúbrica completa con los objetivos “La prueba primero”, “Estimaciones y métricas”, “Codificación” y “Funcionamiento”.

Alumnos	Metodología									
	La prueba Primero				Estimaciones y Métricas					
Apellido y nombre	Cantidad y Calidad de los casos de prueba	Archivos generados	Documentación del Lote de Pruebas	Nota LP	Tratamiento de la prueba	Definición de Incrementos	LOC	LOC/H	Medición de Errores	Nota MT
Manto_Macri	5	3	5	13	3	3	3	3	3	15
Cuelli_Magurno										
Ponchioni_Pizarro										
Garcia_Funes										
Garayalde_Vergara										
Fernandez-Cutri										
Aranda - Gentile										
DiazSalamanca-Drzewko										

Figura 2

Programación												
Código				Funcionamiento de acuerdo a los casos de prueba propuestos								
Algoritmo utilizado				Caso 0	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso Fatiga		
Nivel de eficiencia del algoritmo utilizado	Soporta los casos de fatiga?	Indican correctamente la complejidad computacional?	Nota CD								Nota Func.	Nota final

Figura 3

La nota de la evaluación se obtiene a partir de una fórmula matemática que contempla que se hayan alcanzado los objetivos esenciales. En el momento de definir esta fórmula, es cuando el cuerpo docente delibera sobre la ponderación de los objetivos y su peso o importancia en la nota final. Es de suma importancia que llegado el momento de la evaluación y estando frente a los alumnos, no queden dudas respecto a los criterios de evaluación, la rúbrica debe hablar por si sola y es la que definirá la nota.

4. Discusión

Estas prácticas están siendo aplicadas y consideradas en la UNLaM, como modelo aplicable en las asignaturas de programación correlativas a Programación Avanzada. Al ser una metodología que toma técnicas propias de la Industria del Software, le da un valor agregado a su utilización en el aula.

En [33] los autores exponen que "la evaluación para el aprendizaje es el proceso de buscar e interpretar evidencias para que estudiantes y profesores conozcan dónde se encuentra el alumno en relación a su aprendizaje, dónde necesita estar y cuál es el modo mejor de llegar allí", se ha comprobado en nuestra experiencia, que el uso de rúbricas para la corrección de las evaluaciones, contribuyen de manera eficiente tanto para el alumno, como para el docente, en conocer las competencias alcanzadas y el nivel desarrollado en cada una.

Que el alumno conozca el nivel alcanzado para cada competencia, implica también conocer sus puntos fuertes y débiles lo que le permite tomar decisiones de como plantear su proceso de aprendizaje [35].

Como ya se ha comentado la diversidad de soluciones propuestas a un mismo problema de programación, implica tener objetivos claros al momento de la evaluación, más aún si la misma es llevada a cabo por varios docentes, el uso de rúbricas ha contribuido a reducir el nivel de subjetividad en el momento de la calificación [34, 35]. Así mismo el uso de lotes de prueba diseñados por el cuerpo docente, para verificar el correcto funcionamiento del trabajo generado por los estudiantes, es otro elemento que contribuye a eliminar la subjetividad.

5. Conclusión

Se han logrado avances significativos en cuanto al trabajo colaborativo y a la adaptación a los cambios. Los estudiantes son capaces de enfrentarse a situaciones problemáticas nuevas y lograr resolverlas exitosamente. El conocimiento de la rúbrica, la prueba continua, las estimaciones, registro de métricas y el trabajo de a pares hace que los alumnos conozcan antes que el docente el resultado de su propio trabajo. Se ha mejorado considerablemente el ambiente en el aula, los alumnos investigan y colaboran para lograr los objetivos y consignas planteadas.

En lo que respecta a las evaluaciones, éstas son tomadas por los alumnos de manera menos traumáticas y con menos presión al trabajar de a pares. Cada evaluación es claramente una experiencia motivadora y enriquecedora.

La rúbrica ayudo a eliminar subjetividades ya que le asegura a cada alumno ser evaluado con los mismos criterios que sus compañeros. El estudiante entiende porqué razón obtiene una nota determinada, ya que la rúbrica le proporciona información sobre los conocimientos que le faltan para alcanzar los distintos niveles de rendimiento esperados.

Todo esto ha contribuido a mejorar la motivación, a disminuir sensiblemente la deserción de quienes comienzan la cursada y finalmente a aumentar la tasa de aprobados al final del curso.

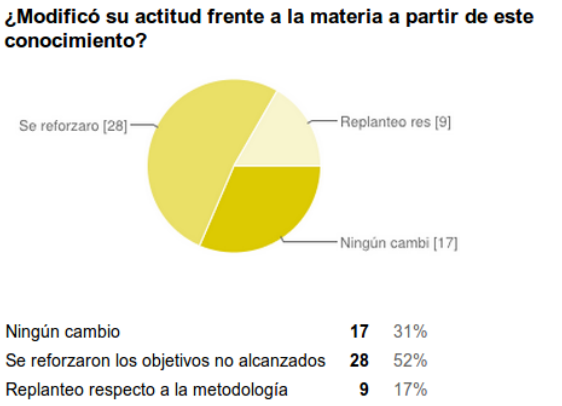
5.1 Impacto en los alumnos – Encuestas

Para conocer el impacto que la corrección con rúbricas produjo en los alumnos se han elaborado encuestas anónimas, las cuales han sido respondidas al final del cuatrimestre.

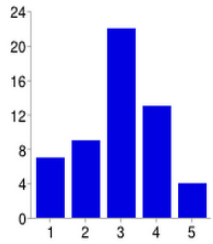
Se consultaron 4 aspectos:

- De qué manera la evaluación por objetivos modificó la actitud respecto al proceso de enseñanza - aprendizaje
- El impacto de la rúbrica en el proceso de enseñanza aprendizaje.
- La importancia de conocer la rúbrica antes de la evaluación.
- La rúbrica ayuda a los estudiantes a conocer sus debilidades y fortalezas.

Se muestran a continuación los resultados obtenidos:



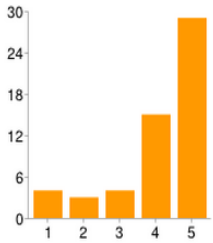
¿Considera de utilidad para su aprendizaje el hecho de haber sido corregida su evaluación mediante una rúbrica.?



1	7	13%
2	9	16%
3	22	40%
4	13	24%
5	4	7%

1: poca utilidad (peor)
5: mucha utilidad (mejor)

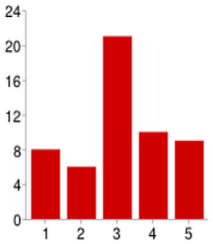
¿Considera de utilidad conocer la rúbrica de corrección antes de la evaluación parcial?



1	4	7%
2	3	5%
3	4	7%
4	15	27%
5	29	53%

1: poca utilidad (peor)
5: mucha utilidad (mejor)

¿En qué medida la rúbrica de correción le permitió conocer sus debilidades y fortalezas?



1	8	15%
2	6	11%
3	21	39%
4	10	19%
5	9	17%

1: poco (peor)
5: mucho (mejor)

5.2 Ventajas y desventajas de la evaluación por rúbricas

Ventajas	Desventajas
En toda tarea que se les asigne a los alumnos deben de establecerse de forma clara y precisa los criterios que fundamentan los objetivos de enseñanza.	Conlleva un trabajo extra de análisis de resultados posibles por parte de los docentes, antes de asignar cualquier tarea. Sin este análisis previo, la rúbrica carece de sentido
Asegurar de forma precisa el nivel de aprendizaje que se desee de los alumnos.	
Fijar con anterioridad objetivos esenciales y no esenciales.	
Los docentes pueden mejorar la calidad de su enseñanza al enfatizar y precisar los detalles particulares que consideren más pertinentes para garantizar trabajos de excelencia por sus alumnos.	
Permite a los docentes obtener una medida más precisa tanto del producto como del proceso.	Trabajo previo extra y sobre todo, exige la unicidad de criterios que no siempre se logra.
Los estudiantes tienen una guía de forma explícita para realizar sus tareas de acuerdo a las expectativas de sus docentes.	Tendencia al conductivismo.
Asegurar de que todos los objetivos son evaluados, y que todos los docentes usarán el mismo criterio de evaluación.	Correcciones más detalladas y extensas. Frecuentemente consumen una o dos clases.

6. Agradecimientos

A las autoridades del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza, por los cambios puestos en marcha durante estos últimos años. A Marcela Imperiale y Fabiana Grinsztajn por sus sugerencias y comentarios.

7. Referencias

[1] Pozner, P. "Competencias para la profesionalización de la gestión educativa: diez módulos destinados a los responsables de los procesos de transformación educativa." Instituto Internacional de Planeamiento de Educación IIPE–UNESCO, sede regional Buenos Aires, Argentina, 2000.

[2] García González, J. L. y García Ruíz, R. “Aprender entre iguales con herramientas Web 2.0 y Twitter en la universidad. Análisis de un caso”. EDUTEC. Revista Electrónica de Tecnología Educativa. Núm. 40 / Junio 2012.

[3] Consejo Federal de Decanos de Ingeniería (CONFEDI), Competencias Genéricas. Desarrollo de competencias en la Enseñanza de la Ingeniería Argentina. San Juan, Argentina: Universidad Nacional de San Juan, Argentina, 2007.

[4] Aubin, V.; Blautzik, L.; Dejean, G. “Mejoras en el Proceso de Enseñanza-Aprendizaje de Programación Utilizando Metodologías de la Industria Del Software”, primera edición CoNAIISI Facultad regional Córdoba de la Universidad Tecnológica Nacional. Noviembre, 2013.

[5] Beck, K. “Extreme Programming Explained: Embrace Change”. Reading, Massachusetts: Addison-Wesley. 2000.

[6] Cockbun, A.;Williams, L. "The Costs and Benefits of Pair Programming". Humans and Technology Technical Report. 2000.

[7] Letelier, P.; Penadés, C. “Metodologías ágiles para el desarrollo de Software: eXtreme Programming (XP)”, Universidad Politécnica de Valencia 2006.

[8] Humphrey, Watts S. “An Introduction to the Personal Software Process”, Addison-Wesley, 1997.

[9] Mertler, C.A. 2001, Designing scoring rubrics for your classroom. Practical Assessment, Research y Evaluation, 7 (25).

[10] Roblyer, M.D. & Wiencke, W.R. (2003). Design and use of a rubric to assess and encourage interactive qualities in distance courses. The American Journal of Distance Education, 17 (2), 77-97.

[11] Brunning R.H., Schraw G.J., Ronning R.R., Cognitive Psychology and Instruction 2nd ed., Englewoods Cliffs (ed.) New Jersey: Prentice Hall, 1995.

[12] Johnson, D.; Johnson, R. y Holubec, E. “El aprendizaje cooperativo en el aula”. Buenos Aires. Piadós. 1999.

[13] Panitz T. “Collaborative versus cooperative learning-a comparison of the two concepts which will help us understand the underlying nature of interactivelearning”. 2001.

[14] Venturelli, J. Educación médica. Nuevos enfoques, metas y métodos. Medical Education: New Approaches, Goals and Methods. Washington: Pan American Health Organization, PALTEX, 1997.

[15] Cope B.; Kalantzis M. “Aprendizaje ubicuo”. Traducción. Quintana Emilio. Grupo Nodos Ele. 2009.

[16] Villena Moya, A. A. “Un modelo empírico de enseñanza de las metodologías ágiles”. Tesis para optar al grado de Magíster en Ciencias, Mención Computación. Universidad de Chile. Santiago de Chile – Enero 2008

[17] Cockburn, A.; Laurie W. "The costs and benefits of pair programming." Extreme programming examined 2000: 223-247.

[18] Williams, L. “Experimenting with industry’s pairprogramming model in the computer science classroom”. Computer Science Education. 2001.

[19] Faria, E. S. J. de; Adan-Coello, J. M.; Yamanaka, K. “Forming groups for collaborative learning in introductory computer programming courses based on students programming styles: An empirical study”. In 36th ASEE/IEEE Frontiers in Education Conference, pages 6-11. 2006.

[20] Beck, K. “Test Driven Development: By Example”, Addison-Wesley Professional, 2002.

[21] Humphrey, W.S. "The Personal Software Process (PSP)". Technical Report. The Software Engineering Institute. 2000.

[22] Towhidnejad, M., & Hilburn, T. Integrating the Personal Software Process (PSP) across the undergraduate curriculum. IEEE Computer Society , 162 - 168. 1997.

[23] Humphrey, W.S. "Why Should You Use A Personal Software Process?". Software Engineering Notes. Vol. 20, Issue 3, pp. 33-36. 1995.

[24] Hilburn, T.B. "PSP Metrics in Support of Software Engineering Education". Proceedings of the 12th Conference on Software Engineering Education and Training, pp. 135-136. 1999.

[25] Dejean, G.; Aubin, V.; Blautzik, L. “El Uso De Técnicas PSP Para El Logro De Competencias” Las Jornadas Chilenas de Computación (JCC) 2011. XIII Congreso Chileno de Educación Superior en Computación (CCESC'2011). 2011.

[26] Marielle, S.; Forgette-Giroux, R. "A rubric for scoring postsecondary academic skills." Practical Assessment, Research & Evaluation 7.8. 2001.

[27] Vélez, Lamberto V.; Palomares, A. Medición, assessment y evaluación del aprendizaje. Publicaciones Puertorriqueñas, 2008.

[28] Cebrián, M., Buenas prácticas en el uso del e-portafolio y e-rúbrica, In El practicum: buenas prácticas en el Espacio Europeo de Educación Superior por A. Cid, M. Raposo y A. Pérez (coords.), 67-87, Tórculo, Santiago de Compostela 2007.

[29] Raposo, M., and Martínez, E. "La rúbrica en la enseñanza universitaria: Un recurso para la tutoría de grupos de estudiantes." Formación universitaria 4.4 (2011): 19-28.

[30] Mertler, C. A. "Designing scoring rubrics for your classroom." Practical Assessment, Research & Evaluation 7.25 (2001): 1-10.

[31] Moskal, B. M. Scoring rubrics: what, when and how? Practical Assessment, Research & Evaluation, 7(3). 2000

[32] Shipman, D, et al. "Using the analytic rubric as an evaluation tool in nursing education: The positive and the negative." Nurse education today 32.3 (2012): 246-249.

[33] Benito, A. y Cruz, A., Nuevas claves para la docencia universitaria en el Espacio Europeo de Educación Superior, Narcea, Madrid –España. 2005.

[34] Martínez, M.E. y Raposo, M., Seguimiento de trabajos tutelados en grupo mediante rúbricas, In La docencia en el nuevo escenario del EEES, por Vicerrectoría de Formación e Innovación Educativa, 567-570, Universidad de Vigo, Vigo. España. 2010.

[35] Raposo, M. y Sarceda, M. C. Como avaliar unha memoria de prácticas? Un exemplo de rúbrica no ámbito das novas tecnoloxías, eIn Prácticas educativas innovadoras na universidade, 107-124, Universidad de Vigo, Vigo. España. 2008.

Aplicación de Experimentos Controlados en la Enseñanza de la Ingeniería de Software

Sergio Gustavo Zapata, Gustavo Antonio Sevilla, Estela Liliana Torres
Instituto de Informática
Universidad Nacional de San Juan
San Juan, Argentina
{szapata,gsevilla,etorres}@iinfo.unsj.edu.ar

Resumen

El uso de métodos empíricos en actividades de investigación en Ingeniería de Software es cada vez más extendido. Mediante los métodos empíricos es posible evaluar nuevos aportes de la Ingeniería de Software antes de ser incorporados en los procesos de desarrollo de software de las organizaciones. Resulta beneficioso que la educación superior aplique métodos empíricos en el proceso de enseñanza y formación de los estudiantes. Este artículo presenta el resultado de un análisis, desde la perspectiva educativa, de seis experimentos controlados liderados por profesores de la Universidad Nacional de San Juan en asignaturas de Ingeniería de Software pertenecientes a los últimos años de las carreras de Informática de dicha universidad. La utilización de experimentos controlados es una interesante forma de hacer converger las actividades y metas de la docencia con las propias de la investigación, un objetivo siempre buscado en la academia.

1. Introducción

Las técnicas de Ingeniería de Software requieren ser evaluadas experimentalmente si se pretende que tal disciplina crezca en madurez. La Ingeniería de Software es una "gran ciencia"; y la experimentación es un ingrediente necesario de cualquier gran ciencia [1].

Mediante los métodos empíricos es posible evaluar nuevas herramientas, técnicas o métodos de la Ingeniería de Software antes de que éstos sean incorporados en los procesos de producción de las empresas [2].

Los estudios empíricos más utilizados en ambientes educativos son los experimentos controlados [3] en donde la participación del alumno toma relevancia tanto por ser sujeto de los mismos como destinatario de una acción pedagógica.

La formación de profesionales que incorporen nuevos conocimientos basándose en métodos empíricos fiables, como los experimentos controlados, debería ser un objetivo de la educación superior. Habilidades sociales adicionales pueden ser incentivadas en los estudiantes mediante las prácticas grupales experimentales.

El presente trabajo se justifica en la necesidad de obtener un conjunto de indicios o hallazgos respecto de los beneficios de aplicar prácticas pedagógicas que involucren instancias experimentales en temáticas de Ingeniería de Software.

Este artículo presenta el resultado de un análisis, principalmente desde la perspectiva educativa, de seis experimentos controlados liderados por profesores de la Universidad Nacional de San Juan en asignaturas de Ingeniería de Software pertenecientes a los últimos años de las carreras de Informática de tal universidad.

Este análisis pone énfasis en el impacto que estos experimentos controlados tiene en los estudiantes y en los profesores, en su doble rol de docentes e investigadores.

En la siguiente sección se desarrolla una descripción breve de cada uno de los experimentos controlados objeto del análisis que se presenta en este artículo. En la sección 3 se expone el análisis conjunto de los experimentos controlados haciendo especial énfasis en los aspectos educacionales de los mismos. En la sección 4 se presenta una discusión sobre los beneficios y debilidades de estos experimentos, y por último en la sección 5 se detallan las conclusiones y los trabajos a futuro.

2. Descripción de los Experimentos Analizados

A continuación se describen brevemente los seis experimentos objeto de análisis en este trabajo. Como características principales se exponen los objetivos de cada experimento, el tipo de diseño experimental usado, sujetos que intervienen, modelo instruccional aplicado,

contexto curricular utilizado y cantidad de profesores que guiaron el experimento.

2.1. Experimento #1. Comparación de Técnicas de Estimación de Tamaño de Software

Este experimento [4] tuvo la finalidad de comparar dos reconocidas técnicas de estimación de tamaño de software, MK-II [5] e IFPUG-FPA [6]. La comparación principalmente se centró en las características de efectividad, facilidad de aplicación e intención de uso de cada una de las técnicas. Participaron alumnos avanzados de la carrera de grado de Licenciatura en Sistemas de Información de la Universidad Nacional de San Juan y también, en menor número, egresados recientes de esta misma carrera; 36 sujetos en total. El experimento no formó parte del programa oficial de la carrera, aunque estaba vinculado con temáticas dictadas en la asignatura Calidad de Software de la misma; los profesores de dicha asignatura, en un total de tres, fueron los que lideraron la ejecución del experimento, el cual fue presentado bajo la forma de taller de actualización profesional.

El diseño experimental elaborado fue ad-hoc, sin seguir un método de diseño existente. Esto se debió principalmente a que este experimento fue el preliminar de una serie que tenían como objetivo principal cumplir metas educativas y no científicas. No obstante se respetaron criterios generales que buscaban dar fiabilidad a los resultados obtenidos.

Para medir la efectividad de ambas técnicas se utilizó una fórmula basada en el tiempo insumido en la aplicación de la técnica y el tamaño o puntos de función obtenidos al finalizar la aplicación de la misma.

La facilidad e intención de uso de las técnicas se midieron subjetivamente por medio de encuestas a los sujetos participantes.

El modelo instruccional utilizado en este experimento consistió en una sesión teórica inicial, donde se expusieron los conceptos de cada una de las técnicas de estimación a aplicar; y dos sesiones prácticas, la primera introductoria al uso de las técnicas y la segunda en donde se desarrolló la práctica experimental propiamente dicha que finalmente sirvió para realizar los estudios comparativos, foco del experimento.

Evaluando la calidad del trabajo de aplicación de las técnicas realizado por los sujetos se determinó que los mismos habían alcanzado los objetivos de aprendizaje establecidos.

Se pudo apreciar, en los resultados de las encuestas, que la mayoría de los sujetos consideraron que el procedimiento a aplicar para cada una de las técnicas estuvo claro al momento de iniciar la sesión practica, como resultado de un entrenamiento previo adecuado.

En cuanto a la utilidad de las técnicas, muchos de los sujetos consideraron que las mismas son de gran utilidad para la práctica profesional.

Esta fue la primera de la serie de experiencias realizadas, y fue en donde se comenzaron a vislumbrar los beneficios, especialmente pedagógicos, de estas prácticas experimentales.

2.2. Experimento #2. Comparación de Técnicas de Lectura de Requisitos de Software

En este experimento [7] se propuso como objetivo comparar dos técnicas de lectura de requisitos de software usadas durante el proceso de inspección de requisitos de software. Se evaluó la técnica Basada en Perspectivas (PBR por sus siglas en inglés) [8] frente a una técnica ad-hoc de lectura de requisitos. Se compararon las características de eficiencia, eficacia, facilidad de uso e intensión de uso de ambas alternativas.

El despliegue del experimento (diseño experimental) incluyó tres etapas o sesiones; la primera consistió en una sesión de introducción donde se desarrollaron los conceptos teóricos acerca de lectura de requisitos de software. A continuación se llevaron a cabo dos sesiones prácticas en las cuales los participantes aplicaban alguna de las dos técnicas para leer requisitos de software durante un proceso de inspección de requisitos de software.

La muestra estuvo compuesta por alrededor de 40 sujetos, vinculados a la carrera de grado Licenciatura en Sistemas de Información de la Universidad Nacional de San Juan, entre estudiantes avanzados y egresados recientes. Fueron tres los profesores que guiaron el experimento. Al igual que el experimento anterior, éste no formó parte del programa oficial de la carrera aunque tuvo fuerte vinculación temática con algunas asignaturas de la misma, aún así se obtuvo una activa participación de alumnos y egresados quienes se mostraron interesados en aprender y practicar nuevas técnicas de Ingeniería de Software.

El diseño experimental utilizado, basado en la propuesta de Wohlin [9], tuvo dos tratamientos (técnica de lectura PBR y técnica Ad-Hoc) y un solo factor (el proceso de inspección de software), se trató de un diseño experimental simple intra-sujetos, en el que a cada sujeto se le asignó ambos tratamientos.

Las conclusiones obtenidas se basaron en indicadores objetivos obtenidos durante la ejecución del experimento, tales indicadores fueron cantidad de defectos detectados, tiempo consumido, eficiencia (cantidad de defectos detectados por el sujeto sobre el tiempo consumido) y eficacia (cantidad de defectos detectados por cada sujeto sobre la Cantidad Total de Defectos del Documento de Requisitos) por cada técnica. Los resultados que se obtuvieron específicamente respecto a las hipótesis

planteadas y analizadas estadísticamente, fueron los siguientes:

- **Cantidad de defectos detectados:** se pudo comprobar que la técnica de lectura PBR detecta más defectos que la técnica de lectura Ad-hoc.
- **Tiempo insumido:** para esta variable, se pudo revelar que la técnica de lectura Ad-hoc insume menos tiempo en su aplicación.
- **Eficiencia:** los resultados estadísticos evidencian una mejor eficiencia de la técnica Ad-hoc. Es decir esta técnica obtuvo menor tiempo promedio para encontrar un defecto.
- **Efectividad:** en este caso, la técnica de lectura PBR logró una mayor efectividad que la técnica Ad-hoc.

También se obtuvieron indicadores subjetivos basados en encuestas realizadas a los sujetos, específicamente respecto de facilidad de uso, preferencia de uso e intención de uso.

Como puede observarse en (Figura 1), la facilidad de uso promedio de la técnica de lectura PBR es mayor que la de la técnica Ad-hoc.

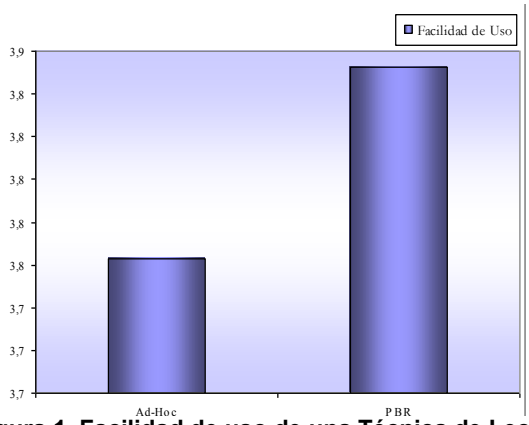


Figura 1. Facilidad de uso de una Técnica de Lectura.

Se estima que la mayor facilidad de uso es justamente debido a que la técnica de lectura PBR guía el proceso, versus la incertidumbre que provoca la técnica, que no tiene procedimiento explícito.

Se observa que 8 de cada 10 inspectores tienen una preferencia hacia la técnica PBR (Figura 2).

La intención de uso muestra a 9 de cada 10 con una predisposición de uso futuro de la técnica de lectura PBR (Figura 3).

Respecto a las variables subjetivas, se pudo observar que por parte de los sujetos experimentales hay una inclinación hacia la técnica de lectura PBR.

Al finalizar el experimento los sujetos participantes demostraron haber alcanzado los objetivos pedagógicos, más allá de los resultados científicos obtenidos en la figura de hallazgos o indicios que deberían ser corroborados mediante replicaciones sucesivas del experimento.

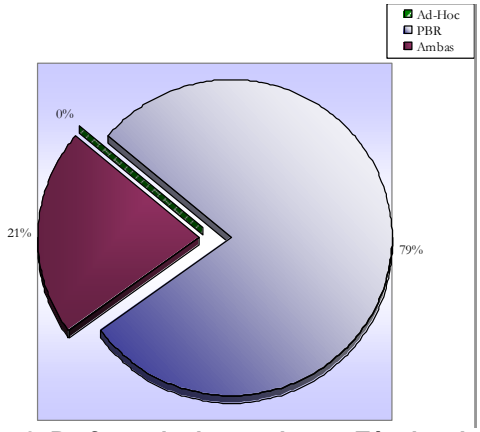


Figura 2. Preferencia de uso de una Técnica de Lectura.

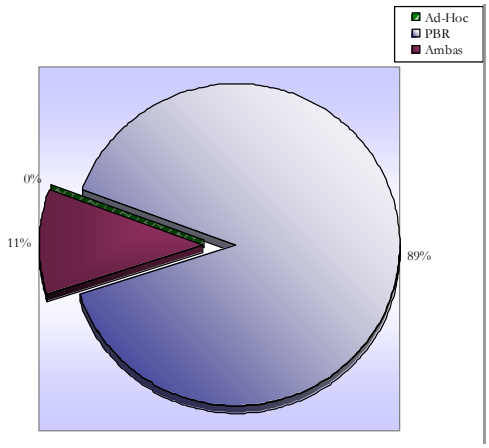


Figura 3. Intención de uso de una Técnica de Lectura.

2.3. Experimento #3. Enseñanza de Ingeniería de Software en un Escenario Distribuido y Colaborativo

En el marco de un proyecto de cooperación científica, que vincula a la Universidad del Cauca (Colombia), Universidad de Chile (Chile) y Universidad Nacional de San Juan (Argentina), se ejecutó un experimento controlado [10] con la participación de alumnos de Informática y Computación de las tres universidades. Los estudiantes participaron de un proceso conjunto de aprendizaje colaborativo aplicando un proceso instruccional, diseñado para tal fin, que utilizó software de comunicación soportado por Internet. La temática objeto del proceso de enseñanza aprendizaje fue Estimación de Tamaño de Software.

Participaron 62 alumnos de los últimos años de las carreras de informática de las universidades participantes; 19 alumnos de Argentina, 22 de Colombia

y 21 de Chile. Seis profesores de las distintas universidades participaron guiando el proceso de aprendizaje distribuido.

El modelo instruccional propuesto para este experimento, como se puede ver en la Figura 4, plantea cinco etapas: Preparación, Clase Teórica, Práctica Local, Práctica Distribuida y Evaluación.

En la preparación se realizaron actividades de planificación, puesta a punto de la tecnología de comunicación a utilizar y aprestamiento de los grupos de alumnos.

Durante la clase teórica, un profesor dictó una exposición introductoria a los estudiantes geográficamente distribuidos utilizando videoconferencia IP; en la práctica local, los estudiantes resolvieron un problema práctico trabajando en forma grupal con sus pares locales; luego, en la práctica distribuida, los estudiantes resolvieron un nuevo problema práctico conformando grupos de trabajo con pares de otras universidades. Por último, en la etapa de evaluación, los estudiantes y profesores realizaron una revisión de la experiencia con el fin de encontrar oportunidades de mejora.

La fase de preparación se realizó en la Universidad Nacional de San Juan, con la participación de profesores de las tres universidades, los cuales acordaron el contenido temático, el despliegue del mismo, las consignas a implementar en las sesiones prácticas, las tecnologías a usar y finalmente el calendario de actividades.

Un profesor de la Universidad de Chile impartió la clase teórica a distancia.

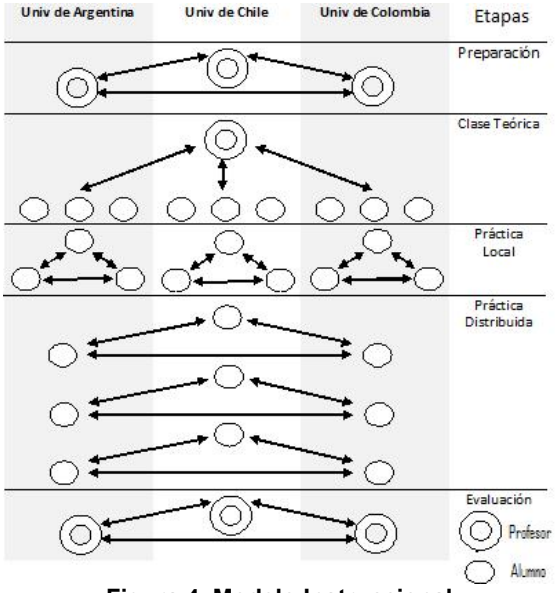


Figura 4. Modelo Instruccional

La tercera etapa, práctica local, se realizó tal cual lo propone el modelo, es decir se conformaron grupos de alumnos pertenecientes a la misma universidad y ellos trabajaron conjuntamente, en forma presencial, para resolver el problema planteado. Un docente investigador local de cada universidad guió esta práctica.

La práctica distribuida, se llevó a cabo a través de sesiones de trabajo distribuidas. Se definieron 21 grupos de trabajo, conformados aleatoriamente por 3 estudiantes, respetando la condición de uno por cada país involucrado en la experiencia.

En general los alumnos se mostraron muy motivados a interactuar con pares de otras universidades y, más allá de las limitaciones que impone la comunicación mediada por las tecnologías, esa motivación se mantuvo durante todo el proceso.

Los estudiantes tuvieron que "negociar" entre ellos para arribar a acuerdos respecto de las tecnologías a utilizar para la comunicación, las formas de resolver el problema, los horarios de comunicación y la distribución de tareas, entre otros. Estas instancias de negociación los obligaron a poner en práctica habilidades y competencias poco usadas por ellos, pero de importancia para su formación profesional.

Por último se realizó la instancia de evaluación de la experiencia, para ello los alumnos contestaron una encuesta de satisfacción general del proceso y otra de co-evaluaciones (evaluación entre pares).

Este experimento fue también altamente positivo para los profesores pues se generaron instancias de discusión y acuerdos entre ellos respecto de la enseñanza e investigación en Ingeniería de Software.

En este caso, la experiencia formó parte del programa curricular de la carrera de grado de la Universidad Nacional de San Juan. La asistencia de los alumnos fue obligatoria y casi el 80% de ellos, según la encuesta realizada, valoró como satisfactoria la experiencia.

La mayoría de los alumnos, el 93%, dijo que aprendieron el tema propuesto, Estimación de Tamaño de Software. Esto se vio corroborado por una evaluación individual final. De manera que el objetivo pedagógico principal fue alcanzado ampliamente

La gran mayoría de los alumnos encuestados, casi el 90%, manifestó que sienten haber enriquecido sus habilidades sociales como proponer y sostener ideas ante pares, participar positivamente en grupos de trabajo y adquirir conocimiento desde pares. Un aspecto negativo de esta experiencia fue las dificultades de operación (interrupción, lentitud, etc.) de las herramientas de comunicación, las cuales impactaron negativamente en el proceso.

2.4. Experimento #4. Comparación de Técnicas de Testing de Software

El objetivo general de este experimento [11] fue obtener una comparación de dos técnicas de testing de

software, focalizando en los aspectos de eficacia y eficiencia de su aplicación. Las técnicas de testing funcionales comparadas fueron Partición de Equivalencia (PE) [12], [13] y Tablas de Decisión (TD) [13].

Participaron como sujetos del experimento 19 alumnos avanzados de la carrera Licenciatura en Sistemas de Información de la Universidad Nacional de San Juan, formando el experimento parte del programa oficial de tal carrera. En este caso fueron dos los profesores que guiaron el experimento.

Se aplicó un diseño experimental, siguiendo las consignas de Wohlin [9], que contó con dos factores y dos tratamientos. El primer factor involucra las dos técnicas de testing; mientras que los dos tratamientos fueron dos programas (códigos ejecutables) que fueron objetos de las pruebas (testing). Cada programa resolvió un problema distinto. Se analizó la aplicación de cada técnica de testing con cada uno de los programas mencionados.

Se implementaron una sesión teórica y dos sesiones prácticas las que se desarrollaron en el transcurso de una semana, acumulando entre las tres sesiones aproximadamente 7 horas de trabajo.

Con este experimento se comprobó, analizando los resultados estadísticos que la técnica PE es más eficiente (Figura 5) y eficaz (Figura 6) en el hallazgo de defectos que la técnica TP, independientemente de los dos tratamientos (programas probados).

En función de las encuestas a los sujetos se observó que el 53% tienen intención de uso de la técnica PE, mientras que TD alcanzó el 26%. Respecto de la facilidad de aplicación casi el 90% de los sujetos encontró más fácil la técnica PE que TD.

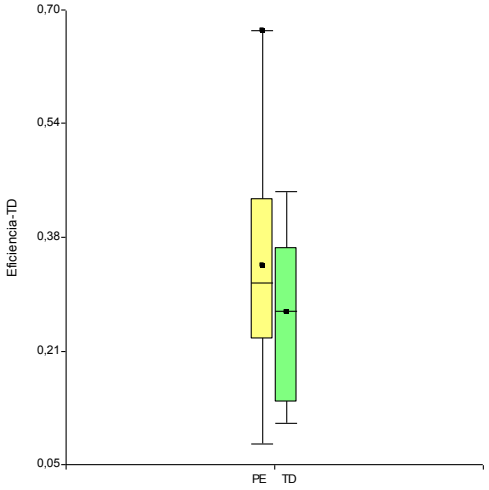


Figura 5. Eficiencia lograda con una y otra técnica

Así este experimento arrojó indicios de interés para la industria del software, específicamente referidos a la efectividad y eficacia de las técnicas de testing. No obstante, es necesario realizar nuevas replicaciones del experimento para afianzar conclusiones.

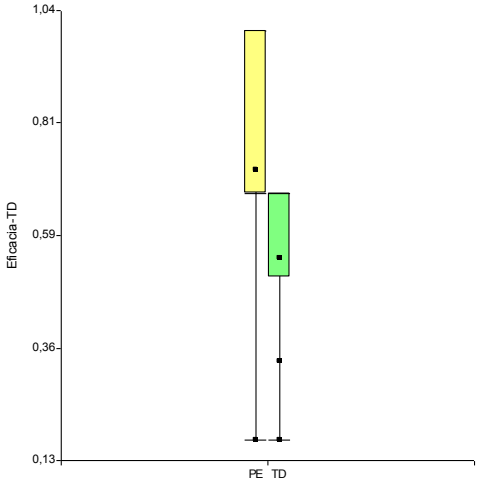


Figura 6. Eficacia lograda con PE y TD

Según las encuestas finales de satisfacción los estudiantes valoraron como "muy buena" la experiencia, destacando que la misma hizo un aporte significativo a su propia formación académica.

2.5. Experimento #5. Comparación de Técnicas de Elicitación de Requisitos de Software en Ambientes Distribuidos

El proceso de desarrollo de software tiende, cada vez con mayor énfasis, a ser distribuido o global, donde los participantes se encuentran dispersos geográficamente. Este escenario obliga a prestar atención a tres aspectos identificados como distancia física, distancia temporal y distancia cultural. Es aceptable sostener que estos nuevos contextos de alguna manera impactarán en el proceso de desarrollo de software, especialmente en aquellas etapas en donde hay exigencias de mayor comunicación y colaboración entre los miembros del equipo distribuido.

En esta experiencia se intentó adquirir un conocimiento más acabado de la etapa de elicitación distribuida de requisitos de software [14], a la vez que se analizó la utilización de escenarios universitarios para llevar a cabo estas validaciones. En forma conjunta con profesores de la Universidad Nacional de San Juan, de la Universidad del Cauca (Colombia) y Universidad de Quindío (Colombia), se planificó un experimento controlado con dos factores: contexto de elicitación (distribuido/colocalizado) y técnicas de elicitación utilizadas (3 distintas combinaciones de técnicas). Así, se ejecutaron dos fases experimentales similares, la primera aplicada a un contexto distribuido de desarrollo de software; mientras que la segunda se aplicó en un contexto tradicional colocalizado (presencial). El diseño

experimental finalmente fue un diseño factorial 2x3 como se muestra en la Tabla 1.

Tabla 1. Diseño Experimental con 2 factores

		Técnicas de Elicitación Aplicadas		
		Técnica 1	***Técnica 2***	***Técnica 3***
Contexto de Elicitación	Fase 1: Distribuido	Grupos de Alumnos Tec1-Dist	Grupos de Alumnos Tec2-Dist	Grupos de Alumnos Tec3-Dist
	Fase 2: Colocaliza-do	Grupos de Alumnos Tec1-Coloc	Grupos de Alumnos Tec2-Coloc	Grupos de Alumnos Tec3-Coloc

Se aplicaron tres combinaciones de técnicas de elicitación de requisitos de software: entrevista/cuestionario, entrevista/brainstorming y entrevista exclusivamente.

Los datos experimentales obtenidos sugieren que la combinación de técnicas más efectiva en un contexto distribuido sería Entrevista-Cuestionario, la cual alcanza un nivel de efectividad superior al 9% respecto de la siguiente combinación de técnicas más efectiva, que fue entrevista-brainstorming. También se advierte que la técnica Entrevista es la menos efectiva de las técnicas evaluadas en este contexto distribuido.

En cuanto a la satisfacción general de los alumnos, en la Figura 7 y en la Figura 8 se muestra la satisfacción de los alumnos en ambos contextos de trabajo. Los datos surgen de encuestas de 5 puntos realizadas a los estudiantes. Los gráficos de frecuencia muestran altos valores de satisfacción en ambos escenarios, aunque en el escenario distribuido el porcentaje de alumnos altamente satisfechos es mayor.

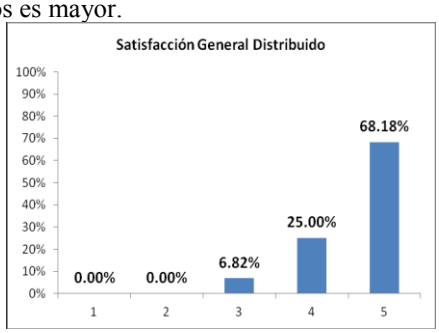


Figura 7. Satisfacción en Contexto Distribuido

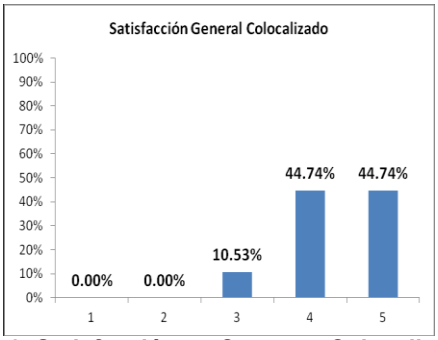


Figura 8. Satisfacción en Contexto Colocalizado

2.6. Experimento #6. Utilización de Herramienta Wiki en el proceso de Elicitación de Requisitos de Software

El éxito de un proyecto de software depende en gran medida de una correcta elicitación de requisitos de software, así mismo, la calidad de estos requisitos mejora en la medida que hay mayor participación de las personas involucradas. Esta participación se dificulta cuando la elicitación de requisitos de software se realiza en un entorno distribuido, es decir, cuando los actores están geográficamente dispersos. Por ello el proceso de elicitación en ambientes distribuidos necesita de una plataforma que promueva y facilite la colaboración efectiva y eficiente de los involucrados, pudiendo ser las herramientas wikis útiles para este propósito. Esta experiencia propone un experimento controlado, realizado para evaluar el uso de una wiki como herramienta de elicitación de requisitos distribuidos. El estudio, que abordó una problemática real, evidenció que en un contexto distribuido y mediante el uso de una herramienta wiki se obtiene un Documento de Requisitos de Software con mayor precisión.

El experimento se realizó en un contexto universitario, en donde los sujetos del experimento fueron estudiantes avanzados y profesores de la carrera de grado en Ciencias de la Computación de la Universidad. Se conformaron 9 grupos o equipos de elicitación integrados por dos estudiantes cada uno, más un profesor que hacía las veces de cliente, en un contexto de elicitación distribuido simulado. Participaron 18 alumnos avanzados y 7 profesores. La asignación de alumnos a los grupos fue aleatoria, teniendo en cuenta que los alumnos tenían perfiles de conocimiento y experiencia previa similares. Igualmente aleatoria fue la asignación de profesores, hubo dos profesores que cumplieron el rol asignado en dos grupos diferentes.

Con el fin de simular el ambiente distribuido, en todo momento la comunicación entre ingenieros de requisitos y clientes se realizó por email, chat, videoconferencia o wiki. En ningún momento los ingenieros de requisitos y los clientes interactuaron cara a cara.

En cuanto a la única técnica de elicitación de requisitos empleada en este estudio, se escogió una de la más usada en pequeñas empresas de software de Argentina, entrevista [15].

En la Tabla 2 se muestran los resultados obtenidos por cada grupo o equipo de elicitación respecto de tres atributos de calidad del Documento de Requisitos de Software: completitud, ausencia de ambigüedad y precisión. Otro índice medido es el grado o nivel de uso de la herramienta wiki que hizo cada grupo.

Tabla 2. Resultados Obtenidos

Grupo	Índice Uso de Wiki	Índice q1 de Completitud	Índice q2 de No Ambigüedad	Índice q3 de Precisión
G1	1,725	0,500	0,800	0,650
G2	1,529	0,540	0,941	0,765
G3	2,345	0,520	0,840	0,800
G4	2,300	0,460	0,700	0,750
G5	2,020	0,600	0,800	0,520
G6	1,618	0,500	0,618	0,382
G7	1,750	0,780	1,000	0,538
G8	1,339	0,520	0,731	0,346
G9	1,283	0,520	0,652	0,174

Haciendo un análisis de correlaciones entre el Índice de Uso de Wiki y cada uno de los índices q1, q2 y q3 se pueden arribar a diversas conclusiones o indicios. No hay indicios que el mayor uso de wiki promueva documentos de requisitos más completos y con menor ambigüedad en ambientes distribuidos. Sin embargo hubo un alto nivel de correlación entre el Índice de Uso de Wiki y q3, lo que despierta la sospecha que el mayor uso de wiki mejora la precisión del documentos de requisitos de software. Este es un indicio que debería ser corroborado en futuras experimentaciones.

La Figura 9 muestra el gráfico de frecuencia de la satisfacción general obtenida de la encuesta realizada a los alumnos que participaron en esta experiencia.



Figura 9. Gráfico de Frecuencia Satisfacción General

Se observa que 81,3% de los resultados obtenidos son mayores o iguales a 4, lo cual significa que la experiencia es altamente satisfactoria desde la perspectiva de los alumnos.

3. Análisis de los experimentos

En esta sección se analizan distintos aspectos importantes de los experimentos controlados objeto de análisis en este trabajo. Especialmente se focaliza en la perspectiva de los alumnos, la participación de los profesores y los aspectos de fiabilidad de los experimentos.

En general los alumnos percibieron como beneficioso la realización de este tipo de actividades experimentales. Lo novedoso, al menos para ellos, del modelo instruccional aplicado los predispone de mejor manera para participar; lo nuevo despierta en ellos motivaciones y actitudes positivas hacia el aprendizaje. Muy probablemente lo anterior colaboró, aunque no excluyentemente, para obtener un alto nivel de satisfacción, como la mayoría de los alumnos manifestaron en las encuestas finales.

El alto componente práctico de las actividades es uno de los aspectos que más satisfizo a los alumnos; esto probablemente se debe a que los estudiantes eran de cursos avanzados y en esas instancias de sus carreras académicas, próximos a la salida laboral, priorizan la adquisición de habilidades prácticas respecto de conocimientos teóricos.

Los alumnos valoran positivamente la interacción grupal para la resolución de problemas, esta valoración se enfatizó cuando la interacción se realizó con pares de universidades extranjeras. Las diferencias culturales y sociales enriquecieron la formación de los alumnos.

De las encuestas a los alumnos se infiere que los mismos pasaron por instancias de discusión con pares, que los llevaron a poner en práctica y aprender habilidades como la negociación de acuerdos, sostener argumentos, analizar argumentos ajenos, aceptar idiosincrasias distintas, etc. Estas habilidades "blandas" se estiman de gran importancia en la formación profesional de los alumnos más allá de la adquisición de conocimientos técnicos.

En los experimentos en donde se aplicaron estrategias de aprendizaje colaborativas no se pudo medir efectivamente en qué medida estas estrategias aportan al aprendizaje. No obstante, se presume que las mismas son de importante utilidad y se hará un seguimiento más preciso en próximos experimentos para validar estas presunciones. En este sentido hay que remarcar que intentar que estas actividades experimentales incorporen prácticas colaborativas implica un esfuerzo adicional significativo en el diseño del experimento y no siempre es posible lograrlo, por las características propias de las temáticas abordadas.

Los profesores deben realizar un esfuerzo adicional significativo en la preparación, ejecución, y análisis de resultados de los experimentos controlados. Se debe tomar especial cuidado en las actividades del diseño experimental, clave para obtener resultados fiables y factibles de generalizar. La selección de los sujetos (alumnos y/o egresados), el tamaño de la muestra, la aleatoriedad en la asignación de los tratamientos, las

actividades previas de inducción (clases teóricas), el despliegue de las actividades prácticas experimentales, la definición y medición de variables experimentales y la encuesta final de satisfacción son los aspectos principales a tener en cuenta en el diseño del experimento. En los experimentos analizados se aplicaron técnicas ad-hoc como también las propuestas de Wohlin [9] para definir el diseño experimental; siendo estas últimas más complejas de implementar pero de mayor confiabilidad respecto de los resultados estadísticos obtenidos.

Los profesores advierten de lo beneficioso de este enfoque experimental, aunque coinciden en que el mismo no es aplicable a todos los temas de los programas curriculares, sólo a aquellos que tienen mayor independencia con otros temas y además alta factibilidad de ser llevados a la práctica. Las temáticas abarcadas en los experimentos analizados en este trabajo fueron: estimación de tamaño de software con técnicas de puntos de función, técnicas de lectura de requisitos basada en perspectivas, técnicas de testing y finalmente técnicas y herramientas de elicitación de requisitos de software.

En los experimentos analizados en este trabajo el modelo instruccional aplicado consta de una sesión teórica y dos o tres sesiones prácticas, habiéndose observado que el mismo es muy factible y eficaz en su aplicación. La duración de los experimentos no superó la semana, excepto en un caso, experimento en el que participaron varias universidades, que demandó tres semanas debido, principalmente, a las dificultades de coordinación y comunicación de los sujetos. Así, la extensión temporal del dictado de estas temáticas bajo la modalidad de experimentos controlados es similar a la insumida cuando estos mismos temas se imparten en una modalidad tradicional no experimental.

Se han observado moderadas mejoras en los niveles de aprendizaje, comparando evaluaciones realizadas a los alumnos que trabajaron con procesos instruccionales tradicionales con evaluaciones realizadas bajo esta nueva modalidad experimental. No obstante, uno de los mayores beneficios de los experimentos controlados analizados reside en el aprendizaje de habilidades sociales por parte del alumno, aspecto complejo de medir objetivamente mediante un indicador cuantitativo. Las encuestas finales a los alumnos confirman esta afirmación.

Excepto en el experimento #1, referido a técnicas de medición de tamaño de software, en el resto de los experimentos se alcanzaron indicios significativos sobre técnicas aplicadas en la Ingeniería de Software; especialmente referidas a efectividad, eficacia y facilidad de uso de las mismas. Estos resultados son fiables estadísticamente, más allá que por ahora son sólo indicios. Este es un aporte importante para la industria del software.

4. Discusión

Claramente se observa que estos experimentos controlados son útiles tanto para los objetivos pedagógicos de la enseñanza de la Ingeniería de Software como para obtener indicios empíricos de valioso aporte científico a la disciplina. Ahora bien, para que estos beneficios se hagan realidad un esfuerzo adicional, al menos inicial, debe ser realizado por parte de los cuerpos docentes. Especialmente en una correcta definición del diseño experimental, cuya importancia es relevante para la obtención de resultados empíricos fiables, requiriendo conocimientos específicos sobre Ingeniería de Software Experimental.

La elaboración de paquetes experimentales reutilizables que puedan distribuirse y aplicarse en distintas universidades, mediante replicaciones, no sólo disminuiría considerablemente el esfuerzo docente antes mencionado, sino que además potenciaría fuertemente los beneficios científicos de estas actividades experimentales al contar con mayor cantidad de muestras para analizar.

La replicación de paquetes experimentales fortalecerá las conclusiones científicas a que se arriben después de aplicar y analizar los experimentos. En este sentido la conformación de redes de universidades es una acción que puede impulsar fuertemente esta iniciativa.

Para alcanzar la replicación de estos experimentos se deben diseñar paquetes experimentales que sean lo suficientemente generales para soportar los posibles escenarios candidatos de aplicación, siendo esto un desafío futuro relevante. Las universidades latinoamericanas son un potencial dominio de aplicación de estas replicaciones.

Las temáticas de Ingeniería de Software involucradas en los experimentos fueron cuidadosamente seleccionadas, observando que no todas las temáticas son factibles de aplicar. Temáticas de corto despliegue pedagógico y de alto contenido práctico son las más adecuadas para estas actividades experimentales. Las fuentes temáticas no sólo debieran ser los programas curriculares de las asignaturas de las carreras universitarias, sino también las necesidades de la propia industria del software. Esto último favorecería la vinculación entre la universidad y las empresas productoras de software.

Los alumnos, actores claves del sistema de educación superior, tuvieron una postura favorable hacia los experimentos, según los resultados de las encuestas de satisfacción realizadas. Ellos valoran, de los experimentos, su énfasis en las actividades prácticas, las interacciones con pares de otras universidades y la claridad del modelo de enseñanza aprendizaje. También expresan que es relevante para la exitosa ejecución del experimento un plan de ejecución ordenado y controlado, tecnología de comunicación (chat, email,

videoconferencia, etc.) eficiente y alta vinculación temática entre la teoría y la práctica propuesta.

No se observa que los costos monetarios de este modelo instruccional experimental de Ingeniería de Software sean inaccesibles para universidades latinoamericanas. Los recursos necesarios son los estándares con los cuales ya cuentan estas universidades, es decir, acceso a internet, laboratorio de computadores, proyector y tecnología de comunicación IP estándar.

Un aspecto interesante a evaluar es la incorporación de profesionales de la industria del software como sujetos de estos experimentos. No se han advertido resultados distintivos en aquellos experimentos analizados en donde han participado graduados. Tal vez esto se debe a que estos profesionales eran de reciente graduación y la experiencia profesional era similar a la de alumnos avanzados participantes en los experimentos.

Los experimentos controlados llevados a cabo en ambientes industriales son más onerosos, principalmente por el costo del recurso humano involucrado, que aquellos llevados a cabo en ambientes educativos. Una ventaja de los experimentos controlados, como los presentados, es que ellos pueden ser ejecutados repetidas veces en ambientes universitarios hasta alcanzar la madurez de los mismos, momento en el cual pueden ser transferidos a la industria del software para su aplicación en dichos escenarios. Esto incrementa las probabilidades de éxito de los experimentos en ambientes más costosos como los señalados.

5. Conclusiones y trabajo a futuro

La utilización de experimentos controlados es una interesante forma de hacer converger las actividades y metas de la docencia con las propias de la investigación, un objetivo siempre buscado en la academia. Es una forma de reunir en una actividad ambos roles de los profesores, docencia e investigación.

La motivación de los alumnos en estos experimentos controlados es un impulso importante que impacta muy positivamente en el aprendizaje, no sólo de aspectos técnicos sino también de habilidades sociales valiosas. También es relevante destacar que los alumnos adquieren, con la aplicación de estos experimentos, los principios de la Ingeniería de Software Empírica que los guía a aplicar mayor rigurosidad científica al momento de incorporar nuevos métodos, técnicas o herramientas en el proceso de Ingeniería de Software.

El modelo instruccional aplicado en los experimentos, basado en sesión teórica y dos sesiones prácticas, ha dado buenos resultados. Sumo cuidado se debe tener en el diseño experimental, en las herramientas de comunicación a utilizar en el caso que hicieran falta y en la incorporación de prácticas colaborativas distribuidas.

Los indicios o hallazgos alcanzados en cada experimentación son un importante aporte a la industria del software, aunque la realización de replicaciones de esas experimentaciones dará mayor validez y generalización a las mismas.

La conformación de bibliotecas de experimentos controlados en Ingeniería de Software aplicables a ambientes educativos parece ser el desafío próximo, especialmente para las universidades latinoamericanas que cuentan con las condiciones adecuadas para conformar redes experimentales en esta disciplina.

6. Agradecimientos

A los alumnos, de distintas cohortes, y profesores del programa de grado en Ciencias de la Computación de la Universidad Nacional de San Juan. A los profesores del grupo de investigación del Instituto de Informática de la misma universidad. A los estudiantes y profesores de distintas universidades latinoamericanas.

7. Referencias

[1] Basili V., "The Past, Present, and Future of Experimental Software Engineering", *Journal of the Brazilian Computer Society (JBCS)*, 2006.

[2] Host, M., "Introducing Empirical Software Engineering Methods in Education", in *Proceedings of the 15th Conference in Software Education and Training (CSEET'02)*, pp. 170-179, 2002.

[3] Baresi, L., Morasca, S. and Paolini, P., "Estimating the Design Effort of web Applications", in *Proceedings of the 9th International Software Metrics Symposium (METRICS'03)*, pp. 62-71, 2003.

[4] Zapata, Sergio G., Gomez, Claudia P. , Herrera, Myriam, Ruiz, Susana, Torres, Estela y Lund, M. Inés, "Diseño Experimental Para Comparar Métodos De Estimación Del Tamaño Funcional Del Software", *X International Conference on Engineering and Technology Education*, São Paulo, Brasil, 2008.

[5] MK II, *Function Point Analysis Counting Practices Manual*, United Kingdom Software Metric Association, Version 1.3.1, 1998.

[6] Abrahão, S. and Pastor, O., "Estimating the Applications Functional Size from Object-Oriented Conceptual Models", in *International Function Points Users Group Annual Conference (IFPUG'01)*, Las Vegas, USA, 2001.

[7] Lund, M. Inés, Herrera, Myriam, Aballay, Laura y Zapata Sergio, *Comparación de Técnicas de Lectura de Documentos de Requisitos de Software: Diseño de un Experimento*, Revista Avances en Sistemas e Informática, Vol.4 No. 3. ISSN 16577663. pp. 117 - 129, Medellín Colombia, 2007.

[8] Basili V., Green S., Laitenberger O., Lanubile F., Shull F., Sørumgård S. and Zelkowitz M.V. et al. "The Empirical

Investigation of Perspective Based Reading", *Empirical Software Eng. - An Int'l J.*, Vol. 1, No. 2, pp. 133-164, 1996.

[9] Wohlin, Runeson, Höst, Ohlsson, Regnell, Wesslén, *Experimentation in Software Engineering: An Introduction*, Kluwer Academic Publishers, 2000.

[10] Lund, María Inés, Zapata, Sergio, Aballay, Laura, Herrera, Myriam, Torres, Estela, Collazos, César, Giraldo, Fáber y Ochoa, Sergio, *Ingeniería de Software para Ambientes de Aprendizaje Distribuidos*, Revista Avances en Sistemas e Informática, ISSN: 1657-7663 Medellín - Colombia, 2009.

[11] Soto, Ariadna y Sánchez, Pablo, *Comparación de Técnicas de Testing de Software*, www.idei.unsj-cuim.edu.ar\tesis\tesis_sotosanchez.pdf

[12] Myers, G., *The Art of Software Testing*, New York, Wiley, 1979.

[13] Mosley, D., *The Handbook of MIS. Application Software Testing*, ISBN 0-139-07007-9, PrenticeHall, 1993.

[14] Zapata, Sergio, Torres, Estela, Sevilla, Gustavo, Aballay, Laura y Reus, Marcelo, "Eficacia de Técnicas Tradicionales de Elicitación de Requisitos de Software Aplicadas en Escenarios Distribuidos de Desarrollo de Software", *Conferencia Latioamericana de Informática. Simposio de Ingeniería de Software*. 2013.

[15] Antonelli, L. y Oliveros, A., "Fuentes Utilizadas por Desarrolladores de Software en Argentina para Elicitar Requerimientos", *5th Workshop on Requirement Engineering*, (WER'02) Noviembre, 11-12, 2002, pp. 106–116.

Una Propuesta Pedagógica de Adecuación a Tendencias Emergentes del Uso de las TICs en la Universidad

Mirta Peñalva, Stella Maris Calderón, Ernesto Girbal, Carlos Gardella
Departamento de Ingeniería en Sistemas de Información
Facultad Regional La Plata, Universidad Tecnológica Nacional
Av.60 esq. 124 s/n - CP 1900 - La Plata - Buenos Aires
E-mail:mirpenalva@hotmail.com, stella_m_calderon@yahoo.com.ar, ergirbal@yahoo.com.ar, hugarde2002@yahoo.com.ar

Abstract

La actual sociedad del aprendizaje con su dinamismo, crecimiento y veloz depreciación de los conocimientos, exige a los actores del sujeto pedagógico estar en activa observación, evolución de sus saberes y habilidades. Este paradigma exige un profundo compromiso del docente proponiendo intervenciones pedagógicas adecuadas para andamiar procesos cognitivos que no sólo involucren aprendizaje disciplinar sino la auto-gestión y macrohabilidades. El docente deberá co-evolucionar a la par de las TICs para sostener un proceso que incluya mecanismos y estrategias que le faciliten al alumno adquirir habilidades para enfrentar escenarios cambiantes que lo desafían a actuar en situaciones inéditas. Es indiscutible el impacto que ha logrado el desarrollo de las TICs en casi todas las actividades humanas, particularmente nos enfocaremos a nuevos hábitos de búsqueda, lectura, procesamiento e intercambio de información digital en los ámbitos universitarios de grado. Hemos identificado nuevas formas de lectura en los estudiantes, lo que empuja al docente a reflexionar sobre la flexibilidad de sus propuestas didácticas ante la diversidad de medios y fuentes de información. En este artículo se aborda el estado del arte sobre los hábitos de lectura, se plantean interrogantes y se detalla una experiencia realizada con alumnos de segundo año de la carrera de Ingeniería en Sistemas de Información, en la cual hubo tareas de exploración de repositorios digitales en Internet, selección de artículos académicos, lectura, comprensión, y abstracción modelando mapas conceptuales. Posteriormente se relatan los resultados obtenidos de la experiencia como así también las conclusiones, extensiones y acciones a futuro.

Palabras claves: TICs, información digital, aprendizaje, mapas conceptuales, modelado.

1. Introducción

El conocimiento es una relación entre un sujeto y un objeto, en constante revisión desde hace más de dos mil años. Dilucidar cuál es el objeto, quién es el sujeto y de qué tipo de relación se trata, ha sido el objetivo de diversos enfoques que enuncian que el objeto de conocimiento puede ser considerado práctico o teórico, experiencial o racional, sustantivo, procedimental o comunicacional y el sujeto mismo puede tratarse de un ser individual, grupal, colectivo o social, incluso puede ser una organización, región o país. Con respecto a la relación, se presenta como un tipo de actividad que entrelaza lo racional con planos motores y sensibles sostenida por intereses, deseos y emociones [15]. Esto nos lleva a denotar la complejidad de la gestión de la enseñanza y del aprendizaje, su diversidad, riqueza y que se encuentra en constante resignificación.

Inmersos en la sociedad del conocimiento, devenida en sociedad del aprendizaje [12], tomamos conciencia del crecimiento exponencial del conocimiento y su alto grado de obsolescencia, que lo destaca no ya como producto, sino como proceso que nos conduce a la educación permanente.

Este paradigma exige al docente intervenciones pedagógicas adecuadas para andamiar procesos cognitivos que no sólo involucren lo disciplinar sino que optimicen el proceso mismo a través del auto-conocimiento y control (metacognición) [1]. La masividad en el uso de nuevas tecnologías y comunicación global TICs[1], va modificando las formas en las cuales el hombre se expresa, interactúa y aprende.

[1] La tecnología se desarrolla con mayor velocidad que los procesos de transformación y aprendizaje social, debemos ocuparnos de disminuir esta asimetría señalada en [8]: "No se puede asir la herramienta moderna y tener una mente antigua". La coevolución del usuario y las TICs, ha elevado los requerimientos de calidad Totalmente convergente con Ludvall [12] "…aprender a enfrentarse y usar el potencial completo de las nuevas tecnologías es lo que permite transformarse de "viejos a nuevos".

Es esencial la co-evolución del docente [8] con las nuevas tecnologías para revalidar su autoridad pedagógica y contribuir a una efectiva mediación en el proceso de aprendizaje del alumno. Crear y ejercitar estrategias y mecanismos son actividades fundantes del proceso de adquisición de macro-habilidades requeridas para enfrentar escenarios futuros que desafían las capacidades.

Con la expansión de información en Internet, sin calificación formal de sitios, se han creado ciertos mitos que hay que manejar con cautela tales como: *todo está allí* (con volumen inmanejable y desconociendo los criterios de prioridad utilizados por motores de búsqueda, no hay jerarquización de la información), el *costo "cero" y facilidad de obtención* no se debe confundir con que lo primero que se encuentra es lo mejor, *el enlace entre páginas* no siempre se debe entenderse como cohesión temática.

Adicionalmente, la conceptualización colectiva del verbo *googlear* acuñado como sinónimo de *buscar,* va más allá, acoplándole las propiedades de ser efectiva, barata, fácil, rápida y neutral [5] nos alerta a ser críticos y no quedar atrapados por la *metáfora de las lentes*[2].

Conjuntamente con la disponibilidad y abundancia de información desestructurada en la Web, se pone de manifiesto la complejización del proceso de extracción de piezas de valor en relación a nuestros objetivos. Ante esto se ha abierto el camino hacia la *búsqueda inteligente* ya que según afirma el autor en [14] "no es lo mismo recuperar información que recuperar datos".

El presente artículo se refiere al tratamiento de una propuesta de acciones de mejora del sistema que conforman la enseñanza y el aprendizaje ante la detección de nuevos hábitos de lectura y abordaje de objetos de enseñanza.

2. Motivaciones

Durante el desarrollo de encuentros pedagógicos de nuestra asignatura hemos detectado una merma significativa en la intención de uso de la biblioteca institucional por parte de los alumnos y el aumento de lectura de archivos bajados de Internet. Este hecho se constató con la opinión de referentes de la biblioteca en forma preliminar. Con la solicitud de compra anual de libros de la asignatura ya gestionada, continuamos con la investigación acerca de la posibilidad de comprar la bibliografía en formato digital. Encontramos que las editoriales de textos de grado no editan en dicho formato adhiriendo al derecho de autor. Avizoramos cambios con respecto a la futura forma de autoridad intelectual y de la producción literaria mundial según lo discute el autor en [16], aunque no vigentes en este momento.

3. Estado del arte

Ante las circunstancias previamente descriptas, nos propusimos indagar espacios en los cuales se hayan estudiado más profundamente los hábitos de lectura.

Encontramos basta documentación bibliográfica como: Encuesta de hábitos de lectura en Argentina 2011[17], Encuesta de hábitos de lectura y compra de libros en España 2012 [18] y Comportamiento lector y hábitos de lectura en América Latina 2011 [19]. Todo este material fue analizado para entender la problemática y encontrar puntos comunes con las costumbres de nuestros alumnos. Se identificaron en estas publicaciones una visión convergente ya que todas apuntaban a la indagación del comportamiento del lector y las formas de acceso al libro.

En Argentina, según lo expresado en [17], la lectura en pantalla amplía la cantidad de lectores, incluso en papel. Esta conclusión se basa en la duplicación de lectores en medio digital mayores de 18 años entre 2001 y 2011, que pasaron del 21% al 44%. Una observación destacable es que una vez que se atraviesa el umbral de la lectura ya sea en papel o pantalla se tiende a leer más en ambos soportes. El análisis de grupos etáreos muestra que los jóvenes leen distinto y que la lectura en pantalla está fuertemente ligada a la edad. El segmento de 18-25 años alcanza al 73% de lectura digital (exclusivamente) como lo muestra la Figura 1:

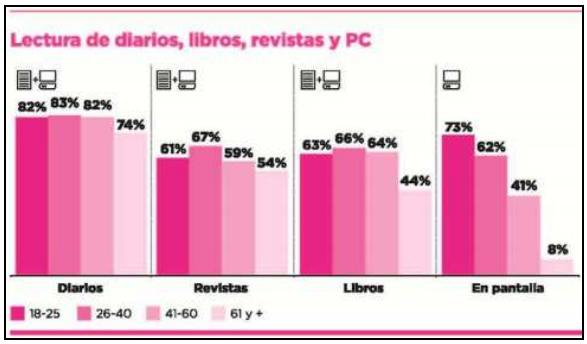


Figura 1- Porcentajes de lectura digital por grupo etáreo. Fuente:Encuesta 2011- Consejo Nacional de Lectura (Argentina)

Respecto a la concurrencia a bibliotecas el informe revela que las mismas son consideradas como el resguardo de la educación y la cultura pero *"no son visitadas"*.

En España, según lo analizado en [18], el 58% de la población mayor a 14 años lee en soporte digital. La

[2] Según Pierre Bourdieu, las lentes definen lo que se ve y lo que no se ve.

evolución de este segmento entre 2010 y 2012 fue de 10 puntos, subiendo del 48% al 58% como se muestra en Figura 2. Se observa que la tendencia de la evolución es creciente en ambos países.

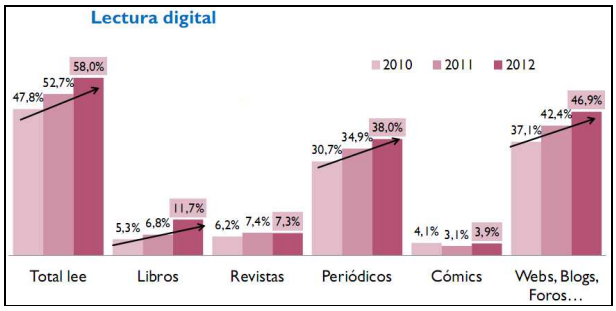


Figura 2- Gráfico de evolución porcentual de lectura digital. Fuente: Hábitos de lectura y compra de libros en España 2012

Complementariamente los resultados indican que el 90,4% de los estudiantes lee en soporte digital. Esta medida es congruente con lo percibido en nuestras aulas. Respecto de cómo obtienen los libros digitales, el 64,4% respondió que los baja gratuitamente de Internet y el 38% que los consigue por copia (USB, mail, CD), lo que muestra que se antepone el acceso por sobre la valoración del derecho de autor. Con respecto al uso de bibliotecas, 70% de los entrevistados *no las usan*. Más adelante veremos que nuestras investigaciones locales arrojaron el mismo valor de indicador. Un dato para tener en cuenta es que los menores de 10 a 13 años buscan información o documentación en Internet diariamente en un 74% por lo que vemos que este medio se consolida como concentrador mayoritario del acceso a la información.

En Latinoamérica la CERLALC[3], desde 2002 promueve *No se quede por fuera del mapa*, un proyecto que investiga la diversidad de modelos, problemas, concepciones, métodos de trabajo y prácticas sociales para construir un mapa de acciones y políticas de desarrollo de la lectura en países hispanoparlantes. De su última versión 2012 se desprende que Argentina y España tienen órdenes de no lectores cercanos del 45% y 39% respectivamente, mientras que el gusto por leer alcanza el 70% y 86 % para los mismos países. La lectura por motivos de exigencia escolar en Argentina se encuadra en el 32% mientras que España un 7%. La razón predominante de no lectura es la falta de tiempo. En España representa un 51% mientras que en Argentina el panorama es diferente: el mayor motivo es el desinterés, seguido por problemas económicos y en tercer lugar por falta de tiempo. En contraposición, en Argentina y España es donde se elige usar el tiempo libre para leer en un 66% y 58% respectivamente, mientras que en otros países la mayor competencia es el medio audiovisual. Respecto de cuántos libros promedio se leen España lidera con 10,3 mientras que en América latina le siguen Chile con 5,4 y Argentina con 4,6 per cápita.

Con respecto al acceso al libro a través de la biblioteca/escuela la encuesta muestra que en Argentina solo el 10% toma préstamos en ellas.

En Figura 3 se muestra cuadro comparativo:

	No lectores	*Gusto por la lectura*	*Leer por exigencia escolar*	*Lectura en tiempo libre*	*Libros per cápita*	*Lec.digital Edad:18-25*	*Uso de bibliotecas*
Argentina	*45%*	*70%*	*32%*	*66%*	*10,3*	*73%*	*10%*
España	*39%*	*86%*	*7%*	*58%*	*4,6*	*90,4*	*30%*

Figura 3 – Cuadro comparativo de indicadores de lectura entre Argentina y España. Fuente:CERLALC

La recopilación de estos estudios posiciona al lector argentino en general, como interesado y con buen hábito. En los segmentos poblacionales más jóvenes coexisten por un lado una tendencia creciente hacia la lectura en medios digitales bajados de Internet, y por otro una marcada disminución del uso de las bibliotecas. Con respecto a este último punto, se deja entrever que en algunos países de Latinoamérica con menor poder adquisitivo, el libro se considera un bien social y se adhiere en mayor medida al préstamo y su rotación.

4. Hipótesis

Luego de identificar nuevas tendencias en los hábitos de lectura de nuestros alumnos concordantes con el contexto regional, nos planteamos los siguientes interrogantes acerca: ¿En qué medida los alumnos asisten a la biblioteca? ¿Leen material en papel o en digital y en qué proporción? ¿El material que leen es de calidad? ¿Las búsquedas en Internet son eficaces? ¿Eligen el material? ¿Leen en otros idiomas? Consideramos que las respuestas de estas preguntas son centrales para un efectivo proceso de enseñanza que deberá ir regulando y adecuando las prácticas docentes para sostener y fortificar el proceso de aprendizaje de los estudiantes. Estas deberán acompasarse a las políticas públicas acerca de lectura y de acceso equidistante al conocimiento a través del libro.

5. Investigación de campo

Teniendo en cuenta las circunstancias contextuales descriptas decidimos llevar adelante un proceso de

[3] Centro Regional para el Fomento del Libro en América Latina y el Caribe

investigación formal con el fin de corroborar o desestimar nuestras hipótesis. Para lo cual realizamos una encuesta a los alumnos para ampliar las opiniones y cuantificar lo percibido.

A continuación detallaremos la metodología utilizada en esta instancia: la encuesta fue dirigida a una población de alumnos de segundo año de la carrera de Ingeniería en Sistemas de Información. El tamaño de la muestra fue n=31 de un total poblacional de N=150. La proporción seleccionada para el cálculo del error fue p=0,4839, mientras que el error de muestreo aleatorio máximo fue de 17,59% con un nivel de confianza del 95%. La toma de datos se realizó a fines del primer cuatrimestre de 2014.

Se diseño una encuesta para cuantificar y evaluar los aspectos de la realidad en estudio, denominada *Hábitos de Acceso y Lectura Universitaria*. La misma se implementó digitalmente con la asistencia de la herramienta SurveyMonkey. Se invitó a los cursos a que respondieran la encuesta especificando claramente que era no obligatoria y anónima. Las métricas asociadas a las preguntas de la encuesta fueron:

Métricas de Escala Numérica y Continua:

1. "Proporción de preferencia del soporte papel para leer material de la cátedra"
2. "Proporción de lectura en inglés como segundo idioma"
3. "Proporción de alumnos que poseen carnet de lector de la biblioteca de la facultad"
4. "Proporción de alumnos que han sacado de biblioteca al menos un libro en su carrera"
5. "Proporción de alumnos utilizan otras bibliotecas"
6. "Proporción de alumnos que compran libros relacionados a la carrera"

Métricas de Escala Categórica:

7. "Frecuencia de uso de material fotocopiado"
   Valores: *poco frecuente, frecuente, muy frecuente*
8. "Forma de acceso al material de lectura más utilizado" Valores: *libro, fotocopias, Internet, otros (apuntes)*
9. "Deficiencias más frecuentes del material de lectura bajado de Internet"
   Valores: *Caso de páginas (Sin acreditación, Sin fuente que la elaboración, Sin fecha de vigencia, Con errores de escaneo por OCR, Versiones obsoletas, Traducción de baja calidad, Falta de citas bibliográficas, Elaboración por copy&paste, Errores de ortografía y sintaxis, Traducciones automáticas, Sitios no académicos, Deficiencia en búsquedas, Lectura dispositivos pequeños, Errores de escaneo a imagen.)*
10. "Ventajas identificadas en el material digital y su bajada de Internet" Valores: *Búsqueda automática, Fácil copiado, Bajos gastos insumos, Manejo grandes volúmenes, Almacenamiento en la nube, Lectura en micro-dispositivos, Inmediatez, Uso Wi-fi institucional, Tratamiento de imágenes, Afinidad al formato, Material inalterable.*

6. Resultados

Los resultados de la encuesta indicaron que existe todavía una fuerte preferencia por la lectura en papel en 71 % de los casos y el resto en digital, el idioma de lectura predominante (excepto castellano) es inglés en el 100% de los casos, seguido por una menor proporción del 4% en portugués y chino. Nos ha sorprendido el bajo porcentaje de alumnos que son socios de la biblioteca institucional representando sólo un 20% (coherente con última columna de Figura 3). Esta circunstancia se refuerza con el 85% de encuestados que nunca sacaron un libro durante los años de cursada, lo que adicionalmente muestra la baja actividad de uso de la biblioteca. Destacamos la necesidad de difundir esta evidencia a un plano superior para que las instituciones tomen acciones compensadoras reviendo alternativas de uso y diversificando sus repositorios de información.

Por otra parte, consultamos a los alumnos si tomaban préstamos de libros en otras bibliotecas, la respuesta fue afirmativa sólo en el 10% de los casos. Respecto de la compra de libros relacionados a la ingeniería, respondieron afirmativamente el 13%.

A continuación consultamos sobre el soporte más usado que resultaron ser "las fotocopias" en el 94% de los casos mientras que el "material bajado de Internet" se ubica en segundo lugar con una preferencia del 71%.

A partir de este último indicador indagamos el tipo de material que los alumnos leían bajado de Internet, y descubrimos que adolecía de problemas que perjudicaban el proceso de aprendizaje. En la Figura 4 se detallan dichas características relevadas como desventajosas. De la lista de opciones sugeridas las alternativas "*No se disponía de la fuente que la elaboró"* y "*Material reelaborado por técnica de copy & paste"* fueron seleccionadas en aproximadamente el 50% de los casos, mientras que "*Material bajado de páginas sin acreditación"*, "*No se disponía de fecha de elaboración o vigencia" y "Los sitios de bajada frecuente son no académicos"* fueron marcados por un 40% de la muestra. Con un nivel de elección del 20% se ubican las opciones *5, 9, 10, 12, 13, 1. 4* de la Figura 4.

DESVENTAJAS	
1.	Material bajado de páginas sin acreditación
2.	No se disponía de la fuente que la elaboró
3.	No se disponía de fecha de elaboración o vigencia
4.	El material escaneado por OCR tenía errores que dificultaba la lectura de corrido
5.	Los libros bajados en formato digital si bien correspondían a los autores definidos en el programa eran de versiones antiguas
6.	Los libros en castellano provenían de traducciones de baja calidad
7.	El material no contenía las citas bibliográficas que pudieran solventar pertinencia y calidad
8.	Material reelaborado por técnica de copy & paste
9.	Material con errores ortográficos y construcciones sintácticas deficientes que entorpecen la comprensión
10.	Utilización de traductores automáticos de idiomas que no logran textos comprensibles
11.	Los sitios de bajada frecuente son no académicos
12.	Búsquedas simples en la Web (sin especialización) devuelven información de baja calidad y relevancia
13.	Dificultosa lectura en pantallas de tamaño reducido desde dispositivos móviles
14.	Algunos presentaban escaneo descuidado y de baja calidad

Figura 4- Lista de deficiencias encontradas en textos digitales tomados de Internet

Por último investigamos los beneficios por los cuales preferían el material digital, las apreciaciones se detallan en la Figura 5 y se destaca *"Posibilidad de búsqueda automática en el texto"* con el 75% de adhesión.

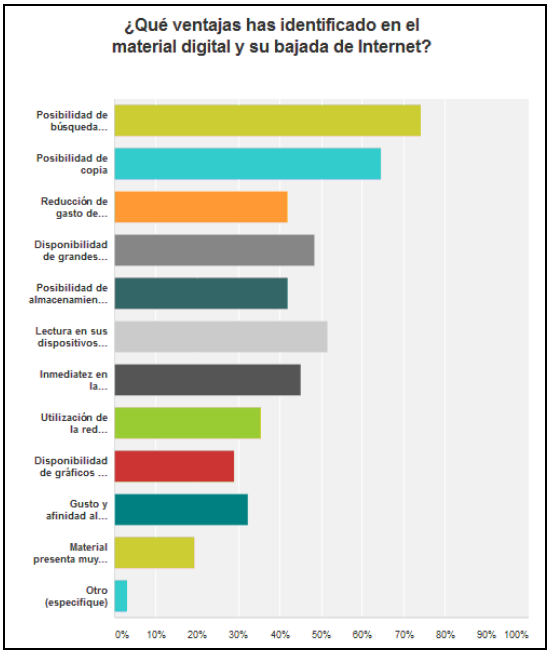


Figura 5- Lista de aspectos de calidad percibidos por alumnos

Seguido por *"Posibilidad de copia" con un 65%, "Lectura en sus dispositivos móviles" y "Disponibilidad en ...Pendrive"* cercanos al 50% e *"Inmediatez en la disponibilidad*", "*Reducción de gasto de toner/tintas*" y "*Almacenamiento en la nube"* lograron un 45%.

El resto de beneficios obtuvo muy buena marca superando el 20%.

A modo síntesis podemos describir nuestro grupo estudiantes encuestados como: mayoritariamente no asisten ni solicitan libros de bibliotecas (20%) pero leen de fotocopias (94%) y texto digital (71%), abordan material en inglés (100%) por sobre otros idiomas excepto el nativo, no compran libros (13%) y han experimentado desventajas en el uso de material digital como así también muchos beneficios.

En las categorías que se abrió la posibilidad del ítem *Otros* para que pudiera especificar aperturas no previstas, hubo muy poca ampliación por lo que podemos inferir, a priori, que la lista de opciones definida es completa.

Ante el conocimiento de esta situación los docentes decidimos reflexionar sobre nuevas estrategias para adecuar las actividades aportando mecanismos para validar la pertinencia de los contenidos, correctitud de uso de información y extendiendo y/o robusteciendo la integración entre diversas herramientas.

7. Propuesta Práctica

Enmarcados en este contexto y bajo la premisa que esta propuesta sólo cubrirá el problema en forma parcial es que apuntamos a contribuir para que los alumnos elijan información digital de valor y cuya comprensión se potencie a través del modelado conceptual.

En este sentido, la propuesta dirigida a una comisión de alumnos de la materia Análisis de Sistemas de la carrera Ingeniería en Sistemas de Información, comenzó administrando esfuerzos en la exploración de repositorios digitales, en especial se interactuó con la Biblioteca Electrónica de Ciencia y Tecnología [9], de la cual todas las Universidades argentinas tienen un nodo de acceso gratuito y cuya estructura contiene un meta-buscador para acceder a otros repositorios certificados, además se trabajó con el repositorio Sedici [10] y con Google Académico [11].

Posteriormente se definió la tarea de buscar artículos académicos sobre la unidad temática *"Proceso de Elicitación de Requerimientos"* correspondiente a la currícula de la materia en dichos repositorios. Para darle mayor significación a su aprendizaje [3] y arraigar el tema al contexto, se pautó que la elección apuntara a una visión local o regional del tema, para lo que debieron priorizar palabras clave, identificar los autores más citados, luego se procedió a la selección del material digital. Dicha pieza hace referencia a la indagación de

fuentes utilizadas por desarrolladores de software en Argentina, para elicitar requerimientos [2]. A continuación se efectuó la lectura individual, comprensión, para el posterior procesamiento y abstracción. En una primera etapa, se bosquejó un esquema conceptual y en una segunda, se utilizaron técnicas y herramientas de modelado en UML a través de Diagrama de Clases [4] que constituyen otra unidad temática de la misma currícula. La actividad se desarrolló en forma grupal, reforzando la circularidad e integración de contenidos con la utilización de la herramienta CASE Enterprise Architect.

8. Resultados de la experiencia

Tal como se subrayó en la introducción, la disminución de los ciclos de vigencia del conocimiento imponen la jerarquización del proceso de aprendizaje por sobre el producto. Debido a esto, se llevó adelante esta experiencia que intentó promover la autogestión de los alumnos en una secuencia de tareas pautadas donde ellos la transitaban eligiendo aspectos y tomando decisiones.

Los resultados del proceso fueron mapas conceptuales que interpretaron el contenido del artículo. Se realizaron intercambios entre los diferentes grupos que refinaron las visiones y se lograron los diagramas que se muestran en las Figuras 6 y 7, que representan una jerarquía de Técnicas y Fuentes de Elicitación respectivamente. Estos diagramas de Clases servirán como soporte en la revisión y consolidación del conocimiento.

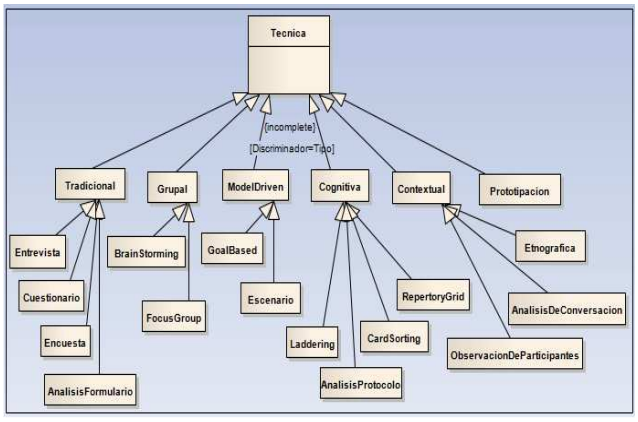


Figura 6- Diagrama de Clases modelando taxonomía de Técnicas de Elicitación

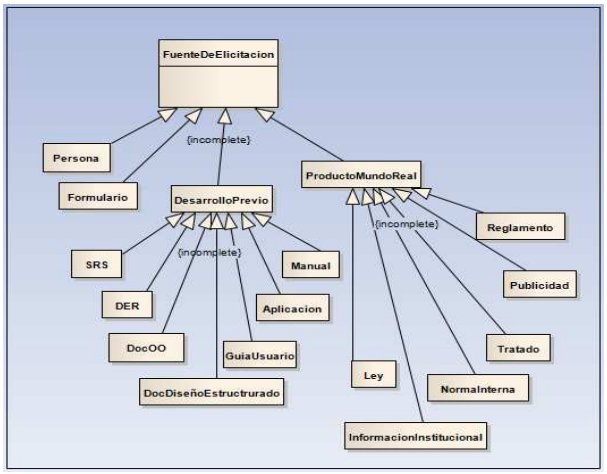


Figura 7- Diagrama de Clases representando Clasificación de Fuentes de Elicitación

Como productos complementarios podemos mencionar:

1. Los estudiantes incursionaron en los espacios de producción científica, lo que les permitió conocer la producción académica nacional y su relación con el mundo.

2. Se esbozaron criterios de selección de material de calidad, especialmente útiles, al momento de elegir y descartar en Internet.

3. A través de la exploración de la producción científica, se promovió el acercamiento del alumno al mundo de la investigación.

4. Los estudiantes descubrieron y visualizaron el patrón subyacente en todo artículo académico con sus secciones: título, autores, sus filiaciones, resumen, introducción, planteo de hipótesis o problema a discutir, marco teórico, estado del arte, propuesta o caso prácticos, resultados, conclusiones, futuras acciones y fuentes bibliográficas.

5. Los estudiantes se familiarizaron con las fuentes de información y autores reconocidos en la disciplina. Se abordó la diferenciación entre fuentes directas e indirectas [6].

6. Otras observaciones importantes de este proceso radican en que los alumnos ubicaron el estado del arte de la temática referida en el contexto propio de las prácticas en Argentina. De éstas se desprende una configuración particular y cultural de llevar adelante procesos de ingeniería regionalmente lo cual constituye un ejercicio de autoconocimiento.

7. Finalmente consideramos un valor adicional de este trabajo al desplazamiento de tareas aisladas e individuales, propias del vínculo dispositivo tecnológico-humano, hacia un espacio de socialización e intercambio.

9. Conclusiones

En el presente artículo hemos expuesto una situación emergente del uso de las TICs que se prevé en incremento y diversidad de conductas del colectivo de alumnos con formación nativa en ellas. La detección de que el 85% de la población de alumnos consultada nunca había sacado libros de la biblioteca, focalizó el objetivo de llevar adelante una investigación particular. A modo de contextualización se tomaron indicadores regionales sobre lectura. De ellos se desprende que en Argentina la lectura digital correlacionada con la permanencia frente a dispositivos tecnológicos, es de uso intensivo en el sector más joven de la población. Es decir que los jóvenes no leen menos sino distinto.

Como resultante de la indagación de una muestra poblacional de estudiantes encontramos que solo el 20% asiste o extrae de libros en bibliotecas, el 94% leen de fotocopias y el 71% prefiere el texto digital, la totalidad lee material en inglés y muy pocos compran libros. Las mayores ventajas del material digital se corresponden con: la búsqueda automática, facilidad de copia, lectura en dispositivos personales, inmediatez y ahorro de insumos. Las mayores desventajas se refieren a déficit de calidad del contenido debido a: fuentes de información desconocidas, material reelaborado por copy&paste, sitios sin acreditación (no académicos), sin fecha de vigencia, libros con edición obsoleta, errores ortográfico/traducciones deficientes.

Como acción compensadora a estas asimetrías presentamos una propuesta para realizar un encadenamiento de tareas para desarrollar las capacidades cognitivas de: explorar, buscar, filtrar, organizar, clasificar, comparar, elegir, relacionar contextos, abstraer en modelos gráficos UML en herramienta informática, tomando como insumo el material digital de calidad seleccionado previamente. Complementariamente se identificaron situaciones oportunas para el desarrollo del conocimiento procedimental como es el caso del descubrimiento del template de la escritura académica, la aproximación a la comunidad de autores y organizaciones relacionadas a un tema, la presentación del mundo de la investigación científica por medio de la exploración de su producción documental, el cambio de escala de lo global a lo regional, la interacción grupal como acción de balance a la tendencia de encuentro individual con la tecnología.

10. Extensiones y acciones futuras

Consideramos nuestra experiencia positiva y estimamos pueda ser profundizada, extendida y aplicada en otros espacios, sobre todo porque los artículos académicos son altamente utilizables en cualquier asignatura siendo un campo que se encuentra en constante producción y evolución. No obstante, se sugiere investigar y observar nuevos hábitos y conductas para que, en forma temprana, el docente reajuste sus intervenciones.

Se prevé como etapa complementaria posterior, trabajar sobre un proyecto de integración con la materia Inglés (indicado en la encuesta como el más usado). Dicho proyecto se basará en la traducción de material en ese idioma sobre la temática de ¿cómo leer e interpretar un artículo académico? Ya se han seleccionado dos fuentes y están siendo consideradas por la cátedra de Inglés [13],[7] para planificar una interacción conjunta.

Por último es importante destacar la necesidad del tratamiento del rol y adecuación de la biblioteca como entidad dinámica en espacios académicos superiores.

11. Bibliografía

[1] Allueva P., Desarrollo de habilidades metacognitivas: programa de intervención. Zaragoza, Consejería de Educación y Ciencia. Diputación general de Aragón. (2002).

[2] Antonelli L., Oliveros A., Fuentes utilizadas por desarrolladores de software en Argentina para elicitar requerimientos, LIFIA, Facultad de Informática, UNLP. (2003)

[3] Ausubel D., Adquisición y retención del conocimiento. Una perspectiva cognitiva. Barcelona: Ed. Paidós. (2002)

[4] Booch G., Jacobson I., Rumbaugh J. El lenguaje unificado de Modelado. Ed. Pearson Educación, 2da. Edición. (2006)

[5] Cassin B., Googléame: la segunda misión de los Estados Unidos - Fondo de la Cultura Económica. Bs. As. (2008)

[6] Eco U., Cómo se hace una tesis - Técnicas y procedimientos de estudio, investigación y escritura. Cap.III Búsqueda del Material (1977).

[7] Griswold W., How to Read an Engineering Research Paper, CSE, UC San Diego disponible en http://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html

[8] Hernández R. D.: Desarrollo económico, Las industrias de Tecnología Avanzada UNSAM (2004).

[9] http://www.biblioteca.mincyt.gob.ar/

[10] http://sedici.unlp.edu.ar/ (Servicio de Difusión de la Creación Intelectual)

[11] http://scholar.google.com.ar/schhp?hl=es-419

[12] Lundvall B.: ¿Por qué la nueva economía es la economía del aprendizaje?, Seminario: "Economie basée sur la connaissance et nouvelles tecnologías cognitives ", Université Technologique de Compiègne" (Enero 2002). http://www.littec.ungs.edu.ar/pdfespa%F1ol/Cap1.%20Lundvall.pdf

[13] McConnell S., How To Read A Technical Article, IEEE Software, Vol. 15, No. 6, November/December 1998

[14] Olivas Varela J. A., Búsqueda eficaz de información en la Web, Cacic, SEDICI (2011)

[15] Salas M.: El aprendizaje en el mundo del trabajo, Inédito UNSAM.

[16] Vercelli A., Google Books y la privatización de la inteligencia comunitaria. Análisis de los procesos de privatización / comunización del acervo literario de la humanidad, IESCT - UNQ (2014)

[17] Consejo Nacional de Lectura, Encuesta Nacional de hábitos de lectura año 2011. Sintesis de resultados Mayo 2012. http://sinca.cultura.gov.ar/archivos/documentos/ENHL_2011.pdf

[18] Federación de Gremios de Editores de España y Mrio. Educación Cultura y Deporte, Encuesta de hábitos de lectura y compra de libros en España 2012. http://www.clibromadrid.es/HabitosLecturaCompraLibros2012.pdf

[19] Centro Regional para el Fomento del Libro en América Latina y el Caribe, CERLALC y UNESCO. Comportamiento lector y hábitos de lectura en América Latina 2011 http://www.cerlalc.org/cole-lectura-escritura.php

Avances en la aplicación de Mundos Virtuales en Educación

Calixto Alejandro Maldonado, Patricia Etcheverry
Universidad Empresarial Siglo 21, Córdoba
Mario Groppo
Universidad Tecnológica Nacional, Fac. Reg. Córdoba
Manuel Pérez Cota
Universidad de Vigo, España

Resumen
Se presentan los resultados de la investigación y desarrollo aplicando Ingeniería de Software y los aspectos didácticos en la aplicación de TICs, específicamente software de Mundos Virtuales (MV) conectado a un Learning Management Software (LMS). Se diseñaron actividades y registro de las transacciones realizadas dentro del mismo, que se relacionan con el dictado de materias de las carreras de Ingeniería Informática. El estudio presentado abarca la detección de requerimientos, estudio de fundamentos didácticos, diseño y construcción de un software de Mundos Virtuales, diseño de actividades, de modelos de datos, formularios y reportes que se unen para lograr un mejor aprendizaje a los asistentes. Se detallan los resultados de los estudios sobre didáctica, Blended Learning, software de MV y las propuestas realizadas. Se eligió la aplicación de lo construido en las materias de Bases de datos de dos carreras de Ingeniería de una universidad pública y otra Privada dictadas en formato Blended Learning con modalidades a distancia y semi-presencial.

Palabras Clave
Ingeniería de Software, Mundos Virtuales, Inmersividad, Blended Learning, Didáctica.

Introducción

La problemática abordada en este artículo refiere un proyecto de investigación de la aplicación de una herramienta de Mundos Virtuales (MV) junto con estrategias de enseñanza en materias de la Ingeniería, con estudiantes habituados a interactuar con videojuegos y escenarios en MV y las actividades específicamente diseñadas para lograr un aprendizaje más significativo.

Se incluyen los objetivos, fundamentos, y resultados obtenidos durante la elaboración una tesis de doctorado de Ingeniería de Software en la Universidad de Vigo (España).

El objetivo de esta tesis es mejorar el aprendizaje de estudiantes universitarios que cursan las Ingenierías de sistemas de Información y de Software de dos universidades locales y de nivel medio a través de interactuar con escenarios creados en un Mundo Virtual (MV).

Como punto de partida y fundamento inicial del trabajo de investigación se define que se deben determinar los requerimientos que tiene el profesor al aplicar este recurso, para mejorar la llegada del contenido a enseñar, lo prescripto, captando la atención del estudiante.

Estos requerimientos para enseñar a los estudiantes una materia de Ingeniería, debe abarcar a los esquemas significativos de recepción, la teoría socio-cultural y las herramientas colaborativas que promueven captar la atención de un estudiante que presenta características generacionales propias, diferentes a las de sus actuales instructores.

Desde el punto de vista de la herramienta, el proyecto constituye una propuesta de software de MV, escenarios y actividades a desarrollar dentro del MV, es decir, una plataforma Multi-Usuario On-line de MV en la pantalla de la computadora de cada uno de los asistentes.

Durante el trabajo de elaboración se han estudiado diversas experiencias de capacitación realizando actividades en Second Life [19], SLOODLE como interfaz de SL y MOODLE, como parte de la recopilación de antecedentes y experiencias.

Para lo anterior se define que durante las actividades en el MV creado, se reunirán sincrónicamente a todos los participantes

del dictado de una materia, estudiantes y profesores, representados por un avatar, y a través del uso de micrófono y auriculares, se generará una señal de audio envolvente que replique el sonido de la presencia física de los compañeros y del instructor, logrando una particular sensación de inmersión en el lugar.
Dado esto se ha definido como supuesto de trabajo en la investigación que la experiencia de realizar actividades en un MV debiera ser aplicada, en un comienzo, en materias dictadas con la modalidad Blended Learning (BL) que requiere una actividad presencial, permitiendo que la la misma sea virtual, recibiendo el nombre de Blended Learning 2.0. (BL2.0) [2] y [8]
Esta propuesta ha generado atención e interés de una de las Universidades en las que se llevan a cabo la presente investigación, logrando ser incorporado como proyecto financiado. En la otra Universidad ha prestado colaboración para iniciar un proyecto en el próximo año con aplicación en la modalidad semi presencial de la materia Gestión de datos.

Se encontró y contactó a grupos de investigación que llevaron a cabo proyectos de aplicación de MV para educación en otra Facultad Regional de la UTN, que aplica OpenSim para realizar tareas de enseñanza, se ha previsto la comunicación con la dirección e integrantes de los mismos para realizar un intercambio de experiencias y desarrollos.

La idea de usar software de MVs para educación fue presentada en el marco del congreso mundial de ingeniería 2010 en Buenos Aires en una Mesa Redonda sobre Tecnología de Informática y Comunicaciones. [12]

A modo completar el contexto podemos enumerar las siguientes fuentes sobre el tema tratado:

“Es notable la motivación desarrollada por los juegos en los jóvenes y una de las hipótesis que se busca demostrar en este trabajo es el uso de esta atracción a fin de abrir un renovado canal por el cual llegar a la motivación, atención y comprensión genuina de contenidos por parte del estudiante.” [1]

“Referido a esta excelente predisposición cognitiva que generan los video juegos, y la negativa recomendación a luchar contra esta fascinación, sino por el contrario, usarla con fines pedagógicos, es que se ha logrado atraer la atención de autores como Marc Prensky.” [1]

“No deberá sorprender que las personas que actualmente realizan videos y juegos de computadora tienen la gran oportunidad de que sus mensajes sean positivamente escuchados” [2]

“El aprendizaje significativo ocurre cuando una nueva información "se conecta" con un concepto relevante o subsunsor (estructura y conocimiento previo que sirven de base para la adquisición de nuevos conocimientos) pre existente en la estructura cognitiva” [3]

“Se ha avanzado en el estudio de los fundamentos de Didáctica buscando definir los requerimientos que esta disciplina exige cumplir para lograr un aprendizaje significativo y duradero” [4]

“SLOODLE provee un amplio rango de herramienta que permiten a los mundos virtuales el aprendiza y la enseñanza inmersiva…” [5]

Elementos del Trabajo y metodología
El supuesto previsto en el proyecto para que esto sea aplicado a una materia en modalidad BL 2.0 surge del estudio de la forma tradicional de BL definido como una forma de la educación a distancia mediada por tecnología, denominada Blended Learning pudiendo ser traducida como Aprendizaje Mixto que puede definirse como:

"La convergencia actual de dos ambientes arquetípicos de aprendizaje. De un lado el tradicional ambiente de aprendizaje "Cara a Cara" que está presente desde hace siglos y, por el otro lado, los ambientes distribuidos de aprendizaje que han empezado a crecer exponencialmente junto con el hecho que las TICs han expandido las posibilidades para la comunicación e interacción…"[9]

Para poder desarrollar escenarios y actividades en MV se debe tener claro el objetivo principal de la investigación, esto es, mejorar el aprendizaje usando MVs.

Como primer paso se analizan y enumeran los requerimientos didácticos del profesor y las actividades a desarrollar. Estos han sido recopilados, analizados y los resultados publicados en recientes congresos [4], se sintetizan para esta presentación.

a) Se debe seleccionar un contenido curricular del prescripto a fin de valorar su metodologia más adecuada.
b) Se deben plantear situaciones conflictivas. Desde el formato "juego interactivo" o simil, exigir distintas actividades a ser realizadas en pos de un objetivo dado y así posibilitar distintas estrategias y proveer herramientas para su posible resolución.
c) Se deberá introducir, previo estudio de las características socio etarias de la población de los asistentes, a los mismos mediante un ejemplo corto, sea éste narrado o animado a fin de introducirlo en el efecto sugestivo y predisponerlo del mejor modo para lo que vendrá, el efecto motivacional.
d) Es un imperativo considerar las consecuencias didácticas y metodologicas del trabajo cooperativo colaborativo, entre grupo de pares. Es decir, es esencial plantear actividades colaborativas desde las herramientas y actividades a desarrollar.
e) Debe existir un registro de la elaboración discursiva en los distintos niveles de los contenidos trabajados. Por ejemplo, si se planteara un juego con postas, o desafíos a modo de diálogo es esencial, llegar a construir colaborativamente la síntesis del trabajo. Así se irá chequeando en espiral dialéctico y se ajustarán las variaciones surgidas. Además el juego también podría avanzar dialécticamente, complejizándose cada vez más a partir de contradicciones y nuevas síntesis a medida que se va recrea el contenido.
f) Considerar la creación de un espacio dentro de la plataforma donde se plantee un "recreo", en donde se puedan volcar comentarios, pareceres y opiniones casuales. Ambiente que en general no se considera en la educacion a distancia, sin embargo, éste sera un espacio donde podrán surgir cuestiones de importancia, conocido desde la dinámica de grupos, como lo instituyente y adquiriendo valor de verdad frente a lo establecido.
g) Finalmente, todo este despliegue desarrollado en MV, MOODLE y demas, debe estar contenido, como un componente de una actividad integral que, se asume, debe estar planificada y articulada con acceso a recursos educativos como textos académicos, debates y todo aquello que forme parte de la planificación en su totalidad.

En el estudio de los fundamentos didácticos, se buscó definir los requerimientos que ésta disciplina exige para lograr un aprendizaje significativo y duradero. Se encontró que la didáctica es una disciplina con la que se construye nuevo conocimiento, que está en constante

desarrollo y en relación a diferentes contextos de aplicación, ubicando al profesor en el rol de docente-investigador, que aporta hacia la construcción de su corpus o marco conceptual [6]. Además se observó que se debe tener claro el rol y la intencionalidad del dispositivo didáctico (Software de MV y escenarios) en función de provocar cambios en el sentido de aprendizaje en los jóvenes, lo que les permitirá incorporarse activamente en la cultura y la sociedad. Así el profesor y el dispositivo promueven, con las situaciones de enseñanza orientadas hacia la transmisión de saberes, transformaciones en lo cognitivo, afectivo y social. [7]
Por lo marcado anteriormente, se estudiaron las actuales estrategias de aprendizaje y su aplicación de TIC's, específicamente orientado a explorar los posibles aportes de las distintas distribuciones de software de MV disponibles.

Se realizaron estudios para determinar cual campo del conocimiento presentaría ventajas y dificultades para la aplicación de la propuesta. Se vió que la Filosofia, en algunos temas muy abstractos, podría presentar dificultades en hallar metáforas y actividades que usen MV. En cambio la Física, Química, Historia y Geografía, tienen abundantes posibilidades de crear metáforas y actividades que mejorarían el aprendizaje de los participantes.

En lo que hace a la investigación de estrategias didácticas se han realizado estudios y análisis que derivaron en conclusiones que figuran como aportes publicados y de los que podemos destacar:

"El profesor con su grupo clase, se enfrenta con, por lo menos, dos cuestiones. Por un lado y en cuanto al contenido, la incertidumbre acerca de los saberes previos de su grupo, es decir, aquellos saberes ya internalizados en sus estudiantes que le permitirían a modo de anclaje sostener aquellos que vendrán vehiculizados a través de actividades didácticas y situaciones de aprendizaje nuevas. La segunda cuestión se refiere a la incertidumbre acerca del nivel cognitivo-madurativo alcanzado por los estudiantes lo que, de saberse, permitirá la toma de decisión y puesta en práctica de líneas de acción didácticas bien precisas y funcionales al grupo clase y así ajustar acertadamente la didáctica aplicada a fin de generar conocimiento genuino.

Con todo esto, se contaría con un diagnóstico del grupo clase que posibilitaría la construcción de "organizadores previos" en función de garantizar código compartido y activar esquemas relacionales que garanticen la significatividad de los contenidos impartidos y al mismo tiempo el pasaje hacia la retención de los conceptos. [8]

Esto último condujo a desarrollar una actividad en un escenario de MV, donde el profesor que requiere determinar si los estudiantes tienen conocimientos previos del tema próximo a exponer, referido a una Unidad Temática (UT) podrá tener reportes de cuanto saben de ésta UT y lo que conocen poco o nada de ella.

Esta actividad estaría situada en una simulación de programa de televisión de preguntas y respuestas y en un panel se mostraran preguntas y opciones de respuestas. El profesor toma el rol de conductor del programa, informa de las características de la actividad, pudiendo reforzar con un beneficio de participar, como el enterarse de qué se trata lo que viene, en un juego lo que le ayudará a mejorar su desempeño próximo. Luego leerá las preguntas que además aparecerán en un tablero luminoso visible para todos, indicando el momento a responder con la letra de la opción correcta y luego de un lapso lee la respuesta correcta. Los estudiantes escucharán la pregunta, la observarán en el tablero y a través del teclado ingresarán su opción elegida y podrá enviarse a su correo el texto de las preguntas y respuestas dadas, ya que quedará registrado en la base de datos del MV, el curso, código de materia, unidad

temática, id del estudiante, número de pregunta respondida, opción elegida y si es correcta. El diseño de las tablas para registrar las respuestas se creará en el motor de bases de datos usado por MOODLE, MySql. También se prevé diseñar un formulario de carga de preguntas por unidad temática de cada materia y uno o varios reportes de las respuestas dadas por los estudiantes, lo que ofrecerá un valor cuantitativo del conocimiento previo de los conceptos importantes de la UT a dictar.

Se realizó también un estudio del estado del arte de las herramientas de software de MV encontrando que han avanzado notablemente y son utilizadas para el entretenimiento en juegos de distintos tipos. Es para destacar la existencia de distribuciones Open Source de alguno de ellos lo que facilitará la tarea de construcción del software producto del trabajo de investigación.

Del software de MV encontrado, se analizaron características, ventajas, desventajas, potencialidades de ampliación, capacidad requerida para su instalación y ejecución, performance, test de carga, escalabilidad y seguridad. Puntualmente de Second Life se realizaran pruebas de concepto, construyendo en ése MV los artefactos para revisar su factibilidad y potenciales aplicaciones. Con la información resultante del análisis se prevé establecer las bases para producir un software de servidor y cliente de MV que permita impartir contenidos de materias de nivel medio, universitarias y de capacitación laboral, cumpliendo los requerimientos de las estrategias didácticas más actuales estudiadas y que fueron enunciados en citadas publicaciones [11] [13] [14].

Para construir las herramientas de software que interactuarán con los docentes y estudiantes en la proyectada infraestructura de MV, se estudiarán varias opciones:

(a) la tecnología ofrecida en licencia Pública del originalmente llamado Project DarkStar y que, recientemente, ha sido renombrada como RedDwarf Server [15],
(b) de otra fuente de licencia pública como es OpenWonderland [16] que es una solución de middleware de código abierto para el desarrollo del lado del servidor de juegos multi jugador masivo en línea. En la actualidad es posible descargar el código fuente e iniciar un desarrollo independiente debiendo seguir con la licencia Open Source.
(c) También se analizó el servidor Open Croquet [17], diseñado para ser usado en el desarrollo de aplicaciones de MV en SmallTalk. Las aplicaciones creadas con el SDK Croquet son automáticamente colaborativas ya los objetos creados comparten un protocolo común que permite colaborar entre ellos, usando el principio de sincronizacion basado en mesajería basada en pares y no en el servidor. Esto facilita la replicación de cómputo entre pares para evitar la sobrecarga sobre un servidor que deba soportar toda la implementación de MV colaborativos. [18]

En conjunto con las actividades realizadas en la plataforma de mundo virtual a desarrollar, se vió necesario incluir un Software de Gestión de Aprendizaje (Learning Management Software para su sigla LMS). Acorde al análisis realizado, se concluyó que el LMS más conveniente y adaptable a nuestras necesidades será MOODLE. Este software gestionará la información de los asistentes, cursos, resultados y, conjuntamente con la información propia de MOODLE, se crearán y utilizarán tablas propias para atender los requerimientos didácticos encontrados y por descubrir. Las tablas diseñadas para contener la información de perfil de los participantes, las referencias a las evaluaciones realizadas en MOODLE junto con las respuestas de cada estudiante con la opción elegida junto con las respuestas abiertas redactadas que las complementarán se crearan en el esquema

del LMS con un prefijo para su identificación “MV”.

Se ha determinado importante la posibilidad de que el estudiante pueda acceder a todas sus respuestas pasadas y poder compararlas con nuevas respuestas y poder mejorar sus participaciones en las actividades previstas.

Las Universidades implicadas en esta investigación, han financiado en un caso con la aceptación de un proyecto de tipo I y en el otro ha autorizado a realizar experiencias en la materia elegida.

Resultados

Una vez elegidos los requerimientos que serán atendidos por lo producido en el proyecto se avanzó en los siguientes resultados desarrollados a continuación agrupados por el requerimiento:

a) “Se debe seleccionar un contenido curricular del prescripto a fin de valorar su metodologia más adecuada.”

Con este requerimiento se elaboraron pautas para la selección de la materia a elegir, determinando que el contenido debe tener:

a) contenidos posibles de ser base de metáforas significativas y posibles de ser construidas en un MV.
b) una estructura posible de aplicar en las actividades a diseñar.
c) Una experiencia asociada que puede ser recreada en un ambiente y con una actividad.

Se determinó que la materia Gestión de Datos del tercer año de la carrera de Ingeniería en Sistemas de Información de la UTN, en su primer UT en la que trata de arquitecturas de RDMS, podria servir para crear una metáfora de este complejo software. Una visita guiada a artefacto en funcionamiento, ingresando una consulta simple primero y luego una que utilice índices que cambien el plan de ejecución, podría ayudar a visualizar los pasos en los que transforma de una Sentencia Select a las filas resultado. Lo mismo puede aplicarse a Base de Datos 1 de la carrera de Ingeniería en Software de la UES21.

También se enunció que una materia que involucre en una UT, procesos abstractos como la Filosofía sería un poco complicado aplicar la propuesta. En cambio, se pensó en materias de otros campos como las ciencias naturales para encontrar casos de aplicación de lo diseñado.

b) “Se deben plantear situaciones conflictivas. Desde el formato “juego interactivo” o simil, exigir distintas actividades a ser realizadas en pos de un objetivo dado y así posibilitar distintas estrategias y proveer herramientas para su posible resolución.”

Este requerimiento permitió desarrollar la actividad del concurso de preguntas y respuestas, en las que los asistentes son puestos a responder preguntas sobre conceptos de una UT que aún no han cursado. Otra actividad lúdica que se ha diseñado es un juego de preguntas y respuestas de todo el programa con puntaje y determinación de un ranking entre todos los participantes del grupo que acepten la publicación de sus resultados para que finalmente se compartan en la clase.

Por el formato ideado y la asociación a los juegos mas difundidos en la actualidad se espera que en las pruebas de concepto esta actividad acerque sus conocimientos previos de la UT con el objetivo dado que es competir o compararse con sus compañeros, de paso dejando información sobre cuanto conocen del proximo tema al profesor, y en el segundo caso, difundir las respuestas correctas a preguntas importantes de la materia estudiada, difundiendo el conocimiento específico entre los estudiantes, estimulando una participación mas provechosa, según lo estudiado en el proyecto[1][3][4].

c) “Se deberá introducir, previo estudio de las características socio etarias de la

población de los asistentes, a los mismos mediante un ejemplo corto, sea éste narrado o animado a fin de introducirlo en el efecto sugestivo y predisponerlo del mejor modo para lo que vendrá, el efecto motivacional."

La plataforma MOODLE cuenta con el registro del perfil del estudiante con los datos requeridos. Se espera que la actividad de concurso de preguntas y respuestas que cubre el requerimiento anterior, tambien tenga efecto sugestivo en el asistente, ya que está conociendo los conceptos a estudiar en la próxima UT y en el caso de que ya los conozca puede generar un efecto de confianza que permitiría una mayor adaptacion en el próximo periodo de estudio. Si no tuviera idea acerca de lo preguntado, al menos, por la característica de la actividad se habrá logrado una atención sobre éstos y al mostrarse cuales son las respuestas correctas, se espera producir algun residuo cognitivo en él.

d) "Es un imperativo considerar las consecuencias didácticas y metodológicas del trabajo cooperativo colaborativo, entre grupo de pares. Es decir, es esencial plantear actividades colaborativas desde las herramientas y actividades a desarrollar."

La sensación de inmersividad a través del audio del escenario en el que se esta trabajando incluye escuchar y hablar al participante, lo que lo conecta a través de este sentido con el grupo clase.

Ademas, este requerimiento motivó al trabajo de investigación y diseño a buscar y definir especificaciones de herramientas para estimular y plasmar trabajo colaborativo acerca de la UT. Las herramientas que se han especificado como útiles son un dispositivo de chat, para que los participantes puedan conversar y ser registrado en el sistema y una herramienta que permita construir Wikis, para que en grupo, de a pares, se puedan elaborar documentos colaborativos como actividad a realizar en el MV como el chat y en la plataforma MOODLE conectada con SLOODLE.

e) Debe existir un registro de la elaboración discursiva en los distintos niveles de los contenidos trabajados. Por ejemplo, si se planteara un juego con postas, o desafíos a modo de diálogo es esencial, llegar a construir colaborativamente la síntesis del trabajo. Así se irá chequeando en espiral dialéctico y se ajustarán las variaciones surgidas. Además el juego también podría avanzar dialécticamente, complejizándose cada vez más a partir de contradicciones y nuevas síntesis a medida que se va recrea el contenido.

Este requerimiento nos ha permitido estudiar los roles y grupos presentes en la plataforma MOODLE para que sean aplicables en el escenario MV, cambiando comportamientos o aspectos y en las actividades, aumentando o disminuyendo las dificultades y poder registrar los resultados de todos estos elementos para su posterior análisis y evaluación de las actividades.

f) Considerar la creación de un espacio dentro de la plataforma donde se plantee un "recreo", en donde se puedan volcar comentarios, pareceres y opiniones casuales. Ambiente que en general no se considera en la educacion a distancia, sin embargo, éste sera un espacio donde podrán surgir cuestiones de importancia, conocido desde la dinámica de grupos, como lo instituyente y adquiriendo valor de verdad frente a lo establecido.

Se han analizado y definido las características de un area de relax en el MV, y un foro en MOODLE donde se planteen temas no relacionados con la materia y que permitan expresar ideas descontracturadas sin ser evaluadas como correctas o incorrectas, sino como aportes importantes a la hora de generar un buen clima que permita el aprendizaje.

g) Finalmente, todo este despliegue desarrollado en MV, MOODLE y demas, debe estar contenido, como un componente de una actividad integral

que, se asume, debe estar planificada y articulada con acceso a recursos educativos como textos académicos, debates y todo aquello que forme parte de la planificación en su totalidad.

Este requerimiento se cumple con la adopción de SLOODLE como plataforma de software que conecta a MOODLE y al software de MV, para registrar las transacciones durante la actividad en MV y ser el soporte del material como textos académicos, videos y foros, wiki y demás recursos para cumplir con la planificación del dictado de la materia.

Discusión

El primer supuesto de la investigación es que el uso de MV y las actividades desarrolladas dentro de él, se darán inicialmente, en materias dictadas en la modalidad BL y reemplazarían la parte Cara a Cara "física", por una actividad Cara a Cara remota, por que se obtendría "Avatar a Avatar". Si fuera exitosa esta intervención se avanzará en materias con otra forma de dictado, para ampliar el uso de la plataforma desarrollada.

El fundamento de elegir esta modalidad de impartir la materia en la que intervenir esta dado por lo siguiente:

"Existen muchas razones por las que un profesor o estudiante debiera adoptar BL. Guthorpe y Graham (2003) identificaron seis razones para elegir un sistema BL:

1) riqueza pedagógica,
2) acceso al conocimiento,
3) interacción social,
4) mediación,
5) eficiencia en el costo y
6) facilidad de revisión.

En la literatura de BL se suele usar el sentido común de la frase que dice que BL combina "lo mejor de los dos mundos". [9]

También se consideró en este estudio, que el hecho de no tomar en cuenta las necesidades del docente y la estrategia utilizada, es decir, imponer técnicas, diseños y tareas, con el fin de que la herramienta funcione y no para cumplir el objetivo de la estrategia didáctica a aplicar por el docente, o sea priorizar el aparato o dispositivo tangible y no ubicar en el plano principal la finalidad del proceso que subordina al dispositivo, junto con la estrategia es como poner el carro antes que el caballo. Se ha visto experiencias similares que no han tenido éxito en su aplicación y la causa que se ha encontrado en este estudio, es una prioridad por el dispositivo sin el objetivo claro.

Analizando las áreas de conocimiento que forman parte del corpus de las ingenierías existen pocas materias sin metáforas o con metáforas que tengan dificultad de expresarse en un MV o que la actividad no pueda realizarse en ellos. Las materias como Filosofía y Sociales, pueden tener algunos contenidos difíciles de graficar con metáforas dado su alto nivel de abstracción. Pese a esta situación y generalización, uno de los escenarios ideados durante el trabajo fueron los portales cubiertos del Liceo de Aristóteles conocidos como *perípatoi* [10] por donde el filósofo leía y enseñaba a sus discípulos, que ofrece una interesante oportunidad de realizar experiencias que ayuden a los estudiantes a aprender sobre esta materia. En otros campos, en cambio existen grandes oportunidades de aplicación del proyecto. En Biología, existen mecanismos fisiológicos, como en una célula, un tejido o un órgano completo. Lo que podria ser fuente de metafora y de actividades de recorrer, abrir, introducirse y observar determinados procesos internos, a veces muy dificiles de visualizar.

Se realizó un estudio del estado del arte de las herramientas de software de MV que han avanzado notablemente y son utilizadas para el entretenimiento en juegos de distintos tipos. Más aún la capacidad de los ordenadores hay permitido su disposición, poniendo al alcance de los estudiantes juegos muy exitosos que utilizan MV.

Existen distribuciones de software abierto que pueden usarse para construir MV exclusivamente para educación y almacenar en ellos todos los dispositivos y materiales usados para las clases sincrónicas remotas en MV.
Pese a existir MV en sitios con mucha asistencia de usuarios, se ha visto como un valor importante controlar toda la instalación en la universidad para evitar pérdidas de los insumos allí guardados y preservar el acceso a los estudiantes de la casa sin interferencias de desconocidos, o de diversos sitios de contenidos inapropiados dentro de esos MV públicos.
Experiencias en SL, lecturas.

La asistencia prevista a los cursos de Bases de Datos en los casos analizados, no superan los 300 estudiantes por semestre, número que nos ubica fuera del grupo de Juegos Masivos, lo que alivia la necesidad de equipamiento y soporte de ancho de banda de red.

Conclusiones

La didáctica ofrece requerimientos que pueden atenderse con la aplicación de tecnología para su desarrollo.

El BL ofrece un marco de aplicación de las propuestas excelente, generando la posibilidad de que materias dictadas en BL2.0 evite los costos de viajes hacia las sedes de los encuentros presenciales entre otras ventajas, sin perder la presencialidad por esto.

Las materias de la Ingeniería presentan oportunidades de aplicación de la propuesta, se ha visto que esto no es igual en todos los campos, pero algunos serán muy productivos.
Se decidió sobre qué materias aplicar el proyecto, son las relacionadas a Bases de Datos y se ha avanzado en desarrollar metáforas en otras materias, más abstractas aún, que permitan a los estudiantes a aprender conceptos con ellas.

La ingeniería de software cuenta con tecnología y técnicas adecuadas para registrar los requerimientos, transformarlos en casos de uso y de prueba, desarrollo de las especificaciones de los procesos, metodologías de organización de trabajo con metodologías ágiles que permiten llevar a cabo el desarrollo del software derivado de la investigación y su prueba de errores. Todo esto hace que el proyecto sea de Ingeniería de Software aplicada a la Didáctica.

La disponibilidad del software de MV Open Source, permite a este proyecto completar la propuesta aplicando la Ingeniería de Software en la construcción de un motor de MV como prototipo de prueba de los conceptos y diseños desarrollados.

Dada la cantidad de estudiantes que pueden usar simultáneamente la plataforma el requerimiento de Hardware no será tan importante como los juegos de asistencia Masiva, donde se cuentan por decenas de miles de usuarios conectados simultáneamente. [19]

Se ha contactado con otros grupos de Investigación que han elaborado experiencias en el uso de mundos virtuales. Esto, de alguna manera convalida la elección del tema y el camino elegido, resta sumar esfuerzos para mejorar los resultados.

Agradecimientos
A la Universidad Empresarial Siglo 21 (UES21), al Dr. Aldo Merlino Director del Área de Investigación, a la Lic. Patricia Etcheverry directora del Proyecto de Investigación Situado en la UES21.
A la Universidad de Vigo y al Dr. Manuel Pérez Cota por su dirección en el desarrollo de la investigación publicada.

Referencias

[1] Perez Cota, Manuel y otros “Mundos Virtuales: Estrategias didácticas” Cisti 2012 Pag. 212 Aisti Madrid, España 2012 ISBN: 978-989-96247-7-1
[2] M. Prensky, Marc (2006) Mom don’t bother me, I’m Learning. http://www.marcprensky.com/ dont-bother-me-mom-im-learning/ visitado en enero 2012

[3] Maldonado, C.; Etcheverry, P.; Perez Cota, M.; Groppo, M. "Hacia el uso de Mundos Virtuales en Blended Learning 2.0" Conaiisi 2013 Congreso Nacional de Ingeniería Informática/Sistemas de Información (En línea) ISSN: 2347-0372 Córdoba.

[4] Pérez Cota, Manuel y otros "Resultados del estudio de requerimientos de la Didáctica para una aplicación de Mundos Virtuales en educación". Pag. 1157. XV Workshop de Investigadores en Ciencias de la Computación WICC 2013. ISSN 978-950-766-082-5 ISBN: 9789872817961 Paraná Corrientes 2013

[5] Torres Vallejo, y otros. CREACIÓN DE UN METAVERSO EN OPENSIM PARA LA UNIVERSIDAD DISTRITAL DENTRO DE LA RED RITA-UD. REDES DE INGENIERIA, 3(2), 51-60. 2013.

[6] Morin, Edgar; Le Moigne, Jean-Louis. La Inteligencia de la complejidad (Morin, Edgar; Le Moigne, Jean-Louis, L'intelligence de la complexité, L'Harmattan,1999. ISBN 2738480853)

[7] Souto, Marta. Grupos y dispositivos de formación. Colección Formación de Formadores. Ediciones Novedades Educativas y Facultad de Filosofía y Letras, UBA. 1999. Buenos Aires.

[8] Maldonado, C; Etcheverry, P. "Blended Learning 2.0 con Mundos Virtuales" en la Revista de Ciencia y Tecnología, Nro. 13, 2013, pág. 189-202 ISSN 1850-0870. Universidad de Palermo. CABA

[9] Charles R. Graham capítulo del libro Bonk, C. J., & Graham, C. R. (2012). The handbook of blended learning: Global perspectives, local designs. John Wiley & Sons. San Francisco CA.

[10] Wikipedia artículo sobre el Liceo de Aristoteles http://es.wikipedia.org/wiki/Liceo Visitado Julio 2014

[11] Pérez Cota, Manuel et al "Requerimientos Didácticos para usar en Mundos Virtuales" XIV Workshop de Investigadores en Ciencia de la Computación 2012: Wicc 2012 Compilado por Horacio Daniel Kuna 1a Edición, Posadas, Universidad Nacional de Misiones, 2012 E-book ISBN 978-950-766-082-5

[12] "Congreso Mundial de Ingeniería 2010" - Mesa Redonda T. Información y Comunicación http://www.ingenieria2010-argentina.info/programa/programaExtendido.php?casillero=1111151000&sala_=11&dia_=11&idioma=

[13] Maldonado, C.; Perez Cota, M.; Groppo, M. "Nuevas Formas de Comunicación: Second Life" Encuentro Informático Universidad Nacional de La Rioja. 2007

[14] Maldonado, C. Pérez Cota, M.; Groppo, M. "Uso de Mundos Virtuales en aplicaciones de educación" CNIT 2009 - Congreso Nacional de IT – Córdoba, Argentina - ISBN: 978-987-24343-2-8

[15] Wikipedia Project RedDwarf Visitado Febrero 2012 http://en.wikipedia.org/wiki/RedDwarf_Server

[16] www.openwonderland.org/ visitado en Abril del 2012

[17] Www.opencroquet.org en funcionamiento restringido solo a bajar el SDK v 1.0,

[18] http://en.wikipedia.org/wiki/Croquet_Project Visitado en Febrero de 2012.

[19] Second Life (2013) http://secondlife.com-/whatis/faq.php#02. – visitado en Febrero 2013

Accesibilidad en el Ambiente Educativo Virtual Moodle 2.7

Diana Lorena Salinas Gomez[1], Liliana Raquel Cuenca Pletsch[2], Franco Escobar[3], María Antonieta Lopez Lotero[4], Valeria Celeste Sandobal Verón[5]
Grupo de Investigación Educativa
Universidad Tecnológica Nacional – Facultad Regional Resistencia
Resistencia, Chaco (Argentina)
[1]dianalorenasg@gmail.com; [2] cplr@frre.utn.edu.ar; [3] franco_m77@hotmail.com; [4] lopezloteroma@gmail.com; [5] valeriasandobal@hotmail.com

Abstract

El presente trabajo se enmarca en el proyecto de investigación titulado "Modelización y desarrollo de un Repositorio de Objetos de Aprendizaje para la gestión del conocimiento en la UTN-FRRe". Uno de los objetivos del mismo es garantizar la accesibilidad con el objeto de incluir a personas con discapacidad en la utilización de los contenidos del repositorio.

Para cumplir con el mencionado objetivo se ha realizado una evaluación del entorno virtual Moodle, versión 2.7, en lo referido a las características de accesibilidad y cumplimiento de las Pautas de Accesibilidad para Contenidos Web (Web Content Accessibility Guidelines, WCAG 2.0).

Para llevar adelante esta actividad se contrastaron las caracteristicas del entorno con el template Clean respecto del segundo nivel de conformidad "AA" del estándar.

Se utilizaron, como herramientas, extensiones para el navegador Firefox como evaluadores manuales, así como dos observadores para registro de prueba de usabilidad con el software NVDA.

Se espera que los resultados de estas pruebas constituyan información útil, no sólo para mejoras en el entorno en lo que respecta a la accesibilidad, sino también como información decisoria para las instituciones educativas que deben elegir una plataforma virtual, ya sea para extender su oferta académica como para reforzar la presencialidad.

Palabras Clave: Accesibilidad, Moodle, evaluación.

1. Introducción

La accesibilidad es una característica cada vez más requerida en los contenidos web [1], y se considera fundamental en sistemas educativos en línea, en los cuales la integración de usuarios con diferentes capacidades físicas y cognitivas, forma parte de las políticas globales para la inclusión.

El entorno virtual Moodle es una de las plataformas de enseñanza y aprendizaje más utilizada por las instituciones educativas [2], entre ellas la Universidad Tecnológica Nacional. La versión 2.7 tiene como carta de presentación la mejora en diversos aspectos de la plataforma, entre los que se destaca la accesibilidad. La definición del template Clean por defecto y del editor de contenido Atto, que reemplaza al editor anterior TinyMCE, consolidan esta evolución [3].

El presente trabajo busca comprobar las características de accesibilidad que describe Moodle 2.7 y calificarlas mediante el estándar WCAG 2.0 de la W3C [4][5], con el objetivo de definir el nivel de accesibilidad presente en la versión analizada.

La selección de esta plataforma en particular se debe a que es la utilizada en la Universidad Tecnológica Nacional y se prevé la integración de la misma con el repositorio de objetos de aprendizaje de la Facultad.

En primer lugar se presentan los conceptos utilizados para el desarrollo del trabajo. Luego, se analiza la accesibilidad de la plataforma, evaluada conforme las mismas bases sentadas en un trabajo previo sobre Accesibilidad en Objetos de Aprendizaje [6]. Finalmente, se presentan los resultados más relevantes, las conclusiones y sugerencias de mejora para el Moodle 2.7.

2. Elementos del trabajo y metodología

WCAG 2.0 define como crear contenido web accesible para personas con discapacidad. La accesibilidad considera un amplio rango de discapacidades, tales como las visuales, auditivas, físicas, del habla, cognitivas, del lenguaje, de aprendizaje y neurológicas.

Para poder satisfacer estas necesidades se proporcionan varios niveles de orientación: Principios, Pautas, Criterios de conformidad y Técnicas suficientes y recomendables.

Las pruebas se efectuaron utilizando una instalación local del Moodle 2.7 con su template por defecto denominado Clean. Se evalúan las facilidades ofrecidas por la plataforma, no los contenidos disponibles actualmente, ya que estos están relacionados con otro proceso de evaluación según el tipo de recurso digital de que se trate [9].

Para realizar las evaluaciones se utilizaron herramientas manuales y ayudas técnicas y el navegador Firefox 31.0 sobre Ubuntu 12.04. Todas pertenecen a la categoría de software libre y, además, de acceso gratuito y se detallan a continuación:

- Accessibility toolbar para Firefox [10]: Desarrollada por la Universidad de Illinois ("Illinois Center for Information Technology and Web Accessibility)". Se encuentra disponible en el repositorio de extensiones de Firefox y permite generar reportes sobre el cumplimiento de ciertos criterios de accesibilidad del sitio a analizar.
- Web developer toolbar [11]: Al igual que la herramienta anterior, se encuentra disponible en el repositorio de Firefox. Permite evaluar elementos individuales, así como el uso de las pautas WCAG 1.0 [12].
- Contrast Checker: Se trata de una herramienta web que permite medir la relación de contraste entre dos colores, característica especialmente importante en personas con dismución visual.
- NVDA: se trata de una herramienta que facilita a usuarios ciegos, o con severas disminuciones visuales, realizar la navegación en el sitio mediante un lector de pantalla.

2.1. Pautas de accesibilidad

En el presente apartado se describen las pautas y principios de accesibilidad que se tomaron en cuenta para las evaluaciones realizadas.

2.1.1. Pautas de accesibilidad para la WCAG 1.0. Las pautas de accesibilidad establecidas para la WCAG 1.0 son 14 [13]. , y están vinculadas con: la posibilidad de que el *contenido visual y auditivo sean equivalentes* para garantizar su *accesibilidad* por parte de usuarios *ciegos*; que los *textos y gráficos* se *comprendan* aún si se presentan *sin color*; que se *utilicen apropiadamente* los *marcadores y hojas de estilo*; que se *utilicen marcadores* que *faciliten* la *pronunciación* de *texto abreviado* o *extranjero*; que las *tablas* se *transformen* correctamente; que las *páginas* sean *accesibles* mediante *tecnología anterior* a la utilizada; que los *objetos* o *páginas* que se *mueven*, *parpadean*, se *desplazan* o se *actualizan automáticamente*, puedan ser *detenidos* o *parados* por *personas* cuyas *discapacidades cognitivas* o *visuales* les impiden leer textos que se mueven con la suficiente rapidez o en absoluto; que se *asegure la accesibilidad directa de las interfaces de usuario incrustadas;* que las *aplicaciones* sean *independientes del dispositivo*; que se *utilicen soluciones provisionales* para garantizar el funcionamiento de las *ayudas técnicas* y los *navegadores antiguos*; que se utilicen las *tecnologías* y *pautas W3C* y, si no es posible utilizarla o, usándola, se obtengan materiales que no se transforman correctamente, se proporcione una *versión alternativa* del contenido que sea *accesible*; que se *proporcione información* de *contexto* y *orientación*, *mecanismos claros* de *navegación* y que los *documentos* sean *claros* y *simples.*

Asociados a cada pauta se presentan puntos de verificación que explican cómo se aplica la misma en situaciones típicas de desarrollo de contenido. Cada punto de verificación, a su vez, está asociado a una de 3 posibles prioridades: La *prioridad 1* se asocia a puntos de verificación que el desarrollador de contenidos de páginas web *debe* satisfacer ya que, de otra forma, ciertos grupos de usuarios *no podrían acceder* a los documentos Web. La *prioridad 2* se asocia a puntos de verificación que el desarrollador de contenidos de páginas Web *debería* satisfacer ya que, *si no lo hiciese*, a ciertos grupos de usuario les resultaría muy difícil acceder a los documentos Web. La *prioridad 3* se asocia a puntos de verificación que el desarrollador *puede* satisfacer para mejorar la accesibilidad de los documentos Web.

Además se han definido 3 niveles de adecuación:

Si se satisfacen todos los puntos de verificación de prioridad 1 se obtiene un Nivel de Adecuación *A (A)*.

Si se satisfacen todos los puntos de verificación de prioridad 1 y 2 se obtiene un Nivel de Adecuación *Doble A (AA)*.

Si se satisfacen todos los puntos de verificación de prioridad 1, 2 y 3 se obtiene un Nivel de Adecuación *Triple A (AAA)*.

2.1.2. Principios de WCAG 2.0. En WCAG 2.0 [14] se definen 4 principios que proporcionan los fundamentos de la accesibilidad web (Perceptible, Operable, Comprensible y Robusto), 12 pautas que llevan asociados criterios de conformidad estando definidos 3 niveles de conformidad (A, AA y AAA). A su vez, por cada pauta y sus criterios de conformidad se han documentado una serie de técnicas, algunas consideradas suficientes y otras recomendables para satisfacer los criterios de conformidad. A continuación se ofrece una breve descripción de los principios y pautas:

Principio 1: Perceptible - *La* información *y los* componentes *de la interfaz de usuario deben ser* presentados a *los* usuarios *de modo que* ellos puedan percibirlos. *Las pautas que se desprenden de este principio son: proporcionar* alternativas textuales para *todo* contenido no textual *de modo que se pueda convertir a otros formatos que las personas necesiten, tales como textos ampliados, braille, voz, símbolos o en un lenguaje más simple;* p*roporcionar* alternativas *para los* medios tempodependientes*; crear* contenido *que pueda* presentarse de diferentes formas *(por ejemplo, con una disposición más simple)* sin perder información o estructura; *facilitar a los usuarios* ver y oír el contenido,

incluyendo la separación entre el primer plano y el fondo.

Principio 2: Operable - Los componentes de la interfaz de usuario y la navegación deben ser operable. Las pautas que se desprenden de este principio son: proporcionar acceso *a toda la funcionalidad* mediante *el* teclado; *proporcionar a los usuarios el* tiempo suficiente *para leer y usar el contenido; considerar, en el diseño de contenidos,* criterios de diseño *que* eviten *la ocurrencia de* ataques, espasmos *o* convulsiones*; proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran*.

Principio 3: Comprensible - La información y el manejo de la interfaz de usuario deben ser comprensibles. Las pautas que se desprenden de este principio son: hacer que los contenidos textuales *resulten* legibles *y* comprensibles; *hacer que las* páginas web aparezcan *y* operen *de manera* predecible; *ayudar a los usuarios a* evitar y corregir *los* errores.

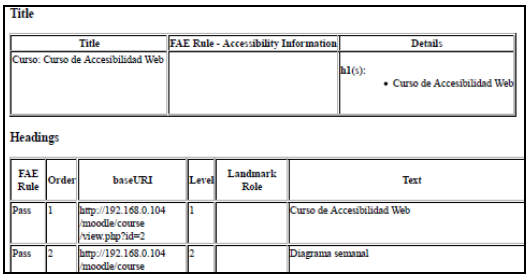
Principio 4: Robusto - El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas. De este principio se desprende una única pauta: *maximizar* la *compatibilidad* con las *aplicaciones* de usuario *actuales* y *futuras*, incluyendo las ayudas técnicas.

2.2. Primera evaluación

Para realizar la primera evaluación se utilizaron las herramientas para evaluar la accesibilidad, mencionadas previamente.

Como objetos de evaluación se seleccionaron cuatro páginas distintas de la plataforma: *Inicio*, *Ingreso*, *Curso* y *Perfil*. La opción se basó en que estas páginas son fundamentales para utilizar Moodle ya que, cualquier persona que desee utilizar las funcionalidades básicas de la plataforma debe acceder a las mismas. En cada una, se evaluó el cumplimiento de WCAG 1.0 y 2.0.

La Accessibility toolbar realiza una evaluación minuciosa de cada elemento HTML de la página, como puede observarse en la extracción de parte de un reporte en *Figura 1*. Estos resultados sirven como base para el análisis realizado en la primera evaluación de este documento. La prueba consiste en verificar el cumplimiento de todas las pautas de la norma. En los casos de no cumplimiento, se identifican los incumplimientos.



Title

Title	FAE Rule - Accessibility Information	Details
Curso: Curso de Accesibilidad Web		h1(s): • Curso de Accesibilidad Web

Headings

FAE Rule	Order	baseURI	Level	Landmark Role	Text
Pass	1	http://192.168.0.104/moodle/course/view.php?id=2	1		Curso de Accesibilidad Web
Pass	2	http://192.168.0.104/moodle/course	2		Diagrama semanal

Fig.1: evaluación de herramienta Accesibility toolbar

2.3. Segunda evaluación

Para realizar una evaluación del Moodle 2.7, considerando la accesibilidad para personas ciegas o con grave disminución visual, se solicitó a personas sin problemas visuales pero no familiarizadas con la interfaz de la plataforma, que hicieran las pruebas sin acceso visual a la pantalla utilizando el software de ayuda técnica NVDA. El procedimiento para realizar la prueba fue el siguiente:

1. Se colocó una cartulina, a modo de cobertura, delante del monitor para evitar que el individuo, sentado en frente, pudiera ver el entorno.
2. Se consideró que una persona ciega está familiarizada con el teclado, por lo cual a los voluntarios se les permitió visualizar el mismo.
3. Se requirió a los voluntarios que, bajo las condiciones previas, llegaran a un objetivo, a partir de la navegación a través de cada página específica del Moodle 2.7, que consistía en acceder a links, buscar o rellenar formularios, dependiendo el caso de prueba planteado.

3. Resultados

3.1. Primera evaluación

Luego de efectuar la evaluación, se pudo observar el incumplimiento de los siguientes criterios definidos por al WCAG 2.0:

- *Contraste mínimo*: La relación de contraste observada en el Template Clean es de 2.61:1 (resultado obtenido con la herramienta Contrast Checker), mientras que el mínimo exigido por la Pauta 1.4.1 es de al menos 4.5:1.
- *Cambio de tamaño del texto*: Según la pauta 1.4.4, el texto debería poder ser ajustado sin ayudas de software de terceros. El entorno virtual Moodle no provee opciones para ajustar dicho tamaño.
- *Comprensible*: Algunos textos en controles, como botones, no proveen una experiencia de usuario simple, ya que no describen claramente la función que realizan al presionarlos. Este podría ser un inconveniente para personas que no están familiarizadas con las

Tecnologías de la Información y tienen como primer contacto el uso de la plataforma Moodle, o bien, personas con déficit de atención o dificultades de aprendizaje.

3.2. Segunda evaluación

En la *tabla 1* se informan los problemas observados durante la experiencia y, además, se incluyen recomendaciones desde el punto de vista del desarrollador de software.

A partir de las observaciones realizadas, en base al checklist incluido en las pautas WCAG 2.0 [13], se incluyeron en el cuadro características consideradas positivas y negativas. En la columna de sugerencias, se realizan propuestas para remediar estas últimas.

Las conclusiones se basan en la observación del comportamiento de los individuos que realizaron la prueba y las dificultades que se les plantearon al momento de las mismas.

Tabla 1: Resultados de las pruebas realizadas a con NVDA sobre Moodle 2.7

Pantalla	Objetivo	Características Positivas	Características Negativas	Sugerencias
Pantalla Inicial	Acceder al inicio de sesión de la aplicación.	Navegable mediante tecla tab con un orden lógico de contenido. Las secciones que necesitan permisos dirigen a la pantalla de autenticación	El usuario hace el recorrido completo reiteradas veces y no encuentra el enlace de Inicio de Sesión porque no posee una etiqueta que lo identifique correctamente.	Colocar en un contenedor, con título adecuado, el inicio de sesión. Asociar un Tab con el texto "Usted no se ha identificado" *(referencia a punto de verificación 2.4).*
Pantalla de inicio de sesión.	Iniciar sesión en la plataforma.	Navegable mediante tecla tab con orden lógico de contenido. Se puede ingresar al campo Usuario y Contraseña con facilidad. Funciona correctamente el botón de ayuda seleccionable.	El nombre no es lo suficientemente representativo y no cuenta con instrucciones que orienten al usuario sobre lo que debería hacer para iniciar sesión.	Además de la etiqueta “Entrar” se podría agregar un texto más descriptivo que el lector de pantallas pudiera utilizar para ayudar al usuario a identificarlo rápidamente. *(referencia a punto de verificación 3.1.1.)*
Pantalla principal con usuario logueado	Ingresar al curso de accesibilidad Web.	La navegación es correcta		
Pantalla Curso de accesibilidad Web con usuario logueado	Encontrar las novedades para la primera semana de agosto.	En un solo recorrido a través del camino de tecla tab cumple el objetivo.	El calendario no estaba en el mismo idioma del entorno. El usuario reingresaba en reiteradas ocasiones a Novedades por la existencia de un menú recursivo.	Los plugins, al ser independientes de la plataforma, pueden contar con inconsistencias de idiomas y disminuir el nivel de accesibilidad obtenido por el entorno. Deberían definirse normas para el desarrollo de complementos que aseguren el cumplimiento de los estándares. Se podría obviar la selección mediante Tab de la página en la cual se encuentra actualmente *(referencia a punto de verificación 3.1.1.)* Los links que llevan a la pantalla actual deberían estar desactivados *(referencia a punto de verificación 3.2.3.)*

4. Discusión

El objetivo de las evaluaciones efectuadas fue verificar el cumplimiento de las pautas a un nivel de accesibilidad AA. Luego de un análisis detallado de los resultados se puede observar que para cumplir dicho nivel conforme las pautas de accesibilidad WCAG 2.0, aún resta ajustar algunos detalles, tanto desde el punto de vista de la inclusión de nuevos requerimientos funcionales como desde el punto de vista de los requisitos no funcionales relacionados con la interfaz de usuario.

Pudo observarse que Moodle depende de un gran número de plugins o complementos que agregan funcionalidad y son de simple instalación. En términos generales y, para usuarios que no presentan inconvenientes en el uso de las Tecnologías de la información, estos complementos son de gran utilidad. El inconveniente es que al ser independientes de la plataforma, y en algunos casos desarrollados por equipos ajenos al Moodle, existen incompatibilidades con las pautas de accesibilidad, e incluso con los idiomas ya que no logran integrarse en su totalidad a la plataforma y sus configuracionescomo es el caso del idioma.

En este sentido se propone a las instituciones educativas que al momento de añadir un complemento para apoyar sus procesos de enseñanza, sometan al mismo a un control de accesibilidad a efectos de garantizar un cumplimiento acorde al que se obtiene con toda la plataforma.

En relación con la interfaz gráfica, si bien presenta importantes mejoras, no cumple con los requerimientos de contraste entre los grises de las cajas que agrupan las opciones.

No se debe perder de vista que la evaluación fue realizada sin tener en cuenta el contenido disponible en la plataforma, es decir, el que es generado por los usuarios (profesores y alumnos). Para asegurar la accesibilidad completa, se debe verificar que el contenido proveniente de diversas fuentes cumpla también con las pautas de accesibilidad. Esto sin duda demanda gran esfuerzo por parte de quienes los desarrollan, pero se considera que es el único camino para garantizar la inclusión a todos los potenciales usuarios.

5. Conclusiones

Teniendo en cuenta las evaluaciones realizadas, en lo que respecta a la WCAG 2.0, el Moodle 2.7 no es accesible en el nivel AA atento a que no cumple con todas las pautas de dicho nivel. Específicamente las pautas de Contraste Mínimo, Cambio de tamaño de Texto y el principio Comprensible.

Aún queda camino por recorrer en lo referido a accesibilidad web en la plataforma de aprendizaje Moodle, lo cual requerirá de una evolución, en conjunto, de las herramientas de desarrollo y de evaluación.

Además, para garantizar que la accesibilidad provista por el entorno no sea disminuida por complementos externos, se debe analizar el cumplimiento de las pautas correspondientes antes de decidir la incorporación de éstos a la plataforma.

La evaluación de plataformas virtuales de enseñanza y aprendizaje en lo relacionado con la accesibilidad constituye un aporte importante a las Instituciones Educativas, ya que se trata de un requerimiento fundamental para garantizar una mayor inclusión en la oferta universitaria. Es fundamental reconocer la necesidad de cumplir con ciertos estándares para el desarrollo Web que permita evolucionar hacia diseños que garanticen el acceso desde diferentes dispositivos y a personas con discapacidades visuales o auditivas. La responsabilidad social, universitaria y empresaria debe centrar su atención en este aspecto y avanzar en este sentido. Las conclusiones de este trabajo serán sistematizadas y publicadas para aportar información a la comunidad de desarrollo Moodle y a los usuarios de esta importante plataforma de enseñanza y aprendizaje.

6. Referencias

[1] Bravo García, Alejandro Gonzalo, "Accesibilidad web. Un problema pendiente.", *Universitat Autonoma de Barcelona*, 2006.

[2] Leonor Betegón Sánchez, Marta Fossas Olalla, Elena Martínez Rodriguez, Mª del Mar Ramos González, *"Entornos virtuales como apoyo a la docencia universitaria presencial: utilidad de Moodle"*, Anuario Jurídico y Económico Escurialense, XLIII (2010) 273-302 / ISSN: 1133-3677

[3] Nuevas características de Moodle 2.7, disponible en: http://docs.Moodle.org/all/es/Nuevas_caracter%C3%ADsticas_de_Moodle_2.7, 10 de Julio 2014.

[4] W3C (sf) Introducción a la accesibilidad web. Recuperado el 3 de abril de 2013 de http://www.w3c.es/Traducciones/es/WAI/intro/accessibility

[5] Fundación Sidar - Acceso Universal (2009) *Web Content Accessibility Guidelines (WCAG) 2.0*. Recuperado el 3 de abril de 2013 de http://www.sidar.org/traducciones/wcag20/es/

[6] Liliana Cuenca Pletsch, Diana Salinas Gómez, *"Accesibilidad en Objetos de Aprendizaje*", Universidad Tecnológica Nacional, Facultad Regional Resistencia, 2013, pp 4-6.

[7] NonVisual Desktop Access (NVDA), disponible en: http://www.nvaccess.org/.

[8] Jaws, disponible en: http://webaim.org/articles/jaws/

[9] Ramón Voces, Lluís Codina, *"La accesibilidad potencial y real del formato pdf: análisis de diarios digitales españoles"*, Volume 17, Number 2 / March-April 2008.

[10] Firefox Accessibility Toolbar Extension, disponible en: http://firefox.cita.illinois.edu/ University of Illionois, sf.

[11] Web developer toolbar, disponible en: http://chrispederick.com/work/web-developer/

[12] Traducción por Sergio Luján Mora, Evaluación de la accesibilidad de sitios web con la barra de herramientas de Firefox Web Developer Toolbar, disponible en: http://accesibilidadweb.dlsi.ua.es/?menu=hr-webdevelopertoolbar.

[13] W3C (sf) Introducción a la accesibilidad web. Recuperado el 3 de abril de 2013 de http://www.w3c.es/Traducciones/es/WAI/intro/accessibility

[14] Fundación Sidar - Acceso Universal (2009) *Web Content Accessibility Guidelines (WCAG) 2.0*. Recuperado el 3 de abril de 2013 de http://www.sidar.org/traducciones/wcag20/es/

[15] WCAG 2.0 CheckList, disponible en: http://webaim.org/standards/wcag/WCAG2Checklist.pdf

Metodología para determinar la exactitud de una respuesta, escrita en forma textual, a un interrogatorio sobre un tema específico

Paz Menvielle, María Alejandra; Ligorria, Karina; Guzmán, Analía; Casatti, Martín; Groppo, Mario A.; Higa Tamashiro, Seiyu Ricardo; Gimenez, Juan Pablo.
Universidad Nacional Tecnológica Nacional Facultad Regional Córdoba

Abstract

El presente trabajo describe el proceso que realzará análisis de texto en un párrafo, de redacción libre, que es respuesta a preguntas relacionadas a un dominio específico, con el objetivo de detectar si esa respuesta es correcta. El ámbito de prueba se realizará usando como sujetos de prueba a textos escritos por alumnos de una determinada cátedra, que responderán a preguntas de exámenes usando texto de redacción libre.

Para este análisis se definirá el dominio de aplicación tanto en relación a la temática que se trate como en la forma en que el alumno deba dar las respuestas. Se estudiará la necesidad de establecer para las respuestas ciertas restricciones que facilitarían su posterior estudio.

Es importante indicar que este trabajo no pretende profundizar en el conocimiento de la minería de textos ya que, en base a la recuperación de la información desde el texto escrito por el alumno, se tratará de identificar cual es la respuesta provista, y saber si es correcta, en comparación con todas las posibles respuestas a una pregunta dentro del dominio de trabajo.

Palabras Clave: análisis de textos, grafos, análisis sintáctico de textos, análisis semántico de textos, bases de datos NoSQL.

1. Introducción

Ante la necesidad de encontrar el modo de realizar evaluaciones objetivas a exámenes resueltos en computadoras, consistentes en preguntas que los alumnos deben responder en forma de texto de redacción libre, surge este proyecto, cuyo objetivo es realizar análisis de texto en párrafos, de redacción libre, que son respuestas a preguntas relacionadas a un dominio específico, con el fin de detectar si esa respuesta es o no correcta. Para ello se escogió la cátedra Paradigmas de Programación del 2do año de la carrera Ing. en Sistemas de Información en UTN FRC, ya que la misma posee contenidos con dominios simples y directos que facilitan la interpretación de las posibles respuestas de los alumnos.

Es importante indicar que el presente trabajo no pretende profundizar en el conocimiento de la minería de textos ya que, en base a la recuperación de la información desde el texto escrito por el alumno, se buscará identificar cual es la respuesta provista, y si es correcta, en comparación con todas las posibles respuestas a una pregunta dentro del dominio de trabajo.

En cambio, text mining es el análisis de recursos de textos que son nuevos, en los que información previamente desconocida es descubierta. Éste es un campo interdisciplinario que toma técnicas del campo general de data mining y, adicionalmente, combina metodologías de varias otras áreas tales como extracción de información (Information Extraction - IE), recupero de información (Information Retrieval - IR), lingüística computacional, categorización, temas de seguimiento y concepto de vínculo.

Es frecuente confundir el campo de Information Retrieval (IR) y el de text mining, véase [6] y [8]. Esto sucede porque ambos tratan con el texto y sus particularidades, entonces ambos tienen cuestiones de apariencia similar. La diferencia entre estos 2 campos es principalmente su acierto final. En IR, el objetivo es recuperar documentos que parcialmente concuerdan con una consulta y seleccionar de esos documentos algunos de los mejores aciertos. Text mining trata acerca de descubrir hechos desconocidos y verdades ocultas que deben existir en el léxico, la semántica o relación estadística de las colecciones de textos.

Otro campo que ha dejado metodologías a text mining es Extracción de Información (IE). IE difiere de text mining en que extrae datos estructurados específicos (Ej. nombre de personas, ciudades, títulos de libros) y relaciones re-especificadas tal que se descubren nuevas relaciones y generan nuevos diseños. En text mining la información encontrada es no sospechada, en cambio en IE es predefinida y se concuerda con un interés especificado por el usuario. Las técnicas de IE pueden ser parte de las tareas de text mining con el fin de facilitar la extracción de conocimiento.

Las metodologías mencionadas servirán de aporte al presente trabajo ya que, si bien se busca definir si una respuesta es o no correcta, en esta investigación sabemos el contenido de la información que se está tratando (a diferencia de TM), no buscamos palabras o contenidos exactos (a diferencia de IE) y no buscaremos en múltiples documentos relacionados a un tema determinado (a diferencia de IR).

2. Estado actual de conocimiento del tema

El tratamiento de textos con diferentes finalidades, es tema de estudio en diversas disciplinas como medicina, ingeniería, biología, etc. En todos los casos se emplean metodologías originadas en otras áreas tales como extracción de información, recupero de información, lingüística computacional, minería textual, categorización de textos, etc. También se han realizado numerosos trabajos relacionados con el análisis de textos, como son la generación automática de resúmenes y los sistemas de búsquedas de respuestas, entre otros.

A continuación se mencionan algunas de las técnicas conocidas y trabajos realizados que tienen estrecha relación con el presente proyecto.

El estudio de los lenguajes estructurados

El orden lineal de las frases, como su estructura jerárquica, son el tema principal en los formalismos para el análisis sintáctico. Los distintos enfoques consideran esa jerarquía como relaciones entre combinaciones de las palabras o entre palabras mismas.

En el Paradigma de Chomsky se han desarrollado muchos formalismos para la descripción y el análisis sintácticos. El concepto básico de la gramática generativa es simplemente un sistema de reglas que define de una manera formal y precisa un conjunto de secuencias (cadenas a partir de un vocabulario de palabras) que representan las oraciones bien formadas de un lenguaje específico. Las gramáticas son bien conocidas en otras ramas de la ciencia de la computación, las expresiones regulares y las gramáticas independientes del contexto, son gramáticas generativas también.

Chomsky (1965) y posteriormente sus seguidores desarrollaron y formalizaron una teoría gramatical basada en la noción de generación. El trabajo que se realiza en la gramática generativa descansa en la suposición acerca de que la estructura de la oración está organizada jerárquicamente en frases (y por consiguiente en estructura de frase), véase [1] y [2], y éste es el punto de partida de nuestro estudio.

Generación automática de resúmenes

En [3] se define el problema de la generación de resúmenes automáticos a dos diferentes niveles: a nivel de documento y a nivel de grupos de documentos. En el primer caso los resúmenes también pueden ser con enfoque extractivo o abstractivo. Los métodos extractivos se basan en los mismos principios usados en la identificación de términos, consisten en una colección de términos, frases o párrafos significativos que definen el significado del texto original. Los métodos abstractivos dependen de técnicas de parafraseo para producir las síntesis, las técnicas aún están siendo desarrolladas. Un problema común es el de la existencia de múltiples documentos sobre un mismo tema, en este caso se habla de resúmenes a nivel de colecciones de documentos que agrupan o separan los documentos por tópicos y destacan las similitudes y diferencias de la información contenida en ellos. Los contenidos se relacionan entre ellos en un sentido semántico: cubren el mismo tópico y tienen similares categorías semánticas o conceptos estrechamente relacionados.

Categorización de textos

Dado un grupo de objetos, la tarea de clasificarlos consiste en asignarlos a un conjunto pre especificado de categorías. Si estamos dentro del dominio de gestión documental, la tarea se la conoce como categorización de texto, y consiste en hallar uno o más tópicos en los que encajen los contenidos de los documentos; teniendo como entrada un grupo de categorías (sujetos – temas) y un conjunto de documentos de texto. La categorización automática de documentos es una forma de clasificación de patrones, que es necesaria para la gestión eficiente de sistemas de información de texto. Se aplica en el indexado de texto para entrega comercial personalizada de texto, filtrado de spam, categorización de páginas web bajo catálogos jerárquicos, generación automática de metadatos, detección de género de textos, entre otros. Véase [4]. Hay dos enfoques principales a la categorización de textos. Un enfoque de ingeniería del conocimiento en el que mediante reglas de clasificación se introduce conocimiento experto (reglas) y otro el de aprendizaje automático (ML: Machine Learning), en el que procesos inductivos generales construyen un clasificador con aprendizaje basado en ejemplos preclasificados.

Sistemas de búsquedas de respuestas

En [7] se describen los sistemas de búsqueda de respuestas como sistemas diseñados para tomar una pregunta en lenguaje natural y entregar una respuesta. De esta manera los usuarios no tendrían que navegar y leer una o varias páginas de resultados de búsqueda. Estos

sistemas se construyen sobre motores de búsqueda y requieren contenido como fuente para descubrir las respuestas. Deben tener métodos para entender las preguntas del usuario y determinar el tipo de respuesta que debe dar, generar una búsqueda significativa de la consulta, y finalmente calificar los resultados obtenidos. De estos tres problemas el más difícil de enfrentar es determinar el tipo de respuesta. Para hacerlo se realizan tres pasos: entrenamiento, fragmentación y solo entonces la determinación del tipo de respuesta. Para construir un sistema de respuestas se necesita aplicar técnicas de PLN como: Recupero de Información, Extracción de Información con algoritmos de reconocimiento de entidades y etiquetado, comparación de secuencias de caracteres, entre otras.

3. Elementos de trabajo y metodología

El grupo ha investigado técnicas y aplicaciones de software, existentes en el mercado, que permiten realizar análisis sintáctico de oraciones, como también de correctores ortográficos.

Simultáneamente se ha buscado información acerca de la teoría de grafos, su aplicación en la representación de oraciones y su relación con los lenguajes formales, las gramáticas y los autómatas desarrollados por Chomsky, el estudio de su teoría y su evolución.

Tareas clave:

Para realizar el análisis del texto, se inicia con el análisis ortográfico de cada palabra (ya que es necesario que la palabra esté correctamente escrita a fin que el sistema la comprenda) y se obtiene una segunda versión de la oración sin errores ortográficos, así será posible saber si las palabras pertenecen o no al dominio de trabajo.

Para lograr un correcto análisis del texto, es de vital importancia que el mismo no posea errores del tipo ortográfico ni gramatical (aspecto necesario para que el sistema pueda comprender el texto recibido). Debido a la posibilidad que un usuario del sistema ingrese un texto que no cumpla adecuadamente ambas características, surge la necesidad de brindar ayuda de contexto que permita identificar las falencias mencionadas para poder ser tratadas adecuadamente.

Existe gran cantidad de proyectos open source (modelos de desarrollo de software que promueven su acceso universal, de libre uso y libre distribución) que tratan las características del tema abordado, es por ello que se ha optado por investigar las librerías ya desarrolladas y escoger la que mejor se adapte a nuestro propósito. La posibilidad de reutilizar código ya existente ahorra tiempo fundamental en el desarrollo del software. Sin embargo, es factible que el código seleccionado posea errores que representen un costo futuro al modificar y utilizar el mismo, es por ello que para su selección se tuvieron en cuenta las cuestiones mencionadas.

Las librerías estudiadas presentan diferentes formas de abordar el análisis de textos así como diversidad en las interfaces e integración con lenguajes de programación. Luego del proceso de selección, concluimos en dos de ellas que se adaptan mejor al presente trabajo, ya que cumplen con los requerimientos y principalmente poseen una fácil integración en JAVA, ellas son JOrtho y LanguageTool.

Luego de un exhaustivo análisis de las diferencias entre ambas herramientas, se seleccionó LanguageTool. Entre las ventajas que encontramos con respecto a JOrtho, podemos mencionar que la primera posee una implementación con Maven, lo cual garantiza un diccionario actualizado, al igual que las reglas gramaticales. Otras ventajas son la facilidad de la implementación de la librería y la posibilidad de modificar los diccionarios y reglas ya existentes y/o agregar más.

LanguageTool es un proyecto que se encuentra bajo la licencia GNU Lesser General Public License(2.1), el cual brinda verificación gramatical y ortográfica. El proyecto está preparado para ser utilizado en diversos tipos de aplicaciones, lo que garantiza su portabilidad. Entre las implementaciones posibles encontramos el complemento para LibreOffice y OpenOffice, el complemento para el navegador Firefox versión stand-alone de escritorio, entre otras.

Al Investigar en la Wiki de LanguageTool y en los foros del proyecto, pudimos observar que la comunidad de usuarios que han implementado esta herramienta es superior a las otras herramientas estudiadas, esta situación redundará en mayores posibilidades de evacuar las inquietudes que surjan y obtener soporte técnico.

4. Ayuda de contexto

Tanto en la etapa de ingreso de preguntas como de respuestas, el sistema llevará a cabo el análisis del texto en busca de errores ortográficos y/o gramaticales.

El sistema estará diseñado para ser utilizado básicamente por dos tipos de usuario (o dos roles): quien da de alta las preguntas y quien dará respuesta a dichas preguntas. Pudiendo un mismo usuario ocupar ambos roles, si así lo requiere. Solo el usuario/actor que ingresa las preguntas tendrá la posibilidad de agregar palabras

y/o reglas gramaticales a la base de datos del sistema. Al momento de ingresar una pregunta, el sistema señalará las palabras que forman parte de la pregunta y que no se encuentran en la base de datos, solicitando al usuario que agregue dicho término a la base de datos o, caso contrario, que cambie el término empleado por otro que si se encuentre en la base de datos.

En el lenguaje natural las palabras pueden contar con sinónimos, el sistema no empleará términos sinónimos propias del idioma sino que solicitará al usuario gestor de las preguntas que ingrese términos equivalentes. Esto es necesario porque para determinar si un término es equivalente o no a otro es necesario contar con el expertise del dominio de conocimiento al que pertenece la pregunta, ya que un término puede ser equivalente a otro en un dominio de conocimiento y no serlo en otro dominio de conocimiento.

Por ejemplo, si buscamos en un diccionario español de sinónimos la palabra “objeto” encontramos materia, asunto, finalidad, pieza, etc. Pero si usamos el término “objeto” desde el punto de vista del Paradigma Orientado a Objetos estos sinónimos no son aplicables, o al menos no todos. En este caso el término o palabra “objeto” tiene como equivalencia el término “entidad”, pero el mismo término “objeto” en el Paradigma Lógico tiene como equivalencia el término “constante individual” y no entidad.

Frente a este requerimiento, se permitirá la carga de palabras y equivalencias por dominio de conocimiento. Para ello se propone que previamente al agregado de los nuevos términos los docentes puedan consultar una lista de sinónimos o realizar una búsqueda asociada a los conceptos. La misma posiblemente obtenga un término adecuado ya existente y evite de esta forma la carga de una palabra nueva.

Al momento de completar las respuestas (por parte del alumno), se considera conveniente dejar al usuario gestor de las preguntas la decisión de utilizar o no esta herramienta. En caso de no emplearla, se minimiza el impacto que generaría en el servidor, y en los enlaces de las redes de comunicaciones, las correcciones en tiempo de ejecución durante el desarrollo de un examen. Otra importante razón de esta elección se basa en la importancia que los errores sean identificados por el propio usuario, ya que es probable que existan palabras y/o reglas de escritura no contempladas en la aplicación y que podrían ser agregadas a la base de datos del sistema logrando, de esta manera, su actualización permanente.

El agregado de nuevas palabras a la ayuda de contexto debe ser hecho de forma ordenada y coherente, para evitar sobrecargar la base de conocimientos con términos que tengan poco impacto en la evaluación de los conceptos estudiados.

Se implementará también un mecanismo de depuración que periódicamente recorra la base de datos y reporte los conceptos que más palabras contextuales tengan, lo que las haría candidatas para su simplificación y de esta manera mantendría una base de datos cohesiva y minimalista, con conceptos de gran valor para la corrección pero con una terminología acotada.

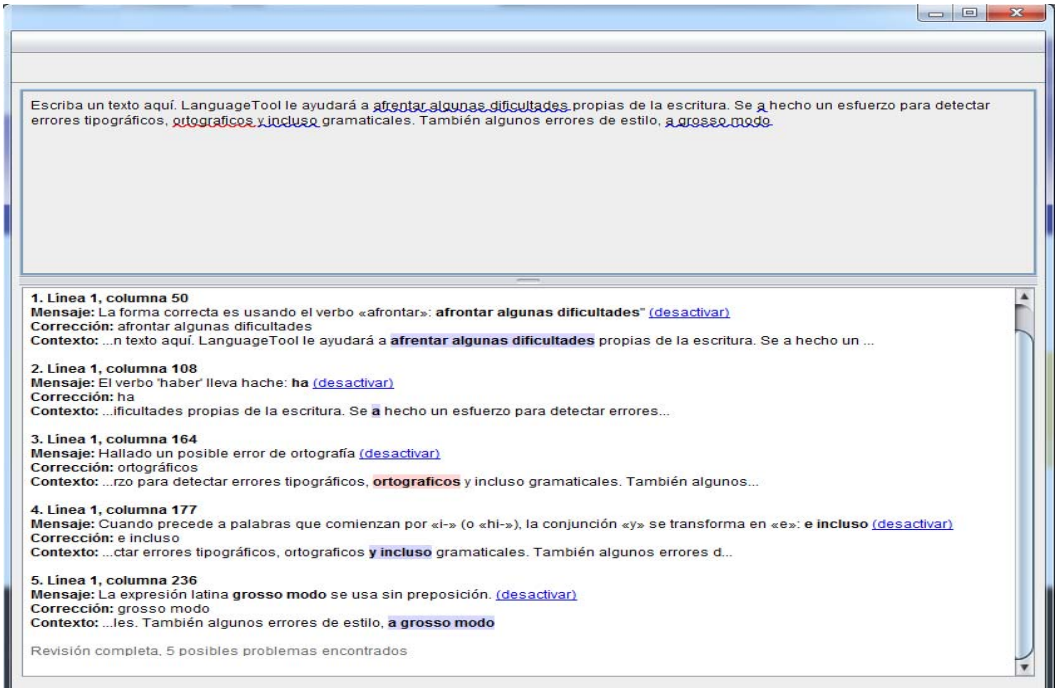


Figura 1. Ejemplo UX

5. Implementación

Una de las principales ventajas de que nos otorga la herramienta incluida en este proyecto, es el soporte provisto en la Wiki del proyecto, véase [15]. Para la implementación de esta herramienta, utilizaremos una aplicación JAVA, véase [16].

Como parte del proyecto de investigación, se está diseñando una interfaz que reúna las características necesarias para este trabajo en particular, logrando una óptima UX (User Experience) y adecuando tanto el diccionario como las reglas de gramática, al dominio abordado.

Una prueba de concepto, preliminar, es la imagen que mostramos previamente en la figura 1.

Una vez obtenido el texto (al que llamaremos texto respuesta) sintácticamente correcto, la tarea más importante y de mayor complejidad es el p*re-procesamiento y transformación* del mismo. Dado que estaremos en presencia de párrafos de escritura libre, es necesario transformar el texto respuesta, en datos aptos para realizar el proceso de análisis cuidando de no alterar el significado de los mismos.

A este proceso se lo puede subdividir en:

a. Selección de datos relevantes: busca eliminar entradas duplicadas o redundantes y detectar anomalías en el conjunto de datos.

b. Reducción de datos: seleccionar las características adecuadas para el proceso de análisis de los datos

c. Preparación: realiza la limpieza, integración, transformación y reducción de dimensiones sobre los datos seleccionados.

Es importante destacar que gran parte del tiempo y esfuerzo computacional de todo el proceso se invierte en esta etapa, y que puede ser la más importante ya que los resultados obtenidos dependerán, en gran parte, de la calidad de los datos con los cuales se trabaja.

En relación al procesamiento del texto, durante la investigación preliminar surgió la necesidad de realizar el tratamiento en diferentes niveles: desde el nivel de palabra a palabra, frase (no usada realmente en la actualidad), respuesta, conjunto de respuestas, conjunto de respuestas con enlaces y aplicación. Cada nivel de procesamiento revela diferente información sobre el texto. Como ejemplo, a continuación se expone brevemente el nivel de palabra para procesar un texto, véase [9], donde se pueden procesar las palabras en base a diferentes criterios y propiedades:

Propiedades: Homónimos, sinónimos, hiponimia.

Stop-Words: Palabras que desde un punto de vista no-lingüístico, no portan ninguna información. Lo normal es quitarlas para procesar el texto. En Español pueden ser palabras como: a, hacia, y, para, por, cuando, etc.

Stemming: Normalmente una misma palabra se puede representar de diferentes maneras. Esto representa un problema porque se escriben diferente pero tienen un significado similar (ej. aprender, aprendía, aprendiendo). Stemming es el proceso de transformar una palabra en su raíz (stem). Existen algoritmos muy buenos para el idioma inglés, pero para otros idiomas el estado del arte no está tan avanzado. El Stemming en inglés, se basa en un conjunto de reglas, en las que dado un sufijo de una palabra, lo cambian por otro, por ejemplo "TIONAL"-> "TION" (ej. conditional->condition).

N-Grams Frecuentes: N-Gram es una secuencia de N palabras consecutivas, por ejemplo "fer martin" es un 2-Gram. N-Grams frecuentes son los que aparecen en el texto una o más veces.

Relaciones Léxicas: o relaciones de las palabras.

Este procesamiento es la clave del proyecto. Es importante optimizar la resolución del problema y la solución deberá provenir desde la presente investigación. No es sencillo, un lenguaje consiste en un alfabeto, una gramática y un conjunto de reglas que definen la sintaxis. El alfabeto es el conjunto de símbolos usados por un lenguaje. La gramática de un lenguaje es un conjunto de reglas que definen cómo los símbolos de un alfabeto pueden interactuar uno con otro, mientras la sintaxis consiste en reglas que capturan el modo que las palabras pueden ser unidas para formar una sentencia, véase [5] y [10]. Todas las “fugas de las gramáticas” suceden porque la gente tiende a usar el lenguaje libremente, sin adherirse a las reglas. Esto se evidencia, por ejemplo, en los e-mails y diálogos de chat donde expresiones mal formadas son frecuentemente usadas en aras de la simplicidad.

Describir texto partiendo sólo de la gramática puede conducir a identificaciones erróneas, inhabilidad para capturar errores sintácticos en el texto o para identificar ciertos ítems tales como nombres, aunque las reglas sintácticas básicas puedan capturar patrones claves en la estructura del lenguaje.

6. Resultados

Durante el estudio e investigación de técnicas y metodologías existentes aplicables al reconocimiento de textos, más los antecedentes aplicados a distintas áreas como ciencia, medicina, administración, industria, etc.; se detectó su limitación y que sería necesario incursionar en otras áreas. En virtud del análisis realizado, se decidió aplicar teoría de grafos para la descripción del dominio de estudio.

Existe numerosa bibliografía sobre teoría de grafos, y se han construido herramientas de uso libre que permiten graficar y construir el camino crítico entre otras tareas.

La teoría de grafos (*Graph Theory*), Dieter (2004), es ampliamente utilizada tanto en ingeniería como en otras disciplinas relacionadas como la investigación operativa, diseño de redes, ingeniería de organización industrial, la logística y el transporte, etc. Un grafo puede representar en forma de red un modelo de una realidad empresarial. Este modelo podrá ser analizado desde distintos puntos de vista. En el presente proyecto se incluyó la aplicación de grafos para establecer la relación entre las palabras que conforman oraciones dentro de un dominio específico.

Sabiendo que las palabras están escritas correctamente, se está realizando la aplicación de la técnica de grafos tanto para conocer si las oraciones pertenecen al dominio en cuestión, si están construidas correctamente y que correspondan a su respuesta correcta, así como para explicar el dominio de trabajo.

Se continúa el análisis definiendo detalladamente el sub-dominio de aplicación. Se analiza la necesidad de establecer restricciones tanto en relación a la temática que se trate como en la forma en que el alumno deba dar las respuestas. La pregunta deberá estar estructurada de tal manera que propicie una respuesta directa y acotada. Así, por ejemplo, no se podrá preguntar ¿cuáles son las características fundamentales del paradigma orientado a objetos? Sino ¿qué denota la característica fundamental de la abstracción?, lo que limita la respuesta. Así también, el gestor de preguntas, podrá agregar una frase/palabra que de inicio a la posible respuesta del alumno. Siguiendo el ejemplo anterior, a la pregunta ¿qué denota la característica fundamental de la abstracción? A la respuesta del alumno el sistema le puede dar la introducción (previamente ingresada por el gestor de preguntas): “La característica fundamental de la abstracción denota:”.

7. Conclusiones

El análisis de textos desde esta perspectiva continuará ampliando el aprendizaje sobre la materia y las distintas técnicas y metodologías existentes, pudiendo de esa manera acercar a la comunidad académica un poco más a la solución de la problemática permanente que significa que un computador pueda analizar textos correctamente.

La utilización de la metodología propuesta busca establecer un marco de evaluación común a todos los docentes de la cátedra a la vez que reducir los tiempos de corrección.

Al estar la solución creada sobre una base de conocimientos ampliable, los docentes tienen la posibilidad de ir agregando casos no vistos anteriormente y extender de esa manera tanto los temas a corregir, como los conceptos involucrados en la evaluación.

Hay que mencionar, no obstante, que la metodología y la herramienta asociada no buscan reemplazar la intervención de un docente en la evaluación de los conocimientos de la materia sino ayudar a que dicha evaluación sea consistente, sin importar el docente que la realice. La minimización de los tiempos de análisis también permitirá al docente enfocarse más en los casos particulares y específicos que en la generalidad de la evaluación, que al ser previsible y repetitiva, puede ser fácilmente reemplazada por un proceso automatizado. Además el gestor de preguntas y el responsable de brindar respuesta a dichas preguntas, podrán disponer de manera instantánea una aproximación de la exactitud o no de la respuesta brindada. En relación al examinado, a diferencia de los sistemas de evaluación múltiple opción, en este caso a través del desarrollo de respuestas escritas en forma textual, se permitirá al alumno enriquecer su capacidad de expresión y tender a mejorar su ortografía, entre otras ventajas.

Está prevista la creación del sistema que aplique en forma on-line esta metodología, para que el aplicativo expanda las posibilidades de potenciales usuarios, pudiendo generar intercambio de ideas con personas relacionadas al tema desde otras cátedras. El desarrollo propio será necesario porque las herramientas existentes no permiten la integración total de los nuevos métodos de evaluación de las respuestas de examen que se prevé implementar en este proyecto.

El sistema podrá ser utilizado no solo a nivel universitario sino aplicado en cualquier contexto que requiera el análisis de texto derivado de preguntas a desarrollar, a nivel escolar o no.

8. Bibliografía

[1] “Teoría de la Computación”, J Glenn Brookshear. Ed Addison Wesley – 1989.

[2] “Teoría de Autómatas, Lenguajes y Computación”. J. Hopcroft, R, Motwani, J. Ullman. Ed. Pearson- Addison Wesley – 2008.

[3] “Aplicaciones de Procesamiento de Lenguaje Natural”. Hernández M._ Gómez J. Revista Politécnica, Julio 2013, Vol. 32, No. 1, Páginas: 92–96

[4] “The text mining handbook”, R. Feldman and J. Sanger. Cambridge University Press. Diciembre

11, 2006. También disponible on line: www.safaribooksonline.com

[5] “Lenguajes, Gramáticas y Autómatas, un enfoque práctico”. Isasi, P., Martines, P., Borrajo, D. Ed Addison-Wesley. 1997.

[6] “Minería de Datos basada en sistemas inteligentes”. Britos, P., García Martínes, R., Hossian, A. y Sierra, E. Editorial NUEVA LIBRERÍA. 2005.

[7] “Introduction to information retrieval”. C. D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Cambridge University Press. Julio 2008. También disponible on line: www.safaribooksonline.com

[8] “Tratamiento de noticias empresariales en la Web, mediante minería textual”. Ismael Esquivel Gámez. Universidad Popular Autónoma del Estado de Puebla, México. 2007.

[9] “Minería de texto para actualización automática de documentos”. Pérez Abellería, M.A. y Cardoso, C.A., J. Universidad Católica de Salta. 2007.

[10] “Una mirada por el procesamiento de texto no estructurado para la extracción de conocimiento e información: Aplicado a la medicina”. Montaño Ramirez, D. Universidad de Colombia. 2007.

[11] “Recuperación de Información de BD no estructuradas utilizando técnicas de minerías de textos que sean aplicables en el ámbito médico”. Aguilar J.P. Univ Nacional de Colombia, Bogotá. 2008

[12] “Clasification of text, automatic”. In The Encyclopedia of Language and Linguistics. 2da Ed. Sebastianini, F. 2006.

[13] “La Minería de Texto: perspectiva metodológica para la realización de resúmenes documentales”. Castillo Zayas, M. y Leiva Mederos, A. ACIMED v.15 n.5 Ciudad de La Habana. Mayo 2007.

[14] “Herramientas de gestión del conocimiento”. Grau, A. Fundación Iberoamericana del Conocimiento. 2003.

[15] http://wiki.languagetool.org/

[16] http://wiki.languagetool.org/java-api

Datos de Contacto:
Paz Menvielle, M.A .pazmalejandra@gmail.com
Ligorria, Karina. karinaligorria@hotmail.com
Guzmán, A. aguzman@sistemas.frc.utn.edu.ar
Casatti, Martín. mcasatti@gmail.com
Groppo, Mario A. proyale@groppo.com.ar
Higa Tamashiro, SR..rickyssht@gmail.com
Gimenez, Juan. juan.gimenez10@yahoo.com

Universidad Nacional Tecnológica Nacional –Facultad Regional Córdoba, Marcelo López esq Cruz Roja Argentina. Córdoba

Análisis de stakeholders en el desarrollo de proyectos para e-learning en contextos universitarios

Lucila Romero
Universidad Nacional del Litoral, Facultad de Ingeniería y Ciencias Hídrica
Ciudad Universitaria. Ruta Nacional N° 168 Km 472,4 (3000) Santa Fe, Argentina
Universidad Católica de Santiago del Estero, Dirección Académica Rafaela
lucila.rb@gmail.com

Marcela Andrea Vera
Universidad Nacional del Litoral, Facultad de Ingeniería Química
Santiago del Estero 2829 (3000) Santa Fe, Argentina
Universidad Católica de Santiago del Estero, Dirección Académica Rafaela
Bv. Hipólito Irigoyen 1502 (2300) Rafaela, Argentina
marcelaandrea.vera@gmail.com

Abstract

The success of an e-learning project depends largely on the quality of the applications and tools used to support the teaching-learning process. Meanwhile, the quality of the software and its implementation depends largely on proper identification of stakeholders in the stage of requirements elicitation of it.

While there are studies that have identified key stakeholders in e-learning contexts, most of them do not perform a deep analysis of them, which is necessary to consider the pedagogical and cultural issues that favor the formation of organic and functional space necessary for the administration of education mediated by technologies.

The purpose of this paper is to present the application of a method for the analysis of stakeholders in a software project adapted to the context of e-learning in university environments showing a specific case study.

The result of its implementation cooperates with the construction of a balance between pedagogical components that are kept and others that are introduced to facilitate the changes in the new educational model based on the use of technologies.

1. Introducción

La educación basada en las Tecnologías de Información y Comunicación (TIC) impone el reto de desarrollar un entorno de enseñanza que permita el diseño de material flexible y personalizado que se ajuste a ambientes de trabajo heterogéneos y dinámicos. Como parte de esta dinámica, la Web se constituyó en un universo virtual en constante evolución, exigiendo el desarrollo y la adaptación por parte de los sistemas educativos, generando nuevas estructuras didácticas incorporadas a estos entornos tecnológicos [1].

Cada vez hay más universidades e instituciones educativas que comienzan programas de e-learning. Esta situación genera la constante búsqueda de recursos para afrontar una serie de cuestiones complejas como la planificación, diseño e implementación de cursos y programas adecuados a esta realidad. Por otra parte, la creciente disponibilidad de e-learning presenta un espectro de problemas generados por accesos universales, con marcadas diferencias culturales.

El avance tecnológico en entornos distribuidos posibilita patrones de transmisión de contenidos educativos como la educación a distancia o la educación semipresencial, con la consiguiente proliferación de herramientas que asisten en el desarrollo de estas modalidades de aprendizaje. Por este medio, se intentan resolver limitaciones de disponibilidad en escenarios tradicionales de enseñanza [2]. Se puede contar con interfaces flexibles e intuitivas mediante el aprovechamiento de los recursos tecnológicos, estableciendo ambientes adaptativos donde el proceso de enseñanza se ajusta a las necesidades de cada usuario.

Por otro lado, con el fin de que el e-learning sea aceptado por los educadores, se necesitan herramientas que den soporte al diseño y creación de materiales válidos y confiables, desde una perspectiva pedagógica y técnica [3]. Es decir, es necesario establecer una alineación de la enseñanza, el aprendizaje y el uso de las TIC. En un sentido pedagógico, la enseñanza en ambientes abiertos demanda mayor personalización y eficacia para proveer orientación a los alumnos y soporte en el acceso individual. Es importante tener en cuenta que los recursos educativos son accedidos por personas que difieren en una diversidad de características que demandan la producción de estrategias y materiales de naturaleza mucho más facilitadora que prescriptiva [4].

La identificación de los *stakeholders*/*interesados* representa una actividad fundamental a realizar en el proceso de definición de requerimientos en el desarrollo de proyectos de software. La identificación del grupo de interés clave en este desarrollo conduce a determinar fehacientemente necesidades, responsabilidades y, de esta manera, colaborar con la utilización exitosa de las TIC desarrolladas para e-learning. Esta clara identificación, permite además, descubrir conflictos de intereses entre estos, negociar estos puntos contrapuestos para lograr un proyecto exitoso.

Es de destacar que el éxito de un proyecto de software para e-learning implica el descubrimiento de los *interesados*, de manera de tener en cuenta agentes que pueden afectar o ser afectados por las herramientas desarrolladas y que deben ser considerados como un elemento esencial en la planificación estratégica de dicha herramienta. De esta forma, se busca ampliar el espectro considerado generalmente cuando se menciona este tipo de proyectos en los que se consideran sólo *interesados* evidentes tales como profesores, tutores y alumnos. Luego de la correcta y completa identificación de *interesados*, es necesario el análisis de éstos en el contexto en el que se produce su desarrollo teniendo en cuenta aspectos variados que influyen en su identificación.

Si bien existen trabajos que tienen como finalidad identificar *interesados* en contextos de e-learning [5-6], la mayoría de ellos no presentan un análisis profundo de los mismos, necesario para considerar las cuestiones particulares pedagógicas y culturales que marcan este tipo de contextos. Además, las propuestas encontradas en la bibliografía carecen de la utilización de métodos, técnicas y herramientas para ejecutar esta etapa del desarrollo de software fundamental para garantizar su éxito.

El objetivo de este trabajo es presentar a través de un caso de estudio un método que permita analizar los *interesados* en proyectos de software en contextos de e-learning. Para lograr un correcto análisis, se propone aplicar el método para el análisis de *interesados* propuesto en [7], el cual fue adaptado en el presente trabajo para entornos específicos de enseñanza soportada por las tecnologías focalizando el estudio en universidades privadas. En anteriores trabajos se presentó la aplicación de este método [8] pero considerando una red interorganizacional de universidades públicas y privadas. En esta oportunidad se focaliza el estudio en la dinámica particular de las universidades privadas.

El presente trabajo se organiza de la siguiente manera. En primer lugar se presentan los principales conceptos en torno al caso de estudio utilizado, y en secciones siguientes, se presentan los pasos del método utilizado para el análisis de los *interesados* involucrados en este proceso. Finalmente, la última sección resume y concluye este trabajo.

2. Contexto

El Universo de Información (UdI) es de donde se extraerá la información necesaria para la tarea de elicitación de requerimientos. De esta manera se identifica el contexto general en el cual el software deberá ser desarrollado y operado. Para esto se debe tener en cuenta todas las fuentes de información y las personas relacionadas al software.

A. Entorno operacional

Un LMS (Learning Management System) [9] ofrece una colección de funcionalidades diseñada para transmitir, realizar seguimiento, reportar y administrar contenido de aprendizaje, así como realizar el seguimiento del progreso del alumno y sus interacciones con el material de estudio. LMS puede aplicarse a un sistema simple de administración de cursos, o a un entorno distribuido altamente complejo. Un LMS habitualmente administra accesos, actividades y permisos de los usuarios (inscripción, control de contenidos accedidos, notas de evaluaciones, generación de informes y estadísticas, etc.) y puede proporcionar herramientas de comunicación tanto síncronas (chat, videoconferencia, tutorías en tiempo real, etc.) como asíncronas (foros de discusión, tableros de anuncios, etc.).

Generalmente un LMS se limita a implementar estrategias de enseñanza tradicionales y en muchos casos no están preparados para soportar las diferentes necesidades de los usuarios individuales limitándose a enfocar el "promedio de los usuarios" [10]. Con la adaptabilidad a capacidades y preferencias individuales, e informes de calificaciones devueltos casi en tiempo real [11], la enseñanza basada en este modelo expande las posibilidades de los modelos tradicionales, brindando la posibilidad de alcanzar niveles significativos de personalización.

Un proyecto de software para e-learning puede ser el desarrollo de un LMS en sí mismo, o el desarrollo de alguna herramienta que se pueda integrar a un LMS.

B. Caso de estudio

El presente trabajo comprende la enseñanza universitaria, tanto de grado como extracurricular y en las modalidades presencial, semipresencial y a distancia. Como institución de referencia se considera la Universidad Católica de Santiago del Estero (UCSE).

Los documentos del UdI comprenden formularios, políticas de la organización, manuales, actas de reuniones, contratos con desarrolladores, etc. Debido al tipo de institución involucrada en el desarrollo de software, se tuvieron en cuenta las directivas y normas ministeriales de la nación, es decir, se consideraron normativas del Ministerio de Educación y del Ministerio de Ciencia y Técnica de la Nación, siempre considerando el grado de aceptación del proceso de enseñanza-aprendizaje mediado por TIC. De la misma manera, se tuvieron en cuenta directivas propias de la institución, como

ser: resoluciones de Directores, Resoluciones del Consejo Superior, Resoluciones del Consejo Directivo, planificaciones de cátedra aprobados por el Coordinador Académico, y otros documentos asociados.

Entre las resoluciones de la institución se encuentra la creación del Sistema de Educación Multimedial de UCSE (SEM-UCSE), Resolución N° 150 del año 2012, cuyo objetivo principal es promover e impulsar la Educación a Distancia en la UCSE. Este sistema fue creado para definir líneas de acción tendientes a diseñar la estructura organizacional de educación a distancia de la institución. Para ello, propone redefinir las ofertas académicas, las pautas, los criterios y procedimientos para garantizar la calidad de la educación a distancia y, de esta manera, potenciar la universidad sin fronteras haciendo efectivas la integración en redes institucionales universitarias.

La misión del Sistema de Educación Multimedial de UCSE, según la descripción realizada en la resolución de su creación es "Constituir orgánica y funcionalmente un espacio para la administración de la Educación Asistida destinada a los alumnos libres y de un Sistema de Educación a Distancia en la UCSE". La incorporación de la virtualización en la UCSE, es un factor transformador de sus estructuras y funciones, un instrumento para mejorar su cobertura, calidad, pertinencia y equidad de acceso y una manera de construir una nueva identidad en la *sociedad del conocimiento*.

Teniendo presente este contexto de trabajo particular, se aplica el método para identificar los *interesados* para un proyecto de software.

3. Identificación de interesados para e-learning

Los *interesados* representan la fuente primaria de información. Pueden ser clientes, usuarios, expertos del dominio u otros actores. Un interesado es cualquier individuo, grupo u organización que puede afectar o ser afectado (positiva o negativamente) por el sistema bajo estudio y que tiene influencia directa o indirecta en sus requerimientos [12].

La identificación de los interesados es una tarea crítica para el éxito en un proyecto de software. Es imprescindible tomar esta tarea de manera metódica y sistemática, fundamentalmente cuando se trabaja sobre entornos que involucran diversas organizaciones. Para el trabajo de selección de los *interesados* se seguirá el método propuesto por Ballejos y Montagna [7]. El método es apropiado para ambientes interorganizacionales y consta de los siguientes pasos:

(a) Especificar tipos de *interesados*
(b) Especificar roles de *interesados*
(c) Seleccionar *interesados*
(d) Asociar roles con *interesados*
(e) Analizar influencia e interés.

En los siguientes apartados se irán desarrollando cada uno de los pasos que propone el método adaptado al contexto de e-learning.

A. Especificar tipos de interesados

Cada proyecto de software involucra diferentes *tipos* de *interesados*, teniendo cada al menos un rol asociado. Los tipos están especificados por el dominio, el entorno, la organización: es la clasificación de un conjunto de *interesados* que comparten las mismas propiedades y atributos en la dimensión bajo análisis. Los *interesados* de la organización representan intereses de una institución en particular.

También se pueden distinguir entre *interesados* internos, si pertenecen a la organización; y externos, si tienen interés por el proyecto. Un ejemplo de interesado interno es un profesor universitario y un ejemplo de un interesado externo sería el Ministerio de Educación de la Nación dado que regula el accionar de las universidades de su dependencia. Como parte de la dimensión interna, el método propone considerar el aspecto organizacional y el aspecto interorganizacional (Red interorganizacional, RIO).

Dadas las características del dominio e-learning, en el presente trabajo se propone una nueva división en las dimensiones propuestas por el método: la dimensión interna e-learning. Esta modificación está basada en la consideración que los procesos de enseñanza aprendizaje soportados por TIC imponen una dinámica diferente a una institución educativa. Por ejemplo, el uso de las TIC para la educación permite otorgarle a una Universidad un alcance y una visibilidad diferente y la enseñanza involucra, entre otros aspectos, diseños de contenidos apropiados para el entorno, formación de los docentes en el uso de tecnologías específicas, entre otras cuestiones particulares.

El método propone diferentes criterios de análisis que consideran diferentes puntos de vista y necesidades para la especificación de tipos de *interesados*, estos son:

Funcional: Funciones o tareas que serán afectadas directa (porque el sistema las soportará) o indirectamente (porque sus resultados sirven de entrada o salida para el sistema). Aplicado a la red, este criterio analiza las actividades principales que toman lugar y que son la base para la colaboración entre organizaciones.

Dispersión geográfica: Emplazamientos o áreas geográficas que deben ser incluidos en la selección. Este criterio permitirá la selección de interesados ubicados en lugares geográficamente dispersos, con diferencias idiomáticas y culturales.

Conocimiento/capacidades: Conocimiento o capacidad importante en relación al dominio de desarrollo e implementación del sistema.

Tabla 1. Especificación de tipos de stakeholders

	Dimensión interna			**Dimensión externa**
	Organizacional (ORG)	***Interorganizacional (RIO)***	***e-learning (ELE)***	
Criterio Funcional	-Enseñanza aprendizaje -Planificación de asignaturas -Aprobación de planificación de materias por parte de Secretaría Académica -Diseño o producción de contenido	-Investigación -Elaboración de convenios entre universidades, intercambios docentes	-Enseñanza mediada por TICs, educación a distancia --Aprobación de planificación de materias pertenecientes a carreras mediadas por TICs por Secretaría Académica	
Criterio Dimensión Geográfica	-Universidad Católica de Santiago del Estero -Zona de influencia (alumnos)	-Zonas de influencia de los departamentos académicos	-Zona de influencia e-learning (nacional, internacional) -Dispersión geográfica de docentes que realizan tareas a distancia	
Criterio Conocimiento/ capacidades	-Profesores o docentes universitarios Personal de apoyo a la Gestión	-Director de proyecto de investigación, investigadores, integrantes, becarios -Docentes que realizan tareas de enseñanza en diferentes instituciones pertenecientes a la red -Tesistas, directores de tesis	-Diseñadores de contenidos (contenidistas) - DocentesTutores - Equipo técnico	-Asesores pedagógicos -Profesores de ciencias de la educación -Consultores en TICs aplicadas a la educación -Leyes nacionales
Criterio Nivel Jerárquico	-Niveles jerárquicos de regionales: Directores, Coordinadores Académicos, directores de departamento, directores de carrera -Niveles jerárquicos docentes: Profesores titulares, prof. Adjuntos, Asociados, Jefes de trabajos prácticos, ayudantes	-Ministerio de Educación de la Nación (Gestión Universitaria, Asesoramiento de Gestión Curricular) -Ministerio de Ciencia y Técnica	- Coordinador Administrativo y Tecnológico.	-Niveles jerárquicos ministeriales

Nivel jerárquico: Niveles jerárquicos afectados por el sistema.

La Tabla 1 muestra la especificación de tipos de *interesados* combinando los criterios de análisis y las dimensiones.

En la tabla se puede observar que la dimensión interna presenta una subdivisión entre criterios organizacionales e interorganizacionales (originalmente propuestos por el método) y la dimensión e-learning que expresa cuestiones particulares a tener en cuenta en la enseñanza soportada por las TIC (modificación propuesta en el presente trabajo).

B. Especificar roles de interesados

Un rol de un interesado se define como una colección de atributos definidos que caracterizan un grupo de *interesados* y su relación con la RIO. De los roles propuestos por el método se consideró un subgrupo atendiendo a los roles que se presentan en desarrollos de proyectos de software en el dominio bajo estudio:

- Beneficiario funcional: Se benefician con la implementación del sistema a través de las funciones del mismo.
- Responsable: Encargados del proyecto en todas las etapas del ciclo de vida del sistema.
- Decisor: Controlan el proceso de desarrollo y avance del proyecto. Toman decisiones para lograr acuerdos.
- Usuarios: Usarán el sistema a desarrollar. Interactúan con el sistema y utilizarán sus resultados.
- Experto: Conocen ampliamente el dominio de implementación y pueden brindar mucha colaboración en la elicitación de requerimientos.
- Consultor: Incluye cualquier rol que pueda brindar apoyo para cualquier aspecto del desarrollo del sistema. Son generalmente externos y tienen conocimiento específico sobre cuestiones particulares.
- Desarrollador: Están involucrados de manera directa en el desarrollo del producto de software.

C. Seleccionar interesados

Este paso sirve de guía en la selección concreta de *interesados* que reúnen las condiciones especificadas en el paso A. Esta selección está basada en la Tabla 1. Analizando las características de los criterios en cada dimensión, se deben identificar *interesados* concretos que concuerden con el perfil.

En la Tabla 2 se muestran los interesados seleccionados.

Tabla 2. Selección de interesados

ID	*Interesado*	*Descripción*	*Criterio de identificación*	*Dimensión de Identificación*
DU	Docente universitario	Responsable del proceso de enseñanza aprendizaje de una asignatura correspondiente a una carrera de las universidades involucradas.	-Funcional - Conocimiento	ORG
SA	Coordinador Académico	Responsable de la Coordinación Académica de la Regional. Regulador	- N jerárquico	ORG
DCA	Director de Carrera	Director de carrera en una Facultad. Beneficiario Regulador	- Conocimiento -N jerárquico	ORG
DPI	Director de Proyecto de investigación	Director de Proyecto de Investigación. Desarrollador	-Funcional -Conocimiento	ORG/RIO
IPI	Integrante de Proyecto de Investigación	Integrante de Proyecto de Investigación Desarrollador	-Funcional - Conocimiento	ORG/RIO
DI	Docente Investigador	Responsable del proceso de enseñanza aprendizaje de una asignatura que, a su vez, participa de un proyecto de investigación. Beneficiario del producto.Usuario	-Funcional - Conocimiento	ORG/RIO
CEAD	Coordinador Administrativo y Tecnológico	Responsable del área de Educación a Distancia, de la coordinación de las carreras que se dictan con esta modalidad, los profesores involucrados, tutores, diseñadores de contenido, etc. Beneficiario del producto, experto, regulador	-Funcional - Conocimiento -N jerárquico	ORG/ELE
AG	Alumno Grado	Usuario, beneficiario del producto.	-Funcional	ORG
APOS	Alumno Posgrado	Usuario beneficiario del producto.	-Funcional	ORG
APFC	Alumno de grado que se encuentra realizando un PFC	Desarrollador del producto. Alumno de grado que se encuentra realizando un Trabajo Final de Carrera relacionado con el producto de software.	-Funcional - Conocimiento	RIO
AP	Asesor pedagógico	Asesor pedagógico perteneciente o no a las instituciones de referencia. Experto, Consultor	- Conocimiento	ORG
CTIC	Consultor de TIC	Consultor de TIC aplicadas a la educación.	- Conocimiento	RIO
DIR	Responsable proyecto	Docente Investigador responsable del proyecto Responsable, Decisor	-Funcional - Conocimiento	RIO
AT	Asesor Técnico	Asesor Técnico sobre conectividad y cuestiones relacionadas con los entornos en donde se instalará el producto de software. Consultor	-Funcional -Conocimiento	ORG//ELE
DC	Diseñador de contenido	Diseñador de contenido de material para educación a distancia. Usuario. Beneficiario funcional	-Funcional - Conocimiento	ORG//ELE
TE	Equipo Técnico	Operador de plataforma de elearning. Usuario, consultor	-Funcional - Conocimiento	ORG//ELE
PT	Profesor tutor de carrera	Tutor de carrera de educación a distancia.	-Funcional - Conocimiento	ORG/ELE
CE	Corrector de estilo	Correctores del material para una adecuada comunicación con el alumno. Consultor	-Funcional - Conocimiento	ORG/ELE
TE	tesistas	Tesistas de maestria, doctorales.	-Funcional - Conocimiento	ORG/ELE
DTE	Director de tesis	Directores y codirectores de tesis de posgrado	-Funcional - Conocimiento	ORG/ ELE

Las filas muestran las entidades identificadas y en las diferentes columnas se muestra en forma sucesiva, un identificador, el nombre del interesado, una breve descripción que permita comprender la función que desempeña el interesado en el dominio, junto con la documentación del criterio y dimensión de identificación. UCSE busca desarrollar ofertas académicas a distancia como respuesta a demandas directas del medio, de Rectorado, de las Direcciones Académicas y/o surgidas de la iniciativa de docentes e investigadores que se pueden materializar en: carreras de pregrado, grado y posgrado; carreras de formación técnico profesional; cursos de extensión; cursos de posgrado; módulos del curso de ingreso, asignaturas de carreras de pregrado o grado; etc.

Para esto, el SEM-UCSE administra las aulas virtuales de las cátedras presenciales en la que se combinan diferentes formas de mediación entre los alumnos y los objetos de conocimiento y se crean ambientes de aprendizajes variados y flexibles, según la necesidad. El sistema apela a la creación y desarrollo de diversas estrategias metodológicas, medios y materiales de estudio para establecer una relación adecuada entre quienes participan conjuntamente en el proceso educativo. Esta modalidad propicia y se basa en los procesos de

estudio autónomo, individual o grupal. Y se caracteriza por un contacto diferido entre facilitador y estudiante, el cual se establece a través de medios que permiten el flujo de información y las interacciones entre los actores educativos.

En UCSE, el SEM es el medio facilitador para diferentes propuestas:

- Asesorar, incentivar y apoyar proyectos educativos institucionales que adopten la opción pedagógica y didáctica a distancia.
- Brindar asistencia técnica y servicios en temas pedagógicos y didácticos relativos a la modalidad de educación a distancia a personas físicas o jurídicas, públicas o privadas, ajenas al ámbito de la universidad y de acuerdo a las reglamentaciones para la provisión de servicios vigente.
- Producir materiales educativos e informativos de apoyo a la Educación a Distancia.
- Entablar una comunicación fluida y periódica con las Unidades Académicas y la comunidad educativa.
- Propender a la formación y capacitación continua del profesorado y de los diferentes actores del sistema de Educación a Distancia de la UCSE.
- Entablar, mantener y fortalecer relaciones de cooperación con otros organismos nacionales e internacionales referidos a la Educación a Distancia.
- Implementar y coordinar los cursos y carreras que se ofrecen con la modalidad de Educación Libre Asistida.

En este contexto, un profesor puede estar motivado a implementar e-learning en sus cursos, ya sea por sugerencia de su institución, o puede estar buscando una mayor audiencia para sus cursos, o puede querer beneficiarse con la aplicación de las TIC a la enseñanza para el dictado de los mismos [5], pero también la utilización de TIC resulta imprescindible cuando de educación a distancia se trata.

En este sentido, el profesor se encontrará con el desafío de aprender a utilizar las tecnologías pero contará también con los beneficios correspondientes. Si bien el uso de las TIC beneficia en gran parte el proceso de enseñanza en general permitiendo nuevas posibilidades, demanda mayor tiempo de dedicación a los profesores para la generación de material apropiado para conformar una plataforma y esa plataforma debe ser atendida en tiempo y forma.

Un desafío en educación soportada por TIC es el desarrollo de un entorno de enseñanza que facilite el diseño de materiales de estudio flexibles y personalizados, adecuados a los ambientes heterogéneos de trabajo, que garantice el intercambio y reutilización de los mismos. Un profesor cuenta con diseñadores de contenido que pueden asistirlo en la generación de material de estudio apropiado para las diferentes modalidades de enseñanza y correctores de estilo para una correcta comunicación con el alumno. A través del uso de las TIC se estimula la exploración por parte de los alumnos, otro de los actores principales en el proceso de e-learning [5], su autoexpresión y un sentido de la propiedad y control sobre el proceso de aprendizaje. Se favorece la comunicación, la cooperación entre profesores y alumnos, lo cual naturalmente vuelve al modelo de enseñanza más flexible. De esta manera, se enfatiza la construcción del conocimiento por parte de los alumnos a través de una variada gama de recursos y no la mera recepción pasiva del conocimiento. Estudios recientes muestran que los alumnos esperan un alto uso de las TIC en sus estudios aunque también consideran que esto no asegura necesariamente que el proceso de aprendizaje mejore [6]. El resultado de la utilización de estas herramientas induce a la utilización de componentes en formatos tecnológicos diversos que difícilmente se pueden reutilizar. Esta situación deriva en la necesidad de la administración de componentes educativos que permitan manejar la heterogeneidad [13]. En entornos educativos de este tipo, la interoperabilidad de la información se presenta como un factor clave, dada la necesidad de funcionamiento y operación sobre información heterogénea, proveniente de fuentes diversas, autónomas y dinámicas. Para ello, es necesario contar con asesores técnicos con conocimiento específico.

Sobre esta base, se hace necesario establecer un mecanismo de entendimiento que establezca una conceptualización común, de manera de contar con una estructura integrable que propicie la especialización del trabajo y la posibilidad de compartir y reutilizar los contenidos educativos.

D. Asociar roles con interesados

En la Tabla 3 se muestran algunas asociaciones de roles con *interesados*. Cada interesado puede estar asociado con diferentes roles. En la tabla se muestra la asociación de los diferentes *interesados* resultantes del análisis de varios criterios sobre diferentes dimensiones con los diferentes roles que un interesado con esas características puede asumir. Por ejemplo, un profesor será beneficiado por el producto de software ya que verá automatizado su trabajo de generación de contenido. Los docentes de UCSE serán beneficiados al poder emplear las TIC para optimizar el proceso de enseñanza y se verán acompañados por la institución mediante el asesoramiento, el incentivo y el apoyo a los proyectos educativos institucionales que adopten la opción pedagógica y didáctica a distancia o mediada por TIC. El profesor también será operador o usuario del software y experto en el dominio de trabajo ya que de su experiencia se desprenderán cuestiones a tener en cuenta en el desarrollo.

E. Analizar influencia e interés

Como último paso se debe determinar la influencia y el interés del participante en el proyecto. Para ello,

Tabla 3. Análisis de influencia e interés. Asociación de roles con interesados

Dimensión		*Criterio*	*Interesados*	**Roles**							
				Beneficiario	***Responsable***	***Decisor***	***Regulador***	***Usuario***	***Experto***	***Consultor***	***Desarrollador***
Interna	***ORG***	-Funcional - Conocimiento	Docente universitario	√				√	√		
		N jerárquico	C Académico				√				
	ELE	-Funcional - Conocimiento -N jerárquico	Coordinador Administrativo y Tecnológico	√			√	√	√	√	
		-Funcional - Conocimiento	Profesores Tutores	√				√			
		-Funcional - Conocimiento -N Jerárquico	Director de carrera	√			√	√		√	
		-Funcional - Conocimiento	Diseñadores de contenido	√				√			
		-Funcional - Conocimiento	Correctores de estilo	√				√			
	RIO	-Funcional - Conocimiento	Director de proyecto de investigación	√	√	√					√
		-Funcional - Conocimiento	Integrante de Proyecto de Investigación	√				√			√
		-Funcional - Conocimiento	Docente investigador	√				√			√
Externa		-Funcional - Conocimiento	Consultor de TICs aplicadas a la educación						√	√	

se deben distinguir la Influencia, que indica el poder relativo del interesado en el proyecto y el interés, medida que se deriva de la relación entre las necesidades del interesado y los propósitos y objetivos del proyecto. En la Tabla 4 se muestra el análisis de influencia e interés.

4. Conclusión

Resulta ineludible que diferentes interesados tengan diversas visiones de la importancia y prioridad de los requerimientos y, algunas veces, dichas visiones están en conflicto. Durante el proceso, usted deberá organizar negociaciones regulares con los interesados, de forma que se alcancen compromisos. Es imposible complacer por completo a cada interesado, pero, si algunos suponen que sus visiones no se consideraron de forma adecuada, quizás intenten deliberadamente socavar el proceso de Ingeniería de Requerimientos [14].

El análisis de los *interesados* es imprescindible para llegar a una correspondencia de responsabilidades entre las partes interesadas y las partes involucradas en el proceso de enseñanza aprendizaje para maximizar las posibilidades de éxito del e-learning en las instituciones de educación superior.

En este trabajo se presentó la aplicación de un método para el análisis de *interesados* en la etapa de especificación de requerimientos en el desarrollo de un proyecto de software para e-learning en ambientes universitarios. Este análisis involucra diferentes aspectos, criterios y dimensiones que favorecen en la identificaciones de los principales roles. La aplicación de un método permite sistematizar la tarea de identificación de los *interesados* a la vez que propone criterios y sirve de guía en esta etapa del desarrollo del software. Se presentaron las distintas tareas que el método propone y su adecuación para entornos de e-learning en ambientes universitarios aplicándolas a un caso de estudio. El método utilizado, y su modificación propuesta, son adecuados para el desarrollo de herramientas para e-learning.

De esta manera se propone una forma ordenada y sistémica para identificar las necesidades, intereses y conflictos que se presentan en un entorno de e-learning de manera de colaborar en el desarrollo de herramientas adecuadas que resultan fundamentales a la hora de aplicar tecnologías a los procesos de enseñanza aprendizaje en ambientes universitarios. Así, se busca colaborar con la implementación y coordinación de cursos y carreras que se ofrecen con la modalidad de educación mediada por TIC y permitir el mantenimiento y la actualización

constante en el equipamiento necesario para atender a la evolución permanente de la modalidad.

Tabla 4. Análisis de influencia e interés

		Influencia	
		BAJA	***ALTA***
Interés	***BAJO***	CE	SA DCA CTIC
	ALTO	AG – AP- APFC-APOS TE	DPI – IPI – DI - DIR CEAD PT AP DC

5. Agradecimientos

Los autores agradecen a la Universidad Católica de Santiago del Estero Sede Rafaela (UCSE DAR) y a la Universidad Nacional del Litoral (UNL) por su financiamiento.

6. Referencias

[1] J. Gama, Z. Gomes de Olivera, E. da Silva, *"La evaluación del aprendizaje en los programas de educación a la distancia: Construcción de procesos y alcance de calidad"*, Universidade do Estado de Rio de Janeiro. Etic@net .ISSN: 1695-324X. Año II Número 4.

[2] V. Cristea, S. Trausan-Matu, O, Udrea. Sintec elearning collaborative environment. Proceedings of CSCS14, vol. 2, pp. 63-67. oan Dumitrache (ed.).

[3] Pei-Chen Sun, Ray J. Tsai, Glenn Finger, Yueh-Yang Chen, Dowming Yeh. *"What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction"* Computer and education. Elsevier, Vol 50, pp 1183-1202, 2008

[4] L. Guárdia Ortiz, A. Sangrà Morer, *"Diseño instruccional y objetos de aprendizaje; hacia un modelo para el diseño de actividades de evaluación del aprendizaje online"*, Red. Revista de educación a distancia, julio, año/vol. IV, número monográfico OIV. Universidad de Murcia.

[5] N. Wagner, K. Hassanein, M. Head, M. Who Is Responsible for E-Learning Success in Higher Education? A Stakeholders' Analysis. Educational Technology & Society, 11(3), 26-36. Retrieved February 17, 2014 from http://www.editlib.org/p/75266. 2008.

[6] R. Ellis, T. Hubble, A. Applebee, M. Peat. Perspectives of stakeholders on eLearning in science education at university. Proceedings of The 23rd Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education ASCILITE Who's Learning? Whose Technology? 2006.

[7] L. Ballejos M. Montagna, *"Method for stakeholder identification in interorganizational environments"*, Requirements Eng. vol 13 pp. 281-297. DOI 10.1007/s00766-008-OO69-1. Springer-Verlag, 2008.

[8] L. Romero, M. Gutierrez, M. Caliusco. "Análisis de interesados en el desarrollo de proyectos de software para e-learning en contextos universitarios". CISTI 2014. Sistemas y Tecnologías de la Información. Actas de la 9na Conferencia Ibérica de Sistemas y Tecnologías de la Información. Vol I, p. 149-154. Alvaro Rocha, David Fonseca, Ernest Redondo, Luis Paulo Reis, Manuel Perez Cota editores. Barcelona, España, 18 al 21 junio 2014. ISBN 978-989-98434-3-1. 2014.

[9] SCORM 2004. 3rd edition. Sharable Content Object Referente Model. Overview. Versión 1.0. 2006. Advanced Distributed Learning.

[10] C. Chenti-Belcadhi, L. Henze, N., Rafik-Braham., *"An assessment Framework for eLearning in the Semantic Web"*, Proceedings of the Twelfth GI-Workshop on Adaptation and User Modeling in interactive Systems (ABIS 04).

[11] K. Scalise, B. Gifford, "Computer-Based Assessment in E-Learning: A framework for Constructing *"Intermediate Constraint" Questions and Tasks for Technology Plataforms"*, The Journal of Technology, Learning and Assessment. A Publication of the Technology and Assessment Study Collaborative Caroline A.& Peter S. Lynch School of Education, Boston Colege.Volumen 4, número 6. ISSN 1540-2525.

[12] I. Alexander, S, Robertson, *"Understanding project sociology by modeling stakeholders"*, IEEE Software IEEE Computer Society 21(1): pp. 23–27, 2004.

[13] J. Luo, W. Li, J. Cao. Integrating Heterogeneous E-learning Systems. Telecommunications, 2006. AICT-ICIW apos;06. International Conference on Internet and Web Applications and Services/Advanced International Conference. volume , Issue , 19-25 Feb. 2006 pp: 9 – 9. 10.1109/AICT-ICIW.2006.115.

[14] I. Sommerville, *"Ingeniería de Software"*, Addison-Wesley, 9° edición. Pearson Educación, México, 2011, ISBN 978-607-32-0603-7

Estrategias de evaluación y motivación del aprendizaje para la materia Simulación

Rosana M. Portillo, Mariana Cóccola, Ana Rosa Tymoschuk.
Departamento Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional – Facultad Regional Santa Fe
Lavaise 610 - S3004EWB Santa Fe - Argentina
rportill@frsf.utn.edu.ar, marcoccola@gmail.com, atymoschuk@frsf.utn.edu.ar

Abstract

En vistas a la respuesta de los alumnos respecto al aprendizaje de la materia Simulación, los docentes de la cátedra hemos implementado mejoras para el dictado y la evaluación de la asignatura a partir del ciclo lectivo 2013 y 2014.

Junto a los indicadores tradicionales de desempeño de los alumnos como números de regularizados y de promocionados y de intercambios informales de opinión, hemos considerado una encuesta de los estudiantes al final del cursado sobre los aspectos relacionados con el desarrollo de las unidades temáticas, los temas más interesantes, la metodología didáctica, la metodología de evaluación y la principales dificultades encontradas, con el fin de reconocer las fortalezas y detectar las debilidades en el dictado y el desarrollo de la materia.

Introducción

La asignatura "Simulación" del cuarto año en la carrera de Ingeniería de Sistemas de Información de la UTN plantea enseñar y aplicar modelos y simulación de sistemas como herramientas de predicción, mediante una serie de criterios asociados a los sistemas de información y las tecnologías.

Un aspecto importante de la materia para el alumno es aplicar el enfoque sistémico para estimar cuantitativamente comportamiento de casos de estudio. Esta metodología es útil para otras materias de la carrera como Sistemas de Gestión, Inteligencia Artificial, Investigación Operativa, entre otras.

En la materia también se abordan los comportamientos dinámicos de los sistemas en forma continua o discreta. Por esa razón son necesarios conceptos de Probabilidad y Estadística, Matemática Discreta, Paradigmas de Programación, y otras, favoreciendo la integración de conocimientos por parte del estudiante.

Otros aspectos de importancia para el aprendizaje son las representaciones de los sistemas reales en los modelos, con los parámetros de entrada obtenidos a partir de registros y observaciones del mundo real y su correspondiente tratamiento estadístico, la programación del modelo en un lenguaje de propósitos generales o de un simulador, las etapas de verificación y de validación y los experimentos por computadoras, cuyas variables de salidas permiten detectar comportamientos y buscar configuraciones favorables para el desempeño del sistema. Banks [1], Kelton y col. [2].

El estudiante debe aplicar conocimientos de probabilidad, estadística y programación para simular sistemas y obtener información que debe ser correctamente interpretada. Esto se favorece con una práctica intensiva de modelos y simulaciones de una variedad de sistemas.

La planificación de la materia prevé la regularización y la aprobación mediante una serie de instrumentos que consideran la participación del alumno en forma grupal e individual en trabajos donde se aplican conceptos teóricos y prácticos.

Con el objeto de mejorar el aprendizaje de los alumnos y establecer una forma dinámica de actualización de contenidos los docentes de la cátedra hemos considerado los resultados de las evaluaciones de ciclos lectivos anteriores y las opiniones de los alumnos de cada año para implementar cambios.

Por esta razón y según Niremberg y col., los procedimientos sistemáticos de recolección, análisis e interpretación de información permiten emitir juicios valorativos fundamentados y comunicables sobre las actividades, resultados e impactos de esos proyectos o programas y formular recomendaciones para tomar decisiones que permitan ajustar la acción presente y mejorar la futura. [3]

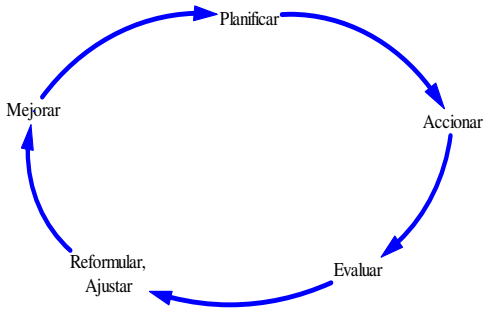


Figura 1. Proceso de evaluación

Según la Figura 1, la evaluación del aprendizaje de lo planificado permite observar aspectos para plantear mejoras en un proceso cíclico, con una orientación hacia el aprendizaje y no sólo a la comprobación del mismo.

Desde la visión sistémica el proceso de enseñanza-aprendizaje forma relaciones circulares (causalidad circular). Es decir, el alumno realiza su aprendizaje y al mismo tiempo se efectuarán reiterados procesos valorativos de enjuiciamiento y crítica que le servirán para tomar decisiones respecto a su ubicación en el proceso de estudio. En este sentido, la evaluación le permitirá tomar conciencia de los conocimientos adquiridos, por lo tanto deberá ser una actividad que promueva el aprendizaje.

El alumno dispone de instrumentos que le permiten percibir su progreso en el conocimiento y habilidades. Asimismo el docente puede hacer con ellos el seguimiento del aprendizaje. Así, las evaluaciones completas de contenidos curriculares, la realización de ejercicios y trabajos, planteados como desafíos para el desarrollo de soluciones creativas y un espíritu de superación por parte de los alumnos, las evaluaciones continuas a través de guías, ejercitaciones, pruebas de laboratorios, entre otros, con la interacción alumno-profesor y el trabajo individual y grupal y con las evaluaciones Cualitativas y Cuantitativas, permiten lograr un juicio de valor sobre el desempeño del alumno en la asignatura.

Otra cuestión importante para el aprendizaje es motivar al alumno con propuestas de ejercicios de amplia variedad para las instancias de evaluación con el fin de presentar mayores desafíos y reducir el acostumbramiento a una forma de resolución.

Estas consideraciones han sido tenidas en cuenta en las planificaciones de la materia Simulación con más intensidad a partir de 2013.

Hasta entonces la regularidad de la materia consistió en cumplir un porcentaje mínimo de asistencias a clases, la aprobación de un parcial luego de las primeras 4 unidades y de un trabajo práctico grupal de resolución de un problema dado por la cátedra sobre una temática específica, con entrega de un informe en un tiempo establecido y un coloquio individual sobre el trabajo práctico grupal en presencia del grupo de trabajo.

Se solicitaba un porcentaje mínimo de aprobación del 60 % en esas instancias y una asistencia del 75 % a las clases para regularizar.

Como resultado de esa planificación, en general se observó un alto porcentaje de alumnos que regularizaron la materia, pero mucho menor fue el número de los que se presentaban a examen final en el mismo año en que la cursaron.

Como dato ilustrativo, durante el período 2009 al 2012 los alumnos que regularizaron al final del cuatrimestre fueron entre el 65 y 80% del total. En ese mismo período los que debieron recuperar la evaluación parcial fueron entre el 20 y 35% del total y finalmente no regularizaron entre 12 y 17%.

Asimismo la materia se aprobaba y promocionaba con un examen escrito individual en alguno de los llamados en el año consistente en la modelización de un problema mediante el planteo de modelos de simulación discreta, expresados en lenguaje de simulación, modelos de sistemas continuos e interpretación de resultados.

En el año 2013 se implementó una nueva estrategia de evaluación y según el análisis e interpretación de los resultados académicos obtenidos, se implementaron cambios para mejorar el aprendizaje en la modelización y la simulación de los sistemas discretos y continuos durante el ciclo lectivo siguiente.

Dicha evaluación consistió en alcanzar la regularidad con la aprobación de evaluaciones parciales al final de cada unidad, del trabajo práctico grupal y del coloquio, además de un porcentaje de asistencias. De esta forma, se pasó de un solo parcial que incluía varias unidades a una evaluación de cada unidad.

La aprobación de cada parcial y del trabajo práctico, con su informe y coloquio, con un porcentaje mínimo de un 70% cada instancia les permitió a los alumnos presentarse al examen final y promocionar la materia con un examen práctico conceptual, en donde se presentaba un problema y se planteaban posibles soluciones, técnicas de resolución en cada etapa del problema, interpretación de resultados, aspectos teóricos relacionados con el problema, con respuestas del tipo Falso Verdadero, sin tener que plantear un modelo completo de simulación discreta, o un modelo de sistema continuo.

En caso que los alumnos aprobaran alguna de las instancias con porcentajes entre el 70 % y 60%, solo les permitió regularizar y promocionar la materia con un examen final completo como se describió anteriormente. Si los porcentajes en cualquier instancia fueron menores del 60 % se debía realizar el recuperatorio de las unidades no aprobadas.

En el ciclo lectivo 2013 hubo un total de 55 alumnos cursantes quienes en su totalidad regularizaron la materia. En la tabla 1 se observan los resultados de evaluaciones parciales correspondientes a las unidades temáticas 1, 2, 3, 4, 6, 7 y 8.

Un total de 33 alumnos aprobaron parciales con porcentajes mayores o iguales al 70% y 22 alumnos realizaron recuperatorios de una o más unidades, de los cuales 13 aprobaron con porcentajes iguales o mayores al 70% y 9 lo hicieron con menos de este último valor.

Tabla 1. Resultados evaluaciones parciales 2013

Evaluación	Cantidad de alumnos		
	Notas entre 0 -59%	Notas entre 60 – 69%	Notas entre 70% - 100%
Unidad 1	0	13	32
Unidad 2	0	2	53
Unidad 3	0	4	51
Unidad 4	2	15	38
Unidad 6	1	4	50
Unidad 7	0	9	46
Unidad 8	1	9	45

Es de destacar que desde la finalización del cursado de la asignatura en julio de 2013 hasta el llamado a examen final de julio de 2014 se presentaron a rendir el examen final un total de 44 alumnos, habiéndolo aprobado y promocionado en todos los casos. Esto indicó la efectividad de hacer una evaluación y seguimiento más frecuente, lo que motivó a los alumnos a buscar el 70 % mínimo para la promoción de la materia con un examen práctico conceptual.

En virtud de estos resultados y teniendo en cuenta por un lado los juicios de valor que manifestaron los alumnos respecto a la metodología, y por otro, la articulación horizontal con las demás materias del cuarto nivel respecto a fechas de parciales y a la cantidad de evaluaciones, numerosas aunque sencillas, se decidió mejorar la estrategia de evaluación implementada en 2013 y aplicarla en el ciclo lectivo 2014.

Además de los resultados académicos y para tener datos cualitativos por parte de los alumnos, se realizó una encuesta al finalizar el cursado para conocer sus opiniones respecto a contenidos de la materia, el nivel de los conceptos desarrollados, la metodología de evaluación, los temas que resultaron más interesantes y cuáles fueron sus mayores dificultades durante el cursado.

Elementos de trabajo y Metodología

Desde el punto de vista estratégico y dados los momentos, las evaluaciones se consideran en la categoría continua (evaluación del proceso) y con función formativa mediante el desarrollo de guías prácticas. Y también reviste el carácter de final o sumativa mediante pruebas objetivas donde se realiza la valoración de todo el recorrido que ha seguido el alumno.

Los cambios de la planificación 2014 respecto a la de 2013 han incluido el requisito de la regularidad con la aprobación de cuatro parciales, cada uno incluyendo dos unidades, del trabajo práctico grupal con la resolución mediante simulación de un sistema de cierta complejidad y del coloquio, además del porcentaje de asistencia que se conserva como en el año anterior.

El porcentaje mínimo de aprobación de las instancias de evaluación para la regularidad es 50 y si se alcanza un 70 % o más en todos los casos, el examen final es del tipo práctico conceptual como se describió anteriormente. El criterio de considerar un 50 % como mínimo para regularizar la materia es porque el alumno realiza varias instancias de evaluación a partir del ciclo lectivo 2013, siendo antes un solo parcial que incluía 4 unidades, con mayor inclusión de temas, y el trabajo práctico grupal. En todos los casos se toma un recuperatorio de las unidades no aprobadas.

Los criterios de evaluación de la cátedra se presentan en el primer día de clases, informando a los alumnos las condiciones para la regularidad y la aprobación y promoción de la materia, así como las instancias de evaluación y un cronograma tentativo de las actividades.

Explicitar los criterios que se van a tener en cuenta para evaluar alude a cuál será la base de referencia que permitirá afirmar si un alumno alcanzó o no los objetivos formativos y cuáles son los requisitos mínimos para decidir su aprobación. Avolio de Cols y Lacolutti [4]

Las clases de consulta se dan durante todo el año, como un espacio para un mayor acercamiento con el alumno, con horarios establecidos, priorizando las fechas previas a las entregas de trabajos, parciales y exámenes.

El empleo del campus virtual es otra herramienta de mejoras ya que facilita la comunicación con los alumnos, mediante el uso de foros y de la consulta a través de mensajes o envío de emails, incluso para informar cambios o novedades. Además se publica el material de estudio y de prácticas necesarios para el estudio.

El uso de las técnicas informales, que forman parte de la evaluación formativa, como son la observación de las actividades realizadas por los alumnos y las preguntas formuladas por el docente durante la clase, permite en cada clase la percepción de acciones y reacciones de los alumnos cuando están trabajando en grupo, con especial atención a indicadores de expresiones paralingüísticas como los gestos de atención, de sorpresa, de gusto, disgusto, de aburrimiento, y la forma en que expresan sus dudas y como se comunican, discuten y debaten con sus pares del grupo. Estos elementos suman para realizar evaluaciones de seguimiento o cualitativas.

Por otro lado, las preguntas planteadas en las clases se formulan con el fin de obtener el nivel de comprensión de los alumnos en un tema.

Para la asignatura la observación se sistematiza mediante registros anecdóticos y diarios de clases, a completar en el transcurso del proceso de enseñanza-aprendizaje, tanto por cada alumno como por grupo de trabajo.

Es importante que los docentes permitan la expresión de las dudas de los alumnos mediantes preguntas y

charlas para detectar el avance de la comprensión correcta de los temas, además de promover el pensamiento. Esta instancia de intercambios requiere de la formulación de preguntas para favorecer la reflexión por parte de estudiantes y docentes, siendo en este último caso importante para brindar nuevas explicaciones, aclaraciones, repeticiones, reformulaciones.

En coincidencia con los autores F. Díaz y A. Barriga [5], es importante dar un tiempo a los alumnos para pensar las respuestas, considerar una muestra amplia de estudiantes y no trabajar solo con los alumnos voluntarios quienes por lo general se ubican en los primeros lugares de la clase, delante del profesor.

También se utilizan las Técnicas Semi Formales como parte de la evaluación formativa, consistentes en trabajos y ejercicios que los alumnos realizan en clase y fuera de las mismas. Las actividades se incluyen dentro del campus virtual de UTN-FRSF.

En clase, se realizan las guías prácticas, en donde además de alumnos de la carrera de Ingeniería en Sistemas de Información también los hay de la carrera Ingeniería Industrial para quienes la materia es electiva. Por esta razón también se incorporan ejercicios y casos de estudio aplicables a la gestión, temática principal de la mencionada carrera, como logística, planificación de la distribución en planta, instalaciones industriales, mecánica de fluidos, planificación y control de producción, transporte, entre otros.

Respecto a la práctica de la materia hay una guía por cada unidad de teoría, que se resuelven en grupo y durante el desarrollo de las clases de práctica en laboratorio o en aula. Cada guía práctica se revisa en cada ciclo lectivo a los fines de actualizar los ejercicios que contienen.

Para el desarrollo de la guía práctica correspondiente a la unidad 5 (Traslación del modelo a la computadora) se utiliza la herramienta Software GPSS (General Purpose System Simulation) [6]. Es un lenguaje de propósito general orientado a la simulación de sistemas discretos cuya estructura de funcionamiento está basada en el enfoque de Interacción del Proceso.

En el desarrollo de modelos continuos se utiliza el simulador VensimPle [7], es una herramienta visual de modelización que permite conceptualizar, documentar, simular, analizar y optimizar modelos de dinámica de sistemas. Provee una forma simple y flexible de construir modelos de simulación, sean diagramas causales o diagramas de Forrester (stock y flujo).

Cabe mencionar que la cátedra analiza y prueba la implementación de simuladores más avanzados o actuales, con interfaz gráfica para facilitar la modelización. En particular se está investigando el software de simulación de evento discreto SImulation Modeling framework base don Intelligent Objects, Simio [8], orientado a objeto que ofrece la ventaja de permitir crear un modelo y luego reutilizar el trabajo inicial para adaptarlo a futuros modelos del mismo tipo, a diferencia de simuladores orientados a proceso.

Para el desarrollo del Trabajo Práctico Grupal (TPG) se solicita la organización de grupos de no más de 4 alumnos para asignarles en la primera semana de cursado un caso de estudio de sistemas discretos.

Se solicita la entrega de 2 informes del TPG, según un cronograma relacionado con el dictado de las unidades, de manera que los alumnos vayan realizando el trabajo en la medida del avance de conocimientos en el procedimiento de la simulación.

La primera entrega del TPG consiste en un informe que describe el sistema en estudio, según los objetivos planteados, la presentación de una tabla con las entidades, atributos, estados y eventos afines a los objetivos buscados. Como el trabajo incluye un documento con datos “crudos” del sistema, el grupo debe realizar el análisis estadístico, con la construcción de la tabla de frecuencias y la aplicación de pruebas de bondad de ajuste para obtener las distribuciones de probabilidad que parametrizan al modelo.

En la segunda entrega se presenta el modelo del sistema en un software de simulación de sistemas discretos, para obtener los resultados según los objetivos planteados. Se requiere un minucioso análisis del reporte estándar del simulador, indicando e interpretando las salidas de cada una de las variables de interés propuestas y de otras variables del reporte. Es importante en esta etapa identificar y enumerar variables de interés en su evolución en el tiempo y para cada una de ellas mostrar gráficos de evolución e interpretación del comportamiento mostrado.

Posteriormente se realiza el diseño y análisis de experimentos de simulación, donde se deben identificar y enumerar todos los factores controlables, con su justificación, seleccionar aquellos que a priori pueden ser más influyentes en las variables de respuesta en estudio, indicar para cada uno de ellos la cantidad de niveles propuestos y sus valores, proponer diferentes tipos de diseños de experimentos y seleccionar uno de ellos para ejecutarlo en el software para obtener el análisis correspondiente. Una etapa de mucha importancia es la interpretación de los resultados obtenidos y la influencia de los factores propuestos y sus interacciones para las conclusiones y recomendaciones finales.

Estas dos entregas completan la resolución del trabajo práctico grupal solicitado. El informe que presentan los alumnos cumple con los requerimientos solicitados en la cátedra. De esta forma se orienta al grupo para analizar y detallar los aspectos de importancia en el proceso de modelado. Todos los enunciados de TPG están asociados a la materia en la plataforma del campus virtual, con

fechas límites de presentación de los informes correspondientes. Se realiza la devolución una vez que los trabajos prácticos son revisados y calificados por los docentes, a través del campus virtual para realizar las correcciones o analizar las observaciones. Se debe informar a cada grupo por separado todas las correcciones, observaciones y errores si los hubiera y luego el grupo debe corregirlos bajo la supervisión del docente.

La nota obtenida por el grupo en este TPG es considerada en la nota final individual de cada alumno integrante.

A fin de integrar la realización de las guías prácticas y el TPG se planificaron las clases de práctica, dedicando parte de la clase para resolver los ejercicios de la guías prácticas y parte al TPG, logrando participación de la mayor cantidad de alumnos en la clase de práctica en los grupos de cada TPG, y esto ha permitido identificar a aquellos alumnos o grupos a quienes tienen más dificultades para resolver los ejercicios propuestos.

Durante el desarrollo de la cátedra y especialmente el TPG, los docentes de la cátedra coordinan horarios de consulta durante todo el cursado.

Las Guías Prácticas y el TPG permiten a los alumnos realizar un análisis reflexivo, y de esta forma se los ayuda a pensar y aprender sobre los conceptos o procesos que se enseñan.

Entre las Técnicas Formales que se utilizan en la materia para la evaluación se cuentan con:

- Exámenes Parciales, como evaluación formativa.
- Coloquio Grupal con preguntas individuales, considerada como evaluación formativa
- Examen Final, considerada como evaluación final o sumativa

La aprobación de los exámenes parciales es condición necesaria para poder obtener la regularidad en la materia y su nota se tiene en cuenta en el examen final, durante un ciclo lectivo.

Luego de cada parcial se realiza la devolución en clase de teoría, para observar los errores.

El Coloquio Grupal con preguntas individuales se realiza al finalizar el cuatrimestre. En esta actividad cada grupo prepara una exposición sobre el caso de estudio del TPG y expone su análisis y conclusiones al resto del curso. Dicha exposición grupal tiene una duración de no más de 20 minutos. Durante ese lapso de tiempo el grupo expone y posteriormente los docentes realizan preguntas a cada integrante del grupo. La aprobación del coloquio es condición necesaria para poder obtener la regularidad en la materia. Es una muy buena herramienta de prueba oral que permitirá al alumno expresarse frente a sus compañeros y docentes. Permiten comprobar el dominio de conocimiento o habilidad intelectual de cada alumno del grupo

El Examen Final Práctico conceptual lo realizan aquellos alumnos que aprobaron con más del 70% las evaluaciones parciales, TPG y coloquio. Se considera la situación de cada alumno en el proceso de evaluación.

El examen final tiene una duración de no más de 60 minutos, es escrito (instrumento de evaluación: papel y lápiz) y se propuso que sea mediante pruebas objetivas que incluyan items Verdadero/Falso, ítems de correspondencia, ítems de selección simple y múltiple, con enunciados que deben ser claros y sin ambigüedades.

Resultados

Al finalizar el cursado se realizó una encuesta sobre la materia y su desarrollo en el ciclo lectivo, en la que participaron 40 alumnos.

A continuación se exponen los resultados obtenidos en cada una de las evaluaciones parciales de unidades de la materia, realizadas en horarios de clases de práctica o teoría y con una duración menor a las 2 horas cátedra.

También se muestran los resultados de la encuesta que realizó la cátedra.

Resultados de las evaluaciones parciales.

Cabe detallar que en el ciclo lectivo 2014 hubo un total de 48 alumnos quienes en su totalidad regularizaron la materia y 26 alumnos realizaron recuperatorios.

Los resultados finales del cuatrimestre, luego de la realización de recuperatorios de las unidades, un total de 41 alumnos regularizaron teniendo un 70% o más en sus instancias de evaluación, Trabajo Práctico Grupal y coloquio, por lo tanto, accedieron a un examen final práctico-conceptual con la modalidad múltiple choice.

Los demás alumnos sólo regularizaron la materia, teniendo menos del 70% en sus instancias de evaluación, Trabajo Práctico Grupal y coloquio.

En las tablas siguientes figuran las notas expresadas en rangos de porcentajes obtenidos y correspondiente frecuencia, en los diferentes parciales. Esto brinda una primera vista sobre el nivel alcanzado de los alumnos en cada etapa del proceso. Además se presentan los resultados en forma gráfica para su mejor interpretación.

Mediante esta tabla o gráfica se concluye que la cantidad de alumnos que aprobó las evaluaciones parciales con porcentajes mayores o iguales al 70 % es superior a quienes aprobaron con menor porcentaje.

Tabla 2 . Resultados evaluaciones parciales 2014

Evaluación	Cantidad de alumnos		
	Notas entre 0 - 49%	Notas entre 50 – 69%	Notas entre 70% - 100%
Unidad 1	0	3	45
Unidad 2	0	0	48
Unidad 3	0	5	43
Unidad 4	0	4	44
Unidad 6	0	3	45
Unidad 7	0	4	44
Unidad 8	0	5	43

Resultados de las encuestas.

Por primera vez la cátedra realizó una encuesta a los alumnos, con el propósito de conocer la opinión de los alumnos relativa a los contenidos de la materia y el nivel de los conceptos desarrollados, a la forma en que se abordaron los contenidos (metodología didáctica), a la metodología de evaluación, el material de apoyo y bibliografía, y para saber cuáles fueron los temas que les resultaron más interesantes, las mayores dificultades que tuvieron durante el cursado y sobre el nivel de satisfacción de sus expectativas respecto a "Simulación".

En las tablas siguientes figuran los resultados obtenidos, expresados en porcentajes, para los 40 alumnos encuestados.

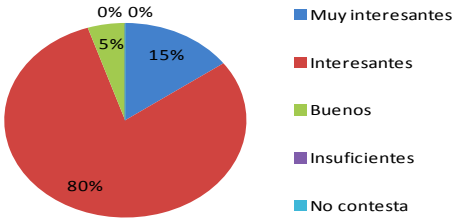


Figura 2. Contenidos de la materia y el nivel de los conceptos desarrollados

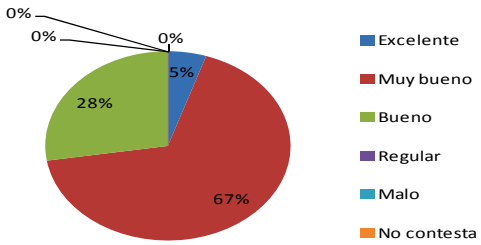


Figura 3. Forma en que se abordaron los contenidos (metodología didáctica).

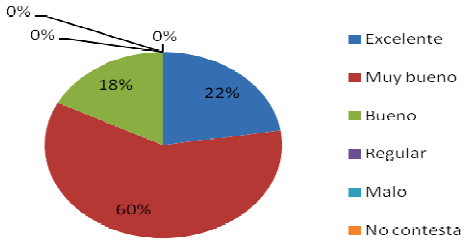


Figura 4. Metodología de evaluación

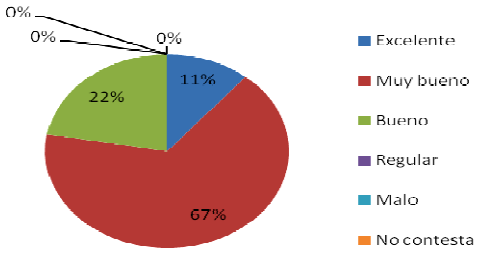


Figura 5. Material de apoyo y bibliografía.

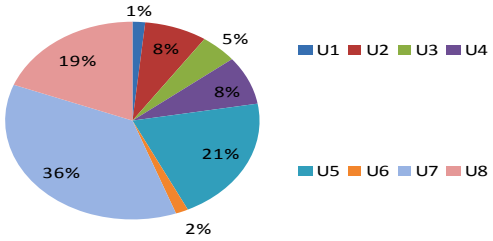


Figura 6. Temas o unidades temáticas que le resultaron más interesantes.

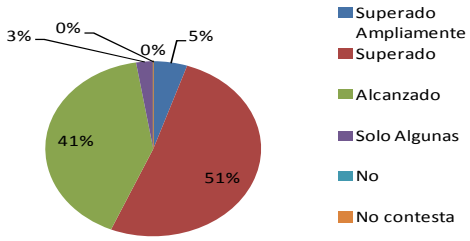


Figura 7. Satisfacción de expectativas previas

Respecto a la consulta en la encuesta sobre los temas que les hubiera gustado profundizar más y/o adquirir otros conocimientos, se obtuvo un alto porcentaje de alumnos que manifestaron el aprendizaje de otro software de simulación, seguido por el diseño de experimentos y modelado continuo.

Además casi en su totalidad los alumnos manifestaron que la mayor dificultad en el cursado de la materia fue el

tiempo que debieron dedicar al estudio y cumplimiento de los requisitos de regularidad y promoción.

Discusión

Según los autores Lambert y col. [9], el aprendizaje es un proceso de construcción en el cual lo que se aprende tiene sentido cuando el aprendiz está activamente implicado en la creación de su propio conocimiento y comprensión a través de la conexión de lo que aprende con el conocimiento y la experiencia anteriores.

Considerando estos conceptos y coincidiendo con Niremberg y col.[3] en que la evaluación implica analizar qué es lo que se está haciendo (o se hizo), de qué modo, si ello se orienta según la direccionalidad deseable, cuáles escollos y cuáles facilidades u oportunidades se presentan y cuáles logros se han obtenido; a eso se hace referencia cuando se dice que la evaluación es reflexión sobre la acción, se ha propuesto mejorar la metodología de evaluación propuesta en el año 2013.

El principal objetivo ha sido que el alumno pueda construir un conocimiento reflexivo, significativo y a promover un pensamiento crítico, acompañados por los docentes como guía en el proceso de aprendizaje.

Mejoras en la metodología de enseñanza-aprendizaje.

Hasta el ciclo lectivo 2012 se evaluaba a los alumnos 4 momentos, siendo el primero mediante un parcial que incluía las primeras cuatro unidades, aproximadamente en la sexta semana del inicio del cursado. El parcial consistía de la resolución de ejercicios de práctica y constituía una evaluación esporádica, con agrupamiento de varias unidades temáticas.

La segunda evaluación consistía en la presentación del informe del TPG, con el desarrollo de dos tipos de modelos, uno discreto y otro continuo. La entrega del informe se desarrollaba en dos etapas para el caso discreto y en una etapa para el caso continuo.

La tercera evaluación era el coloquio grupal sobre el TPG. La aprobación de todas estas evaluaciones era condición necesaria para poder obtener la regularidad en la materia.

Finalmente la aprobación de la materia se hacía mediante el examen final con modelización de un sistema discreto, conceptos e interpretación de diseño de experimentos de simulación y modelización de un sistema continuo. En esta evaluación final no se incluían los contenidos conceptuales de teoría, dónde es necesario comprobar una comprensión de los mismos por parte del alumno.

En ciclo lectivo 2013 se presentó una estrategia de aprendizaje y de evaluación continua y sumativa, en diversos momentos del cursado. Se realizaron parciales al final de cada unidad, con una devolución posterior en clase de teoría con el fin de ver los errores y discutir los resultados. Se continuó con el desarrollo de un Trabajo Práctico Grupal (TPG) de un modelo discreto, de mayor complejidad que los ejercicios de las guías prácticas. Se solicitaron cuatro entregas de informes para hacer un seguimiento de la comprensión de los temas y además, un coloquio grupal con preguntas individuales sobre los informes.

Se hizo la consideración del 70 % de aprobación en todas las instancias para promocionar la materia con un examen final práctico conceptual con la totalidad de los conocimientos de la materia.

La estrategia antes descripta permitió mayor interacción docente-alumno, mayor aprendizaje progresivo por parte del alumno y sostener la motivación durante todo el cursado. En virtud de los resultados obtenidos, teniendo presente el concepto de evaluación y además la opinión de los alumnos respecto a la metodología se revisó la metodología para hacer ajustes y mejorar la estrategia.

Los cambios para mejorar la estrategia de evaluación de aprendizaje, continua y sumativa en diversos momentos del cursado, en el ciclo lectivo 2014 fueron:

- La realización de parciales cada 2 unidades, con un total de 4 momentos de evaluación sobre los contenidos desarrollados en las unidades, excepto de la unidad 5 que se incluye en el TPG.
- La devolución y discusión de los resultados de las evaluaciones en clases para observar los errores y discutir los resultados.
- El desarrollo de un Trabajo Práctico Grupal (TPG) de un modelo discreto con 2 entregas para optimizar tiempos de realización y correcciones.
- La defensa del TPG en un coloquio con el grupo con preguntas individuales.
- La consideración del 70 % de aprobación en todas las instancias con un examen final práctico conceptual globalizador.
- La consideración del 50 % como porcentaje mínimo de aprobación de las evaluaciones parciales por que incrementaron en número respecto a ciclos anteriores al 2013 y en mayor contenido a evaluar.

Estos cambios se hicieron para:

- Mejorar el proceso de construcción del conocimiento.
- Fomentar y entrenar de forma progresiva la planificación, regulación y evaluación de los propios aprendizajes
- Incrementar la interacción docente alumno
- Aumentar el aprendizaje progresivo

- Mantener la motivación y atención durante todo el cursado

Conclusión

En la metodología planteada por la cátedra se desarrolló una estrategia para desarrollar las capacidades de los alumnos dando lugar a habilidades a través de la práctica y mediante diferentes procedimientos.

Esta metodología permite comprender el proceso de simulación de sistemas continuos o sistemas discretos, desde el modelado hasta la simulación.

Se utilizó para el planteo del modelo y su simulación un software específico al enfoque discreto, GPSS, en su versión estudiantil. A partir del año 2014 se ha comenzado a investigar otro simulador, Simio, basado en objetos para emplear herramientas actualizadas e incorporar en el próximo ciclo lectivo. De esta forma se pretende satisfacer una de las propuestas planteadas por los alumnos en las encuestas sobre el aprendizaje de un software de simulación con soporte gráfico.

Para los modelos de sistemas según el enfoque continuo se construyen los diagramas causales, los diagramas estructurales o de Forrester junto con las ecuaciones correspondientes, para poder hacer la simulación en el simulador VensimPLE, y así evaluar comportamientos y tomar las decisiones según los objetivos planteados para el sistema en estudio.

Para obtener capacidad analítica y práctica en la resolución de problemas se desarrollaron tanto en las horas de práctica como en teoría diferentes modelos utilizando el software que les permitió evaluar la evolución de diferentes variables involucradas, realizar posteriormente las modificaciones indicadas en el enunciado e interpretar las nuevas salidas.

Los contenidos antes descriptos fueron evaluados en diferentes instancias y en forma continua, brindando a los alumnos herramientas para mejorar el aprendizaje durante el cuatrimestre y para promocionar la materia en un tiempo no lejanos al final del cursado, tratando de sostener la motivación.

Se lograron resultados satisfactorios, principalmente dados por la cantidad de alumnos que pudieron aprobar y promocionar la materia en un tiempo no lejano al final del cursado y por las opiniones obtenidas en las encuestas en cuanto a la metodología de enseñanza-aprendizaje.

Además los docentes pudimos realizar un seguimiento de la evolución del aprendizaje de los alumnos, pudiendo hacer los cambios necesarios oportunamente. A pesar que esta estrategia es exigente e intensiva en la elaboración de parciales y ejercicios y TPG y en sus correcciones durante el cuatrimestre, se ha incrementado la relación docente-alumno y mejorado el interés por la materia, ayudando a avanzar en la carrera.

Referencias

[1] Banks Jerry, Carson II John S., Nelson Barry. *Discret-Event System Simulation.* Prentice-Hall. Quinta edición, 2009.
[2] Kelton D., Law A. *Simulation Modeling and Analysis*. McGraw-Hill. Cuarta edición, 2006.
[3] Nirenberg, Olga, Brawerman, Josette y Ruiz, Violeta. *Evaluar para la Transformación: innovaciones en la evaluación de proyectos y programas sociales*. Colección Tramas Sociales. Volumen 8. Ed. Paidós. Bs. As. Argentina, 2000.
[4] Susana Avolio de Cols y María Dolores Lacolutti . *Enseñar y evaluar en formación por competencias laborales: Orientaciones conceptuales y metodológicas*. Coordinado por Ana María Catalano. Primera Edición, Buenos Aires, Banco Interamericano de Desarrollo,Argentina, 2006.
[5] Díaz, F. Y Barriga, A. *Estrategias Docentes para un Aprendizaje Significativo: una interpretación constructivista*. México. McGraw Hill, 2002.
[6] GPSS World Student Version, 2010 http://www.minutemansoftware.com
[7] Ventana Systems, Inc., The Ventana Simulation Enviroment. Vensim-PLE for Windows Version 6.0, 2012. http://www.vensim.com
[8] Simio LLC. Simio version 4.68.8352, 2012, http://www.simio.com
[9] Lambert, N. y McCombs, B. *How students learn, reforming schools through learned centered education* (3ª ed., 2000). Washington: American Psychological Association.

Entornos virtuales de aprendizaje colaborativos aplicados a Sistemas y Organizaciones bajo el marco de trabajo de la Metodología Scrum

Susana Rey[1] , Silvia Lanza Castelli [2]

Departamento de Sistemas de Información - Facultad Regional Córdoba –
Universidad Tecnológica Nacional
Córdoba, Argentina (5000)
srey@sistemas.frc.utn.edu.ar[1], slcastell@sistemas.frc.utn.edu.ar [2]

Resumen

En la dinámica de la sociedad actual tanto en lo profesional como organizacional se hacen uso de modelos de intercambio colaborativo que se lleva a cabo mediante la interrelación de los miembros de un grupo haciendo uso de herramientas colaborativas y Tics. En particular, en nuestra disciplina de desarrollo del software, el intercambio de los integrantes de un grupo de proyecto se hace complejo y lleva un tiempo importante de comprensión de las herramientas colaborativas y Tics a utilizar dentro de una determinada metodología.

Con el fin de lograr aproximar estas dinámicas profesionales a los alumnos del primer año de la materia Sistemas y Organización de la carrera de Ingeniería de Sistemas, hemos propuesto un conjunto de actividades para una experiencia pedagógica tomando de referencia buenas prácticas organizadas en un patrón pedagógico. Como marco de trabajo dinámico tomamos los aportes de las metodologías ágiles, en particular Scrum, ésta propuesta forma parte de una de una investigación que venimos desarrollando y fue implementada utilizando los recursos de la plataforma Open Knowledge (Moodle).

Palabras Claves: ENTORNOS VIRTUALES DE APRENDIZAJE COLABORATIVO, SCRUM, PATRÓN PEDAGÓGICO, RECURSOS DE MOODLE, TICs

1. Introduction

Actualmente los escenarios de formación tanto en el ámbito académico como organizacional complementan las instancias de aprendizajes presenciales, con escenarios virtuales colaborativos a través del uso de las Tics y de herramientas colaborativas.

La demanda de profesionales dentro de la Ingeniería de Sistemas, en particular en lo relacionado a proyectos de desarrollo de software con requerimientos ágiles y cambiantes fomenta que se incremente la demanda de recursos humanos con una preparación competente y con una ágil reacción al cambio. Para tal fin formar a nuestro alumnos, futuros profesionales, en este cambio de actitudes y aptitudes nos llama al desafío de proponer un conjunto de actividades en un entorno virtual colaborativo como parte de nuestro proyecto de investigación que permita lograr mayor abstracción y cambios sustanciales al pensamiento y permita crear una experiencia didáctica en un escenario pedagógico bajo la filosofía de los conceptos claves de las metodologías ágiles, en particular Scrum [1].

A lo largo de este trabajo presentaremos la experiencia desarrollada en un entorno virtual de ambiente colaborativo (EVAC) donde los participantes están en colaboración con sus pares con el fin de producir conocimiento de valor y con el fin de crecer como seres humanos, siempre en interacciones y mediadas por Sistemas y Tecnologías de la Información y la Comunicación (S/TIC o Sistemas y TIC).

Además consideramos tomar conceptos de las metodologías ágiles ya que son aplicables a ambientes donde los requerimientos son dinámicos y cambiantes y permiten mejorar los procesos productivos.

Scrum permite obtener los resultados de manera incremental estableciendo períodos cortos de trabajo, motivando el equipo de trabajo y tener una ágil reacción al cambio.

Comprendemos que Scrum ha sido considerado para obtener como producto el desarrollo de software y que aplicarlo a un producto donde no precisamente el resultado final es software tendremos un desafío interesante y para ello tomaremos algunos artefactos para poder definir el framework de trabajo. Al pensar en un marco de trabajo conocíamos que no existe la metodología ideal y que cada una en particular tiene puntos fuertes y débiles siendo consciente de los riesgos que pueden suceder tomar de referencia pura los

principios de Scrum en un ambiente distinto al que originalmente se destina.*[2]*

Las preguntas que guían ésta investigación se refieren a:

¿Qué mejoras se pueden obtener en los procesos educativos aplicando Scrum bajo entornos colaborativos virtuales apoyado por Tics en la cátedra de Sistemas y Organización? ¿Será posible que el alumno pueda comprender y conocer las particularidades de la temática a tratar antes de la clase presencial? ¿Será posible aplicar en la dinámica del aprendizaje los beneficios de la metodología Scrum?

¿Este nuevo modelo de aprendizaje generará un cambio de actitudes y aptitudes en los alumnos permitiendo una mayor abstracción y sustanciales aporte al pensamiento antes de la clase presencial? ¿ El uso de las TIC podrán ayudar a proporcionar soporte a la construcción del conocimiento de manera colaborativa? ¿El aprendizaje grupal colaborativo mediado por Tics contribuirá a generar sinergia e innovación a través de la complementariedad y diversidad de habilidades?

2. Líneas de investigación y desarrollo de la experiencia

El estudio toma como unidad de análisis la cátedra de Sistemas y Organizaciones de primer año de la carrera Ingeniería en Sistemas de Información de la Facultad Regional Córdoba, pretendiendo identificar en las instancias prácticas de laboratorio los beneficios y desafíos de esta experiencia pedagógica de aprendizaje colaborativo en la que se hace necesario el uso de Tics y proponer una dinámica bajo el marco de trabajo de la metodología Scrum. Dichas prácticas serán implementadas en plataforma Moodle bajo el seguimiento del cuerpo docente en el laboratorio y a distancia.

2.1 Entornos Virtuales de aprendizaje colaborativo y Tics

Según Barrera [3] el trabajo colaborativo es "*un patrón de relación entre diversas personas en una organización en el que prevalece la interacción y el compartir para la consecución de un propósito común*.". Dicho concepto permite dilucidar la necesidad de compartir el conocimiento entre los miembros de un grupo para lograr un aprendizaje en conjunto. [4], que permita enriquecer la experiencia sobre una determinada situación o problema que se identifica y se desea desarrollar. Para alcanzar un trabajo en conjunto con beneficios positivos, se deben definir las metas que se intentan conseguir y se requiere trabajar de manera colaborativa sobre la solución y el modo para alcanzarla.

Hay que tener en cuenta que con la convergencia de las tecnologías de la información y la comunicación (TIC) y el surgimiento de la llamada sociedad de la "información" o "del conocimiento" (SI), el conocimiento y la información se han convertido en un factor determinante de nuestra vida económica, social y cultural. Por lo tanto, es importante considerar que a través de esta metodología se está generando conocimiento aplicable que opera sobre productos software, cuyo mercado está en constante cambio y expansión mejorando de esta manera la relación calidad / costo. [5]

Al interactuar entre pares se logra una inserción más sencilla y un diálogo natural, debido a que los hábitos de cooperación se encuentran incorporados en cada individuo. [6]

El aprendizaje colaborativo permite que un grupo de personas pueda colaborar, interactuar, negociar, crear un punto de vista diferente de las cosas y compartir información con el fin de producir conocimiento de manera coordinada y coherente. Es otro de los postulados constructivistas que parte de concebir a la educación como proceso de socio construcción que permite conocer las diferentes perspectivas para abordar un determinado problema, desarrollar tolerancia en torno a la diversidad y pericia para reelaborar una alternativa de solución conjunta.

Los entornos de aprendizaje constructivista se definen como «un lugar donde los alumnos deben trabajar juntos, ayudándose unos a otros, usando una variedad de instrumentos y recursos informativos que permitan la búsqueda de los objetivos de aprendizaje y actividades para la solución de problemas»[7].

Se tomará como eje de investigación la identificación de los beneficios y desafíos del aprendizaje virtual colaborativo apoyado por Tics tomando de referencia un patrón pedagógico siguiendo las prácticas propias de Scrum que pone foco en el seguimiento de avance permanente de los grupos y permite una retroalimentación constante a través de ciclos cortos de trabajo. De esta manera cada alumno asume su propio ritmo y potencialidades, impregnando la actividad de autonomía, pero cada uno comprende la necesidad de aportar lo mejor de sí al grupo para lograr un resultado sinérgico, al que sería más complejo acceder de manera individual; se logra así una relación de interdependencia que favorecer los procesos individuales de crecimiento y desarrollo, las relaciones interpersonales y la productividad.

Se sumarán los aportes de las Tics donde aparecen nuevos retos que es necesario afrontar y no que suponen por si mismas una garantía de cambio

positivo sino una adecuación oportuna a cada situación. Teniendo en cuenta que los involucrados tienen una posibilidad de acceso constante a las nuevas tecnologías tanto en lo particular como dentro de la universidad. En nuestra experiencia si bien se han incorporado herramientas para el intercambio sincrónico como chat y videoconferencia y otras asincrónicas no se ha logrado la coordinación adecuada de los tiempos de todos los participantes siendo necesario incorporar una dinámica que eleve el nivel de productividad.

"El aprendizaje colaborativo implica una forma de enseñanza muy útil para que los estudiantes y los profesores trabajen conjuntamente para agilizar la formación de cualquier materia. Si a este método le incorporamos los avances de la tecnología, sobre todos de las tecnologías de la información y la comunicación (TICS), incrementamos el aprendizaje y lo incentivamos con el uso de Internet, los chats, el e-mail…" [8]. Es importante considerar que el trabajo colaborativo cambiaría radicalmente la forma de interactuar de los alumnos con sus pares y de los docentes moderando mediante las TIC; pero también debe considerarse el impacto que esto causará.

2.2 Scrum

En el año 2001, Ken Schwaber y Mike Beedle. [Schwaber, 2001] describieron la metodología ágil en el libro Agile Software Development with Scrum. Tomando de allí la definición de Scrum " como un modelo de referencia que definen un conjunto de prácticas y roles, y que pueden tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto".

Hemos decidido tomar para estas prácticas Scrum por ser una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo y por sus raíces teóricas basadas en la auto organización. Más bien delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles. [9]

Al principio del proyecto se definió el "Product Backlog", que contiene todos los requerimientos funcionales y no funcionales que debía contener el caso práctico a desarrollar. Los mismos estarán especificados de acuerdo a las convenciones de la organización ya sea mediante: features, casos de uso, consignas y/o tareas, etc. El Product Backlog se definió durante reuniones de planeamiento con los stakeholders. A partir de ahí se definieron las iteraciones, conocidas como Sprint en las que se irá evolucionando la aplicación de las prácticas diseñadas para el caso de estudio. Cada Sprint produjo su propio "Sprint Backlog" como subconjunto del Product Backlog con los requerimientos a ser construidos en el Sprint correspondiente. La duración recomendada del Sprint es de un mes.

Se identificó el Scrum Master que equivale al Líder de Proyecto quien llevó a cabo la gestión de la iteración, convocando diariamente al Scrum Daily Meeting que representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan. De esta práctica se hicieron prácticas apoyadas por el uso de Tics. Al final de cada Sprint, se realizó un Sprint Review para evaluar los artefactos construidos y comentar el planeamiento del próximo Sprint.

3. Diseño de la investigación

3.1 Planteo del problema

Debido a la necesidad de fomentar un aprendizaje que genere un cambio de actitudes y aptitudes en los alumnos permitiendo una mayor abstracción y sustanciales aporte al pensamiento y contribuir a generar sinergia en el aprendizaje grupal colaborativo mediado por Tics nos propusimos a realizar estar prácticas pedagógicas bajo el marco de una metodología que permita generar una dinámica y un seguimiento en todas las etapas conociendo las limitaciones del contexto que es una práctica de aprendizaje.

Las tareas entre los grupos de trabajo se hizo compleja debido a la escasez de herramientas colaborativas que ayuden en la gestión y control de las prácticas y a tener que decidir sobre los recursos adecuados que la plataforma de Moodle ofrecía para cada situación de aprendizaje sumado a la preparación de los alumnos en la comprensión de las consignas y el logro del objetivo de cada caso para desarrollarlos en tiempo y forma.

3.2 Solución

Nuestra propuesta consistió en diseñar instancias teórica-práctica utilizando los recursos de la plataforma de Moodle y herramientas TICs que permitió crear procesos de reflexión, análisis y promoción de nuevas habilidades y puntos de vistas en los participantes.

Tomando las ideas expuestas por Piaget [10] quien argumenta que "Solo en un ambiente de métodos activos el alumno alcanza su pleno rendimiento". De esta manera el equipo docente pudo propiciar un ambiente dinámico, en el cual incorporaron técnicas colaborativas y tecnológicas que incentiven y ayuden generar nuevos conocimientos.

Para generar un marco de trabajo coordinado y establecer objetivos de las experiencias pedagógicas se decidió buscar una metodología que se ajuste a los objetivos, que permita integrar las distintas fases del ciclo de desarrollo logrando trazabilidad que permita referirse a otras fases del proyecto y fusionarlo con las fases previas.

Para esto se decidió tomar Scrum independiente de la tecnología con que se contaba con el objetivo de ajustar de manera iterativa y lograr en cada etapa los objetivos planteados incrementando los logros que permitan la posibilidad de extrapolar el modelo a situaciones similares. A través de las actividades que el equipo docente impartió, los alumnos modificaron la forma de percibir el conocimiento que permitieron tomar decisiones más acertadas y los conceptos fueron comprendidos con menor dificultad y mayor interés, estos datos obtenidos de una encuesta realizada a los alumnos al final de la experiencia.

3.3 Desarrollo de la Solución

3.3.1 Contexto de aplicación

La experiencia se realizó con estudiantes universitarios de 1° año de la materia Sistemas y organización. Dichos alumnos son nativos tecnológicos. Se trabajó en grupos con conocimientos homogéneos sobre la actividad a desarrollar.

Se realizaron prácticas presenciales para que se pudieran adquirir conocimientos de los principios de la metodología Scrum y se planificó con tiempo la actividad.

Se establecieron períodos cortos de trabajo bajo una lista de requisitos (consignas) que se acordaron antes de comenzar el trabajo. A diario se realizan encuentros para ver el avance de la resolución de las tareas y cuáles han sido las dificultades encontradas.

Se distribuyeron los roles e identificación de artefactos.

La experiencia se desarrollo a través de casos de estudios propuestos en una guía que se publicó anticipadamente en el aula virtual a través de la plataforma Moodle.

Se propusieron modelos de casos de estudio similares a proyecto de desarrollo de software con consignas y sugerencias de resolución con aplicación de buenas prácticas.

Se propuso la formación de los grupos de no más de 4 integrantes.

Se propusieron las herramientas colaborativas y Tics utilizando los servicios de correo web tal como Gmail y Hotmail y a través de la plataforma Moodle los foros y servicio de mensajes. También se integraron los servicios de Skype, el chat de Gmail y redes sociales como WhatsApp y Facebook como así también el seguimiento de las versiones de los archivos que se compartieron a través de Google Drive [Fig. 1] pudiendo definir la forma de acceso y los permisos. Lo interesante de esta herramienta es que simultáneamente quienes tienen permiso de acceso pueden efectuar modificaciones sobre el archivo visualizando en forma las acciones de los distintos usuarios colaboradores.

Se establecieron las consigna, fecha inicio y fin de la actividad, resultado de la evaluación, elaboración de la conclusión.

Los participantes más allá de manejar herramientas tecnológicas que implican dependencias se trataron de aprender cómo manejar eficientemente el manejo del tiempo y las emociones.

Luego se obtuvo y se compartió la retroalimentación de la actividad.

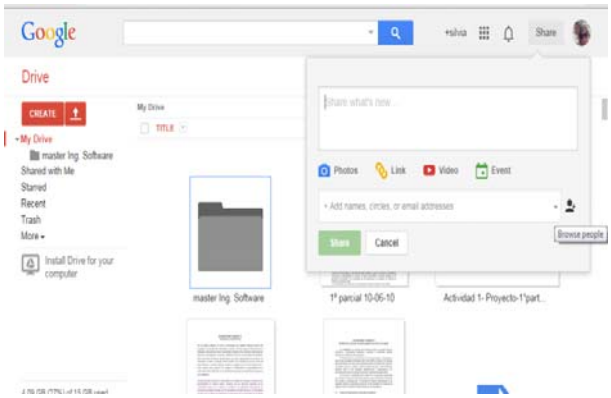


Figura 1. Formas de Acceso compartido a Archivos en Google Drive

Se instruyó a los participantes en los principios de las metodologías ágiles y en particular en los artefactos y los principios de Scrum.

La asignación de roles, artefactos y reuniones diarias (*daily stand up meeting*), según el framework de Scrum. [fig. 2]

La planificación de la práctica didáctica fue a través del patrón pedagógico [fig. 3].

El Diseño e implementación fue con los recursos de la plataforma Open Knowledge (Moodle) [fig. 4].

La recogida de datos de la experiencia fue a través de encuesta a todos los participantes, observación del trabajo en laboratorio y los informes de la plataforma Moodle.

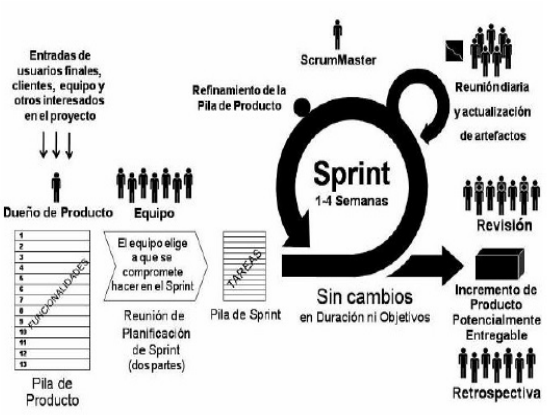


Figura 2- Framework de Scrum

	EXPERIENCIA DIDÁCTICA		
	SPRINT 1	SPRINT 2	SPRINT 3
SCRUM MASTER	Reuniones sincrónicas de trabajo en equipo	Reuniones sincrónicas de trabajo en equipo	Reuniones sincrónicas de trabajo en equipo
	PLANIFICACIÓN	PLANIFICACIÓN	PLANIFICACIÓN
	EJECUCIÓN/ SEGUIMIENTO	EJECUCIÓN/ SEGUIMIENTO	EJECUCIÓN/ SEGUIMIENTO
	ENTREGA	ENTREGA	ENTREGA
	RETROSPECTIVA	RETROSPECTIVA	RETROSPECTIVA
	INCREMENTO 1	INCREMENTO 2	PRODCUTO FINAL
	APROBACIÓN	APROBACIÓN	APROBACIÓN
	EVALUACÓN GENERAL DE LA EXPERIENCIA		

Figura 3 Patrón pedagógico

Resulta de interés el concepto de patrones pedagógicos, estructuras que buscan formalizar en modo simple y resumido las estrategias de enseñanza-aprendizaje aplicadas a una situación o problema determinado y recurrente, intentando capturar no sólo la forma en que se desarrolló la actividad sino también los detalles de su contexto, los factores que influyeron en ella, los roles de los participantes, recursos implicados y otros aspectos relevantes, de forma de facilitar la comunicación de la experiencia en forma transferible y repetible.[11].



Figura 4 Portada del curso en la plataforma Open Knowledge (Moodle)

4. Elementos del trabajo y metodología

Los métodos empleados para el desarrollo del presente trabajo, son de corte cuantitativo y están basados en el registro detallado de las encuestas realizadas a todos los alumnos de un curso al final de las prácticas en el laboratorio , en donde se solicitan que expliquen las acciones individuales y del grupo y las interacciones entre los mismos a la hora de trabajar de manera grupal. Aplicar Scrum como marco de trabajo implicó tomar como referencia un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y que llevó un tiempo y dedicación importante en la coordinación y en la comprensión de los artefactos a utilizar para lo cual se vino haciendo una adaptación continua a las circunstancias de la evolución del proyecto.

Consideramos que esta experiencia puede ser extrapolada a otras cátedras que desarrollen procesos similares.

El diseño del caso de estudio está conformado por un conjunto de materiales didácticos y tecnológicos, en pos de cumplir con un objetivo medular de la cátedra y busca la integración de los contenidos de sistemas y organizaciones. Como objetivos derivados, con el proyecto se pretende incorporar el uso de medios tecnológicos, para la mejor resolución de cada caso de estudio y valorar el uso de un lenguaje preciso y técnico, como organizador de las habilidades y destrezas individuales y/o grupales.

En el apartado 3 se presentaron diversos recursos colaborativos, artefactos de la metodología Scrum y de un patrón pedagógico para planificar la experiencia a los cuales en esta sección se les concede un uso específico.

Dicho uso responde exclusivamente a la necesidad de mejorar el proceso de enseñanza-aprendizaje de los niveles iniciales de de la carrera de ingeniería de sistemas.
La propuesta de actividades se plantea bajo los pasos de la siguiente propuesta metodológica:

1. Se realiza la Planificación del Sprint (*sprint planning*) en un caso de estudio donde se establecen las consignas a realizar.
2. Luego de realizado el caso se intercambian las resoluciones para que el rol de algunos grupos sea de revisores y verifique las falencias o cambios a la situación problemática. Seguimiento Diario (*daily scrum)*
3. Luego el equipo docente hace una puesta en común haciendo uso de un retroproyector y propone reflexiones. Revisión del Sprint y Retrospectiva (*sprint Review/retrospective*).

Se deja abierta la posibilidad de buscar alternativas diferentes de análisis para futuras prácticas.

Caries Monereo [12] identifica y explica algunas características de este nuevo grupo:

- Manejan una variedad de recursos para obtener información: páginas webs, discos rígidos, teléfonos celulares, comunidades virtuales, redes sociales, etcétera.
- Utilizan y decodifican diferentes tipos de lenguaje que, además, no se presentan secuencialmente, sino en forma simultánea, como animaciones, fotografías, gráficos, textos, hipertextos;
- Crean nuevas producciones a partir de partes de otros productos (copiar-pegar);
- Respecto del conocimiento, son relativistas por excelencia; por un lado, porque la web se actualiza permanentemente, y por el otro, porque toda información es considerada válida.

La realidad relevada en nuestros alumnos nos muestra que sobre 100 alumnos encuestados el 13% son mujeres y el 87% hombres.

El 43 % tiene entre 17 y 19 años, edad probable de los ingresantes. Un 20% entre 20 y 21 años.

El 92 % ha empleado internet a lo largo de sus estudios secundarios para la búsqueda de información, consideran que les ayudó a profundizar los temas y realizar sus tareas. También lo emplearon para el entretenimiento, como un medio de comunicación: para recibir y enviar materiales a los profesores, compañeros de grupo, entre otras.

El acceso a internet lo realizan empleando distintos medios. Observando que el 79% cuenta con Pc para hacerlo, 58% tiene celular, el 45% tiene notebook y el 24% netbook.

En cuanto a los recursos de internet que conocen tenemos que manejan servicios de correo electrónico (69% Gmail y Hotmail, servidor de correo de la universidad y el del aula virtual, 38% Yahoo y 24% el Outlook), redes sociales, buscadores, whatsApp, wikis, etc.[Figura 6]

A continuación mostramos la comparación de los recursos de internet, marcando la diferencia entre lo que se conoce y lo que se utiliza.

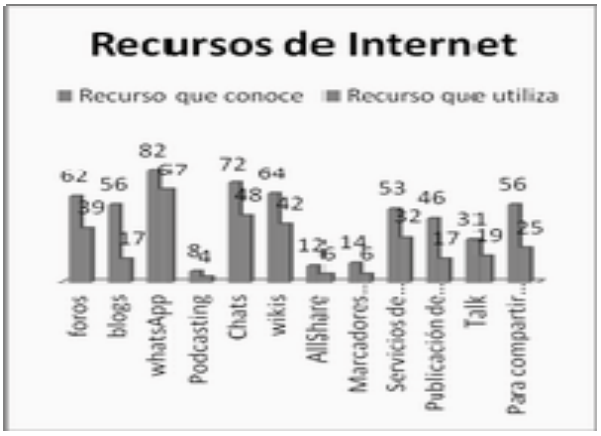


Figura 6: Recursos de Internet

En cuanto a la frecuencia de uso de la plataforma Moodle y de los recursos de internet tenemos que el 34 % los usa una o dos veces por semana y el 29% diariamente y el 28 % sólo cuando lo necesitan.[Figura 7]



Figura 7 Frecuencia de uso del Aula virtual Open Knowledge (Moodle)

Se puede apreciar que el 43% de los alumnos disponen en su equipo de auriculares con micrófono integrado y webcam o cámara de video incorporada al monitor. [Figura 8]

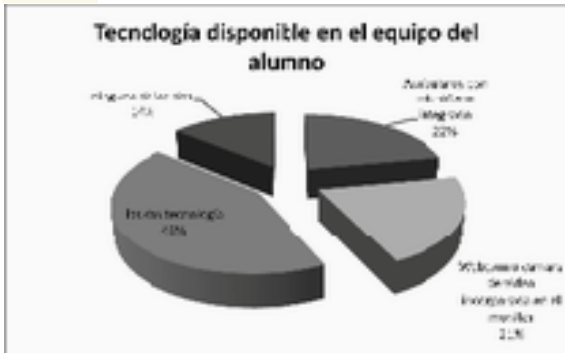


Figura 8 Recursos tecnológicos disponibles

Consideramos que éste manejo diverso de herramientas tecnológicas constituye un aspecto propicio para el aprendizaje y así favorecer diferentes usos del tiempo, los espacios, las formas de agrupamiento.

Como así también promover la evaluación continua, la autoevaluación entre pares, la del docente, la producción escrita y oral , que a su vez involucre instancia de meta cognición , es decir, de reflexión de los estudiantes sobre sus propios modos de aprender y sobre lo aprendido.

Por ello, dado el carácter colaborativo del proyecto, permitirá focalizar el desarrollo del modelo que permita compartir y potencializar habilidades de la comunicación.

5. Resultados obtenidos/esperados

La propuesta se encuentra en un grado de avance importante con dos años de aplicar la misma experiencia que comparada a la modalidad anterior ha permitido obtener alta motivación en involucrarse en el aprendizaje de los alumnos.

Este análisis da evidencias de que sumar distintos recursos tecnológicos disponibles en la web y sumarlos a la forma de trabajo habitual bajo una metodología ágil ha permitido obtener beneficios y desafíos.

Se propone permanentemente acuerdos comunes entre cada grupo y se modera cada forma de participar complementando las resoluciones en la clases presencial.

Por los aportes que propone la metodología ágil permite tener resultados incrementales y en lapsos cortos de tiempo que permite hacer retroalimentación activa entre los participantes.

Es importante destacar que desde las actividades académicas y profesionales de los participantes se han abordado distintas instancias experiencias de aplicación de los recursos de la plataforma Moodle y de los aportes de las herramientas colaborativas y Tics que complementan las necesidades de comunicación.

Aspectos identificados como beneficios

- Mejora en la motivación del aprendizaje de los alumnos preparándolos para la dinámica de la realidad profesional.
- Desarrollo de competencias transversales.
- Lograr aportes favorables de compartir nuevos conocimientos y reafirmar los actuales.
- Incorporación de las oportunidades que permite la comunicación a través de las TICs.
- Superación de las barreras del tiempo y de la distancia.
- Proporciona mayor autonomía a los participantes.
- Permite la autogestión del aprendizaje.
- Mejora continua del proceso de aprendizaje colaborativo.

Desafíos a lograr

- Desafíos innovadores de los docentes
- Formación en la configuración de nuevos recursos colaborativos.
- Planificación de seguimiento de los dispositivos de aprendizaje para cumplir los plazos determinados.
- Mayor dificultad de evaluar a los participantes.
- Dificultades en la gestión del tiempo y en los acuerdos internos de cada grupo.

6. Discusión

Se abre la posibilidad como desafío de incorporar Scrum como metodología de trabajo en grupo colaborativos para aplicarlo en ambientes diferentes a la construcción de software como producto final dado que este proyecto se enfoca en la obtención de un producto educativo. También se pretende lograr que los integrantes del grupo de investigación adquiera conocimientos y experiencia en el desarrollo de un proyecto bajo una metodología ágil poniendo énfasis en pautas de Scrum y profundizar el conocimiento del uso de una metodología ágil (aplicada en la actualidad en proyectos de desarrollo de sistemas), y considerar su aplicación en los procesos críticos de las organizaciones [13]. De esta manera se sostiene que los desafíos impredecibles pueden ser fácilmente enfrentados de una forma predictiva y planificada. Poniendo en foco la tendencia de maximizar la capacidad del equipo y no de los procesos.

La experiencia nos permitió comprobar que el trabajo colaborativo apoyado por Tics [14] ha permitido obtener beneficios significativos .

7. Formación de recursos humano Transferencia de los resultados

En el plano académico, los resultados que se generaron de las experiencias obtenidas a partir de ésta investigación se transforman en insumos para la concreción de nuevas instancias de aprendizaje colaborativo dentro de esta cátedra y la extrapolación en cátedras de similar perfil dentro de la carrera.

Se ha logrado la creación de equipos auto organizado impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros involucrados en el proyecto.

Reconocimiento de que durante un proyecto los participantes pueden cambiar de idea sobre lo que quieren y necesitan.

La experiencia ha permitido formar a los integrantes del grupo en los recursos de la plataforma Moodle [15] y proponer estrategias de aprendizaje innovadoras para aplicar a las experiencias.

Se ha logrado formar a los alumnos becarios que se suman al proyecto para que puedan participar en los congresos de esta temática.

8. Conclusión

Los cambios significativos que ocurren en nuestra disciplina de sistemas hacen que tengamos la necesidad de implementar nuevos recursos de enseñanza aprendizaje. Esto genera que se logre integración entre los alumnos y el equipo docente, y de esta manera tener que introducir cambios en los contenidos curriculares de la cátedra debiendo diseñarse actividades que permitan lograr dinámicas similares a las de un profesional dentro de un proyecto de sistemas.

El Trabajo Colaborativo presenta un conjunto de características que hacen a la integración del alumno dentro de un grupo y permite una adquisición distinta del conocimiento, respetando los tiempos no solo del entendimiento individual, como así también la posibilidad de acomodar los tiempos de dedicación personal con la utilización de herramientas colaborativas.

Se logró identificar los beneficios y desafíos que puede aportar el aprendizaje colaborativo a través de un conjunto de actividades pedagógicas organizadas en un patrón pedagógico utilizando los recursos de la plataforma Moodle bajo el marco de trabajo de la metodología Scrum pretendiendo mejorar el razonamiento individual, auto aprendizaje, y una formación integral de un futuro profesional de sistemas.

9. Referencias Bibliográficas

[1] SChwaber K. , Beedle M. "Agile Software Development with Scrum" **,** Octubre 2001.

[2] Marquez, Maximiliano,Pedroza M.Paula y Rojas Amaya Ma. Florencia, ¿Cuáles son los problemas de la implementación de Scrum? ConaIISI2013

[3] Barrera Corominas, Aleix. "El trabajo colaborativo en la Administración. Aportaciones del programa" Compartim. Cataluña : Generalitat de Cataluña, 2010. pág. 40.

[4] Sainz M.S Ibarra, Gómez G. Rodríguez El trabajo colaborativo en las aulas universitarias: reflexiones desde la autoevaluación. 334 septiembre- diciembre de 2007, Revista de Educación. pp 355-375.

[5] Mitaritonna, Alejandro Daniel"Una innovadora metodología para el desarrollo de software en ambientes de trabajo virtuales", Ciudad Autónoma de Buenos Aires, 2010, pp 12

[6] Vilaboa Pablo, Paris Mariela y Vargas Navarro Rocío "Aplicando tecnología colaborativa para generar un ámbito de integración y aprendizaje"Conaiisi2013, pp 2

[7] Calzadilla, María Eugenia."Aprendizaje Colaborativo y Tecnologías de la Información y la Comunicación". Venezuela: s.n., 2002, OEI-Revista Iberoamericana de Educación, pág. 7

[8] Moreno, Edgardo Javier; Vera, Pablo Martín; Rodríguez, Rocío Andrea; Giulianelli, Daniel Alberto; Dogliotti, Mariano Gastón; Cruzado, Graciela " El Trabajo Colaborativo como Estrategia para Mejorar el Proceso de Enseñanza-Aprendizaje – Aplicado a la Enseñanza Inicial de Programación en el Ambiente Universitario" pp2.

[9] Schwaber K. , Beedle M. Agile Software development with Scrum **,** Octubre 2001.

[10] Vilaboa Pablo, Paris Mariela y Vargas Navarro Rocío "Aplicando tecnología colaborativa para generar un ámbito de integración y aprendizaje"Conaiisi2013, pp 5

[11] YAZYI, Sergio "Una Experiencia práctica de Scrum a través del aprendizaje basado en proyectos mediado por TIC en un equipo distribuido, Salamanca Junio 2011."

[12] Monereo, Caries y Castelló M. "Estrategias de Enseñanza y Aprendizaje, 2001.

[13] Rey,Susana, Lanza Castelli, Silvia. "Aplicación de Software colaborativos o groupware empleando metodología SCRUM en Sistemas y Organizaciones" , Wicc 2013.

[14] Las tecnologías de la información y la comunicación (TIC) en la educación.
http://www.unesco.org/new/es/unesco/themes/icts/

[15] Cano Gallego, José C. "Guía de Moodle– Plataforma a distancia "

Yo Virtual

Rosana Hadad Salomón[1], Jorge Buabud[1], Fernando Rubén Araujo[1], María Concepción Caporale [1], Elizabeth M. A. Dufour[2], Mario A. Paredi[2]
[1] Docente - Investigador UTN Facultad Regional Tucumán, Yerba Buena, Tucumán, Argentina {rosanahadad, jbuabud, araujorf, concecaporale}@gmail.com
2 Graduado Ingeniería en Sistemas de Información- Investigador UTN Facultad Regional Tucumán, Monteros, Tucumán, Argentina {aledufour, marioaparedi}@gmail.com

Resumen

En el presente trabajo nos proponemos explicar la manera en la que se implementó un servidor que aloja una Plataforma de Aprendizaje 3D en la Universidad Tecnológica Nacional - Facultad Regional Tucumán, (UTN-FRT), el cual servirá de apoyo a las clases tradicionales.

A tal fin, se creó un espacio 3D llamado Isla, en donde se realizaron pruebas con un grupo de estudiantes y docentes, quienes previamente desarrollaron ciertas actividades dentro del espacio virtual a través de un personaje llamado Avatar.

Tanto las herramientas para implementar el servidor como el conjunto de software utilizados para el desarrollo del espacio virtual, son de características open source.

Introducción

Con "Yo Virtual" hacemos referencia a las actividades que podemos realizar los seres humanos a través de un entorno virtual 3D, que puede tener o no cierta semejanza con el mundo real, en donde los participantes pueden interactuar a través de personajes llamados avatares, realizar diferentes actividades y construir objetos, que podrán luego ser utilizados por los demás usuarios, haciendo de esta una experiencia muy completa y enriquecedora, la cual pone énfasis en la imaginación e interacción de los avatares en el aspecto pedagógico y social.

Existen diferentes entornos o mundos virtuales, dependiendo del uso que quiera dársele: recreativo, comercial, profesional o educativo.

Un mundo virtual educativo es una gran red social, compuesta por estudiantes y profesores, que permite desplegar la intervención de los sentidos para alcanzar mejor la captación del conocimiento. Es una forma innovadora de enseñar y de aprender en forma amena, como apoyo a las clases tradicionales [1].

El presente trabajo tiene como objetivo contar la experiencia de desarrollar e implementar el Mundo Virtual de la UTN-FRT, que actualmente lleva su segunda versión [2]. La idea principal es relatar tanto su aspecto técnico como la organización del equipo de trabajo.

La integración técnica y el desarrollo de su software fueron actividades que consideramos importantes compartir con la comunidad educativa, difundiendo así la importancia de implementar nuevas herramientas de enseñanza y formando equipos de trabajo integrados por docentes, graduados y estudiantes.

Metodología de Trabajo

Tecnologías utilizadas

Para el desarrollo del Mundo Virtual y la puesta en marcha del servidor se utilizaron los siguientes software:

OpenSim: Simulador 3D multiplataforma de código abierto, el cual se utiliza para crear un entorno virtual (mundo virtual), al que se puede acceder a través de una variedad de clientes, los cuales pueden moverse y realizar actividades dentro del mismo[3].

Moodle: Plataforma tecnológica de aprendizaje, tipo LCMS (Learning Content Management System). Es una aplicación web de Ambiente Educativo Virtual, un sistema de gestión de cursos, de distribución libre, que proporciona a los educadores, administradores y estudiantes un único sistema robusto, seguro e integrado para crear comunidades de aprendizaje en línea personalizadas [4].

Sloodle: Es un proyecto libre y de código abierto que integra los entornos virtuales multiusuario de SecondLife (Mundo Virtual más usado en el mundo) y/o OpenSim con el Sistema de Gestión de Aprendizaje Moodle. Además proporciona una gama de herramientas para apoyar el aprendizaje y la enseñanza en el mundo virtual inmersivo; las cuales están totalmente integradas con un sistema de gestión de probada eficacia basada en la web de aprendizaje, utilizado por cientos de miles de educadores y estudiantes de todo el mundo, como lo es el Moodle [5].

SingularityViewer: Es un proyecto de código abierto, un cliente o visor para SecondLife y OpenSim, que se

esfuerza por combinar la última y mejor tecnología disponible para mantener la compatibilidad con los futuros cambios y características de SecondLife [6].

Base de Datos MySQL: Sistema de gestión de bases de datos relacional, multihilo y multiusuario, utilizado por el servidor OpenSim y Moodle [7].

Debian GNU/Linux: Sistema Operativo basado en software libre, distribución Linux de alta calidad, estable y escalable. Se destaca por ser el primero en incluir un sistema de gestión de paquetes para permitir una fácil instalación y desinstalación del software [8].

Para comprender cómo se compone nuestro mundo virtual debemos describir sus componentes individuales dentro de cada software. El mundo virtual que se formó en este caso es una fusión entre la plataforma Moodle y Opensim los cuales se conectan a través de un módulo de Moodle llamado Sloodle.

Al instalar, configurar, e inicializar Opensim, podremos acceder a través del visor al mundo virtual y encontrar los siguientes elementos:

Avatar: personaje 3D creado generalmente a semejanza del usuario.

Prim: forma geométrica a través de la cual se forma nuestro Mundo Virtual, todo lo que está contenido en él (excepto los avatares y los árboles) están hecho a partir de éstos.

Isla o Sim: es una porción de tierra o territorio al cual se otorga un nombre. Por ejemplo nuestra isla se denomina UTN-FRT.

Estos tres elementos son los únicos que se visualizan al "comenzar" o crear un mundo virtual.

Por otro lado, tenemos corriendo la plataforma Moodle con el módulo Sloodle instalado. Este último nos permite realizar una serie de asociaciones entre las actividades y recursos del Moodle con OpenSim.



Figura 1. Correspondencia entre objetos Moodle y Sloodle.

Lo que se quiere decir es, por ejemplo, que nuestro Cuestionario en 2D se transforma en un QuizChair en 3D, el cual es su objeto compatible en Sloodle y se visualiza de esta forma en Opensim:

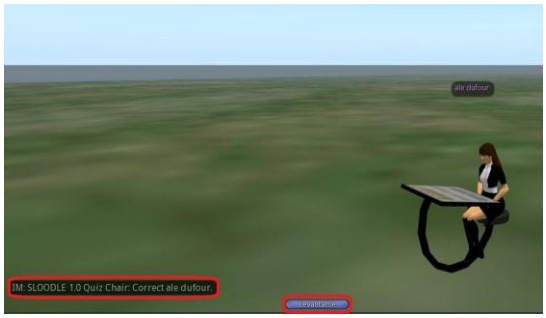


Figura 2. QuizChair.

De igual manera se realiza con los demás objetos vistos en la Figura 1, y esto demuestra una forma diferente de realizar actividades e incentivar a los alumnos y docentes a utilizar esta nueva tecnología de aprendizaje. La plataforma Moodle grabará todos los datos resultantes de las actividades en Opensim en su base de datos, tal como lo realizaría en el entorno 2D, lo cual facilita el control y la revisión de las actividades propuestas por los docentes.

También existen otros objetos propios de Sloodle que sólo se visualizan en el mundo virtual como por ejemplo una Cabina de Matriculación, la cual hace analogía con la matriculación a un curso, existe también una zona de login, la cual permite al avatar teletransportarse hacia su clase en Moodle en la cual tiene que estar registrado, y por último una Máquina Expendedora la cual hace de repositorio de objetos, en ella se puede guardar todo lo que queramos que hayamos creado en el mundo virtual, por ejemplo todo el mobiliario para conformar un aula, de manera que pueda ser reutilizado por todos los docentes. Además en la última versión de sloodle se ha incorporado una Tabla de Puntuación, la cual conectada con Moodle realiza una tabla de posición con las actividades que van realizando cada usuario.

Todo lo nombrado es visualizado en un visor, en este caso el Singularity, el cual puede ser descargado de manera gratuita desde su página web.

Integración de tecnologías

Para conformar nuestro piso tecnológico e integrar los distintos software utilizados para el desarrollo del Mundo Virtual UTN FRT hubo que superar distintas dificultades en cuanto a la compatibilidad y funcionamiento de las versiones que utilizamos[9].

En esta segunda etapa se decidió actualizar las versiones, lo cual nos permitió tener nuevos objetos y lograr el funcionamiento completo de otros. El detalle es el siguiente:

OpenSim: pasamos de utilizar la versión 0.7.1 a la versión 0.7.6, lo que nos permitió mejorar algunas características de nuestro mundo virtual.

Moodle: pasamos de utilizar la versión 1.9 a la versión 2.5, lo que nos permitió mejorar en cuanto a nuevos recursos y actividades, que nos pueden ser de utilidad para nuestro mundo virtual.
Sloodle: pasamos de utilizar la versión 1.0 a la versión 2.1.11 para moodle 2.5, la cual tiene nuevos objetos para hacer más atractivas y dinámicas nuestras aulas del mundo virtual.
Visor: dejamos de utilizar el visor Imprudence y comenzamos a utilizar el *Visor Singularity*, el cual nos permite subir objetos en distintos formatos realizados con software externos a nuestro mundo virtual. La versión de nuestro Visor es 1.8.4 (5434) para Windows 32-64bits.
Sistema Operativo: este es el principal cambio que realizamos, el cual nos permitió tener una mejor performance de nuestro mundo virtual. A pesar de no contar específicamente con un Servidor, se destinó una computadora (Notebook) que hiciera las veces de Servidor, instalando el S.O. Linux Debian.
Base de Datos: Opensim en su distribución viene configurado por defecto para trabajar con la base de datos SQLite. A efectos de estandarizar el uso de las bases de datos, ya que Moodle emplea como sistema gestor de base de datos (SGBD) MySql, e incrementar las prestaciones del SGBD, decidimos unificar y emplear MySql también para Opensim.

Con estos cambios se solucionó los inconvenientes de compatibilidad, y a su vez con la implementación de un servidor se pudo sortear los inconvenientes de trabajar en una sola computadora lo cual complicaba el desarrollo y avance de nuestro Mundo Virtual.

La primera versión de nuestro mundo virtual parte de una tesis de grado que realizaron dos personas que pertenecen al equipo, la misma "corría" en una máquina personal haciendo de servidor y cliente a la vez, lo cual hacía difícil la conexión de dos o más avatares al mismo tiempo, sin embargo las pruebas de conexión entre los objetos Moodle y Sloodle fueron correctas así como también su interacción con la base de datos, logrando así demostrar que el mundo virtual propio podía ser un hecho aplicado a nuestra facultad.

Conformación del equipo de trabajo

La conformación del equipo de trabajo para continuar con la investigación y puesta en marcha de un servidor único para el mundo virtual se realizó con la participación de docentes y alumnos de distintas áreas de la ingeniería, por ejemplo Matemática, Modelos y Simulación, Programación, etc.

El equipo de investigación se divide en 3 áreas:

- Infraestructura, que es la encargada de controlar el servidor y de proveer los estándares básicos de cómo será la imagen y visualización del mundo virtual.
- Programación, la cual se encarga de desarrollar el comportamiento que se necesita sobre ciertos objetos.
- Aplicaciones e Interfaz, aquí se realiza el estudio, diseño e implementación de los objetos Moodle y Sloodle.

A lo largo de nuestra investigación hemos logrado poner a prueba el trabajo en equipo y la coordinación del mismo, trabajando primeros con las ideas y objetivos que necesitábamos plantear antes de comenzar, ideas que luego fueron convirtiéndose en realidad con la suma de docentes, alumnos y graduados en nuestro equipo.



Figura 3. Equipo de Trabajo.

Desarrollo de la prueba

Para el desarrollo de la primera prueba se realizó la instalación de los visores en un Laboratorio principal de la Facultad con el objetivo de poner a prueba la conectividad simultánea de varios avatares al mismo tiempo.

Para que un mundo virtual funcione correctamente se requiere de un servidor que provea el servicio en línea de forma persistente, activo y disponible 24 horas al día y todos los días. Por tal motivo, una de las etapas del proyecto consistió en verificar si el servidor con el que se cuenta iba a poder soportar varios usuarios activos simultáneamente.

La notebook que funciona como servidor tiene las siguientes características: Procesador Intel Core i5, Memoria RAM de 6GB DDR3, Disco Rígido de 750 GB, Unidad Óptica CDRW/DVDRW, VGA Intel HD Graphics 4000, Resolución 1366x768 16:9 panel.



Figura 4. Notebook servidor.

Para la realización de la prueba se hizo uso de un Laboratorio de la UTN FRT que cuenta con cuarenta computadoras de última generación. Esto nos permitió contar con una cantidad considerable de usuarios conectados concurrentemente, realizando actividades de construcción de nuevos objetos dentro del mundo virtual.

Además se contó con la posibilidad de realizar cambios en los Avatares (personajes de los usuarios) a través de las herramientas propias del simulador y el cliente utilizado, como así también con creaciones de vestuario que fueron colocados dentro del espacio virtual para que los usuarios puedan hacer uso de ellos y experimentar los distintos cambios que se pueden realizar en el personaje.

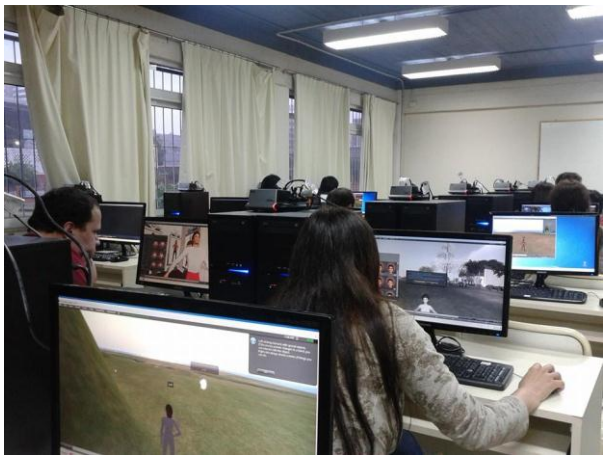


Figura 5. Prueba en el laboratorio.



Figura 6. Realización de cambios en la apariencia de un avatar.

Resultados

En el escenario de prueba realizado se pudo experimentar claramente con las herramientas de creación de objetos de OpenSim, como así también con objetos y herramientas externas a la plataforma.

Además se pudo observar la performance del servidor, el cual se mantuvo en línea constantemente durante las pruebas realizadas.

De más está decir la sorpresa y aceptación con la cual interactuaron los alumnos al descubrir sobre qué realizarían las pruebas a las que fueron convocados, y de cómo se piensa aplicar en nuestra facultad.

Se realizaron pruebas de vestuario y manejo de los avatares, así como también un recorrido general sobre la estructura de prototipo que acompañaba el entorno.

La prueba de fuego era la conexión de más de 20 personas en el servidor que habíamos implementado y como así también si el ancho de banda de internet sería suficiente, lo cual fue superado.

Se concluyó como satisfactoria la prueba debido tanto a la aceptación de los alumnos como a los resultados de la experiencia.

Conclusiones

A través de la conformación de las diferentes áreas de trabajo e investigación, se abrieron oportunidades de realizar prácticas supervisadas y proyectos finales de carrera para los alumnos de nuestra facultad, dando mayor relevancia al proyecto de investigación.

Con la realización de la primera evaluación del desempeño del prototipo del Mundo Virtual, en la cual participaron alumnos, se pudo concluir a través de una encuesta online que fue muy interesante y divertido poder interactuar y desplazarse dentro de un Mundo Virtual.

También se pudo verificar que el servidor no tuvo inconvenientes para soportar la cantidad de usuarios que estuvieron participando de la prueba.

Aún queda continuar con: una segunda etapa de implementación, pruebas y evaluaciones. Realización de capacitaciones, para describir cómo debe ser usado el software por parte del usuario. Elaboración de un documento con los resultados obtenidos, para su publicación y difusión como parte del trabajo de comunicar la ciencia.

Con la investigación sobre mundos virtuales se pretende:

Que los alumnos y docentes se interesen por el tema, como innovación en la forma de interactuar y recibir el conocimiento.

Un uso didáctico de los mismos, que se traducirá en estudio de los contextos educativos, de las actitudes de los profesores y de los estudiantes frente al uso de estas tecnologías y de las estrategias de formación del docente en los medios de enseñanza.

Implementar una herramienta innovadora y atractiva como apoyo al dictado de clases, la cual sirva de nexo con la plataforma Moodle agregándole a esta un condimento especial y revalorizando la misma.

Lograr que los alumnos se sientan contenidos y cómodos como nativos digitales que son [10,11].

Referencias

[1] Badilla Quintana, Vera Sagredo, Carripán Sáez, Quilodrán. "Experiencias y reflexiones sobre el aporte de los mundos virtuales inmersivos en las habilidades pedagógicas y tecnológicas de alumnos en Formación Inicial Docente: la experiencia del proyecto TYMMI en OpenSim" (2014). http://www.virtualeduca.org/ponencias2014/347/BadillaVeraCarripanQuilodran.pdf . Accedido el 31 de Julio de 2014.

[2] Hadad Salomón, Dufour, Paredi (2013). http://www.42jaiio.org.ar/proceedings/simposios/Trabajos/JSL/06.pdf. Accedido el 31 de julio de 2014.

[3]Comunidad Open Simulator. What is OpenSimulator?.Opensimulator.http://opensimulator.org/wiki/Main_Page. Accedido el 31 de Julio de 2014.

[4]Comunidad Moodle. Bienvenido a la Comunidad Moodle. Moodle.http://moodle.org/.Accedido el 02 de agosto de 2014.

[5]Comunidad Sloodle. AboutSloodle.Sloodle.http://www.sloodle.org/moodle/. Accedido el 03 de agostode 2014.

[6] Comunidad Singularity. Singularity Viewer.Singularity.http://www.singularityviewer.org/. Accedido el 02 de agosto de 2014.

[7] Comunidad MySQL. The world's most popular open source database.MySQL. http://www.mysql.com/ . Accedido el 30 de Julio de 2014.

[8] Comunidad Debian. DebianThe Universal OperatingSystem. Debian. https://www.debian.org/index.es.html. Accedido el 01 de agosto de 2014.

[9] Ramón, Russo, Esnaola, Alonso, Fochi, Padovani. "Implementation of a 3D Virtual Environment at the National University of the North West of the Province of Buenos Aires" (2013). http://sedici.unlp.edu.ar/handle/10915/32050 . Accedido el 15 de Julio de 2014.

[10] Nativos e inmigrantes digitales, Marc Prensky (2011) EDICIONES SM.

[11] Nativos e Inmigrantes Digitales, Alejandro Piscitelli (2009) Santillana.

Las Competencias Estudiantiles de Programación y las Tecnologías, como nuevos medios para la Formación e Integración

Marta Castellaro[1], Malva Alberto[1], Daniel Ambort[1],
[1] *Universidad Tecnológica Nacional. Facultad Regional Santa Fe.*
Lavaisse 610, (3000) Santa Fe, Argentina.
***mcastell@frsf.utn.edu.ar*, *mtoso@frsf.utn.edu.ar*, *dambort@gmail.com*,**

Abstract

Despertar vocaciones hacia la ingeniería, atraer a los estudiantes de las escuelas secundarias para estudiar carreras del área Informática/Sistemas de Información, motivar y retener a los alumnos que cursan los primeros años y favorecer la búsqueda y arraigo de habilidades adecuadas para el desarrollo profesionalizante en los alumnos más avanzados, son algunos de los desafíos que encontramos quienes nos responsabilizamos de la formación de estas nuevas generaciones. Como equipo docente y de investigación, nos preocupa el proceso que se inicia en la articulación con la escuela secundaria en un espacio escolar incipiente como es la informática y continúa dentro de la universidad con el desarrollo de secuencias didácticas que integran una buena selección de contenidos y la ejecución de actividades para el afianzamiento de las habilidades y destrezas más requeridas dentro del área de la programación. En este trabajo describimos una experiencia extracurricular, que tiene como objetivo la reunión de estudiantes de distintos niveles de ingeniería, junto a alumnos de escuelas secundarias y que, apoyada por los nuevos medios tecnológicos, permite desarrollar en los involucrados aptitudes y habilidades genéricas y tecnológicas. Se trata del proceso de concientización, preparación y realización de una competencia/jornada denominada TECNOMATE, que la hacen posible, docentes y alumnos de distintos niveles educativos y es apoyada por empresas y organismos del medio regional. Las experiencias 2012 y 2013 han sido incrementalmente significativas y el proceso iniciado en 2014, permite avizorar mayor participación y continuidad.

1. Introducción

Acercar la Universidad a las Escuelas para dar el mensaje que estudiar ingeniería es una posibilidad, despertar vocaciones científico-tecnológicas, atraer a los jóvenes de las escuelas secundarias para acompañar en la decisión de continuar en el nivel superior, motivar y retener a los alumnos que cursan los primeros años universitarios y generar procesos de intervención favorecedores del afianzamiento de habilidades y destrezas adecuadas para el desarrollo pre-profesional en los más avanzados, son algunos de los desafíos que encontramos los docentes de ingenierías. En un momento en que los ciclos de innovación son cada vez más breves, los equipos universitarios se ven obligados pero a la vez dispuestos, a una mayor versatilidad y flexibilidad en cuanto a la selección de contenidos, actividades y recursos necesarios, que les permitan anticipar, prever, adaptar, gestionar y dar solución a estos nuevos desafíos. Para alcanzar éstos y otros nuevos modelos pedagógicos surgentes, se requiere articular tempranamente, desarrollar y participar en estrategias conjuntas entre todos los actores y asegurar que el acceso al conocimiento sea genuino y significativo.

Algunos aspectos que definen el contexto en el que deben afrontarse esos desafíos, y algunas acciones precursoras de la experiencia que presentamos en este trabajo son:

1.1. Motivar y despertar vocaciones y dar orientaciones esclarecedoras

La problemática del pregrado requiere una mirada integral, incluyendo todos los puentes necesarios de articulación con el nivel medio. El acceso de los alumnos a estudios superiores constituye un reto permanente, un problema complejo y con muchas brechas. La coordinación de esfuerzos sociales debe hacer asequible y accesible el ingreso de la comunidad escolar y en especial de los sectores más vulnerables, así como garantizar posibilidades genuinas de permanencia de los jóvenes que decidieron estudiar Ingeniería. Señala Cullen [1] que los mecanismos de articulación con el nivel anterior deben incluir acciones para que cada vez más alumnos decidan inicialmente continuar estudios superiores, luego

que puedan hacerlo, en función de su esfuerzo y perseverancia, pero sin encontrar obstáculos insalvables por falta de formación básica.

Esto resulta aún más significativo cuando se trata de carreras de ingenierías, que implican duraciones y disciplinas a abordar que no son las habituales en el imaginario colectivo de los adolescentes que finalizan la escuela. A esto se suma la responsabilidad de la universidad, de atender una situación local, nacional y mundial por la falta de graduados y estudiantes de carreras de ingeniería. Por otro lado, los medios de comunicación y los encuentros de especialistas [2] revelan esta falta de ingenieros como una carestía para el desarrollo social y destacan la necesidad de generar nuevas propuestas para promover vocaciones y decisiones por parte de quienes van a formarse en niveles superiores.

La concientización sobre la falta de ingenieros en general y la escasez de graduados para atender las demandas sociales crecientes en ramas afines a la informática y sistemas de información deben ser encauzadas para constituirse en motivadoras y esclarecedoras para revertir la situación actual del área.

Sintetizando, dar posibilidades genuinas de ingreso y permanencia de los jóvenes en nuestras aulas de Ingeniería requiere atender al menos a tres aspectos estrechamente vinculados: la difusión; la articulación inter-niveles y la permanencia de los ingresantes vista como una adaptación gradual e integral del joven al mundo universitario.

1.2. Generar experiencias que dejen huella

Es nuestro propósito generar e implementar estrategias, dirigidas a los alumnos de los últimos años de la escuela secundaria y compartidas con los docentes de pregrado, para favorecer la toma de decisiones respecto de la necesidad de continuar estudiando; orientar en el proceso de esclarecimiento y elección, y colaborar para atemperar la marcada brecha desde el "ser aspirante" hasta convertirse en "ser ingresante".

"Argentina necesita ingenieros, sumate al desafío" no es sólo un slogan de nuestra Facultad, es una realidad que da cuenta de la falta de ingenieros. La ingeniería es una profesión de pleno empleo donde la demanda supera con creces la oferta. Sin embargo en nuestro país son escasas las vocaciones ingenieriles tempranas. En nuestra Facultad, mantener el número de aspirantes durante los últimos años ha exigido esfuerzos adicionales. Los jóvenes aspirantes, afrontan instancias de indecisión, incertidumbres, contradicciones y la responsabilidad que implica construir un proyecto de vida [3]. A esto se agrega una amplia variedad de ofertas de capacitación, una publicidad poco orientadora y un imaginario social sobre las carreras de ingenierías que las concibe como difíciles, largas, duras, no aptas para mujeres. Se suman además otras cuestiones con efecto multiplicativo: superación del desarraigo familiar, adaptación a nuevos ámbitos y metodologías de estudio; superación de carencias previas y de habilidades y destrezas escasamente ejercitadas.

Este equipo docente trabaja para ser parte responsable de la creación de mecanismos de inclusión social. Intenta definir y/o colaborar en líneas de trabajo compartidas, para mejorar los índices de aspirantes-ingresantes; socializar la importancia que tiene la formación en competencias ingenieriles en general y en programación, en particular, para atender al sector productivo del país y promover acciones tendientes a democratizar el conocimiento en temas disciplinares.

A continuación, una síntesis de los distintos tipos de actividades realizadas en los últimos años lectivos, relacionadas con experiencias de articulación que dejan su impronta para la difusión y promoción de la oferta académica puede resumirse en:

1. Atención de alumnos (y familiares) que se contactan en forma particular y de grupos que se contactan a través de Escuelas para visitar la facultad (de la ciudad y del interior).
2. Asistencia a escuelas que nos convocan a integrarnos en actividades escolares (charlas, jornadas temáticas, ferias)
3. Aprovechamiento de oportunidades de difusión que se presentan no sistemáticas (no planificadas).
4. Participación en distintas Exposiciones, Expocarreras, Ferias anuales organizadas por otras instituciones locales y regionales.
5. Organización de la Jornada de Facultad Abierta al Aspirante en la Facultad Regional Santa Fe desde 2010. A partir del año 2014, sumamos la participación en la Jornada "Escuela-Universidad" (Marzo) y Jornada "Semana de la Ingeniería" (Junio).
6. Difusión y certificación de participación de estudiantes secundarios en charlas y eventos abiertos o congresos de estudiantes que se organizan en la Facultad.
7. Contacto continuo a través de correos electrónicos (en formato de cartillas) y llamadas telefónicas, con docentes y directivos de escuelas, con noticias, avisos e información de interés sobre temas científico tecnológicos.
8. Participación en programas de apoyo que organizan otras entidades (caso Fundación Sadosky con Sistemas de Información; Jornadas de Conectar Igualdad y Voluntariados Universitarios de la Secretaría de Políticas Universitarias).
9. Adhesión al sistema de prácticas profesionalizantes del Ministerio de Educación de la Provincia de Santa Fe.
10. Organización de visitas de escuelas periféricas, con apoyo de traslado por parte de la Municipalidad.
11. Atención de peticiones específicas (caso de una

escuela de zona periférica, que solicita un apoyo en matemática para sus alumnos, al considerar que se encuentran en condiciones no favorables para afrontar el curso de ingreso; atención a la visita de docentes de escuelas técnicas en su plan de formación continua)

12. Difusión en medios (diario, televisión, suplementos; que se presentan en forma gratuita)
13. Organización y dictado de cursos para docentes y alumnos de Escuelas Secundarias en Matemática y Programación.

Entre las experiencias que dejan huellas para la apertura a estudios en Sistemas de Información mencionamos los Cursos, Jornadas o Talleres de jornada diaria en las Escuelas o en la Facultad, tales como:

El Taller "*Inicio a la programación de un robot Lego nxt 2.0*" que convoca a alumnos avanzados de Sistemas que trabajan con alumnos de las escuelas secundarias desde cuarto año en adelante y comparten el desarrollo de apuntes para la enseñanza de RoboMind con conceptos de programación, descripción de las funcionalidades básicas del software a utilizar y descripción del lenguaje de programación usado en el simulador RoboMind; desarrollo de guías de ejercicios prácticos; desarrollo de apuntes para la enseñanza de programación de robots Lego utilizando el lenguaje LEGOS basado en Java; desarrollo y puesta a punto del simulador para robots; apoyo para el uso del manual de instrucciones; acciones para el entrenamiento en el uso y programación [4]. Durante el año 2013 se dictaron 7 cursos a distintas escuelas. El robot formó parte de varias exposiciones, tal como lo muestran las Figuras siguientes:



Figura 1. Robot Lego en Expocarrera

El Apoyo al Programa Nacional "*Dale Aceptar"* que lleva adelante la Fundación Sadosky [5], al cual adhirió el Departamento de Ingeniería en Sistemas de Información, y se traduce en la formación dirigida a los alumnos de las escuelas secundarias para despertar vocaciones en computación, para que descubran las oportunidades y beneficios que implica estudiar una carrera relacionada con las Tecnologías de la Información y Comunicación.



Figura 2. Descripción de movimientos de Robot

La capacitación en la Jornada "*Animaciones en Mundos 3D"* con el uso del software Rebeca constituye un momento de encuentro y articulación entre la comunidad escolar y la universitaria, mediado por las posibilidades de expresión y creatividad de los estudiantes puestas en escena en las actividades propuestas con la utilización del software. La Figura 3 muestra un momento de esta actividad:



Figura 3. Jornada Mundos 3D

1.3. Desarrollar competencias de ingeniería

El CONFEDI (Consejo Federal de Decanos de Ingeniería), realizó un trabajo preliminar que contribuyó a la normativa de los estándares, con fines de una "Reforma Curricular de las Ingenierías" [6]. En el mismo se establecieron criterios generales que deben contemplarse en la formulación de los Planes de Estudio de Ingeniería, tales como:

- Desarrollar en los estudiantes capacidad de manejo de situaciones bajo incertidumbre, consolidando actitudes para la solución de problemas no tradicionales.
- Estimular la creatividad, la iniciativa personal, el trabajo interdisciplinario y la innovación en el área tecnológica.
- Desarrollar en los estudiantes capacidad de

abstracción y de reflexión crítica.
- Dominio del idioma inglés y desarrollo máximo de su capacidad para el uso de las herramientas que le brinda la informática.

Para ello establece que es necesario adquirir distintos tipos de competencias/ habilidades, las que agrupa en 4 clases:
- *Competencias básicas para ingresantes*: lectura comprensiva y rápida, escritura, expresión oral y matemáticas básicas.
- *Competencias genéricas*: conocimientos generales para realizar comportamientos laborales y habilidades que empleen tecnología, que incluye el caso del manejo de algunos equipos y herramientas.
- *Competencias específicas*: conocimientos especializados para realizar labores concretas propias de una profesión.
- *Competencias laborales*: articulación de conocimientos, aptitudes y actitudes para el mundo del trabajo.

También se define que para comenzar a cursar una carrera de ingeniería, un alumno debe responder a las competencias básicas y al inicio de las competencias genéricas, y en el marco de los primeros años de las carreras se debe asegurar la adquisición de las competencias genéricas y el inicio de las competencias específicas.

Las *competencias genéricas*, a su vez, las agrupa en tres clases:
- Competencias *sociales*, entre ellas:
- Trabajar en equipo a partir de la construcción de metas comunes, a través de un entendimiento interpersonal y en forma comunicativa; participar en actividades que contribuyan a la formación de juicio crítico y toma de decisión; adaptar y resolver las situaciones complejas haciendo uso de las capacidades adquiridas.
- Competencias *metodológicas*, entre ellas: buscar, seleccionar y utilizar estratégicamente los recursos disponibles para el estudio; resolver problemas a partir del uso estratégico y heurístico de los saberes construidos y manejar tecnologías de la información y comunicación para la resolución de problemas y construcción de nuevos aprendizajes.
- Competencias *científico–técnicas*, entre ellas relacionadas con: el lenguaje simbólico; la resolución de problemas; los modelos de simulación.

Los autores, como equipo docente, nos planteamos ¿de qué manera podemos colaborar para integrar la escena con los actores: aspirantes, ingresantes, motivación, habilidades, programación y resolución de problemas?. En el punto siguiente efectuamos algunas consideraciones particulares sobre la formación en Programación.

2. Enseñanza de Programación

La enseñanza-aprendizaje de la programación de computadoras utiliza entre otras estrategias, la resolución de problemas a través del uso de computadoras. Este proceso se descompone en varias etapas: interpretación del enunciado del problema, modelado de una solución, selección de las estructuras de datos más adecuadas a la situación planteada, escritura del algoritmo, implementación en un lenguaje de programación de alto nivel [7]. También es importante tener en cuenta la optimización del uso de los recursos disponibles para el desarrollo de un programa.

La actividad práctica es fundamental para comprender y adquirir capacidad respecto a este proceso. La intervención docente para el armado de propuestas que guíen estas acciones requiere especial atención y el empleo de técnicas y herramientas que la favorezcan.

El empleo de distintas metodologías y de las nuevas tecnologías para la formación puede apoyar la formación desde otros aspectos.

El *aprendizaje basado en problemas* es un proceso de indagación que permite resolver preguntas, dudas o incertidumbres, que se desarrolla en grupos de trabajo pequeños, persiguiendo la resolución de un problema complejo y/o desafiante, que ha sido planteado por el docente, con la intención de promover en los alumnos un aprendizaje autodirigido. Esto implica un cambio de roles, donde el estudiante debe ser sujeto activo, que trabaja en forma cooperativa.[8]

Un importante recurso tecnológico de apoyo lo constituyen los sitios web denominados *"Jueces en línea de Problemas de Programación"* [9], [10], [11] y [12], que son sitios de internet donde hay publicados una serie de retos de programación en los que se envían las soluciones y un juez virtual las acepta o las rechaza. Los problemas difieren bastante entre sí, en cuanto a dificultad y temática. Pero hay tres requerimientos comunes que las soluciones deben satisfacer:
- Corrección: el programa se ejecuta sobre distintos casos de prueba y se comprueba la salida para cada uno.
- Tiempo: todos los problemas tienen un límite máximo de tiempo. La solución que ejecute por arriba de dicho límite, será descartada.
- Memoria: en algunos problemas se establecen también limitaciones al uso de este recurso.

El requisito de corrección siempre está presente, y los de tiempo y memoria, aparecen cuando se pretende poner énfasis en la eficiencia (más alla de la eficicacia), porque se requiere optimizar el uso de uno ó ambos recursos, y establecer un orden en la calidad de las soluciones.

En cuanto a los lenguajes a emplear, casi todos los sitios admiten Pascal, C, C++, Java, aunque algunos permiten una extensa lista de lenguajes adicionales.

Los jueces más conocidos y utilizados son Sphere Online Judge (SPOJ) [13] y UVa Online Judge [14] , aunque existe ya alrededor de una centena de ellos, en varios lenguajes y con prestaciones muy diferentes.

3. Acciones previas a esta propuesta

3.1. Competencia Internacional

Como elemento motivador y superador para los alumnos avanzados de Ingeniería en Sistemas de Información (ISI) , desde hace unos años se comenzó a promover la participación de grupos de alumnos en la Competencia Internacional de Programación (Annual ACM International Collegiate Programming Contest) [15] organizada a nivel internacional por ACM, y auspiciada por IBM, que es la más antigua competencia a nivel internacional de programación.

ACM [16] es la mayor sociedad científica de computación a nivel internacional. El objetivo de la competencia es promover la creatividad, la innovación y el trabajo en equipo en pos del logro de la solución a un problema determinado y permite a los alumnos el reconocimiento y puesta en práctica de habilidades necesarias en contextos que plantean el actuar "bajo presión".

La competencia Internacional se inicia con competencias regionales en todo el mundo, de las cuales salen los equipos que participan cada año en las ACM-ICPC World Finals. En los últimos años equipos de Argentina han clasificado en la Regional Sudamericana y han participando en la Final Mundial, obteniendo en varias oportunidades lugares muy destacados. Estos equipos argentinos contaron con el apoyo del Ministerio de Ciencia, Tecnología e Innovación Productiva y del Ministerio de Educación de la Nación.

El entusiasmo demostrado por los primeros grupos participantes desde la Facultad, motivó a los docentes a generar un espacio/gabinete de apoyo para la preparación durante el año de los alumnos interesados. Luego se progresó a la realización de una competencia interna, a nivel de la Facultad, para definir que grupos se encontraban en mejores condiciones para participar y recibir el apoyo para ello.

3.2. Asignatura electiva "Programación Competitiva"

Estas acciones motivaron a los docentes a diseñar una asignatura electiva que profundiza los aprendizajes necesarios para el desempeño en los contextos asociados a este tipo de programación. Así se abrió en el año 2009 la asignatura "Programación Competitiva", la cual se viene dictando de manera ininterrumpida hasta el año en curso. El objetivo de esta asignatura es abordar técnicas algorítmicas avanzadas, habitualmente puestas en práctica para resolver problemas de competencias de programación, prestando especial atención al desarrollo de habilidades en los alumnos que favorezcan el trabajo en equipo.

4. La experiencia "Tecnomate"

La experiencia TECNOMATE surge con el objetivo de promover las siguientes competencias en los aspirantes y los estudiantes ingresantes y de niveles siguientes:

- Desafío: de competir, de participar en algo nuevo, de enfrentarse a problemas no conocidos.
- Autoestima: de superarse, de ir más allá de los conocimientos que se han impartido en los cursos y las evaluaciones realizadas.
- Integración de equipos: Conformarlos especialmente para la competencia o sumarse a equipos incompletos.
- Empleo de nuevos medios: el sitio juez para subir los problemas, probar y calificar; el software gráfico para los alumnos de escuela media.

Bajo estas premisas y teniendo en cuenta los antecedentes descriptos en la introducción, se diseñó la experiencia "Tecnomate" que se propone como una competencia local, con características básicas de las internacionales, pero ideada para que puedan participar alumnos de los distintos niveles de la carrera ISI, así como también alumnos de escuelas medias (posibles aspirantes) en una jornada común, con momentos y algunas pautas comunes, buscando los objetivos antes planteados, pero permitiendo que cada alumno pueda hacerlo desde su nivel.

La experiencia se planteó como opcional y en el caso de las escuelas secundarias se trabajó con los directivos y docentes para que faciliten la participación de los alumnos interesados, en las etapas de preparación y en la jornada.

La invitación y preparación de los alumnos que concurren a la universidad se realizó a través de varios medios: en el cursado de las asignaturas obligatorias del área Programación, en el cursado de la asignatura electiva, en el gabinete extracurricular y con tutorías ofrecidas.

En el caso de la escuela media, para los alumnos que cursan orientaciones de informática, se los invitó a participar de las reuniones del gabinete.

Para incorporar a los alumnos con poca actividad previa de programación, se propuso integrar este proyecto con acciones derivadas de la participación de la Facultad en el "Desafío Dale Aceptar", descripto en el punto 1.2., que se enmarca dentro del programa "Vocaciones en TIC" [17]. Trabaja con una herramienta denominada "Alice", un programa muy intuitivo y fácil de usar para quienes jamás hicieron una animación o un juego. Desde la página de clases se puede ver y/o bajar los videos tutoriales y aprender desde el inicio a crear los

escenarios, los personajes y las situaciones que se imaginen.

Así los alumnos de la escuela secundaria pueden optar por sumarse a grupos de alumnos del primer año de la Facultad, y trabajar en lenguajes como C o C++, o bien participar de la categoría de juegos.

Se destaca la estrategia de reunir en un mismo evento a estudiantes de escuelas medias, de primeros años de la universidad y de años siguientes. Si bien trabajan con problemas y herramientas distintas, lo hacen en un mismo recinto y con pautas comunes.

Se dio amplia difusión el encuentro; en la Figura 4 se muestra el afiche. Para reforzar la importancia de este encuentro, se realizaron acciones para lograr reconocimientos (como el Ministerio de la Producción de la provincia de Santa Fe - Subsecretaría de empresas de Base Tecnológica, y el Colegio de Ingenieros Especialistas) y también se interesó a empresas privadas para que aporten premios que estimulen los esfuerzos de los grupos. Así se lograron notebooks, tablets, radios, calculadoras, etc. En la Figura 5, se indican las distintas categorías y los premios asignados, además de otros premios simbólicos (carpetas, remeras, gorras, jarras).



Figura 4. Afiche de difusión

Categ	Descripción	1er premio	2do premio	3er premio
A	Escuela Secundarias Programación en C	1tablet 1carpeta		
B	Escuelas Secundarias con Rebeca	1tablet 1carpeta	1 bolso	1calculadora
C	Primer año ISI y Escuelas Secundarias	1 notebook	1tablet	1 calculadoras
D	2do año ISI	1notebook 1radio	1 bolso 1jarro	1 calculadoras
E	3ero y 4to ISI	1notebook 1 radio	1 bolso	1 radio

Figura 5. Premios

La competencia duró 3 horas, durante las cuales los equipos (de 3 integrantes) tuvieron que resolver un conjunto de problemas. Para los alumnos de la UTN (en cualquier nivel), los mismos debían codificarse en lenguaje C ó C++, y ser subidos al sitio juez (Spoj), para su evaluación. La categoría "Rebeca", para alumnos secundarios, se basó en una consigna que planteaba una situación de enseñanza de un tema particular, para la cual los equipos debían desarrollar una animación que permitiera explicar el mismo de forma didáctico. En este caso, un jurado compuesto por docentes, evaluó las animaciones desarrolladas, en cuanto a idea, potencial y utilización de las facilidades que brinda Rebeca. Se realizó una puesta en común y devolución a los equipos, y se definió un orden que permitió establecer los equipos ganadores.

5. Resultados

- Año 2009: 1er. Encuentro Santafesino de Programación Competitiva. Participaron equipos de la Universidad Nacional de Rosario, Universidad Tecnológica Nacional F. R. Rosario, Universidad Nacional del Litoral, y Universidad Tecnológica Nacional F. R. Santa Fe
- Año 2010: Competencia de Programación en el marco del CNEISI realizado en la UTN F.R.S.F. Equipos participantes de Córdoba, Mendoza y Santa Fe.
- Año 2011: Competencia de Programación Interna de la cátedra de Programación Competitiva.
- Año 2012: Competencia sólo para estudiantes universitarios, apoyo del Ministerio de la Producción de la Provincia de Santa Fe (Sistema de Empresas de Base Tecnológica).
- Año 2013: integración del proyecto de Fundación Sadosky, integración de alumnos de escuelas secundarias con sus docentes. Acompañamiento de empresas de la región con premios "motivadores". Participaron 24 equipos, 5 de escuelas secundarias, y los restantes formados por alumnos de 1ro. a 4to. año de la carrera de Ingeniería en Sistemas de Información, cada uno de ellos con nombres de fantasía como: "Null Exception", "Legendarios Leviatanes", "Tiroteo Multivariable", "Backtracking", entre otros.

El momento de la entrega de premios permitió una reflexión conjunta de todos los participantes, luego de una tarde de trabajo en equipo, que marcó una experiencia diferente dentro de las actividades universitarias cotidianas.

La competencia era de elección personal y grupal, y se resaltó la actitud que es necesaria para emprender este tipo de acciones.

El acompañamiento de los docentes y de los compañeros viendo la competencia desde afuera y

festejando la entrega de premios, pone de manifiesto el compromiso a nivel institucional, y el reconocimiento a todos los participantes.

El esfuerzo voluntario de transitar las instancias previas de preparación y conformación de equipos y la posibilidad de poner en juego en una experiencia concreta conceptos y buenas prácticas de programación, la percepción en primera persona de experiencias que muchas veces los docentes transmiten verbalmente: *para un mismo problema no hay una sola solución, ni siquiera cuando lo resuelven sujetos con preparación similar, la eficiencia depende de cómo se ponderan los recursos considerados, las soluciones más simples son la opción a considerar en primer lugar*, etc.; convierten a la competencia (en palabras de los participantes) en "una experiencia que brinda muchos aprendizajes".

6. Conclusiones

La atracción de aspirantes y la motivación de estudiantes universitarios en curso, requiere decisiones institucionales, participación creativa de docentes y una verdadera tarea de integración. En esta experiencia, se observa una decisión institucional integrando los equipos docentes que trabajan en la Dirección de Acceso (Ingreso), junto con docentes de las asignaturas del Área Programación de la carrera ISI; a esto se suma la articulación con las escuelas medias (a través de programas de Voluntariado Universitario), así como también con la Fundación Sadosky (en su proyecto de apoyo vocacional en TICs) y con empresas que avalan estos esfuerzos.

Los alumnos participaron voluntariamente con entusiasmo, en el proceso preparatorio y en la jornada.

Los ambientes de trabajo evidenciaron el involucramiento y la atención.

Las Figuras 6,7 y 8 muestran imágenes de esa jornada.



Figura 6. Vista de grupos de categorías A, C, D y E



Figura 7. Vista de grupos de categorías B



Figura 8. Entrega de premios

En el año 2014, ya se han desarrollado actividades con varias escuelas, tomando contacto con cientos de alumnos. En la primera jornada de capacitación del programa de la Fundación Sadosky participaron 120 alumnos, lo que permite prever para Tecnomate de este año una mayor participación e integración.

7. References

[1] Cullen, P. *Universidades para el siglo XXI*, EdUTecNe, Buenos Aires, 2009.

2] Opiniones en los medios de comunicación sobre la Falta de Ingenieros:

Clarín, 08/08/10 Opiniones en http://www.ieco.clarin.com/empleos/falta-ingenieros-debate-empresas-universidades_0_313168968.html.

Diario BAE 16/01/12 http://www.cessi.org.ar/ver-noticias-bae-la-industria-del-software-todavia-esta-lejos-de-satisfacer-su-demanda-laboral-843A.

Clarín, 10/09/12. Opiniones en http://www.clarin.com/opinion/suficientes-ingenieros_0_771522968.html.

[3] Rascovan, S. *Los jóvenes y el futuro. ¿Y después de la escuela qué?* Psicoteca. Buenos Aires. 2000

[4] Simulador para robots Lego: https://github.com/sedugri/proyecto. © 2014 GitHub, Inc.

[5] www.daleaceptar.gob.ar/Sadosky

[6] *CONFEDI - Consejo Federal de Decanos de Ingeniería, XXXVII Reunión Plenaria, Mayo 2004*."Proyecto Estratégico de Reforma Curricular de las Ingenierías 2005 – 2007".

[7] González, A.; Madoz, M.C. "Utilización de TIC para el desarrollo de actividades colaborativas para la enseñanza de la programación", *RedUNCI. Red de Universidades con Carreras en Informática - TE & ET 2013.* http://sedici.unlp.edu.ar/handle/10915/27525

[8] Morales Bueno, P. y Fitzgerald V. (2004). "Aprendizaje Basado en Problemas". *Revista Theoria, Vol. 13* ISSN 0717-196X pág.145-157

[9] Kurnia, A.. *Online Judge. Honors year thesis,* School of Computing, National University of Singapore. 2001.

[10] English, J., & Siviter, P.. "Experience with an automatically assessed course".*In Proceedings of the 5th Annual.SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education,* (pp. 168–171).(2000, July)

[11] Cheang B., Kurnia A y otros. "On automated grading of programming assignments in an academic institution". *Computers & Education 41 (*2003) 121– 131.

[12] Tonin N., Bez J. URI. " Online Judge: A New Interactive Learning Approach". Computer Science and Engineering Department, Universidade Regional Integrada, Erechim, RS, Brazil. *Computer Technology and Application 4* (2013) 34-38.

[13] SPOJ. Sphere Online Judge: http://www.spoj.com/

[14] UVa Online Judge: http://uva.onlinejudge.org

[15] http://icpc.baylor.edu/

[16] http://acm.fjnu.edu.cn/

[17] Fundación Sadosky- Vocaciones en TICs http://www.fundacionsadosky.org.ar/Programas-Proyectos/vocaciones-en-tic

La educación popular en la universidad obrera
Dinámicas participativas para aprender a programar en el Paradigma de Objetos

Sanzo, Alfredo
Universidad Tecnológica Nacional (F.R. Buenos Aires) – Universidad Nacional de Quilmes.
Spigariol, Lucas
Universidad Tecnológica Nacional (F.R. Buenos Aires / F.R. Delta)

Abstract

El presente trabajo conforma un estudio del caso sobre la propuesta pedagógica con características propias de la educación popular llevada a cabo por un equipo docente en la Universidad Tecnológica Nacional - fundada originalmente como Universidad Obrera- para la explicación de los conceptos centrales del paradigma de programación orientada a objetos. Se presentan diversas experiencias que se observan en el aula, donde con recursos y estrategias didácticas que combinan actuación, humor, imágenes y metáforas, buscan lograr un mayor protagonismo del estudiante en su proceso de aprendizaje, mantener su atención y motivación, facilitar la comprensión de conceptos técnicos y establecer una buena integración entre el alumno y el docente. Esta experiencia, desde la teoría educativa, se puede encuadrar alineada con las pedagogías críticas, a pesar de la falta de formación formal específica en educación del cuerpo docente y de ser un contexto institucional no habituado a este tipo de prácticas. Bajo un análisis participativo y en concordancia con la palabra de Paulo Freire, se concluye que valorizando aspectos claves como la participación y la motivación, asumiendo el riesgo de incorporar el humor y el juego y fundamentalmente adoptando una posición crítica de creación y recreación continua, se logra una mejora tangible en los procesos de aprendizaje de los estudiantes. Por último, es una invitación a todo docente, especialmente aquéllos de materias "duras", y de todos los niveles de educación formal y particularmente en la educación superior, a intentar poner alguna de estas ideas en práctica y a revalorizar la educación popular como herramienta pedagógica apropiada.

1. Introducción

Siendo cuatro los términos que plantea el título del presente trabajo -leyes de combinatoria mediante- son seis los pares que se pueden armar y son múltiples las tensiones que sugieren

En primer lugar, los explícitos:

Educación-popular. Esta tensión entre la educación, tradicionalmente propia de los sectores privilegiados de la sociedad, y lo popular, peyorativamente asociado a lo inculto, ya ha sido trabajada por numerosos autores y tiene una larga historia de conformación como campo disciplinar propio. De todas maneras, esa identidad ganada, muchas veces signada por la resistencia y la alternatividad, mantiene una difícil vinculación con los sistemas educativos formales. Partiendo de la experiencia y la palabra de Paulo Freire, emblema de esta corriente, se intentará un acercamiento.

Universidad-obrera: Rompiendo con el preconcepto dominante de que el ámbito propio de formación para un obrero lejos estaba de ser la Universidad, reservada a otra clase social, en 1948, en pleno auge del gobierno peronista, se crea por ley la Universidad Obrera Nacional. Con un sistema orientado a estudiantes que debían simultáneamente trabajar en la especialidad y sedes distribuidas en lugares estratégicos del país, su objetivo era formar integralmente profesionales de origen laboral, destinados a satisfacer las necesidades de la industria argentina, otorgando un título de Ingeniero de Fábrica. En 1959 pasaría a denominarse Universidad Tecnológica Nacional, y con los años, aunque fue perdiendo esa impronta fundacional, continúa manteniendo ciertos rasgos particulares. Es difícil precisar cuánto le queda de "obrera" a la UTN y no es ese el objetivo de este trabajo, pero vale la pena al menos recuperar ciertos rasgos característicos, como por ejemplo y a riesgo de estereotipar, señalar cómo lo "frío" de lo universitario -o más aún, de lo ingenieril- fue ganando terreno sobre la idiosincrasia obrera.

Luego, vienen los aparentemente más obvios:

Educación-Universidad: Que la Universidad tiene por esencia un rol educativo no es ningún descubrimiento, ¿pero de qué educación estamos hablando? ¿para qué educa? ¿cómo educa? Sin pretender hacer un análisis del sistema universitario argentino como tal, son interrogantes que se mantienen presentes a lo largo del trabajo.

Popular-Obrera: Categoría difícil la de "popular", con pluralidad de significados, alcances diversos y múltiples planos de análisis. Años atrás probablemente era más simple asociarla al ser obrero, pero en la actualidad tanto por la transformación de la misma estructura productiva que requiere proporcionalmente de menos "obreros" -en su sentido más elemental- como por la resignificación social del carácter identitario del término, se torna más prudente hablar de trabajadores. Más aún en el ámbito de sistemas, donde lo virtual de la materia prima con que se trabaja hace más forzada la identificación.

Por último, los que faltaban:

Universidad-Popular: Quizás sea la más fecunda de las tensiones. ¿En qué sentido lo popular se introduce en la Universidad? ¿por el sujeto que conduce el proceso de enseñanza o por sus destinatarios? ¿relativizando el nivel de sus contenidos o lo exigente de sus evaluaciones? ¿por su potencial revolucionario en tanto cantera de intelectuales orgánicos? ¿planteando un modelo alternativista que discute el carácter profesionalista de la Universidad y construye conocimiento y relativiza certificaciones? Sin desestimar esta multiplicidad, se agudizará una mirada más metodológica de lo popular en cuanto a un modo de ser y actuar, donde, entre otros aspectos, el estudiante -el trabajador- se asume protagonista del proceso en el que está inmerso y no un simple destinatario, o peor aún, un insumo.

Educación-Obrera: Simplemente para completar el esquema, aunque sus principales aristas pueden encontrarse entre los pares anteriores.

En este contexto institucional, el subtítulo presenta más directamente el alcance del trabajo, que propone una sistematización de la práctica de un equipo docente en particular, que encarna las idas de la educación popular en el ámbito de la Universidad Tecnológica Nacional. Principalmente, se recuperan y fundamentan una variedad de recursos y dinámicas, que con una impronta participativa y apelando al protagonismo del alumno en su proceso de aprendizaje, se utilizan para enseñara a programar dentro de los términos del paradigma de la Programación Orientada a Objetos.

.

2. Objetivos

Se busca analizar la propuesta pedagógica del equipo docente de una de las asignaturas troncales de la carrera de Ingeniería en Sistemas de Información, en la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional, para descubrir en ella la presencia de rasgos propios de la educación popular, en un contexto institucional no habituado a este tipo de prácticas.

La materia elegida se denomina "Paradigmas de Programación", está ubicada en el segundo año de la carrera y tiene por temas centrales la programación orientada a objetos, la programación funcional y la programación lógica.

No se busca señalar errores ni elogiar virtudes, sino, desde la recuperación de una experiencia, aportar ideas y criterios valiosos para enriquecer los procesos de enseñanza de la ingeniería en sistemas, por ejemplo valorizando aspectos claves como la participación y la motivación o asumiendo el riesgo de incorporar el humor y el juego.

3. Metodología utilizada

Por la modalidad de la investigación educativa, y con el propósito de interpretar el significado de una experiencia, para llevar adelante lo medular del trabajo se optó por el método de estudio de caso, que permitió analizar situaciones puntuales y reales de enseñanza y aprendizaje, ubicando el fenómeno estudiado en tiempo y espacio, utilizando la observación participante como instrumento privilegiados de recolección de datos. El acento estuvo puesto en la profundización y el conocimiento y no en una generalización de los resultados que fuera por encima de lo concreto de éstos. En este sentido, se asumió una perspectiva cualitativa e interpretativa de la situación educativa estudiada.

Para sintetizar la concepción epistemológica utilizada, cabe mencionar una distinción fundamental entre los distintos métodos: "A diferencia de los métodos cuantitativos que se enmarcan en una concepción positivista, que aplican controles rígidos a situaciones 'artificiales' y en cuya aplicación el investigador intenta operar manteniendo cierta distancia y neutralidad, en los métodos cualitativos se actúa sobre contextos 'reales' y el observador procura acceder a las estructuras de significados propias de esos contextos mediante su participación en los mismos”[1]. En este sentido, tratándose de un proceso tan complejo y multidimensional como es el aprendizaje, evitamos reducir el resultado de la investigación a una mera medición de resultados de aprobación de la materia o realizar comparaciones artificiales entre cursos o materias. No se pretende establecer una afirmación de validez universal ni de extrapolar mecánicamente las conclusiones a cualquier otro contexto de enseñanza, pero sí poder interpretar la significatividad de la propuesta pedagógica y cuál es el aporte que hace al campo de la enseñanza de la ingeniería.

4. Marco teórico

La corriente pedagógica cuyos aportes son más significativos para el presente trabajo es la de la educación popular, que nacida en el marco de las pedagogías críticas, encuentra en Paulo Freire su exponente más emblemático de un grupo nutrido de especialistas en el tema. Partiendo de asumir al estudiante no como un cuenco vacío que debe ser llenado de contenidos sino como portador de saberes y habilidades y como verdadero protagonista de su propio proceso de aprendizaje, son numerosos los aportes de la pedagogía crítica a tener en cuenta: la reconstrucción de la curiosidad, la mirada crítica de la realidad, la importancia de leer el contexto, la interrelación entre teoría y práctica, la capacitación permanente, la actitud de crear y recrear permanentemente, la relación con otros, la interpretación dialéctica del mundo, la comprensión de la educación como arte, las reflexiones acerca de la esperanza, la generosidad, el coraje y el amor que requiere la tarea docente, etc.

Analizando la experiencia de materias iniciales de programación, que combinan conceptos teóricos con elementos prácticos de los propios lenguajes de programación, se perciben numerosas tensiones que atraviesan a los estudiantes: entre la polisemia de un lenguaje natural cargado de sobreentendidos y ambigüedades y la rigurosidad del lenguaje de programación, la emoción del estudiante y la frialdad de la computadora, la frustración de ver que el programa "no compila" y la ilegibilidad de los errores que arroja la máquina, la lógica de sentido común del modo de razonamiento humano y la incuestionable mecánica computacional, la comprensión de una idea y su utilización para plantear una buena solución, la descripción del problema a resolver y la implementación de su solución mediante un programa concreto, la multiplicidad de formas en que las personas interpretamos la información y lo acotado de los tipos de datos con que los lenguajes se manejan, entre otras.

Estas intuiciones son consistentes con los numerosos aportes teóricos desde diferentes corrientes de la pedagogía que permanentemente intentan plantear una interrelación fecunda entre teoría y práctica. Entre ellas, es enriquecedora la perspectiva crítica, que interpreta la relación práctica/teoría desde la acción/reflexión: "Si los hombres son seres del quehacer esto se debe a que su hacer es acción y reflexión. Es praxis. Es transformación del mundo. Y, por ello mismo, todo hacer del quehacer debe tener, necesariamente una teoría que la ilumine. El quehacer es teoría y práctica. Es reflexión y acción. No puede reducirse ni al verbalismo ni al activismo." [2]

El principal problema que se percibe, desde las características específicas de la materia, consiste en el carácter abstracto de los conceptos técnicos de programación que están en juego, que se suma a un cierto preconcepto que traen los estudiantes –construido por años de transitar instituciones educativas y ratificado en su primer año en la universidad- acerca de asociar en una tristemente reiterada tríada la "seriedad" del docente, la "pasividad" del alumno y la "frialdad" de su explicación, que se acentúa conforme a la complejidad del tema.

La convicción que fundamenta este trabajo es que precisamente lo complejo, abstracto y teórico de los conceptos de programación orientada a objetos, requiere de un esfuerzo adicional de parte de los docentes de crear las condiciones que faciliten el proceso de aprendizaje del alumno. En otras palabras, la distancia que separa al estudiante de la programación no es menor ni trivial, aunque tampoco es una dificultad insalvable: requiere de mediaciones pedagógicas.

5. Recursos pedagógicos

Un relevamiento acerca del uso actual de recursos pedagógicos específicos por parte de los docentes en la materias Paradigmas de Programación, sistematizado en cuatro grandes categorías, muestra las siguientes experiencias:

Estrategias áulicas clásicas: Utilizando "lápiz y papel", es de uso frecuente una serie de diagramas que representan los elementos centrales de los programas en cuestión, y naturalmente, la escritura manuscrita del propio código del programa de acuerdo al correspondiente lenguaje. Entre ellos, los diagramas de objetos o de clases.

Recursos Tecnológicos Genéricos: Mediante videos, proyecciones y presentaciones, por citar algunos recursos que también son utilizados en múltiples disciplinas, ciertamente es posible desarrollar los conceptos y ver su aplicación práctica de una manera apropiada y dinámica. Por ejemplo, existen tutoriales y guías de uso de herramientas de desarrollo que existen en formato de video, en las que se combina un audio de parte del docente que explica los pasos a dar, con una captura de pantalla de lo que va sucediendo en la interfaz de la aplicación. Ocasionalmente, se hace uso de proyectores para presentar diapositivas con síntesis de los contenidos, evitando abusar de su carácter de ayudamemoria.

Mediaciones tecnológicas específicas: Herramientas de desarrollo, tales como lenguajes, entornos integrados de desarrollo, editores, recursos gráficos y otras tecnologías disponibles, en especial todo tipo de recursos tecnológicos creados y desarrollados con fines didácticos. En primer lugar, como recurso para enseñar ciertos

conceptos se recurre a los mismos entornos de desarrollo que proveen los lenguajes de programación, en particular Haskell, Prolog y Smalltalk, que son los lenguajes que utiliza la cátedra. Su utilización consiste en hacerlas funcionar sobre el equipamiento correspondiente, desarrollando, ejecutando y probando soluciones a problemas concretos y viendo los resultados que proveen. Por otra parte, un caso que se destaca en la materia es en relación al lenguaje Smalltalk, donde los docentes previamente a utilizarlo con todas las variantes que proporciona, como estrategia didáctica utilizan un software denominado *Ozono*[3], destinado a facilitar el aprendizaje de dicho lenguaje de programación -y de esta manera comprender los conceptos del paradigma de programación orientado a objetos- que fue desarrollado por el mismo equipo docente y está inserto en la propuesta pedagógica que llevan adelante [4].

Recursos Participativos Innovadores: Tomando prestadas ideas de ámbitos típicamente lejanos a la informática, como lo es la educación popular, es interesante ver la utilización de representaciones, juegos y otras técnicas que apelan al movimiento y la participación, como parte del proceso de aprendizaje en el ambiente universitario.

Sin dudas, es sobre esta última categoría en la que se centra el trabajo.

Las dinámicas, juegos y recursos que se ponen en práctica en el ámbito de la asignatura son diversos y su autoría corresponde a diferentes docentes. La circulación de ayudantes por cursos de diferentes profesores, así como las frecuentes reuniones entre todos los docentes, permiten que se haya ido armando un repertorio de recursos, que con matices y variantes en las sucesivas ejecuciones, se reiteran entre los distintos cursos y se recrean año tras año. Para contextualizar, son más de quince los cursos que se dictan anualmente, son unos diez los profesores a cargo de cursos y alrededor de cincuenta los ayudantes que de una u otra manera participan de la cátedra. Ciertamente se pueden descubrir internamente matices y enfoques diferentes y hay docentes más propensos que otros a este tipo de dinámicas, pero a modo analítico, en función de los objetivos del presente trabajo se toma al equipo docente como una unidad y cada uno de los recursos son descriptos de una forma representativa del conjunto. A modo de ejemplo, en el año 2013. Su desarrollo se ubica en momentos clave dentro de las alrededor de diez clases que se dedican a la enseñanza del paradigma de programación orientada a objetos dentro del cronograma de la materia. Mayormente, se utilizan en las primeras clases que se donde resultan más útiles para comprender precisamente la dinámica de funcionamiento del paradigma, aunque hay algunas que por la temática que abordan, se desarrollan sobre las últimas clases. Siendo una materia de modalidad anual, y teniendo reservado el primer cuatrimestre para los paradigmas de programación “funcional” y “lógico” –donde se llevan adelante otras estrategias participativas no descriptas en el presente trabajo- estas técnicas se desarrollan en el segundo cuatrimestre.

A continuación, una explicación detallada de las dinámicas más utilizadas:

5.1. Personificación de objetos

Quienes ya décadas atrás definieron los principios básicos del paradigma de programación orientada a objetos, al elegir los términos con los cuales denominar a los nuevos elementos insinuaron cierta metáfora. Al concepto principal lo denominaron precisamente "objeto" ya que es una entidad computacional, una abstracción, que representa a cualquier cosa, elemento -o precisamente objeto- que existe en la realidad del problema que se quiere resolver. Hasta ahí, no es para destacar su gran creatividad, pero es un término preciso y sencillo. Más interesante es que al concepto que representa la forma en que se requiere la realización de una determinada acción por parte de un objeto se la denominó "mensaje" y al hablar en el contexto de envío de mensajes se lo identifica al objeto como "receptor" del mensaje, estableciendo de esta manera una terminología propia de las ciencias de la comunicación. Además, al conjunto de procesos que debe realizar el sistema, como desde la perspectiva del paradigma se los distribuye entre diferentes objetos, se dice que cada objeto tiene su "comportamiento" y a su vez, que posee "atributos", lo cual sugiere cierta influencia desde la psicología. La propuesta de los docentes consiste en redoblar la apuesta y extender el sentido metafórico de los conceptos, sustituyendo el término "objeto" por "sujeto" y reforzando de esta manera el carácter autónomo, activo y comunicativo de cada uno de ellos. No se trata de plantear al término “sujeto” como un concepto nuevo que pudiera llevar a la confusión con su sentido en otras áreas de sistemas, ni de pretender refundar el paradigma, sino simplemente una comparación con fines didácticos. Por lo tanto, en vez de ver la programación con un conjunto de tareas frías o estáticas, se la presenta como una situación social, donde un "sujeto" le envía un mensaje a otro, y él -porque ya no es "algo" sino "alguien"- de acuerdo a cómo sea su comportamiento le responde o decide a su vez enviar un nuevo mensaje a un tercero. Otra posibilidad es que el mensaje que recibe un "sujeto" provoque una modificación en sus atributos, lo que se expresa como un cambio en su propia "interioridad" o "subjetividad".

Toda esta terminología crea las condiciones propicias para que los docentes puedan realizar una representación

de índole teatral en la que interviene el profesor con sus ayudantes y con la colaboración de estudiantes voluntarios. Cada uno asume el rol de un objeto diferente y la clave es preguntarse y responderse entre ellos de acuerdo al comportamiento propio de cada uno. Se acuerda previamente un sencillo guión como para que la situación tenga cierta lógica y coherencia y se lo pone en escena ante la mirada de los estudiantes. A veces, para captar más la atención y reforzar visualmente la asociación entre objeto y sujeto, se suele utilizar algún tipo de disfraz o escenografía modesta.

5.2. Flechas, hilos y carteles.

Continuando con la lógica de subjetivación de los objetos, habiendo ya incorporado la idea comunicativa de los mensajes -o quizás como introducción a la misma-, esta dinámica permite incluir una complejidad adicional a partir del concepto técnico de referencia. La idea central es que los atributos que posee un objeto son referencias a otros objetos.

Entre los conocimientos previos que tienen los estudiantes a partir de lo aprendido en las materias correlativas anteriores, concretamente en *Algoritmos y estructuras de datos* que aborda la programación estructurada, el concepto que se relaciona con esta idea es el de punteros. Es uno de los temas que resulta más difícil para comprender y generalmente la noción que los estudiantes traen está más asociada a su funcionamiento interno que a su utilidad. Es decir, la preocupación por su mecánica de direccionamiento e indireccionamiento, la validación de los tipos de datos, la asignación y liberación de memoria dinámica, suele dejar en un segundo plano la multiplicidad de aplicaciones que tiene el concepto. Volviendo a la programación orientada a objetos, las referencias, si bien tienen en su implementación interna una mecánica similar a los punteros, para quien hace tareas de desarrollo de sistemas en el marco de este paradigma esto no representa un problema ya que los mismos lenguajes lo resuelven de manera transparente: importa más el "para qué" que el "cómo". En definitiva, abstrayendo lo fundamental del concepto, se trata de algo que hace referencia, que apunta, que señala a otra cosa.

En la dinámica, cada referencia se representa con una flecha, dibujada en un cartel junto con una palabra que la identifica y sostenida por un hilo. Una persona, actuando de objeto, sostiene la flecha tomando un extremo del hilo y el otro extremo termina en otra persona, que hace las veces de otro objeto; la dirección de la flecha indica el sentido de la relación. La forma metafórica de nombrar la referencia es diciendo que un objeto "conoce a" otro objeto. Así, siguiendo un guión actoral acordado previamente, incorporando más actores que hagan de objetos se mantiene la comunicación mediante mensajes como en la dinámica anterior. Ahora, un objeto le envía un mensaje al objeto que conoce mediante la flecha, y se refiere a él con el nombre que está escrito en el cartel. De a poco, las referencias se van multiplicando. Si un objeto tiene un atributo más, pasa a conocer a un nuevo objeto, y el actor ahora sostiene dos hilos con sendas flechas etiquetadas. Una característica importante de las referencias, conceptualmente hablando, es que pueden ser múltiples; asociado a esto, que los objetos no tienen un nombre único sino que son referenciados mediante diferentes identificadores, según cómo cada otro objeto denomine a sus atributos. Resulta muy gráfico mostrar esta situación con flechas que partiendo de diferentes objetos confluyen en el mismo, los identificadores de cada una de las flechas son diferentes, porque representan lo que significan para ellos, es la forma en que cada uno lo conoce. Son múltiples puntos de vista de una misma realidad. Pese a estas diferentes identificaciones de un objeto, todos los demás objetos que lo conocen le pueden enviar los mismos mensajes y él los responde por igual, porque el objeto conserva su comportamiento y no discrimina según quién sea el que se comunica con él.

Una de las situaciones visualmente más interesante es cuando se produce una asignación en un atributo: para representar que el objeto deja de conocer mediante ese atributo al objeto anterior y pasa a conocer a otro, lo que se hace es mover el extremo final de la flecha, de manera que una nueva persona toma el extremo del hilo que antes sostenía la otra. Este desplazamiento deja en claro además cómo no se destruye el objeto anterior al cual referenciaba el atributo, sino que es simplemente dejado de conocer por ese objeto y no quita que otros objetos los sigan conociendo. A su vez, cabe resaltar que el origen y el identificador de la flecha no cambia, cambia su destino. De manera que pese al movimiento, mediante la misma referencia un objeto para a conocer a otro objeto y es de esperar que le pueda seguir enviando los mismos mensajes que le enviaba al anterior. Esta última característica da pie a un nuevo concepto -polimorfismo- para el cual se utilizan otras estrategias pedagógicas.

Lo que Freire afirma respecto que "estudiar es desocultar, es alcanzar la comprensión más exacta del objeto, es percibir sus relaciones con los otros objetos"[5] se aplica doblemente en este caso, como mirada pedagógica del proceso general que llevan adelante los docentes y en este caso en particular, con un significado informático del término "objeto" con el cual seguramente el autor nunca imaginó.

5.3. Mini Director de Orquesta

Otro de los conceptos a abordar, en este caso uno muy interesante en cualquier lenguaje de programación y

en particular en una orientación a objetos, es polimorfismo, cuya idea central es diseñar el sistema para hacer uso de valores diferentes uniformemente, ó mejor dicho, para poder hacer un diseño genérico y que sea sencillo reemplazar y agregar partes específicas del funcionamiento del mismo.. Éste es sin duda el aporte más significativo del paradigma orientado a objetos, y es consecuentemente otro de los conceptos que en la cátedra han asociado a una actividad.

Para realizar esta actividad los docentes suelen llevar diversos instrumentos musicales a clase y disponen un ambiente de objetos nuevamente subjetivizados, donde hay mínimamente tres actores: dos músicos y un director de orquesta. El objetivo es que el director de orquesta de alguna manera produzca música usando uno de los músicos.

Para dejar en evidencia la diferencia entre la resolución de un problema sin aplicar el concepto de polimorfismo y haciéndolo, la dinámica consiste en teatralizar y comparar dos situaciones, donde el resultado final es el mismo, pero la forma en que se lo resuelve es diferente. Es por otra parte interesantes esta distinción, porque el enfoque de los docentes tiende a valorar el proceso, el razonamiento y la lógica puesta en juego en el desarrollo de la solución que el resultado en sí. En materia de desarrollo de sistemas es un tema fundamental, ya que si bien el usuario final de una determinada aplicación informática nunca accede ni necesita conocer cómo está hecho por dentro el código que la hace funcionar, desde la perspectiva del profesional de sistemas es fundamental que el código sea de calidad, con todo lo que ello implica, y en ese sentido el concepto de polimorfismo es fundamental. Pensando las propuestas pedagógicas no sólo en cuanto al tema en sí, sino poniendo en perspectiva la formación de profesionales para un proyecto de país, aunque parezca un detalle, constituye un elemento importante, sobre todo desde la perspectiva de la educación popular que permanentemente contextualiza las situaciones.

Retomando la dinámica, en la primera escena el director de orquesta se ocupa de enviar una serie de mensajes al músico que conoce, de manera de obtener información necesaria para luego indicarle precisamente qué debe interpretar o cómo debe hacerlo. Básicamente, se trata primero de indagar acerca de su naturaleza preguntándole qué es y posteriormente recién pedirle que haga algo determinado. Siguiendo con la actuación, se cambia al músico por otro -lo cual remite a la dinámica anterior de los hilos donde las referencias se modifican dinámicamente-, el nuevo músico recibe las mismas preguntas y como sus respuestas son diferentes, el director actúa diferentemente. Y he ahí el quid de la cuestión: es el director quien dirige el comportamiento de cada músico y quien necesita conocer lo más posible sobre ellos.

La segunda alternativa, superadora, propone enviar un único mensaje al músico (algo similar a “haz lo tuyo” o “interpretar”) y que el músico tenga la responsabilidad de ejecutar lo que corresponda. De esa manera, cuando cambia el músico, el comportamiento cambió sin necesidad de que el director tuviera que prever todos los casos posibles. Nuevamente, el juego se torna ilustrador cuando se eligen nuevos músicos de entre los alumnos (siempre se suele encontrar alguien que toque la guitarra o cante), y se demuestra el poder del polimorfismo con la facilidad de agregar músicos al esquema, evidenciado en la indiferencia del director de orquesta ante la utilización de un nuevo músico. Contando con un guión previo y algunos participantes cómplices (ayudantes, por ejemplo) el resultado es que, por unos minutos y a modo de distensión, se puede escuchar una melodía interpretada por varios a la vez.

Una variante incluye darle diferentes instrumentos al músico y observar cómo el músico puede utilizar la misma técnica para tocar diferentes instrumentos, lo que refuerza la idea de polimorfismo.

5.4. Fanáticos de la camiseta

El concepto central que aborda esta dinámica el de clase. Para que se pueda interpretar el sentido de la dinámica, una rápida aproximación al concepto: una clase es una generalización de los objetos: cuando varios objetos se comportan de la misma manera y tienen los mismos atributos, son de la misma clase de objetos. La utilidad consiste en que para evitar repetir una innumerable cantidad de veces el código con el cual cada objeto responde a los mensajes que recibe, si se trata de objetos de la misma clase, se define una única vez ese comportamiento en la clase y todo objeto, al saber a qué clase pertenece, sabe también cómo comportarse. En otras palabras, continuando con las metáforas personalistas anteriores aunque con una impronta más sociológica, se traduce como que el comportamiento del individuo -el objeto- está determinado por la clase -¿social?- a la que pertenece. Sin entrar en debates políticos o filosóficos sobre la implicancia social de afirmaciones de esta naturaleza y evitando hacer comparaciones que afecten aspectos sensibles de la forma de pensar o vivir de cada uno, el juego propone incursionar a un territorio donde la mística y el folcklore reinante permite que cierta estereotipación resulte simpática antes que ofensiva: el futbol.

El primer paso consiste en pedir que los estudiantes se agrupen de acuerdo al equipo de fútbol del que son simpatizantes y se sienten juntos. Luego de unos minutos de desplazamiento donde quedan formados varios grupos y siempre quedan, algunos más numerosos que otros,

eventualmente alguno con un único integrante, lo cual tampoco es un problema.

Luego, se le pide a cada grupo que se pongan de acuerdo en tres acciones que realiza un hincha típico de ese equipo. Además, se les indica que describan la forma en que el hincha realiza dichas acciones. Pueden ser más de tres o menos, no cambia las cosas; lo que es importante es que sean tareas cuya forma de concretarla sea bien propias y característica del equipo, de manera que cada hincha la asuma como propia sin ningún tipo de discusión. Este es precisamente el punto central del juego que da sentido al concepto de clase: el hincha -el objeto- está compenetrado de tal manera con su equipo -su clase- que se vuelve su identidad y define su comportamiento. Se puede sugerir algunas acciones, como festejar un gol o alentar a los jugadores, ya que si varios grupos eligen la misma acción no es un problema sino todo lo contrario, permite además repasar e integrar el nuevo concepto con el de polimorfismo, aprendido con anterioridad. Por ejemplo, el festejo de un hincha típico de Boca Juniors consiste en gritar fuerte, el de Rosario Central es levantar los brazos y el de Belgrano de Córdoba es agitar una bandera celeste.

Un paso importante es hacer que pasen por escrito en una única hoja por equipo el listado de acciones típicas y la forma en que se las desarrolla y que pongan bien claro el nombre del equipo.

Por último, se disuelven los grupos y los estudiantes regresan a sus lugares originales, de manera que queden todos mezclados. Las hojas con las indicaciones escritas por cada grupo pueden quedar sobre alguna mesa o en cualquier lugar, basta que estén todas separadas.

Habiendo preparado todo lo anterior, sin haber dicho aún una palabra de los conceptos, comienza la representación. El docente le dice a un hincha cualquiera algunas de las tareas, por ejemplo, "Juan festejá". Habiendo hecho previamente otras actuaciones, a esta altura ya el alumno se dio cuenta que se trata de un mensaje y que él en tanto objeto debe reaccionar. Lo nuevo, es que ahora debe recurrir a la hoja de su equipo donde tiene escrito las indicaciones de cómo comportarse, en este caso de cómo festejar, y hacerlo. Entonces, por ejemplo, Juan grita fuerte. Inmediatamente, el docente continua con "Ana festejá", en cuya respuesta la alumna se fija en la hoja de su equipo y actúa: gritará fuerte, en caso que ella también fuera simpatizante de Boca, o levantará los brazos si fuese de Central. Podría suceder que fuera de un equipo -por ejemplo y sin herir suceptibilidades- como Racing que no tiene en sus indicaciones la tarea de festejar, frente a lo cual su respuesta será que no entiende el mensaje. En definitiva, hará lo que esté prescripto en su equipo, y lo hará de la manera que allí está determinada. Lo poderoso de la didáctica es su intuitividad: cuando el alumno recibe el mensaje suele haberse olvidado ya las decisiones tomadas en su grupo. Consecuentemente (y agregando un poco más de humor a la interpretación) debe estirarse a buscar o a pedir la hoja en la que fue escrito ese comportamiento, lo cual representa aún de mejor manera lo que sucede técnicamente en la articulación de un objeto con su clase.

Se continua con varios mensajes similares más haciendo que participen la mayor cantidad posible de alumnos, explorando las diferentes alternativas, cuidando de terminar antes que se vuelva demasiado reiterativo y por lo tanto aburrido, lo que en algunos manuales básicos de dinámicas participativas le llaman "matar el juego antes que muera".

Una precaución importante es no preguntar nunca "de qué equipo sos" sino hacer la pregunta de lo que realmente se necesite, para evitar que una mala comprensión del concepto de clase entre en conflicto con el de polimorfismo. Por otra parte, es recomendable que el docente esté atento a lo que los grupos vayan escribiendo, como para tener más argumento a la hora de preguntar.

Otro aspecto conceptual que esta actividad hace evidente es que -al menos en el esquema original del paradigma de objetos y lo que en el alcance de la materia- un objeto no cambia de clase, así como un hincha nunca traiciona los colores de su equipo, lo cual una vez entrado en la dinámica prácticamente se explica solo. De hecho, una característica deseable de un relato de carácter metafórico -como así también de la fábula o de la parábola religiosa- es que se explique por sí mismo, o con el mínimo posible de claves de interpretación. En este sentido, la "moraleja" es que un objeto, desde que "nace" hasta que "muere", es de una única clase y "vive" como tal. Por este motivo, hay que tener cuidado de que si se modifica el argumento del relato reemplazando el equipo de fútbol por otra cosa, sea algo lo suficientemente estable, como por ejemplo el país de origen. Por lo contrario, si se quisiera aplicar este recurso para enseñar un enfoque alternativo o más flexible del paradigma de objetos donde los objetos pueden cambiar de clase dinámicamente, se deberían elegir categorías tales como la profesión, el lugar de residencia o porqué no -si el clima de confianza con los alumnos es el adecuado- el género.

5.5. Otras dinámicas

Además de las explicadas, que son las que requieren de una preparación mayor para su implementación o que resultan más útiles, hay otras actividades más puntuales o simples que persiguen también un objetivo didáctico en el marco de la asignatura.

Dentro del mismo paradigma de objetos, por ejemplo hay algunos conceptos que permiten establecer analogías

con elementos de la realidad bastante directas que resultan de fácil comprensión por los estudiantes, como las colecciones, que es la forma en que se denominan genéricamente a las estructuras que permiten contener y organizar en su interior secuencias o conjuntos de objetos - y herencia -que remite a la posibilidad de que una clase de objeto posea rasgos propios de otras clases de objetos a los que se considera en la jerga sus "padres". La utilización en clase de un elemento físico, como por ejemplo una mochila, un clasificador de estampillas o algún recipiente para mostrar los distintos tipos de colecciones, son un recurso frecuente, sencillo y que ayuda a fijar un concepto, a destacar con claridad un momento del desarrollo de la cursada, al cual se puede hacer referencia en el futuro con sólo sugerir la imagen.

También, alguna vez se utilizó plastilina para mostrar la composición de objetos y se dibujaron botones en hojas para que los alumnos lo aprieten y así ejemplificar el funcionamiento de evaluación de los bloques de código. Otras veces se recorre a la actuación no necesariamente para explicar un concepto en particular, sino para ambientar e introducir al problema que se quiere resolver de una forma que capte la atención. Otro recurso es hacer preguntas a modo de desafíos y dar premios a quienes los responden correctamente, o plantear problemas que van más allá de lo explicado en clase, para fomentar la curiosidad y la iniciativa.

6. Implicancias pedagógicas

Estas dinámicas teatrales tienen su importancia pedagógica por varios motivos, que sin pretender jerarquizarlos por algún orden en particular, sino a modo de elenco, son los siguientes:

6.1. Participación de los estudiantes.

Si bien generalmente es el profesor y sus ayudantes quienes asumen los roles iniciales como objetos, progresivamente se van incorporando actores entre los alumnos del curso. Quienes participan protagónicamente de la escena, puede que en su primer actuación no terminen de cumplir su papel a la perfección, pero repitiendo la secuencia varias veces -a modo de sucesivos ensayos que van puliendo la actuación- se llega a una escena lograda donde difícilmente se olviden de los conceptos puestos en juego. Aún entre quienes por lo numeroso del curso y lo reducido de los papeles disponibles no puedan actuar, la cercanía con sus compañeros, la posibilidad de ayudarlos soplándole la letra si se olvidan, la "amenaza" latente de ser convocados a actuar en cualquier momento, hace que sean también parte de la situación y sigan con atención el tema. Es como señala Freire al criticar a la *educación bancaria*, que "enseñar no puede ser un simple proceso de transferencia de conocimientos del educador al aprendiz"[6] que supone al estudiantes como un cuenco vacío que debe ser llenado. El permitir que el estudiante sea protagonista de la clase es además una forma de reforzar su convencimiento de que asuma el protagonismo de su propio proceso de aprendizaje. En este sentido, se busca precisamente que los alumnos sean protagonistas de su aprendizaje, valorando su esfuerzo, animándolos y entendiendo su participación como central y no accesoria en el proceso de aprendizaje.

6.2. Motivación.

Además de mantener la atención, lo cual no es menor, este tipo de dinámicas provoca un efecto motivacional significativo. Sin demasiado esfuerzo previo de creatividad humorística -aunque con un poco de preproducción sale mejor-, la situaciones despiertan rápidamente la sonrisa de los alumnos y un ambiente agradable, habiéndose superado las inhibiciones iniciales. A los alumnos le dan ganas de ir a cursar y los niveles de asistencia se mantienen relativamente estables sin tener que recurrir a medidas disciplinarias. Si bien no falta algún alumno al que le resulta un poco desubicado o incluso infantil, que o bien se acerca personalmente a fin de año a comentarlo o su opinión circula informalmente por los espacios intersticiales de la facultad, en términos generales es bien recibido y valorado por los estudiantes que se sienten a gusto. En el otro extremo, hay también quienes reconocen que haber cursado esta materia los reconcilió con la carrera. Como afirma Freire "Si el estudiar no fuese para nosotros casi siempre una carga, si leer no fuese una obligación amarga que hay que cumplir, si por el contrario estudiar y leer fuesen fuente de alegría y placer, de la que surge también el conocimiento indispensable con el cual nos movemos mejor en el mundo, tendríamos índices que revelarían una mejor calidad en nuestra educación."[6]

6.3. Clima de confianza.

Sumado a lo anterior en tanto factor motivacional, el ambiente agradable predispone de otra manera para que los alumnos se desenvuelvan con confianza, ya sea para preguntar dudas, opinar o para expresar sus inquietudes. En otras palabras, habiendo ya actuado en público o jugado con sus compañeros ¿qué vergüenza le puede quedar que lo inhiba a preguntar delante de todos? Ayuda a crear un vínculo entre alumnos y docentes mucho más cercano, que no anula la asimetría constitutiva de la relación docente-alumno, sino que permite una interacción más fluida que facilite los mutuos aprendizajes. "Las relaciones entre educadores y educandos son complejas, fundamentales, difíciles, son

relaciones sobre las que debemos pensar constantemente. Y bueno sería, además, que intentásemos crear el hábito de evaluarlas o de evaluarnos en ellas también como educadores y educadoras."[7]. El equipo docente asume el desafío de reconocer a la enseñanza es un proceso complejo donde es fundamental la comunicación entre docente y alumno, y entender al aprendizaje como apropiación de sentidos y significados de la realidad, mediante la experimentación, la escucha, el diálogo, reconociendo aportes ya existentes y generando nuevos aportes. La formulación de estrategias específicas para cada concepto relevante de la materia parte de entender el propio rol docente como facilitador del proceso de aprendizaje, favoreciendo la comunicación y participación de todos. Para esto, dominar los contenidos específicos y hacerlo de manera planificada y organizada es condición necesaria aunque no suficiente.

6.4. Articulación teoría práctica.

Las dinámicas planteadas inherentemente condesan una aspecto teórico y uno práctico. Las propias tramas argumentales, sean músicos, hinchas de fútbol, golondrinas o personas, son situaciones prácticas de aplicación de los conceptos teóricos. A su vez, generalmente a continuación de las representaciones o a veces incluso simultáneamente con la colaboración de algún ayudante, ya sea escribiendo en el pizarrón o con una computadora y un proyector, se van plasmando los ejemplos mismos de código, en el lenguaje de programación correspondiente, para tener una articulación más fluida entre teoría y práctica. Es la forma que la cátedra ha encontrado de asumir uno de los postulados básicos de la educación: la relación conflictiva entre teoría y práctica, no como una dicotomía, sino como interrelación fecunda que da lugar a nuevos aprendizajes.

7. Conclusiones

Los resultados, difíciles de cuantificar, se pueden percibir desde diferentes enfoques. Desde lo cuantitativo, un análisis estadístico de porcentajes de aprobados o promedio de notas no es indicador, ya que los criterios de aprobación han ido cambiando respecto de años anteriores. Por otra parte, en un análisis cualitativo, la constatación que hacen los docentes acerca de la mayor complejidad de los ejercicios que se resuelven en clase, de los trabajos prácticos que se realizan y de los exámenes que se plantean, como así también la naturaleza de las consultas y errores reiterados, son indicadores más claros de la significatividad de los procesos de aprendizaje, que hacen posible dicho aumento de complejidad sin perjuicio de la aprobación.

Desde la observación participante como herramienta metodológica privilegiada, se constata que se genera un buen clima de participación, de confianza, de motivación y de trato cercano con los estudiantes. Es evidente que el desarrollo de la propuesta curricular que lleva adelante la cátedra presenta rasgos claramente diferentes a otras asignaturas de la carrera.

Desde la mirada de los docentes, también es clara la intencionalidad que persiguen y los criterios pedagógicos que ponen en práctica, y aunque los docentes en su mayoría no cuentan con una formación profesional desde las ciencias de la educación, sino que son egresados y estudiantes de la misma carrera de Ingeniería en Sistemas, en el análisis de su práctica se descubren en acción varios de los principios de la educación popular y las pedagogías críticas.

Otro elemento interesante que da cuenta del impacto de la propuesta de la materia no sólo del aprendizaje mismo de sus contenidos sino de la motivación y del compromiso que genera en los estudiantes en relación a la Universidad, es la cantidad de alumnos que habiendo terminado de cursar la materia quieren sumarse como ayudantes de la cátedra. De hecho, la cantidad mencionada de unos cincuenta jóvenes en calidad de ayudantes en el año 2014, -de los cuales una escasa proporción son rentados y hay algunos que incluso no pudieron ser nombrados ad-honorem por estar a la vez en otras asignaturas u otras dificultades administrativas-, no contempla a quienes han pasado por Paradigmas de Programación y ahora están como docentes en otras materias. Analizar sus motivaciones o la propia dinámica formativa que la cátedra tiene para sus ingresantes ameritaría otro trabajo, pero en líneas generales, son muchos los que a partir de esta experiencia quieren incorporarse a la carrera docente, a seguir investigando y profundizando.

Además de los diálogos informales que se sostienen entre alumnos y docentes durante la cursada de la materia, al final de cada curso los docentes plantean un cuestionario a modo de encuesta donde consultan a los estudiantes acerca de la materia en general, y en particular hacen referencia explícita a la dinámica utilizada, al clima de la clase y a la comprensión de ciertos temas. Dichos diálogos informales y encuestas, más los análisis docentes clase a clase, más las devoluciones orales formales para las que se hace espacio, más la compartición de experiencias puntuales entre docentes, son herramientas necesarias pero no suficientes. El enfoque sinérgico guiado por la actitud autocrítica permanente es el verdadero factor clave para crear y recrear las dinámicas y recursos pedagógicos empleados año tras año.

En concordancia con lo anterior, es destacable valorar que estas propuestas que se llevan adelante desde hace un tiempo no surgieron espontáneamente, sino que son fruto

de repensar lo ya pensado, de revisar permanentemente las posiciones desde las cuales se lleva adelante la práctica docente. Así, para reconstruir los caminos de la curiosidad de los alumnos, primero los docentes tuvieron que asumir el riesgo de su propia curiosidad y estar abiertos a una mirada crítica. Desde allí es que se promueve en los alumnos una actitud crítica y cuestionadora respecto de lo que los mismos docentes plantean y evitando que lo asuman ingenua y pasivamente. Afirma Freire: "como preparación del sujeto para aprender, estudiar es en primer lugar un quehacer crítico, creador, recreador"[8].

Por último, teniendo en cuenta que la programación orientada a objetos es una temática indispensable para cualquier carrera universitaria de sistemas o informática, la experiencia se demuestra directamente replicable en otras instituciones. Aún más, considerando que la materia se encuentra dentro de la rama más dura del temario de sistemas, consideramos desmitificada una trillada traba a la replicabilidad de estas ideas en un contexto académico más amplio.

Con el objetivo de persistir la intención motivadora de la retórica utilizada para introducir el presente trabajo, para cerrar, dos apreciaciones de Adrián Paenza de su conferencia ante la Unión Matemática Internacional, Corea 2014, en ocasión de su designación como mejor divulgador matemático del mundo.

Refiriéndose al problema de actualización docente: “Entonces ¿cómo resolvemos el problema? Con lo que yo llamo educación horizontal. En vez de una relación vertical, donde los maestros están arriba y los alumnos abajo –el mismo sistema de yo soy el que sabe y ustedes son los que no–, establezcan una relación horizontal. Aprendemos juntos. Hay que tragarse el orgullo, como un padre que aprende con los hijos, cosa que he visto. Si hay alguien que sabe algo, que simplemente venga y comparta lo que sabe.”[9]

Y refiriéndose a las matemáticas -aunque válido para la programación también- Paenza agrega "tiene todas las herramientas para demostrarles que se puede jugar, que van a seguir jugando. Como los magos, ¿qué hacen ellos con los chicos y también con los adultos? Los cautivan, los desconciertan, suscitan su interés, y ellos se quedan asombrados”[10]. Y el asombro es probablemente el ingrediente que mejor sintetiza el aporte original de la propuesta pedagógica de Paradigmas de Programación: quien acepte el desafío de recrearla para su propio ámbito, lo hará a su manera, de acuerdo a sus convicciones y sus posibilidades, pero es de esperar que mantenga esa chispa mágica que desconcierta y cautiva a los estudiantes.

8. Referencias

[1] Vasilachis de Gialdino, I. . *Métodos Cualitativos. Los problemas teórico-epistemológicos*; Ed. Centro Editor de América Latina; Buenos Aires. 1992

[2] Freire, P. *Pedagogía del oprimido*. Bs.As., Siglo XXI. 1973.

[3] Griggio C., Leiva G., Polito G., Decuzzi G., Passerini N. *A programming environment supporting a prototype-based introduction to OOP*, IWST '11 Proceedings of the International Workshop on Smalltalk Technologies, ACM New York, NY, USA 2011.

[4] Spigariol, L. y Passerini, N., "Enseñando a programar en la orientación a objetos". *1er Congreso Nacional de Ingeniería Informática y Sistemas de Información*, Córdoba, 2013.

[5-8] Freire, P. *Cartas a quien pretende enseñar*. México, D.F.: Siglo Veintiuno Editores. 1994.

[9-10] Paenza, A *La puerta equivocada*, en diario Pagina/12 edición del día 24 de agosto 2014.

Definición y medición de indicadores específicos para aulas virtuales de apoyo al dictado presencial

Marcela Andrea Vera
Universidad Nacional del Litoral, Facultad de Ingeniería Química
Santiago del Estero 2829 (3000) Santa Fe, Argentina
Universidad Católica de Santiago del Estero, Dirección Académica Rafaela
Bv. Hipólito Irigoyen 1502 (2300) Rafaela, Argentina
marcelaandrea.vera@gmail.com

Lucila Romero
Universidad Nacional del Litoral, Facultad de Ingeniería y Ciencias Hídricas
Ciudad Universitaria. Ruta Nacional N° 168 Km 472,4 (3000) Santa Fe, Argentina
Universidad Católica de Santiago del Estero, Dirección Académica Rafaela
lucila.rb@gmail.com

Abstract

En los países que desarrollan procesos de reformas e innovaciones en el ámbito educacional, una de las preocupaciones centrales es el tema de la "calidad". Cada modelo de evaluación de calidad, para su evaluación centra su atención en diferentes aspectos de la organización educativa y de su proceso. El objetivo común de los modelos es medir el grado en que una organización alcanza el nivel de calidad acorde con el modelo aplicado.

La educación a distancia (EaD) y el b-learning se encuentran en un momento de auge, debido a la extensa y variada oferta académica que proveen, por lo cual es fundamental definir un modelo adecuado de evaluación de la calidad, que brinde las herramientas necesarias para la mejora continua de sus servicios, pensados estos como agentes de cambio en una sociedad que requiere la democratización del conocimiento.

La utilización de un modelo de evaluación de la calidad adecuado permite analizar en forma estructurada la situación de la institución y proponer los procesos de mejora necesarios [1].

En este trabajo se muestra:

- *La definición de indicadores específicos para aulas virtuales, que permitan obtener medidas de desempeño de las propuestas de b-learning.*
- *Los resultados de las mediciones realizadas durante un cuatrimestre, en el cual se implementaron aulas virtuales de apoyo al dictado presencial en la carrera de Ingeniería en Informática.*
- *Análisis comparativo de dos plataformas educativas.*

1. Introducción

Es fundamental, evaluar el impacto de las TIC, de manera de poder definir "buenas prácticas" en el uso de la tecnología en la enseñanza universitaria. La evaluación es el mecanismo que permite conocer el verdadero impacto de las TIC, y además será la garantía de mejora de la calidad educativa [2].

En nuestro país, no existe un modelo específico que atienda las particularidades de la modalidad de EaD y b-learning, y para obtener la acreditación se utilizan los modelos existentes, generados y probados para la educación presencial. Estos, no permiten evaluar las particularidades de la modalidad, y además no existen indicadores específicos que colaboren en la medición y evaluación de desempeño de estos nuevos modelos de enseñanza-aprendizaje.

La situación referida, implica que las implementaciones de plataformas virtuales de apoyo al dictado presencial, se realizan en muchos casos de forma intuitiva, sin contar actualmente con experiencias previas, ni tampoco modelos pedagógicos específicamente diseñados para esta situación particular.

La realización de mediciones periódicas, de manera sistemática, que permitan generar métricas específicas, es parte fundamental de la generación de un modelo completo para las instituciones universitarias que brindan

herramientas de apoyo al dictado presencial dentro de la variada oferta de las plataformas virtuales.

La medición asigna números o símbolos a atributos de entidades en el mundo real. Para lograr esto, se requiere un modelo de medición que abarque un conjunto consistente de reglas. Una medida proporciona un indicio cuantitativo de la extensión, cantidad, dimensión, capacidad o tamaño de algún atributo de un producto o proceso. La medición es el acto de determinar una medida.

Una métrica relaciona en alguna forma las medidas individuales (por ejemplo, el tiempo promedio de permanencia en el aula por semana de los alumnos).

Un indicador es una métrica o combinación de métricas que proporcionan comprensión acerca del proceso de aprendizaje, el proyecto de enseñanza-aprendizaje o el producto de apoyo al dictado (Aula Virtual) en sí. Un indicador proporciona comprensión que permite a los decisores ajustar el proceso, el proyecto o el producto para hacer mejor las cosas, y que estas sigan los objetivos planteados por la institución educativa [3].

2. Dimensiones

Encontrar inconsistencias entre los componentes básicos de una institución universitaria o de educación superior en una determinada comunidad, constituye el punto de partida y la guía precisa para tomar decisiones optimizantes en una dirección específica.

En consecuencia, se propone agrupar y ordenar los indicadores en tres grandes grupos: indicadores de funcionalidad, de eficacia y de eficiencia:

Funcionalidad: La determinación de los indicadores de funcionalidad, aplicables fundamentalmente a los niveles de institución y de sistema universitarios, plantea dificultades en las sociedades pluralistas y democráticas, dados, por un lado, la diversidad de valores, aspiraciones y necesidades sociales que coexisten en ellos y, por otro, la falta de acuerdo sobre las metas educativas deseables. En todo caso, existen técnicas para determinar los niveles de aceptación de los valores, metas y productos de las instituciones universitarias, base principal de los indicadores de funcionalidad.

Eficacia: La determinación de los indicadores de eficacia se apoya fundamentalmente en la evaluación del producto, tomando como instancia de referencia las metas y objetivos del sistema o de la institución.

Eficiencia: La determinación de los indicadores de eficiencia se apoya fundamentalmente en la valoración de los costos de todo orden (personales, temporales, sociales, materiales, económicos, renuncia a otros logros, etc.) que suponen los resultados obtenidos. Se trata, pues, de expresar la relación medios-logros y su dificultad radica también en el carácter elusivo y ambiguo de estos conceptos en el ámbito universitario [4].

Adaptar los indicadores convencionales a la modalidad b-learning implica tener en cuenta algunos efectos posibles. Algunos de ellos pueden ser transferidos, pero exigen juicios e interpretaciones específicos. En otros casos, se requieren indicadores específicos para la modalidad de enseñanza a distancia. Por esto agregamos a estos, tres características específicas de la modalidad: disponibilidad, información e innovación.

Disponibilidad: Representa la coherencia entre los objetivos propuestos por la institución y los recursos materiales, humanos y económicos de los que puede disponerse al iniciar el proceso.

Información: Coherencia existente entre los resultados obtenidos y las propuestas de mejora. Toda información debe difundirse de manera adecuada, de manera de que resulten indicadores reales y conocidos por la comunidad involucrada. De nada sirve tener un excelente modelo de evaluación si los indicadores no son informados a los actores involucrados.

Innovación: Coherencia entre el listado de mejoras precisas, y la decisión de innovar. El modelo debe proponer nuevas metas, nuevos proceso y nuevas entradas que ataquen los puntos débiles que se detectan [5].

3. Indicadores generados

Como primer parte de este proyecto de investigación, se generaron las aulas virtuales para las materias dictadas en el primer y segundo cuatrimestre del año 2014, en el ámbito de la Universidad Católica de Santiago del Estero, Departamento Académico Rafaela.

Para poder realizar un estudio comparativo, se generaron estas aulas, en dos plataformas distintas: E-ducativa y Moodle, como se indica en la Tabla 1.

Tabla 1: Correspondencia de materias y plataforma educativa

Materia	Plataforma
Base de datos II	E-ducativa
Administración de Proyectos	Moodle
Trabajo Final	Moodle
Base de Datos I	E-ducativa
Ingeniería del Software	Moodle
Métodos Ágiles de Desarrollo de Software	E-ducativa

Las aulas generadas, desde el primer día de dictado estuvieron activas, y durante el transcurso de la primera semana se asocio a todos los alumnos inscriptos, de forma tal, que la participación e interacción fuera natural desde el principio del periodo lectivo.

Se incentivo la participación constante, mediante la generación de foros de bienvenida y actividades iníciales que lograron en el alumno un sentido de pertenencia y pertinencia, lo cual permite plantear actividades de manera periódica.

Principalmente se procede a adaptar el material de estudio para compartirlo con los alumnos en las plataformas. Se busca que esta adaptación no sea la simple "subida" de archivos digitalizados al entorno educativo, los mismos que se utilizan en el dictado habitual. Se busca repensar los materiales de manera de obtener resultados superadores optimizando los recursos tecnológicos de los que se dispone [6].

Para ello se diseña el material pensando en que sea utilizado por los alumnos sin la asistencia del profesor, como material de apoyo a las clases presenciales. En este caso el material debe ser lo suficientemente asistido, conteniendo la explicación detallada que el docente considera apropiada para avanzar en el proceso de enseñanza-aprendizaje.

Como práctica principal, se considera la utilización de objetos de aprendizaje (OA). Investigaciones sobre estándares para la formalización de material educativo emergen y evolucionan constantemente pero en gran medida convergen en la definición de objetos de aprendizaje [6] para la optimización de los procesos en la creación de materiales para desarrolladores de contenidos educativos y diseñadores de materiales de enseñanza.

En este contexto pedagógico-tecnológico, la evaluación del proceso de enseñanza-aprendizaje se convierte en un factor clave. Basado en la evaluación, el aprendizaje con la utilización de las nuevas tecnologías puede enriquecerse con una retroalimentación de calidad a través de una enseñanza personalizada, convirtiendo a este proceso en una experiencia altamente interactiva basado en tecnologías web. En este sentido, se hacen imprescindibles estrategias tendientes a mejorar los procesos de conceptualización, diseño, planificación y ejecución de las actividades de evaluación donde se controla el proceso de enseñanza en contextos de trabajo distribuidos. Los objetivos de aprendizaje que son definidos por un profesor, deben evaluarse en uno de estos dos sentidos: para la clasificación de los alumnos y para mejorar la calidad del curso en futuros dictados [7].

Se plantean evaluaciones on-line de seguimiento, de manera quincenal, que a partir de cada unidad temática de la materia finalizada, les permite a los alumnos y docentes evaluar los conocimientos adquiridos, puntos a revisar, comportamiento del grupo, etc.

La complejidad de estas evaluaciones está planificada, partiendo al principio del dictado con evaluaciones sencillas, con dos oportunidades de realización por alumno, sin penalización, y a partir de que estos se familiaricen con el aula y el formato de evaluación, generar propuestas más complejas que sean vistas como desafíos por parte de los alumnos, de forma tal, que no se pierda el interés en el instrumento.

La planificación detallada y periódica de los contenidos de las aulas, permite además verificar constantemente la participación, entusiasmo y resultados de la implementación.

Se generar por cada uno de los temas dados, diferentes foros de debate, con preguntas disparadoras de reflexión y/o investigación, donde los alumnos son los motores que impulsan y gestionan los temas de interés.

A partir de la efectiva utilización de la plataforma, deseamos mejorar el dictado presencial, usando el aula virtual como un espacio que permite continuar el proceso de enseñanza-aprendizaje de forma ubicua y mediada por tecnologías que son amigables y naturales para nuestros alumnos.

La ubicuidad nos permite trascender el horario y lugar de dictado, y continuar este proceso sin perder el contacto y apoyando el proceso de aprendizaje del alumno.

Se generan espacios colaborativos, en los cuales todos los alumnos participan de manera proactiva, generando propuesta innovadoras, originando discusiones totalmente enriquecedoras, en las que, en muchos casos, no necesitan la mediación del docente/tutor, sino que ellos mismo autogestionan el proceso de aprendizaje.

A continuación, en la tabla 2, se muestran algunas de las métricas ya definidas.

Tabla 2: Indicadores definidos

Métrica	Descripción	Fórmula
Dedicación	Cantidad promedio de horas por semana de participación en el aula virtual por alumno	$Promedio\ Hs\ Dedicadas = \sum \frac{Hs\ semana\ total}{total\ alumnos}$
Participación en foros	Porcentaje total de alumnos que participan en los foros	$\%Part.Foros = \sum \frac{N°\ particip\ foros}{Cant\ de\ foros * total} * 100$
Alumnos regulares	Porcentaje de Alumnos que regularizan la materia	$\%Regulares = \frac{Total\ Regulares}{Total\ Alumnos} * 100$
Alumnos promocionados	Porcentaje de Alumnos que promocionan la materia	$\%Promocionados = \frac{Total\ Prom.}{Total\ Alumnos} * 100$
Actuación académica	Promedio notas en evaluaciones online	$Promedio\ Notas = \sum \frac{Notas\ obtenidas}{Cantidad\ evaluaciones}$

A partir de estas métricas, y otras que están en proceso de definición, se ha comenzado la definición de indicadores, que nos permitan medir y controlar la pertinencia y cumplimiento de los objetivos planteados tanto a nivel institucional, como también en cada cátedra.

4. Valores obtenidos

A continuación se muestra un resumen de las materias que han finalizado el cursado durante el primer cuatrimestre de 2014, y sobre las cuales ya se han obtenido resultados en las mediciones realizadas.

Tabla 3: Resultados obtenidos

Métrica	Base de datos II	Administración de Proyectos
Dedicación	1,5 hs x semana	1,45 hs x semana
Participación en foros	11%	18%
Alumnos regulares	83,333%	75%
Alumnos promocionados	8,33%	25%
Actuación académica	60,83% (Aprobado)	66,39%

Las mediciones realizadas, nos permiten revisar la actuación y cumplimiento de los objetivos planteados al inicio del dictado de las materias.

Estos valores, serán significativos, cuando podamos compararlo tanto con los anteriores periodos de dictado de la materia, como así también en posteriores dictados, para verificar las mejoras en los indicadores de actuación académica y participación de los alumnos en el Aula Virtual.

Generar una propuesta pedagógica que favorezca la participación activa de los alumnos en el proceso de enseñanza-aprendizaje, como así también el aprendizaje colaborativo y permita incentivar en estos el auto-aprendizaje, es el objetivo primero de la implementación de propuestas de semipresencialidad (b-learning), como complemento a la educación presencial. Este objetivo requiere propuestas que cumplan con pautas de calidad que además deben ir en concordancia con el plan estratégico de cada institución educativa.

La definición de métricas especificas, que nos permitan medir la coherencia, funcionalidad, disponibilidad, eficiencia y eficacia entre otras características permitirá evaluar y corregir errores o problemas a tiempo, generando en la institución una fuerte cohesión en el cumplimiento de los objetivos estratégicos.

5. Retos de la EaD y b-learning en nuestro país

Muchos proyectos de implementación de plataformas educativas, como apoyo al dictado presencial, aún tropiezan con resistencia y muchas veces, las propuestas no ofrecen la calidad mínima que las instituciones educativas exigen, como así también la sociedad [8].

Los objetivos que se deben plantear las propuestas de b-learning son:

- Planificación estratégica: es fundamental en toda propuesta educativa, y en estas propuestas especialmente requiere una evaluación rigurosa y exigente.
- Acceso ubicuo: debe explotarse el hecho de que se rompen las fronteras espaciales y de tiempo, ya que todos los actores pueden participar en cualquier momento y lugar, solo accediendo al aula, para participar en el proceso de enseñanza-aprendizaje.
- Individualización-colaboración: estas nuevas tecnologías permiten el trabajo grupal, aprendizajes cooperativos y colaborativos. Equilibrar las dos vertientes de la formación del alumno (individual y grupal), nos brinda una apuesta personalizada de gran valor.
- Adaptación de la institución a los requerimientos tecnológicos. De forma constante, se debe ajustar el modelo de la institución para integrarlos a las nuevas tecnologías, siempre pensando que las tecnologías deben estar al servicio del modelo pedagógico.
- Innovación y superación de la resistencia al cambio. En muchos casos existirán resistencias dentro de la institución, y también docentes plantearán resistencias a estos nuevos modelos pedagógicos. Es un reto modificar estas resistencias.
- Actualización permanente de los contenidos. Las tecnologías no nos permiten quedar en un estado estanco, se debe fomentar la innovación en momentos en que el saber crece a pasos agigantados.
- Atención a las necesidades reales de educación y formación. Atender las demandas y necesidades sociales en cuanto a la formación de nuevas capacidades por parte de los profesionales es fundamental, tanto en la formación inicial como en la formación continua.
- Impulsar los grados de interacción. Generar materiales de calidad que incentiven la interacción de todos los actores del proceso: docentes, alumnos, tutores, gestores.
- Capacitación docente. Las nuevas implementaciones de aulas virtuales requieren nuevas capacidades de parte de los docentes, que deben realizar una formación de calidad, de forma de realizar un uso efectivo y eficaz de este nuevo modelo pedagógico.

- Trabajo en redes institucionales y personales. Las instituciones deben comprender el valor de conformar redes institucionales con el fin de potenciar la inteligencia colectiva.

6. Análisis comparativo de las plataformas educativas

Para poder evaluar y comparar dos plataformas de e-learning, se generaron aulas para complementar el dictado presencial en dos plataformas conocidas:

- E-ducativa.
- Moodle.

Se implementaron materias en estas plataformas, con iguales características y al final del dictado se realizo una encuesta a los alumnos para detectar la evaluación que ellos realizan de ciertos ítems específicos que se implementaron.

Se pidió a los alumnos que evaluaran los siguientes ítems:

1. Contenidos:
    a. Estructura.
    b. Distribución y temporización.
    c. Claridad.
    d. Nivel de profundización.
    e. Información: nivel de aprendizaje mediante autoevaluaciones, test, etc.

Los resultados obtenidos de esta evaluación, en la plataforma E-ducativa, sobre 10 alumnos son los que se muestran en la Imagen 1.

Luego en la Imagen 2, se muestran los resultados obtenidos en Moodle, sobre los mismos puntos evaluados.

Podemos ver en estos resultados, donde se muestran únicamente los referentes a los contenidos, que los alumnos han evaluado de forma satisfactoria la implementación de las aulas virtuales, que les permitieron participar de forma familiar y sencilla para ellos y que les permitió construir el conocimiento, generando saberes a partir de la participación activa, y una actividad mediada por las nuevas tecnologías.

Se evaluaron además, otros conceptos que consideramos importantes en la implementación del aula virtual, se resume en la tabla 3 la totalidad de los conceptos encuestados.

Tabla 3. Conceptos evaluados en la encuesta final

COMUNICACIÓN	Grado de importancia de la interacción con los compañeros que ayudó en su proceso de aprendizaje
	La rapidez de respuesta por parte del profesor/tutor a sus dudas (en un plazo de aproximadamente 48 horas).
	El grado en que se ha sentido guiado por el profesor en el proceso de aprendizaje de la asignatura.
	El nivel de trabajo colaborativo entre alumnos fomentado por el profesor.
	Su motivación hacia el curso.
	La implicación de los estudiantes en el desarrollo de la asignatura.
	El nivel de interacción con los otros alumnos en los foros de debate.
	El nivel de interacción con el profesor en los foros de debate.
CONTENIDOS	La estructura de los contenidos didácticos.
	La temporización y distribución de los temas.
	La presentación general de los contenidos didácticos.
	La claridad en la exposición de los contenidos didácticos.
	El nivel de profundización de los contenidos didácticos
	El nivel de información que se le ha proporcionado sobre su aprendizaje de la asignatura a través de los ejercicios de autoevaluación, test, etc.
ACTIVIDADES	El número de actividades programadas (foros, trabajos, etc.).
	El tipo de actividades propuestas.
	La contribución de las actividades al aprendizaje comprensivo.
	El grado en que las actividades han facilitado el aprendizaje colaborativo entre los compañeros de la asignatura.
EVALUACIONES	El número de evaluaciones en línea

ONLINE	
	Grado de Complejidad
	Utilidad en el aprendizaje
	Utilidad como herramienta de autoevaluación
USABILIDAD	Acceso al aula
	Descarga de contenidos
	Acceso a Foros
	Chat

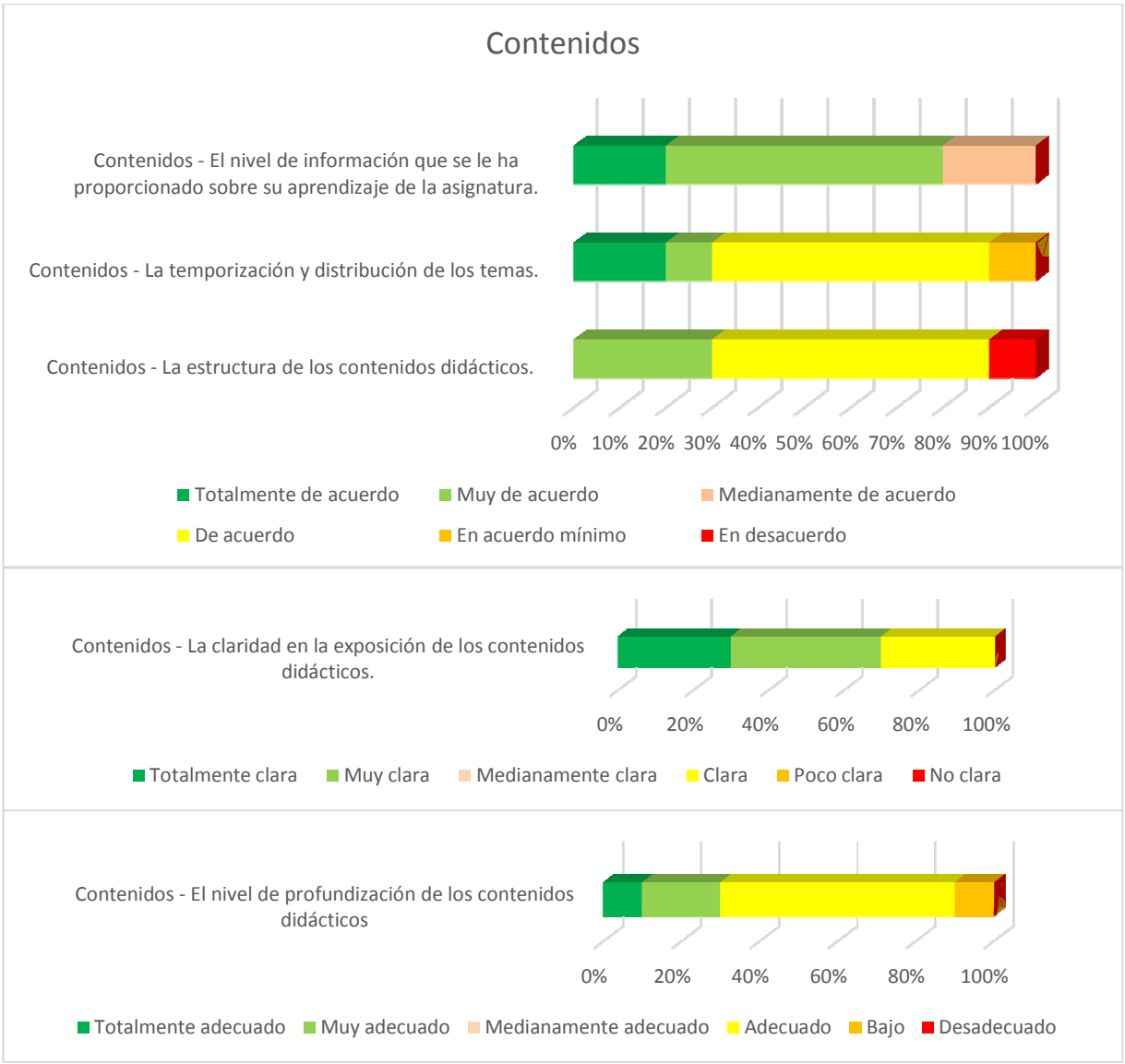


Imagen 1: Resultados obtenidos en E-ducativa.

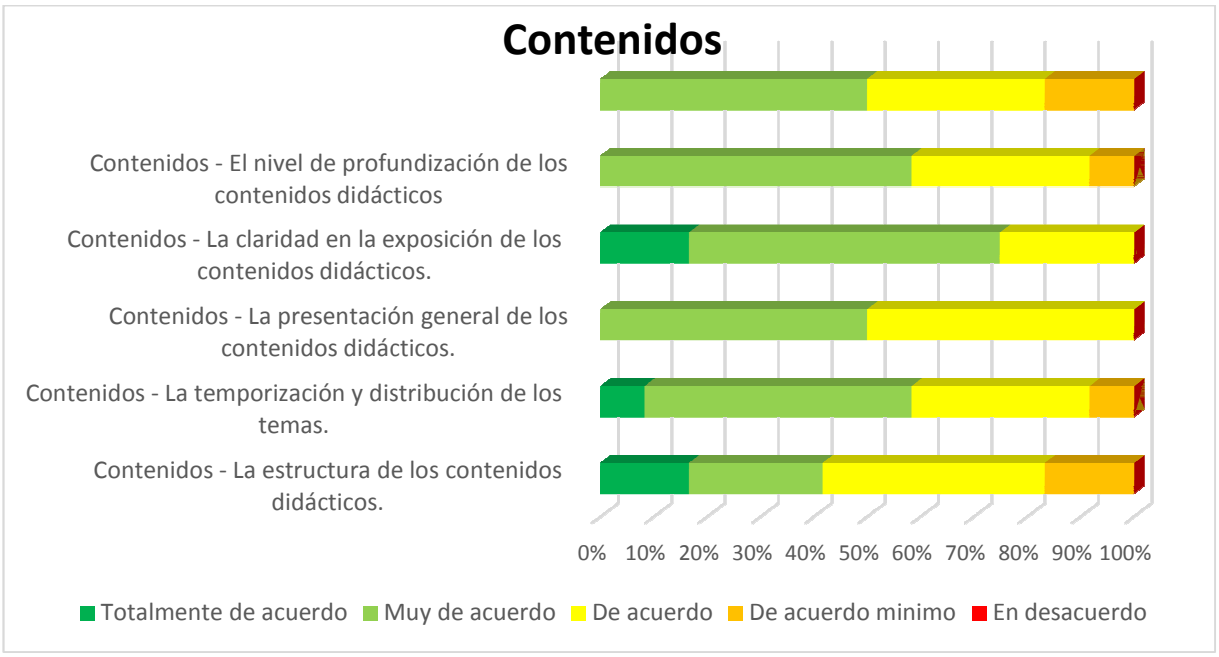


Imagen 2: Resultados obtenidos en Moodle.

7. Conclusiones

En este proyecto I+D, en el que ya han transcurrido quince meses, hemos logrado:

- Generar en los alumnos, un sentido de pertenencia del aula virtual, reconociéndolo como una extensión del dictado presencial.
- Participación natural en las actividades planteadas, enriqueciendo los contenidos dictados, y brindando además pautas de temas de mayor interés al docente.
- Generar un proceso de aprendizaje ubicuo, donde los alumnos se apropian del aula, y socializan sus conocimientos con los demás compañeros, en un ámbito para ellos natural, en el cual todos participan tomando de manera alternativa los diferentes roles.
- Se realizaron mediciones sobre las diferentes plataformas, que nos permitieron generar las métricas necesarias, y realizar un estudio comparativo acerca de la actuación académica de los alumnos en un dictado b-learning, y el dictado solo presencial de los años anteriores.
- Se generaron encuestas específicas que permitieron a los alumnos puntuar las dos plataformas de trabajo, de forma que nos permita realizar un análisis comparativo cualitativo.

Los alumnos fueron muy críticos en el aspecto de disponibilidad de la plataforma. Requieren que el aula se encuentre disponible para el trabajo las veinticuatro horas, los siete días de la semana. Con respecto a este ítem, evaluaron de forma mucho más favorable la estabilidad y disponibilidad de Moodle, sobre E-ducativa.

Este proyecto ha cumplido la mitad de su periodo, y notamos con gran entusiasmo, una mejora en el dictado de las materias, a partir del ejercicio constante de repensar la propuesta pedagógica desde la perspectiva de las buenas prácticas de enseñanza-aprendizaje, y brindando a los alumnos un protagonismo que les permite verse como autores de su aprendizaje, creciendo de forma personal y formando una conciencia colaborativa en este proceso enriquecedor.

Debemos reconocer que la utilización de las nuevas tecnologías como apoyo al dictado presencial, en muchos casos fue inicialmente una práctica improvisada, donde la buena voluntad de los docentes fue el mayor capital.

Actualmente, nos replanteamos estas estrategias, y pensamos en una implementación organizada, planificada y que responda a las pautas de calidad necesarias para permanecer en el tiempo, y mejorar nuestras prácticas docentes.

8. Agradecimientos

Este trabajo es posible, gracias a la gran colaboración y financiamiento de la Universidad Católica de Santiago del Estero, Dirección Académica Rafaela, que puso además a disposición toda su infraestructura tecnológica para llevar a cabo este proyecto de investigación.

9. Referencias

[1] Seuret, M.; Justiniani, A. M. “Algunas consideraciones sobre la calidad en instituciones de educación a distancia” en *“Construyendo la nueva agenda de la educación a distancia”* de Mena, Marta (coordinadora), La Crujía, Buenos Aires, 2007.

[2] Sangrá, A.; Sanmamed, M. G. “La transformación de las universidades: a través de las TIC: discursos y prácticas”, Editorial UOC, Barcelona, 2003.

[3] Pressman, R.S. "Ingeniería del Software. Un enfoque práctico", Mac Graw Hill, México, 2010.

[4] Lesourne, J. “Educación y sociedad. Los desafíos del año 2000”, Gedisa editorial, Barcelona, 1993.

[5] García Aretio, L. “Indicadores para la evaluación de la enseñanza en una universidad a distancia”, Revista Iberoamérica de Educación a Distancia, Vol 1, Nº 1, 1998.

[6] SCORM, 3rd edition. “Sharable Content Object Referente Model. Overview”. Versión 1.0. 2006. Advanced Distributed Learning.

[7] Mödritscher, F.; Spiel, S.; Garcia-Barrios, V.M.. “Assessment in E-Learning Environments: A Comparison of three Methods”. In C. Crawford et al. (Eds.), Proceedings of Society for Information Technology & Teacher Education International Conference 2006. pp. 108-113. Chesapeake, VA: AACE.

[8] Garcia Aretio, L. "Bases, mediaciones y futuro de la educación a distancia en la sociedad digital", UNED, Editorial Sintesis, España, 2014.

Estudio de Perfil de Rendimiento Académico: Un Abordaje desde Datawarehousing

David La Red Martinez, Marcelo Karanik, Mirtha Giovannini, Noelia Pinto
Facultad Regional Resistencia, Universidad Tecnológica Nacional
French 414, Resistencia, Chaco

Abstract

Con la necesidad de definir perfiles de estudiantes y determinar patrones que conduzcan al éxito o fracaso académico, se propone evaluar el rendimiento de los estudiantes mediante técnicas de Minería de Datos. La propuesta no se enfoca en analizar el perfil del estudiante a través de sus calificaciones, sino también estudiar el desempeño académico en base a otras variables que lo pueden afectar.

En este trabajo se describe un modelo de Data Warehouse para determinar los perfiles de rendimiento académico en la asignatura Algoritmos y Estructuras de Datos de la carrera Ingeniería en Sistemas de Información de la UTN-Facultad Regional Resistencia (UTN-FRRe), con el objetivo de instrumentar estrategias que ayuden a aquéllos estudiantes con perfil tendiente al fracaso o deserción. La característica principal del modelo es que combina aspectos relacionados a las calificaciones del estudiante junto a otras variables tales como factores socioeconómicos, demográficos, culturales, entre otros, en base a lo cual permite clasificar diferentes perfiles de alumnos.

1. Introducción

La Universidad enfrenta actualmente el desafío de mejorar su calidad académica sin incremento presupuestario en cantidad de cargos docentes y de tutorías, y al mismo tiempo, adaptarse a las demandas de nuevos escenarios producto del contexto globalizado del cual forma parte. Este reto no implica enfocarse sólo en el sistema de enseñanza-aprendizaje, sino que debe contemplar otras tantas variables, como por ejemplo la sistematización de procesos de evaluación permanentes que permitan monitorear cuestiones ligadas a la calidad académica y retroalimente la propuesta de mejora para la Universidad [1]. En este sentido uno de los factores más críticos que debe evaluarse continuamente es el rendimiento académico.

Se define al rendimiento académico como la productividad del sujeto, matizado por sus actividades, rasgos y la percepción más o menos correcta de los cometidos asignados [2].

Generalmente al evaluar el rendimiento académico, se analizan en mayor o menor medida los elementos que influyen en el desempeño como ser, entre otros, factores socioeconómicos, la amplitud de programas de estudio, las metodologías de enseñanza, conocimientos previos del alumno [3]. Por esto, no resulta adecuado evaluar el desempeño general de los alumnos a través de porcentajes de aprobación, notas obtenidas, etc., ya que este proceso de evaluación no brinda información que pueda ser utilizada para detectar, y corregir, problemas cognitivos, de aprehensión, de discernimiento, etc.

Surge, entonces, la necesidad de implementar un mecanismo que permita determinar las características propias del estudiante, analizando la existencia de relaciones y patrones de comportamiento estudiantiles que posibilite la definición clara de perfiles de alumnos. Para ello, una alternativa viable es desarrollar los métodos de evaluación en base a las tecnologías de información disponibles, aprovechando los beneficios y avances de éstas al momento de resolver problemas en diversos ámbitos.

Actualmente existen varios métodos para determinar y clasificar patrones que se utilizan en el área de la Inteligencia Artificial (del inglés Artificial Intelligence – AI) y del Aprendizaje de Máquinas (del inglés Machine Learning – ML) [4]. La Minería de Datos (del inglés Data Mining - DM), es un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias en grandes volúmenes de datos, utilizando técnicas de AI y ML. Estas técnicas permiten extraer patrones y tendencias para describir y comprender mejor los datos y predecir comportamientos futuros. Para dar soporte a los grandes volúmenes de información se necesita un almacén de datos (del inglés Data Warehouse - DW). Un DW es una colección de datos orientada a un dominio, integrada, no volátil y variante en el tiempo para ayudar a tomar decisiones [5]. Los DW surgieron por dos razones: primero, la necesidad de proporcionar una fuente única de datos limpia y consistente para propósitos de apoyo para la toma de decisiones; segundo, la necesidad de hacerlo sin afectar a los sistemas operacionales [6].
Existen algunos trabajos relacionados con la utilización de DW y DM para la determinación de rendimiento

académico [7][8][9], en ellos se modela al estudiante tomando como base ciertas características que le son propias.

En este trabajo se propone la utilización de técnicas de DM sobre información del desempeño de los alumnos de la cátedra Algoritmos y Estructura de Datos de la carrera Ingeniería en Sistemas de Información que se dicta en la Facultad Regional Resistencia de la Universidad Tecnológica Nacional.

El artículo se organiza de la siguiente manera: en la sección 2 se describen el estado del arte y conceptos introductorios relacionados a DW. En la sección 3 se describe el diseño del modelo propuesto de DW junto al proceso de DM utilizado para la clasificación y determinación de perfiles. Luego, en la sección 4 se describe el caso de estudio sobre el cual se validará a futuro el diseño del DW. Y finalmente, en la última sección se exponen conclusiones y trabajos futuros a realizarse.

2. Data Warehouse

Un DW es un repositorio de datos que proporciona una visión global, común e integrada de los datos y presenta las siguientes características [10]:

- Orientado a un tema: organiza una colección de información alrededor de un tema central.
- Integrado: incluye datos de múltiples orígenes y presenta consistencia de datos.
- Variable con el tiempo: se realizan fotos de los datos basadas en fechas o hechos.
- No volátil: sólo de lectura para los usuarios finales.

El término DW puede utilizarse indistintamente para hacer referencia a la arquitectura en sí misma o a uno de los componentes que la integran. Los bloques funcionales que se corresponden con un sistema de información completo que utiliza un DW se resumen en la Figura 1 [11]:

1. Nivel Operacional: hace referencia a los sistemas operacionales/transaccionales de la organización y a fuentes que forman parte del proceso de DW.
2. Nivel de acceso a la información: es la capa de interacción del usuario cuya finalidad es la conversión de los datos almacenados en información fácil y transparente para las herramientas de los usuarios finales.
3. Nivel de acceso a los datos: comunica el nivel de acceso a la información con el nivel operacional de forma universal.
4. Nivel de directorio de datos (metadatos): repositorio de metadatos de los datos almacenados que proporcionan información sobre el origen y sobre la transformación de los mismos en el proceso de DW.
5. Nivel de gestión de procesos: planificación de las tareas y procesos para la construcción y mantenimiento actualizado del DW.
6. Nivel de mensaje de la aplicación: determina el transporte de información a lo largo del entorno de computación de la organización a modo de middleware pero más allá de meramente protocolos de red.
7. Nivel DW (físico): es el repositorio central altamente flexible de información donde residen copias de los datos operacionales y/o externos optimizados para su acceso para la consulta.
8. Nivel de organización de datos: incluye todos los procesos necesarios para seleccionar, editar, resumir (normalmente sumarizar), combinar y cargar en el DW y en la capa de acceso a la información los datos operacionales y/o externos.

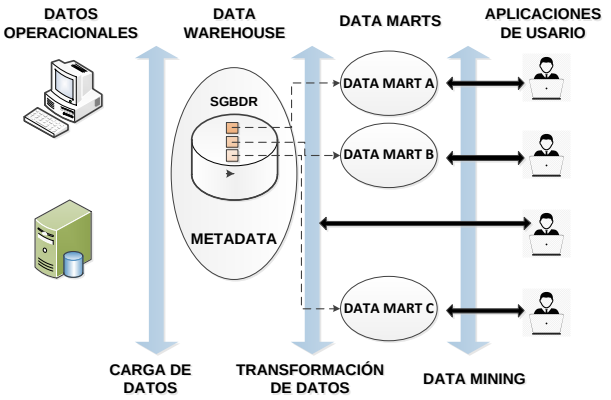


Figura 1. Arquitectura Básica de un DW

Detrás de la arquitectura de componentes del DW, existe un conjunto de procesos básicos asociados, los procesos ETL (del inglés Extract, Transform, Load – Extracción, Transformación y Carga). Los procesos ETL hacen referencia a la recuperación y transformación de los datos desde las fuentes orígenes, cargándolos en el DW.

En primer lugar los datos se analizan desde las fuentes, y se extraen aquéllos que serán de utilidad para el proceso en ejecución. El proceso de extracción debe ser sumamente cuidadoso, para evitar perturbar el normal

funcionamiento de los sistemas orígenes desde dónde se está recuperando información [12].

Luego de extraer los datos, se los carga al DW, pero, en muchas ocasiones, éstos requieren pasar por un proceso de transformación. La transformación de los datos significa un formateo y/o estandarización de los mismos, convirtiendo ciertos números en fechas, eliminando campos nulos, etc. Es necesario que antes de completar el DW con los datos, se realicen varios controles para enviar información cualitativamente correcta.

Finalmente, se procede a aplicar alguna técnica para realizar el análisis de los datos almacenados en el DW. El método más utilizado es el proceso de DM, que aplica la inteligencia artificial para encontrar patrones y relaciones dentro de los datos permitiendo la creación de modelos, es decir, representaciones abstractas de la realidad, pero es el descubrimiento del conocimiento (KDD, por sus siglas en inglés) que se encarga de la preparación de los datos y la interpretación de los resultados obtenidos, los cuales dan un significado a estos patrones encontrados [13]. Existen varias alternativas del DM, por ejemplo la Minería de Datos en Educación (Educational Data Mining, EDM).

El objetivo de la EDM es el desarrollo de métodos para la exploración de tipos de datos únicos provenientes de plataformas educativas, y usando estos métodos para entender mejor a los estudiantes en el aprendizaje [14].

Existen diversos estudios y publicaciones que abordan la evaluación de rendimiento académico utilizando técnicas de Minería de Datos [15] [16] [17].

3. Modelo Propuesto

La estructura del DW, como se muestra en la Figura 2, consta de una tabla de hechos y varias tablas de dimensión.

Una tabla de hechos o una entidad de hecho es una tabla o entidad que almacena medidas para medir el negocio, como las ventas, el coste de las mercancías o las ganancias [18]. Cada medida se corresponde con una intersección de valores de las dimensiones y generalmente se trata de cantidades numéricas, continuamente evaluadas y aditivas. Se pueden distinguir dos tipos de columnas en una tabla de hechos, columnas de hechos y columnas llaves. Las columnas de hechos almacenan las medidas del negocio que se quieren controlar y las columnas llaves, forman parte de la clave de la tabla.

Una tabla de dimensiones o entidad de dimensiones es una tabla o entidad que almacena detalles acerca de hechos. Por ejemplo, una tabla de dimensión de hora almacena los distintos aspectos del tiempo, como el año, trimestre, mes y día. Además incluye información descriptiva sobre los valores numéricos de una tabla de hechos. Por ejemplo, las tablas de dimensiones para una aplicación de análisis de mercado pueden incluir el tipo de período de tiempo, región comercial y producto.

Asimismo las tablas de dimensiones describen los distintos aspectos de un proceso de negocio. Por ejemplo, si se desea determinar los objetivos de ventas, se puede almacenar los atributos de dichos objetivos en una tabla de dimensiones. Cada tabla de dimensiones contiene una clave simple y un conjunto de atributos que describen la dimensión [19].

Las columnas de una tabla de dimensiones se utilizan para crear informes o para mostrar resultados de consultas. Por ejemplo, las descripciones textuales de un informe se crean desde las etiquetas de las columnas de una tabla de dimensiones.

El modelo que se presenta en este trabajo, se compone de la tabla de hechos “ALUMNOS” y varias tablas de dimensiones asociadas a la misma, que incluyen características que se desean estudiar. En la Figura 2, se representa gráficamente esta estructura.

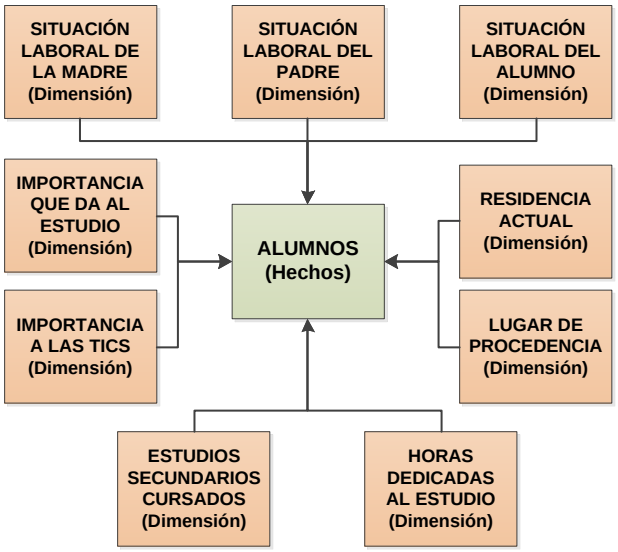


Figura 2. Modelo Propuesto del DW

Para llevar a cabo este trabajo se ha utilizado la suite de herramientas incluidas en el IBM DW Edition (DWE) V.9.5, un paquete de productos que combina la potencia de DB2 Universal Database (DB2 UDB) con la robusta infraestructura de inteligencia empresarial de IBM [20]. Se compone de varios productos pero a los fines de este trabajo se han usado los siguientes componentes:

- El Sistema de Gestión de Base de datos DB2 Enterprise Server Edition (DB2 ESE), que permite conexiones multiusuarios y presenta alta escalabilidad.

- Design Studio (DS): Una plataforma a modo de herramienta que usan los analistas de negocios y los administradores de almacenes de datos para diseñar las reglas de las cargas de trabajo, los flujos de transformación de datos y los flujos analíticos para la minería de datos y el análisis de textos.
- Intelligent Miner (IM): Conjunto de funcionalidades que permiten realizar análisis de información de acuerdo a conceptos de Business Intelligence (BI).

A continuación se describen los campos de cada una de las tablas que componen el DW con el que se trabajó.

3.1. Descripción de las Tablas

En la Tabla 1, se observa la estructura de la tabla de hechos "Alumnos" junto al detalle del significado de cada campo.

Tabla 1. Variables de la Tabla "Alumnos".

Nombre de Variable	Significado
LU_ALUM	Número de Legajo
DNI_ALUM	Número de DNI
CARRERA_ALUM	Carrera
SEXO_ALUM	Sexo
EDAD	Edad
ESTCIVIL_ALUM	Estado Civil
FECHA_NACIM	Fecha de Nacimiento
PAIS_NACIM	País de Nacimiento
PROV_NACIM	Provincia de Nacimiento
CIUDAD_NACIM	Ciudad de Nacimiento
FECHA_ENCUESTA	Fecha de la Encuesta
GRUPO_SANGUINEO	Grupo Sanguíneo
NOTA_PP_ALUM	Nota de Primer Parcial
NOTA_PR_ALUM	Nota de Primer Recuperatorio
NOTA_SP_ALUM	Nota de Segundo Parcial
NOTA_SR_ALUM	Nota de Segundo Recuperatorio
NOTA_EXT_ALUM	Nota de Extraordinario
SF	Situación Final del Alumno
ANIO	Año Lectivo

En la Tabla 2 se enumeran las variables de la tabla "Importancia que da al estudio", junto a la descripción correspondiente de cada una.

Tabla 2. Variables de la Tabla de "Importancia que da al Estudio".

Nombre de Variable	Significado
LU_IES	Número de Legajo
DNI_IES	Número de DNI
IMPORTANCIA_IES	Importancia dada al estudio

A continuación se detalla y explica en la Tabla 3, el significado de la tabla "Lugar de Procedencia".

Tabla 3. Variables de la Tabla de "Lugar de Procedencia".

Nombre de Variable	Significado
LU_P	Número de Legajo
DNI_P	Número de DNI
PROVINCIA_P	Provincia de procedencia
CIUDAD_P	Ciudad de procedencia

En la Tabla 4 se pueden observar las variables que conforman la tabla "Importancia otorgada a las TICS", junto a su correspondiente descripción.

Tabla 4. Variables de la Tabla de "Importancia Otorgada a las TICS".

Nombre de Variable	Significado
LU_TIC	Número de Legajo
DNI_TIC	Número de DNI
UTIL_TIC	Utilización de TICS

Luego en las Tablas 5 y 6 se enumera y se explica el significado correspondiente a las tablas "Estudios Secundarios cursados" y "Residencia Actual".

Tabla 5. Variables de la Tabla de "Estudios Secundarios Cursados".

Nombre de Variable	Significado
LU_ES	Número de Legajo
DNI_ES	Número de DNI
COLEGIO_ES	Nombre de Escuela
DEPENDENCIA_ES	Dependencia de Escuela
PROVINCIA_ES	Provincia a la que pertenece la Escuela
CIUDAD_ES	Ciudad a la que pertenece la Escuela
TITULO_ES	Título otorgado
FEGRESO_ES	Fecha de Egreso

Tabla 6. Variables de la Tabla de "Residencia Actual".

Nombre de Variable	Significado
LU_RA	Número de Legajo
DNI_RA	Número de DNI
TIPO_RA	Tipo de Residencia
CALLE_RA	Dirección actual
PROVINCIA_RA	Provincia donde reside
CIUDAD_RA	Ciudad donde reside

En la Tabla 7, se observa la descripción de la tabla de dimensión "Horas dedicadas al estudio".

Tabla 7. Variables de la Tabla de "Horas Dedicadas al Estudio".

Nombre de Variable	Significado
LU_ES	Número de Legajo
DNI_ES	Número de DNI
HS_ES	Horas dedicadas al estudio

A continuación se describen, en las tablas 8 y 9, las dimensiones "Situación Laboral de la Madre" y "Situación Laboral del Padre".

Tabla 8. Variables de la Tabla de "Situación Laboral de la Madre".

Nombre de Variable	Significado
LU_SLM	Número de Legajo
DNI_SLM	Número de DNI
ULT_EST_SLM	Grado de escolaridad de la madre
SIT_LAB_SLM	Situación laboral de la madre
HS_TRAB_SLM	Horas semanales trabajadas
REL_TRAB_SLM	Rama de la actividad laboral de la madre
CATOCUP_SLM	Categoría ocupacional de la madre

Tabla 9. Variables de la Tabla de "Situación Laboral del Padre".

Nombre de Variable	Significado
LU_SLP	Número de Legajo
DNI_SLP	Número de DNI
ULT_EST_SLP	Grado de escolaridad del padre
SIT_LAB_SLP	Situación laboral del padre
HS_TRAB_SLP	Horas semanales trabajadas
REL_TRAB_SLP	Rama de la actividad laboral del padre
CATOCUP_SLP	Categoría ocupacional del padre

En la Tabla 10, se enumeran y describen las variables que conforman la tabla "Situación Laboral del Alumno".

Tabla 10. Variables de la Tabla de "Situación Laboral del Alumno".

Nombre de Variable	Significado
LU_SL	Número de Legajo
DNI_SL	Número de DNI
TIPO_SL	Categoría ocupacional
ACTIV_ECONOM_SL	Rama actividad económica
HS_TRAB_SL	Horas semanales trabajadas
REL_TRAB_SL	Relación del trabajo con la carrera elegida
OBRA_SOCIAL_SL	Tiene Obra Social
CATOCUP_SL	Categoría ocupacional
SIT_LAB_SL	Situación Laboral

4. Caso de Estudio

El estudio se realizó sobre alumnos regulares que estaban cursando el 2°, 3°, 4° o 5° año de la carrera de Ingeniería en Sistemas de Información, de la Facultad Regional Resistencia, Universidad Tecnológica Nacional (FRRe-UTN) y particularmente se trabajó sobre el rendimiento que estos estudiantes tuvieron en la materia Algoritmos y Estructuras de Datos.

4.1. Etapa de Recolección de Datos

Tal como se planteó en la sección 1, el estudio del desempeño académico de los estudiantes no sólo debe evaluarse teniendo en cuenta los resultados de las instancias de evaluaciones previstas por asignatura, sino que también deben analizarse otros factores culturales, sociales y/o económicos que afecten el rendimiento del alumno.

Por ello, para este trabajo resultó determinante la participación directa del estudiante, pues era necesario conocer datos sobre aspectos personales que no se podían obtener de otra manera que no fuera a través de respuestas directas por parte de cada alumno.

A tal fin, se dispuso la elaboración de una aplicación web que permitió contar con una Encuesta On-Line compuesta por preguntas relacionadas a situación familiar, historial de estudios secundarios, entre otras cuestiones. En la Figura 3, se muestra un ejemplo de pantalla correspondiente al Sistema de Encuesta Web que contiene un formulario que fue utilizado por los estudiantes para responder sobre diversas cuestiones.

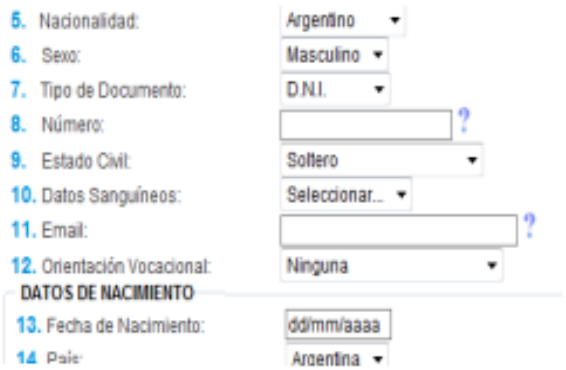


Figura 3. Formulario Inicial del Sistema de Encuesta Web.

A continuación en las Tablas 11, 12, 13, 14, 15, 16, 17 y 18 se enumeran y describen los datos solicitados en la Encuesta web.

Tabla 11. Datos de la Encuesta Web – 1ra. Parte.

Campo	Significado	Opciones
Facultad	UTN-FRRE (por defecto)	
Carrera	Ing. en Sist. de Inf. (por defecto)	
Año Académico	Año en que se evalúa el desempeño del alumno	Ingresar

Apellidos	Del Alumno	Ingresar
Nombres	Del Alumno	Ingresar
Nacionalidad	Del Alumno	Argentino
		Extranjero
		Naturalizado
		Por opción.
Sexo	Género del Alumno	Femenino
		Masculino
Tipo de Documento	Del Alumno	DNI
		LC
		LE
		PASAPORTE
Número	Número de DNI del Alumno	Ingresar

Tabla 12. Datos de la Encuesta Web – 2da. Parte.

Campo	Significado	Opciones
Estado Civil	Del Alumno	Soltero
		Casado
		Divorciado
		Viudo
		Unión de hecho
Grupo Sanguíneo	Del Alumno	Ingresar
Email	Del Alumno	Ingresar
Orientación Vocacional	¿El alumno obtuvo orientación vocacional antes del ingreso a la carrera?	Ninguna
		Entidad oficial
		Entidad privada
		Profesional
DATOS DE NACIMIENTO		
Fecha	Del Alumno	Ingresar
País	Del Alumno	Seleccionar en lista de países
Provincia	Del Alumno	Seleccionar en lista de provincias
Departamento	Del Alumno	Ingresar
Localidad	Del Alumno	Ingresar
Código Postal	De la Localidad	Ingresar

Tabla 13. Datos de la Encuesta Web – 2da. Parte.

DATOS DEL SECUNDARIO		
Nombre	Nombre del Colegio de donde egresó el alumno	Ingresar
Título Obtenido	Del Alumno	Ingresar
Dependencia	Del Establecimiento	Seleccionar entre la lista
Calle	Del Establecimiento	Ingresar
País	Del Establecimiento	Seleccionar entre la lista
Provincia	Del Establecimiento	Seleccionar entre la lista
Departamento	Del Establecimiento	Ingresar
Localidad	Del Establecimiento	Ingresar
Código Postal	Del Establecimiento	Ingresar
Año de Egreso	Del Alumno	Ingresar

Tabla 14. Datos de la Encuesta Web – 3ra. Parte.

DATOS FAMILIARES		
Apellido/s del Padre	Del Alumno	Ingresar
Nombre/s del Padre	Del Alumno	Ingresar
Apellido/s de la Madre	Del Alumno	Ingresar
Nombre/s de la Madre	Del Alumno	Ingresar
DOMICILIO PERIODO LECTIVO ACTUAL		
Tipo de Residencia	Del Alumno	Con Familiares
		Independiente
		Residencia Universitaria
		Otra Situación
Calle	Donde vive actualmente el Alumno	Ingresar
País	Donde vive actualmente el Alumno	Seleccionar entre la lista
Provincia	Donde vive actualmente el Alumno	Seleccionar entre la lista
Departamento	Donde vive actualmente el Alumno	Ingresar
Ciudad	Donde vive actualmente el Alumno	Ingresar
Código Postal	Donde vive actualmente el Alumno	Ingresar

Tabla 15. Datos de la Encuesta Web – 4ta. Parte.

DOMICILIO DE PROCEDENCIA FAMILIAR		
Calle	De residencia familiar del Alumno	Ingresar
País	De residencia familiar del Alumno	Seleccionar entre la lista
Provincia	De residencia familiar del Alumno	Seleccionar entre la lista
Departamento	De residencia familiar del Alumno	Ingresar
Localidad	De residencia familiar del Alumno	Ingresar
Código Postal	De residencia familiar del Alumno	Ingresar
Teléfono	De residencia familiar del Alumno	Ingresar
SITUACIÓN DEL ALUMNO		
Vive alguno de los padres		Si
		No
Cantidad de Familiares a cargo		Ingresar
Situación Laboral		Ocupado
		No trabaja

		Sub-Ocupado
		Desocupado
		Jubilado
Categoría Ocupacional		Seleccionar entre la lista
Rama de Actividad Económica		Seleccionar entre la lista
Horas semanales trabajadas		No trabaja
		Hasta 20
		De 21 a 35
		De 36 o más
Relación del trabajo con la carrera elegida		Total
		Parcial
		No se relaciona
Tiene Obra Social		Ninguna
		De los padres
		Del cónyuge
		Propia
		Otros

Tabla 16. Datos de la Encuesta Web – 5ta. Parte.

SITUACIÓN DEL PADRE		
Ultimo Nivel de estudios cursados	Del Padre	Seleccionar entre la lista
Situación Laboral	Del Padre	Ocupado
		No trabaja
		Sub-Ocupado
		Desocupado
		Jubilado
Categoría Ocupacional	Del Padre	Seleccionar entre la lista
Rama de Actividad Económica		Seleccionar entre la lista
Horas semanales trabajadas		No trabaja
		Hasta 20
		De 21 a 35
		De 36 o más

Tabla 17. Datos de la Encuesta Web – 6ta. Parte.

SITUACIÓN DE LA MADRE		
Ultimo Nivel de estudios cursados	De la Madre	Seleccionar entre la lista
Situación Laboral	De la Madre	Ocupado
		No trabaja
		Sub-Ocupado
		Desocupado
		Jubilado
Categoría Ocupacional	De la Madre	Seleccionar entre la lista
Rama de Actividad Económica		Seleccionar entre la lista
Horas semanales trabajadas		No trabaja
		Hasta 20
		De 21 a 35
		De 36 o más

Tabla 18. Datos de la Encuesta Web – 7ma. Parte.

ACTITUD HACIA EL ESTUDIO		
Prioridad otorgada al estudio		Más que al trabajo
		Más que a la Familia
		Más que a la Diversión
Número de horas semanales dedicadas al estudio	Sin considerar horas de clases	Hasta 10 inclusive
		Más de 10 y hasta 20 inclusive
		Más de 20
Estudia para		Aprobar las asignaturas
		Aprender integralmente y aprobar
		Aprender acerca de los contenidos
		Otras motivaciones
Utiliza TICS en el estudio porque		Están de moda
		Facilitan la enseñanza
		Son una realidad de la actualidad
		Para la vida profesional

4.2. Etapa de Depuración y Preparación de Datos

Para la realización de una correcta explotación del DW, se debe asegurar que los datos obtenidos en la etapa anterior sean consistentes y mantengan la coherencia entre ellos. De esta forma, en esta etapa se procedió a un proceso de limpieza en los datos que significó la eliminación de aquéllos registros con todos sus campos en blanco, corrección de errores tipográficos, llenado de algunos campos nulos, entre otros. Además la Encuesta no permite la carga, por parte de los estudiantes, de calificaciones de la asignatura en estudio. Esto se dispuso así para evitar errores en los datos ya sea por olvido, o confusión al momento de ingresar los valores. Por ello la carga de notas correspondientes al 1° Parcial, 2° Parcial, 3° Parcial, Recuperatorios, Examen Final y Situación del alumno (Regular, Promocionado, Libre), fue realizada por el equipo responsable de este trabajo de investigación. La información se obtuvo a partir de la base de datos histórica de la cátedra respecto a calificaciones de los alumnos.

Con esta información depurada se deberá proceder a trabajar en las próximas etapas:

- Carga de Datos al DW: Mediante la ejecución del flujo de datos, la información almacenada en la tabla ENCUESTA se distribuirá a las tablas pertenecientes al modelo del DW.
- Obtención de Resultados finales: Mediante el diseño y ejecución de flujos de minería a partir

de la información disponible respecto a diversos aspectos (Lugar de Procedencia, Horas dedicadas al estudio, Situación Laboral de los padres, etc.)

5. Discusiones y Trabajos Futuros

En este artículo se propone un modelo para la determinación de perfiles de rendimiento académico de alumnos, utilizando técnicas de DM. Hasta el momento se ha completado la primera etapa, que implicó el diseño del modelo del DW sobre el cual se implementarán técnicas de DM a fin de determinar perfiles de estudiantes vinculados a su desempeño académico en la asignatura Algoritmos y Estructuras de Datos de la carrera Ingeniería en Sistemas de Información de la UTN-FRRe.

En el avance que aquí se presenta respecto del Proyecto, se pudo comprobar que la etapa de depuración y preparación de los datos ha demandado mucho tiempo y esfuerzo debido principalmente a la poca integridad y coherencia que existía en la información que se utilizará para realizar la evaluación final.

Como trabajos futuros se prevé finalizar con el proceso de minería de datos a fin de evaluar y comparar patrones que se obtengan para definir los perfiles de estudiantes. La evaluación, análisis y utilidad de estos patrones permitirá soportar la toma de decisiones eficaces por parte del cuerpo docente de la asignatura analizada, quienes, en el marco de los programas institucionales que adhieren al Plan Estratégico pata la Formación de Ingenieros, deben enfocarse en formular políticas y estrategias relacionadas con los programas de retención estudiantil, que actualmente se encuentran establecidos, mejorando las estadísticas de regularización y aprobación aportando al ciclo de mejora continua respecto a desempeño académico en la UTN –FRRe.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del proyecto "**Determinación de Perfiles de Estudiantes y de Rendimiento Académico Mediante la Utilización de Mineria de Datos**". Universidad Tecnológica Nacional, Facultad Regional Resistencia. Código 25/L059 (programa de Incentivos a Docentes Investigadores del Ministerio de Educación). Código EIINRE571. (01/01/2013 – 31/12/2014).

Referencias

[1] Briand, L. C., Daly, J., and Wüst, J., "A unified framework for coupling measurement in objectoriented systems", *IEEE Transactions on Software Engineering*, 25, 1, January 1999, pp. 91-121.

[2] Maletic, J. I., Collard, M. L., and Marcus, A., "Source Code Files as Structured Documents", in *Proceedings 10th IEEE International Workshop on Program Comprehension (IWPC'02)*, Paris, France, June 27-29 2002, pp. 289-292.

[3] Marcus, A., *Semantic Driven Program Analysis*, Kent State University, Kent, OH, USA, Doctoral Thesis, 2003.

[4] Marcus, A. and Maletic, J. I., "Recovering Documentation-to-Source-Code Traceability Links using Latent Semantic Indexing", in *Proceedings 25th IEEE/ACM International Conference on Software Engineering (ICSE'03)*, Portland, OR, May 3-10 2003, pp. 125-137.

[5] Salton, G., *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, 1989.

[6] Molina López, J., García Herrero, J.: "Técnicas de Análisis de Datos". Universidad Carlos III. Madrid, España (2006).

[7] La Red Martínez D. L., Podestá Gómez C. E.: "Contributions from Data Mining to Study Academic Performance of Students of a Tertiary Institute American". Journal of Educational Research, 2014, Vol. 2, No. 9, 713-726

[8] La Red Martínez D. L, Acosta J. C., Uribe V. E., Rambo, A. R.: "Academic Performance: An Approach From Data Mining". Jounal of Systemics, Cybernetics and Informatics, 2012, Vol. 10, No. 1, 66-72.

[9] La Red Martínez D. L, Acosta J. C., Agostini F., Uribe V. E., Rambo, A.: "La importancia otorgada al estudio y su relación con el rendimiento académico". Revista Documentación, 2011, No 24, 54-63.

[10] Curto Días, J.: "Introducción al Business Intelligence". Editorial UOC. Barcelona, España. ISBN: 978-84-9788-886-8 (2010).

[11] "Arquitectura de un datawarehouse". Recuperado en http://informationmanagement.wordpress.com/2007/06/03/arquitectura-de-un-datawarehouse/

[12] Rainardi, V.:"Building a Data Warehouse: With Examples in SQL Server".Apress.USA(2008).

[13] Valcárcel Asencios, V.: "DataMining y el Descubrimiento del Conocimiento".

[14] Baker, R., Yaceff, K.: "The State of Educational Data Mining in 2009: A Review and Future Visions". Journal of Educational Data Mining, Articulo 1, Vol 1, No 1 (2009).

[15] Formia, S., Lanzarini, L., Hasperué, W.: "Characterization of University Drop-Out at UNRN Using Data Mining. A study case." Mar del Plata, Buenos Aires. CACIC 2013.

[16] Dapozo, G., Porcel, E., López, M., Bogado, V., Bargiela, R.:"Aplicación de minería de datos con una herramienta de software libre en la evaluación del rendimiento académico de los alumnos de la carrera de Sistemas de la FACENA-UNNE".

[17] Timarán Pereira, R.: " Detección de Patrones de Bajo Rendimiento Académico y Deserción Estudiantil con Técnicas de Minería de Datos".

[18] Información disponible en http://pic.dhe.ibm.com/infocenter/idm/docv3/index.jsp?topic=%2Fcom.ibm.datatools.dimensional.ui.doc%2Ftopics%2Fc_dm_fact_tables.html

[19] La Red Martínez, D.; Acosta, J.; Cutro, L.; Uribe, V.; Rambo, A.: "Data Warehouse y Data Mining Aplicados al Estudio del Rendimiento Académico". Dpto. de Informática. FACENA. Universidad Nacional del Nordeste (2010).

[20] Información disponible en ftp://public.dhe.ibm.com/ps/products/dwe/V8.2.1/es_ES/getstart821.pdf

Herramienta de seguimiento del rendimiento académico en la Facultad de Ingeniería (Sede Puerto Madryn)

Romina Stickar
Dto. de Informática, Fac. de Ingeniería
Univ. Nacional de la Patagonia San Juan Bosco
Puerto Madryn, Argentina
+54-0280-4472885-int. 117
romistickar@gmail.com

Laura Biscayart
Dto. de Informática, Fac. de Ingeniería
Univ. Nacional de la Patagonia San Juan Bosco
Puerto Madryn, Argentina
+54-0280-4472885-int. 117
lalibiscayart@gmail.com

Sandra Alvarez
Dto. de Informática, Fac. de Ingeniería
Univ. Nacional de la Patagonia San Juan Bosco
Puerto Madryn, Argentina
+54-0280-4472885-int. 117
s_alvarez@speedy.com.ar

Sergio Salvatierra
Dto. de Informática, Fac. de Ingeniería
Univ. Nacional de la Patagonia San Juan Bosco
Puerto Madryn, Argentina
+54-0280-4472885-int. 117
sersalvatierra@gmail.com

Resumen

En este trabajo se relevan y estudian los datos del rendimiento de los alumnos que ingresan a primer año de la Facultad de Ingeniería (Sede Puerto Madryn), a través de un seguimiento de su primer año académico y primer cuatrimestre de segundo año.

Se determinan una serie de indicadores, que se van a utilizar para lograr definir acciones de retención y/o motivación para el alumnado.

La información obtenida ratifica una preocupación permanente de las universidades públicas argentinas que revela que son muy pocos los estudiantes que completan la carrera y en el plazo teórico de duración.

Palabras clave: Rendimiento académico, indicadores, universidades, alumnos

1. Introducción

El **rendimiento académico** es uno de los indicadores del nivel de aprendizaje alcanzado por el alumno, de ahí la importancia del mismo para el sistema educativo. Además el rendimiento académico está relacionado con un problema serio, como es la deserción del alumno.

La información de las universidades públicas argentinas revela que son muy pocos los estudiantes que completan la carrera y en el plazo teórico de duración. Para 20 carreras de todas las universidades públicas el retraso promedio de la graduación es de alrededor del 60% (Cociente entre la duración efectiva y la teórica en años). En base a datos del Ministerio de Cultura y Educación de la Nación (1996) varios factores explican el retraso: sólo el 60% de los estudiantes no trabaja; el fracaso vocacional lleva a iniciar una segunda carrera; es baja la participación de las becas en el financiamiento de los estudiantes y la programación curricular es inadecuada (Gertel, 1999).

Estos indicadores revelan un pobre desempeño de los estudiantes universitarios que ha generado preocupación y motivado la búsqueda de factores explicativos que permitan diseñar y evaluar políticas para el sector. Estas políticas, entre otras, comprenden los mecanismos de admisión a las Facultades y las transiciones que siguen al ingreso.

Un punto de partida útil para entender el proceso que siguen los estudiantes es comenzar con la "contabilidad de los logros académicos" o "contabilidad del rendimiento estudiantil". La información disponible se organiza para responder a preguntas tales como: ¿Cuántos alumnos completan en el tiempo teórico el primer cuatrimestre de primer año? ¿Cuántos alumnos continúan en el segundo cuatrimestre del primer año? ¿Los números varían en materias no correlativas con respecto a materias correlativas? ¿Se mantiene en segundo año la cantidad de alumnos que aprobaron las materias de primer año? ¿Cuántos no pueden aprobar ninguna materia? Estos y otros interrogantes se pueden contestar a partir de la información recopilada para diferentes cohortes.

La finalidad del proyecto es diseñar una herramienta (indicadores) que permita evaluar y analizar permanentemente el desempeño académico en los primeros años de estudios

universitarios de la Facultad de Ingeniería sede Puerto Madryn. Asimismo, se pretende lograr un registro estadístico que permita planificar acciones de retención y motivación para los alumnos.

2. Desarrollo de la Investigación

Esta investigación, abarcó **4 (cuatro) etapas de trabajo**:

Etapa Nº 1: Capacitación del equipo de investigación.

Se realizó una búsqueda en la web de documentos y profesionales investigadores en la temática.

Se realizó el Seminario/taller "Las tareas de investigación" enfocado a las necesidades del proyecto.

Etapa Nº 2: Selección de métricas e indicadores candidatos.

Se dio inicio a la normalización, sistematización y un primer análisis de los datos de la Encuesta elaborada por el equipo, a alumnos del último año de 9 (nueve) escuelas secundarias que visitaron la Sede Puerto Madryn UNPSJB durante el evento anual "Venite a la uni".

También se trabajó con datos suministrados por las autoridades sobre las carreras: Analista Programador Universitario – Licenciatura en Informática [1].

Se hizo un muestreo completo cuantitativo con registros existentes y en campo controlando planillas de materias cedidas por el Departamento de Alumnos de la facultad.

Etapa Nº 3: Elaboración de métricas e indicadores.

Los primeros datos recogidos a través de la Encuesta del "Venite a la Uni" fueron:

- Cantidad de alumnos por escuela
- Cantidad de alumnos que desean seguir estudiando
- Carreras que desean estudiar
- Cantidad de alumnos que estudiaran en otra provincia
  - En qué provincia
  - En qué ciudad
  - En qué universidad
- Cantidad de alumnos que estudiarán en Chubut
  - En qué ciudad
  - En qué universidad
- Dificultades que cree puede tener en su futuro estudio
  - Económicas
  - Socio-afectivas
- Materias
  - Matemática
  - Lengua
  - Técnicas de Estudio
  - Otras

Los indicadores obtenidos de dicha Encuesta fueron orientativos para saber el porcentaje de alumnos que estudiarían las carreras de Ingeniería en la Sede Puerto Madryn de la UNPSJB.

Indicadores que provienen de la encuesta:

Tabla 1. Indicadores

Indicador	Cálculo	Resultado
% de encuestados en Puerto Madryn	Cantidad de encuestados / Cantidad de egresados	Dado que un alto porcentaje de colegios secundarios asisten al evento Venite a la Uni se obtuvo un porcentaje alto (89%) de encuestas respecto a la cantidad de alumnado de la ciudad.
% de intencionalidad de seguir estudiando	Cantidad de encuestados / Cantidad de respuestas positivas (P)	Da un porcentaje muy alto de intencionalidad : 93%
% de alumnos que estudiarían en Chubut	Cantidad de alumnos estudiarán en Chubut(CHUB) / Cantidad de respuestas positivas	Ver Figura 1
% de alumnos por ciudad en Chubut	cantidad alumnos que eligen ciudad X/ CHUB	X: Ciudades en Chubut (Puerto Madryn, Trelew, Comodoro, Esquel y Rawson) Ver Figura 2

% de alumnos por establecimiento en Chubut	cantidad alumnos que eligen esta-blecimiento X/ CHUB	X: establecimiento (Universidad /Instituto/ etc.) (por ej. UNPSJB, UTN, ISFD 803, etc.)
% de alumnos por carreras que estudiarán en la UNPSJB	cantidad alumnos que eligen la carrera X / cantidad alumnos que eligen establecimiento UNPSJB	X: carreras dictadas en la UNPSJB (Abogacía, Administración de Empresas, Analista Programador Universitario, Biología, Contador Público, Ingeniería, Licenciatura en Ciencias Económicas, Licenciatura en Informática, Psicología, Turismo, etc.) Ver Figura 3

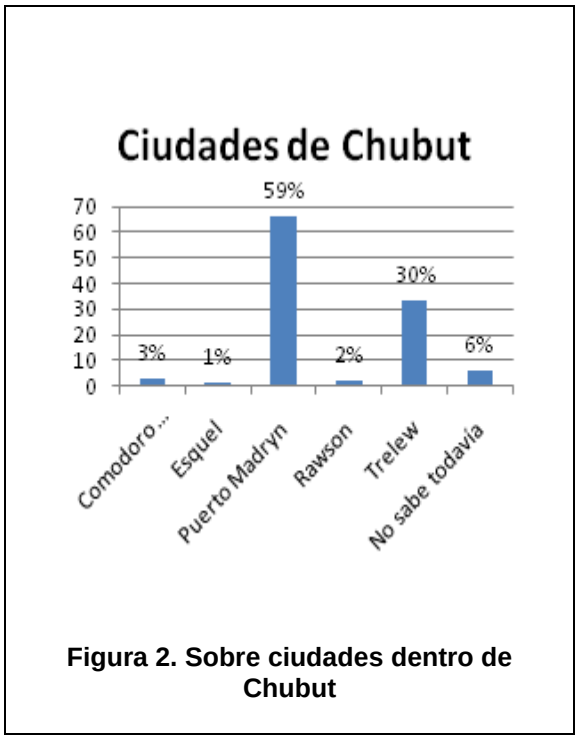


Figura 2. Sobre ciudades dentro de Chubut

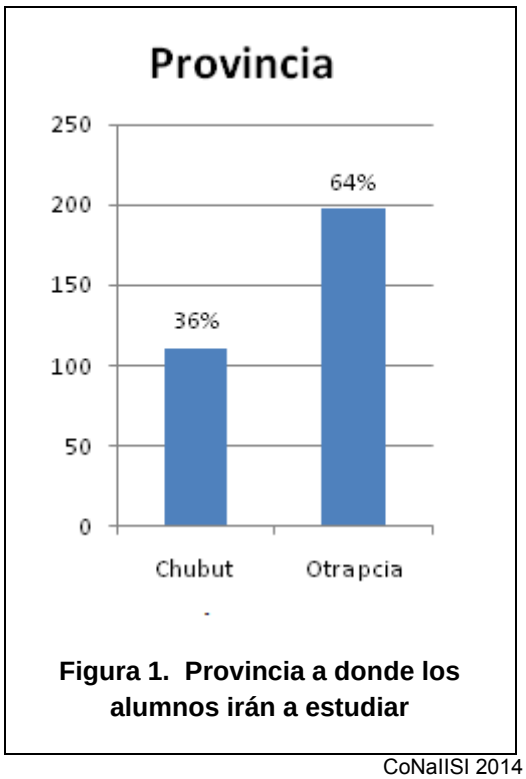


Figura 1. Provincia a donde los alumnos irán a estudiar

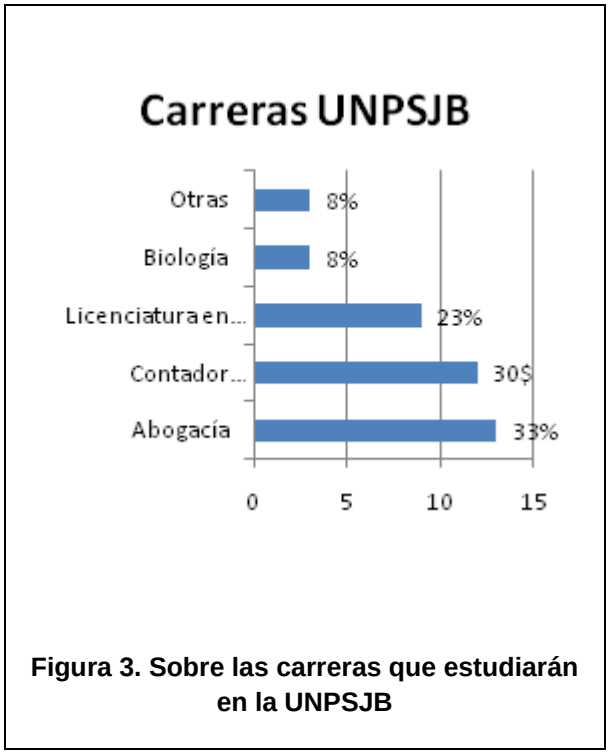


Figura 3. Sobre las carreras que estudiarán en la UNPSJB

Los datos de los alumnos fueron extraídos del departamento de alumnos de la Facultad de Ingeniería, sede Puerto Madryn (ver Figura 5):

- Cantidad de alumnos postulantes por año
- Cantidad de alumnos ingresantes por año
- Materias del 1er cuatrimestre del primer año de carrera
- Cantidad de inscriptos por 1ra vez (ingresantes)
- Cantidad de inscriptos por 2da vez (recursantes)
- Cantidad de alumnos que aprobaron
- Cantidad de alumnos que desaprobaron
- Cantidad de alumnos ausentes
- Materias del 2do cuatrimestre del primer año de carrera (datos ídem 1er cuatrimestre)
- Materias del 1er cuatrimestre del segundo año de carrera

De acuerdo a los datos registrados en el Boletín Estadístico Nº 3 de la UNPSJB, más del 85% de los estudiantes provienen de escuelas de la provincia de Chubut (se toma el Lugar de residencia estable).

Entonces, posibles indicadores de alumnos:

Tabla 2. Indicadores de alumnos

N°	Indicador	Cálculo	Obs.
1	% de ingresantes respecto a los Postulantes	Cantidad de ingresantes / Cantidad de postulantes	Se debe aprobar un examen de ingreso en los períodos diciembre y marzo
2	% de aprobados/ desaprobados / ausentes en cada materia del 1er y 2do cuatrimestre (para los ingresantes y para los recursantes)	Cantidad de aprobados Cantidad de desaprobados Cantidad de ausentes	
3	Deserción de alumnos ingresantes a mitad de 1er año, para todas las materias	Cantidad de de inscriptos (ingresantes) en cada materia del 2do cuatrimestre/ Cantidad de ingresantes	Se toma un promedio de inscriptos en las materias del 2do cuatrimestre para determinar qué porcentaje de alumnos continúan estudiando a mitad de año del 1er año
4	Deserción de alumnos ingresantes a mitad de 1er año, para materias correlativas	Cantidad de de inscriptos (ingresantes) en cada materia del 2do cuatrimestre/ Cantidad de ingresantes	Ídem indicador 3 pero para materias correlativas
5	Aprobados en materias del 1er cuatrimestre de 1er año respecto a los inscriptos en 2do cuatrimestre	Cantidad de aprobados en materias del 1er cuatrimestre/ cantidad de inscriptos (ingresantes) del 2do cuatrimestre	Se determina porcentaje de alumnos inscriptos en el 2do cuatrimestre que aprobaron materias del 1er cuatrimestre y que porcentaje continúan sin aprobar materias del 1er cuatrimestre (con materias no correlativas)
6	Deserción de alumnos (ingresantes del año anterior) a principio de 2do año	Cantidad de de inscriptos (ingresantes del año anterior) en cada materias del 1er cuatrimestre de 2do año / Cantidad de ingresantes del año anterior	Se determina cuántos de los alumnos ingresados en el año anterior del año de estudio siguen estudiando
7	Deserción de alumnos (ingresantes	Cantidad de de inscriptos (ingresantes	Ídem 6 sumando los

	del año anterior) a principio de 2do año sumados los recursantes	del año anterior) en cada materia del 1er cuatrimestre de 2do año + Cantidad de inscriptos (recursantes) en materias del 1er cuatrimestre de 1er año / Cantidad de ingresantes del año anterior	recursantes

3. Resultados

A continuación se analizaron y presentaron los resultados de algunos de los indicadores para el período 2007-2011:

Tabla 3. Indicador 1. Porcentaje de ingresantes respecto a los Postulantes

Año	A.P.U. Lic. en Informática
2007	24%
2008	26%
2009	22%
2010	50%
2011	47%

Tabla 4. Indicador 3. Deserción de alumnos ingresantes a mitad de 1er año (para todas las materias)

Año	A.P.U Lic. en Informática
2007	32%
2008	42%
2009	37%
2010	75%
2011	60%

Estos números indican el porcentaje de alumnos que continúan estudiando.

Tabla 5. Indicador 4. Deserción de alumnos ingresantes a mitad de 1er año (para materias correlativas)

Año	A.P.U. Lic. en Informática
2008	84%
2009	71%
2010	65%
2011	81%

Estos números indican el porcentaje de alumnos que continúan estudiando.

Tabla 6. Indicador 5. Aprobados en materias del 1er cuatrimestre de 1er año respecto a los inscriptos en 2do cuatrimestre

Año	A.P.U Lic. en Informática
2007	47%
2008	67%
2009	50%
2010	56%
2011	55%

Se observó que alrededor del 60% de los alumnos inscriptos en el 2do cuatrimestre corresponde a alumnos que han aprobado materias del 1er cuatrimestre. El 40% restante son alumnos que a pesar de haber desaprobado esas materias siguen estudiando.

Tabla 7. Indicador 6. Deserción de alumnos (ingresantes del año anterior) a principio de 2do año

Año	A.P.U. Lic. en Informática
2007	
2008	48%
2009	26%
2010	26%
2011	46%

Estos valores indican el porcentaje de alumnos que continúan estudiando el 2do año.

Tabla 8. Indicador 7. Deserción de alumnos (ingresantes del año anterior) al inicio de 2do año sumados los recursantes

Año	A.P.U Lic. en Informática
2007	
2008	52%
2009	53%
2010	31%
2011	54%

Estos números indican el porcentaje de alumnos que continúan estudiando el 2do año más los que recursan materias de 1er año.

4. Conclusiones

Se diseñó una herramienta (indicadores) para el relevamiento de la información del rendimiento académico de los alumnos de la Facultad de Ingeniería en la sede Puerto Madryn de la UNPSJB.

Los porcentajes observados indicaron que: en el ingreso a la facultad, se pierde entre de un 20 a un 30% de postulantes; y de los ingresantes, aproximadamente un 50% dejan sus estudios a mitad de año.

De acuerdo a estudios exploratorios y análisis similares en lugares geográficos del país y universidades con mayor cantidad de alumnos, se han observado resultados similares en lo que respecta a ingreso y deserción [2] [3].

A partir de los indicadores presentados, la Facultad de Ingeniería posee datos concretos para poder realizar y/o implementar Proyectos relacionados con el Ingreso, Permanencia y Egreso de los estudiantes denominado Programa Integrado de Ingreso, Permanencia y Egreso**.**

En relación al PIPE Ingreso, mediante las siguientes líneas de acción;

- **I.** Articulación con el nivel secundario.
- **II.** Información y orientación sobre las carreras y el campo laboral.
- **III.** Ciclo Integrado de Ingreso.
- **IV.** Sistema de tutorías y becas.
- **V.** Prácticas técnicas desde el primer año.

En relación al PIPE Permanencia, mediante las siguientes líneas de acción:

- I. Contención socio- económica.
- II. Contención académica.

Otra línea de investigación es determinar las capacidades y competencias no adquiridas de los alumnos que egresan de las escuelas secundarias de la provincia del Chubut que resultan más críticas en el ingreso al nivel superior y distinguir los factores asociados con el rendimiento académico y la deserción de los alumnos.

5. Referencias

[1] Anexo I - Res CAFI 274-04 Parte II y III. Anexo I - Res. CAFI 274-04-2005. Anexo I - Res. CAFI 274-04-2007. Anexo I - Res. CAFI 274-04-2008. Anexo I - Res. CAFI 274-04-2009. Anexo I - Res. CAFI 274-04 2010.

[2] Corengia, Ángela, Mesurado, Belén y Redelico, Federico, "Las aptitudes educacionales y su relación con el rendimiento académico y la deserción: un estudio

exploratorio". Revista Internacional de Estudios en Educación 2006, Año 6, Nº 2, 75-83

[3] Corengia, Ángela y Pita, María "Las aptitudes educacionales y su relación con el rendimiento académico y la deserción: un estudio exploratorio", V Coloquio Internacional sobre Gestión Universitaria en América del Sur, 2005.

[4] Girón Cruz, Luis Eduardo y González Gómez, Daniel Enrique, "Determinantes del rendimiento académico y la deserción estudiantil", programa de Economía de la Pontificia Universidad Javeriana de Cali.

[5] Ibarra, María del Carmen y Michalus, Juan Carlos, "Desempeño académico de estudiantes de ingeniería: análisis de factores incidentes". X Encontro Gaúcho de Educação Matemática. Comunicação Científica 02 a 05 de junho de 2009, Ijuí/RS.

[6] Ministerio de Educación, Ciencia y Tecnología. "Informe de gestión 2003 – 2007".

[7] Muriete, Raúl "Perspectivas actuales de la investigación educativa" Características predominantes en la organización de las Ciencias de la Educación a partir del estudio de campo de las prácticas investigativas. Revista Aprendizaje Hoy Año XX Nª 45, pág. 17 a 31.

[8] Ministerio de Educación, Ciencia y Tecnología. "Programa de Calidad Universitaria- PACENI" – Proyectos de Apoyo para el Mejoramiento de la Enseñanza en Primer Año de Carreras de Grado de Ciencias Exactas y Naturales, Ciencias Económicas e Informáticas.

[9] Universidad Nacional de la Patagonia San Juan Bosco. "Informe Proyecto de Tutorías 2007-2008"

Datos Materias Facultad Ingeniería

Año lectivo	Cuatrimestre	Materia Codigo	Nombre	Correlativa	Inscriptos 1ra	Inscriptos Rec	Inscriptos Total	Aprobados 1ra	Aprobados Rec	Aprobados Total	Desaprobados 1ra	Desaprobados Rec	Desaprobados Total	Ausentes 1ra	Ausentes Rec	Ausentes Total	% Aprob 1ra	% Aprob Rec	% Aprob Total	% Desaprob 1ra	% Desaprob Rec	% Desaprob Total	% Ausentes 1ra	% Ausentes Rec	% Ausentes Total
2011	1cr	MA045	Algebra		39	4	43	14	0	14	21	3	24	4	1	5	36%	0%	33%	54%	75%	56%	10%	25%	12%
2011	1cr	IF001	Elementos de informatica		39	6	45	13	2	15	17	2	19	9	2	11	33%	33%	33%	44%	33%	42%	23%	33%	24%
2011	1cr	IF002	Expresión de Problemas y Algoritmos		39	7	46	18	1	19	10	2	12	11	4	15	46%	14%	41%	26%	29%	26%	28%	57%	33%
2011	2do	MA046	Análisis Matemático I		29	15	44	7	2	9	6	3	9	16	10	26	24%	13%	20%	21%	20%	20%	55%	67%	59%
2011	2do	MA008	Elementos de Lógica y Matemática Discreta		29	4	33	10	3	13	13	1	14	6	0	6	34%	75%	39%	45%	25%	42%	21%	0%	18%
2011	2do	IF005	Algorítmica y Programación I	IF001	20	2	22	7	0	7	9	1	10	4	1	5	35%	0%	32%	45%	50%	45%	20%	50%	23%
2010	1cr	IA001/MA045	Algebra y Geometría/Algebra		22	12	34	11	8	19	7	1	8	4	3	7	50%	67%	56%	32%	8%	24%	18%	25%	21%
2010	2do	MA046	Análisis Matemático I		17	17	34	4	5	9	4	4	8	9	8	17	24%	29%	26%	24%	24%	24%	53%	47%	50%
2010	1cr	IF002	Expresión de Problemas y Algoritmos		21	0	21	12	0	12	1	0	1	8	0	8	57%	0%	57%	5%	0%	5%	38%	0%	38%
2010	1cr	S1008/IF001	Introducción a la Computación/Elementos de informatica		25	2	25	11	1	12	6	1	7	6	0	6	48%	50%	48%	28%	50%	28%	28%	0%	24%
2010	2do/1cr 2	S1009	Introducción al Análisis de Sistemas		0	1	1	0	1	1	0	0	0	0	0	0	0%	100%	100%	0%	0%	0%	0%	0%	0%
2010	2do	S1001/IF005	Algorítmica y Programación	S1008 /IF001	12	1	13	7	0	7	1	1	2	4	0	4	58%	0%	54%	8%	100%	15%	33%	0%	31%
2010	2do	MA008	Elementos de Lógica y Matemática Discreta		17	7	24	10	5	15	3	1	4	4	1	5	59%	71%	63%	18%	14%	17%	24%	14%	21%
2009	1cr	MA001	Algebra y Geometría		21	10	31	5	1	6	16	9	25	0	0	0	24%	10%	19%	76%	90%	81%	0%	0%	0%
2009	1cr	MA002	Análisis Matemático I		21	14	35	3	1	4	18	15	33	0	0	0	14%	7%	11%	86%	93%	89%	0%	0%	0%
2009	1cr	S1005	Introducción a la Computación		21	5	26	7	3	10	10	0	10	4	2	6	33%	60%	38%	48%	0%	38%	19%	40%	23%
2009	2do	S1009	Introducción al Análisis de Sistemas		12	5	15	11	2	13	1	0	1	0	1	1	92%	67%	87%	8%	0%	7%	0%	33%	7%
2009	2do	S1001	Algorítmica y Programación	S1008	6	3	9	6	2	8	0	0	0	0	1	1	100%	67%	89%	0%	0%	0%	0%	33%	11%
2009	2do	MA008	Elementos de Lógica y Matemática Discreta		12	7	19	4	4	8	4	1	5	4	2	6	33%	57%	42%	33%	14%	26%	33%	29%	32%
2008	1cr	MA001	Algebra y Geometría		15	14	29	5	4	9	4	2	6	6	8	14	33%	29%	31%	27%	14%	21%	40%	57%	48%
2008	1cr	MA002	Análisis Matemático I		16	21	37	2	6	8	5	5	8	9	12	21	13%	29%	22%	31%	14%	22%	56%	57%	57%
2008	1cr	S1005	Introducción a la Computación		16	8	24	6	2	8	6	1	7	4	5	9	38%	25%	33%	38%	13%	29%	25%	63%	38%
2008	2do	S1009	Introducción al Análisis de Sistemas		9	6	15	7	2	9	2	4	6	0	0	0	78%	33%	60%	22%	67%	40%	0%	0%	0%
2008	2do	S1001	Algorítmica y Programación	S1008	5	8	13	5	3	8	0	2	2	0	3	3	100%	38%	62%	0%	25%	15%	0%	38%	23%
2008	2do	MA008	Elementos de Lógica y Matemática Discreta		9	9	18	4	1	5	4	4	8	1	4	5	44%	11%	28%	44%	44%	44%	11%	44%	28%
2007	1cr	MA001	Algebra y Geometría		21	12	33	3	7	10	9	3	12	9	2	11	14%	58%	30%	43%	25%	36%	43%	17%	33%
2007	1cr	MA002	Análisis Matemático I		21	17	38	3	4	7	3	5	8	15	8	23	14%	24%	18%	14%	29%	21%	71%	47%	61%
2007	1cr	S1005	Introducción a la Computación		21	4	25	5	2	7	4	1	5	12	1	13	24%	50%	28%	19%	25%	20%	57%	25%	52%
2007	2do	S1009	Introducción al Análisis de Sistemas		7	5	12	5	3	8	1	0	1	1	2	3	71%	60%	67%	14%	0%	8%	14%	40%	25%
2007	2do	S1001	Algorítmica y Programación	S1008	5	3	8	5	2	7	0	0	0	0	1	1	100%	67%	88%	0%	0%	0%	0%	33%	13%
2007	2do	MA008	Elementos de Lógica y Matemática Discreta		8	4	12	5	3	8	4	0	4	1	1	2	38%	75%	50%	50%	0%	33%	13%	25%	17%
2006	1cr	S1005	Estructura de datos y algoritmos	MA008 y S1001	7		7	4		4	1		1	2		2	57%	0%	57%	14%	0%	14%	29%	0%	29%
2006	1cr	S1005	Arquitectura de los sistemas de Computo	S1008	15		15	5		5	4		4	4		4	33%	0%	33%	33%	0%	33%	33%	0%	33%
2009	1cr	S1005	Estructura de datos y algoritmos	MA008 y S1001	4	5	9	3	2	5	0	0	0	1	3	4	75%	40%	56%	0%	0%	0%	25%	60%	44%
2009	1cr	S1005	Arquitectura de los sistemas de Computo	S1008	6	9	15	1	2	3	3	4	7	2	3	5	17%	22%	20%	50%	44%	47%	33%	33%	33%
2010	1cr	S1005	Estructura de datos y algoritmos	MA008 y S1001	6	6	12	1	4	5	0	0	0	5	2	7	17%	67%	42%	0%	0%	0%	83%	33%	58%
2010	1cr	S1005	Arquitectura de los sistemas de Computo	S1008	9	11	20	2	2	4	5	5	8	1	6	7	22%	18%	21%	56%	45%	42%	15%	55%	37%
2011	1cr	IF005	Algorítmica y Programación (ex S1005)	MA008 y IF003	8	3	11	4	1	5	0	0	0	4	2	6	50%	33%	45%	0%	0%	0%	50%	67%	55%
2011	1cr	IF006	Arquitectura de Computos (ex S1003)	IF001	15	5	20	1	1	2	9	3	12	5	1	6	7%	20%	10%	60%	60%	60%	33%	20%	30%
2011	1cr	IF004	Sistemas y Organizaciones		9		9	6		6	0		0	3		3	67%	0%	67%	0%	0%	0%	33%	0%	33%

Figura 4. Datos sobre las materias de la Fac. de Ingeniería

"Simulador de Redes Virtuales para Laboratorios de Sistemas RVL"

Gibellini, Fabian Alejandro; Ignacio José Sanchez Balzaretti; Germán Parisi
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Laboratorio de Ingeniería en Sistemas de Información
Uladislao Frias S/N, Ciudad Universitaria, Córdoba
fgibellini@bbs.frc.utn.edu.ar, ignacio@bbs.frc.utn.edu.ar, germannparisi@gmail.com

Abstract

"*El Simulador de Redes virtuales para Laboratorios de Ingeniería en Sistemas de Información de UTN, Es una solución con entornos virtualespensada para docentes de cátedras y alumnos relacionadas con las redes informáticas y la seguridad informática, la cual busca suplir el gran vacío existente de simuladores libres y que se adapten al uso de las carreras de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional y que tengan interfaces amigables y en nuestro idioma, basándonos en GNU/Linux y en especial en la distribución de Debian GNU/Linux 6.0. Este simulador será de utilidad para uso educativo y se propone también acompañar al simulador las prácticas que se pueden realizar en los horarios de Laboratorio de las materias que den uso al mismo, acompañando al cumplimiento de los objetivos de aprendizajes de las cátedras involucradas."*

Palabras Claves

Simulador de Redes, Máquinas Virtuales, Simulador de servidores, Redes de Información, Seguridad informática, GNU/Linux, Sistemas operativos, Laboratorio de sistemas.

Introducción

Este simulador trabaja con Máquinas virtuales factible de simular routers, servidores, estaciones de trabajo, switch básicos permitiendo la conformación de topologías de diferentes dificultades. Utilizando el lenguaje Python hemos concretado la primera interfaz amigable para los estudiantes que permite hacer laboratorios de enrutamiento, configuración de servicios, seguridad, análisis de protocolos y otros.

Para poder llegar a la primer versión del simulador se realizó un análisis de campo, donde se pudo detectar la falta de una herramienta libre y que dé el soporte necesario para el desarrollo de las actividades prácticas de las materias que hacen uso en la UTN, entre ellas Redes de Información (RIN) materia del cuarto año de la carrera de Ingeniería en Sistemas de Información de la UTN FRC. Posteriormente se realizó una investigación y selección de todos los simuladores que podrían permitir realizar las prácticas de RIN libres y propietarias. Entre ellas Common Open Research Emulator (CORE) de la Naval Research Laboratory Washington [1] y Marionnet a virtual network laboratory de la Université de Paris 13 [2], la cual la descartamos por no estar traducido a nuestro idioma y por ultimo Packet tracer la cual solo nos permite configurar equipos de propiedad CISCO y el cual solo puede ser usado por las academias de CISCO y además solo permite configuraciones de equipos de networking y no se pueden configurar servidores.

Los simuladores que se detallaron anteriormente tienen una serie de ventajas tales como tener interfaces muy amigables y la posibilidad de realizar diferentes tipos de topología rápidamente y desventajas tales como no poder adquirir licencias, no estar en nuestro idioma nativo, o tener errores básicos.

Objetivos del Simulador:

- Contar con prácticas asociadas a las materias de UTN FRC y que las mismas puedan realizarse en los laboratorios y/o de modo externo a las instalaciones de la Institución.
- Lograr un simulador de redes informáticas para que pueda ser implementado en cátedras de otras regionales.
- Realizar una interfaz amigable para la generación de los prácticos por parte de los profesores.
- Facilitar al estudiante la realización de prácticas simulando varios dispositivos en una interfaz amigable y con posibilidad de migrarlo de un equipo a otro.
- Implementar configuraciones realizadas en el simulador en infraestructura real.
- Capacitar a los docentes de la cátedra involucrada.
- Realizar encuestas de satisfacción para evaluar con el tiempo la aceptación del simulador en la Facultad Regional Córdoba.

Elementos del Trabajo y metodología

De acuerdo a la investigación exploratoria previamente realizada sobre, las aplicaciones propietarias y de código abierto, que permiten simular redes informáticas, disponibles a la fecha en mercado; y luego de analizar las prestaciones de las mismas, y tras laboriosas pruebas de comparación de las citadas prestaciones y adecuación de las mismas a los requerimientos del proyecto, se desarrolla un el simulador de redes virtuales.

Con el propósito de analizar, estudiar e investigar con que herramienta correr el simulador, se trabaja con una máquina virtual, que se ejecuta desde distintas máquinas por medio de la aplicación VirtualBox [3] (ver Figura 1). La máquina virtual tiene instalada una distribución DebianSqueeze.

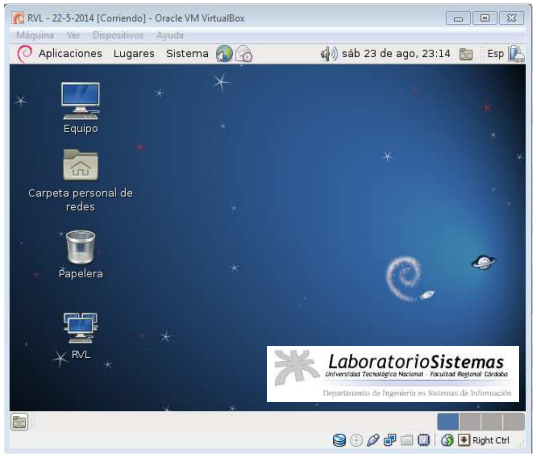


Figura 1 - Máquina Virtual

La metodología utilizada en este proyecto de investigación está basado en el método empírico, de orientación cuantitativa observacional en la toma, análisis y asociación de los datos cuantitativos arrojados por las aplicaciones bajo estudio (Bunge; 1998) [4], (Barchini; 2005) [5]. Pues, la informática utiliza los métodos empíricos, que toman conocimiento del objeto mediante el uso de la experiencia.

Para el desarrollo de los programas se tendrá en cuenta las normas CMM y ISO/IEC 12207 [6] / ISO/IEC 25000:2500 [7] e ISO 9001[8] para asegurar la mejor calidad posible y capacitar a los integrantes en las mejores prácticas de Ingeniería de Software. Se utilizarán metodologías ágiles del tipo SCRUM (1986) [9] para el desarrollo del Software.

Se decidió que la aplicación funcionará en una máquina virtual GNU/Linux Debian 6.0, el simulador será desarrollado con el lenguaje de programación Python [10] con GTK+ [11] y para el diseño de las topologías GTK Canvas [12].

Para el armado de la topología por parte del docente, hacemos uso del editor de diagramas DIA [13]. Herramienta sencilla de uso y aprendizaje.

Resultados

En base a las pruebas realizadas con la máquina virtual con un GNU/LINUX Debian 6.0 anteriormente mencionada, se seleccionó esta herramienta para desarrollar el sistema RVL porque en la misma se pueden correr varias máquinas virtuales con User Mode Linux [14] sin perder performance del equipo local.

En las primeras pruebas de desarrollo se pudo definir una distribución de la interfaz para el estudiante, el mismo puede observar la topología propuesta por el docente y las consignas del práctico a seguir y cada dispositivo que interviene en la topología (ver Figura 2).

Últimos Avances:

- se realizaron prácticas que permitieron descubrir problemas con herramientas de apt-get y dhcp.
- Se agregaron herramientas como wireshark y nmap y un front end para nmap,
- se realizaron las actualizaciones a la última versión de Debian Squezze
- Se agregaron servicios de VPN y tshark.
- Se configuro la consola para que mc y mcedit funcione correctamente. Correcciones para que el hostname no falle.
- Se creó un repositorio en la imagen principal de la máquina virtual.
- Se implementaron múltiples tap para poder realizar análisis de protocolos y permitir la interconexión de topologías simuladas por distintos estudiantes
- Se cambió la metodología de una lista enlazada a un grafo para mejorar la performance de tiempos de desarrollo.
- Se realizó un parse de comandos, que permite el armado del archivo de configuración desde la línea de comandos sin necesidad de ingresar a la interfaz gráfica.
- Se comenzaron estudios de desarrollo con CANVAS GTK para la interfaz gráfica para la generación de topologías
- Se realizaron modificaciones para que las ventanas realicen un resize al hacer ampliaciones de los marcos de las ventanas.
- Se incorporó al RVL interfaz de selección de prácticos
- Se definió la confección de una API para mejorar el código
- Se incorporaron 7 prácticos, los cuales fueron desarrollados en conjunto con la cátedra de Redes de Información de la UTN FRC.
- Se capacito a docente para el uso del simulador
- Se llevó a cabo una reunión con la cátedra de redes de información, para colaborar con la planificación de las prácticas.

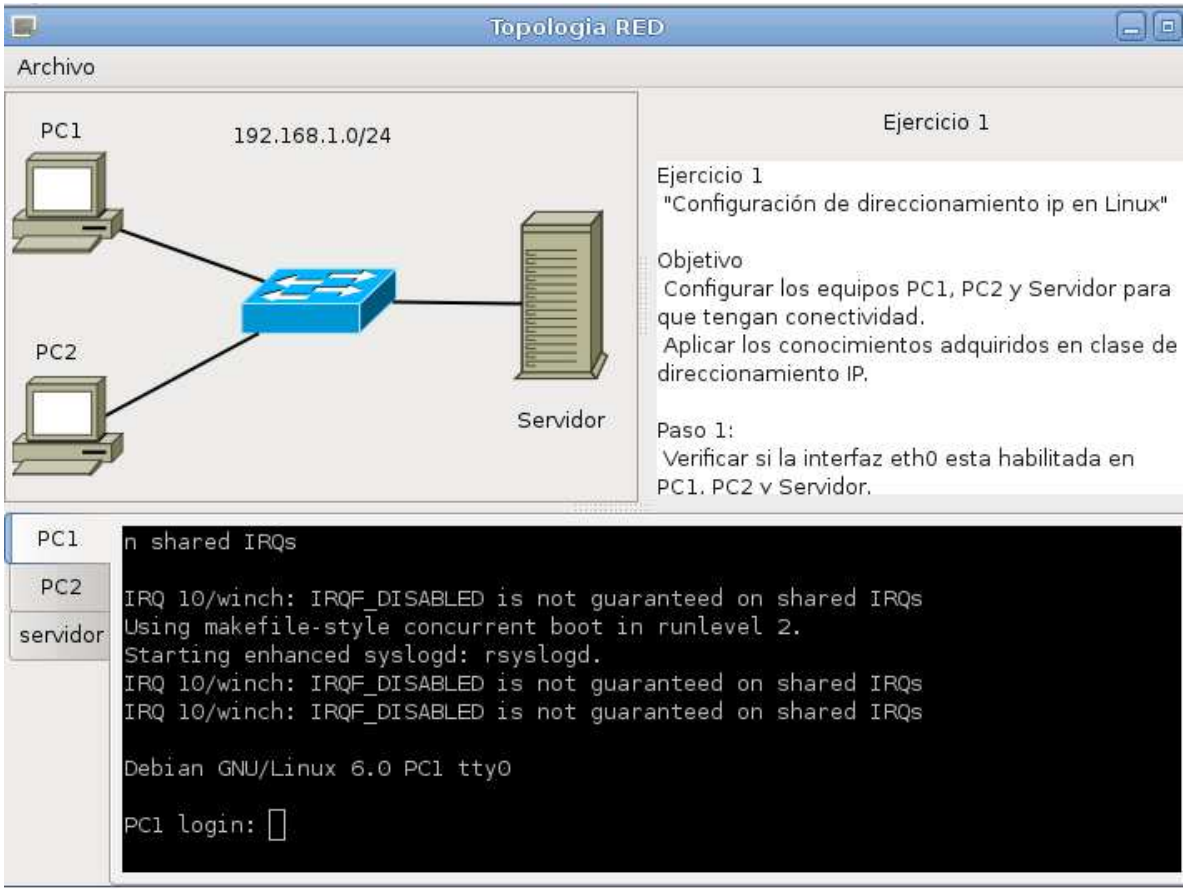


Figura 1: Interfaz del Simulador

Discusión

El Laboratorio de Sistemas ha realizado muchos desarrollos siempre con el objetivo de brindar un mejor servicio a las cátedras y estudiantes de la carrera de Ingeniería en Sistemas de Información, adaptándose a la infraestructura del momento y a los alcances tecnológicos que tienen nuestros usuarios, por eso se promueve el uso de código abierto, ya que el mismo está a disposición de quien lo requiera.

El simulador RVL, ha sido pensado con tal fin, para que los estudiantes y los docentes puedan realizar prácticas y si fuese necesario adaptarlo a sus necesidades futuras.

La continuidad del desarrollo de este simulador se justifica teniendo en cuenta que para la realización de estas prácticas se debería contar con una cantidad de equipos de redes a disposición de los estudiantes que no es factible para ninguna Universidad Pública.

Se han dado clases prácticas con el simulador, en la cual los docentes han demostrado satisfacción de uso.

Conclusión

Este simulador contribuye a la formación y crecimiento de los estudiantes y docentes.

Durante el período de desarrollo del simulador se contempla la formación de alumnos de la carrera de Ingeniería en Sistemas de Información.

La concreción de esta aplicación favorece a las cátedras de redes de información de la Universidad Tecnológica Nacional ya que la misma al estar desarrollada con código abierto, permitirá ser adaptada a las necesidades locales de cada Regional.

Referencias

[1] Common Open Research Emulator (CORE) de la Naval Research Laboratory Washington

http://cs.itd.nrl.navy.mil/work/core/

[2] Marionnet a virtual network laboratory de la Université de Paris 13

http://www.marionnet.org/EN/whoswho.php

[3] Máquina virtual VirtualBoxhttps://www.**virtualbox**.org/

[4] Bunge, M. 1998. La ciencia su Método y su Filosofía. Editorial Siglo Veinte. Buenos Aires.

[5] Barchini, 2005. G. Métodos "I+D" de la Informática. Universidad Nacional de Santiago del Estero, Argentina. 2005.

[6][7] www.iso.org

[8] NORMA ISO 9001(2000). Calidad en el desarrollo de software.

[9] Alonso Alvarez Garcia, 2012, METODOS AGILES Y SCRUM. ANAYA MULTIMEDIA.

[10] Python Programming Language http://www.python.org/

[11] The GTK+ http://www.gtk.org/

[12] CANVAS CAIRO http://stackoverflow.com/questions/8608686/python-gtk-canvas

[13] Editor de Diagramas Día

http://dia-installer.de/index.html.es

[14] User Mode Linux

http://user-mode-linux.sourceforge.net/

* Modalidad Académica 2014 de RIN UTN FRC. http://www.institucional.frc.utn.edu.ar/sistemas/noticias/ACA/Modalidades/2014/rin__2014.pdf

Bibligrafía utilizada para la investigación y el desarrollo

1. TRANSMISIÓN DE DATOS Y REDES DE COMUNICACIONES. 4ta. edición". Behrouz A. Forouzán. Editorial Mc Graw Hill. 2006.

2. "COMUNICACIONES Y REDES DE COMPUTADORAS 7ma. Edición". Williams Stallings. Prentice Hall 2004.

3. "REDES DE COMPUTADORAS Cuarta Edición". Andrew Tanenbaum. Prentice Hall 2003

4. REDES DE COMPUTADORES. Un Enfoque Descendente Basado en Internet 2da. Edición". Kurose - Ross. Addison Wesley. 2004.

La visualización como estrategia pedagógica para la enseñanza del Análisis Matemático con GeoGebra

Lamas, Roberto Daniel; García, Adelina
Universidad Nacional de Jujuy, Facultad de Ingeniería

Abstract

La siguiente experiencia se realizó con el propósito de innovar en la forma de enseñar el Análisis Matemático a los alumnos de Primer Año de la Facultad de Ingeniería. Teniendo en cuenta que el estudio de la matemática, se caracteriza por su precisión, por su carácter formal y abstracto, por su naturaleza deductiva y por su organización axiomática, requiere de didácticas especiales como ser la resolución de problemas, el aprendizaje significativo y la visualización matemática.

Se emplea la visualización como herramienta pedagógica para la enseñanza del análisis matemático ya que nos ofrece enormes posibilidades. A los docentes, nos permite compartir conocimientos desde distintas perspectivas y a los estudiantes, la oportunidad de complementar, abordar y profundizar los conceptos a través de diferentes puntos de vistas (analítico, visual, evaluativo, aplicativo).

Para tal fin se diseñaron aplicaciones específicas (applets) desarrolladas con GeoGebra utilizadas en las clases teóricas presenciales para la presentación del tema Funciones en variable real y Teoremas de derivabilidad. Materiales que están a disposición de los alumnos en el aula virtual de la cátedra.

Como resultado de esta innovación los estudiantes mejoraron su actitud hacia el aprendizaje del análisis matemático al ser motivados a incorporar el uso de la tecnología a sus procesos de enseñanza y aprendizaje, propiciando una mejor comprensión de los conceptos teóricos.

Este camino iniciado con la integración de las TIC en las clases presenciales por medio de applets será profundizado en forma gradual a los todos los temas que son de difícil comprensión de la materia.

Palabras Clave
Visualización, GeoGebra, Análisis Matemático, TIC, Resolución de problemas, Aprendizaje significativo

Introducción

Las nuevas tecnologías de la información y la comunicación se han convertido en una herramienta insustituible y de indiscutible valor y efectividad en el manejo de la información con propósitos didácticos [1].

La incorporación de las nuevas tecnologías debe constituir una nueva oportunidad para transformar la docencia universitaria y optimizar la calidad de los aprendizajes de los alumnos, aunque por sí mismas las TIC no mejoraran la enseñanza.

Juega con ventaja el que posee información, o bien conoce como acceder a ella y cómo interpretarla, el gran desafío de las instituciones educativas será conseguir que sus alumnos dominen los conocimientos y también las operaciones cognitivas vinculadas a los aprendizajes para que éstos se vuelvan permanentes y propicien nuevos aprendizajes en situaciones diversas[2].

En este nuevo escenario, el alumno se enfrenta al aprendizaje con un nivel de desarrollo (biológico, motriz, emocional, social, cognitivo y de comunicación y lenguaje), con un nivel de competencia curricular y un Estilo de Aprendizaje que ha de generarles competencias específicas (Saber hacer), competencias académicas (Conocer) y competencias para la vida (Convivir) [3].

Aprender a aprender es una necesidad para nuestro desarrollo como seres humanos e implica la capacidad de reflexionar en la forma en que se aprende y actuar en consecuencia, autoregulando el propio proceso de aprendizaje mediante el uso de estrategias flexibles y apropiadas que se transfieren y adaptan a nuevas situaciones, tal como demanda un mundo globalizado [4].

Diferentes autores definen las ideas y estrategias que impulsarán el proceso enseñanza-aprendizaje, Cabero señala las siguientes estrategias didácticas [5]:

- Cada vez se asume que el aprendizaje es un proceso activo y no pasivo, donde la participación del estudiante es clave en el proceso.
- Se asume que el gran desafío con el cual se enfrenta el estudiante, y los profesores, es que los estudiantes a su nivel deben producir conocimiento y no simplemente reproducirlo.
- Se admite que el aprendizaje es un proceso social, donde el sujeto en la interacción con sus compañeros y con el resto de variables curriculares, modifica su estructura cognitiva.
- La importancia del aprendizaje colaborativo, es integrado, contextualizado y situado en función

de los problemas, necesidades y ubicación espacial del sujeto.
- Que debemos movilizar diferentes sistemas simbólicos, para desarrollar e impulsar los diferentes tipos de inteligencias.
- Que la educación será más personalizada, donde se respeten los ritmos, los estilos e inteligencias múltiples de cada alumno.
- Y que su evaluación no debe referirse únicamente a los productos, sino a los procesos que lo han generado, y que no debe limitarse solo a uno de los actores del proceso sino a todos

En definitiva, las TIC pueden ser un factor que ayuden a construir y desarrollar un modelo de enseñanza más flexible, donde prime más la actividad y la construcción del conocimiento por parte del alumnado a través de una gama variada de recursos que a la mera recepción pasiva del conocimiento a través de unos apuntes y/o libros. Este es, y será, uno de los retos pedagógicos de la docencia universitaria a corto y medio plazo [6].

Los roles que los docentes tendrán que asumir son:

- Consultores y facilitadores de información.
- Facilitadores de aprendizaje.
- Diseñadores de medios.
- Moderadores y tutores virtuales.
- Evaluadores continuos y asesores.
- Orientadores [7].

Dentro de este contexto universitario, la matemática es una de las disciplinas que juega un papel importante en la formación intelectual del alumno e incide directamente sobre las estructuras mentales, requiere un proceso de enseñanza y aprendizaje adecuado que facilite en el alumno un desarrollo lógico matemático apropiado, pero que a la vez satisfaga sus necesidades [8].

La matemática, se caracteriza por su precisión, por su carácter formal y abstracto, por su naturaleza deductiva y por su organización a menudo axiomática. Sin embargo, tanto en la génesis histórica como en su apropiación individual por los alumnos, la construcción del conocimiento matemático es inseparable de la actividad concreta sobre los objetos, de la intuición y de las aproximaciones inductivas activadas por la realización de tareas y la resolución de problemas particulares. La experiencia y comprensión de las nociones, propiedades y relaciones matemáticas a partir de la actividad real es, al mismo tiempo, un paso previo a la formalización y una condición necesaria para interpretar y utilizar correctamente todas las posibilidades que encierra dicha formalización. El aprendizaje y la enseñanza deben tener en cuenta que es natural que los alumnos tengan dificultades y cometan errores en su proceso de aprendizaje y que se puede aprender de los propios errores. Esta es la posición de las teorías psicológicas constructivistas sobre el aprendizaje de las matemáticas, las cuales se basan a su vez en la visión filosófica sobre la matemática conocida como constructivismo social [9].

Tipos de aprendizaje matemático

Aprendizaje significativo

Según Ausubel el aprendizaje del alumno depende de la estructura cognitiva previa, que ya tiene, y el establecimiento de relación con la nueva información que recibe. El aprendizaje significativo ocurre cuando la nueva información "se conecta" con un concepto relevante ya existente en la estructura cognitiva, esto implica que, las nuevas ideas, conceptos y proposiciones pueden ser aprendidos significativamente en la medida en que otras ideas, conceptos o proposiciones relevantes estén adecuadamente claras y disponibles en la estructura cognitiva del individuo y que funcionen como un puente. Así se produce la asimilación. Se trata de un aprendizaje por descubrimiento, se realiza frente a lo que se denomina aprendizaje por recepción. En el aprendizaje por recepción el alumno recibe los contenidos que debe aprender en su forma final, acabada; no necesita realizar ningún descubrimiento más allá de la comprensión y asimilación de los mismos de manera que sea capaz de reproducirlos cuando le sea requerido. El aprendizaje por descubrimiento implica una tarea distinta para el alumno; en este caso el contenido no se da en su forma acabada, sino que debe ser descubierto por él. Este descubrimiento o reorganización del material debe realizarse antes de poder asimilarlo; el alumno reordena el material adaptándolo a su estructura cognoscitiva previa hasta descubrir las relaciones, leyes o conceptos que posteriormente asimila [10].

Aprendizaje basado en problemas.

El aprendizaje basado en problemas (ABP) puede definirse como un proceso de indagación que resuelve preguntas, curiosidades, dudas, incertidumbres, sobre fenómenos complejos de la vida. Se trata de resolver problemas mediante la indagación de los alumnos. Los objetivos del aprendizaje basado en problemas (ABP) se centran en desarrollar habilidades de pensamiento de orden superior, presentando al alumno problemas y casos auténticos y complejos. Este enfoque ofrece un contexto más real para el aprendizaje e involucra a los alumnos en tareas reales. A través del proceso de trabajar en equipo (de forma autónoma y/o guiados por el docente), articular teorías, crear hipótesis y discutir de forma crítica las ideas de otros, los alumnos alcanzan un nivel mucho más profundo en la comprensión de los problemas. Las estrategias de aprendizaje auto-dirigido que se utilizan en el ABP pueden servir para estimular el aprendizaje permanente [11].

El ABP permite:

- Compartir fuentes de información discutiendo críticamente sus contenidos.
- Consultar a expertos para dar respuesta a las necesidades de aprendizaje detectadas por los alumnos.
- Enfatizar la identificación y comprensión de los principios y no memorizar detalles.

Como método de aprendizaje integrador de conocimiento, es una herramienta fundamental en la docencia, desde una perspectiva cognitivista, que pretende:

- Promover la responsabilidad del propio aprendizaje.
- Desarrollar aprendizaje profundo de lo fundamental, lo relevante.
- Desarrollar habilidades para las relaciones interpersonales, intelectuales, emocionales y de trabajo en grupo.
- Desarrollar el razonamiento eficaz y la creatividad.
- Detectar faltas de conocimientos y habilidades y orientar la mejora.
- Desarrollar capacidad para enfrentar situaciones ambiguas.
- Desarrollar habilidades para búsqueda de información y tener capacidad crítica.
- Desarrollar capacidad para identificar y explicar los aspectos importantes de cada problema.
- Apertura para aprender de los demás y compartir sus aprendizajes [12].

El alumno adquiere el máximo del protagonismo al identificar sus necesidades de aprendizaje y buscar el conocimiento para dar respuesta al problemas planteado, lo que volverá a generar nuevas necesidades de aprendizaje [11].

Conocimiento matemático y visualización.

Entre las verdades comunes aportadas por el empirismo filosófico se encuentra que la percepción y observación son fuentes privilegiadas del conocimiento humano. Cada ser humano lleva en sí mismo, en sus facultades de percepción sensorial, una vía de acceso al conocimiento. Los sentidos son los cauces por los que los seres humanos recibimos información del exterior. Por ello, resulta adecuado considerar la intervención de los sentidos en la transmisión, adquisición y construcción de cualquier tipo de conocimiento. El conocimiento matemático se recibe y se transmite, prioritariamente, mediante dos canales de información: el auditivo y el visual (y, de manera complementaria, por el tacto). Enunciados verbales y organizaciones visuales gráficas o simbólicas son los medios usados con mayor frecuencia en la emisión, transmisión y recepción de conocimiento matemático. De ahí que los dos sentidos mencionados tengan una importancia primordial en los procesos que intervienen en la enseñanza y aprendizaje de las matemáticas. Y esto es así por la función esencial de los enunciados declarativos y las expresiones gráficas o simbólicas en la formación y comunicación del conocimiento matemático. Además de comunicar nuestro conocimiento sobre conceptos y operaciones también necesitamos pensar sobre tales objetos; en este caso formamos imágenes mentales que se denominan representaciones internas. No todas las imágenes mentales consideran características figurativas o gráficas del concepto; cuando en una representación mental predominan los componentes figurativos o gráficos hablamos de visualización. La visualización es usual cuando imaginamos objetos geométricos familiares; así ocurre cuando pensamos en las figuras de un cuadrado, un círculo o un triángulo equilátero. Menos usual es tener una imagen gráfica de otros conceptos (como el de término general de una sucesión), o de algunos procedimientos (como divisibilidad de polinomios). Las personas con facilidad para formarse imágenes mentales gráficas emplean éstas en sus razonamiento matemáticos; se dice que son buenos visualizadores. Lo que caracteriza a una imagen mental es hacer posible la evocación de un objeto sin que él mismo esté directamente presente. Cuando se usan las representaciones gráficas de conceptos matemáticos como herramientas para interpretar conceptos o resolver problemas, la visualización no es un fin en sí misma sino un medio para llegar a su comprensión o resolución. Se trata de expresar alguna propiedad específica de un concepto o alguna relación importante para la resolución de un problema a través de un diagrama, un dibujo o una gráfica. Se usa el término de pensamiento visual para describir los aspectos del pensamiento matemático que están basados o se pueden expresar en términos de imágenes mentales [13].

Zimmermann considera que se puede adquirir habilidad respecto a la visualización y divide estratégicamente esta habilidad en cinco categorías, a las que considera como objetivos de aprendizaje. Tales objetivos son:

Objetivos básicos

Entender el álgebra y la geometría como lenguajes alternativos para expresar las mismas ideas matemáticas.

Entender la información matemática implícita en una representación gráfica.

Extraer información de un diagrama así como utilizar las gráficas y los diagramas para representar información matemática.

Objetivos funcionales

Contemplan la capacidad para entender qué conceptos están representados en un diagrama y utilizarlos para realizar demostraciones y para resolver problemas.

Objetivos generales

Son aquellos aspectos de la visualización que tienen amplia aplicación en distintas áreas de las matemáticas; incluye aquí la habilidad para realizar estimaciones y

aproximaciones en un contexto geométrico, reconocer y explicar simetrías, apreciar la periodicidad, captar la similitud, entender y reconocer patrones, entender trasformaciones geométricas, o conseguir un amplio repertorio de imágenes visuales.

Objetivos relacionados específicamente con el cálculo
Incluyen la habilidad de entender conceptos como el de diferenciación, visualizar elementos infinitesimales en figuras geométricas y visualizar superficies y figuras de tres dimensiones.

Objetivos de alto nivel
Hacen referencia a la habilidad de apreciar la belleza de las matemáticas, interpretar fenómenos y experiencias visuales de la vida real, explorar y descubrir visualmente ideas matemáticas [14].

Por otra parte, los sujetos que realizan un dibujo de un concepto lo aprenden mejor que aquellos que sólo conocen su definición verbal. Una explicación admitida de este fenómeno es que la generación activa de una imagen visual por parte del que aprende facilita más el aprendizaje que la simple presencia de la imagen visual. Es una hipótesis generalmente admitida que mejorando la educación visual en matemáticas aumenta la intuición y se proporciona al sujeto una mayor capacidad de entendimiento [15].

Hay un alto consenso entre investigadores y especialistas relativos a que el desarrollo de las capacidades que caracterizan el pensamiento visual proporciona a los alumnos nuevos caminos para pensar y hacer matemáticas.

Software educativo GeoGebra

La utilización de software libre ha surgido con gran fuerza en el contexto universitario, en el campo de la matemática, el software GeoGebra es uno de los más utilizados debido al carácter abierto de su código, lo cual permite que esté disponible de manera libre y gratuita para todas las plataformas. GeoGebra no es solamente geometría (Geo), también es álgebra (Gebra), cálculo, análisis y estadística. La gran variedad de opciones que ofrece hacen que sea utilizado para proponer a los estudiantes sencillas tareas de investigación y experimentación, las cuales en su mayoría, no requerirán demasiados conocimientos técnicos, solamente algunas herramientas básicas y comandos.

Algunas de las características del GeoGebra son: es un software libre diseñado para ser utilizado en los procesos de enseñanza y aprendizaje de las matemáticas en todos los niveles educativos. Tiene la característica de integrar las capacidades de los programas de geometría dinámica con las capacidades de los programas de cálculo simbólico y de las hojas de cálculo. Además, permite el uso simultáneo de los sistemas de representación simbólico (algebraico/numérico) y gráfico en tres ventanas simultáneas (algebraica, gráfica y hoja de cálculo). Las manipulaciones o modificaciones de objetos geométricos o matemáticos en una de las ventanas, tiene de forma inmediata repercusión en las ventanas restantes. Además, cuenta con una interface "amigable" que facilita su uso, lo cual lo convierte en un instrumento útil para la enseñanza de las matemáticas [16].

Otra característica del GeoGebra es la diferenciación entre dibujo y figura, la cual es útil para describir la forma en la cual los estudiantes interpretan las representaciones realizadas en la pantalla del ordenador.

Construcción dibujo: realizada de la misma forma que los dibujos hechos con lápiz y papel, es decir, dibujo a mano alzada. Por ejemplo, si un estudiante construye un rectángulo basándose únicamente en elementos de medida, las propiedades del objeto construido no se mantienen al desplazar uno de los vértices de la figura. Este objeto se considera como un dibujo y no mantiene las propiedades geométricas de la figura.

Construcción figura: realizada utilizando las propiedades euclídeas que el programa ofrece a través de las herramientas propias o bien creadas por el usuario (herramientas personales). Por ejemplo, para trazar una recta perpendicular a un segmento debemos utilizar la herramienta Recta Perpendicular [17].

En este caso, el estudiante debe conocer las herramientas necesarias de construcción pero también las propiedades geométricas del objeto, lo cual requiere conocimiento del software y de las matemáticas [18].

Una característica más del software, es la necesidad de definir explícitamente los objetos geométricos, es decir, los objetos que se construyen con GeoGebra tienen que ser definidos explícitamente para que sean reconocidos como tales y por tanto manejables por el programa. Por ejemplo: cuatro segmentos consecutivos y cerrados no definen un cuadrado para el programa, y por tanto, no es posible calcular su área, salvo que sea definido o construido con la herramienta correspondiente. Finalmente, el Principio de Geometría Dinámica significa que las construcciones realizadas con GeoGebra pueden ser modificadas dinámicamente a través del arrastre de objetos básicos o bien introduciendo deslizadores. Las propiedades o relaciones geométricas establecidas explícitamente al construir la figura a través de las herramientas disponibles, permanecen al arrastrarse los objetos. Este principio hace posible la devolución que hace el arrastre en las figuras del programa: se conservan las propiedades explícitas usadas por el usuario al realizar la figura y además se hacen visibles al usuario todas aquellas propiedades o relaciones de la figura que se deducen de las anteriormente explicitadas por el usuario [17].

GeoGebra no requiere inversión para su uso y su continua evolución hace que surjan nuevas versiones o se incorporen nuevas herramientas, con las cuales se incrementa su potencia y sus posibilidades didácticas. Por

lo expuesto anteriormente, el GeoGebra se está convirtiendo en un recurso imprescindible para los docentes que quieran incorporarlo a su práctica.

Experiencia

En esta experiencia se presenta la visualización como herramienta pedagógica que contribuye a desarrollar el pensamiento matemático en los alumnos, utilizándola en la enseñanza del Análisis Matemático por medio de aplicaciones específicas para las clases teóricas y prácticas desarrolladas por docentes con el software Geogebra.

La misma se lleva a cabo dentro del Proyecto de Investigación "Análisis y mejoramiento del entorno virtual de enseñanza-aprendizaje como apoyo de las clases presenciales en las cátedras de primer año ". En el mismo están integrados docentes de Análisis Matemático, Algebra y Geometría Analítica y Matemática Discreta de la Facultad de Ingeniería, Universidad Nacional de Jujuy.

La integración de las TIC a la práctica docente permite disponer de recursos que utilizados de forma adecuada, se convierten en una potente herramienta y con funcionalidades interesantes para la enseñanza y aprendizaje de las matemáticas. Su carácter motivador y su eficacia para favorecer metodologías activas y participativas permiten a los estudiantes que se sientan partícipes de su propio aprendizaje. En particular, en el Análisis Matemático se puede trabajar de forma experimental a través del software GeoGebra, al interactuar con objetos matemáticos, construirlos, analizar comportamientos, comprobar propiedades, hacer conjeturas, realizar demostraciones.

A partir del año 2012 la Universidad dispone de una plataforma educativa Moodle, en la cual la cátedra tiene el aula virtual, que funciona como complemento en el dictado de la materia, la misma además de ser un nexo para la comunicación asincrónica con los alumnos y demás profesores, sirve para desarrollar actividades adicionales colaborativas de los alumnos. En dicha aula se habilito una sección especial donde el alumno tiene acceso en forma libre y atemporal a estas aplicaciones para que complemente su conocimiento acerca de estos temas en forma visual y además cuenta con un tutorial para su utilización, acompañado con actividades de aplicación de los temas teóricos desarrollados.

Desarrollo

La experiencia se inicia con applets aplicados a funciones elementales reales en variable real y luego continua con la demostración de Teoremas del Análisis Matemático.

Actividad 1

Objetivo: Que el alumno pueda reconocer y relacionar las funciones con la forma y fórmula correspondiente.

1.1 Así se partió de funciones elementales, considerando en primera instancia la función lineal, la fórmula general de la misma es:

$$y = a*x + b \qquad (1)$$

Donde "a" representa la pendiente de la función lineal y "b" la ordenada al origen, controlados cada una de las mismas por los deslizadores respectivos.

De esta manera al desplazar los deslizadores el alumno aprecia la fórmula correspondiente a dicha recta, permitiéndole relacionar la misma con la gráfica de la función (Fig. 1).

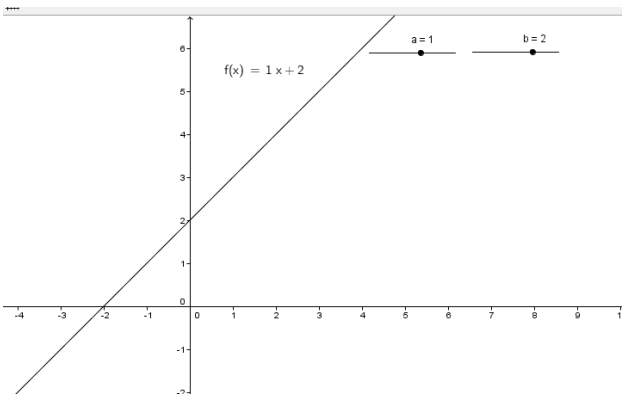


Fig. 1. Función lineal

La actividad que se pidió al alumnado fue la siguiente:

1) Siendo a = 2, b = 2, cual es la ecuación de la recta obtenida?
2) Grafique la recta que representa la función lineal dada por la fórmula es f(x) = – 4 x + 1.
3) Desplace los deslizadores de tal manera que la recta pase por los puntos (0, 4) y (3, 0), cuál es la ecuación de la recta obtenida en el este caso? La pendiente es positiva o negativa?.

1.2 La siguiente función es la función polinómica, consideramos un polinomio de grado cuarto,

$$f(x) = ax^4 + bx^3 + cx^2 + dx + e \qquad (2)$$

Donde a, b, c, d y e son los coeficientes del polinomio.

Se utilizaron cinco deslizadores, cada uno controlando uno de los coeficientes de la función polinómica (Fig. 2)

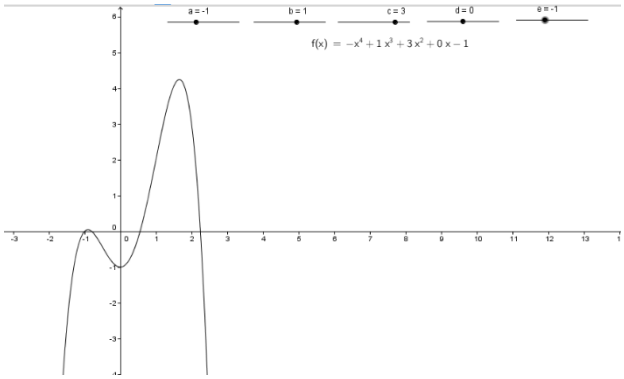


Fig.2. Función polinómica

Así con esta herramienta se le pide al alumno la siguiente actividad:

1) Considerando a = 0, b = 2, c = 3, d= – 2, e= 0, determine cuáles son las raíces gráfica y analíticamente.
2) Qué valores tiene que dar a los deslizadores si desea obtener una parábola que pasa por el origen?
3) Si deseo que las raíces sean –1, 1 y 0, que coeficientes debe tener la función cúbica correspondiente?

1.3 Finalmente se le plantea la función radical cuya fórmula general es:

$$y = \sqrt[n]{x} \qquad (3)$$

Con n entero.
Se utilizaron tres deslizadores, el deslizador "a" controlando el grado de la raíz, el deslizador "b" controlando el desplazamiento horizontal y el deslizador "c" para controlar el desplazamiento vertical (Fig. 3)

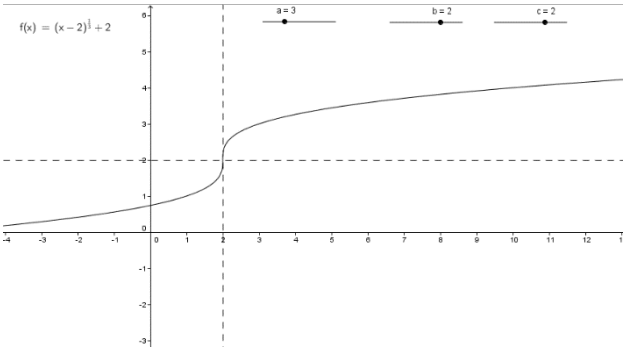


Fig. 3. Función radical

Algunas de las actividades planteadas eran las siguientes:

1) Siendo a = 2, b = 2 y c = –1 , cual es el dominio y la imagen de la función dada?
2) Para que la función tenga por dominio [– 3 , ∞) e imagen [2 , ∞) , que valores debo tomar en los deslizadores ?.
3) Para que la función tenga por dominio todos los reales y esta desplazada 3 unidades a la izquierda y 1 hacia abajo, que valores debo marcar en los deslizadores.

Actividad 2
Otras de las dificultades encontradas por los alumnos es comprender los teoremas dados, en estos deben tener presentes las hipótesis y determinar si se cumple la tesis de los mismos. Consideramos que el problema radica en intentar comprender lo que expresan, al verlos en el pizarrón como algo estático.
Para tal fin se consideraron los teoremas sobre funciones derivables, teorema del valor medio de Lagrange y teorema de Rolle.
El desafío es hacer que el alumno verifique las hipótesis del teorema para luego verificar la tesis.

2.1 Teorema del Valor Medio de Lagrange:
El teorema establece que:
Si f es continua en [a, b] y derivable sobre (a, b), entonces existe un número c en (a, b) tal que;

$$f'(c) = \frac{f(b)-f(a)}{b-a} \qquad (4)$$

Siendo f(a) y f(b) los valores que toma la función en a y b respectivamente y f '(c) el valor que toma la derivada en c.
En este caso las hipótesis a considerar son las siguientes:
I) La función debe ser continua en el intervalo [a, b]
II) La función debe ser derivable en el intervalo (a,b)
Lo que debemos verificar es que debe existir un número c, comprendido en el intervalo (a,b) tal que la pendiente de la recta secante sea igual a la pendiente de la recta tangente.

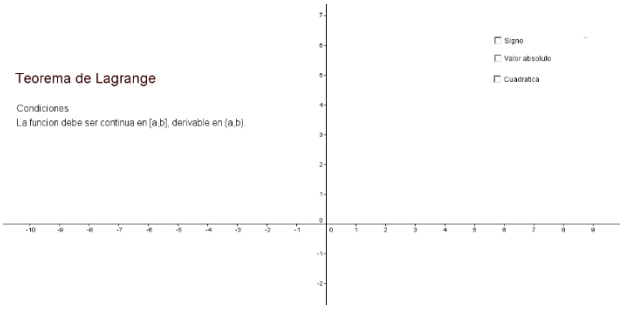


Fig. 4. Pantalla inicial para la aplicación del Teorema de Lagrange.
Se plantea al alumno la elección de tres funciones, con las cuáles tiene que verificar las condiciones del teorema correspondiente, esto lo observamos en la Fig.4, debe

marcar una de las tres opciones que aparecen a la derecha de la pantalla.
Al marcar la primera opción, se trata de la función signo, la cual definimos como

$$sg(x) = \begin{cases} 1 \; si \; x > 0 \\ -1 \; si \; x < 0 \end{cases} \qquad (5)$$

Esta función no es continua en x = 0, por lo tanto al desplazar el deslizador j, cuando marca 0, aparece un mensaje de aviso, en donde se explica que la función no cumple con la condición de continuidad, y por lo tanto no se puede aplicar el teorema (Fig.5).

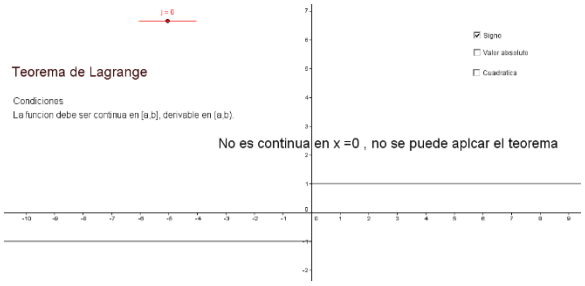


Fig. 5. Mensaje indicando la imposibilidad de aplicar el teorema en la función signo.

Al optar por la segunda opción que es la función valor absoluto (Fig.6), se desplaza el deslizador d, y cuando el mismo marca 0, aparece el mensaje que dicha función no es derivable en ese punto, ya que es un punto anguloso, por lo tanto no cumple con una de las hipótesis estipuladas para la aplicación del teorema.

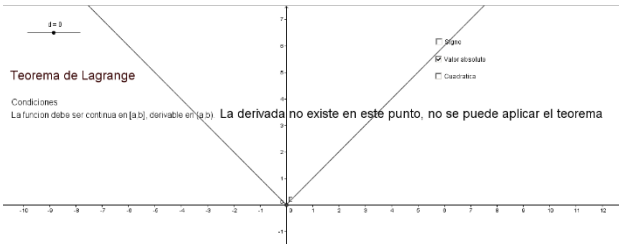


Fig. 6. Mensaje indicando la imposibilidad de aplicar el teorema en la función valor absoluto.

Finalmente si el alumno opta por la tercer casilla, la función cuadrática (Fig.7), vemos allí, que se disponen de tres deslizadores, con los deslizadores denominados "a" y "b" se regulan las coordenadas de los puntos A y B que se encuentran sobre la curva y por medio de los cuales se traza la recta secante a la curva dada, mientras que con el deslizador denominado "c" sirve para ubicar el punto de tangencia a la curva dada y por el cuál trazo la recta tangente a la misma.

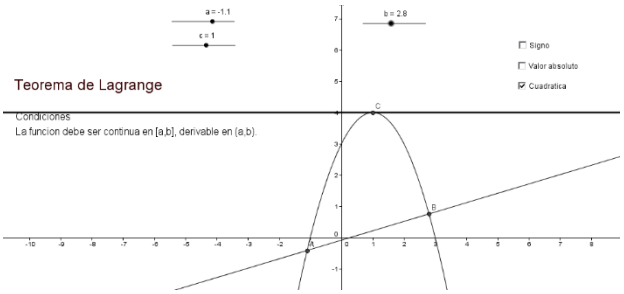


Fig. 7. Función cuadrática, cumple las condiciones para aplicar el teorema de Lagrange.

El alumno debe observar que la función cuadrática verifica las hipótesis del teorema, por lo tanto debe comprobar la tesis, para tal fin puede desplazar los deslizadores a y b, de tal forma que se aprecia la recta secante a la curva, luego desliza c hasta que observa que la recta tangente cambia de color (Fig. 8), señalando que en ese punto las dos rectas son paralelas, es decir encontró el valor "c" que verifica el teorema. Al desplazar los deslizadores "a" y "b", puede encontrar otras rectas secantes y para cada una de ellas el valor "c" correspondiente, pero siempre verificando las condiciones del teorema.

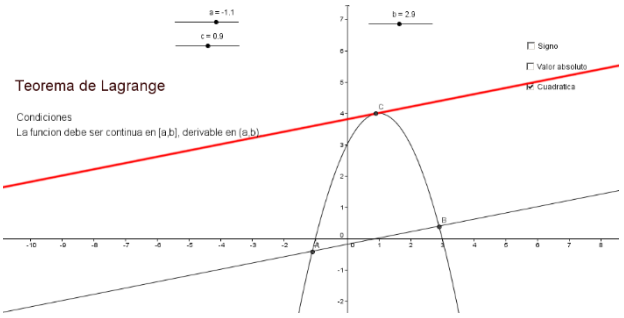


Fig. 8. Punto de verificación del teorema de Lagrange.

De esta manera el alumno comprueba que dado el intervalo (a,b) , siempre que la función verifique las condiciones del teorema encontrara diferentes valores de "c" que hacen cumplir el mismo, es decir, encuentra puntos en donde las dos rectas son paralelas, esta aplicación permite al alumno pasar de lo estático, que es lo que se vio en el pizarrón, a lo dinámico, que es la aplicación de la tecnología a la nueva enseñanza.

Para este caso se empleo el siguiente problema de verificación:

Compruebe que se cumple el Teorema del Valor Medio, y encuentre el valor de c que lo verifica para la función f, definida por

$$f(x) = -(x-1)^2 + 4 \text{ en el intervalo } [-1, 2]$$

La función, al ser un polinomio, es continua y derivable en todo su dominio, por lo tanto en el intervalo dado, conserva estas propiedades.
El alumno puede ubicarse con ayuda del deslizador "a", en el punto (– 1 , 0) y con el deslizador "b" , en (2 , 3). Entonces al mover el deslizador "c", en el intervalo dado, vera que hay un valor, c = ½ , en donde la recta tangente se torna roja, indicándole que en ese punto las dos rectas son paralelas, la secante con la tangente, hallando c que era lo que se busca y efectivamente c pertenece al intervalo [– 1, 2].

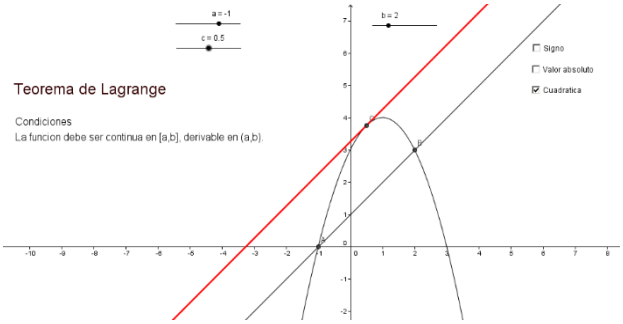


Fig. 9. Resolución grafica del problema planteado

2.2 Teorema de Rolle:
El teorema establece que: Si f es continua en [a, b] y derivable sobre (a , b) , y f(a) = f(b) , entonces existe un número c en (a, b) tal que f '(c) = 0.

En este caso las hipótesis a considerar son las siguientes:

I) La función debe ser continua en el intervalo [a, b]
II) La función debe ser derivable en el intervalo (a,b)
III) f (a) = f (b)

Lo que debemos comprobar es que debe existir un numero c, comprendido en el intervalo (a,b) en donde se verifique;

$$f'(c) = 0 \qquad (6)$$

es decir, que la pendiente de la recta tangente sea nula, equivale a decir que la recta tangente sea horizontal.

Para el caso de la función signo, es similar a lo visto anteriormente para el teorema de Lagrange, avisando al alumno la imposibilidad de aplicar el teorema porque no cumple con la condición de continuidad, tal como se vio en la Fig. 5.

Para el segundo caso, valor absoluto, sucede algo similar , pero en este caso con el deslizador i se puede ver que los valores de la función son los mismos, cumpliéndose la tercera condición, pero no se cumple la de derivabilidad, siendo señalada al desplazar el deslizador d, como en el caso del teorema de Lagrange (Fig.10).

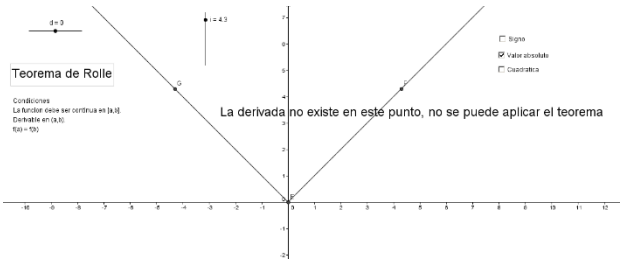


Fig.10. Función valor absoluto no verifica las condiciones del teorema de Rolle.

Para el tercer caso (Fig.11) tenemos los dos deslizadores "a" y "b" que toman el mismo color para advertir que se está cumpliendo la tercera condición, tras lo cual se puede desplazar el deslizador "c" hasta obtener el valor pedido.

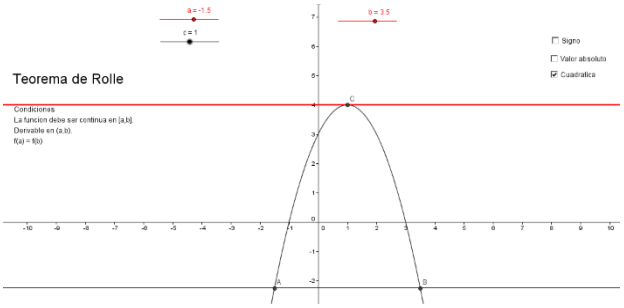


Fig. 11. Punto en donde se verifica el Teorema de Rolle

Evaluación

Se hizo una evaluación puntual con respecto a la aplicación utilizada para presentar en forma visual el teorema de Lagrange. La misma consistió en la resolución de un problema de aplicación de este teorema, el cual fue presentado a dos grupos de alumnos, estos grupos son resultado de un muestreo al azar de cada comisión.
El Grupo 1 está compuesto por 30 alumnos, que concurren a la comisión en la cual se utilizó la aplicación del GeoGebra para explicar el Teorema, la presentación del problema se hizo en forma dinámica utilizando la misma; el Grupo 2 formado por 20 alumnos, que asisten a la

comisión en la cual se explicó el Teorema utilizando simplemente el pizarrón, en este grupo el problema fue presentado en la forma tradicional es decir una hoja impresa con el enunciado del mismo y el gráfico que describía la situación planteada.
El problema formulado para ambos grupos fue el siguiente:
Un automóvil pasa por el control de una autopista, en la cual la velocidad máxima es de 50 km /h , a las 12 del mediodía, le entregan un comprobante de la hora por la cual él paso por ese punto, se encamina a un segundo control ubicado a 125 km , por el mismo pasa justo 2h 30 minutos (2,5 hs) después del primer punto , le cobran el correspondiente peaje, pero también le cobran una multa por haber excedido la velocidad máxima permitida, como sabían que violó la velocidad máxima?

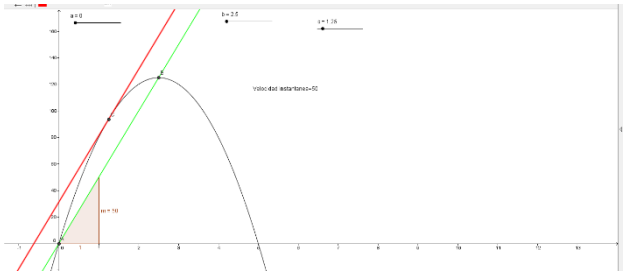


Fig. 12. Descripción de la situación planteada en el problema

Responda las siguientes preguntas:

a) De acuerdo al gráfico, que representa m = 50?
b) Cuál es su velocidad final?
c) En que instante se verifica el Teorema de Lagrange?
d) Antes de que instante el conductor del vehículo sobrepasa la máxima velocidad permitida?
e) Luego de ese instante que ocurre?

Los resultados de esta evaluación se presentan en la tabla 1 que corresponde al Grupo 1 y en la tabla 2 que corresponde al Grupo 2.

Tabla 1.Grupo 1 (30 alumnos)

	Correcto	Incorrecto
A	66%	33%
B	86%	14%
C	60%	40%
D	53%	47%
E	56%	44%

Tabla 2.Grupo 2 (20 alumnos)

	Correcto	Incorrecto
A	40%	60%
B	65%	35%
C	75%	25%
D	35%	65%
E	30%	70%

Comparando ambas tablas se observa un porcentaje mayor de respuestas correctas en casi la mayoría de los ítems en el Grupo 1, lo que nos lleva a una conclusión preliminar favorable con respecto a la inclusión de la visualización dinámica para la comprensión de los teoremas del análisis matemático.
Esta evaluación puntual realizada forma parte de una evaluación general, de la cual tendremos resultados una vez finalizado el presente periodo lectivo.

Conclusión

En la plataforma Moodle la interactividad con los alumnos se incrementó, dado que aumentaron las consultas y la participación de los mismos en el aula virtual de la cátedra.
La participación de los alumnos en las clases teóricas desarrolladas con la implementación de applets de GeoGebra es mayor en relación a las clases tradicionales. Se pudo apreciar que el alumno aprende más rápido y con mejores fundamentos, ya que se le facilito la representación gráfica de los conceptos y procedimientos matemáticos.
En este ambiente virtual, por medio de los applets, se proporcionó al alumno herramientas que le permiten conjeturar hipótesis, comprobar propiedades y analizar situaciones que en otro ámbito se tornarían más difíciles de visualizar, logrando desarrollar esta capacidad y que esta tienda a mejorar.
Este trabajo proporcionó mecanismos para salir del tipo de aprendizaje memorístico y repetitivo. El GeoGebra facilita la relación entre las Matemáticas y las TIC con la finalidad de propiciar un cambio en la metodología de trabajo con los estudiantes y por ende, en los procesos de enseñanza y aprendizaje.
Esta experiencia nos sirvió para experimentar con applets sencillos que ayudan mucho al alumno y que se puede seguir con los temas que más dificultad presentan en las distintas materias referidas con las matemáticas.
A través de los applets elaborados con GeoGebra se pone de manifiesto su potencial para abordar las dimensiones semántica y sintáctica de manera conjunta en los conceptos matemáticos, propiciando en los estudiantes la inquietud por explorar e indagar sobre aspectos teóricos de las matemáticas evitando así dedicar más tiempo al desarrollo de los mismos en las clases teóricas y facilita así el desarrollo de todo el programa de la materia.

Referencias

[1] Canos, L. y Ramón, F. "*Una experiencia sobre la interacción entre la formación universitaria y el uso de nuevas tecnologías"*. Jornadas de la Red Estatal de Docencia Universitaria (REDU), Valencia. (2007).

[2] Vaillant y Marcelo, C. *Las tareas del formador*. Ed. Aljibe. Málaga. (2001).

[3] García-Valcarcel, Muñoz-Repiso A.. *Competencias en Tecnologías de Información y Comunicación (TIC) de los estudiantes universitarios*. Revista Latina de comunicación social ISSN -e 1138-5820 N° 66 (2011).

[4] Gonzalez de la Higuera Aparicio J.J. *Aprender a aprender: Una perspectiva curricular para el asesoramiento e intervención educativa*. http://www.scridb.com/doc/9051313/aprender-a-aprender (Consultado 24/03/14). (2008).

[5] Cabero Almenara J.. *La formación en la sociedad del conocimiento. En Las TICs en los contextos de formación universitaria.* Indivisa Boletín de Estudios e Investigación. Monografía X. Madrid. (2008).

[6] Fernandez Muñiz B., Suarez Álvarez L., Alvarez Arregui E.. *El camino hacia el Espacio Europeo de Educación Superior: deficiencias metodológicas y propuestas de mejora desde la perspectiva del alumno*. ICE. Universidad de Oviedo Aula Abierta, 88, pp: 85-106. (2006).

[7] Cabero Almenara J.. *Las necesidades de las TIC en el ámbito educativo oportunidad, riesgo y necesidades*. Revista electrónica Tecnología y Comunicación Educativas, nº45,

http://investigacion.ilce.edu.mx/stx.asp?id=2281 (Consultada 28/09/13). (2007).

[8]Riveros, Víctor . *Implicaciones de la Tecnología Informatizada en el proceso de enseñanza y aprendizaje de la matemática*. Tesis Doctoral, Facultad de Humanidades y Educación, LUZ, Maracaibo, Venezuela. (2004).

[9] Departamento de Didáctica de la Matemática Facultad de Ciencias de la Educación Universidad de Granada. *Didáctica de las matemáticas para maestros.* 18071 Granada Gami, S. L. Fotocopias Avda. de la Constitución, 24. Granada. Distribución en Internet:

http://www.ugr.es/local/jgodino/edumat-maestros/

[10] Ausubel, D.P., Novak j. y Hanesian, H.. *Psicología educativa: un punto de vista cognoscitivo*. 2da. edición. Trillas. México. (1983).

[11] Barell, J.. *El aprendizaje basado en problemas. Un enfoque investigativo*. Buenos Aires, Manantial. (1999).

[12] Branda, L.A.. *El Aprendizaje Basado en Problemas: Una herramienta para toda la vida*. Madrid, Escuela Universitaria Enfermería Comunidad de Madrid. (2004).

[13] Castro, E. & Castro, E. *Representaciones y modelización*. En Rico, L.Coord. La Educación matemática en la enseñanza secundaria 12: 95-13. Barcelona. Esp.: ICE/HORSORI. (1997).

[14] Zimmerman,W.,y Cunningham,S.(Ed.) *Visualizatión in teaching and learning Mathematics*. Mathematics Association of America. (1991).

[15] Presmeg, N.C. *On visualization and generalization in mathematics.Psychology of mathematics education*. Proceedings of the Twenty first annual meeting 1: 151-154. (1999).

[16] Gavilán J. M. & Barroso, R. Manual del curso de Thales-Cica 2009, "*Geogebra: TICs e innovación en la enseñanza de las matemáticas*". (2009).

[17] Gavilán J. M. & Barroso, R. "*GeoGebra como instrumento de la práctica del profesor*". Comunicación presentada en las II Jornadas de Geogebra en Andalucía. Huelva, 1 al 3 de abril de 2011. Organizadas por el Instituto GeoGebra de Andalucía-SAEM Thales. (2011).

[18] Hollebrands, K. "*The Role of a Dynamic Software Program for Geometry in the Strategies High School Mathematics Students Employ*". Journal for Research in Mathematics Education, 38(2), pp. 164-192. (2007).

Datos de Contacto:

Lamas, Roberto Daniel, Facultad de Ingeniería, Universidad Nacional de Jujuy, Código Postal: 4600

rdlamas@fi.unju.edu.ar

Análisis de los Estilos de Aprendizajes de alumnos de Ingeniería Informática a través del Cuestionario CHAEA

García, Adelina; Lamas, Roberto Daniel
Universidad Nacional de Jujuy, Facultad de Ingeniería

Abstract

Este trabajo presenta la experiencia desarrollada para indagar sobre los estilos de aprendizaje de los alumnos de segundo año de Matemática Discreta de las carreras de Ingeniería Informática y Licenciatura en Sistemas Un alumno con dificultades de aprendizaje puede ser recuperable, pero es de conocimiento general que la dificultad de aprendizaje provoca el fracaso universitario y el abandono de los estudios Para implementar metodologías de enseñanza que ayuden a la motivación y retención de los alumnos es necesario tener un conocimiento completo de las maneras en que los mismos desarrollan su aprendizaje. Los estilos de aprendizaje dan cuenta de las diferencias actitudinales, motivacionales, cognitivas personales que se ponen en juego cuando se lleva a cabo el proceso de aprendizaje y se relaciona con los resultados académicos de los estudiantes, su desenvolvimiento en el aula y su disposición para el aprender. Se aplico el Cuestionario Honey-Alonso de Estilos de Aprendizaje CHAEA a una muestra de 82 alumnos, encontrándose un mayor nivel de preferencia por los estilos teórico y pragmático, una menor preferencia por los estilos reflexivo y activo. Los resultados obtenidos acerca del conocimiento de los estilos de aprendizaje de los alumnos nos llevan a realizar las modificaciones necesarias en los métodos de enseñanza presenciales y adecuar las actividades complementarias desarrolladas en la plataforma virtual para atender la diversidad de estilos de aprendizajes presentado.

Palabras Clave
Estilos de aprendizaje, TIC, rendimiento académico, CHAEA

Introducción

En las últimas décadas la sociedad está experimentando una serie de cambios e innovaciones radicales relacionadas con la comunicación y las tecnologías. Cambios que están configurando un nuevo modelo de sociedad, la sociedad de la información, o la sociedad red [1]. En el contexto educativo, la sucesión de cambios o impactos a los que hacemos referencia, lleva a hablar de una nueva revolución educacional, como dice Brunner[2] la cuarta revolución, que conllevará cambios tanto en la organización del conocimiento, en las prácticas y forma de organización social y en los procesos cognitivos del ser humano; como lo hicieron las "tres revoluciones anteriores": la aparición del lenguaje oral, la aparición y difusión de la escritura y la aparición de la imprenta.

Las TIC en el contexto de la educación superior, exigen un modelo educativo caracterizado entre otros rasgos, por el incremento de la capacidad de decisión del alumno sobre su proceso de aprendizaje, así como por una mayor capacidad y libertad para seleccionar y organizar su propio proceso formativo[3].

No olvidemos que la información no es conocimiento, es la materia prima necesaria pero no suficiente y será el proceso formativo el que la transforma en conocimiento. Habrá que arbitrar los medios pedagógicos para pasar de la sociedad de la información a la sociedad del conocimiento [4]. Pasar de cualquier expresión verbal, numérica, gráfica o de otro tipo que posee un significado dentro de un contexto y cuyo objetivo es interiorizar esa información nueva, a nuestras estructuras cognitivas, adquiriendo nuevos conocimientos [5].

Las investigaciones cognitivas han demostrado que las personas aprenden de diversas formas, lo que hace necesaria la reflexión sobre las variables que influyen en la capacidad de aprender [6].

Una de estas variables es el estilo de aprendizaje o tendencia a desarrollar preferencias globales en la elección de estrategias para aprender. Por tanto, podemos establecer diferencias individuales en el aprendizaje: las relacionadas con el potencial de aprendizaje de los individuos y que tienen que ver con el modo de aprender [7].

Estilos de aprendizaje

Las personas perciben y adquieren conocimiento, tienen ideas, piensan y actúan de manera distinta. El término estilos de aprendizaje se refiere a esas estrategias preferidas que son, de manera más específica, formas de recopilar, interpretar, organizar y pensar sobre la nueva información [8]. El aprendizaje es posible sobre todo cuando cada persona encuentra la forma o los procedimientos que más le acomodan y le facilitan el conocimiento, por tales circunstancias se sugiere que se adecue el proceso de enseñanza-aprendizaje en las distintas modalidades de formación a los tipos de inteligencias y estilos de aprendizaje [9].

El concepto mismo de Estilo de Aprendizaje no es común para todos los autores y es definido de forma muy variada en las distintas investigaciones. Desde nuestro punto de vista, una de las definiciones más claras y ajustadas es la que propone Keefe y que hacemos nuestra [10]:

"Los Estilos de Aprendizaje son los rasgos cognitivos, afectivos y fisiológicos, que sirven como indicadores relativamente estables, de cómo perciben los discentes, interaccionan y responden a sus ambientes de aprendizaje"

Existen numerosas teorías e instrumentos de medición respecto de los estilos de aprendizaje. Simplificando mucho y esquematizando la teoría sobre Estilos de Aprendizaje, podríamos distinguir dos tendencias. En primer lugar los autores que se centran de forma prioritaria en los aspectos cognitivos del individuo y en segundo lugar los autores que se centran en el proceso de aprendizaje. Los primeros se basan más en los aspectos psicológicos y los segundos se apoyan más en aspectos pedagógicos. Los primero prefieren hablar de estilos cognitivos, los segundos se refieren a estilos de aprendizaje [6].

Teoría de Aprendizaje por experiencias de David Kolb

Dentro de estas teorías nos enfocamos en la Teoría de Aprendizaje Experiencial, desarrollada por Kolb [11], reflexiona sobre como las distintas formas de aprender de cada sujeto responde a la herencia, las experiencias anteriores y las características y exigencias concretas del actual entorno de aprendizaje. Divide en dos las actividades de aprendizaje:

1. Percepción. Al encontrarnos ante nuevas situaciones, tomamos la información sensorial o emocional y adquirimos la experiencia.
2. Procesamiento. Modo como manejamos o transformamos esa nueva información, la propia experiencia y la hacemos parte de nosotros mismos, creando conexiones significativas.

Los modos de experienciar (percibir y procesar la experiencia) son:

- Modo concreto(EC)
- Modo reflexivo(OR)
- Modo abstracto(CA)
- Modo activo(EA)

El predominio de estas habilidades, son las que desarrollan los estilos de aprendizaje preferidos que tienen sus características.

- Estilo Divergente, enfatiza los modos EC y OR
- Estilo Asimilado o analítico, privilegia los modos CA y OR.
- Estilo Convergente, enfatiza los modos EC y EA.
- Estilo Acomodador, privilegia los modos EC y EA.

Kolb[12], elaboró un instrumento que ha tenido gran influencia en la investigación del proceso de aprendizaje:Inventario de estilos de aprendizaje (Kolb's Learning Style Inventory). Dónde define lo siguiente:

- Estilo activo: se distingue por la implicación en nuevas tareas, mente abierta a lo nuevo y el espíritu entusiasta para emprender nuevas actividades.
- Estilo reflexivo: se define por el análisis en profundidad de un problema antes de tomar una decisión, no se actúa en un primer momento sino que se analizan las posibilidades, se obtiene la mayor información posible, se observa a otros, se buscan distintos puntos de vista.
- Estilo teórico: se caracteriza por el afán de perfección, por el intento de comprender los hechos dentro de marcos globales, lógicos y coherentes. Se analizan los problemas en profundidad y se enmarcan en modelos complejos.
- Estilo pragmático: se basa en la rápida aplicación práctica de las teorías e ideas abstractas. Se lleva a la práctica la utilidad de los nuevos conocimientos.

En las personas están presentes en mayor o menor medida características de todos los estilos con predominio de alguno de ellos, este predominio significa que cada sujeto tiene una tendencia hacia una determinada forma de aprender y que mejorara su aprendizaje si la acción docente se ajusta a ese estilo personal.

Cuestionario de P. Honey y A. Mumford

Partiendo del planteamiento de David Kolb, Peter Honey y Alan Munford [13] desarrollaron un modelo, dónde señalan cuatro etapas de proceso:

1. Tener una Experiencia.
2. Repasar la Experiencia.
3. Sacar conclusiones de la Experiencia.
4. Planificar los pasos siguientes.

A su vez desarrollaron otro instrumento para conocer los estilos de aprendizaje, el LSQ (Learning Styles Questionaire), enfocado al mundo empresarial.

Las escalas del LSQ son unidimensionales y continuas y los sujetos deben señalar su acuerdo o desacuerdo con una serie de ítems, considerándose cuatro escalas que se corresponden con los cuatro estilos definidos por Kolb: activo, reflexivo, teórico y pragmático. Está diseñado para detectar tendencias generales del comportamiento personal e individual. [6].

Cuestionario de Honey y Alonso de Estilos de Aprendizaje (CHAEA)

Los planteamientos teóricos y prácticos de Honey y Mumford, son recogidas por Alonso en 1992 en España, quien, junto con Gallego, adaptaron el cuestionario LSQ de Estilos de Aprendizaje al ámbito académico iberoamericano y al idioma Español. Alonso adaptó el instrumento LSQ, de la lengua inglesa para la española y lo dirigió a los estudiantes de la universidad, añadiendo una serie de preguntas socio-académicas. De este modo nació el CHAEA, Cuestionario de Honey y Alonso de Estilos de Aprendizaje. Asimismo, realizó una investigación con el propósito de comprobar la validez del cuestionario misma que fue altamente comprobada. La fiabilidad y validez del CHAEA ha sido demostrada basándose en las pruebas estadísticas pertinentes, al analizar los estilos de aprendizaje de una muestra de 1.371 alumnos de 25 Facultades de la Universidad Complutense y Politécnica de Madrid [7].

El conocimiento y uso que hasta hoy se tiene del cuestionario resulta bastante abundante y difundido. Cabe señalar que el cuestionario es accesible en línea.

La investigación, en la que se apoya este Cuestionario se inscribe dentro de los enfoques cognitivos de aprendizaje y acepta, una división cuatripartita del aprendizaje en consonancia con lo anunciado por Kolb, Juch, Honey y Mumford. Dividen el proceso de aprendizaje en cuatro etapas [14] [15]:

1. Vivir la experiencia: Estilo Activo
2. Reflexión: Estilo Reflexivo
3. Generalización, elaboración de hipótesis: Estilo Teórico.
4. Aplicación: Estilo Pragmático.

Los alumnos que obtengan un claro predominio de un estilo, manifestaran algunas de las características que presentamos para los distintos estilos [16].

1. Estilo Activo

- Se involucran totalmente y sin prejuicios en las experiencias nuevas
- Disfrutan el momento presente y se dejan llevar por los acontecimientos
- Son entusiastas ante lo nuevo , actúan primero y piensan después en las consecuencias
- Llenan sus días de actividades y tan pronto disminuye el encanto de una de ellas se lanzan a la siguiente
- Les aburre ocuparse de planes a largo plazo y consolidar los proyectos, les gusta trabajar rodeados de gente, pero siendo el centro de las actividades.

La pregunta que quieren responder con el aprendizaje es ¿Cómo?

Los activos aprenden mejor:

- Cuando se lanzan a una actividad que les presente un desafío.
- Cuando realizan actividades cortas de resultado inmediato.
- Cuando hay emoción, drama y crisis

Les cuesta más trabajo aprender:

- Cuando tienen que adoptar un papel pasivo.
- Cuando tienen que asimilar, analizar e interpretar datos.
- Cuando tienen que trabajar solos

2. Estilo Reflexivo

- Los alumnos reflexivos tienden a adoptar la postura de un observador que analiza sus experiencias desde muchas perspectivas distintas
- Recogen datos y los analizan detalladamente antes de llegar a una conclusión. Para ellos lo más importante es la recolección de datos y su análisis concienzudo, así que procuran posponer las conclusiones todo lo que pueden
- Son precavidos y analizan todas las implicaciones de cualquier acción antes de ponerse en movimiento
- En las reuniones observan y escuchan antes de hablar, procurando pasar desapercibidos

La pregunta que quieren responder con el aprendizaje es ¿Por qué?.

Los alumnos reflexivos aprenden mejor:

- Cuando pueden adoptar la postura del observador.
- Cuando pueden ofrecer observaciones y analizar la situación.
- Cuando pueden pensar antes de actuar.

Les cuesta más aprender:

- Cuando se les fuerza a convertirse en el centro de la atención.
- Cuando se les apresura entre una actividad y otra.
- Cuando tienen que actuar sin poder planificar previamente

3. Estilo Teórico
- Los alumnos teóricos adaptan e integran las observaciones que realizan en teorías complejas y bien fundamentadas lógicamente
- Piensan de forma secuencial y paso a paso, integrando hechos dispares en teoría coherentes
- Les gusta analizar y sintetizar la información y su sistema de valores premia la lógica y la racionalidad
- Se sienten incómodos con los juicios subjetivos y las técnicas de pensamiento lateral y las actividades fuera de lógica clara.

La pregunta que quieren responder con el aprendizaje es ¿Qué?
Los alumnos teóricos aprenden mejor:
- A partir de modelos, teorías, sistemas con ideas y conceptos que presenten un desafío.
- Cuando tienen oportunidad de preguntar e indagar

Les cuesta más aprender:
- Con actividades que impliquen ambigüedad e incertidumbre.
- En situaciones que enfaticen las emociones y sentimientos.
- Cuando tienen que actuar sin un fundamento teórico.

4. Estilo Pragmático
- A los alumnos pragmáticos les gusta probar ideas, teorías y técnicas nuevas y comprobar si funcionan en la práctica
- Les gusta buscar ideas y ponerlas en práctica
- Inmediatamente les aburren e impacientan las largas discusiones, sobre la misma idea de forma interminable
- Son básicamente gente práctica, apegada a la realidad, a la que le gusta tomar decisiones y resolver problemas
- Los problemas son un desafío y siempre están buscando una manera mejor de hacer las cosas.

La pregunta que quieren responder es ¿Qué pasaría si…?
Los alumnos pragmáticos aprenden mejor:
- Con actividades que relacionen la teoría y la práctica.
- Cuando ven a los demás hacer algo.
- Cuando tienen la posibilidad de poner en práctica inmediatamente lo que han aprendido

Les cuesta más aprender:
- Cuando lo que aprenden no se relaciona en nada con sus necesidades inmediatas
- Con aquellas actividades que no tienen una finalidad aparente.
- Cuando lo que hacen no está relacionado con la "realidad"

Existe una brecha entre la mala preparación del estudiante al ingresar en la universidad y las exigencias que la institución demanda a sus alumnos. El docente, en su gran mayoría, prepara su programa de aprendizaje sin preocuparse por el tipo de alumno que asistirá a sus clases, o sea, cuáles serán los Estilos de Aprendizaje de sus alumnos y, consecuentemente, cuáles serán los estilos de enseñanza que deberá desarrollar para el grupo, y mucho menos cuales serán los procesos mentales envueltos en una determinada actividad de aprendizaje.
Esta situación, acaba por no favorecer un clima satisfactorio en la clase, dificultando así, la calidad del proceso de aprendizaje-enseñanza. Los profesores, al tener la responsabilidad de enseñar a los estudiantes de la forma en que sea más fácil y accesible para ellos, teniendo en cuenta que cada profesional tiene un modo particular de enseñar, diagnosticar los estilos de aprendizaje es relevante. Diversos estudios parecen avalar que los alumnos obtienen mejores rendimientos cuando se les enseña desde su propio estilo de aprendizaje [17], [18]. El estilo de enseñar es "el modo habitual de acercarse a los discentes con varios métodos de enseñanza"[19].
El proceso se facilita si, cuando se diseña todo el proceso didáctico, se tiene en cuenta el estilo de aprendizaje del alumno, es decir, se particulariza su individualidad.

Experiencia

En base a lo presentado anteriormente, se cree necesario indagar sobre los estilos de aprendizaje de los alumnos de ingeniería, ya que se observa en las materias de primer año mucho abandono y reprobación de las mismas, a pesar de que se tiene implementado un curso de ingreso en el cuál se intenta nivelar los conocimientos antes de su ingreso en la universidad. Se lleva a cabo la experiencia dentro del Proyecto de Investigación "Análisis y mejoramiento del entorno virtual de enseñanza-aprendizaje como apoyo de las clases presenciales en las cátedras de primer año ". En este proyecto están integrados docentes de Análisis Matemático, Algebra y Geometría Analítica y Matemática Discreta de la Facultad de Ingeniería, Universidad Nacional de Jujuy.

El objetivo principal es indagar sobre los estilos de aprendizaje de los alumnos de las carreras de informática.
En base a los resultados se readecuarán las estrategias de enseñanza presencial apoyados en las TIC, como así también se reforzaran las actividades adicionales implementadas en la plataforma Moodle, para un mayor aprovechamiento de las capacidades de los estudiantes y el desarrollo de aquellos estilos que menos emplea, para lograr mejorar la retención de los alumnos y poder ayudarlos a aprender a aprender que es la herramienta que les facilitara su subsistencia social y académica.

Desarrollo

Esta investigación es de tipo exploratoria y descriptiva ya que pretende dar información acerca de los estilos de aprendizaje de los estudiantes que cursan las carreras de informática por medio del Cuestionario de Estilos de Aprendizaje de Honey-Alonso, para lo cual se utilizo como población de estudio los alumnos que cursan Matemática Discreta, materia de segundo año, primer cuatrimestre de las carreras de informática, se realizo esta elección debido a la disponibilidad y accesibilidad de los profesores de esta cátedra, ya que los profesores involucrados en este equipo de trabajo encuentran poca colaboración por parte de sus pares y utilizamos las herramientas que se encuentran disponibles para nuestras investigaciones. Y esta población de alumnos es representativa de los estudiantes de ingeniería informática ya que se encuentran en segundo año de la carrera.

Participantes

La población estuvo constituida por los alumnos de segundo año de la materia Matemática Discreta del ciclo lectivo 2013 y 2014, primer cuatrimestre, la misma fue de 82 alumnos.

Instrumentos

El Cuestionario Honey-Alonso de Estilos de Aprendizaje (CHAEA) es un cuestionario de respuesta dicotómica, de aplicación individual, que consta de 80 ítems, 20 para cada estilo de aprendizaje (activo, reflexivo, pragmático y teórico) distribuidos aleatoriamente. La puntuación absoluta que cada sujeto obtiene en cada grupo de 20 ítems indica el nivel que alcanza en cada uno de los cuatro estilos, permitiendo evaluar las preferencias de los estudiantes en cada uno de los estilos y obtener, su perfil de aprendizaje.

Procedimiento

Se procedió a la aplicación del cuestionario, previamente de informarles a los alumnos para que se realizaba el mismo y pidiéndole su participación voluntaria y dándoles las instrucciones para el llenado del mismo. Los datos fueron procesados con el software SPSS versión 14.0

Resultados

Los datos socio demográficos permiten señalar que la edad promedio de la muestra fue de 21 años. La población estuvo conformada por 13 mujeres que constituyeron el 15.85% y 69 hombres que constituyeron el 84,15%, para un total de 82 estudiantes participantes.
Los resultados presentados en la Tabla 1, son los valores promedios y la desviación estándar de cada estilo de aprendizaje obtenidos a partir del procesamiento de los datos del Cuestionario de Honey-Alonso para una población de 82 estudiantes de Matemática Discreta.

Tabla 1.Valores promedios de los Estilos de Aprendizaje.

Estilo	Media	Desv.Std.
Activo	11,06	2,941
Reflexivo	14,44	2,758
Teórico	12,74	3,388
Pragmático	12,43	2,798

Según las puntuaciones establecidas en el Baremo General de Preferencias en Estilos de Aprendizaje propuesto por Alonso y cols. [7] que se presenta en la Tabla 2, la población de estudiantes analizada se sitúa en el rango moderado en los cuatro estilos, con tendencia de los estilos teórico y pragmático por sobre el reflexivo y activo.

Tabla 2.Baremo General de Preferencias en Estilos de Aprendizaje.

Preferencia	Activo	Reflexivo	Teórico	Pragmático
Muy Baja	0-6	0-10	0-6	0-8
Baja	7-8	11-13	7-9	9-10
Moderada	9-12	14-17	10-13	11-13
Alta	13-14	18-19	14-15	14-15
Muy Alta	15-20	20	16-20	16-20

Los valores promedio de los estilos activo, teórico se ubican por arriba de los valores obtenidos por Alonso para Facultades Técnicas [7], el estilo pragmático coincide aproximadamente y el estilo reflexivo se encuentra por debajo, teniendo en cuenta las desviaciones estándar podemos decir que se encuentran dentro de los valores obtenidos por Alonso.

Tabla 3.Baremo General de Preferencias en Estilos de Aprendizaje de Facultades.

Facultad	Activo	Reflexivo	Teórico	Pragmático
Humanidades	10,94	15,15	10,80	11,99
Técnicas	10,74	15,32	11,55	12,29
Experimentales	10,30	15,75	11,98	12,23

También se analizó las preferencias individuales de los alumnos dentro de cada estilo de aprendizaje, los resultados obtenidos se presentan en la Tabla 4 para el Estilo Activo

Tabla 4.Preferencias en el Estilo Activo.

Preferencia	Cantidad	Porcentaje
Alta	20	24,4%
Baja	7	8,5%
Moderada	39	47,6%
Muy Alta	10	12,2%
Muy Baja	6	7,3%
Total	82	100%

Dentro del estilo Activo la preferencia de los alumnos es moderada con un 47,6%. Luego están las preferencias altas o muy altas que en conjunto son de un 36,6% y un 15,8% tiene una preferencia baja o muy baja.
Para el estilo Reflexivo se presenta la Tabla 5.

Tabla 5.Preferencias en el Estilo Reflexivo.

Preferencia	Cantidad	Porcentaje
Alta	14	17,0%
Baja	25	30,5%
Moderada	35	42,7%
Muy Alta	0	0,0%
Muy Baja	8	9,8%
Total	82	100%

En el estilo reflexivo la preferencia es moderada con 42,7% luego sigue la tendencia baja con 40,2% y solo un 17% con tendencia alta en este estilo de aprendizaje, es decir los alumnos no son reflexivos en su estilo de aprendizaje.
Para el estilo Teórico se presenta la Tabla 6.

Tabla 6.Preferencias en el Estilo Teórico.

Preferencia	Cantidad	Porcentaje
Alta	15	18,3%
Baja	15	18,3%
Moderada	34	41,5%
Muy Alta	17	20,7%
Muy Baja	1	1,2%
Total	82	100%

En el estilo teórico la tendencia es moderada con un 41,5%, solo el 19,5% de los alumnos presentan tendencia baja en el estilo teórico.
Para el estilo Pragmático se presenta la Tabla 7.

Tabla 7.Preferencias en el Estilo Pragmático.

Preferencia	Cantidad	Porcentaje
Alta	22	26,8%
Baja	15	18,3%
Moderada	29	35,4%
Muy Alta	10	12,2%
Muy Baja	6	7,3%
Total	82	100%

En el estilo pragmático la preferencia es moderada con un 35,4%, y con tendencia baja o muy baja solo el 25,6%, alta o muy alta es 39%.
Para obtener los perfiles de estilos de aprendizaje de los alumnos que se muestra en la Tabla 8 se siguieron los parámetros de evaluación de CHAEA. Que dice que los dos puntajes brutos más altos conforman el perfil de estilo de aprendizaje. Se identificaron dos perfiles de mayor frecuencia, el perfil Reflexivo-Teórico y el perfil Teórico-Pragmático, además se identificaron otros tres perfiles con frecuencias bajas.
El perfil Teórico-Pragmático con 29,3% es el que prevalece entre los alumnos, seguido por el perfil Reflexivo-Teórico con 25,6% y el perfil Activo-Pragmático con 17,1%. Estos resultados concuerdan con los valores promedios obtenidos de los estilos de aprendizaje de toda la población de alumnos.

Tabla 8.Perfiles de Estilos de aprendizaje de los alumnos.

Perfil de Estilos de Aprendizaje	Total	
	Cantidad	Porcentaje
Activo-pragmático	14	17,1%
Activo-Reflexivo	8	9,8%
Activo-Teórico	9	11,0%
Reflexivo-Pragmático	6	7,3%
Teórico-Pragmático	24	29,3%
Reflexivo-Teórico	21	25,6%
Total de alumnos	82	100%

Análisis de Resultados

La población de estudiantes analizada se sitúa en el rango moderado en los cuatro estilos, con tendencia de los estilos teórico y pragmático por sobre el reflexivo y activo.

Al hacer los análisis de las preferencias de los alumnos en cada estilo, observamos que los alumnos prefieren el estilo activo con una tendencia alta y muy alta en un 36%, con respecto al estilo reflexivo la preferencia alta es solo del 17%, en cuanto al estilo teórico la preferencia alta y muy alta es del 39% y en el estilo pragmático la preferencia alta y muy alta es del 39%.

Los alumnos con un estilo de aprendizaje pragmático buscan la aplicación práctica del conocimiento, por lo que se debe aprovechar cualquier oportunidad para que ellos experimenten, les gusta desarrollar proyectos prácticos.

Estos alumnos, detonan su aprendizaje con la pregunta ¿qué pasaría si? Las actividades de enseñanza que se sugieren para este tipo de alumnos son: enseñar técnicas para hacer las cosas con ventajas prácticas que permitan la comprobación inmediata del aprendizaje; exponer a los alumnos a modelos y enseñar por proyectos o experimentos; dar indicaciones y sugerir atajos; y presentar gran cantidad de ejemplos.

Los alumnos con estilo de aprendizaje teórico buscan el enfoque lógico de los problemas, necesitan integrar la experiencia en un marco teórico de referencia, abordan los problemas de forma vertical escalonada, por etapas lógicas. Muestran fortaleza para el análisis, la síntesis y el establecimiento de principios, teorías y modelos. Usualmente buscan la racionalidad y la objetividad, descartando lo subjetivo y ambiguo. La pregunta que quieren responder con el aprendizaje es ¿Qué?

Las actividades de enseñanza que se sugieren son: a partir de modelos, teorías, sistemas con ideas y conceptos que presenten un desafío.

Para los alumnos con estilo de aprendizaje reflexivo tienden a adoptar la postura de un observador que analiza sus experiencias desde muchas perspectivas y procuran pasar desapercibidos. Detonan su aprendizaje con la pregunta ¿por qué? Las actividades de enseñanza que se sugieren son: permitir que los alumnos puedan sólo observar y pedir reflexiones sobre las actividades; intercambiar opiniones con otros con previo acuerdo; para ellos se recomiendo no establecer plazos y en la medida de lo posible permitir que trabajen sin presiones; y en la medida en que se pueda utilizar videos .

Los alumnos con estilo de aprendizaje activo aprenden mejor: cuando se lanzan a una actividad que les presente un desafío, realizan actividades cortas de resultado inmediato. Detonan su aprendizaje con la pregunta ¿Cómo?

Conclusión

Para poder proporcionar una educación integral e individualizada acorde con las nuevas teorías del aprendizaje, los docentes debemos aprender a realizar diagnósticos que incluyan los estilos de aprendizaje de cada uno de nuestros alumnos para adaptar las estrategias enseñanza aprendizaje a los diversos estilos de aprendizaje.

Con los resultados obtenidos podemos analizar nuestras prácticas docentes y hacer una readecuación en cuánto al material proporcionado en las clases prácticas, tendiendo a desarrollar el trabajo colaborativo entre ellos por medio de la resolución de problemas.

En cuanto a las clases teóricas presenciales se introducirá paulatinamente la visualización dinámica por medio de software libre que se adapte a los distintos temas de las materias, este grupo de trabajo se encuentra en estudio y producción de applets basados en Geogebra, ya que la mayoría de los alumnos tienen la computadora proporcionada por el Plan Conectar Igualdad.

En la plataforma educativa Moodle que se basa en el aprendizaje constructivista, este grupo de trabajo tiene a cargo las respectivas aulas virtuales de cada materia, con este resultado se readecuarán las actividades tendientes a que el alumno pueda construir su aprendizaje por sí mismo.

Como resultado de nuestra experiencia observamos que los alumnos deben desarrollar el estilo de aprendizaje reflexivo y activo. Tener desarrollado los cuatro estilos de aprendizaje les agregaría una competencia más a los alumnos del siglo XXI. Debido a que en esta sociedad del conocimiento se debe estar preparado para aprender a aprender a lo largo de nuestras vidas, por la vorágine de los cambios tecnológicos, y que hace que todo sea obsoleto en muy poco tiempo.

Podemos concluir que es importante conocer los estilos de aprendizaje de nuestros alumnos ya que esto nos provee de herramientas para ayudarlos a insertarse en la vida universitaria y disminuir la deserción y el recursado de materias.

La información obtenida se utilizará en la planificación y desarrollo del dictado de un curso e-learning de matemática discreta para alumnos recursantes en el segundo cuatrimestre, utilizando actividades desarrolladas en software libre teniendo en cuenta los estilos de aprendizaje detectados.

Referencias

[1]] Castells, M. "La era de la información". *La Sociedad red. .Vol 1.* Alianza Editorial. Madrid. (1996).

[2] Brunner J. *Nuevas Tecnologías y Sociedad de la información*. Preal. Chile. (2000).

[3] UNESCO. *Hacia la sociedad del conocimiento, capítulo 3º*. pp.61. UNESCO París. (2005).

[4] Castells, M. *La era de la información. Vol. 3 Fin de Milenio.* Alianza Editorial. Madrid. (1998).

[5] Castells, M. *La era de la información. Vol. 2 El poder de la identidad.* Alianza Editorial. Madrid. (1997).

[6] Alonso Garcia C.M., Gallego Gil D.J. *Los estilos de aprendizaje: una propuesta pedagógica*, I Congreso Internacional de Estilos de Aprendizaje, En línea: http://www.ciea.udec.cl/trabajos/Alonso-Gallego.pdf (Consultada 14/03/13) . (2004)

[7]Alonso, C.M. *Estilos de Aprendizaje: Análisis y Diagnóstico en Estudiantes Universitarios.* Madrid: Universidad Complutense. (1991)

[8] Alonso C.M. y Gallego, D.J. *Cómo diagnosticar y mejorar los estilos de aprendizaje.* Madrid: UNED, Formación Permanente. (2003).

[9] Revilla, D. *Estilos de aprendizaje. Temas de Educación.* Segundo Seminario Virtual del Departamento de Educación de la Pontificia Universidad Católica del Perú. Recuperado de: http://www.pucp.edu.pe/temas/estilos.html. (1998).

[10] Alonso García C.M., Gallego Gil, D. J. *Si yo enseño bien...¿por qué no aprenden los niños?*, II Congreso Internacional master de educación, Editorial Master libros "Educando en tiempos de cambio", En línea: http://www.congreso.gob.pe/historico/cip/eventos/congreso/IICongreso/Conferencias/conf_extranjeros/Conf_Ext_CatalinaAlonso.doc (Consultado 1/06/10). (2006).

[11] Kolb, D. *Experiential Learning: Experience as the source of Learning and Development*. Englewood Cliffs, New Jersey: Prentice-Hall. (1984).

[12] Kolb, D. *The Learning Style Inventory: Technical Manual.* Boston, Ma: Mc Ver(1976).

[13] Honey, P. y Mumford, A. *Using our Learning Styles.* Berkshire, U.K.: Peter Honey. (1986).

[14] Lago B., Colvin L., Cacheiro M., *Estilos de aprendizaje y actividades polifásicas.* Modelo EAAP, Revista Estilos de Aprendizaje 2(2):2-22.

http://www.uned.es/revistaestilosdeaprendizaje/numero_2/artigos/lsr_2_octubre_2008.pdf. (2008).

[15]Alonso Catalina M., Gallego Domingo J., Honey Peter. *Los estilos de aprendizaje. Procedimientos de diagnóstico y mejora.*. Bilbao. Ediciones Mensajero. (2007)

[16] Alonso C.M., Gallego, D.J. y Honey, P. *Estilos de aprendizaje.* Bilbao. Ediciones Mensajero. (1999).

[17] Alonso Catalina M., Gallego Domingo J., Honey Peter. *Los estilos de aprendizaje. Procedimientos de diagnóstico y mejora.* Ediciones Mensajero. Bilbao. (2007)

[18] Litwin, E. *Las configuraciones didácticas.* Paidos. Buenos Aires. (1997).

[19] Ramírez, B. M. *Estilos de aprendizaje y uso de tecnologías en el desarrollo de competencias profesiones del licenciado en Administración de empresas de la Universidad Autónoma de Baja California.* Universidad de Tijuana. Tésis de Doctorado. (2009).

Datos de Contacto:
García Adelina, Facultad de Ingeniería, Universidad Nacional de Jujuy. Calle Ítalo Palanca N° 10.San Salvador de Jujuy

agarcia@fi.unju.edu.ar

Programa de Tutorías Para Primer Año: Una experiencia con la intervención de monitores alumnos y las tecnologías como soporte comunicacional.

María Alejandra Odetti
Amalia Inés Haefeli
Alejandra María Jewsbury
Roberto Miguel Muñoz
Marcelo Martín Marciszack
Departamento de Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional- Facultad Regional Córdoba

Abstract
El Departamento de Ingeniería en Sistemas de Información, sostenido en experiencias anteriores de tutoría para sus estudiantes, implementa desde comienzos de año 2014 un programa destinado a la retención de sus estudiantes de primer año. Este artículo aborda la descripción de esta experiencia mostrando algunos resultados, limitaciones y propuestas de mejora.

Palabras Clave
Abandono, deserción, tutoría, retención, recuperación.

.

1. Introducción

La deserción en las universidades públicas Argentinas es un tema de agenda de política universitaria. Con números que indican que alrededor del 50% de los ingresantes desertan del sistema universitario, porcentaje que es mayor en carreras de ingenierías, la deserción se constituye en un fenómeno que intenta ser contenido desde las políticas de gestión universitaria[1]. Esto es evidenciado por el interés que el tema suscita en los procesos de evaluación y acreditación de carreras y por acciones de financiamiento tanto a las instituciones para mejorar la contención, como a los propios estudiantes mediante diferentes sistemas de becas universitarias[2].

Pero más allá del interés propio del sistema educativo superior en la retención de sus estudiantes, también es de destacar el papel que las universidades tienen en el desarrollo económico del país. Especialmente las carreras vinculadas a la informática y los sistemas de información tienen un destacado papel en el desarrollo tecnológico nacional y en la capacidad de creación de importante capital económico. Es por esta razón que los gobiernos nacional y provincial propician y fomentan el estudio de carreras informáticas, mediante diversas becas, tanto para alumnos que comienzan sus estudios, como líneas especiales para finalización de carrera.

Por otro lado, el interés de los jóvenes en incorporarse a carreras de ingeniería es bajo. A pesar de los estímulos importantes para la selección de carreras informáticas, las matrícula desciende, o en el mejor de los casos, permanece sin alteraciones entre un año y otro.

La baja matriculación en carreras relacionadas a la informática, sumada a la deserción de los estudiantes, genera egresos de profesionales del sistema universitario que son escasos en relación a las necesidades del sistema productivo.

La deserción constituye un fenómeno susceptible de ser objeto de miradas desde diferentes ángulos, y abordajes variados para su solución.

Actualmente las universidades realizan el análisis desde una mirada globalizadora y del conjunto de los estudiantes, identificando tendencias y generalidades que, en su interior, posee una variada gama de causas posibles, y por ende, variadas maneras de encarar su solución. Es por eso que se torna necesario el desarrollo de mecanismos que generen la identificación de quienes están en potencial riego de abandono, para, una vez detectados, ejercer sobre ellos una serie de acciones tendientes a su contención.

A su vez, es necesario también que estas señales de alerta sean visualizadas con la mayor anticipación posible, de modo de dar rápida respuesta y solución a situaciones problemáticas.

Las estadísticas de rendimiento y deserción realizadas finalizados los periodos permiten la toma de decisiones y realización de acciones de cobertura general: modificación de regímenes de correlatividades, ajustes en

sistemas y modalidades de evaluación, modificación de contenidos y/o modalidad de dictado de los mismos[3].

Dichas medidas encaminadas hacia la totalidad de la población estudiantil son favorables en la atención de necesidades del conjunto de estudiantes de una determinada carrera, pero dejan en el camino a quienes poseen una dificultad especial o problemática de abordaje específico.

Problemas académicos, actitudinales y motivacionales deben ser abordados de una manera específica.

Es aquí donde aparecen las tutorías como un mecanismo capaz de contener a los individuos con problemática específica.

Así mismo, las dificultades que los estudiantes presentan a lo largo de su carrera son diversas: no son las mismas para un ingresante que se enfrenta por primera vez a aulas numerosas, a compañeros desconocidos, una institución muchas veces inabordable, y a veces desarraigo del seno familiar; a las de un estudiante avanzado, afianzado en lo institucional, pero con problemáticas más cercanas a la vida laboral y por lo general con un proyecto familiar propio.

El Departamento de Ingeniería en Sistemas de Información, puso en marcha en el año 2014 un Programa de Tutorías especial para alumnos de primer año de la carrera de Ingeniería en Sistemas de Información.

2. El Programa de Tutorías de Primer Año. Motivaciones

La Carrera de Ingeniería en Sistemas de Información ha desarrollado desde el año 2010 un importante camino de mejoras, tendientes a su evaluación por parte de la CONEAU. Obtenida la acreditación y afianzada sólidamente en el contexto de las ofertas académicas de excelencia en el sistema universitario cordobés y nacional, se vienen implementado acciones en variadas direcciones tendientes a la concreción del Plan de Mejoras. Fortalecimiento de la investigación, implementación de carrera docente, mejoras de infraestructura y recursos en laboratorios y condiciones de estudio y de trabajo para alumnos y docentes, y fortalecimiento de planes de tutorías para estudiantes son ejemplos de ello.

En relación a las tutorías, se vienen realizado de manera sostenida experiencias para estudiantes avanzados de la carrera con dificultades para concretar su egreso, las que han resultado altamente satisfactorias, tanto para la institución como para los alumnos.

Pero el apoyo para los estudiantes en su tramo inicial es una deuda pendiente.

El mayor porcentaje de deserción se produce en estudiantes durante el tramo inicial de la carrera, particularmente durante el transcurso del primer año.[4]

De encuestas obtenidas en el marco del proyecto "Sistema de Monitoreo de la población estudiantil universitaria en riesgo de deserción con marco en el uso intensivo de tecnologías de la información" e investigaciones relacionadas[5], se identifican como las más importantes dificultades de los alumnos las siguientes

- Inseguridad en el manejo de competencias de estudio
  El egresado de nivel medio demuestra en general serias limitaciones en el manejo de herramientas y estrategias que le permitan abordar este nuevo desafío académico. Dificultades en el abordaje de competencias como: comprensión de textos, expresión oral, resolución de problemas, sumadas a escasos hábitos de estudio, dificultades en la organización del material, y planificación del tiempo de estudio son una muestra de esta problemática.

- Integración al contexto estudiantil e institucional
  El alumno que recién ingresa a la universidad se enfrenta con dificultades extras al desafío académico que implica una carrera universitaria: el ingreso a una institución numerosa, sin el acompañamiento/ contención de los pares que los acompañaron en todo el ciclo educativo anterior, ni de su familia. La Universidad es una institución con una lógica de funcionamiento muy diferente a la que el alumno ha tenido que enfrentar con anterioridad, sumado a que en numerosas ocasiones viven un desarraigo de su lugar de procedencia y alejados de su familia.

- Variables motivacionales: perfil de egreso.
  La imagen que el estudiante tiene del perfil del Ingeniero en Sistemas al momento de la elección de la carrera no es correcta, por lo general es ambigua y confusa, y en general está asociada a la de programador.

- Dificultades económicas
  Las dificultades económicas son también obstáculos con los que se enfrentan algunos estudiantes ingresantes. La existencia de programas de becas de variadas fuentes de financiamiento, que van desde becas de material bibliográfico otorgadas por la propia institución, así como becas provenientes del gobierno provincial o nacional ayudan a resolver al menos parcialmente estas situaciones, aunque

muchas veces la dificultad reside en el desconocimiento de estos sistemas de becas y de los mecanismos de tramitación de las mismas.

El Programa de Tutorías para Primer Año se crea con la intención de ofrecer un espacio, en el que el estudiante de primer año *voluntariamente puede participar* y recibir el acompañamiento de estudiantes avanzados quienes intentarán solucionar en parte las problemáticas anticipadas, entre otras problemáticas emergentes.[6]

3. Objetivos

El Programa se propone contribuir en la mejora del desempeño académico y a la disminución de la deserción de los estudiantes en el primer año de la carrera de Ingeniería en Sistemas de Información

Como objetivos específicos podemos mencionar:

- Identificar las causas que producen la deserción estudiantil en los primeros años de la carrera de Ingeniería en Sistemas de Información.
- Involucrar la participación de estudiantes avanzados de la carrera, como monitores, en el trayecto académico de los ingresantes.
- Atender las necesidades de los estudiantes a través de un plan de estrategias de mejora de rendimiento académico y disminución del abandono
- Promover la participación de estudiantes en actividades de socialización que fomenten la integración y faciliten la pertenencia institucional.

4. Descripción del Programa

El programa se concibe como una propuesta de variadas actividades de apoyo y acompañamiento al estudiante de primer año.

- De apoyo: se pretende brindar apoyo a los estudiantes de primer año en los siguientes aspectos.
    - Académico: de soporte a las materias de primer año pertenecientes a Sistemas de Información. Quedan excluidas de estas tutorías las materias pertenecientes a Ciencias Básicas.
    - Metodológico: a través del desarrollo de talleres y consultas relativas a métodos y técnicas de estudio.
    - Administrativo: orientando a los alumnos en consultas sobre trámites y procedimientos administrativos.
- De acompañamiento: mediante la participación de:
    - Tutores: Una Ing. en Sistemas de Información y una Lic. en Cs. de la Educación están a cargo de la coordinación y organización del programa.
    - Docentes: docentes voluntarios de las distintas asignaturas vinculadas al ofrecen horarios de consulta.
    - Monitores: lo constituyen estudiantes avanzados de la carrera que ofician de intermediarios en la comunicación entre los docentes y tutores.

5. Los Primeros Pasos.

Este programa fue inaugurado en abril del año 2014, y presentado a los profesores que integran las Cátedras que constituyen las asignaturas troncales de la Carrera (SOR, AED y MAD) y a todos los alumnos de primer año, utilizando diapositivas para explicar los objetivos y la lógica de funcionamiento, roles y soporte de interacción.

Las presentaciones fueron realizadas por separado: se convoca a los docentes de las cátedras AED, SOR y MAD, una catedra por vez, y se comunicó a los docentes las intenciones y objetivos del programa. A su vez, se los invitó a participar de manera activa en el programa, dado que ellos son los profesionales que pueden realizar una rápida detección de los estudiantes que estén manifestando algún tipo de dificultad: bajas notas en parciales, poca participación, inasistencias a clases, etc.

Posteriormente se realizó la presentación del Programa utilizando diapositivas para exponer los objetivos, roles y medios de interacción. Los alumnos asistieron en cada turno acompañados por los docentes.

Simultáneamente a la presentación del Programa a docentes y estudiantes, se realizó la convocatoria y selección de estudiantes avanzados que funcionarían a modo de monitores.

El proceso de selección de los monitores ha sido cuidadoso y atendiendo a los requerimientos que se habían definido para el rol: estudiantes de tercero a quinto año, que aún estén cursando materias (para garantizar su asistencia a la facultad), con un historial académico bueno, y de preferencia con algún tipo de referencia entre sus compañeros.

Una vez seleccionados los monitores, fueron asignados a los distintos cursos. Cada monitor a cargo de un curso se presentó ante el grupo, expresó sus horarios disponibles, su trayectoria en la universidad, su motivación en acompañarlos y entregó tarjetas con sus datos de contacto, poniéndose a disposición del grupo para atender cualquier problema o inquietud.

Consideramos de suma importancia el rol de los monitores en el programa. Su incorporación se fundamenta en la idea que éstos pueden facilitar la adaptación del ingresante al nuevo contexto educacional. Aspectos como la cercanía etaria de estos estudiantes con los ingresantes, el manejo de códigos, lenguaje e intereses comunes acrecienta la posibilidad de realizar consultas sobre variadas cuestiones y presentar sus problemas con mayor confianza. La imagen de un "hermano mayor" que acompañe el proceso de adaptación al nuevo sistema educativo, vivencias que se transmiten desde la propia experiencia es lo que ha motivado la incorporación de los monitores a este programa de tutorías.

6. Facebook: El uso de las Redes Sociales como espacio de comunicación

Además se creo un grupo moderado en Facebook: Tutorías de ISI Primer Año 2014. Este grupo es administrado por los responsables de las tutorías y por personal del Departamento de Sistemas habilitado a tal efecto.

La motivación para la creación de este grupo radica en la preferencia de los jóvenes por las redes sociales como mecanismo para comunicarse y compartir con otros. Facebook, Twiter, Google+, Instagram, Pinterest son los preferidos. La tecnología disponible en teléfonos móviles, tablets e intenet inalámbrica permiten mantener constante conexión a intenet y a las redes. Estas condiciones posibilitan que cualquier intervención en las redes en las que los alumnos se suscriben, llega instantáneamente.

Si bien la página de Autogestión de la FRC- UTN es el sitio donde los estudiantes obtienen toda la información que necesitan (horarios, inscripciones, notas, fechas de exámenes, materiales brindados por los profesores, etc.), es necesario el ingreso a la página para acceder a las novedades. Esta limitación hace que solo si el alumno se loguea en la página puede ver lo que se le publica, tal cual sucede con el email, mientras en las redes sociales la comunicación es instantánea y el estudiante lo reciba la instante en su teléfono móvil.

En el grupo se admiten docentes de las materias de primer año del Departamento de Sistemas, los monitores, y, por supuesto, los alumnos de primer año.

La elección de Facebook por sobre las otras redes, se debe a que no existe el límite de los 140 caracteres de Twitter y además la posibilidad de compartir y versionar archivos.

7. Actividades.

El equipo de tutorías se conforma, tal como se menciona con anterioridad, por los tutores, los monitores y los docentes voluntarios.

Entre los tutores y los monitores ser realzan reuniones quincenales de organización y de evaluación del avance de las actividades planificadas.

En dichas reuniones, los monitores realizan una pequeña evaluación de las actividades que han tenido con los alumnos: reuniones, clases de consulta, asesoramientos a alumnos, etc.

En cuanto a las actividades realzadas con los alumnos se pueden mencionar:

- Reuniones de monitores semanales con los estudiantes de los cursos que tienen asignados.
- Talleres de resoluciones de problemas de las materias de primer cuatrimestre.
- Talleres de estrategias de estudio de aprendizaje
- Clases de consultas con docentes voluntarios.
- Atención de situaciones particulares de alumnos.
- Facebook: respuestas a preguntas e inquietudes de los alumnos, publicación de material brindado por los docentes, ejercicios prácticos, etc.

Para mantener la historia de las actividades del programa, se realizan minutas de las reuniones con los monitores, y se registran en un formulario de Google las actividades con los alumnos, donde se mantiene información acerca de los responsables de la actividad, la cantidad de ingresantes intervinientes en la actividad, las inquietudes planteadas por los estudiantes, las observaciones realizadas por los monitores o tutor responsable, entre otros.

8. Resultados.

El tiempo transcurrido desde el comienzo de las tutorías es escaso como para evaluar la magnitud del impacto en la población estudiantil.

Como logros obtenidos podemos mencionar:

- Alta participación de estudiantes en las clases dictadas, especialmente de AED, como apoyo a los parciales de la materia.
- Participación de los estudiantes en Facebook, a través de preguntas respondidas por los propios estudiantes, los monitores y tutores.
- Talleres de estrategias y métodos de estudio con moderada participación.
- Participación activa de los docentes de las cátedras, informando situaciones de atención para el equipo de tutorías.
- Asesoramiento administrativo brindado a solicitudes de estudiantes de manera satisfactorias.

Si bien hasta el momento es dificultoso medir el impacto en términos de deserción y el rendimiento académico, sí es posible evidenciar el grado de participación del estudiantado en el programa, posibilitando algún nivel de contención en la resolución de problemáticas de diferente índole.

9. Conclusiones.

A pesar del corto recorrido de la experiencia, reconocemos algunas cuestiones de debemos fortalecer y otras que debemos modificar o más bien regular y sistematizar. La continuidad de las actividades, la información que se infiera de las encuestas, el incremento de docentes voluntarios y monitores, favorecerá el crecimiento y una mayor visibilidad del Programa, y en consecuencia la participación del ingresante.

Respecto a la incorporación monitores, podemos afirmar que la participación de alumnos avanzados con funciones de monitoreo del trayecto estudiantil fue una decisión acertada. La motivación manifestada por los "hermanos mayores", al ofrecer ayuda en la resolución de ejercicios, responder dudas teóricas en facebook, brindar material de estudio, convocar grupos de estudio los sábados, etc, dan cuenta de ello.

La vivencia de cada monitor es diferente y dada la realidad académica de cada estudiante, el tiempo de dedicación fue dispar. Este es un aspecto importante a atender, a la hora de sostener y complementar la presencia de algún monitor, para que el programa persista y sea consistente. Si bien el grupo quedó constituido a comienzos del ciclo académico, somos conscientes del dinamismo que requiere y la flexibilidad que debemos tener ante los cambios que se requieran en la marcha.

Las instancias de participación fueron, hasta el momento, suficientemente aprovechadas por muchos estudiantes para acomodar una rutina de estudios, aumentar la motivación y esclarecer el perfil profesional del Ingeniero en Sistemas de Información.

Sin embargo, no podemos dejar de mencionar la dificultad existente en vencer el temor, o reparo en acercarse a este nuevo espacio de contención. En una próxima etapa, las encuestas serán el instrumento adecuado para indagar a cerca de esta problemática y conocer el comportamiento de otras variables para evitar el bajo rendimiento académico y el abandono.

La contención de estudiantes en riesgo de deserción no es menor en términos de tiempo y recursos, sin embargo es el inicio de un proceso que quizás demande en un futuro decisiones que afecten aspectos administrativos y pedagógicos, y que involucren al profesorado en su conjunto en un plan estratégico mayor.

Sabemos que las tareas de tipo tutoriales, implican funciones de orden cognitivo, social y afectivo en tanto que son al mismo tiempo un lugar para el desarrollo de estrategias intelectuales, para la contención emocional y para la socialización en un espacio nuevo[6] y que dichas tareas deben desarrollarse a lo largo de toda la trayectoria del estudiante, especialmente en sus comienzos.

Estamos convencidos que la intervención en esta primera etapa del trayecto de formación del estudiante, permitirá adelantar decisiones de abandono al clarificar elecciones vocacionales en algunos casos y en el caso opuesto permitirá disminuir el fracaso en el aprendizaje, evitando la desilusión, la falta de motivación y previniendo el abandono. En ese sentido queda mucho por aprender y construir.

Referencias

[1] Resolución 786/09 Ministerio de Educación de la Nación Argentina. Resolución para la acreditación de las carreras de grado de ingeniería en sistemas de información y afines.

[2] Programa Nacional de Becas Bicentenario para carreras científicas y técnicas. Ministerio de educación de la Nación Argentina. www.becasbicentenario.gov.ar

[3] Programa de reinserción de estudiantes avanzados. Marciszack y Otros. Weef 2012. Foro Mundial de educación en Ingeniería. Buenos Aires. 2012.

[4] Aguilar Rivera, M. La transición a la vida universitaria. Éxito, Fracaso, Cambio y Abandono. Año 2007. Versión pdf en:
http://www.enduc.org.ar/enduc4/trabajos/t064-c36.pdf

[5] XI Congreso Nacional de Investigación Educativa / Sujetos de la Educación / Ponencia: Análisis estadístico de las causas de estrés en estudiantes universitarios.

Diagnóstico para acciones tutoriales Alicia López Betancourt, Irma Díaz Unzueta
[6] Mirian Capelari. Las configuraciones del rol del tutor en la universidad argentina: aportes para reflexionar acerca de los significados que se construyen sobre el fracaso educativo en la educación superior. Revista Iberoamericana de Educación. 2009.

Construcción de Mapas conceptuales como estrategia didáctica en la enseñanza de la asignatura Paradigmas de Programación.

Corso, Cynthia Lorena
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

Este trabajo está enmarcado en la asignatura "Paradigmas de Programación" perteneciente a la carrera de Ingeniería en Sistemas de Información, cuyos contenidos mínimos responden a la Ordenanza Nº 1150 que rige el Diseño Curricular de la carrera.
El Programa Analítico de la asignatura, en su Unidad Nº 4 "Paradigma Orientado a Objetos", comprende el estudio de los mecanismos fundamentales que dan origen a este paradigma, como así también la resolución de problemas en un lenguaje con orientación a objetos puro en su concepción y representación.
La metodología aplicada en el desarrollo de esta unidad consiste en exposiciones teóricas y guías de trabajos prácticos con ejercicios modelos.
El presente trabajo describe una metodología de enseñanza enfocada en la teoría con el propósito de facilitar la comprensión e integración de los temas y conceptos abordados en la unidad 4 a través del uso de mapas conceptuales.

Palabras Clave
Mapa conceptual, Enseñanza, Paradigma de Programación, Paradigma Orientado a Objetos.

Introducción

La asignatura Paradigma de Programación pertenece al segundo año de la carrera de Ingeniería en Sistemas de Información. Cuya modalidad de cursado es cuatrimestral y tiene una carga horaria semanal de 8 horas.
[1] Los objetivos que persigue la asignatura es que el estudiante:

- Comprenda los fundamentos de los paradigmas básicos que son utilizados por los lenguajes de programación actuales.
- Conozca el modelo formal o semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación correcto.
- Aplique los diferentes paradigmas en la resolución de problemas.

El programa analítico de la asignatura está conformado por seis unidades temáticas:

Unidad	Denominación	Tiempo Dictado
Nro.1	Introducción a los Lenguajes y Paradigmas de Programación.	1 semana
Nro.2	Elementos constitutivos de los lenguajes y paradigmas.	1 semana
Nro. 3	Paradigma Imperativo: Secuencial o Estructurado.	1 semana
Nro. 4	Paradigma de Programación con Orientación a Objetos.	4 semanas
Nro. 5	Paradigma de Programación Lógico.	3 semanas
Nro. 6	Paradigma de Programación Funcional.	3 semanas

Tabla 1: Unidades Temáticas de la asignatura Paradigma de Programación.

La motivación de este trabajo surge como una estrategia didáctica que intenta optimizar el proceso de enseñanza y aprendizaje en la faz teórica con el propósito de favorecer la integración de los temas aprendidos en la unidad y articulación con otros ya aprendidos.
La mayoría de las unidades de esta asignatura se caracteriza por presentar una cantidad significativa de conceptos teóricos, que en muchos casos hace que el alumno presente dificultades para retomar los mismos conceptos e ideas una y otra vez con el objetivo de adquirir profundidad y comprensión de lo aprendido.

Este trabajo constituye una primera aproximación con el objetivo de dar respuesta a la siguiente interrogante: ¿La utilización de mapas conceptuales facilita a los estudiantes la comprensión de conceptos y la relación entre los mismos?

El contexto de este trabajo se ha planteado en el dictado de la primera clase correspondiente a la Unidad N° 4: Paradigma de Programación con Orientación a Objetos, en la que se imparte una introducción al paradigma abordando los conceptos claves y relevantes del mismo.
Los objetivos específicos que persigue esta unidad es que el alumno:

- Comprenda acabadamente los mecanismos fundamentales que dan origen a este paradigma.

- Utilice para la resolución de problemas un lenguaje con OO puro en su concepción y representación.

Con el propósito de dinamizar el dictado de clases en la teoría y favorecer el proceso de enseñanza y aprendizaje, he considerado como recurso didáctico el uso de mapas conceptuales.

[2] Un mapa conceptual es una técnica cada día más utilizada en diferentes niveles de la educación. Se lo utiliza como herramienta para el aprendizaje y la evaluación, permitiéndole al docente explorar los conocimientos previos de los alumnos; mientras que, al alumno le permite organizar, interrelacionar y fijar el conocimiento del contenido estudiado.

[3] Los mapas conceptuales, desarrollados por Novak y Gowin (1984), se usan como un medio para la descripción y comunicación de conceptos dentro de la "Teoría de Asimilación" propuesta por Ausubel. La teoría está basada en un modelo constructivista de los procesos cognitivos humanos. El aprendizaje significativo pone énfasis en la creación, evolución y relación entre los conceptos.

Aunque Ausubel no menciona a los mapas conceptuales en su teoría, los mismos fueron desarrollados para promover el aprendizaje significativo. Un aprendizaje es significativo cuando una nueva información (concepto, idea, proposición) adquiere significados para el estudiante a través de una interacción con la estructura cognitiva preexistente.

[4] La construcción de un mapa conceptual se inicia con una pregunta central la cual se responderá con el contenido del mapa. Generalmente las preguntas que requieren una explicación, en vez de una simple descripción, llevan a construir mejores mapas conceptuales.

El objetivo del presente trabajo es presentar el diseño de un mapa conceptual correspondiente al dictado de la primera clase de la Unidad Nro. 4: Paradigma Orientado a Objetos; con el propósito de facilitar que el alumno logre afianzar de manera más simple los conceptos como evidenciar las asociaciones entre los mismos y conceptos ya aprendidos.

Elementos del Trabajo y metodología

[5] Es de vital importancia la determinación de la pregunta inicial que se intenta responder con el mapa conceptual, que en este caso es conocer cuáles son los aspectos relevantes del Paradigma Orientado a Objetos. A continuación se detallan las etapas consideradas para la construcción del mapa conceptual.

i) Selección de material: en esta etapa se identifica el material bibliográfico que se considera como base, que en este caso es el Material de la cátedra, complementado con la bibliografía propuesta por la cátedra.

ii) Enumeración de los conceptos claves: se realiza un listado de los conceptos

(palabras, hechos o ideas) relacionadas con el tema que surgen a partir de la lectura de libros, material de cátedra y de las experiencias personales. En ésta etapa no se tiene en consideración la redundancia de conceptos, la importancia relativa o las relaciones entre ellos.

iii) Organización: Se reúnen los conceptos en grupos y subgrupos, identificando términos que representen jerarquías. Se incluyen en ésta etapa posibles términos inicialmente omitidos.

iv) Diseño: Se acomodan jerárquicamente los conceptos de modo de representar las interrelaciones entre los grupos y subgrupos.

v) Conexión: Utilizando líneas y flechas se unen conceptos. Se escribe una palabra o frase corta sobre cada relación. Las proposiciones resultantes contienen dos o más conceptos conectados que forman un enunciado con sentido.

vi) Finalización del Mapa Conceptual: Controlando que todos los conceptos y las relaciones sean correctas y que no falten conceptos importantes.

A continuación se detalla el dominio de conocimiento para la construcción del mapa conceptual que es el Paradigma Orientado a Objetos.
[6] En este punto es importante recordar que un paradigma de programación es considerado como un modelo básico de diseño e implementación de programas, que permite desarrollar software conforme a ciertos principios o fundamentos específicos que se aceptan como válidos. Otra definición lo expresa como un marco conceptual que determina los bloques básicos de construcción de software y los criterios para su uso y combinación.

En el Paradigma Orientado a Objetos, se puede partir de una definición básica que caracteriza a este Paradigma: Un programa es un conjunto de Objetos que colaboran entre sí enviándose mensajes.

Objetos+Mensajes= Programa

Una de las principales ventajas de este paradigma es que suele ser adecuado para el diseño de aplicaciones informáticas relacionadas con todo lo que sea gestión.

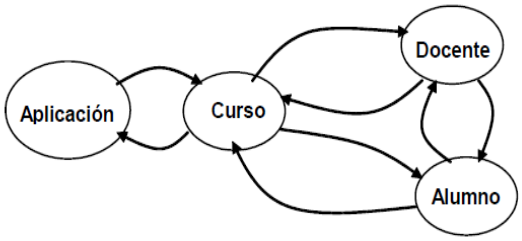


Figura 1: Conjunto de objetos colaborando entre sí.

Algunos de los lenguajes asociados con este paradigma son: Smalltalk, Simula, Java y C++.

Ventajas:

- Facilita el mantenimiento del software.
- Permite crear sistemas más complejos.
- Agiliza el desarrollo de software.
- Facilita la creación de programas visuales.
- Facilita el trabajo en equipo.
- Se basa en lenguajes que soportan sintáctica y semánticamente la unión entre los tipos abstractos de datos y sus operaciones a través de las clases.
- La programación orientada a objetos incorpora en su entorno de ejecución mecanismos tales como el polimorfismo y el envío de mensajes entre objetos.

Limitaciones:

- Tamaño de las aplicaciones. Cuando se heredan clases a partir de clases existentes se heredan de forma implícita todos los miembros de dicha clase aun cuando no todos se necesiten, lo que produce aplicaciones muy grandes.

- Velocidad de ejecución. Esto tiene que ver, en cierto modo, con el punto anterior, una aplicación innecesariamente pesada en muchas ocasiones es más lenta de ejecutar que una aplicación conformada únicamente por los módulos necesarios.

A continuación se detallan dos grupos de conceptos inicialmente reconocidos para la construcción del mapa conceptual: Componentes y Principios de Diseño de Paradigma Orientada a Objetos dentro de cada uno de estos grupos se irán incorporando otros conceptos.

Componentes Principales: Cualquier paradigma de programación condiciona la tarea de programación estableciendo un enfoque que determina la utilización de ciertos elementos o componentes para el diseño y codificación de un programa computacional.

Clase: representa un modelo a partir del cual se crean los objetos; y en el que se definen los métodos y el conjunto de variables (atributos) que tendrán los objetos que se creen a partir del modelo.

*Atributos: s*on variables que se declaran dentro de la clase, y sirven para indicar la forma de cada objeto representado por esa clase. Los atributos, de alguna manera, muestran lo que cada objeto es, o también, lo que cada objeto tiene.

Métodos: son funciones, procedimientos o rutinas declaradas dentro de la clase, usados para describir el comportamiento de los objetos descriptos por esa clase. Los métodos, de alguna manera, muestran lo que cada objeto hace.

Objeto: un objeto es una unidad de software de la cual lo que me importa es: qué le puedo preguntar y/o pedir, y a qué otros objetos conoce. Los objetos responden a los pedidos interactuando con los otros objetos que conoce, y así se va armando una aplicación. Para resumir un objeto posee características y métodos.

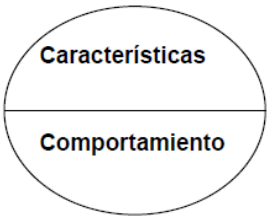


Figura 2: Estructura interna de un objeto

Principios de Paradigma Orientada a Objetos: Todo paradigma de programación se apoya en cierto "pilares" o principios que son indiscutidos e incuestionables. Ellos son:

Abstracción: es la operación intelectual que ignora selectivamente partes de un todo para facilitar su comprensión. En la resolución de problemas: implica ignorar los detalles específicos buscando generalidades que ofrezcan una perspectiva distinta, más favorable a su resolución. El proceso de abstraerse progresivamente de los detalles, permite manejar niveles de abstracción. Existen diferentes tipos de abstracción.

Abstracción de Datos (TDA): es una técnica de programación que permite definir nuevo tipos de datos (tipos de datos definidos por el usuario) adecuados a la aplicación que se desea realizar. Lo que permite el diseño de programas más cortos, legibles y flexibles.

Abstracción Procedimental: se define como un conjunto de operaciones (procedimiento) que se comportan como una operación. Se pone énfasis en que hace el procedimiento y como se invoca, dejando de lado como se implementa.

Abstracción de Iteración: este tipo de abstracción permite trabajar sobre colecciones de objetos sin tener que preocuparse por la forma concreta en que se organizan.

Encapsulamiento: se basa en que cada módulo en un programa computacional no conoce el funcionamiento interno de los demás módulos con los que interactúa sino sólo su interfaz, es decir, la forma en que debe enviarle información adicional en forma de parámetros y cómo va a recibir las respuestas.
Existen conceptos que están íntimamente relacionados con el Encapsulamiento que son la Modularidad y la Delegación.

Modularidad: representa una estrategia central de la programación es buscar la manera de organizar y distribuir la funcionalidad de un sistema complejo en unidades más pequeñas de software con se responsabilizan de tareas específicas y que interactúan entre ellas. Estas unidades reciben nombres diferentes según cada lenguaje de programación, como rutinas, funciones entre otros.

Delegación: consiste en la invocación que desde un módulo se efectúa a otro módulo, de manera que el que invoca explicita qué es lo que pretende y el que es invocado se ocupa de todo lo necesario para realizarlo.

Herencia: la herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente. Es la característica clave de los sistemas orientados a objeto para propiciar entre otros aspectos la reusabilidad. Este concepto tiene como característica la idea de jerarquización entre clases donde se puede la incorporación de otros dos conceptos como clase Madre e Hija.

Clase Madre o Base: Es una clase, que puede ser abstracta o no. Es decir que se puede señalar que de la misma se puedan crear o no objetos. Esta clase por ser madre o base tiene la particularidad de estar conformada por atributos que por lo general se encuentran duplicados en diferentes clases.

Clase Hija o Derivada: Se caracteriza justamente por heredar características y comportamiento de su clase madre o base. Son clases que se caracterizan por ser mas especializadas, por lo que es lógico que de las mismas pueden crearse objetos de las mismas.

Polimorfismo: Este es uno de los conceptos esenciales de una programación orientada a objetos. Así como la herencia está relacionada con las clases y su jerarquía, el polimorfismo se relaciona con los métodos.
Esta propiedad, como su mismo nombre sugiere múltiples formas, se refiere a la posibilidad de acceder a un variado rango de funciones distintas a través de la misma interfaz. O sea, que, en la práctica, un mismo identificador puede tener distintas formas (distintos cuerpos de función, distintos comportamientos) dependiendo, en general, del contexto en el que se halle inserto.

Resultados

Siguiendo las etapas mencionadas en la sección anterior se obtiene el mapa conceptual representado en la Figura 1. El mapa conceptual presentado en este trabajo tiene como objetivo que el alumno pueda tener un panorama general de los conceptos verticales y transversales de la asignatura, haciendo énfasis en la jerarquización y uso de expresiones abreviadas y significativas para los conceptos y palabras de enlace.

Al proporcionarle al estudiante un mapa conceptual como el elemento de presentación de la clase, minimiza la desorientación y la sobrecarga de información y conceptos lo cual resulta en un mejor aprendizaje.

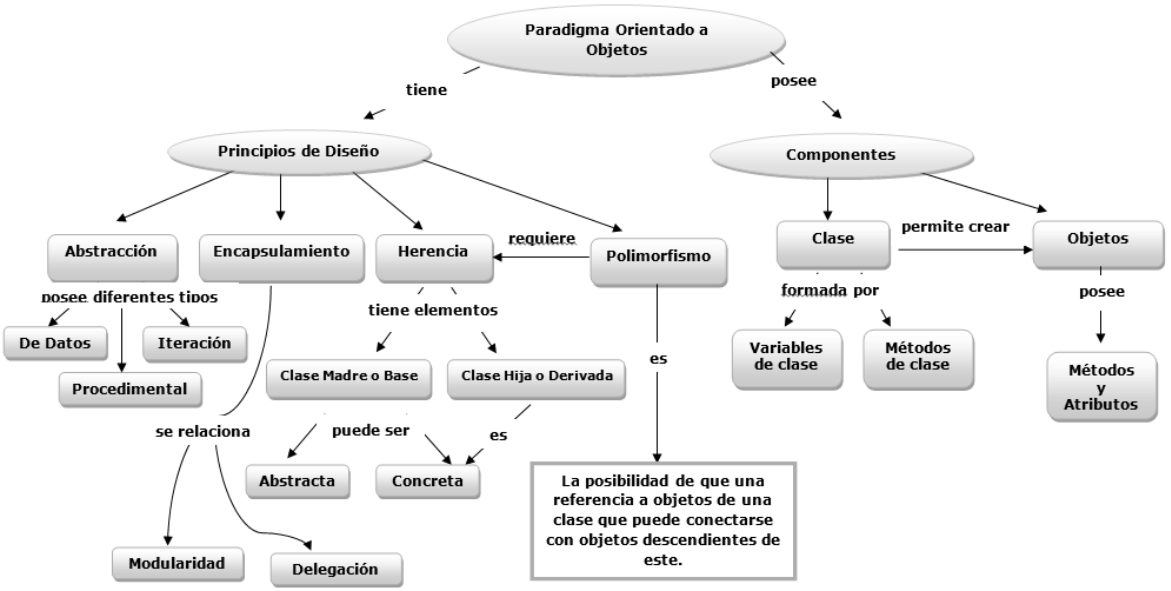


Figura 1: Diseño del Mapa Conceptual

La ventaja es que al tenerlo como recurso didáctico en la presentación de la clase es posible volver al mapa para volver a tener la visión global del tema. Un aspecto positivo de los enfoques interdisciplinares es que ayudan a combatir la segmentación de las enseñanzas entre la teoría y la práctica que pueden dar lugar, entre otros inconvenientes, a conceptos e ideas equivocas.

De una encuesta realizada a los alumnos, la gran mayoría han manifestado que el uso de mapas conceptuales como estrategia de enseñanza les ha facilitado la comprensión de los conceptos, como así también el hecho de poder relacionarlos e integrarlos incluso con conceptos y temas aprendidos en unidades anteriores.

Discusión

Esta experiencia puede ser extendida en otros escenarios dentro del contexto educativo. Cabe mencionar que el diseño del mapa conceptual presentado puede ser enriquecido con otros elementos o recursos tecnológicos digitales asociados a los conceptos. Una de las alternativas es el acceso a documentos que permite describir y caracterizar los conceptos y sus relaciones. Como así también la inclusión de gráficos que faciliten la ejemplificación de ciertos conceptos que componen el mapa conceptual.

Un aspecto no contemplado en esta experiencia, es sugerir a los alumnos finalizada la clase, la revisión del mapa conceptual con el propósito de que el alumno trabaje en el rediseño del mismo. Permitiéndoles la identificación de nuevos conceptos, jerarquizaciones y relaciones que él considere apropiado; generando un espacio para la participación, creatividad y reflexión.

Si bien el mapa conceptual presentado en este trabajo se ha implementado en el dictado de la primera clase de la Unidad 4, considero que les sería beneficioso que los alumnos al finalizar unidad se les pueda presentar a modo de cierre e integración de la unidad un mapa conceptual con todos los conceptos aprendidos en el transcurso de la unidad enfatizando la relación entre ellos, fomentando un aprendizaje significativo.
Estas consideraciones serán tenidas en cuenta en futuros trabajos.

Conclusión

El mapa conceptual construido es utilizado al iniciar la clase, para que el alumno tenga una visión general del mismo. Una vez que

se tiene el panorama completo del tema se continúa en la profundización de dichos conceptos. Los conceptos, al ir relacionándose por medio de las palabras enlace, se van almacenando en la mente de modo organizado y jerárquico de manera que serán más fácilmente comprendidos por el alumno.

A partir de esta experiencia se ha demostrado que es sumamente positivo la incorporación de los mapas conceptuales en la rutina habitual de dictado de clases.

No solo se sugiere la incorporación de mapas conceptuales en el proceso de enseñanza, si no también se recomienda incentivar a los alumnos en la utilización de mapas conceptuales como herramienta que permita favorecer y facilitar su proceso de aprendizaje.

Referencias

[1] Modalidad Académica de la asignatura Paradigmas de Programación, Ciclo lectivo 2014.

[2] Nesbit, J.C, y O. Adescope; Learning with concept and knowledge maps: A metaanalysis, Review of Educational Research: 76(3), 413–448 (2006)

[3] Novak, J.D. y D.B. Gowin; Learning How to Learn, Cambridge University Press Cambridge UK, (1984).

[4] Novak J.D y A.J. Cañas; Theoretical origins of concept maps, how to construct them, and uses in education, Reflecting Education; 29-42 ,2007.

[5] Novak, J.D y A. Cañas; Construyendo sobre Nuevas Ideas Constructivistas y la herramienta Cmap Tools para crear un nuevo Modelo para Educación. Institute for Human and Machine Cognition-
http://eduteka.org/pdfdir/CmapToolsNuevoModeloEducacion.pdf ,2006.

[6] Material de cátedra de la asignatura Paradigmas de Programación. Unidad Nro.1: Introducción a los Lenguajes y Paradigmas de Programación, Link de consulta:
http://uv.frc.utn.edu.ar/pluginfile.php/90703/mod_resource/content/1/Unidad_1%5B2014%5D.pdf,2012.

[7] Material de cátedra de la asignatura Paradigmas de Programación. Unidad Nro. 2: Elementos constitutivos de los Lenguajes y Paradigmas,
Link de Consulta:
http://uv.frc.utn.edu.ar/pluginfile.php/91046/mod_resource/content/1/%5B2014%5D%20Unidad%20Nro2.pdf ,2012.

[8] Deitel & Deitel; Como programar en Java. Séptima Edición; Pearson.Prentice Hall, 2008.

[9] Joyanes Aguilar Luis; Programación Orientada a Objetos; McGraw-Hill, SBN: 84-481-0590-7, 1996.

[10] Miriam E. Presutti, Elisa C. Miceli y Gladys M. Laporte; Mapas Conceptuales como Herramienta en la Enseñanza de Contenidos Comunes a Diferentes Cursos de Ingeniería Agronómica; Formación Universitaria – Vol. 2 Nº 6 - 2009

[11] Yoval, Saulo, Sandoval, García; Valoración Cuantitativa para valorar mapas conceptuales, Concept Maps: Theory, Methodology, Technology Proc. of the First Int. Conference on Concept Mapping Pamplona, Spain 2004

[12] Reinoso Rabago Alfonso; La construcción de los mapas conceptuales en educación a distancia, Concept Maps: Theory, Methodology, Technology Proc. of the First Int. Conference on Concept Mapping Pamplona, Spain 2004

Datos de Contacto:
Cynthia Lorena Corso. Universidad Tecnológica Nacional Facultad Regional Córdoba. Maestro M. López esq. Cruz Roja Argentina. E-mail: cynthia@bbs.frc.utn.edu.ar

Estrategias en la enseñanza de Paradigmas de Programación: de la Programación Estructurada a la Programación Orientada a Objetos.

Juarez, Gustavo Eduardo [(1),] *Nieto Peñalver, Luis Eduardo* [(2)] –
Sánchez, Mariana [(3)] *- Steifensand, Jorge Federico* [(4)]
Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán
[(1)]gjuarez@herrera.unt.edu.ar – [(2)]lnieto@herrera.unt.edu.ar –
[(3)]msanchez@herrera.unt.edu.ar – [(4)]jsteifensand@unsta.edu.ar

Abstract

La enseñanza de la programación requiere el abordaje de diversos aspectos: diseño de algoritmos, estrategias empleadas en la resolución de problemas, el uso de un lenguaje de programación, el empleo de herramientas. En este trabajo se pretenden plasmar las dificultades que se presentan en el abordaje de conceptos de la programación estructurada y orientada a objetos, y las estrategias didácticas para la enseñanza, a implementarse en el eje curricular programación de la carrera de Ingeniería en Computación de la Facultad de Ciencias Exactas de la UNT.

Palabras clave

Enseñanza, Estrategias, Paradigmas, Programación Estructurada, Programación Orientada a Objetos.

1. Introducción

La enseñanza de la programación requiere el abordaje de diversos aspectos: diseño de algoritmos, estrategias empleadas en la resolución de problemas, el uso de un lenguaje de programación, el empleo de herramientas (entornos de programación, compiladores, depuradores, etc).

La carrera de Ingeniería en Computación de la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán tiene en su plan de estudios dos asignaturas fuertemente articuladas entre sí, Programación I y Programación II. Estas corresponden al inicio de la carrera y al eje curricular de la programación. Los paradigmas con los que se presentan la enseñanza de estas asignaturas son

En el caso de las carreras Ingeniería Eléctrica, Electrónica y Computación la asignatura Programación I, se dicta en el primer semestre de segundo año, como correlativa de Informática. Al ser la primera asignatura de programación que se da a los estudiantes, resulta de suma importancia, por lo que se pretende que se desarrollen adecuadamente las técnicas de programación estructurada y programación orientada a objetos para la primera y segunda respectivamente.

Durante el dictado de la primera de ellas (Programación I), donde se trabaja con el paradigma estructurado, se detectó que los alumnos presentaban dificultades al momento de definir una función. Estas dificultades se presentan tanto en la determinación de los parámetros que recibe la función como así también en la identificación del tipo de dato que debe devolver dicha función.

Estas dificultades, que se presentan en Programación I bajo el paradigma estructurado, se vuelven más notorias cuando los alumnos cambian al paradigma orientado a objetos. Si bien los alumnos comprenden la analogía entre una función y un método, nuevamente se observan problemas similares que se presentaron bajo el paradigma estructurado: determinación de los parámetros y tipo de datos devueltos.

En este trabajo se pretenden plasmar ciertas estrategias didácticas para la enseñanza de la programación, en la carrera de Ingeniería en Computación de la Facultad de Ciencias Exactas de la UNT.

2. Antecedentes

La carrera de Ingeniería en Computación de la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán se dicta desde el año 1991, actualmente en el área Programación se dictan tres asignaturas Informática, Programación I y Programación II.

En la asignatura Informática, los alumnos aprenden técnicas de diseño de algoritmos y estrategias para analizar y plantear situaciones problemáticas

programación y el manejo de algún lenguaje de programación.

La asignatura Programación I comenzó a dictarse en el año 1992, con el lenguaje Pascal, en la que se incluían contenidos básicos en cuanto al uso del lenguaje, sintaxis, funciones, tipos de datos y arreglos, tomando como base el paradigma de la programación

estructurada. En el año 1994 se incluye en el programa de la asignatura el tema registros, que hasta ese momento, formaba parte del contenido de la asignatura Programación II. Luego, en el año 1998, por pedido de los estudiantes de las carreras de Ingeniería Electrónica, y con posterior adhesión de las carreras de Ingeniería. Eléctrica y en Computación, se modificó el dictado de la asignatura Programación I para adoptar el uso de Lenguaje C como base para la enseñanza del contenido.

En el año 2004, con la modificación de los planes de estudios, se elimina la asignatura Programación II de las carreras de Ingeniería Eléctrica y Electrónica, por lo que toda la unidad temática referida a archivos pasa a formar parte de los contenidos a dictarse en Programación I.

La asignatura Programación II se dicta en el segundo semestre de segundo año, y comenzó a dictarse empleando el lenguaje C++, como base del paradigma orientado a objetos, para enseñar los contenidos de la misma.

Actualmente la asignatura Programación II trata con la programación orientada a objetos, dejando para asignaturas de años superiores el análisis y el diseño orientados a objetos. Esta asignatura, a diferencia de Programación I, sólo se imparte para la Carrera de Ingeniería en Computación, empleando el lenguaje Java, ya que se consideró el más adecuado para estos alumnos. Además en las asignaturas de los siguientes años se sigue utilizando este lenguaje para profundizar los conocimientos del mismo.

Durante las clases prácticas y teóricas de estas dos asignaturas se detectó que los alumnos presentan dificultades con temas relacionados con conceptos básicos de programación, específicamente la definición de funciones (programación estructurada) o métodos (programación orientada a objetos). Estos problemas tienen que ver con la definición de los parámetros y tipo de retorno de las operaciones. A lo largo del tiempo se realizaron diversas modificaciones en los contenidos y herramientas.

3. Los paradigmas de programación: Programación Estructurada y Programación Orientada a Objetos

La programación estructurada es un paradigma enfocado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora. Surgió en la década de 1960, particularmente del trabajo de Böhm y Jacopini, expresado en el Teorema Fundamental de la Programación estructurada que señala que con solo estructuras de secuencia, selección e iteración combinadas es posible expresar cualquier función computable. Con la evolución de las estrategias de programación se hace uso del diseño modular, mediante el cual se tiene como premisa el dividir el problema en problemas más pequeños hasta el punto en cada uno no pueda dividirse más. En resumen, con el paradigma estructurado se intenta modelar la solución a un problema como un conjunto de pasos, acciones, procedimientos o funciones que se ejecutan en secuencia para arribar a la misma. Bajo este paradigma, se debe pensar cuáles son las cosas que se deben hacer, lo cual lleva a pensar en qué funciones se necesitan.

El paradigma de la programación orientada a objetos (POO) se basa en módulos que encapsulan tanto los datos como las acciones del problema. Bajo este paradigma, se debe pensar por qué cosas está formado el programa, lo cual lleva a pensar en cuáles son los actores principales que participan. A las cosas por las que está formado un programa se las llama objetos, y sus tipos se denominan clases (conceptos fundamentales). Programar bajo este paradigma implica por lo tanto identificar estos objetos o actores principales, reproduciendo adecuadamente sus características y comportamientos.

¿Por qué la necesidad de definir un nuevo paradigma? Distintas razones llevaron a la definición de un nuevo paradigma:

- Fracaso de la programación estructurada en la producción y mantenimiento sistemático de sistemas de calidad.
- Bajo el paradigma estructurado, realizar modificaciones implica mucho esfuerzo.
- Utilizando el paradigma estructurado, ensamblar componentes producidos por otros es difícil y muy propenso a la propagación de errores.

4. Situación problemática

La problemática surgió en las asignaturas del área programación y fue detectada durante las reuniones de articulación de las cátedras que componen la misma.

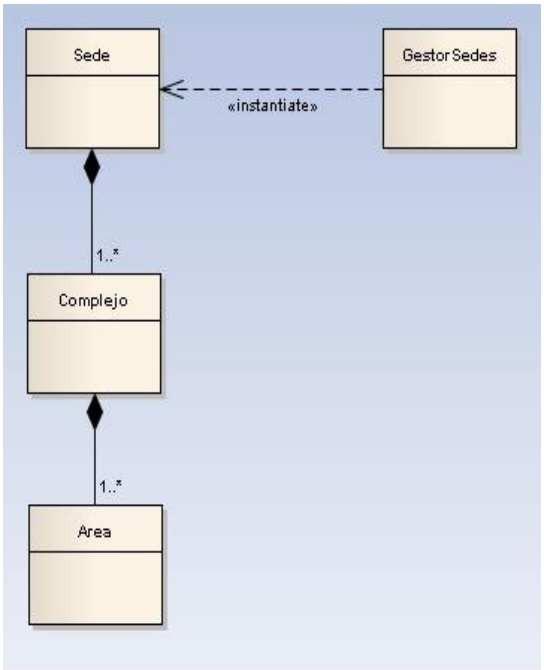


Figura 1. Diagrama de clases

Para detectar los alcances de esta problemática se decidió realizar trabajos prácticos que permitieron realizar un diagnóstico en el dictado de la asignatura Programación II en el año 2013.

En el desarrollo de los trabajos prácticos de la asignatura Programación II se detectaron diversas dificultades en la adquisición de conceptos básicos de programación. A modo de ejemplo se muestra el diagrama de clases de la **Figura 1** y los errores encontrados en la solución dada por los estudiantes.

La solución dada por los estudiantes contiene lo siguiente:

Clase Sede:

```
String editComplejo(C:Complejo); (a)
```

```
String nuevaSede(nombre:String); (b)
```

Clase GestorSedes:

```
String modificaSede(S:Sede); (c)
```

```
String nuevaSede(S:Sede); (d)
```

Clase Complejo:

```
String newA(n:String,C:Complejo);(e)
```

```
List<Complejo> capacidad();(f)
```

Clase Area:

```
List<Area> listaAreas(); (g)
```

Se observan los siguientes errores:

- En los casos (a) y (c) los métodos enunciados no contienen los datos suficientes con los que se van a realizar las modificaciones.
- En el caso (b) se requieren los atributos para crear una Sede. Por otro lado, esta operación no pertenece a esa clase.
- En el caso (d) el método está incompleto ya que no posee los datos con los que se va a realizar la operación: se necesitan los atributos y no la Sede.
- La operación del caso (e) se definió sobre la clase Complejo, por lo que no se necesita el parámetro de tipo Complejo.
- La operación del caso (f) no puede devolver un listado de Complejo.
- En el caso (g), el error se presenta ya que a un Area no se le puede pedir información sobre otras Areas.

Los estudiantes presentan dificultades al momento de relacionar una función o un método con una tarea. Toda tarea realiza una acción para lo cual necesita determinada información (datos de entrada o parámetros de la función) y produce un resultado (datos de salida).

Una de las causales de error es la dificultad para abstraerse del proceso de obtención del dato, ya que al definir una función o método solo se requiere conocer cuáles son los datos de entrada necesarios. Por ejemplo, si una determinada función requiere la fecha actual para realizar su trabajo, la misma debe ser pasada como dato de entrada, independientemente de cómo se la hubiera obtenido (leyendo la fecha actual de la máquina, de un archivo, como resultado de otra función, etc).

Otra dificultad se presenta al momento de descomponer el problema en partes más simples, con un propósito bien definido, los estudiantes intentan incluir en un mismo método o función tareas que pueden ser realizadas por otros métodos. En lugar de descomponer el problema terminan agrupando acciones, que luego dificultan el mantenimiento debido al alto acoplamiento entre los distintos métodos. Esta situación lleva a que los estudiantes definan la mayoría de las funciones para que no devuelvan valor (*void*), sin tener en cuenta que el resultado de una función puede ser la entrada de otra.

5. Propuesta de solución

Dentro de las estrategias didácticas que se vienen implementando durante el año 2014 se encuentran las siguientes:

- Utilizar un vocabulario común que permita identificar claramente los conceptos empleados en Programación I y Programación II. Por ejemplo, con la unidad temática destinada a Registros usar vocabulario que permita hacer una analogía con Clases que se utilizan en Programación II.
- Presentar ejemplos de problemas complejos, resueltos de dos modos diferentes: con una única función o método en el que se involucran varias tareas, y luego dividiendo el problema mediante el uso de varias funciones o métodos más simples y con tareas más específicas, para evidenciar la dificultad en el mantenimiento del código cuando se requiere una modificación.
- Identificar, al momento de analizar el problema, los datos de entrada, proceso y resultados, lo que facilitará la definición de las funciones que se requieran para resolverlo.

Las propuestas antes mencionadas se complementan con la consulta a bibliografía, artículos y foros en Internet, etc.

6. Conclusiones

La estrategia propuesta se comenzó a implementar en el año 2014. Durante el primer semestre se realizó un seguimiento y evaluación de resultados en la

asignatura Programación I. Como medida de la efectividad de las estrategias propuestas se tiene en cuenta la aprobación de los trabajos prácticos.

Al finalizar el dictado de la asignatura Programación I se realizó un estudio comparativo de la proporción de alumnos que aprobaron los trabajos prácticos frente al total de alumnos que cursaron la asignatura desde el año 2011.

En la **Tabla 1** y **Figura 2** se puede observar que como consecuencia de la implementación de las estrategias propuestas, la cantidad de alumnos que aprobaron los trabajos prácticos durante el primer semestre de 2014 se incrementó en un 51% con respecto al promedio de los años 2011 a 2013.

Año	Total	Regulares	Porcentaje
2014	90	61	68%
2013	88	34	39%
2012	93	40	43%
2011	106	45	42%

Tabla 1. Estudio comparativo – Programación I

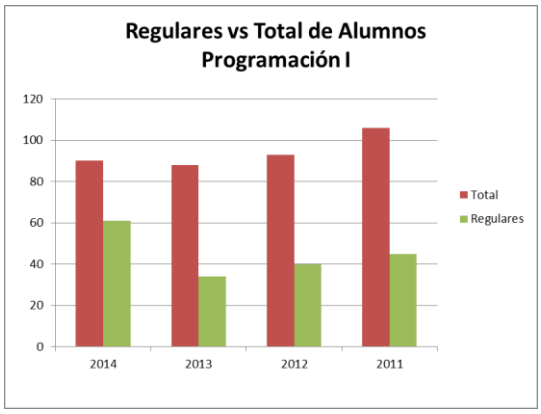


Figura 2. Estudio comparativo – Programación I

En la segunda mitad del año se analizará la eficacia de las acciones implementadas en la asignatura Programación I durante el dictado de la asignatura Programación II.

El resultado de estas acciones, en caso de ser positivos, redundará en beneficios de las asignaturas del eje curricular de programación correspondientes al ciclo superior de la carrera.

7. Referencias

Venegas, Carlos Alberto, "Programación Orientada a Objetos frente a la Programación Estructurada", *Universidad Distrital Francisco José de Caldas, Colombia.* **http://comunidad.udistrital.edu.co/cavanegas/articulos/** (accedido 07/08/2014)

Molina, Manuel, "Programación en Lenguajes Estructurados. Grado Superior ", Mc Graw-Hill – 2006 - ISBN: 8448148703.

Deitel, Harvey M., Deitel, Paul J, "*Cómo programar en C / C++"*, 2da. Edición, Person Educación, México, 1995.

Deitel, Harvey M., Deitel, Paul J, "*Cómo programar en C++"*, Person Educación, México, 2003

Deitel, Harvey M., Deitel, Paul J, "*Cómo programar en Java"*, 5ta. Edición, Person Educación, México, 2004

Integración de Contenidos mediante Programación de Agentes Virtuales en Plataformas de Juegos

Frittelli, Valerio – Steffolani, Felipe – Teicher, Romina – Tartabini, Marcela – Bett, Gustavo – Serrano, Diego – Strub, Ana – Fernández, Julieta – Destéfanis, Eduardo – Cuadrado, Alexis – Domenech, Mauricio
Departamento Ingeniería en Sistemas de Información
Facultad Regional Córdoba - Universidad Tecnológica Nacional
Córdoba (5000) – Argentina
vfrittelli@gmail.com, fsteffolani@gmail.com, rteicher@gmail.com, mtartabini@gmail.com, gfbett@gmail.com, diegojserrano@gmail.com, anastrub@gmail.com, jujulifer@gmail.com, edestefanis1@gmail.com, alexiscuadrado@hotmail.com, mauriciodomenech@icloud.com

Abstract

El desarrollo de juegos para la aplicación de estrategias por parte de jugadores virtuales (o "agentes virtuales") es muy usado en educación superior en carreras informáticas para la integración práctica de contenidos. Existen diversas plataformas de desarrollo genéricas que funcionan como marcos de trabajo que permiten plantear escenarios para juegos de adversarios enfrentados, en los que luego se pueden insertar agentes programados específicamente para cada escenario. Este artículo presenta una clasificación de algunas de las plataformas más divulgadas, así como una breve descripción de las características, ventajas y desventajas de cada una desde la perspectiva de la integración de contenidos en carreras informáticas de nivel superior.

1. Introducción

Un problema común en la implementación de planes de estudio de carreras de nivel superior es el de la integración de contenidos entre asignaturas de distintos niveles o años de cursado, o bien (y quizás con mayor dificultad) entre asignaturas del mismo nivel. Específicamente, este trabajo está centrado en la forma en que se puede contribuir a esa integración utilizando plataformas de desarrollo de juegos (o *motores de juegos*) en carreras universitarias de base informática [1].

En general el problema del desarrollo de juegos y escenarios para la aplicación de estrategias por parte de jugadores y/o agentes virtuales [2] es muy usado actualmente en educación superior en carreras informáticas y/o de base electrónica para el desarrollo de conocimientos y aptitudes referidos a programación, diseño de componentes, algoritmos, estructuras de datos, probabilidad y estadística, inteligencia artificial, diseño de interfaces de usuario y aplicaciones gráficas [3]. Los campos que se acaban de citar son básicamente propios del área de la programación de aplicaciones, pero el desarrollo de juegos permite además ampliar los alcances integrando contenidos y prácticas de otras disciplinas como la física, la trigonometría, el análisis matemático, el álgebra, el cálculo numérico, la simulación y la investigación de operaciones, por citar algunas de las más comunes.

El diseño y programación de un juego de computadora, aún con fines didácticos, es una tarea muy desafiante y no sólo para los estudiantes: el equipo docente debe esmerarse y comprometerse especialmente con el diseño y la planificación de tal actividad para mantenerla acotada en alcances y tiempos de presentación y evitar así que los estudiantes se vean desbordados tanto en sus reales capacidades para lograr los objetivos propuestos, como en los plazos de finalización y entrega. El planteo de un juego (incluso si aparenta ser simple y directo) involucra innumerables factores y campos de aplicación (por lo cual son justamente elegidos para favorecer la integración de contenidos) y la subestimación de cualquiera de ellos puede llevar al fracaso de un proyecto de cátedra o a un recorte de alcances que sin llevarlo al fracaso, lo desvirtuarían.

En ese sentido, está claro que un primer enfoque posible consiste en plantear desde cero el diseño y el desarrollo del juego elegido por parte de los estudiantes. Esta vía es perfectamente aplicable y con esta base se han realizado numerosas experiencias en todo el mundo, con resultados muy bien documentados [3] [4]. Sin embargo,

como se dijo, la propuesta del diseño e implementación desde cero en un curso de programación de un semestre (avanzado o no), puede imponer restricciones de alcance y/o tiempo difíciles de acotar y controlar [5]. Para enfrentar estas restricciones, una estrategia a seguir suele ser la de entregar a los estudiantes algún modelo diseñado y pre-programado por el equipo docente, de forma de exponer en el aula los fundamentos del modelo [6] y luego dejar para los estudiantes la consigna de completarlo agregando o terminando diversas funcionalidades [7].

Un segundo enfoque consiste en la utilización de plataformas de desarrollo de juegos (o *motores de juegos*) [1] que ya provean uno o más escenarios, configurables o no, de forma que la tarea del estudiante sea la de programar y proveer los *jugadores virtuales*, también designados como *personajes virtuales* o más genéricamente, como *agentes virtuales*. Existen muchas plataformas de desarrollo de juegos genéricas ya disponibles en modalidad de *software libre* y/o como proyecto *open source*: se trata de entornos de programación integrados que ya habilitan en forma de *framework* (o marco de trabajo) más o menos expansible la posibilidad de desarrollar escenarios flexibles para juegos y situaciones de adversarios enfrentados, en los que luego puedan insertarse agentes o jugadores programados "ad hoc" (ya se trate de jugadores virtuales o directamente de robots en base a componentes de hardware) [5]. Esto lleva a que cuando se pida incorporar un jugador o agente a un motor ya desarrollado, el mismo debe implementarse e integrarse normalmente en base a la implementación de alguna clase de interfaz que provea el comportamiento esperado para el agente, o la derivación de una clase abstracta que permita un modelado más preciso de cada personaje virtual. De esta forma, el trabajo en general se simplifica: el equipo docente puede partir de la selección de un motor específico, plantear en él un escenario (o modificar alguno pre-existente) y especificar en una clase abstracta o en una clase de interfaz las reglas que los estudiantes deben programar en cada agente, sin preocuparse por los detalles de implementación del propio escenario ni la integración con el mismo.

2. Plataformas y API disponibles

Obviamente, se cuenta con *Interfaces de Programación de Aplicaciones (API)* muy conocidas como *DirectX* [8] que proporciona un amplio conjunto de elementos de programación para gestionar gráficos, sonidos, música, dispositivos de entrada y soporte de red, o como *OpenGL* [9] que provee soporte para el desarrollo gráfico. En esta línea se puede citar también a *XNA* [10], que es un conjunto de herramientas para facilitar el desarrollo y administración de juegos de video. Pero si bien estas API proveen herramientas para el desarrollo, no constituyen una plataforma para juegos específicos ya creados en los que puedan incorporarse agentes externos.

Existen, no obstante algunos productos ofrecidos para el desarrollo de agentes en juegos específicos, tales como *Robocode* (disponible desde 2006 como plataforma *open source* a través de *SourceForge*) [11] que ofrece un framework para el planteo de simulaciones de batallas entre tanques virtuales, tanto para plataforma *.NET* como para plataforma *Java* y también integrable con IDE *Netbeans*.

En forma similar, el motor *M.U.G.E.N.* [12] se ofrece como una plataforma para el desarrollo de juegos de lucha, que puede ser utilizado por programadores con escasa experiencia. En esencia, es simple incorporar nuevos personajes y configurar su aspecto, pero los ajustes de movimientos se hacen a través de un lenguaje de programación propio del motor. Muchos juegos de lucha muy populares tienen versiones basadas en *M.U.G.E.N.* El producto pertenece a la organización *Elecbyte* y se distribuye como software libre para propósitos no comerciales.

En la misma línea, se ofrece el proyecto *Pogamut* [13], que es un *middleware* desarrollado en *Java* para facilitar el control de agentes virtuales en múltiples entornos de algunos motores de juegos. Básicamente, *Pogamut* provee una API para crear y controlar agentes virtuales y una interfaz visual de usuario (un plugin para el IDE *NetBeans*) que simplifica la depuración de un agente ya creado. *Pogamut* contó inicialmente con el soporte del *Ministerio de Educación de la República Checa* y hoy se distribuye bajo licencia *GNU GPL v3*.

Uno de los motores de juegos más usados desde que apareció su primera versión es *RPG Maker*, de la compañía *Enterbrain* [14]. La plataforma *RPG Maker* permite el diseño de juegos en múltiples escenarios, con una interfaz de usuario sencilla e intuitiva. Si bien varios de los subproductos de *RPG Maker* requieren comprar una licencia, se ofrece al menos una versión *free*: *RGP Maker VX Ace Lite*.

En términos de plataformas más generales, pero de aplicación más académica, un producto cada vez más utilizado en enseñanza de Ciencias de la Computación es *GreenFoot* [15]. Se trata de un IDE para creación de personajes que luego son ubicados en *mundos virtuales* que forman parte de juegos, simulaciones y otros programas gráficos. El IDE ofrece una interfaz de trabajo visual e interactiva y los personajes se programan en *Java estándar*. El IDE fue desarrollado en la Universidad de Kent, Canterbury, Reino Unido, por el mismo equipo de trabajo que creó el IDE *BlueJ* para desarrollo y enseñanza de la programación en *Java* [16].

Un clásico en cuanto a entornos para diseño de juegos es *GameMaker*, de la organización *YoYo Games* [17]. Es un IDE integrado que permite crear tanto escenarios

como jugadores virtuales, con extrema sencillez. Los objetos que participan del desarrollo se colocan visualmente en el escenario mediante *drag and drop* y luego se configuran sus propiedades hasta el detalle que el programador requiera. En principio, no se requiere escribir código fuente pero entre sus recursos avanzados *GameMaker* ofrece un lenguaje de programación propio de la plataforma, llamado *GameMaker Languaje (GML)*.

El proyecto *Alice* [18] es una iniciativa multi-universitaria con participación de especialistas de universidades estadounidenses como *Carnegie Mellon*, *Washington* y *Stanford*. Ofrece un entorno de programación gráfica en 3D mediante el cual los estudiantes aprenden los fundamentos de la programación a través de la animación de objetos en pantalla, que ellos mismos ubican con el simple recurso de *drag and drop*. La interfaz interactiva de *Alice* permite que los estudiantes establezcan y comprendan en forma inmediata el uso de una instrucción de programación, ya que el efecto de ésta se aplica en forma directa al comportamiento visual de los objetos diseñados. Esencialmente, *Alice* permite el diseño de animaciones para contar una historia, jugar un juego de video simple o confeccionar un video para ser compartido en la web.

Si se trata de diseñar y desarrollar *robots de hardware* (y ya no agentes virtuales), existen numerosas iniciativas basadas en la programación de placas o tarjetas electrónicas configurables que pueden usarse para controlar dispositivos de todo tipo a partir de sensores, motores paso a paso, leds, relés, baterías y conectividad USB. Si bien estos proyectos no se basan en la inserción de un agente virtual en un escenario predefinido como se explicó para los motores de juegos, constituyen excelentes plataformas de partida para integración de contenidos muy diversos (tanto a nivel superior como nivel de escuela media) a través de una introducción a la robótica. A este grupo de iniciativas pertenece el proyecto *GoGo Board (Stanford University)* [19], de libre acceso tanto para los diseños de las placas como para el firmware de control y el software de programación (normalmente, lenguaje Logo aunque se pueden usar API para C++ y Java). Existen además productos como *Lego Mindstorms* [20] que proveen kits de desarrollo con piezas muy variadas para el armado de robots. Si bien el kit de hardware debe comprarse, las actualizaciones del firmware y el software de desarrollo se pueden descargar en forma gratuita.

3. Competencias de Programación

Se puede citar que en diversos lugares del mundo se organizan cada año competencias de proyectos estudiantiles sobre temas de ciencias de la computación aplicadas en juegos y escenarios de adversarios, que van desde plataformas de juegos con agentes virtuales hasta competencias en las que directamente se presentan robots de hardware. Concretamente, el *AI Challenge* [21] es un concurso internacional de programación e inteligencia artificial inicialmente lanzado por la *Universidad de Waterloo* (Canadá). Originalmente estaba planteado solo para alumnos de esa universidad, pero desde 2010 obtuvo el apoyo de *Google* y se abrió a estudiantes y público en general de todo el mundo. En el concurso, cada participante debe desarrollar un programa auto contenido que sea capaz de jugar un juego frente a un oponente. El programa se sube a un servidor, el cual controla las estrategias aplicadas y determina un ganador. En general, el desarrollo de programas para participar de este evento no es una tarea compleja, ya que el servidor y las reglas aplicadas finalmente proponen un marco de trabajo consistente y transparente para los programadores.

En forma similar, la *ACM (Association for Computing Machinery)* [22] organiza cada año el *ICPC Challenge (International Collegiate Programming Competition Challenge)*, en el cual los participantes deben proveer la programación de jugadores virtuales, y hacer competir a esos jugadores contra otros en un juego específico (que cambia cada año). La competencia implica un proceso tipo *play off* por doble eliminación, al final del cual se proclama al programador vencedor.

El proyecto *Robocode* también ha generado en el mundo numerosas instancias de competencias en las que se busca que distintos equipos de programadores creen el mejor tanque virtual, capaz de vencer a todos los demás en diversas rondas de batallas simuladas. [11] Específicamente, al momento de escribirse este documento, la *Facultad Regional Córdoba de la Universidad Tecnológica Nacional* tiene en pleno desarrollo un curso introductorio sobre *Robocode* para estudiantes de la carrera de *Ingeniería en Sistemas de Información*, que culminará con su primera *Competencia Robocode* en Octubre de 2014 [23].

El proyecto *Alice*, por su parte, es la plataforma de desarrollo elegida para soporte del *Desafío Dale Aceptar* impulsado por la *Fundación Dr. Manuel Sadosky* [24] y orientado al desarrollo de animaciones y juegos por parte de estudiantes de colegios secundarios de toda la Argentina. Los principios de esta iniciativa pueden extrapolarse sin problemas al ámbito de la educación superior como complemento de la enseñanza de la programación en cursos introductorios.

El proyecto *GoGo Board* ha despertado mucho interés en diversos países, como Estados Unidos, Brasil, México o Tailandia, generando competencias y muestras o ferias estudiantiles de muy amplio espectro temático [25]. En Argentina, la *Universidad Blas Pascal (UBP) (Córdoba)* ha instrumentado desde 2011 en forma continua proyectos de investigación [26] [27] orientados al estudio y aplicación de las características de la *GoGo Board*, generando a partir de esos proyectos diversas instancias

de transferencia e integración temática basadas en robótica introductoria hacia escuelas de nivel medio, y también programas internos de integración de contenidos de asignaturas de sus carreras de Ingeniería en Informática e Ingeniería en Telecomunicaciones [28]. La propia *UBP* organiza cada año las *Olimpiadas Nacionales de Informática, Electrónica y Telecomunicaciones (ONIET)* para estudiantes de escuela media. Desde la edición 2012 se incluye en la *ONIET* un capítulo destinado a presentaciones y competencias de proyectos basados en *GoGo Board* [29].

4. Integración de Contenidos: Conclusiones

Como se indicó, uno de los objetivos que se plantean con el uso y aplicación de motores de juegos en la educación superior es el de favorecer la integración de contenidos de asignaturas de carreras de base informática y/o electrónica.

Es sabido el esfuerzo que debe hacerse desde la planificación de las asignaturas de una carrera como las indicadas por integrar asignaturas de diferentes áreas aun cuando existan temáticas, prácticas o incluso docentes en común. Si bien es habitual identificar (al menos implícitamente) alguna estrategia de integración entre asignaturas correlativas (integración vertical), no ocurre lo mismo con asignaturas correspondientes a un mismo nivel (integración horizontal) ni con asignaturas de otras áreas temáticas.

Concretamente, muchas asignaturas correlativas de programación (en distintos semestres) de una carrera informática incluyen el estudio de un amplio conjunto de estructuras de datos y sus algoritmos relacionados, permitiendo al estudiante dominar y poner en práctica técnicas de programación cada vez más complejas. En ese sentido, la integración vertical se da en forma natural, aunque normalmente restringida a situaciones prácticas similares o muy generales.

Sin embargo, el diseño de una actividad (evaluable o no) para asignaturas de áreas de aplicación *diversas* no suele ser simple, por cuanto no termina de quedar claro con qué dominio de problemas trabajar para asegurar que se apliquen en forma combinada conocimientos de esas áreas. Es muy conocido el fuerte impacto que tienen en la carga horaria del plan de estudios de una carrera de Ingeniería en Sistemas o Ingeniería Informática las asignaturas de ciencias básicas como Física, Álgebra, Análisis Matemático o Química, pero también se sabe lo difícil que resulta definir un plan de trabajo en estas asignaturas que permita aplicar lo visto en materias troncales o más propias del núcleo disciplinar de la carrera, como Algoritmos y Estructuras de Datos, Programación Orientada a Objetos o Matemática Discreta (por citar sólo algunos espacios curriculares muy comunes).

En forma recíproca, muchas actividades de las asignaturas del área de la programación podrían verse muy favorecidas si se dispusiese de actividades diseñadas para integrar en forma natural conceptos y prácticas de materias del campo de las ciencias básicas.

El diseño de juegos, en este caso a partir de la base brindada por un motor de juegos que facilite el montaje de escenarios y la configuración de agentes, se presenta como una opción eficiente y comprobable para la propuesta de prácticas y evaluaciones que promuevan la integración.

Inicialmente, es fácil observar que el desarrollo de un juego incluirá el manejo y aplicación de prácticamente todas las *estructuras de control de flujo* disponibles en cualquier lenguaje de programación, cualquiera sea el paradigma en el que se base, y lo mismo vale para el planteo de *subrutinas*. Las *estructuras de datos nativas*, como los *arreglos* o las *listas predefinidas*, surgen un poco más adelante en las asignaturas introductorias, pero son incorporables sin esfuerzo al diseño de la estructura de un juego.

Si se trata de integrar un paradigma de programación como la *programación orientada a objetos o POO*, la mayoría de los motores de juegos aquí nombrados trabajan en forma implícita o explícita con POO por lo que su incorporación resulta muy intuitiva incluso para estudiantes que recién se inician en la POO o tienen escasa experiencia.

La implementación y aplicación de *estructuras de datos abstractas* de muy diversa naturaleza (listas, árboles, montículos binarios, grafos, etc.) queda también abarcada en juegos que requieran de cierta complejidad en el núcleo o motor de inteligencia del juego, para controlar el funcionamiento autónomo de los agentes virtuales, aunque la utilización plena de cada estructura dependerá de la transparencia que permita la plataforma en el control del código fuente. En muchos de los productos citados en este trabajo, la tarea del programador consiste en *implementar una clase de interfaz* para modelar el comportamiento de un jugador virtual, o *instanciar visualmente un objeto* de una clase predefinida, con lo que muchos detalles de la implementación interna final quedan ocultos (básicamente, en *Robocode*, *M.U.G.E.N.*, *Pogamut*, *Alice*, *RPG Maker*), pero en otras plataformas (como *GreenFoot* y *GameMaker*, aunque en alguna medida también las otras plataformas citadas) el programador puede acceder a prácticamente todo el comportamiento del juego mediante un lenguaje de programación estándar o propio del motor.

En cualquier caso, la programación del comportamiento de un jugador virtual requiere la aplicación de *técnicas algorítmicas* variadas que van desde la *fuerza bruta directa* hasta la *programación dinámica*, pasando por *aplicaciones recursivas*,

backtracking, *divide y vencerás*, *algoritmos ávidos*, *algoritmos randomizados* y aplicaciones de la *inteligencia artificial*, incluyendo además el correspondiente *análisis de complejidad* que justifique la aplicación o no de una solución. Por todo lo dicho, de una forma u otra la integración vertical entre asignaturas del área de la programación en particular o de las ciencias de la computación en general, puede planificarse con detalle mediante el uso de plataformas de juegos.

En cuanto a la integración horizontal o entre asignaturas de áreas temáticas distintas, la estrategia de los juegos también aporta mecanismos útiles. Dependiendo del nivel de cursado en el que se encuentre el estudiante se pueden diseñar actividades que requieran mecanismos de control basados en conocimientos muy específicos de las ciencias básicas.

Así, el cálculo de la distancia entre dos oponentes que se enfrentan entre sí surge de la *Geometría* o de la *Trigonometría* si interviene algún valor de desplazamiento angular.

El cálculo con precisión de la trayectoria de un proyectil en un disparo se realiza mediante ecuaciones propias del campo de la *Física*.

Si se requiere simular algún tipo de reacción química en el contexto de juegos de guerra, surge en forma obvia la aplicación de la *Química*.

El cálculo de la posición de un objeto que se mueve en un escenario (cuando no es el propio motor quien lo realiza) requiere de ecuaciones de movimiento provistas desde la *Física* pero con base en el *Cálculo* o *Análisis Matemático*.

En muchos casos, el mismo control del movimiento de un objeto por parte del usuario del juego requiere aplicar algún tipo de modelo de ecuaciones diferenciales típico del *Cálculo Numérico* y la *Simulación*.

Cualquiera sea la aplicación que se requiera de la *Inteligencia Artificial*, involucrará numerosos elementos de cálculo del campo de la *Probabilidad* y la *Estadística*, además de la *Teoría de Juegos* y otros elementos abarcados por la *Investigación de Operaciones*.

En algunos tipos de juegos es necesario aplicar cálculos que permitan rotar la imagen de un objeto en pantalla, lo que lleva técnicas del *Álgebra* basadas en matrices de rotación, y muchas aplicaciones del análisis vectorial se usan para determinar la similaridad entre dos objetos (si el caso fuese necesario).

De más está decir que prácticamente todo el soporte formal del *Análisis de Algoritmos* y la *Complejidad Computacional* se basa en el *Álgebra*, el *Análisis Matemático*, las *Probabilidades* y la *Estadística*.

Todo lo anterior muestra que el campo del diseño de juegos constituye una estrategia hábil y natural para planear la forma de integrar conocimientos y prácticas entre asignaturas de una carrera informática de nivel superior, contando además con numerosas experiencias bien documentadas.

Si se trata de aplicaciones que requieren del control de un robot básico de hardware, las ya citadas herramientas como *GoGo Board* o *Lego Mindstorms* brindan un horizonte prácticamente ilimitado (y también comprobado y documentado) en cuanto a posibilidades de integración.

El diseño de un experimento, un robot o una maqueta controlada por una placa programable requiere la aplicación de múltiples conocimientos teóricos y prácticos del campo de la informática y la electrónica, tales como el armado de la propia placa (en el caso de usar *GoGo Board*), la descarga e instalación del firmware en la placa y la aplicación del lenguaje de programación que se requiera para el desarrollo de la aplicación que controlará el funcionamiento del modelo, la forma de tomar e interpretar los datos entregados por un sensor, y todo esto con el agregado adicional de los conocimientos necesarios ya citados del campo de las ciencias básicas, que seguirán siendo necesarios para controlar un movimiento, evitar una colisión o activar un mecanismo de control que cierre una puerta, encienda un led o arranque un motor.

5. Agradecimientos

Este artículo surgió como parte del trabajo desarrollado en el ámbito del proyecto de investigación *UTN1707 - Diseño de Motores de Juegos y Componentes para Enseñanza y Aplicaciones de la Inteligencia Artificial*, homologado en la *Facultad Regional Córdoba de la Universidad Tecnológica Nacional*, con el aval del *Departamento de Ingeniería en Sistemas de Información*.

6. Referencias

[1] V. Frittelli, A. M. Strub, E. Destéfanis, F. Steffolani, R. Teicher, M. Tartabini, G. Bett, J. Fernández and D. Serrano, "Motores de Juegos e Inteligencia Artificial para la Enseñanza," in Workshop de Investigadores en Ciencias de la Computacion (WICC 2013), Paraná, 2013.

[2] F. Dignum, J. Bradshaw, B. Silverman and W. van Doesburg, Agents for Games and Simulations: Trends in Techniques, Concepts and Design, Berlín: Springer, 2009.

[3] V. Frittelli, M. Tartabini, R. Teicher, F. Steffolani, D. Serrano, J. Fenández, G. Bett and A. Strub, "Desarrollo de Juegos como Estrategia Didáctica en la Enseñanza de la Programación," in CONAIISI 2013, Córdoba, 2013.

[4] C. A. Ochoa Rivera and G. Contreras Vega, "Proyecto AULA: Diseño y Aplicación de una Micro Unidad de Competencia para la Experiencia Educativa Algoritmos y Estructura de Datos II," in Congreso Nacional y Congreso Internacional de Informática y Computación 2009

(CNCIIC 2009), México, 2009.

[5] J. Schell, The Art of Game Design: A book of lenses, Burlington: Morgan Kaufmann, 2008.

[6] V. Frittelli, R. Teicher, M. Tartabini, J. Fernández and G. F. Bett, "Gestión de Gráficos (en Java) para Asignaturas de Programación Introductiva," in Libro de Artículos Presentados en I Jornada de Enseñanza de la Ingeniería - JEIN 2011, Buenos Aires, 2011.

[7] K. Salen and E. Zimmerman, Rules of Play: Game Design Fundamentals, Cambridge: MIT Press, 2003.

[8] Microsoft Corporation, "DirectX [Official Website]," 2014. [Online]. Available: http://windows.microsoft.com/es-es/windows7/products/features/directx-11.

[9] OpenGL, "The Industry's Foundation for High Performance Graphics," 2014. [Online]. Available: http://www.opengl.org/.

[10] Microsoft Corporation, "XNA Game Studio 4.0," 2014. [Online]. Available: http://msdn.microsoft.com/es-ES/library/bb200104%28v=xnagamestudio.40%29.aspx.

[11] SourceForge, "Robocode: Build the best - destroy the rest!," 2014. [Online]. Available: http://robocode.sourceforge.net/.

[12] Elecbyte, "M.U.G.E.N.," 2013. [Online]. Available: www.elecbyte.com/.

[13] European Social Fund, "Pogamut," 2014. [Online]. Available: http://pogamut.cuni.cz/main/tiki-index.php.

[14] Entrebrain, "RPG Maker: Make your own role-playing games," 2014. [Online]. Available: http://www.rpgmakerweb.com/.

[15] M. Kölling and D. McCall, "Greenfoot: Teach & Learn Java Programming," 2014. [Online]. Available: http://www.greenfoot.org/door.

[16] M. Kölling, "BlueJ - The Interactive Java Environment," 2013. [Online]. Available: http://www.bluej.org/. [Accessed Agosto 2013].

[17] YoYo Games , "GameMaker [Official Website]," 2014. [Online]. Available: http://www.yoyogames.com/.

[18] Carnegie Mellon University, "Alice: An educational software that teaches students computer programming in 3D environment," 2014. [Online]. Available: http://www.alice.org/index.php.

[19] GoGo Board - Stanford University, "Gogo Board - Science and Robotics for the Inventive Mind," 2011. [Online]. Available: http://gogoboard.stanford.edu/. [Accessed 16 March 2013].

[20] Lego, "Lego Mindstorms," 2014. [Online]. Available: http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com.

[21] Wikipedia, "AI Challenge," 2014. [Online]. Available: http://en.wikipedia.org/wiki/AI_Challenge.

[22] ACM, "ACMQueue ICPC Challenge," 2014. [Online]. Available: http://queue.acm.org/icpc/.

[23] Proyecto UTN1707, "Robocode UTN," 2014. [Online]. Available: https://www.facebook.com/robocodeutn.

[24] F. Sadosky, "Desafío Dale Aceptar [Official Website]," 2014. [Online]. Available: http://www.daleaceptar.gob.ar/cms/.

[25] F. Silva, L. Tanure and V. Oliveira, "Br-GoGo Project [official website]," 2009. [Online]. Available: http://br-gogo.sourceforge.net/. [Accessed 16 March 2013].

[26] Universidad Blas Pascal, "Proyectos de Investigación - Ingeniería Informática," 2013. [Online]. Available: http://investigacion.ubp.edu.ar/sip/?sec=fichaTecnica&fac=Ing.%20Inform%E1tica&titulo=Ing.%20Inform%E1tica.

[27] Universidad Blas Pascal, "Proyecto GoGo UBP [Official Website]," 1 August 2012. [Online]. Available: http://www.ubp.edu.ar/ciade-it/proyect-gogo/. [Accessed 16 March 2013].

[28] V. Frittelli, C. Osimani, G. Funes, W. Geremía, D. Reyes, M. Aranda, M. J. Blanco and N. Oliver, "Elementos de Robótica: Transferencia en Tecnología Educativa al Nivel Medio," in Workshop de Investigadores en Ciencias de la Computacion (WICC 2012), Posadas, 2012.

[29] Universidad Blas Pascal, "Olimpiadas Nacionales de Informática, Electrónica y Telecomunicaciones 2013," CIADE-IT UBP, 1 August 2013. [Online]. Available: http://www.ubp.edu.ar/onet/onet2012/. [Accessed 2014].

El Mundo Virtual de Matemática Discreta de la Facultad Regional Tucumán de la UTN

Rosana Hadad Salomón, Gladys Mónica Romano, Julia Beatriz Marisel Bedrán, Elizabeth María Alexandra Dufour, Mario Alberto Paredi

Universidad Tecnológica Nacional, Facultad Regional Tucumán, Tucumán, Argentina

Abstract

Este trabajo pretende compartir la experiencia del uso de una Plataforma Virtual de Aprendizaje 3D que se llevará a cabo en la cátedra de Matemática Discreta, materia de 1º año de la carrera de Ing. en Sistemas de Información, perteneciente a la Facultad Regional Tucumán de la Universidad Tecnológica Nacional. Dicha plataforma implementa el llamado V-Learning y viene acompañada de un Mundo Virtual que es una aplicación en 3D semejante a nuestra realidad.

Se trata del uso de la Plataforma Moodle unida a un módulo llamado Sloodle y a Opensim. Para ello, se creó un espacio 3D llamado "Isla", en el cual se aplicaron algunas de las actividades de Moodle para hacer dinámicas las clases y experimentar un cambio en la manera de interactuar entre estudiantes y docentes a través de un representación virtual llamado Avatar.

Todo esto basado en la hipótesis de que el uso de las TIC como así también de las Tecnologías Educativas posibilitan al estudiante aprender de una manera amena y divertida semejante a las herramientas que este utiliza para su entretenimiento e interacción en la Web.

Introducción

Un mundo virtual o metaverso es la representación 3D de la realidad en la cual un usuario experimenta la sensación de estar en la vida real. El concepto de mundos virtuales se asocia al término de realidad virtual que nació en los años 70 como medio de práctica para el departamento de aviación de los EE UU, el cual consiste en el uso de equipos que simulan la inmersión de la persona en una situación determinada. Si sumamos a la realidad virtual la inteligencia artificial, simulación, software e internet podemos crear un entorno virtual diseñado para uso educativo que nos permitirá enseñar, transmitir y compartir conocimientos y experiencias; brindando a los docentes la posibilidad de procesar la información e interactuar con sus estudiantes en un entorno 3D[1][2].

El mundo virtual más usado es SecondLife, que tiene miles de usuarios conectados en todo el mundo, el mismo es arancelado y en él se pueden realizar actividades tales como asistir a un recital de rock o hasta comprar un terreno a través de su moneda virtual denominada linden dólar.

Para conocer de qué se trata el mundo virtual debemos tener claros algunos conceptos como:

Avatar: personaje creado generalmente a semejanza de una persona.

Isla: porción de espacio en la web sobre el que se diseña el mundo virtual

Visor: software que permite visualizar el mundo virtual

Grid: conjunto de islas en un mundo virtual

Prim: forma geométrica de la cual puede nacer cualquier objeto que se quiera diseñar dentro del mundo virtual.

Mediante estos metaversos se pueden realizar diferentes actividades, tal como nos sucede en la vida cotidiana, los mismos se pueden dividir en [3]:

Educativo/Científico: es aquí donde se aplica el denominado V- Learning el cual expande las actividades que se pueden realizar en el proceso de aprendizaje creando un espacio con un sinfín de posibilidades en ambientes diferentes y con experiencias diversas. Precisamente esta es el área que abordaremos en este trabajo.

Entretenimiento/Ocio: su aplicación más conocida es la de los videojuegos.

Comercial: las aplicaciones están destinadas a empresas cuya necesidad es publicitar sus productos.

Desarrollo Sociocultural: a través de los Mundos Virtuales se realizan todo tipo de actividades socio culturales tales como visitas a museos, recitales musicales y actividades turísticas en general.

Es importante destacar que los mundos virtuales son dinámicos a diferencia de un videojuego en 3D, no se persiguen objetivos ni se guardan partidas, aquí simplemente la vida transcurre, por lo cual si nos desconectamos de él, puede que al volver a ingresar éste haya cambiado.

La creación de un mundo virtual es libre, se puede diseñar los objetos y crear las leyes de convivencia que se consideren apropiadas según los objetivos que se persigan. Cada persona puede crear

su propio mundo virtual, diseñando sus objetos y aplicando las leyes de convivencia que desee.

Nuestro mundo virtual académico está en plena etapa de creación y en este trabajo mostraremos las primeras experiencias de aprendizajes realizadas en este tipo de ambiente permitiendo la libre interacción entre docentes y alumnos.

Experiencia en Matemática Discreta

Para desarrollar esta prueba, se realizó la Instalación y puesta en marcha de un servidor en las instalaciones del "Área de Educación a Distancia, Tecnologías Educativas e Innovación" de la universidad a la que pertenecemos, que contiene los siguientes software libres[4]:

Opensim: Simulador 3D libre y de código abierto que utiliza los mismos estándares que SecondLife (SL) para comunicarse con sus usuarios. Esto hace posible que los usuarios puedan utilizar el mismo software que provee Linden Labs (empresa que creó SecondLife) para conectarse a un servidor controlado por organizaciones o individuos sin relación con esta empresa [5].

Sloodle: Proyecto Open Source (de código abierto) cuyo objetivo es unir las funciones de un sistema de enseñanza basado en web (LMS del inglés Learning Management System o VLE de Virtual Learning Environment) con la riqueza de interacción de un entorno virtual multiusuario 3D (MUVE de inglés MultiUser Virtual Environment). Actualmente todo el desarrollo de Sloodle se basa en la integración entre Moodle y SecondLife. De igual manera se puede lograr una combinación de Moodle y OpenSim que es el objetivo de este trabajo [6].

Moodle: Aplicación que pertenece al grupo de los Gestores de Contenidos Educativos (LMS, Learning Management Systems), también conocidos como Entornos de Aprendizaje Virtuales (VLE, Virtual Learning Managements), un subgrupo de los Gestores de Contenidos (CMS, Content Management Systems). Es una aplicación para crear y gestionar plataformas educativas, es decir, espacios donde un centro educativo, institución o empresa, gestiona recursos educativos proporcionados por docentes y organiza el acceso a esos recursos para los estudiantes [7].

Se utiliza la base de datos MySQL en la cual se graban todos los datos que recogerá la experiencia de educar a través de los mundos virtuales.

Para comenzar a experimentar en el mundo virtual debemos acceder a un visor, en este caso utilizamos uno denominado "Singularity" pero existen una diversa cantidad de ellos, los cuales se implementan mediante pasos sencillos de instalación.

Esta experiencia se basa en la implementación de un mundo virtual para la asignatura Matemática Discreta que contiene actividades de Sloodle, las cuales se representan o asocian con las actividades de Moodle de la manera que se representa en la Fig.1:

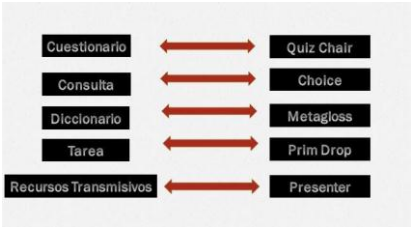


Fig.1. Comparación entre actividades Moodle y Sloodle.

Esto permitirá a los alumnos participar en un mundo 3D en el cual se sentirán cómodos y podrán crear y diseñar sus propios objetos, interactuar con todos los participantes del Mundo Virtual, organizando allí una segunda vida académica y social. En la Fig. 2 observamos avatares y objetos que integran nuestro mundo virtual.



Fig.2. Avatares y objetos en el mundo virtual.

El mundo virtual de Matemática Discreta permitirá ver y descargar el material teórico y práctico con el que se imparten las clases presenciales, hacer actividades de autoevaluación, realizar consultas a los docentes, interactuar con sus pares de un modo agradable compartiendo sus conocimientos, interrogantes, experiencias y demás actividades que estimulen la participación y la dedicación al estudio e investigación. En la imagen de la Fig. 3 se observa parte del equipo en plena actividad. En la Fig. 4 observamos algunos de los avatares de los miembros del equipo.



Fig.3. Equipo de trabajo.



Fig.4. Avatares del equipo de trabajo.

La implementación de la experiencia se realizará durante el mes de Octubre de 2014, con dos comisiones, A y B, las cuales comparten los mismos profesores de teoría y práctica. Debe considerarse que los alumnos ya han rendido el 1º parcial y fueron evaluados en los temas de Lógica Proposicional, Lógica de Predicados e Inducción. En agosto se incursionó en los temas de Teoría de Conjuntos, Métodos de Conteo y Combinatoria, Sucesiones y División en los Enteros. En setiembre se comenzó con el tema de Relaciones Binarias, el cual será el tema con el que se hará la experiencia en el Mundo Virtual.

La elección de este tema fue basada en el hecho de que a esta altura del cursado de la materia al alumno de 1º año se le ha formado y reforzado en los conceptos elementales que son la base de la asignatura y que a la vez fueron dados de una manera muy básica en el ciclo de la enseñanza secundaria.

En la comisión A se seguirá con la misma metodología de enseñanza, la aplicada hace más de 4 años y que consiste en: La clase presencial en aula común una vez por semana de carácter teórico-práctica. Participación obligatoria en el Aula Virtual, donde encuentran el material teórico y práctico, los cuales son guías con las que todos los docentes imparten sus clases, ejercicios de autoevaluación diseñados con las herramientas de la plataforma Moodle. Además de disponer para las comunicaciones entre docentes y alumnos de los foros novedades y de consultas, y la mensajería asincrónica constante desde principio de año.

En la comisión B se hará la experiencia de "El Mundo Virtual de Matemática Discreta" que consistirá en: La clase presencial en Laboratorio, una vez por semana de carácter teórico-práctica, con los alumnos participando activamente en el Mundo Virtual, donde dispondrán del material teórico y práctico a través de Sloodle Presenter, Metagloss y QuizChair. Luego, desde sus hogares contarán con la asistencia virtual del docente - tutor quién estimulará el encuentro virtual a través del Chat. Podrán resolver cuestionarios de carácter prácticos que permiten la autoevaluación de su proceso de aprendizaje. También dispondrán de otros objetos de Sloodle que estarán a modo de prueba para corroborar su funcionamiento debido a que es una versión nueva.

Participación obligatoria en el Aula Virtual, la cual se lleva a cabo desde principio de año y donde cuentan con todo el material y los foros de participación como todos los alumnos.

Al finalizar el dictado del tema, se harán evaluaciones del proceso de enseñanza en ambas comisiones que consistirá en:

- Control de lectura de los aspectos teóricos del tema.
- Pruebas de habilidad para resolver ejercicios del mismo tipo de los impartidos en las clases prácticas.
- Comparación cuantitativa de las participaciones que han tenido ambas comisiones, una en el Aula Virtual y la otra en el Mundo Virtual, con respecto al rendimiento académico de cada grupo.

A la comisión A, la que recibió la enseñanza tradicional se hará una encuesta de opinión considerando los siguientes ítems:

- Agrado por las clases presenciales en el aula tradicional.
- Facilidad en el acceso al Aula Virtual.
- Accesibilidad al material didáctico.
- Grado de participación en los foros.
- Accesibilidad a las autoevaluaciones.
- Agrado por las autoevaluaciones.
- Grado de participación en las autoevaluaciones.
- Agrado por la estrategia de enseñanza empleada.
- Grado de interacción entre docentes y alumnos en el Aula Virtual.

A la comisión B, la que experimentó la experiencia de Mundo Virtual se le hará una encuesta de opinión considerando los siguientes ítems:

- Agrado por la estrategia de enseñanza a través del Mundo Virtual
- Facilidad en el acceso al Mundo Virtual
- Agrado por la construcción de su avatar
- Confort al interactuar con los docentes y pares en el Mundo Virtual
- Accesibilidad al material didáctico
- Accesibilidad a las autoevaluaciones
- Agrado por las autoevaluaciones
- Grado de participación en las autoevaluaciones
- Utilidad de las autoevaluaciones
- Grado de interacción entre los docentes-alumnos y entre pares en el Mundo Virtual

Resultados

Hasta ahora se implementó una prueba piloto con aproximadamente veinticinco estudiantes quienes se mostraron motivados por la creación de su avatar y quedaron entusiasmados por las herramientas que el Mundo Virtual puede ofrecerles. La experiencia definitiva será implementada en octubre de 2014, fecha en la cual podremos presentar los resultados finales de la experiencia.

Discusión

A través de esta experiencia se espera que:

- Se difunda e impulse el uso de Mundos Virtuales para la enseñanza de cualquier asignatura de las carreras de la UTN FRT con el propósito de mejorar los procesos pedagógicos.
- Se conformen nuevos proyectos donde participen docentes y alumnos para desarrollar nuevos objetos de aprendizaje utilizando las herramientas disponibles y que se anexen a los ya creados por Sloodle con el propósito de mejorar el Proceso de Enseñanza Aprendizaje en la asignatura.
- Se fomenten las capacidades de trabajo en equipo, automotivación, esfuerzo y responsabilidad de los alumnos virtuales.
- Se aprovechen las formas de aprender de los nativos digitales, es decir a través de imágenes, videos, mundos 3D, para mejor los rendimientos académicos. [8] [9]

Conclusión

En el Mundo Virtual de Matemática Discreta se pueden recrear diferentes aspectos del ambiente social y académico, incentivando de ese modo el interés en la asignatura y a la vez promover la creatividad y la interactividad en un mismo contexto de aprendizaje.

Aplicar la enseñanza a través de los mundos virtuales nos abre una puerta y nos asigna un gran reto en el cual tanto inmigrantes y nativos digitales aprenderán los unos de los otros, es un escenario en el cual la imaginación y la creatividad se ponen en juego todo el tiempo y nos motiva a innovar en el aula.

En los mundos virtuales está todo por descubrirse.

Referencias

[1]Que es la realidad Virtual (2005). http://www.realidadvirtual.com/que-es-la-realidad-virtual.htm. Accedido el 15 de Mayo de 2014.
[2]Ariel Paz e Silva. Realidad Virtual" (2007). http://www.monografias.com/trabajos53/realidad-virtual/realidad-virtual.shtml?.Accedido el 25 de Mayo de 2014.
[3]Lelen Trujillo. Clasificacion "MUNDO VIRTUAL" (2013). . http://mundovirtual9806.blogspot.com.ar/. Accedido el 20 de Mayo de 2014.
[4]Hadad Salomón, Dufour, Paredi (2013). http://www.42jaiio.org.ar/proceedings/simposios/Trabajos/JSL/06.pdf.Accedido el 25 de Mayo de 2014.
[5]Comunidad Open Simulator. What is OpenSimulator?.Opensimulator.http://opensimulator.org/wiki/Main_Page. Accedido el 20 de Abril de 2014.
[6]Comunidad Sloodle. AboutSloodle. Sloodle.http://www.sloodle.org/moodle/. Accedido el 20 de Abril de 2014.
[7]Comunidad Moodle. Bienvenido a la Comunidad Moodle. Moodle.http://moodle.org/. Accedido el 22 de Abril de 2014.
[8]Nativos e inmigrantes digitales, Marc Prensky (2011) EDICIONES SM
[9]Nativos e Inmigrantes Digitales, Alejandro Piscitelli (2009) Santillana

Datos de Contacto:

- *Rosana Hadad Salomón. UTN – FRT. Rivadavia 1050. S. M. de Tucumán, Tucumán. rosanahadad@yahoo.com.ar*
- *Gladys Mónica Romano. UTN – FRT. Rivadavia 1050. S. M. de Tucumán, Tucumán. gmromano.utn@gmail.com.*
- *Julia Beatriz Marisel Bedrán. UTN – FRT. Rivadavia 1050. S. M. de Tucumán, Tucumán., mbedran@gmail.com.*
- *María Alexandra Dufour. UTN – FRT. Rivadavia 1050. S. M. de Tucumán, Tucumán. aledufour@gmail.com.*
- *Mario Alfredo Paredi. UTN – FRT. Rivadavia 1050. S. M. de Tucumán, Tucumán. marioaparedi@gmail.com.*

Propuesta de Mejora para la Enseñanza de Herramientas Informáticas en la UNAJ

Guzmán, Jésica Vanesa, Montezanti, Diego Miguel y Morales, Daniel Martín
Universidad Nacional Arturo Jauretche, Instituto de Ingeniería y Agronomía

Abstract
El trabajo presenta las primeras experiencias, conjuntamente con una propuesta de mejora, de una asignatura presencial extracurricular de Informática, dictada desde el primer cuatrimestre del año 2013 en la Universidad Nacional Arturo Jauretche (UNAJ). La materia está dirigida a estudiantes de diferentes carreras incluso Ingeniería. La implementación de la asignatura surge como una necesidad frente a profundas limitaciones de los alumnos para la utilización de herramientas informáticas básicas. El número de estudiantes con un escaso dominio de las herramientas informáticas es considerable, en virtud de las dificultades de acceso al sistema educativo por parte de amplios sectores de la población y al perfil promedio de sus estudiantes (primera generación de universitarios, de bajos recursos socioeconómicos, con ocupaciones laborales y que finalizaron sus estudios secundarios hace varios años). La integración de herramientas informáticas a la vida de los estudiantes, cualquiera sea la carrera, es imprescindible para la adquisición de competencias básicas, las cuales fortalecerán su desarrollo profesional.

Palabras Claves
Educación - Herramientas informáticas – Primera generación de universitarios.

1. Introducción

Los primeros años de una carrera universitaria constituyen una etapa crucial en la vida de un estudiante. El salto de la escuela media a la universidad suele ser un proceso conflictivo. Históricamente, uno de los principales problemas que presenta el Sistema de Ecuación Superior en nuestro país es la alta tasa de deserción de estudiantes en los años iniciales de las carreras universitarias [1]. Entre las causas centrales se pueden mencionar:

- pobre formación en el nivel medio;
- situación socioeconómica de los estudiantes;
- carencia de hábitos de estudios y motivaciones para el logro de objetivos.

En general, existe la concepción de que los estudiantes que ingresan a la universidad satisfacen dos condiciones principales: a) son jóvenes que rondan los veinte años de edad, y b) cuentan con un capital cultural más elevado, en comparación con otros sectores de la sociedad. Sin embargo, este no es el patrón habitual para la Universidad Nacional Arturo Jauretche (UNAJ), una novel universidad pública argentina con sede en el partido de Florencio Varela. Desde su inicio de actividades, en marzo de 2011, la UNAJ ha contribuido sustancialmente al desarrollo del área de influencia directa de la universidad, una de las regiones con más carencias del conurbano bonaerense, altamente relegada en cuanto a la educación superior. Esto ha posibilitado sin duda alguna el acceso al sistema educativo a amplios sectores de la población, favoreciendo la inclusión social y educativa de sus habitantes. Actualmente cuenta con más de 10.000 estudiantes. Por lo tanto, en esta universidad se hace muy evidente el contraste con los dos puntos mencionados anteriormente, dado que se trata de estudiantes con un promedio de edad cercano a los 30 años, y que, en su gran mayoría, son primera generación de universitarios (hijos de padres no profesionales). La mayoría de los estudiantes trabajan además de estudiar, han finalizado sus estudios secundarios hace varios años y tienen un nivel socioeconómico relativamente bajo.

Este escenario genera la necesidad de considerar diferentes estrategias institucionales, académicas y pedagógico-didácticas para garantizar la inclusión y favorecer la permanencia de los estudiantes en la universidad. Un aspecto crucial consiste en la alfabetización digital de los estudiantes. La vertiginosa evolución de las aplicaciones informáticas que ha ocurrido en los últimos años ha incrementado la necesidad de capacitación en estas herramientas, cualquiera sea el campo de estudio. Un manejo eficiente resultará en un crecimiento de las posibilidades de inclusión en el sistema laboral de los estudiantes.

En base a indicadores previos que reflejan un pobre conocimiento de las herramientas informáticas básicas por parte de numerosos estudiantes de las diferentes carreras de la UNAJ, la universidad consideró conveniente la implementación de una asignatura denominada "Informática Extracurricular", para los años iniciales de todas las carreras. Entre sus objetivos se encuentran el de realizar un aporte en pos de:

- mejorar el rendimiento académico a través de vías de integración de las herramientas informáticas en los procesos de formación;
- incrementar las capacidades y competencias tecnológicas, incorporándolos a la sociedad de la información y el conocimiento.

Se busca de esta manera fomentar la alfabetización digital y combatir la brecha digital y cognitiva [2] y en consecuencia disminuir la deserción de los estudiantes en sus primeros años de la carrera. A su vez, se espera lograr que los egresados universitarios estén capacitados para aplicar los conocimientos adquiridos en cualquier ámbito en el que deban desempeñarse.

2. Herramientas Informáticas en el Siglo XXI

Las herramientas informáticas, a través de un conjunto de medios electrónicos, permiten manipular información y ofrecer programas necesarios para la adquisición, producción, almacenamiento, comunicación, registro, presentación y transmisión de datos por medio de diversas señales, ya sean acústicas, ópticas o electromagnéticas [3].

El surgimiento de estas herramientas ha revolucionado sin duda la vida de las personas en las últimas tres décadas. En primer lugar, el acceso de la gran mayoría de la población a las computadoras personales (con microprocesadores y dispositivos de memoria digital cada vez más potentes), y posteriormente, el surgimiento de Internet, sumado a la continua mejora en la capacidad de transmisión de información, a través de la fibra óptica y de sistemas inalámbricos, han sido factores fundamentales que han posibilitado la reducción de costos relacionados con el manejo, el almacenamiento y la transmisión de la información.

Las herramientas informáticas están presentes en toda actividad, tanto comercial, industrial, manufacturera, de servicios, educativa o social. Hoy en día, a través del uso de Internet, diferentes dispositivos informáticos y de telecomunicaciones manejan datos en forma digital, que pueden ser compartidos, procesados y empleados en una amplia gama de actividades, en pos de tener una organización y gestión más eficientes, que optimicen los recursos existentes y mejoren la manera de brindar sus servicios.

Para poder manejar eficientemente toda la información disponible, es necesario desarrollar un conjunto de habilidades que permitan determinar los datos relevantes y la manera de obtenerlos y aprovecharlos. Estos aspectos requieren un adecuado dominio de las herramientas informáticas, de forma de lograr rapidez y reducir el esfuerzo.

El conocimiento apropiado del manejo de las herramientas informáticas permite reducir la desigualdad de oportunidades en la población (debido a los diferentes estratos socioeconómicos), preparando adecuadamente a las personas para la denominada sociedad del conocimiento, característica del siglo XXI [4].
Aquellas personas nacidas hace menos de 20 años han conocido desde temprana edad medios interactivos e innumerables recursos tecnológicos, los cuales les resultan familiares o naturales. Este tipo de personas se denominan "nativos tecnológicos" [5], dado que nacieron en la era digital o bien tuvieron un proceso de adaptación acelerado a la tecnología. También se los suele denominar "la generación multimedia" [6], por el uso simultáneo que ejercen de las variadas formas de la tecnología (es decir, navegan por Internet, escriben correos electrónicos y mensajes de telefonía celular, hablan por teléfono, buscan y almacenan información, juegan, escuchan y bajan música). De hecho, muchas veces, son quienes van descubriendo los nuevos usos de las diversas tecnologías. El hecho de que muchos jóvenes, con menores recursos económicos, no puedan adquirir los dispositivos más sofisticados, no implica que no formen parte de esta categoría. En contraste, las personas mayores de 30 años pueden considerarse como "inmigrantes digitales", dado que su lenguaje nativo no incluía a las herramientas informáticas, sencillamente porque este lenguaje aún no se había estructurado [7].
La formación profesional y las competencias requeridas para los nuevos profesionales han cambiado. El mundo actual requiere de profesionales que, además de contar con las competencias específicas de cada carrera, posean un nivel aceptable tanto en el manejo de idiomas, como en habilidades personales y sociales. Fundamentalmente, en la actualidad no resulta suficiente el manejo básico de una computadora, sino que se exige una alfabetización informática-computacional, la cual implica que se debe estar en condiciones de identificar, recolectar, procesar, manejar y gestionar datos provenientes de cualquier tarea o procedimiento inherente a la actividad que desarrollen, ya sea funciones asistenciales, docentes, administrativas, de investigación, de divulgación, de transferencia, etc.
En este sentido, es necesario que el pasaje por la educación superior, sea un camino para que los estudiantes puedan apropiarse de determinados instrumentos, tanto intelectuales como tecnológicos, que los ayuden a enfrentar el mundo actual con las exigencias que conlleva. El conjunto de la población que queda a mitad de camino en el sistema educativo, sin lugar a dudas, tendrá mayores dificultades para poder enfrentar la complejidad del mundo actual, que implican altas probabilidades de quedar marginado, en diferentes grados, de la vida social, política o económica según el nivel de educación formal alcanzado. Por lo tanto, las universidades deben adaptarse a las necesidades de la sociedad actual. El acceso a las herramientas informáticas, favorece la inclusión social y educativa de sus estudiantes y mejora la calidad de vida de la comunidad.

3. Marco Socioeconómico de la Región

La UNAJ se localiza en el Partido de Florencio Varela y tiene como área de influencia directa a los partidos de Almirante Brown y Berazategui. El contexto socioeconómico de la región en la cual se encuadra la universidad incide directamente sobre el perfil de los estudiantes.
En particular, Florencio Varela es uno de los partidos con más carencias del conurbano bonaerense, por lo cual la UNAJ se ha convertido desde sus inicios en un entorno institucional favorable para el desarrollo de la región, a través de la promoción de la educación destinada a los sectores más vulnerables y de la reducción de las disparidades en el acceso a ella, ante el complejo marco social que la comprende. El crecimiento de la matrícula de estudiantes año a

año es reflejo de las demandas de la población, que muestra el considerable impacto que tiene la universidad sobre el desarrollo local y la promoción social en la zona. En este sentido, la retención y motivación de los estudiantes en la universidad constituye todo un desafío.

4. Implementación de la Asignatura "Informática Extracurricular"

4.1. Motivación

Un gran porcentaje de estudiantes de las diferentes carreras de la UNAJ presenta importantes dificultades para utilizar las herramientas informáticas básicas, tales como examinar el correo electrónico, almacenar o buscar un archivo en una carpeta, manejar un dispositivo de memoria externa, etc. Estas falencias son claros indicadores de la necesidad imperiosa de proponer acciones que hagan frente a tal situación. Como una medida para remediar este problema, desde el Centro de Política Educativa, en colaboración con la coordinación de la carrera de Ingeniería en Informática del Instituto de Ingeniería y Agronomía, se dispuso la implementación de una asignatura, denominada "Informática Extracurricular" con el objetivo de brindar conocimientos básicos de informática, destinada a estudiantes de los años iniciales de la universidad. Esta asignatura es de carácter obligatorio y presencial, con una carga horaria de tres horas semanales durante un cuatrimestre. Durante el curso, los estudiantes abordan los conocimientos fundamentales vinculados con el manejo del sistema operativo (manipular carpetas y archivos, agregar nuevo hardware e instalar nuevo software, personalizar el escritorio, etc.), y el uso de utilitarios básicos como procesadores de texto, planillas de cálculo, editores gráficos y presentaciones multimedia. Además, se presentan temas tales como la navegación y búsqueda en Internet, la utilización del correo electrónico y las redes sociales.

4.2. Desarrollo de la Asignatura

Las primeras experiencias de la asignatura "Informática Extracurricular", se llevaron a cabo durante el primero y el segundo cuatrimestre del año 2013 y el primer cuatrimestre del año 2014. El dictado de la asignatura se lleva a cabo en seis aulas informatizadas, cada una de las cuales cuenta con 27 PC´s con MS Windows 7 y paquete de ofimática MS Office 2010, puertos USB y conexión a Internet. Los temas teóricos son exhibidos por medio de presentaciones de diapositivas. Mientras que las actividades prácticas consisten en preguntas y ejercicios para resolver a partir de los contenidos desarrollados en clase. Además, se insta a los estudiantes a la realización de trabajos prácticos utilizando la computadora, a fin de fijar los conocimientos teóricos y aplicarlos a una situación concreta, afianzando al estudiante en el uso de las distintas herramientas presentadas durante el desarrollo de las clases. Durante las actividades prácticas se busca que los estudiantes mantengan un rol protagónico y activo, con la finalidad de que desarrollen un aprendizaje profundo. En tanto, el rol del docente durante la actividad es el de tutor, ofreciendo ayuda y planteando preguntas que guíen a los estudiantes. La comunicación con los estudiantes se realiza por medio de una página web. Respecto de las evaluaciones, éstas constan de una parte teórica y una parte práctica. La parte teórica consiste en una variedad de tipos de preguntas, tales como relación de columnas, opción múltiple, respuesta corta y verdadero/falso, con el objetivo de evaluar contenidos conceptuales. La evaluación práctica en cambio, consiste en la realización de una serie de actividades a través de la utilización de la computadora.

4.3. Dificultades encontradas

1) Determinados contenidos resultan especialmente dificultosos para los estudiantes (como la utilización de gráficos y fórmulas en

las planillas de cálculo, por ejemplo). Sin embargo, habitualmente estas dificultades no se manifiestan durante la realización de las prácticas, ni los docentes reciben consultas por mail ni vía web en las semanas previas a las fechas de evaluación.

2) Uso de las computadoras y versiones de software: muchos estudiantes no tienen PC en sus hogares, o poseen una antigua. Dado que la asignatura contempla que el estudiante aprenda a manejar una PC con versiones actuales de software comercialmente disponibles (como se mencionó anteriormente, MS Windows 7 como Sistema Operativo y MS Office 2010 como paquete de ofimática), no pueden practicar en sus hogares aquellos aspectos en los que existen diferencias, y para los cuales no es suficiente el tiempo durante las clases.

3) Almacenamiento de la información: como se explicó anteriormente, la metodología de evaluación contempla el almacenamiento del trabajo práctico de los estudiantes en dispositivos extraíbles para su entrega a los docentes, formando parte este ítem de los aspectos evaluados. Sin embargo, muchos estudiantes presentan serias dificultades al momento de hacer esto; inclusive, algunos de ellos no han podido encontrar los archivos que ellos mismos han creado en sus PC´s y agruparlos en una carpeta.

4.4. Resultados

Desde que se dictó por primera vez la asignatura "Informática Extracurricular", a comienzos del año 2013, se ha tenido un número de inscriptos cercano a los mil estudiantes. Del total de estudiantes que cursaron la asignatura, el 48% la han promocionado, el 20% aprobó la cursada, mientras que en el 32% restante se encuentran quienes desaprobaron (rindieron una o ambas evaluaciones parciales pero sin llegar a la nota mínima) y los que quedaron libres por superar la cantidad máxima de inasistencias permitidas (Figura 1).

Si bien el hecho de que un 70% de estudiantes posean una calificación satisfactoria, en base a la experiencia compartida por el plantel docente se considera que es posible mejorar la implementación de la asignatura de acuerdo a lo que se presenta en la siguiente sección.

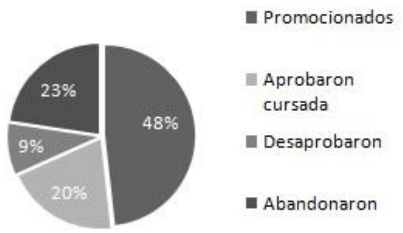


Figura 1. Resultados del rendimiento de los estudiantes en la asignatura "Informática Extracurricular".

5. Propuesta de Mejora para la Asignatura "Informática Extracurricular"

A continuación se presentan algunas propuestas tendientes a contribuir para la reducción de la brecha digital y sociocultural en los estudiantes de la UNAJ. Algunas de ellas se encuentran actualmente en etapa de implementación, mientras que las restantes son proyectos que están siendo evaluados.

5.1. Apertura de una Sala de PC Para Estudiantes

Con el propósito de disponer de mayor tiempo de práctica, se propone la apertura de un espacio informatizado para ser utilizado por los estudiantes fuera del horario de clases.

5.2. Incorporación de Material Audiovisual

Actualmente se está experimentando con la incorporación de material audiovisual producido por el plantel docente de la asignatura. Por medio del sitio web se comparten pequeños videos, tipo tutorial. En ellos, los docentes muestran, paso a paso, la resolución de operaciones sencillas (como personalizar el fondo de escritorio o instalar un programa o un dispositivo desde el Panel de Control). Estos ejemplos son del tipo de los que

pueden encontrarse en las prácticas o que pueden tomarse en las evaluaciones, y pueden ser aprovechados por aquellos estudiantes que tienen mayores dificultades.

5.3. Motivación desde el Ámbito Profesional

Existe la propuesta de incorporar a las prácticas diversos ejercicios sencillos que introduzcan a los problemas básicos propios de las diversas carreras que cursan los estudiantes que participan de la asignatura. De esta forma, los estudiantes ven un posible ámbito de aplicación de la informática ligada a su futura profesión, además de comprender la necesidad de utilización de herramientas tecnológicas. Cabe aclarar que algunas carreras cuentan con asignaturas de Informática Aplicada que son correlativas a Informática Extracurricular. De esta forma, la motivación inicial puede estar articulada con las mencionadas asignaturas correlativas, de modo de establecer una relación entre los contenidos que se estudian en las distintas materias.

5.4. Flexibilización de las Políticas de Administración

Como se mencionó en la sección 4.4, existe un conflicto de seguridad entre las políticas de administración del sistema y de la red en las aulas informatizadas, y ciertos conocimientos que se desea que los estudiantes incorporen. Debe establecerse una comunicación entre la coordinación de la asignatura y el Área de Sistemas para determinar qué contenidos, de los que deben asimilar los estudiantes, son compatibles con los requerimientos de seguridad, de forma de incorporar aquellos que no comprometan seriamente la integridad del sistema.

6. Conclusión

El trabajo presentó las primeras experiencias de la implementación de un curso presencial de la materia "Informática Extracurricular" en la UNAJ. El curso surgió como una necesidad frente a diferentes indicadores que revelaban una escasez de conocimientos y utilización en las herramientas informáticas básicas. El número de estudiantes en estas condiciones no es menor, en virtud de las características socio-económicas del área de influencia de la Universidad y a las dificultades de acceso al sistema educativo por parte de amplios sectores de la población. El curso contribuyó a la formación y capacitación de futuros profesionales. La integración de herramientas informáticas a la vida de los estudiantes, cualquiera sea su campo de estudio, les permitirá poder responder a las demandas que la sociedad reclama de ellos y poder ser incluidos dentro de los sistemas comerciales, industriales, de servicios, educativos o sociales de la región. En base a la experiencia de tres cuatrimestres, el trabajo presentó una propuesta de mejora del curso.

Referencias

[1] García de Fanelli, A. "Acceso, abandono y graduación en la educación superior argentina". *Sistema de Información de Tendencias Educativas en América latina*, 2005.

[2] Cabero Almenara, J. "Las TIC y las universidades: retos, posibilidades y preocupaciones". *Revista de la Educación Superior*. Vol. XXXIV (3), 2005, No. 135, pp. 77-100.

[3] Boude Figueredo Ó., Ruiz Quintero M. "Las TIC: propuesta para el aprendizaje de enfermería basado en problemas". Ed. Aquichan, pp. 227–42 (2008).

[4] Fandiño Parra, Y. "La educación universitaria en el siglo XXI: de la sociedad de la información a la sociedad del conocimiento". *Revista Iberoamericana de Educación*, Nº 55/3, pp. 1-10 (2011).

[5] Cabero Almenara, J. "Los Retos de la Integración de las TICS en los Procesos Educativos. Límites y Posibilidades". *En: Perspectiva Educacional*. Vol. 49, pp 32-61 (2010).

[6] Morduchowicz, R. *La Generación Multimedia. Significados, consumos y prácticas culturales de los jóvenes*. Buenos Aires, Paidós. (2008).

[7] Prensky, M. "Nativos e Inmigrantes Digitales". *En: On the Horizon*. Vol. 9 No. 6. MCB University Press (2001).

Entornos virtuales y herramientas de software libre como apoyo a la enseñanza de programación

Aguirre, Jesús Francisco[1], Viano, Hugo josé[1], García, Berta[1]
[1] *Departamento de Informática*
Fac. de Cs. Física, matemáticas y naturales
Universidad Nacional de San Luis
San Luis, Argentina
{jaguirre@unsl.edu.ar, hviano@gmail.com, bertae.garcia@gmail.com}

Abstract
La experiencia presentada en este artículo se desarrolla en el contexto de la asignatura Programación, de la carrera Ingeniería Electrónica con Orientación en Sistemas Digitales y surge como respuesta a las demandas provenientes de la realidad socio económica actual y su impacto en el ámbito educativo. La idea principal se articula en torno a la incorporación de un modelo de trabajo basado en la utilización de aplicaciones de software libre, con el soporte de un aula en el Entorno para el Aprendizaje Virtual "Aulas virtuales", implementado en Moodle y perteneciente a la UNSL. El uso de estos recursos y estrategias favoreció los procesos de enseñanza y aprendizaje colaborativos e incrementó las posibilidades de seguimiento y tutorización de los alumnos por parte del equipo de cátedra. Además, facilitó la comunicación, mejoró la participación y permitió continuar aprendiendo más allá de los límites del aula presencial.

Palabras Clave
Entornos de aprendizaje virtual, software libre, entornos de desarrollo integrado, trabajo colaborativo.

Introducción

El impacto producido en nuestra sociedad por la constante evolución de las computadoras, permitió el desarrollo de nuevas Tecnologías de la Información y las Comunicaciones llamadas comúnmente TIC. Este desarrollo impactó tanto en el sector público como en el sector privado.
Por otra parte, la filosofía del Software Libre (SL) plantea un nuevo modelo de trabajo sobre las libertades de los usuarios y la propiedad intelectual del software desarrollado. En la actualidad ha demostrado tener viabilidad tanto técnica como económicamente y las estadísticas reflejan un crecimiento sostenido de uso de software con estas características.
En la misma línea de pensamiento, el uso de Entornos de Aprendizaje virtual (EAV) implementados sobre plataformas de SL como Moodle **[1]** por ejemplo, acrecentó la posibilidad de alumnos y docentes de comprender la importancia de la utilización de SL como metodología de trabajo, sobre todo en el ámbito de las UUNN.
La inclusión del entorno "Aulas virtuales" implementado sobre la plataforma Moodle, responde a una necesidad de incrementar la participación de los alumnos que por razones principalmente laborales plantean conflictos horarios lo cual deriva en escasa presencia en clase y, por consiguiente, poca participación en la asignatura. Asimismo, la incorporación de este entorno, aporta un espacio extra de participación que ofrece un gran soporte al proceso de enseñanza-aprendizaje presencial.
Este trabajo presenta una experiencia en el marco de la enseñanza de la programación, mediante el uso de herramientas de SL. El mismo se organiza de la siguiente manera: el primer apartado sintetiza el marco teórico dentro del cual se desarrolla el trabajo; el segundo describe el contexto de trabajo; luego se mencionan algunas herramientas de programación para el desarrollo de software, y posteriormente se describe la metodología utilizada. Finalmente se exponen las conclusiones y las futuras líneas de trabajo.

Marco teórico

La aparición de nuevas formas de organización económica, social, política y cultural trae consigo nuevas maneras de trabajar, comunicarnos, aprender, pensar y

vivir. A este nuevo orden social se lo llama Sociedad de la Información **[2]**.

Las Tecnologías de la Información y las Comunicaciones (TIC) ofrecen la posibilidad de reconstruir y reinterpretar las posibilidades de enseñar, de acuerdo con el marco socio-educativo-cultural de referencia.

El desafío para la educación superior consiste en encontrar modelos más abiertos y flexibles, donde el estudiante pueda hacerse responsable de la construcción de su propio aprendizaje. Para ello la enseñanza debe adoptar metodologías más activas.

En este sentido, el uso de entornos virtuales amplía las posibilidades de aprendizaje personalizado y seguimiento de los alumnos por parte de los equipos de cátedra.

Diseñar un espacio virtual como apoyo a la clase presencial promueve la construcción de conocimiento manteniendo cierta independencia del espacio y el tiempo pedagógicos. Este factor es muy deseable considerando la realidad actual de alumnos que en su mayoría trabajan, y por tanto tienen dificultades para asistir en los horarios habituales de las clases tradicionales. En estas propuestas el docente asume un rol de guía, centrado en el proceso más que en los resultados, organizando y secuenciando el material didáctico de acuerdo a las características de los estudiantes destinatarios, el contexto en el que se produce el proceso, la organización de la información y la tutorización.

Como docentes esperamos promover aprendizajes significativos en los estudiantes. Esto implica una participación más activa de quienes aprenden y da lugar al aprendizaje colaborativo.

El aprendizaje colaborativo es definido por Johnson **[3]** como: “el conjunto de métodos de instrucción o entrenamiento para su uso en grupos, así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social). En el aprendizaje colaborativo cada miembro es responsable de su propio aprendizaje, así como de los restantes miembros del grupo”. Busca favorecer espacios en los cuales se produzca el debate entre los estudiantes al momento de explorar conceptos que interesa interpretar o situaciones problemáticas que se desea resolver. La combinación de situaciones e interacciones sociales debería contribuir a aprender efectivamente tanto en forma personal como grupal. Es aconsejable reflexionar sobre las sesiones de colaboración una vez finalizadas, para detectar acciones útiles o no y reconocer qué es oportuno cambiar.

El aprendizaje colaborativo virtual es entendido como un proceso de construcción social de conocimiento. A partir del trabajo conjunto y las metas comunes, se da una "reciprocidad entre un conjunto de individuos que saben diferenciar y contrastar sus puntos de vista de tal manera que llegan a generar un proceso de construcción de conocimiento. Es un proceso en el que cada individuo aprende más de lo que aprendería por sí solo, fruto de la interacción de los integrantes del equipo" **[4]**. En el desarrollo de un grupo, por tanto, la interacción se convierte en un elemento clave, teniendo en cuenta que es el proceso esencial de juntar las contribuciones de los participantes en la co-creación de conocimiento **[5]**.

En síntesis, el aprendizaje colaborativo permite adquirir competencias para el trabajo en equipo y desarrollar habilidades requeridas en el ámbito profesional y laboral, habilidades deseables para los futuros ingenieros. Enriquecer puntos de vista desde perspectivas diferentes puede contribuir a una formación ciudadana basada en el respeto y la participación constructiva para la solución de problemas sociales de interés común.

Contexto del trabajo

La asignatura “Programación” forma parte del plan de estudios de la carrera Ingeniería Electrónica con orientación en Sistemas Digitales. Dicha asignatura, que se ubica en el segundo año de la Carrera, tiene como objetivo proporcionar al alumno herramientas que le permitan resolver problemas de cálculo numérico. Es la segunda asignatura con contenidos sobre

programación. Recupera e integra el estudio de los aspectos básicos del lenguaje de programación C, tales como el uso de variables, operadores lógicos y relacionales, instrucciones de repetición y condicional, se continúa con aspectos avanzados de C.
La asignatura "Programación" está estructurada en dos partes: la primera de ellas tiene como objetivo el aprendizaje del sistema operativo Linux: distribuciones, comandos, consola, interfaz gráfica y aplicaciones. La segunda parte se desarrolla durante el 70% del tiempo del cuatrimestre y tiene como objetivo profundizar el uso del lenguaje C para la resolución de problemas reales, trabajando con variables punteros, estructuras de datos y archivos, realizando la codificación, compilación y depuración de códigos bajo Linux.
El modelo de aprendizaje anteriormente utilizado fue la realización de actividades en forma presencial e individual, con la ayuda permanente del docente, en su rol de guía y facilitador para la incorporación y búsqueda de nuevos conocimientos.
La inclusión del Entorno de Aprendizaje Vitual (EAV) "*Aulas virtuales*" en la asignatura permitió un marco educativo más flexible como complemento a la enseñanza presencial, ofreciendo nuevas oportunidades, tanto a docentes como a alumnos, con respecto a los contenidos disciplinares. Entre ellas resulta posible mencionar: el seguimiento del aprendizaje, el debate grupal a través de foros y mensajería interna, la posibilidad de entrega de tareas en el entorno y sus correspondientes devoluciones.
Esta nueva forma de trabajo permitió al equipo de cátedra incorporar estrategias para desarrollar nuevas habilidades en los alumnos, tales como el trabajo en grupo y el aprendizaje colaborativo **[6]**

Herramientas de trabajo

Teniendo en cuenta las dificultades de los estudiantes para asistir regularmente a clase se decidió incorporar herramientas que permitan el aprendizaje ubicuo **[7]**. Dichas herramientas se describen en los párrafos siguientes.

Entornos de desarrollo integrado

Un Entorno de Desarrollo Integrado, llamado también IDE (sigla en inglés de "Integrated Development Environment") es un software que provee al usuario de un ambiente de trabajo con un conjunto de herramientas tales como: editor de código fuente, compilador y depurador posibilitando a los alumnos, la implementación de la solución a un problema planteado. La utilización de este tipo de software agiliza el aprendizaje ya que posee una interfaz común para todas sus herramientas.
La selección de una herramienta informática adecuada para que los alumnos desarrollen software no es tarea sencilla. Se deben tener en consideración factores como la complejidad de uso y también su potencia funcional para la resolución de los diferentes tipos de problemas.
Considerando la evolución en la calidad y funcionalidad del SL, que permite grandes ventajas con respecto al uso del software propietario y ofrece una filosofía de trabajo adecuada para el ámbito científico y dado que la asignatura tiene una orientación puramente práctica, se seleccionó CODE::BLOCKS para dar soporte al proceso de enseñanza de programación.
Esta herramienta permite editar, compilar, depurar y ejecutar un programa en los lenguaje C y C++ dentro del mismo ambiente, sin necesidad de utilizar aplicaciones adicionales. CODE::BLOCKS agilizado el trabajo de desarrollar software, evitando utilizar línea de comando para ejecutar los comandos gcc en compilación o gdb para depuración.
Una cuestión a tener en cuenta es que algunos Sistemas Operativos (SO), como Linux, proveen al momento de ser instalados de herramientas que resultan de utilidad para el programador, en cambio en otros SO, como Windows, resulta necesaria una instalación extra.

Sistema de Gestión de Aprendizaje

Un Sistema de Gestión de Aprendizaje (SGA) es una plataforma de software en línea instalada sobre un servidor web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial dentro de una institución u organización.
Dentro de las principales funciones de un SGA podemos enumerar: administrar usuarios y recursos como así también los materiales y actividades de formación, controlar y realizar un seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes estadísticos, gestionar servicios de comunicación como foros de discusión y videoconferencias, entre otros.
Un SGA provee herramientas tanto sincrónicas como asincrónicas las que posibilitan diferentes tipos de funciones **[8]**.

- Gestión de contenidos de aprendizaje: creación, almacenamiento y reutilización de recursos y contenidos de aprendizaje,
- Planificación del currículum: herramientas y capacidad de almacenamiento para planificar el currículum, personalización, diseño de actividades de enseñanza/aprendizaje, etc.
- Administración y participación de los estudiantes: acceso a la información, herramientas y recursos de los estudiantes, asistencia, horarios, portafolios electrónicos e información para la gestión,
- Herramientas y servicios de comunicación: correo electrónico y mensajería, foros de debates, blogs, etc.

Un SGA no incluye generalmente posibilidades de autoría (crear sus propios contenidos), sino que se focaliza en gestionar contenidos creados por fuentes diferentes y externas al SGA. La labor de creación de contenidos se desarrolla mediante el uso de un sistema más complejo denominado Learning Content Management System (LCMS).
Dentro de estos SGA, existen algunos en la categoría de software propietario y otros tantos bajo la filosofía de SL. Dentro de estos últimos, podemos mencionar algunos: Moodle, Atutor, Illias, Dokeos, Chamilo, etc.

Plataforma Moodle

Moodle es una plataforma de aprendizaje en línea, un SGA que se distribuye gratuitamente como SL (bajo licencia pública GNU).
Moodle nos permite administrar, distribuir y controlar actividades de formación a través de Internet. De una forma muy elemental podemos resaltar algunas de las funcionalidades de Moodle:

- Acceder de forma segura a contenidos formativos en línea,
- Gestionar usuarios, recursos así como materiales y actividades de formación,
- Evaluar y hacer seguimiento del proceso de aprendizaje,
- Permitir la colaboración y comunicación entre docentes y alumnos,
- Generar informes, recoger calificaciones y registrar los accesos y las actividades realizadas por los alumnos durante el proceso de formación en línea.

Moodle es una aplicación web a la que se accede por medio de un navegador con cualquier dispositivo conectado a Internet: computadora, celular, tablets, etc.
En la actualidad, Moodle es una de las aplicaciones web de SL más utilizadas en la Educación Virtual ya que es un sistema de gestión de cursos que ayuda a los docentes a crear comunidades de aprendizaje virtuales y en línea

Aulas virtuales

A partir del año 2008 el área de Profesorado del Departamento de informática pone a disposición de los docentes y alumnos de la UNSL el EAV “Aulas Virtuales” implementado sobre Moodle versión 2.3. Esto permite ofrecer una herramienta de apoyo tanto para alumnos que tienen dificultades para asistir a clases presenciales como para quienes desean fortalecer sus propios conocimientos sobre la asignatura.
En la actualidad, Aulas virtuales cuenta con más de 1500 usuarios entre alumnos y

profesores y más de 200 cursos de pregrado, grado y posgrado implementados en la plataforma y pertenecientes a las diferentes unidades académicas de la Universidad Nacional de San Luis.
El ingreso al entorno "*Aulas virtuales*" se reliza desde un navegador web, escribiendo en la barra de direcciones el URL del servidor (http://www.evirtual.unsl.edu.ar/moodle/) o bien puede accederse a través del sitio de la Universidad Nacional de San Luis siguiendo una serie de enlaces. Para poder comenzar a trabajar en la plataforma, se debe contar con un usuario y su correspondiente contraseña. La pantalla inicial de "*Aulas virtuales*" se muestra en la Figura 1.



Figura 1

Teniendo en cuenta el contexto socio económico actual, que impulsa a una considerable cantidad de alumnos a desempeñarse laboralmente además de estudiar, se presentan innumerables inconvenientes para la coordinación de horarios presenciales de clase. Estos problemas se acentúan al considerar que la mayoría de los alumnos cursan otras asignaturas, con sus propios programas de estudio y horarios de clase, que deben compatibilizar y coordinar con los de nuestra asignatura.

Metodología de Trabajo

La metodología de trabajo empleada en la asignatura comprende la realización de actividades con una programación semanal en el EAV que implica:

- lectura de materiales,
- comunicación con los tutores y demás alumnos que realizan el curso,
- realización de actividades colaborativas.
- visualización de videos tutoriales
- respuesta a cuestionarios

Cada semana los alumnos tienen disponible diferentes actividades individuales, tales como la realización de trabajos prácticos en forma presencial. Además, existen actividades planteadas en el EAV como la resolución de cuestionarios, respuestas a foros, entrega planificada de códigos solicitados
Con el objetivo de desarrollar habilidades de colaboración se realiza un trabajo práctico grupal de máquina. El mismo consiste en la implementación de la solución de un ejercicio real, codificado en lenguaje C bajo Linux. Dicha actividad está divida en dos partes a ser entregaaso en dos fechas diferentes. El soporte que ofrece el entorno virtual permite realizar un seguimiento y coordinar las acciones para superar las dificultades que pudieran surgir.
Es importante mencionar que los grupos de trabajo para el práctico de máquina son organizados por la cátedra, teniendo en cuenta la heterogeneidad del alumnado y con el objetivo de evitar el desgranamiento. Cada grupo cuenta con el soporte de un tutor que promueve la colaboración y el intercambio de conocimiento entre pares. Los tutores utilizan foros de discusión y mensajería interna para acompañar el proceso de enseñanza aprendizaje.

Resultados

La incorporación de esta nueva modalidad de trabajo posibilitó una mejor comunicación entre docentes y alumnos, sobre todo en aquellos casos en que existían dificultades para asistir a las clases presenciales de la asignatura.
Por otra parte, la realización de tareas en forma colaborativa entre los alumnos, permitió establecer un contacto más fluido al utilizar diferentes tipos de recursos tecnológicos del EAV, tales como los foros discusión, mensajería interna, cuestionarios, etc.

Existió una activa participación por parte de los alumnos. El uso de materiales de apoyo adicionales, en formato textual y audio visual, fortalecieron el proceso de enseñanza y aprendizaje de los contenidos de la asignatura. Finalmente, los alumnos presentaron el práctico de máquina resuelto en forma grupal y realizaron una defensa del mismo en una instancia presencial. Durante su desarrollo debieron interactuar y comunicarse entre ellos y con el tutor designado para lograr el objetivo final: entregar en la fecha estipulada, cumpliendo los requisitos establecidos.

Entre los criterios de evaluación de la asignatura se consideraron categorías como: participación, colaboración, interacción e integración.

La incorporación de un práctico de máquina usando herramientas de SL y el EAV "Aulas virtuales", ayudó a los alumnos a trabajar colaborativamente (en forma sincrónica o asincrónica) y discutir el desarrollo del código fuente en forma interna entre todos los integrantes del grupo, solicitando soporte técnico a los tutores de la cátedra cada vez que fue necesario.

La utilización de los informes de participación y actividad del EAV ayudaron al seguimiento las actividades grupales, permitiendo ofrecer oportunamente la ayuda necesaria.

Discusión

Teniendo en cuenta los beneficios obtenidos por la implementación de este modelo de trabajo, se piensa fortalecer el uso de las Aulas Virtuales, creando estrategias para incorporar herramientas desarrolladas recientemente, para edición, compilación y depuración dentro del mismo EAV.

Conclusión

Los resultados obtenidos mediante el uso del EAV "Aulas Virtuales" avalan la continuidad de esta línea de trabajo ya que posibilitó una mayor integración de alumnos con dificultades para asistir.

"Aulas Virtuales" permitió contribuir a una formación más flexible durante el proceso de enseñanza-aprendizaje con la realización de diversas actividades no presenciales. Por otra parte, este entorno posibilitó la realización de un seguimiento durante el aprendizaje por parte del docente y la obtención de informes estadísticos los cuales resultan de gran utilidad para la evaluación.

Al usar este modelo, el rol del docente sufrió modificaciones significativas en lo que respecta a la elaboración de materiales pedagógicos, seguimiento del proceso de aprendizaje y comunicación con los alumnos.

Referencias

[1] Sitio oficial de Moodle (www.moodle.org). Última visita: 24/08/2014.

[2] Coll, C. & Monereo, C. (2008) Educación y aprendizaje en el siglo XXI: Nuevas herramientas, nuevos escenarios, nuevas finalidades. En C. Coll y C. Monereo (Eds.) *Psicología de la Educación Virtual* (pp.19-53). Madrid: Ediciones Morata

[3]. Johnson, D.W. Johnson, R.T.,& Holubec, E.J. (1999). El aprendizaje cooperativo en el aula. Barcelona: Paidós

[4] (Guitert, 2000) Guitert, M.; Giménez, F. El trabajo cooperativo en entornos virtuales de aprendizaje. En: Duart, J.M.; Sangra, A. (Ed.) Aprender en la virtualidad (pp. 113 -134). Barcelona: Gedisa. (2000)

[5] (Gunawardena et al, 1997) Gunawardena, Ch., Lowe, C. & Anderson, T. Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. J. Educational Computing Research, vol. 17, núm. 4, pp. 395-429. (1997).

[6] Mondéjar, J., Mondéjar, J. A. & Vargas, M. (2006). Implantación de la metodología e ² learning en la docencia universitaria: una experiencia a través del proyecto Campus Virtual. Revista Latinoamericana de Tecnología Educativa, 5(1), 59 ² 71. Recuperado de http://campusvirtual.unex.es/cala/editi

[7] Cope Bill y Kalantzis, M. (2009) – Aprendizaje ubicuo – Grupo nodos ELE.

[8] Area, M.y Adell, J. (2009): ―eLearning: Enseñar y aprender en espacios virtuales. En J. De Pablos (Coord): Tecnología Educativa. La formación del profesorado en la era de Internet. Aljibe, Málaga, pags. 391-424.

Educación en Ingeniería

Acevedo Cecilia, Moralejo Raúl, Catena Franco
GRIDTICs (Grupo UTN de Investigación y Desarrollo en TICs) Facultad Regional Mendoza, Universidad Tecnológica Nacional, Mendoza, 5500, Argentina

Resumen

Hay una gran demanda de formación en TIC dirigida a los trabajadores, por lo que las instituciones educativas van incluyendo la alfabetización digital, para su gestión y como instrumento didáctico. Es necesario acompañar esta situación para no caer en un atraso tecnológico y asegurar a los ciudadanos las competencias que exige la Sociedad de la Información.

Palabras claves
Ciencia Tecnología y Sociedad (CTS), Casos Simulados, Comunidad de Investigación Solidaria.

1. Introducción

El ritmo acelerado y vertiginoso de la era digital en la que nos encontramos sumergidos, ha modificado nuestro ritmo de vida, nuestras relaciones, nuestras formas de comunicarnos, ha invadido nuestros espacios de trabajo, nuestros hogares y también nos brinda una nueva forma de educación. Una educación que tiene en cuenta las necesidades, intereses, ritmos y tiempos del educando, brindándole espacios de formación continua. Estos cambios sociales también se hacen presentes en nuestros espacios de enseñanza y aprendizaje, en nuestra formación, como necesidad imperante de vincularse y de ser parte de esta sociedad actual. La educación no puede permanecer ajena a estos cambios debe brindar espacios para ayudarnos a comprender, interpretar y aprender haciendo uso de las nuevas tecnologías de la información y comunicación. El uso de las actuales tecnologías en la educación muestra día tras día distintas formas de realizar tareas y plantea permanentemente diferentes modos de ver y pensar las cosas.

2. Tecnología como herramienta de aprendizaje

Pensar informáticamente supone operaciones mentales distintas y por lo tanto una propuesta pedagógica innovadora. No se puede pensar que el poder de la tecnología por sí solo va a conseguir que los viejos procesos funcionen mejor. Su uso debe servir para que las organizaciones sean capaces de romper los viejos moldes y creen nuevas formas de trabajo y funcionamiento. El planteo debe ser cómo usar las tecnologías para hacer las cosas que todavía no podemos hacer y no sólo cómo poder usarlas para mejorar aquellas que ya hacemos.

Las tecnologías de la información y la comunicación estarán cada vez más presentes en el trabajo de los educadores. Crearan nuevos lugares de formación y capacitación, como los espacios virtuales o aulas virtuales, brindando conocimientos y habilidades a amplios sectores de la población, atendiendo a los diferentes ritmos, intereses y tiempos de los usuarios.

Es necesario que todas las personas del siglo XXI conozcan y utilicen estas tecnologías, las valoren como un recurso para mejorar y enriquecer el proceso de enseñanza y aprendizaje, en busca de una sociedad especializada [1][2].

3. TICs Orientados a la educación

Las TIC han desempeñado un papel fundamental en la configuración de nuestra sociedad y nuestra cultura al servicio de la educación.

Pensemos en lo que han significado para la historia de la humanidad la escritura, la imprenta, el teléfono, la radio, el cine o la TV. Desde nuestros antepasados cazadores-recolectores que pintaban figuras en las

paredes de sus cuevas y abrigos hasta nuestros días; la tecnología ha transformado al ser humano, y lo ha hecho para bien y para mal.

Es así que la expansión generalizada de las TIC ha contribuido a modificar la vida de los países y la experiencia de las personas, ha surgido una nueva forma de organización basada en redes, generando transformaciones en los modos en que se organizan, piensan y se articulan los procesos educativos [1][4].

4. Metodología innovadora

La educación tecnológica, va de la mano con la Ciencia, Tecnología, y Sociedad, o comúnmente llamado CTS. El enfoque CTS es una técnica de enseñanza que aspira a que la alfabetización contribuya a motivar a los estudiantes en la búsqueda de información relevante e importante sobre las Ciencias y las Tecnologías de la vida moderna para que puedan obtener información de calidad, puedan analizarla y evaluarla, reflexionar sobre la información circundante, definir los valores implicados en ella y tomar decisiones al respecto.

Existen cinco aspectos que contemplan las unidades curriculares integradas en programas ya establecidos en Ciencia, Tecnología e ingeniería, ciencias sociales, o bien estructuradas como cursos independientes [6]:

Formación de actitudes de responsabilidad personal en relación con el ambiente natural y con la calidad de vida.

Toma de conciencia e investigación de temas CTS específicos, enfocados tanto en el contenido científico y tecnológico, como en los efectos de las distintas opciones tecnológicas sobre el bienestar de los individuos y el bien común.

Toma de decisiones con relación a estas opciones tomando en consideración factores científicos técnicos, éticos, económicos y políticos.

Acción individual y social responsable, encaminada a llevar a la práctica el proceso de estudio y toma de decisiones, generalmente en colaboración con grupos comunitarios (por ej. “talleres científicos, grupos ecologistas, etc.”).

Generalización a consideraciones más amplias de teoría y principios, incluyendo la naturaleza sistémica de la Tecnología y sus impactos sociales y ambientales, la formulación de políticas en las democracias tecnológicas modernas, y los principios éticos que puedan guiar el estilo de vida y las decisiones políticas sobre el desarrollo tecnológico.

La perspectiva CTS permite ir aún más allá del mero conocimiento académico de la Ciencia y la gran implicancia de la Tecnología, preocupándose por los problemas sociales relacionados con lo científico y lo tecnológico, favoreciendo la construcción de actitudes, valores y normas de conducta en relación con estas cuestiones y atendiendo a la formación humana para tomar decisiones con fundamento y poder actuar responsablemente [4].

5. Modelo y estrategia educativa

Si bien no es posible apuntar estrategias de enseñanza específicas del enfoque CTS, sí es cierto que las clases de ciencias con enfoque CTS exigen un repertorio de actividades más variado que lo que habitualmente se hace desde un enfoque tradicional de educación científica centrada en la transmisión de conocimientos. Por ejemplo, Membiela (2001) enumera entre las estrategias utilizadas en CTS las siguientes: trabajos en pequeño grupo, aprendizaje cooperativo, discusiones centradas en los estudiantes, resolución de problemas, simulaciones y juegos de rol, toma de decisiones, debates y controversias.

De acuerdo con los precedentes del movimiento educativo CTS, entendemos que además de aportar criterios sobre el “para qué” enseñar y el “qué” enseñar sobre Ciencia y Tecnología, se deben aportar pruebas de que los resultados de la instrucción dependen esencialmente del

"cómo enseñar" y que la aplicación de "estrategias educativas CTS" puede mejorar la calidad de la enseñanza de las Ciencias y las Tecnologías.

En consecuencia con lo anterior, consideramos que es necesario demostrar que es posible modificar las visiones deformadas sobre Ciencia, Tecnología y sus relaciones con la Sociedad, mediante cambios en el desarrollo curricular de las materias científicas y tecnológicas.

Los estudios CTS se han concentrado principalmente en tres campos: En el plano de la investigación, promoviendo una visión socialmente contextualizada de la ciencia y la tecnología. En el ámbito de las políticas públicas de ciencia y la tecnología, defendiendo la participación pública en la toma de decisiones en cuestiones de política y de gestión científico-tecnológica. En el plano educativo, tanto la educación secundaria como universitaria, contribuyendo con una nueva y más amplia percepción de la ciencia y la tecnología con el propósito de formar una ciudadanía alfabetizada científica y tecnológicamente [4][5].

6. Educación CTS

El objetivo de la educación en CTS, tanto en el ámbito educativo y de formación pública, es la alfabetización para propiciar la formación de amplios segmentos sociales de acuerdo con la nueva imagen de la Ciencia y la Tecnología que emerge al tener en cuenta su contexto social.

Los enfoques en CTS también aspiran a que la alfabetización contribuya a motivar a los estudiantes en la búsqueda de información relevante e importante sobre las Ciencias y las Tecnologías de la vida moderna en la perspectiva de que puedan analizarla y evaluarla, reflexionar sobre esta información definir los valores implicados en ella y tomar decisiones al respecto.

Hacer sensibles los ciudadanos a la cultura científica, mostrarles que la ciencia y la tecnología son construcciones humanas y que, por tanto, también reflejan los deseos, los intereses y los valores de los humanos. Dotarles de las herramientas conceptuales que les permitan comprender el mundo en el que viven y comprometerles con la idea de que las decisiones sobre el rumbo del conocimiento científico o la transformación tecnológica de la realidad no pueden delegarse en los expertos tecnocientíficos porque sus consecuencias afectan a todos. Éstos serían algunos de los objetivos relacionados con la primera de las finalidades descritas [4][5].

7. Metodologías didácticas con enfoque CTS

Las formas de participación pública en cuestiones tecnológicas no podrán resultar ni eficaces ni fundadas si no existe una educación de la tecnología que contribuya a la modificación de la percepción que tradicionalmente se tiene de la tecnología y la ciencia.

La nueva educación tecnológica, que la perspectiva CTS está propiciando, puede tomar muchas formas.

A continuación se exponen dos propuestas prácticas para el trabajo en el aula, La primera propuesta, el trabajo de "casos simulados", la segunda propuesta pretende impulsar un nuevo enfoque de la educación tecnológica propiciando la creación en el aula de una comunidad de investigación solidaria.

7.1. Casos Simulados

Se trata de aparentar una controversia sobre una cuestión que tenga relevancia social, en la que la clase se organiza en diferentes equipos a los que se asigna una posición coincidente con la de uno de los actores sociales que pudiera estar interviniendo en la discusión pública. Estos equipos tendrán que investigar, recopilar, organizar y construir información relevante para defender su punto de vista, primero en una exposición pública y después en un debate.

Los casos CTS educativamente relevantes

han de tener la suficiente apertura y flexibilidad como para propiciar el análisis de distintas alternativas posibles y hacer necesaria la participación de los ciudadanos en los procesos de decisión sobre ellas. ¿Por qué es conveniente simular la controversia?, ¿Por qué no investigar sobre una cuestión presente y totalmente real?, es muy importante introducir la realidad en las aulas, pero cuando se tratan cuestiones tecnocientíficas se corre el riesgo de ser arrollados por esa realidad. Para empezar, los temas que se podrían estar discutiendo no estarían cerrados en la controversia pública, y la cantidad de información sería inmanejable.

No resulta relevante la decisión final que se adopta en cada caso simulado como el nivel de debate público y el contraste racional de informaciones, argumentos y valores que ha podido tener lugar en el proceso.

Por lo tanto el modelo de aprendizaje es muy apropiados para su manejo en el aula y altamente motivador ya que, incluso, permiten un cierto juego creativo al relacionar lúdicamente los planos de la realidad y la ficción. Por otra parte, resultan especialmente adecuados para el aprendizaje de las capacidades necesarias para participar discrepando, negociando y consensuando las soluciones a los problemas que plantea a la sociedad el desarrollo tecnocientífico [3][5].

7.2. Comunidad de investigación solidaria

La idea de una comunidad de investigación solidaria es la apuesta por articular el aula como un espacio educativo en el que sea imprescindible la cooperación para el aprendizaje. Al definir la clase como una comunidad de investigación solidaria se pone el acento en los objetivos del trabajo en clase: investigar sobre ciertos temas, y hacerlo solidariamente, de la misma forma en que el saber se ha desarrollado en la historia: en comunidad.

Así, se han definido cuatro roles a cumplir, los cuales no son ajenos al modo de como los seres humanos se han ocupado de los asuntos que les han preocupado. La clase se dividirá en los siguientes equipos:

Investigación conceptual, Investigación empírica, Investigación creativa, Coordinación.

El trabajo de cada equipo termina con una exposición pública de lo trabajado y un informe por escrito. La idea es que el conocimiento no es unidireccional, sino que la clase se beneficia (y depende) del trabajo realizado por los demás. En el ámbito de la educación en valores, de la que procede esta metodología. Este carácter dinámico de las actividades en cada unidad garantizaría que no sólo motive a los alumnos por la novedad temática, sino que vaya aprendiendo e incorporando conocimiento relacionándolo con raciocinio con la sociedad circundante [3][5].

8. Inclusión en el currículo de propuesta CTS

Se trata de experiencias que rompen con la dependencia de un ámbito disciplinar determinado y tienen una vocación centrada en las cuestiones CTS.

8.1 Contenidos

Comprender la ciencia y la tecnología existente:

Presentación de los contenidos tecnológicos y científicos, con referencias a los contextos históricos y sociales en los que aparecieron. Análisis de la tecnología existente, y de los conocimientos científicos utilizados para diseñar los artefactos tecnológicos. Análisis de los aspectos valorativos de las teorías científicas y los artefactos tecnológicos. Interacción entre los saberes tecnológicos y sociales y creencias sobre la Tecnología y sus interacciones con la Sociedad y la Ciencia.

Participar en decisiones relacionadas con el desarrollo tecnocientífico:

Presentación de los impactos tras los desarrollos tecnológicos y avances científicos que se traten. Análisis de los

aspectos controvertidos sobre los valores inherentes a determinados conocimientos científicos y desarrollos tecnológicos. Identificación de los diferentes intereses y puntos de vista que pueden manifestarse ante determinadas decisiones relacionadas con el desarrollo tecnocientífico.

8.2. Metodologías

Comprender la ciencia y la tecnología existente:
Estudios de casos mediante proyectos de investigación en equipo. Lectura de documento y búsqueda de información referidos a distintos enfoques sobre tecnología. Elaboración y estudio sobre los contextos históricos y sociales en los que aparecieron determinados desarrollos tecnológicos y avances científicos. Exposiciones públicas sobre los trabajos realizados.
Participar en decisiones relacionadas con el desarrollo tecnocientífico:
Encuestas sobre percepción de determinadas cuestiones polémicas relacionadas con la ciencia y la tecnología. Participación en debates sobre controversias. Análisis sobre la información tecnocientífica adquirida, para elaborar propuestas sobre determinadas controversias tecnocientíficas.

8.3. Evaluación

Comprender la ciencia y la tecnología existente:
Sobre la originalidad de trabajaos realizados, la calidad de las exposiciones públicas de los trabajos realizados, la coherencia de proceso seguido y el interés de los resultado logrados.

Participar en decisiones relacionadas con el desarrollo tecnocientífico:
Sobre la capacidad para el intercambio de información, la disposición a comprender las posturas ajenas y a la disposición a la negociación y el consenso en ámbito de debate.

9. Experiencias innovadoras de enseñanza

Se efectuó el uso e implementación de una de las técnicas CTS llamada: Análisis de Casos Simulados.
Se trató de aparentar una controversia sobre una cuestión que tenga relevancia social, con implicancia tecnológica y científica, donde los hechos son inciertos, los valores están en disputa, lo que se pone en juego es alto y las decisiones son de actualidad.

9.1 Ámbito de aplicación

Durante el cursado de la materia Teleinformática, en la Universidad Tecnológica Nacional, Regional Mendoza, Carrera Ingeniería Electrónica, durante el primer cuatrimestre del 2013 y 2014, se puso en práctica el modelo y los estándares de evaluación y mejora correspondiente, para esto, se formaron equipos de trabajo a los que se asignó una posición coincidente con la de uno de los actores sociales que pudiera estar interviniendo en una discusión pública. En nuestro caso se definieron cuatro tipos de grupos y/o actores:
A Favor: aquellos que se ven favorecidos por la propuesta de la implantación tecnológica sugerida.
En Contra: aquellos cuyos intereses o valores se oponen a la propuesta de la implantación tecnológica sugerida.
Experto: aquellos que aportan asesoramiento en la evaluación de esa tecnología.
Coordinador: aquellos que cumplen una función de mediación en la controversia.

9.2 Fases de la experiencia

La puesta en práctica del modelo requirió llevar a cabo cuatro pasos, los cuales se detallan a continuación:
1) se dividió la clase en diferentes equipos cada uno de estos tuvo que investigar, recopilar, organizar y construir información relevante para defender su postura, haciendo uso de las herramientas tecnológicas necesarias para exponer su estudio frente al aula y luego poder confrontar un debate.
2) cada uno de los equipos elaboró

documentación de la controversia; aquí se buscó que cada uno de los grupos pudiera aportar los materiales básicos sobre los que posteriormente se debía debatir.
3) Tras días de trabajo de investigación, se efectuaron las exposiciones y defensas públicas de los mismos, simulando los argumentos que utilizarían los actores reales en una situación verídica.
4) Al final, se planteó un debate abierto entre todos los alumnos, en la que se buscó llegar a una solución consensuada o negociada. Se utilizó una plataforma virtual (libreduca.com), como soporte al dictado de las clases presenciales [7][8]

9.3 Resultado de la experiencia

Se buscó por medio del debate reflexionar entre todos sobre lo que habría sucedido si el caso se hubiera dado en la realidad y cómo puede mejorarse el nivel de participación pública en la decisión sobre un tema como el planteado, pero lo relevante de la experiencia es que se logró efectuar un análisis profundo y detallado sobre aspectos científicos y tecnológicos en relación con la sociedad, haciendo uso de la tecnología de información.
El método comprendió un procedimiento didáctico activo, dado que colocó al alumno en una situación problemática, para la cual tiene que hacer una o más sugerencias de solución, conforme a la naturaleza del problema planteado, es decir, se puso al educando ante una situación problemática o dudosa y se lo desafía a encontrar una solución satisfactoria para la misma. Lo que se buscó es poner énfasis en el razonamiento y la reflexión en busca de:
Desarrollar el raciocinio; sacándolo de la posición de receptividad de datos y soluciones; obligándolo a buscarlos.
Desarrollar aptitudes para el planteamiento, dado que el camino para llegar a la soluciones debe pensarse y estructurarse.
Desarrollar la iniciativa de búsqueda de soluciones, dado que el educando se coloca ante una situación problemática, a la cual tiene que hallar una salida.
Desarrollar el control emocional, dado que tendrá que esforzarse por trabajar con tranquilidad y eficiencia en la resolución del problema que se le proponga.
Hacer que el educando trabaje en base a hipótesis, cuya verificación exige el ejercicio de la reflexión, capacitándolo mejor para tomar decisiones, juzgar hechos y apreciar valores.
Provocar la motivación, debido a la satisfacción que produce la solución, el descubrimiento del conocimiento.
obtener una mejor fijación del aprendizaje.
infundir confianza en sí mismo.
desinhibir para incentivar la iniciativa.
promover el acercamiento entre la teoría y su aplicabilidad.
Desarrollar el espíritu crítico, el cual consiste en una combinación de aptitudes, competencias, actitudes y disposiciones utilizadas en el pensamiento de evaluación razonado y que requiere un juicio cuidadoso o una reflexión continua.

Referencias

[1] Pere Marqués Graells, “Impacto de las TIC en la educación: funciones y limitaciones”. Revista de investigación, 1999 (pdf).
[2] Ernesto Bravo, “Globalización, innovación tecnológica y pobreza”. Aproximación a las nuevas conceptualizaciones en Latinoamérica. Espacio Abierto Cuaderno Venezolano de Sociología, 2012 (pdf).
[3] Mariano Martín Gordillo y Juan Carlos González Galbarte, “Reflexiones sobre la educación tecnológica desde el enfoque CTS”, Revista Ibero Americana de Educación, 2002.
[4] Mariano Martín Gordillo “Cultura científica y participación ciudadana: Materiales para la educación CTS”, Revista Ibero Americana de Ciencia Tecnología y Sociedad, 2005.
[5]Educar para participar. Los casos simulados CTS, Diplomatura CTS+I –Innova
[6] Carlos Osorio, “La Educación Científica y Tecnológica desde el enfoque en Ciencia, Tecnología y Sociedad Aproximaciones y Experiencias para la Educación Secundaria”, Escuela de Ingeniería Industrial y estadística Universidad del Valle, Cali, Colombia.
[7] https://utn-frm.libreduca.com
[8] https://github.com/cirope/libreduca

Simulación Dinámica de Gestión de Tareas en Proyectos Desarrollados Con Scrum

Diego Alberto Godoy, Henry Kotynski, Edgardo Belloni, Eduardo O. Sosa,
Centro de Investigación en Tecnología de la Información y Comunicaciones
Departamento de Ingeniería y Ciencias de la Producción
Universidad Gastón Dachary
Av. López y Planes 6519- Posadas, Misiones, Argentina. Teléfono: +54-376-4438677
{ diegodoy hkotynski,ebellloni,eduardo.sosa}@citic.dachary.edu.ar

Resumen

En este trabajo se presenta el subsistema de gestión de tareas como parte de un "Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software Bajo Scrum". Se presenta un caso de validación y un caso de experimentación de situaciones reales en el cual se pueden ver situaciones de utilidad del modelo. Con la ayuda de esta herramienta los Scrum Masters y miembros del Team podrán observar los resultados de cada decisión tomada y elegir la mejor alternativa para ser aplicada en un proyecto real.

1. Introducción

La complejidad actual de las organizaciones y sus sistemas de software están sujetos a constantes cambios tanto tecnológicos como de requisitos. Esto genera un entorno donde la planificación, el desarrollo, la administración y el seguimiento del software resultan difíciles de estimar o evaluar.

Es por ello que este tipo de escenarios requiere de metodologías de desarrollo de software y gestión de proyectos que permitan generar resultados rápidamente y que los encargados de gestionar proyectos de software puedan utilizarlas de la mejor manera para mantener bajo control el proyecto. Una de las metodologías con este tipo de características es Scrum [1], la cual hace énfasis en las tareas de gestión de proyectos utilizando la planificación adaptativa y el desarrollo incremental del software con entregas funcionales en breves períodos de tiempo.

En escenarios y requerimientos con constantes cambios, sería interesante contar con una herramienta que permita simular proyectos de desarrollo de software gestionados con Scrum, que permita a los Scrum Master y el Team evaluar el impacto de sus decisiones sobre la gestión en el desarrollo de proyectos, sin influir o poner en riesgo el proyecto real y sus recursos. De la misma manera, tal herramienta de simulación ayuda a construir consenso sobre las decisiones a tomar de manera objetiva y no destructiva.

Si bien actualmente existen diferentes trabajos y herramientas que permiten simular la administración de proyectos de software, entre los cuales se pueden citar a los siguientes: "Dynamics of Agile Software Development" [2], "Modeling Agile Software Maintenance Process Using Analytical Theory of Project Investment" [3], "Modelo Dinámico de Simulación de Proyectos de Software con XP" [4] y[5], "Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics" [6],los mismos presentan modelos basados en diferentes metodologías ágiles, en los cuales son propuestos y evaluados diferentes escenarios, algunos de ellos genéricos. No obstante, en ninguno de tales trabajos se presenta un estudio que contemple las fases, variables y roles, particulares de la metodología Scrum para el desarrollo de software. En este caso se presentará el subsistema de gestión de gestión de tareas como parte de un "Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software Bajo Scrum". Cabe destacar que este trabajo se encuentra dentro de una línea de investigación presentada previamente en [7].

El presente trabajo se estructura de la siguiente manera: en la sección 2 se explicará brevemente la metodología de dinámica de sistemas. En la sección 3 se dará una descripción de la construcción del modelo. En la sección 4 se presentará un proyecto real utilizado como caso de validación del modelo y en la sección 5, se realizaran pruebas de experimentación con el modelo de situaciones problemáticas en los proyectos de software desarrollados siguiendo Scrum y se evaluaran dos posibles alternativas para atacarlos. Por último se presentaran las conclusiones y trabajos futuros.

2. Metodología de Dinámica de Sistemas.

El modelo se construyó basándose en la Metodología de Dinámica de Sistemas [8], la cual consta de tres fases.

1. Fase de Conceptualización, se han identificado las variables intervinientes en el modelo que representa el sistema de gestión de un proyecto de software desarrollado con Scrum, para luego construir el Diagrama de influencia que representa la estructura del sistema modelado.
2. Luego, en la segunda fase, denominada Fase de Formulación se construyó el Diagrama de Forrester el cual constituye una traducción de las variables y relaciones del Diagrama de influencias a ecuaciones matemáticas, posibles de ser programadas y simuladas. Este Diagrama de Forrester, que se compone de más de 200 variables, fue además dividido en doce Subsistemas Conservativos de acuerdo con [9] y un subsistema de variables auxiliares y constantes, para facilitar su estudio y análisis.
3. En última fase, la Fase de Evaluación, se han realizado corridas de validación y experimentales con el modelo, utilizando para ello casos reales y escenarios que representan situaciones frecuentes en proyectos gestionados con Scrum.

Para la construcción del modelo se ha utilizado el Software VenSim PLE 5.4c (Versión Académica) [10].

3. Construcción de Modelo.

Para la construcción del modelo se utilizado la metodología presentada en la sección 2.

Los Sub-sistemas en que se ha dividido el modelo son:

- Planificación
- Producción
- Desarrollo de Tareas
- Pruebas de Desarrollo
- Pruebas de Integración
- Presión en el Plazo
- Desarrollo de Tareas Extras
- Promociones de R.H.
- Experiencia de R.H.
- Cansancio de R.H.
- Horas Trabajadas de R.H.
- Inasistencias de R.H.
- Variables auxiliares y Constantes

A modo de ejemplo en este trabajo se presenta el diagrama de Forrester de Sub-sistema de Gestión de Tareas en la Figura 1 y se detallan las variables involucradas a continuación.

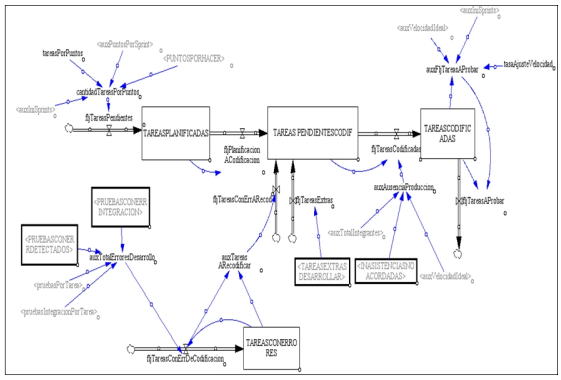


Figura 1 –Sub-sistema de Desarrollo de Tareas

3.1. Variables del Subsistema de Desarrollo de tareas

tareasPorPuntos: esta variable permite establecer la cantidad de tareas por puntos de historia que el Scrum Master estime o considere oportunas dada la complejidad de las tareas a desarrollar. Unidad: tareas

cantidadTareasPorPuntos: representa la cantidad total de tareas que se deben realizar para poder completar cada uno de los puntos de historia establecidos por Sprint. El valor de esta variable surge del producto entre el valor asignado a tareasPorPuntos y auxPuntosPorSprint. Dicho producto se efectuará solamente si PUNTOSPORHACER es mayor a cero. Unidad: tareas.

TAREASPLANIFICADAS: variable de nivel que representa la cantidad total de tareas a desarrollar en los diferentes Sprints. El valor de esta lo determina cantidadTareasPorPuntos. Unidad: tareas.

TAREASPENDIENTESCODIF: este nivel representa la cantidad de tareas pendientes de codificación en cada uno de los Sprints. El valor inicial está determinado por las tareas originales que se planificaron, y se ve modificado de acuerdo a la cantidad de Tareas con Errores que pudieran surgir y de las Tareas Extras. Unidad: tareas,

TAREASCODIFICADAS: almacena la cantidad de tareas que se fueron codificando a lo largo del desarrollo del Sprint. El desarrollo presenta la forma general de un diente de sierra, con puntos altos y descensos lineales, que se ven alterados por la ausencia no planificada de los integrantes del Team o por la aparición de Tareas no planificadas. Unidad: tareas.

auxAusenciaProduccion: esta variable auxiliar permite establecer en cuanto disminuirá la velocidad normal de producción diaria en función de la cantidad de ausencias no acordadas de los integrantes, de cuantos integrantes no asistieron al Sprint diario y de la velocidad de trabajo estimada inicialmente. Unidad: puntos.

tasaAjusteVelocidad: variable que permite ajustar la velocidad ideal establecida en el sub-sistema de Planificación. Ya que el número de tareas en relación a los puntos de historia no siempre es una tarea a un punto, se hace necesario modificar la velocidad de codificación para que los tiempos establecidos sean respetados. Unidad: tasa.

TAREASCONERRORES: variable de Nivel que acumula la totalidad de las tareas con errores sean de Integración o de Codificación. Su valor surge de la suma de la cantidad de tareas con errores de codificación dividido la cantidad de pruebas, más la cantidad de tareas con errores de integración divido la cantidad de pruebas de integración. Unidad: tareas.

auxTotalErroresDesarrollo: variable auxiliar donde se calcula el número de tareas con errores. Su valor se establece a partir de la suma de dos valores, ambos surgidos de dividir, el total de pruebas con errores de integración sobre el número de pruebas de integración, y del total de errores por codificación detectados sobre el número de pruebas realizadas. Unidad: errores

auxTareasARecodificar: variable auxiliar que permite determinar el número total de tareas con error existentes, generando picos que representan el número de tareas que surgen a lo largo del proyecto. Unidad: tareas.

4. Validación del Modelo

Para la validación se han utilizado datos de los siguientes proyectos de software reales gestionados con Scrum. Siendo estos los siguientes: Método ÁgilScrum Aplicado Al Desarrollo De Un Software De Trazabilidad[11], Aplicação Do Processo Ágil De GerenciamentoScrum No Desenvolvimento De UmJogo Digital[12], Case studyon agile estimating and planning using scrum [13] y Automatización de sistemas de desarrollo ágil Scrum: Team & Role[14].

Como ejemplo de validación en[14] se desarrolló en el marco de un convenio entre la Universidad Autónoma de Barcelona y la empresa UNIT4. El trabajo por parte de los programadores es a media jornada, una velocidad del 50% sería equiparable a una velocidad del 100% en un equipo a jornada completa. Para la validación se consideró que 1(uno) punto de historia es equivalente a 8(ocho) horas de trabajo. El detalle de las iteraciones se puede observar en la Tabla 1.

Tabla 1.Iteraciones del proyecto presentado en[14]

Sprint	Requerimientos	Puntos Historias	Duración en Horas	Velocidad Ideal Estimada
1	9	18	80	0,2688
2	8	28	80	0,35
3	12	31	80	0,3875
4	9	21	80	0,2625
5	9	24	80	0,3
6	7	25	80	0,3125
7	10	25	80	0,3125
Total	64	172	560	

De acuerdo con los datos presentados en la Tabla 1, en la Figura 2 se puede observar la planificación puntos a desarrollar en cada uno de los Sprints.

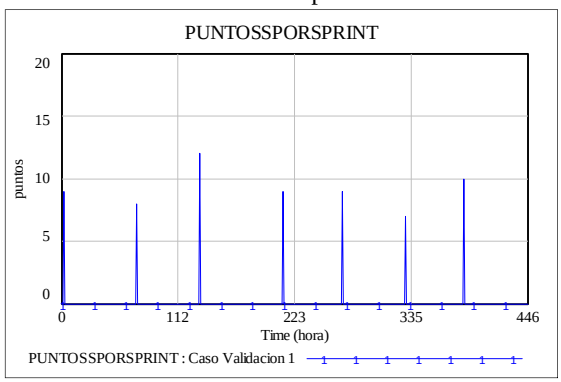


Figura 2 - Puntos Planificados Por Sprint

La velocidad de desarrollo de las tareas planificadas permitirá completar los puntos planificados para cada Sprint. En función de esta velocidad, los Sprints se completaran en tiempo y forma, o bien habrá que quitar puntos de historia para cumplir con el ProductOwner.

La cantidad de Sprints, los puntos a completar y la velocidad de trabajo de cada uno, son valores que el Scrum Master junto al Team pueden estimar previamente al inicio de cada simulación de un proyecto. En este caso de validación los Sprints son siete y la cantidad de puntos en cada uno se puede observar en la Tabla 1. En la Figura 3 se puede observar el gráfico que representa el desarrollo ideal de puntos de historia planificados, presentados en la variable BURDOWNCHART, y de los puntos de historia pendientes a completar en la variable PUNTOSPORHACER.

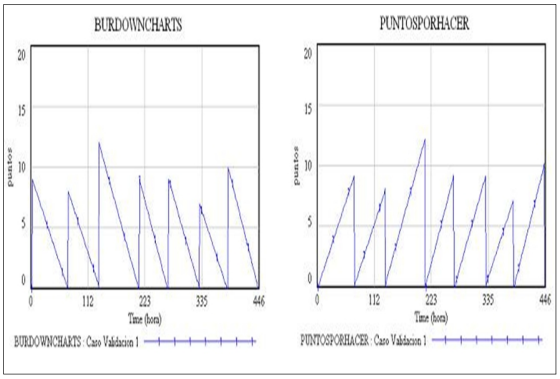


Figura 3–BurdownChart ideal y Puntos por hacer

De esta forma, y como puede verse en la Figura 3, la cantidad de puntos por hacer desciende de manera ideal ya que este grafico se genera en la planificación previa al inicio de cada Sprint. Por otro lado, la cantidad de puntos por hacer asciende de cero al total planificado para dicho Sprint.

Esta situación se repite para todos los Sprints, ya que no se considera ningún tipo de tarea extra, inasistencia o abandono en el Team durante el desarrollo del proyecto.

En la medida que las tareas se planifican, y dadas las condiciones mencionadas, dichas tareas se codifican sin inconvenientes y se completan en un 100% de acuerdo a lo planificado.

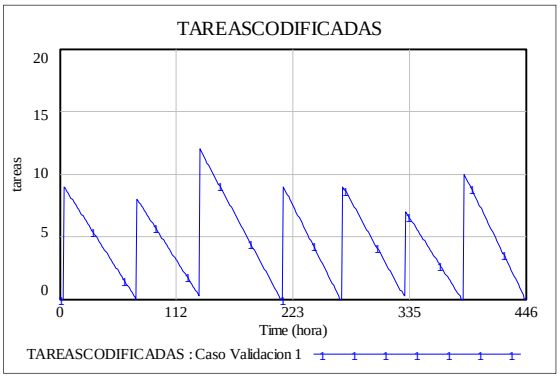


Figura 4 - Tareas Codificadas Por Sprint.

En la Figura 4 se puede ver también que tal como se ha planificado el comportamiento de la variable de nivel TAREASCODIFICADAS se comporta de manera similar al proyecto real validando él modelo.

5. Evaluación del Modelo

En este caso de evaluación del modelo se consideran situaciones donde ciertos parámetros considerados críticos toman valores diferentes a fin de probar el modelo ante situaciones extremas.

5.1. Parámetros Experimento

La Tabla 2 presenta los parámetros generales para el experimento.

Tabla 2–Parámetros Generales del Experimento

Variable	Valor	Observación
Horas Diarias	7	
Días semana	5	
Hs. Extra por Semana	1	
HorasTotalesporSemana	35	
Horas Totales Normales	210	
Horas Totales Extras	6	
Horas Totales Proyecto	216	
Sprints	6	
Puntos de Historia	214	
PruebasPorTarea	2	
Velocidad Prueba	0	
Inasistencias Pactadas	3	
Inasistencias Imprevistas	5	15%
Factor Dedicación Sprint	70,0%	
Factor Dedicación Diario	80,0%	
Días Hombre Ideal	105	
Días Hombre Real	94,5	
Team	6	
Errores por Presión en el plazo	Si	
Errores de Integración	Si	15%
Promociones en el Team	No	
Abandonos en el Team	No	
Tareas Extras	Si	
Tareas Extras No planificadas	Si	20%

La Tabla 3 presenta los valores iniciales utilizados para la corrida.

Tabla3–Experimento - Parámetros Iniciales

Sprint	(a)	(b)	(c)	(d)	(e)
1	36	1,000	35	0	1
2	34	1,000	35	35	1
3	38	1,000	35	70	1
4	35	0,900	35	105	1
5	35	0,900	35	140	1
6	36	0,990	35	175	1
Total	**214**		**210**		**6**

(f)	(g)	(h)	(i)	(j)	(k)
36	2	38	1	1	0
36	3	37	0	0	36
36	1	39	0	1	72
36	2	37	0	0	108
36	3	38	1	0	144
36	2	38	0	1	180
216	**13**	**227**	**2**	**3**	

El detalle de cada columna es el siguiente:

a) Puntos Historia: Determinado por el Team.
b) Duración planificada en horas: Surge de multiplicar las horas normales de trabajo por la cantidad de días semanales de trabajo.
c) Velocidad Estimada: Inicialmente surge de la división de (a) sobre (b). Luego se realizan ajustes para generar el BurdownChart
d) Inicio Planificado: Suma acumulada de los diferentes valores de (b).
e) horas extras: Determinadas según la necesidad del Team. Expresa las horas extras semanales.
f) Duración con Hs. Extras: Surge de la suma de (b) más (e).
g) Tareas Extras: Determinadas por el Scrum Master en función de la necesidad del Team.
h) Puntos con Tareas Extras: Surge de sumar (a) más (g).
i) Tareas Extras No Planificadas: Generado Aleatoriamente.
j) Ausencias No acordadas: Generado Aleatoriamente.
k) Inicio Recalculado: Suma acumulada de los diferentes valores de (f).

5.2. Observaciones

En este caso de experimentación y a fin de poder ejemplificar las situaciones modeladas, se presentan dos escenarios para las corridas, una donde se incluyen las tareas extras planificadas y las tareas a recodificar, y otro que presenta los datos base iniciales presentados en la Tabla 3.

Además se presentan situaciones donde como consecuencia de la inclusión de tareas se debió recalcular porcentualmente la velocidad de codificación estimada inicialmente.

5.3. Resultado del experimento

El inicio de los Sprints para este escenario se observan en la Figura 5.

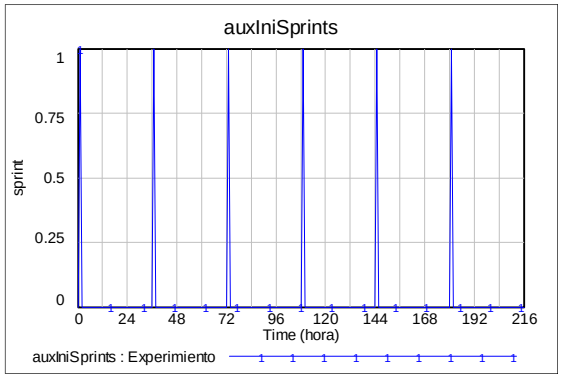


Figura 5 – Inicio de los Sprints

Otro de los gráficos relevantes para el modelo presentando es el de los puntos planificados por Sprint, para actual Experimento dichos puntos se presentan en la Tabla 3, y se muestran en la Figura 6.

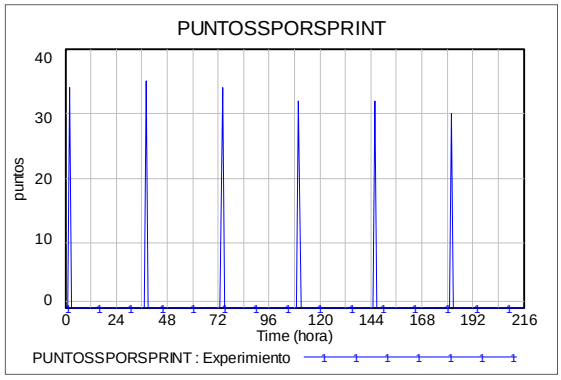


Figura 6– Puntos Planificados por Sprint

En función de los puntos planificados y la velocidad estimada de desarrollo se genera el BurdownChart, que muestra un avance ideal del proyecto y del desarrollo de los puntos planificados. En la Figura 7 se observa el comportamiento de dicha variable de nivel.

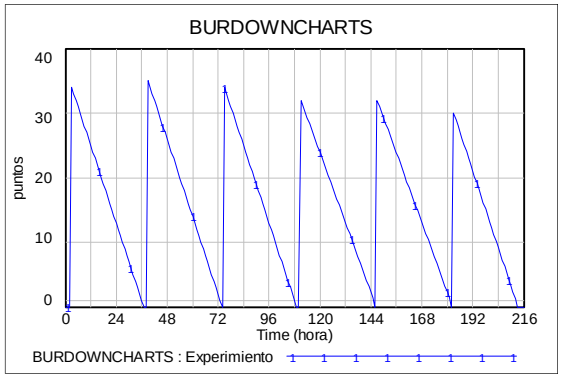


Figura 7– BurdownChart

Como se mencionó anteriormente la velocidad de avance permite completar las tareas planificadas y

acordadas por el Team. En la Figura 8 se presentan las velocidades para cada uno de los Sprints del presente caso.

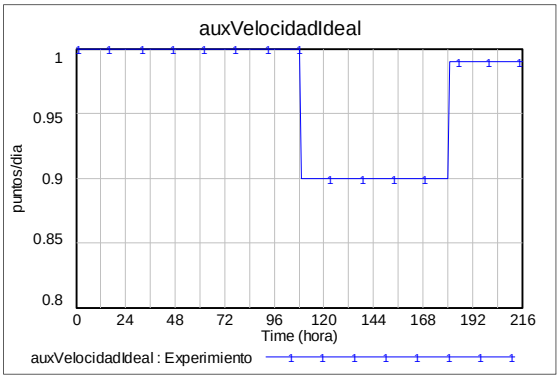


Figura 8– Velocidad Ideal Estimada

En función de la velocidad estimada y de los puntos de historia seleccionados para cada sprint, se genera un gráfico que muestra el avance ideal de cómo se deberían ir completando los puntos en cada uno de los Sprints. En la Figura 9 se observa este comportamiento descripto y representado por la variable de nivel PUNTOSPORHACER.

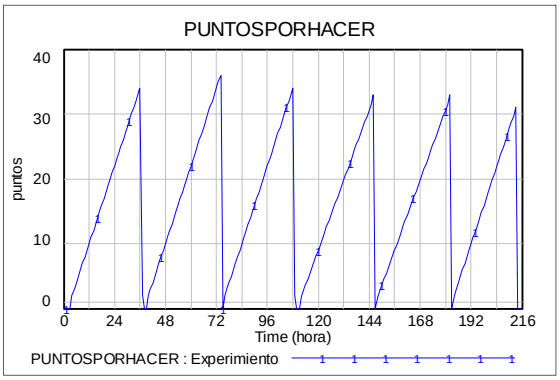


Figura 9–Puntos Por Hacer

Manteniendo el criterio inicial de que a cada punto de historia le corresponde una tarea, en la Figura 10 se observa el comportamiento de la variable TAREASPLANIFICADAS que representa el número de tareas según lo acordado inicialmente por el Team.

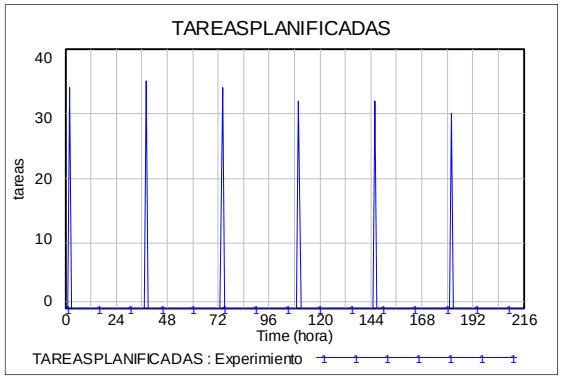


Figura 10 –Tareas Planificadas

Como se mencionó al inicio de este Experimento, y como se estimó en los parámetros iniciales de este caso se consideran las tareas que presenten errores de codificación y las tareas extras planificadas a desarrollar. Los valores que estas dos variables se presentan respectivamente en la Figuras 11y 12 respectivamente.

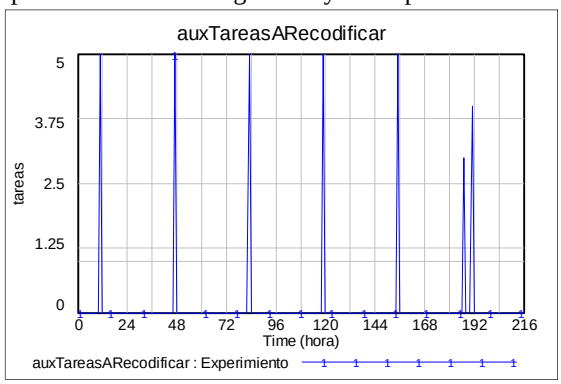


Figura 11 –Tareas a Recodificar

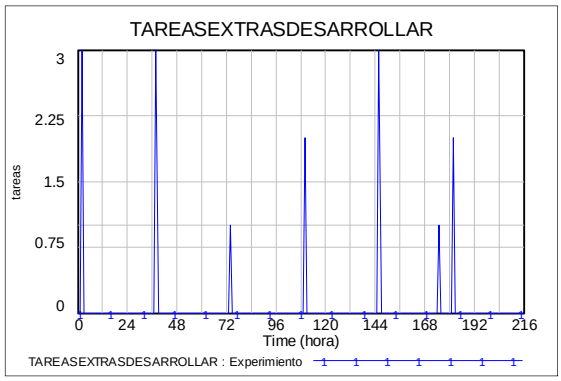


Figura 12 –Tareas Extras a Desarrollar

Dada las características particulares de este escenario dónde se presentan tareas extras planificadas y por errores de codificación (Figuras 11 y 12) se muestran en la Figura 13 de manera superpuesta las tareas originales, y las tareas originales a las que se adicionaron las tareas extras planificadas. Aunque es sutil la diferencia se

aprecia que en la corrida la cantidad de tareas es mayor que en la corrida donde no se incluían las tareas extras.

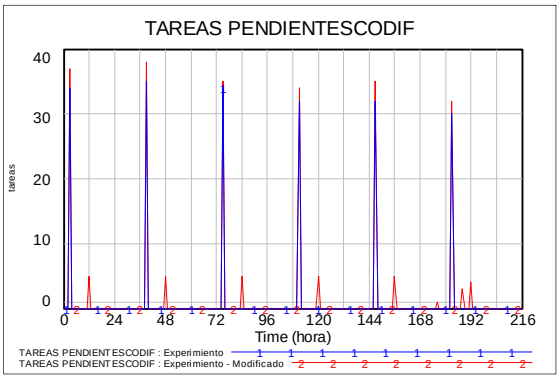


Figura 13 –Tareas Pendientes de Codificación

En la Figura 14 se observan de manera superpuesta las corridas donde se incluyeron las Tareas codificadas Con y Sin tareas extras, de manera similar a la Figura 13 en este caso se observa que al inicio de cada Sprint la corrida involucra mayor cantidad de tareas, y por lo tanto al final de cada Sprint quedan tareas pendientes por completar, dado que se mantuvo la velocidad estimada inicialmente.

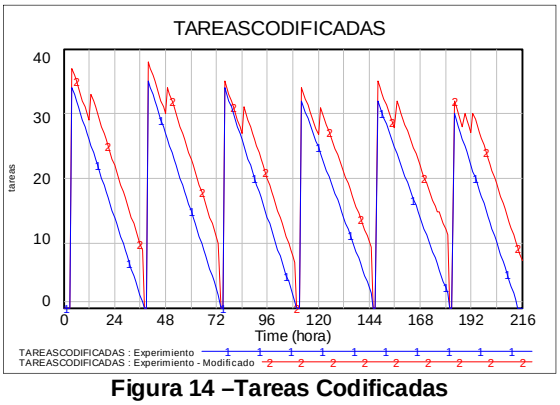


Figura 14 –Tareas Codificadas

En la Figura 15 se observa el comportamiento de la variable que representa a las pruebas que deberán ser realizadas sobre las diferentes tareas codificadas.

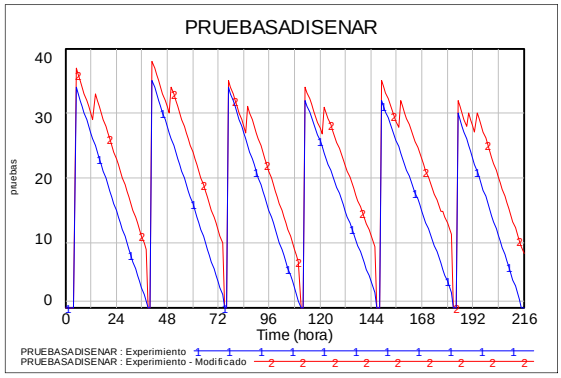


Figura 15– Pruebas A Diseñar

Como se mencionó anteriormente en este experimento, se consideran las tareas que por presión en el plazo pudieran sufrir el Team y repercutir en la codificación de las tareas. Es por ello que en la Figura 16 se observan de manera superpuesta el resultado de las diferentes pruebas realizadas sobre las tareas codificadas y aquellas tareas codificadas que no presentaron errores de codificación.

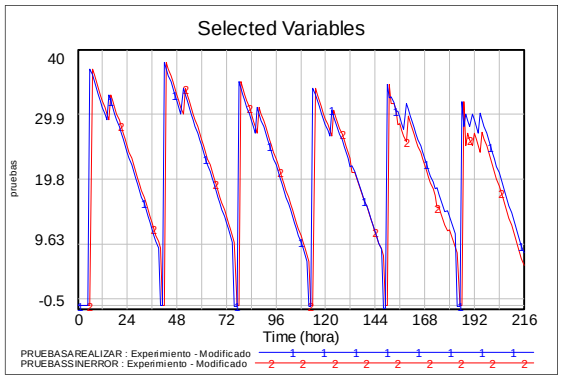


Figura 16– Pruebas Sin Errores de Codificación

En la siguiente Figura 17, se presentan las tareas que no presentaron errores de codificación o lógicos, junto a aquellas que si presentaron errores. Si bien es mínima la diferencia, en la figura correspondiente a las tareas sin errores se presentan pequeñas caídas en los momentos donde las tareas con error fueron detectadas.

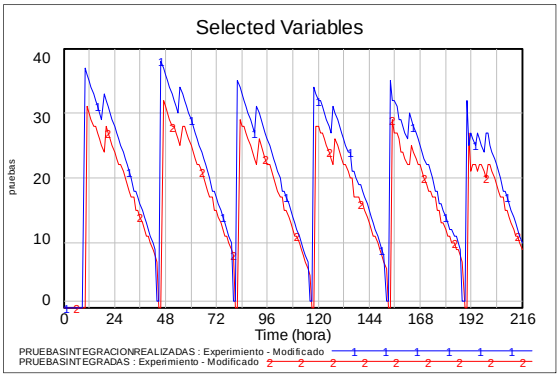


Figura 17– Pruebas Con Errores y Sin Errores

5.4. Mejoras Propuestas

En este apartado se presentan dos posibles mejoras frente a la situación que se presentó en el desarrollo del proyecto bajo sus condiciones estimadas inicialmente.

Una de las soluciones es el resultado de realizar una variación en la velocidad diaria de trabajo manteniendo los tiempos iniciales establecidos para la duración de cada Sprint. La otra consiste en la asignación de más horas a la duración de cada Sprint. Ambas soluciones se presentan de manera conjunta a fin de poder realizar una comparación de los resultados obtenidos. En ambos casos los resultados que se obtuvieron corresponden a las corridas que incluían tareas extras.

Las Figuras 18, 20, 22 y 24, son el resultado de corregir la situación planteada inicialmente con un ajuste de la velocidad de desarrollo de tareas, logrando de esta manera que todas las tareas planificadas y las tareas no planificadas finalicen en el tiempo estimado inicialmente.

Las Figuras 19, 21, 23 y 25, son el resultado de mantener la velocidad inicial, pero donde a los Sprints se le adicionaron más horas de trabajo

Se observa en la Figura 18 donde se realizó un ajuste de velocidad del 1.5%, quedan pequeños espacios de tiempo sin actividad entre los Sprints dado que el ajuste fue similar en todos los Sprints.

En la Figura 19 se destaca el aumento en la duración del proyecto cuando son adicionadas horas de trabajo a los Sprints, lo que derivaría en un retraso del proyecto y en un aumento en el cansancio del Team.

Ambas figuras presentan además el grafico de BurdownChart a modo de comparación entre lo que se estimó inicialmente y la manera en la que se desarrollaron las tareas planificadas más las tareas extras. Es válido aclarar que si bien aparecen unidades diferentes el análisis y la comparación son válidos ya que a cada punto de historia le corresponde una tarea.

Así en la figura 18 el desarrollo estimado inicialmente es similar al desarrollo real a lo largo del proyecto. Por su parte en la figura 19 es notoria la diferencia existente entre lo planificado y lo realizado. Esta última situación llevaría al fracaso del proyecto.

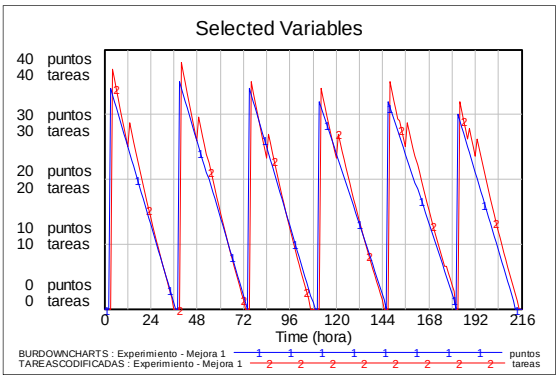


Figura 18 – Experimento – Mejora 1 – Tareas Codificadas

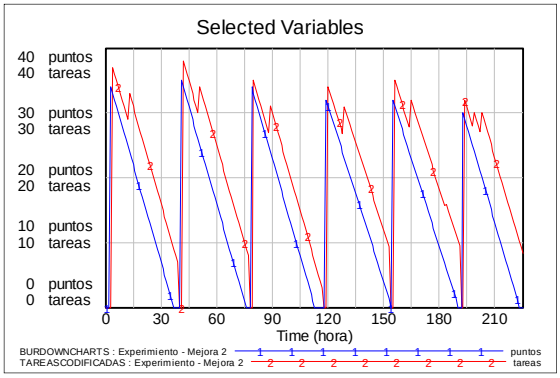


Figura 19 – Experimento – Mejora 2 –Tareas Codificadas

Al igual que en el caso de las Tareas Codificadas, las Pruebas a Diseñar también logran completarse en su totalidad en la primera de las mejoras propuestas Pudiendo observarse esto en la Figura 20.

En la 21 se presenta el resultado de adicionar más horas a los Sprints del Experimento haciéndose evidente que no se alcanzaron a realizar todas las pruebas planificadas.

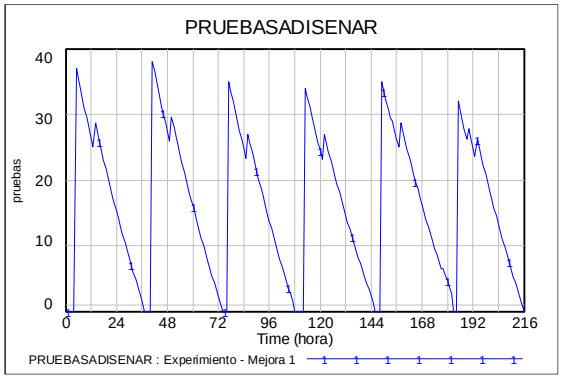


Figura 20 – Experimento – Mejora 1 – Pruebas A Diseñar

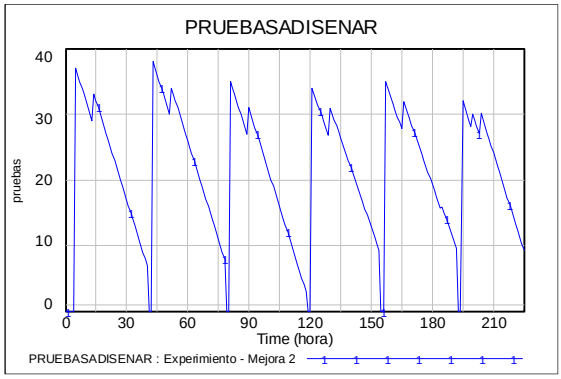


Figura 21 – Experimento – Mejora 2 – Pruebas A Diseñar

En las Figuras 22 y 23 se presentan los resultados de las pruebas realizadas y las pruebas que resultaron exitosas, tanto para la mejora donde se realizó un ajuste de velocidad como, en el caso donde se adicionaron horas a los Sprints respectivamente.

Como ocurrió tanto en las Pruebas a Diseñar como en la Codificación de Tareas, en las Pruebas de Integración Realizadas en la primera de las mejoras propuestas se logra completar la totalidad de lo estimado inicialmente. En cambio las pruebas de integración para la segunda mejora propuesta no llegan cumplirse en su totalidad.

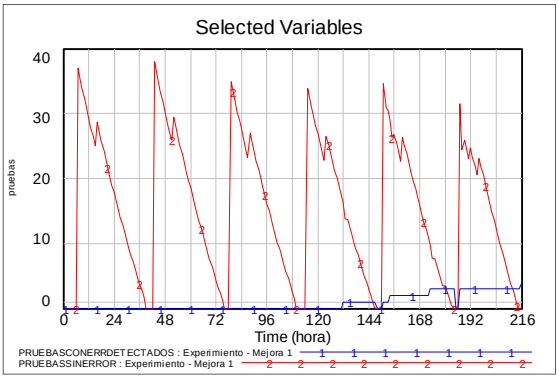


Figura 22 – Experimento – Mejora 1 – Pruebas de Codificación

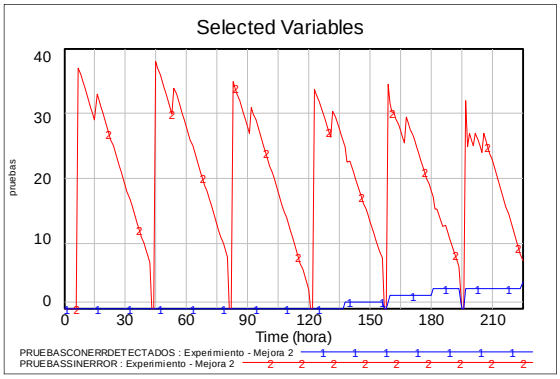


Figura 23– Experimento – Mejora 2 – Pruebas de Codificación

En las Figuras 24 y 25 se presentan los resultados de las pruebas de integración realizadas y las pruebas que resultaron exitosas, tanto para la mejora donde se realizó un ajuste de velocidad como, en el caso donde se adicionaron horas a los Sprints respectivamente.

Puede observarse en este caso como para la propuesta de mejora donde se adicionaron horas a los diferentes Sprints se producen espacios entre la finalización y el inicio de un Sprint, dada esta situación podría decirse que el número de horas adicionados fue mayor al necesario, acarreando esto una demora en la entrega del producto final.

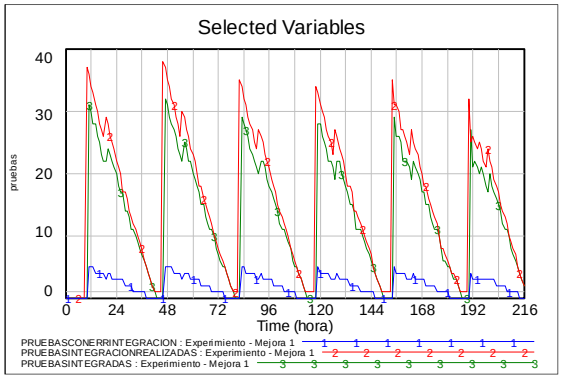


Figura 24 – Experimento – Mejora 1 – Pruebas de Integración

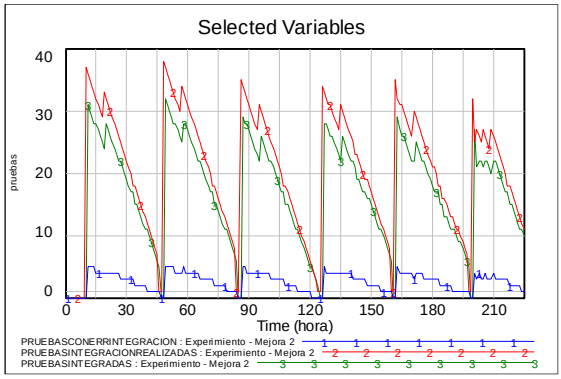


Figura 25 – Experimento – Mejora 2 – Pruebas de Integración

5.5. Conclusión del experimento.

Como se puede apreciar en este experimento la cantidad excesiva de errores y las ausencias de los integrantes del Team hicieron que el proyecto bajo las condiciones iniciales presentara un alto número de tareas pendientes de completar al final de los diferentes Sprints, por lo que bajo esas circunstancias el proyecto fracasaría inevitablemente.

Debido a esto el Scrum Master podría detectar esta situación de modo tal que en el Sprint Planning las plantearía y propondría alguna otra política de gestión del proyecto o bien optar por alguna mejora. En este experimento se optó por un ajuste a la velocidad de trabajo diario y la adición de más horas a los Sprints.

La propuesta de mejora mediante la adición de más horas a los Sprints y en consecuencia al proyecto, no resulta una opción viable ya que aun después de adicionar un promedio de 2 horas a cada Sprint el número de tareas pendientes de desarrollar es elevado. Aun, si se agregaran más horas el problema solamente se extendería ya que la mayor adición de horas traería aparejado una mayor cantidad de errores por cansancio y un atraso mayor en la finalización del proyecto.

Por otro lado la propuesta donde se realizó un ajuste de la velocidad de trabajo diario resulto la opción más viable dado que se pudieron llevar a cabo todas las tareas planificadas y las diferentes pruebas de control de errores.

6. Conclusiones

En este trabajo se presentó un caso de validación y un experimento realizado con un “Modelo Dinámico de Simulación para Proyectos de Software que utilizan Scrum”. Este modelo puede ser de utilidad para los Scrum Master y el Team analizar el efecto del uso conjunto de la metodología Scrum, Bloques de Tiempo, Artefactos y Reglas en la gestión de los mismos en diferentes escenarios. La flexibilidad de modelo permite modificar los valores de los parámetros tanto al inicio de la simulación como al momento de la ejecución de la misma. Dentro de los parámetros que se pueden establecer previo al inicio de cada simulación se encuentran: la duración y la velocidad de cada Sprint, la velocidad estimada de desarrollo de las tareas, Factores de Cansancio, de Presión en el plazo, Cantidad de integrantes del Team según su experiencia en la metodología y las tareas extras que se prevén puedan surgir. A través de la modificación de valores de los parámetros durante su ejecución el usuario puede establecer o modificar la cantidad de integrantes del Team que abandonan el proyecto, clasificar al Team mediante la asociación de estos a su experiencia en Scrum en Juniors o Expertos, cambiar la cantidad de horas estimadas de duración del proyecto, generar horas extras e inasistencia de los integrantes de manera determinística o pseudoaleatoria, entre otros.

Finalmente, luego de validar y experimentar con el modelo, se ha llegado a la conclusión de que el mismo puede ser utilizado como herramienta para evaluar el impacto de políticas alternativas de gestión y la detección de cuellos de botella.

7. Trabajos Futuros

Como trabajos futuros se planea Adicionar otros sub-sistemas que permitan ampliar el ámbito del modelo, lo que permitiría una herramienta que facilite aún más las tareas previas al inicio del proyecto al Scrum Master y al Team. Dentro de estos sub-sistemas podrían nombrarse: cálculo de costos, comunicación en el Team, Intercambio de Roles, Adquisición de Experiencia, por nombrar solamente algunos. Así también, se pretende avanzar con la construcción de modelos similares para otras metodologías consideradas ágiles como Cristal Clear y Crystal Orange [15][16] o incluir prácticas como por ejemplo Test DrivenDevelopment[17].

Por otra parte se planea utilizar este simulador en cátedras relacionadas con la Ingeniería de Software, con el fin de entrenar a futuros Scrum Masters y Teams.

Finalmente, es preciso construir bases de datos de Proyectos Scrum que contengan datos de proyectos de software reales llevados a cabo, ya que actualmente resulta difícil contar con datos post mortem de proyectos gestionados con métodos ágiles.

Bibliografía

[1] Ken y Sutherland, Jeff. Schwaber, *Agile Software Development with Scrum*, Primera ed.: Prentice Hall, 2001.

[2] Kim E. Van, Kishore Sengupta, and Luk N. Van., "Dynamics of Agile Software Development," 2009.

[3] Xiaoying Konga, Li Liu, and Chen Jing, "Modeling Agile Software Maintenance Process Using Analytical Theory of Project Investment," *International Conference on Advances in Engineering 2011*, 2011.

[4] Tamara Kasiak and Diego Alberto Godoy, "Simulación de Proyectos de Software desarrollados con XP," *XIV Workshop de Investigadores en Ciencias de la Computación.*, 2012.

[5] Diego Alberto Godoy and Kasiak Tamara, "Modelo dinámico de simulación para la gestión de proyectos de software desarrollados con XP," in *Actas XVIII Congreso Argentino de Ciencias de la Computación*, 2012, p. 10.

[6] Firas Glaiel, "Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics," Massachusetts Institute of Technology , Tésis de Máster 2012.

[7] Diego Alberto Godoy, Edgardo A. Belloni, Henry Kotynski, Hector H Dos Santos, and Eduardo Omar Sosa, "Simulando Proyectos de Desarrollo deSoftware Administrados con Scrum," in *XVI Workshop de Investigadores en Ciencias de la Computación RedUNCI*, Ushuaia, 2014.

[8] J Aracil, *Dinámica de Sistemas*. Madrid, España: Alianza Editorial, 1997.

[9] Torrealdea J, *Dinámica de Sistemas. Elementos y Estructuras de un Modelo.Construyendo modelos*.: Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco, 2003.

[10] Ventana System Inc. (2013) Vensim. [Online]. http://www.vensim.com

[11] MaríaLaura. Citón, "Método Ágil lScrum Aplicado Al Desarrollo De Un Software De Trazabilidad," Universidad de Mendoza - Facultad de Ingeniería, Mendoza, Trabajo Final de Carrera Ingeniería en Informática 2006.

[12] Tiago Keller Ferreira, "Aplicação Do Processo Ágil De Gerenciamento Scrum No Desenvolvimento De Um Jogo Digital.," UFSM, Informática/UFSM - Biblioteca Digital de Trabalhos de Graduação. , Estudo De Caso Em Empresa 2008.

[13] V. A Mahnic, "case study on agile estimating and planning using scrum," *Electronics & Electrical Engineering*, vol. 111, no. 5, 2011.

[14] Héctor MudarraTeruel, "Automatización de sistemas de desarrollo ágil Scrum: Team & Role -," sitio Web temoa : Portal de Recursos Educativos Abiertos (REA), Memoria delProyecto de Fin de Carrera de Ingeniería Informática Bellaterra, Junio de 2010 2010.

[15] Alistair Cockburn, *Crystal Clear, A Human-Powered Methodology for Small Teams.*: Addison-Wesley Professional, 2004.

[16] Cockburn, Alistair, *Agile Software Development: The Cooperative Game.*: Addison-Wesley Professional, 2006.

[17] L Madeyski, *Test-Driven Development - An Empirical Evaluation of Agile Practice.*: Springer, 2010.

Un Enfoque Integrado para la Gestión de Riesgos y Cambios en la (Re)Planificación de Proyectos en Organizaciones Fractales

Laura Tosselli
Departamento Ingeniería en Sistemas de Información
UTN FRVM, Argentina
ltosselli@frvm.utn.edu.ar

Verónica Bogado
Departamento Ingeniería en Sistemas de Información
UTN FRVM, Argentina
vbogado@frvm.utn.edu.ar

Ernesto Martínez
INGAR (CONICET – UTN)
UTN FRSF, Argentina
ecmarti@santafe-conicet.gob.ar

Abstract

El desarrollo económico y social en los últimos años ha llevado a nuevos paradigmas en las organizaciones, especialmente en las empresas de base tecnológica orientadas a la innovación. En este contexto, surge el concepto de organizaciones fractales basadas en proyectos, donde se establecen alianzas estratégicas entre diferentes organizaciones que se reúnen bajo proyectos comunes, donde la magnitud y complejidad de los mismos es creciente. Los requerimientos iniciales definen el alcance del proyecto y sufren cambios a lo largo de su ejecución, más aún cuando la naturaleza incierta y variable del entorno impactan en el plan original. Este tipo de organizaciones ha tomado consciencia de la necesidad de controlar los riesgos lo más tempranamente posible, ya que es clave para el éxito de las relaciones cliente-servidor. Así, la gestión de los riesgos y el manejo de los cambios en los requerimientos han cobrado importancia. Las propuestas tradicionales de (re)planificación de proyectos no capturan las relaciones complejas entre riesgos, requerimientos y el plan del proyecto.

En este trabajo, se presenta un modelo integrado para (re)planificación de proyectos que incluye la gestión de cambios en los requerimientos así como de los riesgos asociados a cada proyecto. A modo de ejemplificar su aplicación, se ha desarrollado un caso de estudio en el contexto de la industria farmacéutica, ya que la misma responde a las características de las organizaciones fractales basadas en proyectos, donde el proyecto se vuelve la base sobre la cual este tipo de industria se sustenta y los riesgos tienen un tratamiento especial debido a la naturaleza de la industria.

Palabras claves: (re)planificación de proyectos, organizaciones fractales, enfoque *goal-driven*.

1. Introducción

Hoy en día, las organizaciones, para cumplir con sus objetivos de negocio y ser competitivas, necesitan responder rápidamente al cambio y tomar ventaja de las oportunidades emergentes, formando alianzas estratégicas entre las mismas, donde se comparten proyectos. Por ejemplo, completar una orden, diseñar un nuevo producto, satisfacer un requerimiento de recursos, etc. [1,2]. De esta manera, los proyectos pueden considerarse unidades organizacionales autónomas y descentralizadas [3], conformando la Organización Fractal que toma como unidad de gestión a cada proyecto, siendo éste interdependiente y temporal, en el que se combinan distintos tipos de habilidades, conocimientos y recursos para lograr una meta concreta.

En este contexto, el fin último de un proyecto es entregar valor al cliente mediante un producto/servicio y favorecer la alineación estratégica de las organizaciones que participan del mismo. El alcance de un proyecto se determina mediante los requerimientos que ingresan y define los fundamentos para luego determinar y relacionar las diferentes tareas y actividades, donde los recursos son asignados. Las estadísticas relativas a las razones de por qué los proyectos fracasan, apuntan a una abrumadora necesidad de mejora en los procesos de requerimientos y análisis de negocios [4,5]. Los requerimientos sufren cambios a medida que se desarrolla el proyecto y es necesario identificar y controlar dichos cambios, ejecutando acciones que permitan mantener la factibilidad del proyecto. Estos cambios pueden deberse a: personal que no posee las habilidades o experiencia necesarias, personal que abandona el proyecto, presupuestos ineficientes, etc. [6]. El fracaso en los proyectos toma la forma de excesos en el presupuesto y plazos sin cumplir, el cual se deriva de la incapacidad para manejar dichos cambios y evaluar los riesgos asociados al proyecto para obtener un plan realista. Avanzar en un proyecto sin adoptar un enfoque proactivo en materia de gestión de riesgos aumenta el impacto que puede tener la materialización de un riesgo sobre el proyecto y que, potencialmente, podría conducirlo al fracaso [7]. Pero no solamente se debe prestar atención a los riesgos de un solo proyecto, ya que muchas veces una organización lleva a cabo varios proyectos de manera simultánea y, varios de esos

proyectos son compartidos con otras organizaciones, es decir, que el proyecto pertenece a una alianza estratégica entre organizaciones. Por lo mencionado anteriormente, se necesita contar con una correcta gestión de proyectos, específicamente en lo referido a cambios en los requerimientos y gestión de riesgos y su impacto en la (re)planificación.

Con respecto a la literatura relacionada, en [1] se presenta un modelo enfocado en el proyecto dentro de una organización fractal, apuntando a la planificación del mismo haciendo uso de roles a nivel de tareas y recursos. Otros trabajos como el presentado en [8], relacionados a la propuesta de este trabajo, definen modelos enfocados en las relaciones existentes entre requerimientos y cómo un cambio en alguno de ellos afecta a aquellos que se encuentran asociados. No se plantea en el modelo de qué manera esos cambios afectan la planificación de un proyecto. Referido a riesgos, existen modelos conceptuales aislados sin definir como la ocurrencia de un riesgo impacta en el plan general de un proyecto [9]. No existe un modelo que capture los cambios en los requerimientos y la gestión de riesgos, así como su impacto en la planificación y (re)planificación durante la ejecución del proyecto.

El presente trabajo tiene por objetivo general la formulación de un modelo conceptual que permita integrar los conceptos que influyen en la planificación y (re)planificación de los proyectos de organizaciones fractales, así como los conceptos relacionados a la gestión de requerimientos, focalizando en la gestión de cambios, con énfasis en la gestión de riesgos.

El trabajo se estructura de la siguiente forma. La Sección 2 describe cómo influyen los cambios en requerimientos y la ocurrencia de riesgos en la planificación y (re)planificación de los proyectos en las organizaciones fractales. La Sección 3 presenta el modelo propuesto, definiendo agrupaciones de conceptos relacionados según semántica común. En la Sección 4 se describe el caso de estudio utilizado para la instanciación del modelo desarrollado. Finalmente, la Sección 5 incluye conclusiones y trabajos futuros.

2. Gestión de Requerimientos y Riesgos en la (Re)Planificación de Proyectos en Organizaciones Fractales

Actualmente, las organizaciones deben reconocer que los bienes y servicios que producen evolucionan, debido al cambio en el interés del cliente, las acciones de la competencia, condiciones del entorno (gubernamentales, tecnológicas, económicas). Estos factores influyen en los requerimientos definidos inicialmente para satisfacer las necesidades de los clientes y traen aparejados nuevos riesgos u ocurrencia de los riesgos existentes asociados al proyecto que se está llevando a cabo, afectando la calidad final del producto a desarrollar. Para poder subsistir en este ambiente dinámico, manteniendo tiempos, costos y calidad, las organizaciones suelen agruparse bajo alianzas estratégicas que les permitan incursionar en nuevos mercados, mitigar amenazas, aprovechar oportunidades, lograr economías de escala, obtener mejor calidad en productos y servicios [10]. En este contexto, para lograr lo antes expuesto, estas alianzas deben contar con herramientas que soporten sus decisiones por lo que la Gestión de Proyectos (Figura 1) surge como elemento clave dentro de esta agrupación de organizaciones.

La Gestión de Proyecto, según PMI (*Project Management Institute*), incluye los grupos de procesos: iniciación, planificación, seguimiento y control, cierre (Figura 1), e implica: (1) identificar requisitos, (2) abordar las necesidades y expectativas de los interesados, (3) equilibrar las restricciones contrapuestas del proyecto: el alcance, la calidad, el cronograma, el presupuesto, los recursos y los riesgos (Figura 1).

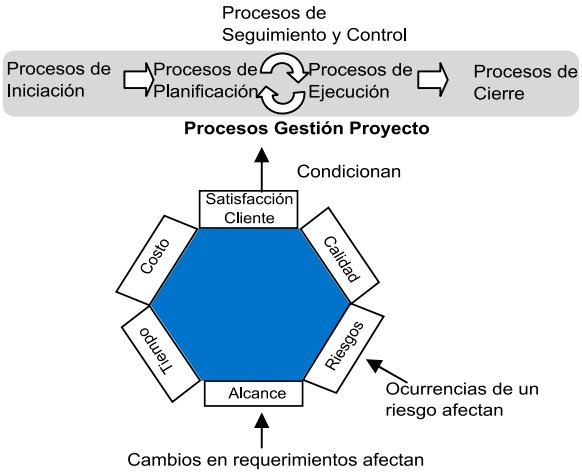


Figura 1. Factores que influyen en la (re)planificación de proyectos.

En la actualidad, la Gestión de Proyectos se enfrenta a varios retos: (1) globalización, donde la organización se expande hacia entornos internacionales y multiculturales, con equipos de desarrollo geográficamente dispersos; (2) complejidad en los proyectos que desarrolla la organización (mayor nivel de personalización por parte de los clientes); (3) necesidad de obtener resultados inmediatos y con alta calidad en proyectos donde existe un mayor grado de incertidumbre sobre la disponibilidad de recursos, duración de las tareas, políticas económicas nacionales e internacionales y multiplicación de los riesgos (debido a cambios en el alcance del proyecto, o a la incorporación de nuevas tareas y (re)programación del plan de trabajo); (4) mantener un buen ambiente en el

equipo de proyecto, adaptarse a las organizaciones planas, a las continuas reorganizaciones y fusiones y a las excesivas dependencias de las tecnologías de la información [2,11]. De igual forma que la calidad es un factor importante en la Gestión de Proyectos dentro de una organización, lo son los requerimientos y los riesgos [7,12]. Estos dos grandes aspectos dentro de la Gestión de Proyectos influyen directamente sobre la planificación llevada a cabo por la organización y seguimiento del plan durante la ejecución, y serán el foco principal de esta propuesta. Como se muestra en la Figura 1, en un proyecto existen restricciones contrapuestas que condicionan los procesos de gestión del proyecto y la relación entre dichos factores es tal que si alguno de ellos cambia, es probable que al menos otro se vea afectado [7]. Particularmente, las ocurrencias de los riesgos y cambios en los requerimientos impactan directamente en el tiempo, presupuesto, recursos y calidad del producto final.

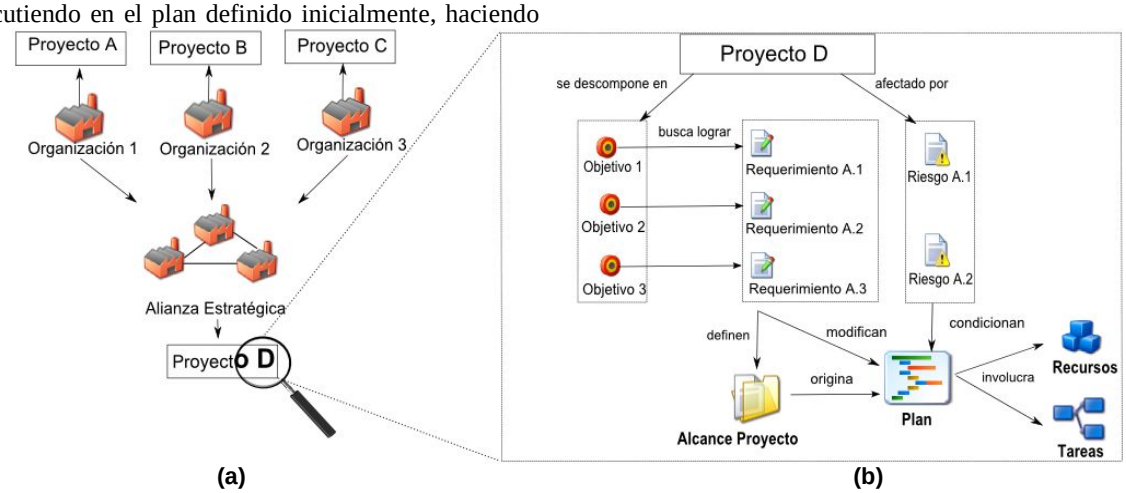
De acuerdo a lo mencionado anteriormente, las alianzas estratégicas entre organizaciones fractales deben basarse en la Gestión de Proyectos como herramienta clave, ya que se necesita realizar un seguimiento de los proyectos compartidos entre los miembros de dichas alianzas, como se muestra en la Figura 2 (a).

Cada uno de los proyectos es tratado como una unidad que posee características de flexibilidad, proactividad, adaptabilidad, auto-organización, auto-optimización, estructura orientada a objetivos, que puede interactuar con otros dentro de esta red de organizaciones, donde la toma de decisiones se realiza de manera descentralizada [13]. A su vez, en cada una de estas unidades, coexisten la gestión de cambios en los requerimientos, la gestión de riesgos y el plan, donde requerimientos cambiantes aumentan los riesgos asociados al proyecto o riegos que surgen como consecuencia de otros factores, repercutiendo en el plan definido inicialmente, haciendo necesaria la (re)planificación de elementos de trabajo y recursos involucrados (Figura 2(b)).

3. Modelo Conceptual para la Gestión Integrada de Requerimientos y Riesgos en la (Re)Planificación de Proyectos

El modelo propuesto tiene como objetivo principal lograr la integración de la gestión de requerimientos y la gestión de riesgos en un modelo de (re)planificación de proyecto. Con dicho modelo se pretende: (1) capturar los conceptos para el tratamiento de riesgos que afectan al proyecto; (2) definir conceptos necesarios para la captura de requerimientos; específicamente, la gestión de cambios asociada a los mismos; (3) plantear los conceptos relacionados a los elementos de trabajo y recursos involucrados en cada uno de ellos; (4) definir entidades que permitan utilizar métricas para cuantificar los factores involucrados en la toma de decisiones relacionadas al plan. El modelo se representa empleando el lenguaje UML [14] y siguiendo los fundamentos Orientados a Objetos [15,16], otorgando: flexibilidad que permite evolucionar en el tiempo, independencia de las diferentes tecnologías en las cuales pueda ser implementado, generalizar y abstraer el conocimiento común del dominio.

El modelo propuesto se basa en el enfoque *goal-driven*, donde los objetivos se plantean a alto nivel y son de negocio, organización o sistema, capturando las razones para las cuales una organización fractal justifica su existencia. Dichos objetivos constituyen la base para la definición de los requerimientos necesarios para alcanzarlos o cumplirlos, y donde a partir de los objetivos del proyecto se identifican y analizan los riesgos que los afectan [17].



(a) (b)

Figura 2. Alianza estratégica. (a) Esquematización de una alianza estratégica (b) Esquematización de riesgos y requerimientos y su relación con el plan dentro de un proyecto.

Los conceptos del dominio se agruparon en diferentes paquetes debido a la complejidad de los mismos y sus relaciones, como se muestra en la Figura 3. En el paquete denominado *Organización* se agrupan las entidades que definen la organización fractal y sus proyectos. El paquete *Riesgo* agrupa los conceptos relacionadas a la gestión de los riesgos asociados a los proyectos que pertenecen al paquete *Organización*; particularmente, a su definición, priorización e identificación de planes de respuesta a aplicar en caso de ocurrencia de los mismos. Dentro del paquete *Requerimientos*, se encuentran los conceptos necesarios para definir requerimientos relacionados a un proyecto y seguir sus cambios e impacto en el plan. Por último, el paquete *PlanProyecto* es el central, el cual contiene las entidades que manejan la información del plan general de cada proyecto que se están llevando a cabo dentro de la organización o red de organizaciones.

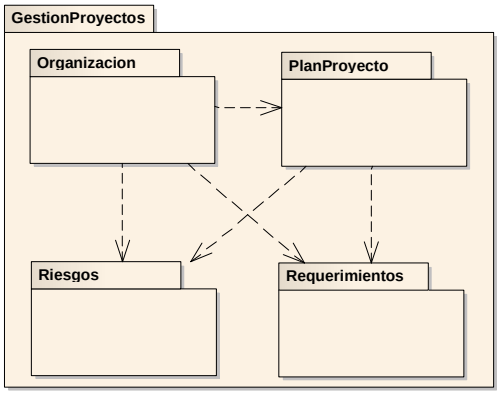


Figura 3. Organización del modelo de (re)planificación de proyectos.

3.1. Conceptos asociados a *Organizacion*

Una alianza estratégica (concepto *RedOrganizaciones*, Figura 4) involucra un conjunto de pequeñas y medianas empresas o más ampliamente organizaciones (mínimo 2, según cardinalidad Figura 4), definidas a través del concepto *Organizacion*, vinculadas mediante relaciones cliente-servidor. Las mismas comparten uno o más proyectos (entidad *Proyecto*, Figura 4), considerado/s como esfuerzo temporal para crear un producto o resultado único para obtener ventajas competitivas [7].

Las organizaciones deben responder al tipo fractal basadas en proyectos, donde cada proyecto se define como una unidad autónoma e interdependiente, y esta estructura se repite a diferentes niveles [1,2], por lo que un proyecto puede descomponerse en subproyectos (rol *subproyecto*, Figura 4).

A su vez, para cada uno de los proyectos, se puede definir más de un *Objetivo*, el cual constituye una meta hacia la cual se debe dirigir el trabajo, una posición estratégica que se quiere lograr o un fin que se desea alcanzar, un resultado a obtener, un producto a producir o un servicio a prestar [7].

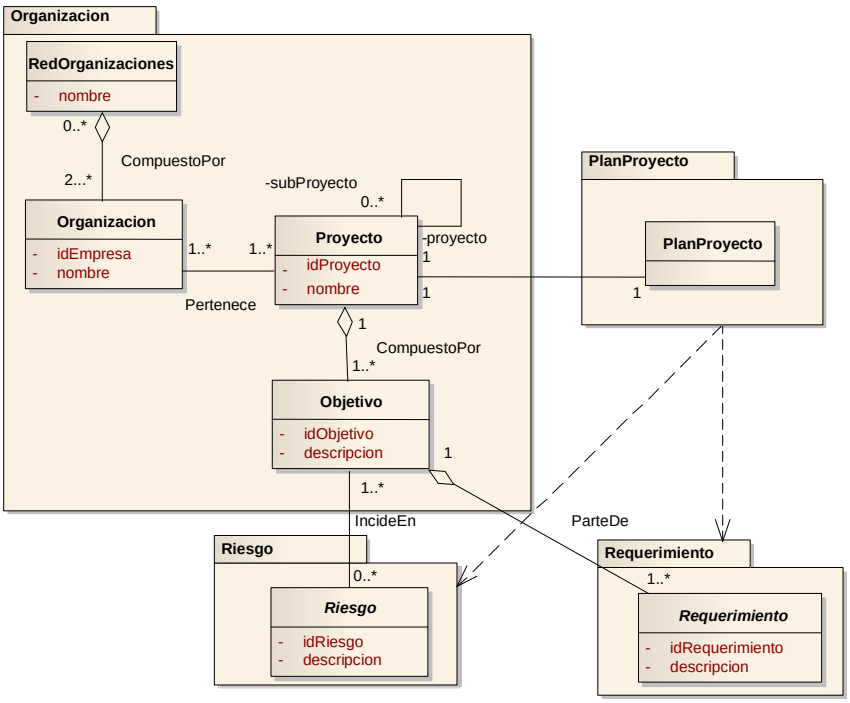


Figura 4. Conceptos relacionados a *Organizacion*.

3.2. Conceptos relacionados a *Requerimientos*

Los requerimientos son la pieza fundamental en un proyecto, entendidos como necesidades de los clientes involucrados en el proyecto, ya que marcan el punto de partida para actividades mencionadas como la planificación, básicamente en lo que se refiere al alcance, estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que constituyen uno de los principales mecanismos de control con los que se contará durante la etapa de desarrollo o ejecución del mismo [18].

De acuerdo con la ingeniería de requerimientos basada en el enfoque *goal-driven*, el análisis de los objetivos de todos los involucrados en el proyecto lleva a un conjunto de requerimientos que cumplen con esos objetivos [19, 20], lo cual se refleja en el modelo propuesto, mediante la relación que existe entre los conceptos *Objetivo* y *Requerimiento*, estableciendo que un *Objetivo* puede descomponerse en más de un *Requerimiento*. Para cada requerimiento, se establece una forma de evaluación cualitativa, mediante clasificación, y cuantitativa, mediante métricas que impliquen un valor objetivo. En la Figura 5, se resume el enfoque *goal-driven* para requerimientos.

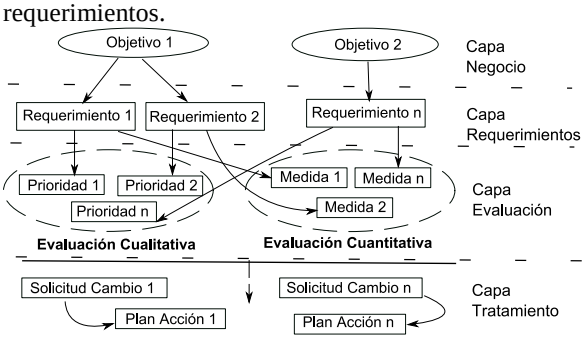


Figura 5. Enfoque *goal-driven* en requerimientos.

La gestión de requerimientos en la (re)planificación es asegurar que el proyecto cumpla con las expectativas de sus interesados, tanto externos como internos. En la Figura 6, se muestran las entidades pertenecientes al paquete Requerimientos.

El concepto principal es el *Requerimiento*, el cual se define como una condición o capacidad que un sistema, producto, servicio, resultado o componente debe satisfacer o poseer para cumplir con un contrato, norma, especificación u otros documentos formalmente impuestos. Los requerimientos incluyen las necesidades, deseos y expectativas cuantificadas y documentadas de los interesados [7]. Un requerimiento puede estar compuesto (*RequerimientoCompuesto*) por otros requerimientos compuestos o atómicos (*RequerimientoBasico*) y existe una relación recursiva entre ellos del tipo *Requiere*, *EnConflicto*, *Contiene* o *Refina* [8].

El concepto *RequerimientoBasico* se asocia a una *Restricción*, la cual se considera como una limitación aplicable, ya sea interna o externa, a un proyecto, la cual afectará el desempeño del mismo [7]. La entidad *Restricción* en este modelo representa los valores estándares que un requerimiento debe alcanzar para cumplirse, mientras que los conceptos *Medida* e *Indicador* representan la información asociada al valor real, resultado de la medición. Este indicador se encuentra definido mediante una métrica, la cual puede ser directa (*MétricaDirecta*) o indirecta cuyo cálculo depende de otras métricas (*MétricaIndirecta*), y ésta se encuentra asociada a una escala de medición (*Escala*). Además, relacionada a un requerimiento básico existe una prioridad (*PrioridadRequerimiento*), la cual permite realizar una clasificación del mismo según su importancia o impacto en el proyecto.

La inserción, eliminación o actualización de un requerimiento se realiza mediante una solicitud de cambio (*SolicitudCambio*) capturando información acerca de la gestión de cambios en requerimientos. Si el cambio es aprobado (*Aceptado*), se establece un plan de acción (*PlanAccion*), el cual se relaciona con un conjunto de acciones (*Accion*), siendo éstas elementos de trabajo cuando se incorporen al plan del proyecto o acciones que repercuten en el mismo.

3.3. Conceptos asociados a *Riesgos*

La Gestión de los Riesgos incluye los procesos relacionados a la identificación, el análisis, la planificación de respuesta a los riesgos, así como su monitoreo y control. Los objetivos de la Gestión de los Riesgos son aumentar la probabilidad y el impacto de eventos positivos, además de disminuir la probabilidad y el impacto de eventos negativos [7]. Al igual que con los requerimientos, se utiliza el enfoque *goal-driven* para la identificación, evaluación y tratamiento en caso de un evento que desencadene un riesgo [19, 20], ver Figura 7.

En la Figura 8, se presentan los conceptos y las relaciones que definen la gestión de los riesgos dentro de la gestión de proyectos. La entidad central representa al *Riesgo*, definido como un evento o condición incierta que, si se produce, tiene un efecto positivo (oportunidad) o negativo (amenaza) en los objetivos de un proyecto [7]. Los riesgos de un proyecto se ubican siempre en el futuro. Un riesgo compuesto (*RiesgoCompuesto*) constituye la entidad superior de una jerarquía de riesgos, donde la unidad atómica es el riesgo básico (*RiesgoBasico*). Un riesgo básico tiene asociado un *Impacto*, el cual determina la incidencia del riesgo en el proyecto (incluyendo un valor cualitativo, definido como una escala, por ejemplo de 1 a 5), y una probabili-

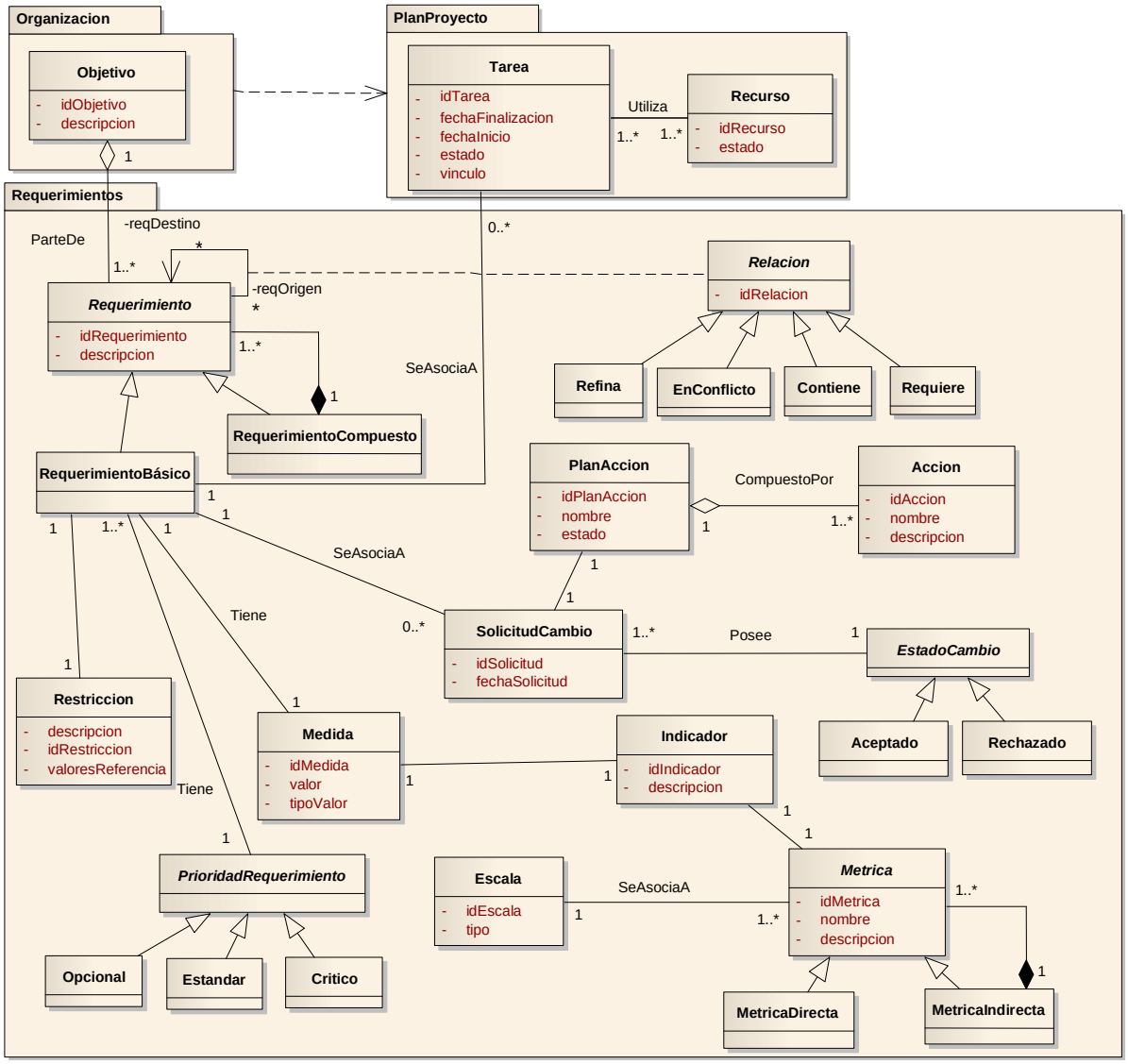


Figura 6. Conceptos relacionados a la gestión de cambios de requerimientos en proyectos.

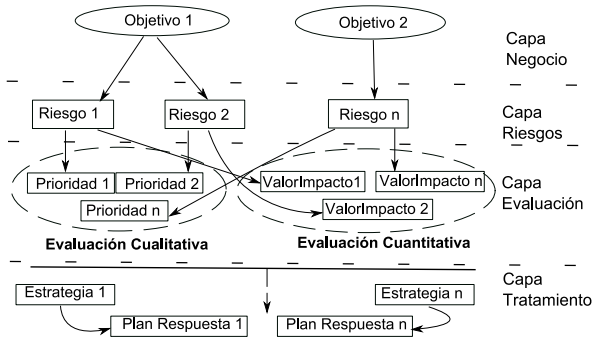


Figura 7. Enfoque *goal-driven* en riesgos.

dad de ocurrencia (*ProbabilidadOcurrencia*) que indica la frecuencia con la cual puede producirse el mismo [7].

A partir de la relación entre estas dos últimas entidades, se obtiene un valor (*Puntaje*) que permite establecer la información necesaria para priorizar el riesgo (*PrioridadRiesgo*), lo que determina la estrategia o política (*Estrategia*) a seguir frente a la ocurrencia del mismo. En caso que el riesgo sea positivo (*Oportunidad*), las estrategias incluyen: (1) explotar; (2) compartir; (3) mejorar; (4) aceptar. Si el riesgo es negativo (*Amenaza*), las estrategias incluyen: (1) evitar; (2) transferir; (3) mitigar; (4) aceptar [7]. Si ocurre un riesgo (es decir, sucede el evento asociado al riesgo, representado a través del concepto *Ocurrencia*),

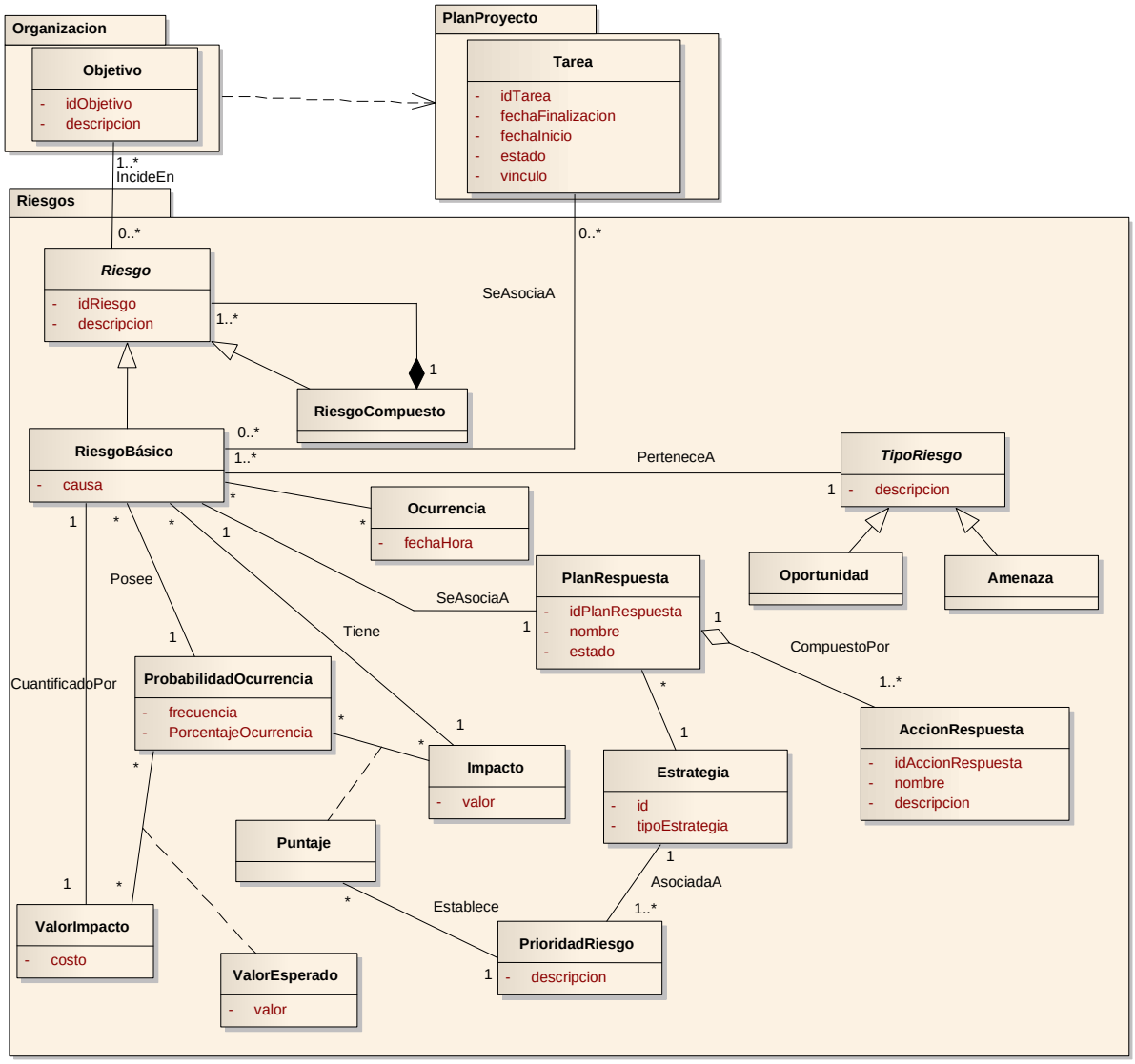


Figura 8. Conceptos relacionados a la gestión de riesgos y su impacto en el plan del proyecto.

de acuerdo a la estrategia establecida para dicho riesgo, existe un conjunto de acciones específicas para implementar dicha estrategia (cada una representada mediante *AccionRespuesta*), agrupadas bajo un plan de respuesta (*PlanRespuesta*). Estas acciones implicarán un impacto en el plan del proyecto, pudiendo incidir en más de una tarea o en más de un recurso, pero no es condición exigida en el modelo propuesto, ya que existen riesgos que afectan al proyecto en general. Además, se define en el modelo (relación entre las entidades *RiesgoBásico* y Ocurrencia, la cual es * a *), que un *RiesgoBásico* puede estar asociado a más de una *Ocurrencia*.

3.4. Conceptos de *PlanProyecto*

La planificación determinará si es factible o no llevar a cabo lo anunciado en el alcance, teniendo en cuenta los riesgos que podrían afectar al proyecto. En caso que sea posible, la planificación deberá detallar cómo se desarrollará el proyecto para cumplir con los objetivos del proyecto [7]. En la Figura 9, se modela la información necesaria para el plan del/los proyectos.

El plan (*PlanProyecto*) se asocia a un conjunto de elementos de trabajo (*ElementoTrabajo*), donde un elemento de trabajo se define como un componente de trabajo realizado en el transcurso de un proyecto [7]. Un elemento de trabajo puede componerse de otros

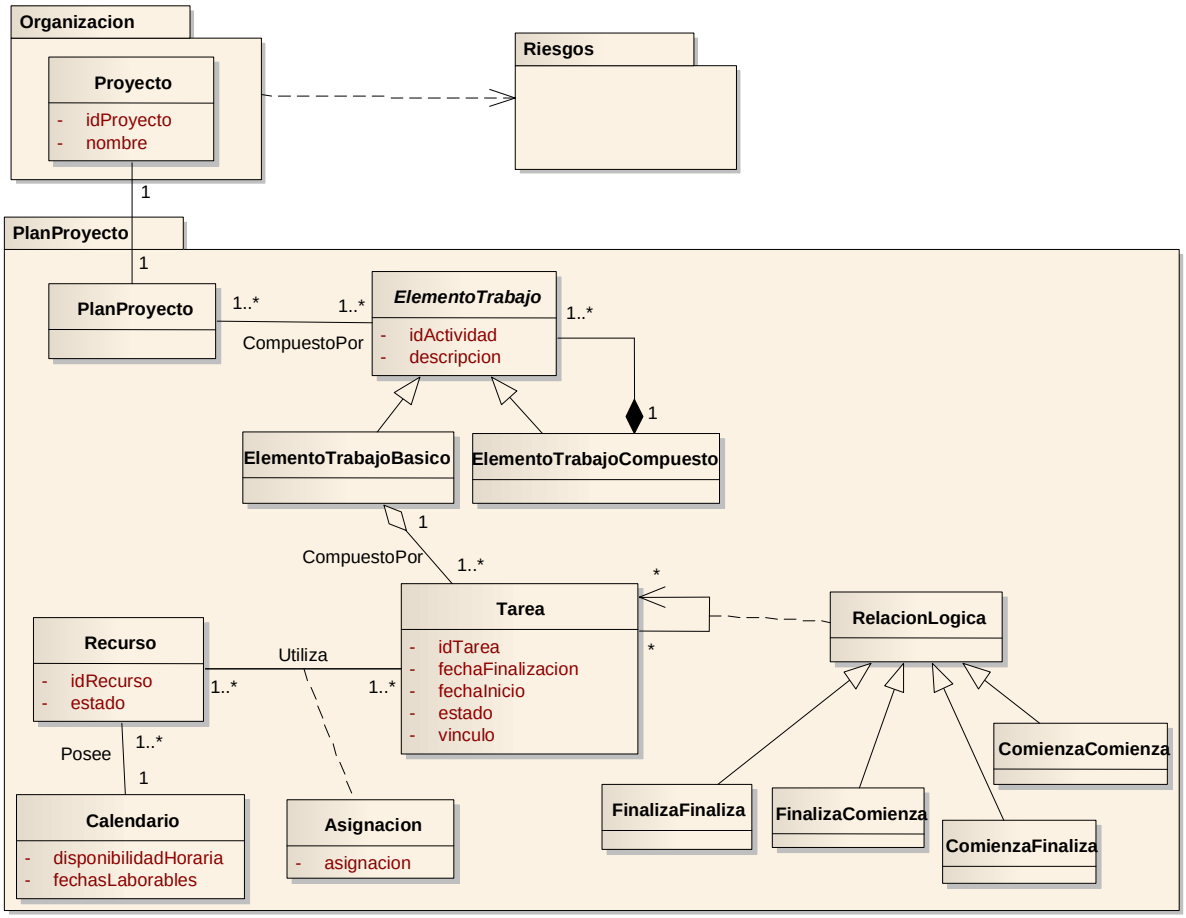


Figura 9. Conceptos relacionados al Plan del Proyecto

elementos de trabajo (por ejemplo, una etapa, fase, actividad, etc.), conformando un elemento de trabajo compuesto (*ElementoTrabajoCompuesto*), o ser atómico (*ElementroTrabajoBasico*), el cual contiene tareas (*Tarea*), considerando las mismas como una unidad de trabajo más pequeña.

Entre tareas se establecen relaciones lógicas (*RelacionLogica*) definidas como dependencias entre dos tareas del cronograma del proyecto [7]. Los cuatro tipos posibles de relaciones lógicas son: *FinalizaComienza* (FC: Una tarea no puede comenzar hasta que no termine la anterior); *ComienzaComienza* (CC: Una tarea no puede comenzar hasta que no comience otra); *FinalizaFinaliza* (FF: Una tarea no puede finalizar hasta que no finalice otra); *ComienzaFinaliza* (CF: Una tarea no puede finalizar hasta que no comience otra). Las tareas se asocian a su vez a recursos (cada uno representado por *Recurso*), necesarios para realizar la tarea relacionada, a través del concepto *Asignacion*. Los recursos pueden ser recursos humanos especializados, equipos, suministros, materias primas, materiales, presupuestos, etc. [7]. Para cada recurso, se definen los períodos de tiempo en los cuales dicho recurso estará disponible, lo cual es plasmado en la entidad *CalendarioRecurso*.

4. Caso de Estudio: Industria Farmacéutica

El caso de estudio que se presenta consiste en el desarrollo de un nuevo fármaco para disminuir la presión arterial, denominado *PRE205X*, lanzado por la compañía *AltoFarmacia*, y de su proceso de producción con el objetivo de evidenciar las etapas involucradas en estos proyectos, las capacidades y recursos requeridos, los tiempos insumidos y el riesgo e incertidumbre asociados con este tipo de proyectos a largo plazo. En el proceso de desarrollo se encuentran involucradas diferentes pequeñas y medianas compañías u organizaciones, que se encargan de llevar a cabo distintas actividades compartiendo un objetivo común: la realización del proyecto. El mismo fue desarrollado en trabajos previos [1], focalizando en la planificación, puntualmente en las tareas y los recursos. El objetivo es extenderlo para contemplar riesgos y gestión de cambios en

requerimientos y como éstos repercuten en la (re)planificación de proyectos.

Independientemente del método utilizado para la obtención de una droga, sea sintetizada químicamente, extraída desde una fuente natural o derivada a través de biotecnología, el ciclo de Investigación y Desarrollo de la misma se divide en cuatro etapas, según lo mostrado en la Figura 10. Para mejor comprensión de los conceptos, se ha partido del estado en progreso del proyecto *PRE205-X*. Las tareas de investigación de moléculas candidatas han concluido y se están realizando las pruebas toxicológicas y farmacológicas en animales de las mismas. Además, se inició la investigación para derivar rutas de síntesis para la molécula.

Focalizando en un instante de tiempo *t*, se proponen los siguientes objetos, identificados por paquetes, mostrando parcialmente la instancia del modelo, por cuestiones de espacio. Entonces, en el ejemplo propuesto, la alianza estratégica (*Red01*) reúne 5 organizaciones (*Org01*,…, *Org05*) con el objetivo de llevar a cabo el Proyecto *P01* denominado *PRE205-X*, el

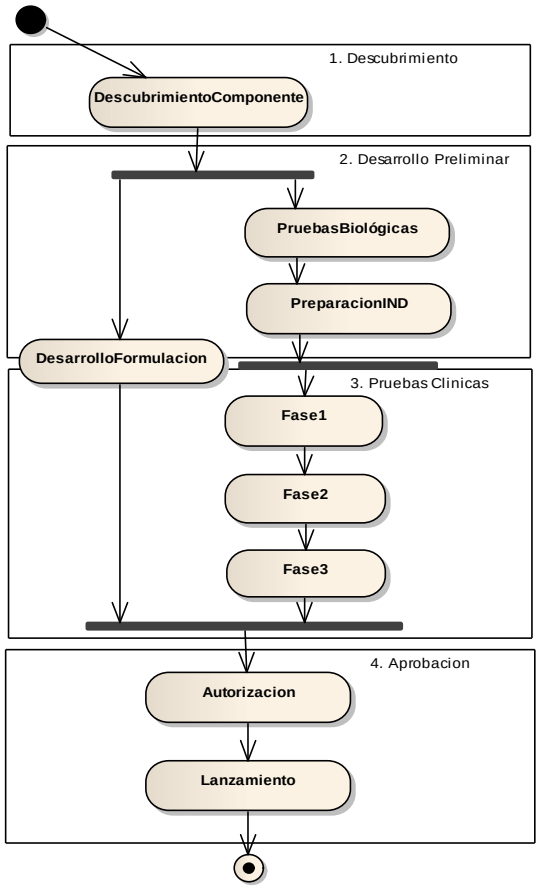


Figura 10. Proceso de desarrollo de nuevo fármaco.

cual se divide en subproyectos, de los cuales se instancian dos: "*Transferir proceso a planta piloto*" (*Sp01*) y "*Pruebas clínicas*" (*Sp02*). Para cada uno de los subproyectos instanciados, se definen algunos de los objetivos a alcanzar a través de la realización de los mismos, como por ejemplo *O1* (Tabla 1).

Tabla 1. Instancias correspondientes a los conceptos definidos para *Organizacion*.

ID	Concepto	Instancia
Red01	***RedEmpresas***	RedEmpresa1
Org01	***Organizacion***	AltoFarmacia
Org02	***Organizacion***	Universidad Nacional
Org03	***Organizacion***	Biotech
Org04	***Organizacion***	Farmatest
Org05	***Organizacion***	Technology
P01	***Proyecto***	PRE205-X
Sp01	***SubProyecto***	Transferir proceso a planta piloto.
Sp02	***SubProyecto***	Pruebas clínicas.
O1 Perteneciente a sP01	***Objetivo***	Diseñar y construir una planta piloto.
O2 Perteneciente a sP01	***Objetivo***	Seleccionar parámetros de reacción que optimizan la eficiencia del proceso.
O3 Perteneciente a sP02	***Objetivo***	Realizar las pruebas clínicas para el compuesto PRE205-X.

Algunos de los requerimientos identificados para los objetivos de la Tabla 1, se observan en la Tabla 2. Existen requerimientos compuestos como *R01*, que se desglosa en dos requerimientos básicos, identificados como *RB01* y *RB02*. Asociada al requerimiento básico *RB01* se encuentra la restricción *rt01*. En el modelo, las restricciones establecen los valores de referencia que deben tenerse en cuenta al momento de evaluar un requerimiento, mientras que las medidas se utilizan para capturar el valor real obtenido de la evaluación del requerimiento. En este caso, la *rt01* se refiere al cumplimiento de estándares para el diseño de planos y la medida (*m01*) refleja el valor obtenido en la evaluación del requerimiento, en este caso 1, lo que significa que se cumple con los estándares, según lo definido por la métrica *md01* y su escala *es01* asociada. El indicador *i01* representa el valor real obtenido.

En el proyecto *PRE205-X*, existe el riesgo *r01* (Tabla 3), con una fecha de ocurrencia (*oC01*) determinada, el cual establece que la transferencia de la ruta de síntesis del compuesto a la planta piloto requiere de una transformación carbón-a-carbón dificultosa de lograr.

Este inconveniente pone en dudas la realización de las pruebas clínicas según lo planificado para poder salir al mercado antes que su competidor [1].

Tabla 2. Instancias correspondientes a las entidades relacionadas a *Requerimientos*.

ID	Concepto	Instancia
R01 Definido para O1	***Requerimiento Compuesto***	Diseño de equipos.
RB01	***Requerimiento Basico***	Construcción de planos.
RB02	***Requerimiento Basico***	Preparación de especificaciones para adquirir equipos y materiales.
rt01 Definida para RB01	***Restriccion***	Cumplir con estándares vigentes para el diseño de planos en la industria farmacéutica.
m01	***Medida***	1
i01	***Indicador***	Cumplimiento de estándares relacionado a Md01.
md01	***MetricaDirecta***	Valores: 1 (se cumplen los estándares para diseño de planos); 0 (no se cumplen estándares).
es01	***Escala***	Nominal.
R02	***Requerimiento Compuesto***	Obtener datos para la Fase I de las pruebas clínicas.
RB03	***Requerimiento Basico***	Producir dos lotes del compuesto para la fase I de pruebas clínicas.
rt02 Definida para RB03	***Restriccion***	Los lotes a producir deben ser de 10 kg. c/u.
RB04	***Requerimiento Basico***	Administrar el compuesto en 100 voluntarios sanos.
rl01	***Relacion - Requiere***	RB03 debe finalizar para que comience RB04.
RB05	***Requerimiento Basico***	Realizar ensayos del compuesto en cápsulas.
rt03 Definida para RB05	***Restriccion***	Las cápsulas deben cumplir con la dosificación de 1 mg.
m02	***Medida***	1
i02	***Indicador***	Cumplimiento de estándares relacionado a Md02
md02	***MétricaDirecta***	Valores: 1 (las cápsulas cumplen la dosificación), 0 (no se cumple con la dosificación).
es02	***Escala***	Nominal

Este riesgo consiste en una amenaza (*am01*) poniendo en peligro el éxito del proyecto *PRE205-X*, por lo que su impacto (*ip01*) se considera Crítico, con una clasificación de 5, y una probabilidad de ocurrencia (*pO01*) igualmente Crítica, clasificada en 5. A partir de la relación entre estos elementos, se obtiene un puntaje (*pun01*) para determinar la priorización del riesgo (*pri01*, que define al riesgo como Crítico) y la estrategia (*e01*, Mitigación) a seguir en caso de ocurrencia del mismo.

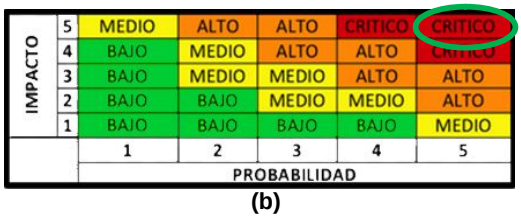
Tabla 3. Instancias correspondientes a las entidades relacionadas a *Riesgos*.

ID	Concepto	Instancia
r01 Afecta al O1	***RiesgoBásico***	Transformación carbón-a-carbón.
oC01	***Ocurrencia***	02/09/2006
am01	***Amenaza***	Amenaza.
ip01	***Impacto***	5 (Critico).
pO01	***ProbabilidadOcurrencia***	valor = 5 (Crítica). porcentajeOcurrencia= 0,8
pun01	***Puntaje***	25.
pri01	***PrioridadRiesgo***	Critico.
e01 Asociada a r01	***Estrategia***	Mitigación.
pR01 Asociado a e01 y r01	***PlanRespuesta***	Reparar el plan del proyecto *PRE205-X*.
aR01 Asociado a pR01	***AccionRespuesta***	Contratar personal especializado proveniente de otras compañías
aR02 Asociado a pR01	***AccionRespuesta***	Re-asignar personal desde las tareas de investigación.
vI01	***ValorImpacto***	$ 100.000
vE01	***ValorEsperado***	$ 80.000
r02 Afecta al proyecto	***RiesgoBásico***	Incompatibilidad y diferencias entre estructuras organizacionales.
am02	***TipoRiesgo***	Amenaza.
ip02	***Impacto***	5 (Critico).
pO02	***ProbabilidadOcurrencia***	4 (Alto).
Pun02	***Puntaje***	20.
pri02	***PrioridadRiesgo***	Crítico.

La Figura 11 muestra las matrices de impacto y probabilidad de ocurrencia, así como los resultados obtenidos del análisis cualitativo del riesgo *r01* (puntaje Figura 11(a) y prioridad Figura 11(b)).



(a)



(b)

Figura 11. (a) Matriz de medición de riesgos (b) Matriz de clasificación de riesgos

En cuanto al análisis cuantitativo, se determinó el costo asociado al riesgo (*vI01*), y el valor esperado (*vE01*) que resulta de la relación entre valor de impacto y porcentaje de probabilidad de ocurrencia, propiedad definida en *pO01*.

A partir de la estrategia seleccionada y de acuerdo al riesgo, se define un plan de respuesta *pR01* (reparar el plan definido para el proyecto *PRE205-X* insertando una actividad de transformación carbón a carbón), el cual posee acciones de respuesta (*aR01*, *aR02*, etc.) que se deberán llevar a cabo en caso de la ocurrencia del riesgo *r01*.

El riesgo *r02* es identificado y analizado cualitativamente, pero no se ha definido el evento o momento de ocurrencia del mismo.

Al desarrollar un proyecto se necesita cuantificar el tiempo y los recursos que insumirá. Para ello, es necesario definir un plan. En el caso del proyecto *PRE205-X*, se define un plan *pp01* (Tabla 4), el cual se puede descomponer en elementos de trabajo, tanto compuestos (por ejemplo *eTC01*) como básicos (*eTB01*). Además, un elemento de trabajo básico se relaciona con una o más tareas (en el ejemplo, definidas como *tr01*, *tr02*, etc.).

Entre tareas existen vínculos que determinan el momento de ejecución de cada tarea con respecto a las demás. Por ejemplo, la tarea *tr02* no puede comenzar hasta que la tarea *tr01* no haya finalizado, según la relación *rlog01*. A cada tarea se le asignan los recursos necesarios para llevarla a cabo, como es el caso, por ejemplo, de la tarea *tr01*, donde se asigna un recurso *rec01* (Ingeniero) a través de *asig01*, cuyo calendario *c01* se define como 6 hs. de lunes a viernes.

Parte de la planificación del proyecto *PRE205-X* se muestra en la Figura 12, destacando el instante de tiempo en el cual se produce el riesgo *r01* y señalando el punto de referencia en donde se deberá insertar el plan de respuesta definido para dicho riesgo.

Tabla 4. Instancias correspondientes a los conceptos de *PlanProyecto*.

ID	Concepto	Instancia
pp01	***PlanProyecto***	Plan proyectoPRE205-X.
eTC01	***ElementoTrabajo Compuesto***	Construir planos equipos
eTB01	***ElementoTrabajo Basico***	Construir planos equipos área trabajo
tr01	***Tarea***	Crear plano equipo área trabajo
tr02	***Tarea***	Obtener aprobación plano área trabajo
eTB02	***ElementoTrabajo Basico***	Construir planos equipos de laboratorio
tr03	***Tarea***	Crear plano equipo laboratorio
tr04	***Tarea***	Obtener aprobación plano laboratorio
rlog01	***Finaliza Comienza***	Entre tr01 y tr02.
rlog02	***Finaliza Comienza***	Entre tr03 y tr04.
rec01	***Recurso***	Ingeniero
c01	***Calendario***	6 hs diarias, de lunes a viernes
asig01 Asociada a tr01	***Asignacion***	6 hs. Diarias, 37 días
asig02 Asociada a tr03	***Asignacion***	6 hs. Diarias, 37 días.

5. Conclusiones y Trabajo Futuro

A partir del modelo de organización fractal orientada a proyectos, en esta propuesta, se presenta un modelo conceptual de gestión integrada de riesgos y requerimientos desde una perspectiva orientada a las metas y objetivos (*goal-driven approach*) de gestión de cambios en requerimientos y gestión de riesgos debido a la naturaleza cambiante de este tipo de organizaciones y su entorno dominado por relaciones cliente-servidor. Al ser los requerimientos los que determinan el alcance del proyecto y éste se desglosa en elementos de trabajo que dan lugar al plan compuesto de tareas y recursos, y los riesgos identificados tempranamente dan lugar a planes de contingencia cuyas acciones de respuesta se

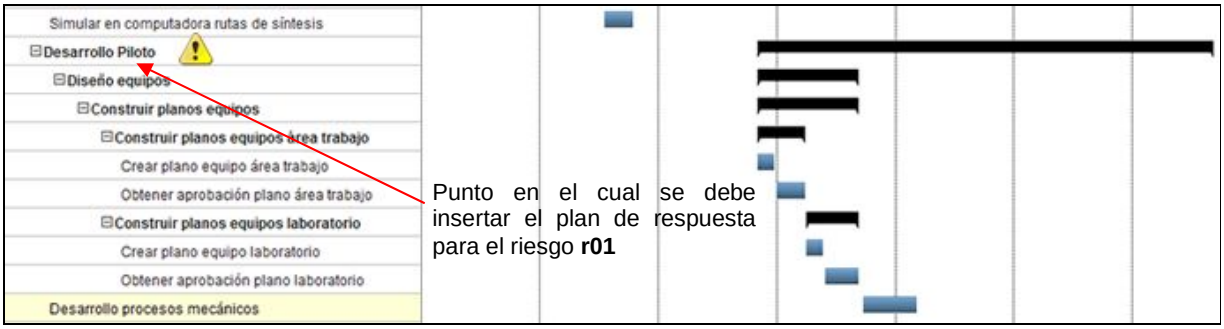


Figura 12. Seguimiento del riesgo y el impacto en el plan

traducen en tareas que utilizan recursos o en la eliminación o modificación de las ya existentes, resulta relevante la correcta gestión de los mismos. Por ejemplo, si durante la ejecución de un plan de un proyecto en un tiempo *t* ocurre uno de los eventos asociados con los riesgos identificados, un conjunto de tareas correspondientes al plan de contingencia debería automáticamente ser integrado en el plan general del proyecto para dar respuesta al riesgo presentado. Lo mismo ocurre en el caso de un cambio en un requerimiento especificado. Así, el modelo propuesto captura e integra los diferentes conceptos para poder realizar una (re)planificación en este tipo de organizaciones, más aún cuando las mismas se reúnan bajo proyectos comunes. Este modelo permite la (re)planificación derivada del seguimiento y control del proyecto, donde la misma es imperativa en la industria, sobre todo en aquellas en las que los riesgos deben planificarse y tratarse en caso de ocurrir, como por ejemplo industrias petroquímicas, de software, etc.

Como trabajo futuro, se pretende desarrollar un entorno multiagente en el cual los agentes de un proyecto (por ejemplo, gestor de requerimientos, gestor de riesgos, gestor de plan) puedan interactuar y, a partir de dicha interacción, aprendan a determinar posibles acciones conjuntas para enfrentar eventos que demandan cambios, referidos a requerimientos o a riesgos asociados a los planes de los proyectos de la organización, asistiendo al administrador de proyecto en la toma de decisiones sobre la actualización y/o ejecución del plan en un contexto dinámico.

6. Agradecimientos

Se agradece el apoyo financiero brindado por el proyecto PID 25/OR01 “Planificación automática en sistemas cognitivos de producción integrando aprendizaje por refuerzos con abstracciones lógicas y relacionales”.

7. Referencias

[1] Canavesio, M. *Un modelo de compañía fractal para la integración entre empresas usando proyectos*. Tesis Doctoral. Universidad Tecnológica Nacional, Santa Fe, Argentina, 2007.

[2] Canavesio, M., Martinez, E. Enterprise modeling of a project-oriented fractal company for SMEs networking. *Computers in Industry*, Vol. 54, 2007, pp. 794–813.

[3] Warnecke, H.J. *The Fractal Company: A Revolution in Corporate Culture*. Berlin: Springer-Verlag. ISBN: 3642781268, 1993.

[4] Heembrock, P. Hitting the Mark: The Impact of Requirements Maturity on Project Outcome. *IAG Consulting*, New Castle DE, Estados Unidos. En: http://www.iag.biz, último acceso 26/01/2014, 2009.

[5] Ellis, K. Diagnosing Requirements Failure: A benchmark of the current capability of organizations in doing business requirements and an assessment of the underlying causes of poor quality requirements. *IAG Consulting*, New Castle DE, Estados Unidos, 2009. En: http://www.iag.biz, último acceso 26/01/2014.

[6] Barker, D. Requeriments modeling technology: A vision for better, faster, and cheaper systems, *Proceeding VIUF '00 Proceedings of the VHDL International Users Forum Fall Workshop* (VIUF'00), 2000, pp. 3.

[7] PMI. *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)*. Project Management Institute, 4ª edición. ISBN 978-1-933-890-72-2, 2008.

[8] Goknil, A, Kurtev, I, Van Den Berg, K, Spijkerman, W. Change impact analysis for requirements: A metamodeling approach. *Information and Software Technology*, Vol. 56, 2014, pp. 950-972.

[9] Braendeland, G. Refsdal, A., Stolen, K. Modular analysis and modelling of risk scenarios with dependencies. *The Journal of Systems and Software*, Vol. 83, 2010, pp. 1995–2013.

[10] Westkamper, E., Huser, M., Van Briel, R. Transicion processes: A survey of German industry. *Journal of Materials Processing Technology*, Vol. 106, 2000, pp. 141–151.

[11] Anavi-Isakow, S. y Golany, B. Managing multiproject environments through constant work-in-process. *Internacional Journal of Project Management*, Vol. 21-1, 2003, pp. 9-18.

[12] Sommerville, I. *Software Engineering*. Addison-Wesley, 9ª Ed. ISBN: 0137035152, 2010.

[13] Berberoğlugil, B. y Satir, S. Fractal Organization Management: University As A Model. *Interactive Collaborative Learning (ICL)*, 2012, pp. 1-7.

[14] UML. Object Management Group. UML 2.4, Superstructure, 2010. En: http://www.omg.org/spec/UML/2.4/Superstructure, último acceso 26/01/2014.

[15] Booch, G. *Object-oriented analysis and design with applications*. Redwood City, CA: Benjamin-Cummings, 1994.

[16] Booch, G. *Object solutions: managing the object-oriented project*. Reading, MA: Addison-Wesley, 1995.

[17] Anton, A. Goal-based requirements analysis. *Requirements Engineering*, 1996, pp. 136-144.

[18] Pressman, R. *Software Engineering: A Practitioner´s Approach*. 7º Edición. McGraw- Hill Higher Education. ISBN: 0073375977, 2010.

[19] Asnar, Y., Giorgini, P., Mylopoulos, J. Goal-driven risk assessment in requirements engineering. *Requirements Eng*., Vol. 16, 2011, pp. 101–116.

[20] Islam, S., Mouratidis, H. Weippl, E. An empirical study on the implementation and evaluation of a goal-driven software development risk management model, *Information and Software Technology*, Vol. 56, 2014, pp. 117–133.

Proceso para Asistir la Migración de Información a Arquitecturas Cloud Computing en PyMEs

Bernal Tomadoni, L., Pytel, P., Pollo-Cattaneo, M
Grupo de Estudio en Metodologías de Ingeniería de Software,
Facultad Regional Buenos Aires, Universidad Tecnológica Nacional.

Abstract

El concepto de Cloud Computing hace referencia a un modelo que permite habilitar acceso a la red, de forma conveniente y en demanda, a un fondo compartido de recursos computacionales. Se ha observado la falta de una metodología íntegra y homogénea que permita analizar la conveniencia y la viabilidad de la adopción de esta tecnología de forma completa, dentro de las Pequeñas y Medianas Empresas (PyMEs). Las técnicas y herramientas existentes dejan de lado varios aspectos importantes a tener en cuenta. A su vez, el estado del arte actual, no ofrece un marco donde se aprecie cómo llevar a cabo posibles implementaciones de dichas metodologías. Por lo tanto, el presente trabajo tiene como objetivo proponer un proceso que permita evaluar las condiciones y características de una PyME para migrar sus datos, y ciertas aplicaciones, a un modelo basado en la Nube. Además, se proponen procedimientos para asistir la toma de decisiones, teniendo en cuenta aspectos técnicos, económicos y organizacionales.

1 Introducción

Según el Instituto Nacional de Estándares y Tecnología de Estados Unidos (NIST), el concepto de Cloud Computing hace referencia a un "modelo que permite habilitar acceso a la red, de forma conveniente y en demanda, a un fondo compartido de recursos computacionales configurables (redes, servidores, almacenamiento, aplicaciones y servicios) que puede ser provisto rápidamente y con un mínimo esfuerzo de administración o interacción con el proveedor" [1].

Este concepto también abarca al hardware y los sistemas de software en los centros de datos que proveen los servicios entregados por demanda [2]. Estos servicios se denominan 'Software como Servicio' (SaaS), mientras que los recursos IT necesarios (hardware y software del centro de datos) son lo que se llama 'Cloud' o 'la Nube'. Esta Nube se basa en la virtualización de recursos de hardware, cuya comercialización se encuentra acompañada de sistemas de software que permiten gestionar la arquitectura subyacente. Por lo tanto, el paradigma de Cloud Computing ayuda a optimizar los procesos de almacenamiento y manejo de datos, haciendo más eficaz la toma de decisiones en una organización.

En [3] se ha observado la falta de una metodología íntegra y homogénea que permita analizar la conveniencia y la viabilidad de la adopción de esta tecnología de forma completa, dentro de las Pequeñas y Medianas Empresas. La mayor parte de los trabajos de investigación en este área, se refieren a organizaciones con dimensiones mucho mayores y dinámicas de trabajo muy diferentes, lo que impide generar un modelo homogéneo que permita aplicarse de forma genérica. Las técnicas y herramientas existentes dejan de lado varios aspectos importantes a tener en cuenta. Entre ellos, se encuentran factores organizacionales en general, como las consecuencias de la integración de la nueva tecnología que se evalúa adoptar con la infraestructura actual, y la adaptación de los empleados. A su vez, el estado del arte actual, no ofrece un marco donde se aprecie cómo llevar a cabo posibles implementaciones de dichas metodologías, sino que se proponen recomendaciones aisladas, dejando de lado la integridad.

En este contexto, el presente trabajo continúa el trabajo presentado en [4] para proponer un proceso que asista a los miembros de una Pequeña y Mediana Empresa (PyME) en la decisión de migrar sus datos (y ciertas aplicaciones) a un modelo basado en la Nube, a tal efecto se identifican y evalúan las características de la organización.

La estructura del trabajo es la siguiente: primero se identifican las características y condiciones que deben ser evaluadas para proyectos de computación en la Nube en el ámbito de las PyMEs (sección 2). Luego se presentan las etapas para asistir la toma de decisiones a la hora de

evaluar la adopción de esta tecnología (sección 3). La implementación de dicho proceso es ilustrado mediante el análisis de dos casos de estudio, uno exitoso y otro que finalizo en fracaso (sección 4). Finalmente, se indican las conclusiones obtenidas con las correspondientes futuras líneas de trabajo (sección 5).

2 Características a ser evaluadas

Antes de emprender todo proyecto software, la organización debe decidir si es conveniente realizarlo o no. Para poder tomar esa decisión, la cual es compleja y depende de una multitud de factores, es necesario conocer el impacto que ese software va a causar en la organización y los riesgos que la organización posee por su construcción [5]. A tal efecto, es necesario estudiar las características del proyecto y la organización a través de una evaluación de la viabilidad técnica y económica del proyecto (también conocida como estudio de factibilidad). Como resultado de esta evaluación se puede determinar si se cumplen las condiciones para garantizar la finalización satisfactoriamente del desarrollo del sistema software [6].

Con la adopción de un modelo basado en la Nube ocurre algo semejante. Pero es importante resaltar que cualquier variación en cuanto a la percepción de los riesgos de alojar determinada información en servidores de terceros, podría llevar al fracaso del proyecto [6]. Por lo tanto el estudio de la viabilidad en estos proyectos tiene una mayor importancia para identificar los riegos que deben ser monitoreados y controlados.

Es importante tener en cuenta las características propias de cada empresa al momento de llevar a cabo cualquier proyecto informático dentro de la organización, y más aún, si se trata de la migración de datos y aplicaciones propias. Los volúmenes de información generados por una empresa dependen de la dimensión de la misma, la cantidad de clientes, empleados, transacciones comerciales, y del volumen de los servicios o cantidad de bienes que produce, entre otras cosas. De esta manera, existen clasificaciones como la de PyMEs, que permiten categorizar a las organizaciones según el rubro al cual pertenecen, su volumen de ventas y cantidad de empleados [7-9]. En este trabajo se consideran las características de las PyMEs descriptas en [4].

Por otra parte, en este dominio se han realizado diversos trabajos para la identificación de las características que una organización debería poseer para migrar a [10-13], pero no existe un proceso completo que permita analizar al comienzo si es posible migrar la información o no.

Asimismo, en [14] se pretende generar un análisis integral del potencial de Cloud Computing para PyMEs. Se plantean métricas de costos interesantes, y se obtiene como resultado, un número que, si se encuentra entre determinado rango de valores, permite decidir si adoptar el modelo Cloud o no. Sin embargo, deja de lado la evaluación de cuestiones importantes sobre aspectos del negocio.

El proceso propuesto en este trabajo se presenta dividido en etapas, cuya implementación y evaluación tienen como objetivo final facilitar y asistir la toma de decisiones sobre la migración de su información; es decir, auxiliar a los miembros de la organización en las etapas iniciales de la adopción de este tipo de tecnologías. Para ello, se tienen en cuenta las características utilizadas por los modelos antes citados para generar un proceso propio que generará como resultado un valor para decidir si es conveniente y oportuna la migración a esta arquitectura.

Dicho proceso engloba las características en cuestión, clasificándolas en dos grupos [7]:

- aquellas características que tienen en cuenta los *factores cualitativos,* como la respuesta al cambio de los miembros de la organización, el grado en que los empleados trabajan en ubicaciones remotas, requisitos de seguridad y el nivel de estandarización de los procesos de la empresa; y
- aquellas en las que intervienen *aspectos cuantitativos* que incluyen desde la estimación temprana de beneficios en la Nube contra los beneficios en un centro de datos interno, el estudio de la infraestructura actual, y el análisis de compatibilidad y portabilidad de aplicaciones.

3 Etapas del Proceso Propuesto

En cada una de las subsecciones que se encuentran a continuación, se describen las cinco etapas del proceso propuesto, el cual se ilustra en la Figura 1.

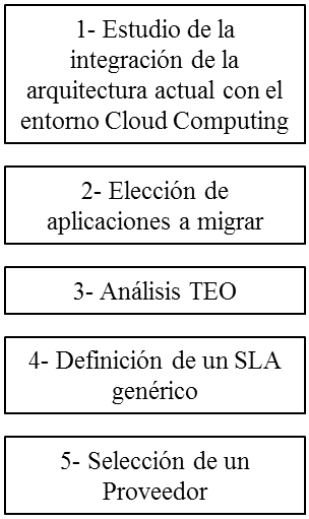


Figura 1. Etapas del Proceso Propuesto.

3.1 Estudio de la integración de la arquitectura actual con el entorno Cloud Computing

Como primer paso es necesario detallar de forma precisa la arquitectura actual de la empresa. Esta fase constituye un primer filtro, para aquellas organizaciones que no reúnen las características necesarias de: virtualización avanzada, gestión automatizada y estandarización en sus procedimientos.

Antes de considerar un modelo Cloud es fundamental tener en cuenta cuáles son las características actuales y potenciales de la infraestructura de hardware y software de la empresa. Idealmente, se debería contar con una infraestructura dinámica, con capacidades de virtualización, estandarización de servicios y provisión automatizada de recursos. Como señala [15], una infraestructura preparada para un modelo Cloud deber tener las siguientes características y capacidades:

- Ser abierta, basada en estándares y orientada a servicios. Al lograr altos grados de estandarización es posible simplificar el manejo de la arquitectura subyacente a las aplicaciones y datos de la organización. Esto depende no sólo de la infraestructura sino de los procesos y el control que se tenga sobre los distintos activos informáticos. De no contar con estas características, la migración a la nube también será posible, pero requerirá un mayor trabajo de preparación previo y, por lo tanto, un mayor costo.
- Tener un grado de virtualización avanzado. Es decir que la disposición de los recursos IT es provista a través de sistemas autónomos y no manualmente.
- Contar con medidas de seguridad, que puedan ser integradas con la nueva arquitectura. Es importante explicar que Cloud Computing no introduce ningún nuevo problema de seguridad que no haya sido analizado para la seguridad informática en general [15]. Sin embargo, es necesario diferenciar el Cloud Privado del Cloud Público:
  - *Cloud Privado*: son escenarios donde las compañías realizan sus operaciones fuera de línea, ejecutando aplicaciones seguras en Datacenters, o 'Internal Clouds'. La principal ventaja es la transferencia y almacenamiento de datos controlados, y el menor riesgo de infiltración.
  - *Cloud Público*: son escenarios donde las compañías necesitan mover datos o aplicaciones desde su interior al exterior. Involucra recursos y servicios que son vendidos a un proveedor de SaaS. El servicio vendido se denomina 'Utility Computing' y permite administrar la infraestructura provista así como, entre otras cosas, aumentar o reducir la cantidad de máquinas virtuales en la medida que sean requeridas. Sin embargo, al tratarse de un tipo de almacenamiento compartido, tienen mayor riesgo de infiltración.
- Asimismo, como se menciona en [4], no se pueden dejar de lado los "requisitos de cumplimiento normativo" es decir, los temas regulatorios en ciertas industrias, que no permiten migrar determinados datos a un modelo compartido. Según [11] se pueden aplicar requisitos normativos estrictos para algunos datos, como puede ser la información financiera o, en salud, ciertos datos de los pacientes como las historias clínicas. Las leyes y normas obligan a la estandarización de los procedimientos organizacionales. En este contexto, las PyMEs tienden a aumentar el nivel de normalización de sus procesos [16] a medida que crecen. Por tal motivo, es vital tener en cuenta cuál es la información estandarizada, ya que ésta será la más adecuada para migrar a la nube.

Posteriormente al análisis de las características internas de la organización, es recomendable preguntarse si conviene o no seguir con las etapas del proceso de evaluación y adopción de Cloud Computing. Lo mencionado en esta sección debe servir como base para esbozar una respuesta temprana. Si los responsables del proyecto consideran que la organización reúne las características mencionadas, entonces se estudiar la compatibilidad e integración de las funcionalidades que se alojarán en la Nube con las plataformas brindadas por los proveedores, lo que lleva a la siguiente etapa.

3.2 Elección de aplicaciones a migrar

Como segundo paso, antes de abordar un modelo Cloud, es prioritario identificar cuáles son las

aplicaciones existentes en la organización que se pueden migrar a este tipo de servicio y qué no. Para ello, Fourcade [12] categoriza las aplicaciones en dos grupos, según la arquitectura de servicio. De esta manera, en el primer grupo propuesto se incluyen las aplicaciones que se alojarán en Nubes Públicas y en el segundo las que estarán en Nubes Privadas. Las primeras corresponden a aplicaciones destinadas al intercambio de información no sensible entre el usuario final y la organización (como aplicaciones de mensajería instantánea, notificaciones, entre otras); dejando fuera toda aplicación que involucre la producción de datos sensibles como los esperados en procesos de explotación de información en bases de datos, entre otros.

Por lo tanto, para ingresar a un Cloud Público las funcionalidades que suelen ser las más apropiadas son:

- Infraestructura para entrenamiento y demostración.
- Infraestructura de Video y Voz sobre IP (también conocida como 'Infraestructura VoIP').
- Aplicaciones relacionadas a mensajería y notificaciones.

Mientras que para ingresar a un Cloud Privado, las más apropiadas suelen ser:

- Bases de Datos transaccionales.
- Minería de Datos y Minería de Texto.
- Data-warehouses y data-marts.
- Aplicaciones ERP.

Al momento de decidir las aplicaciones a migrar en servidores externos es preciso que el usuario tenga noción que deja la seguridad de su información en manos de un tercero, en este caso el proveedor de SaaS. Para garantizar la integridad, confidencialidad y disponibilidad de la información, en [17] se plantean los requisitos fundamentales que una aplicación debería poseer si se quiere integrar a una plataforma SaaS. Se destaca, entre ellas, la importancia de contar con mecanismos robustos de autenticación y autorización de acceso a dichas aplicaciones.

Como conclusión de esta etapa, se puede afirmar que si la organización no llegara a detectar una necesidad real que involucre un cambio de tecnología para almacenar, mantener y distribuir sus aplicaciones, o expandir los repositorios donde persiste actualmente la información, significa que tampoco necesita migrar esos datos a través de un modelo Cloud. Ahora bien, si la organización identifica una o varias aplicaciones que, por ejemplo, debe distribuir a empleados que acceden de forma remota, o que producen mucha información y son accedidas constantemente, se deberá llevar a cabo un evaluación cuantitativa, como la propuesta en la etapa siguiente.

3.3 Análisis Técnico Económico y Operativo (TEO)

En esta etapa se realiza el análisis cuantitativo considerando las dimensiones técnicas, económicas y operativas de la organización. Es decir, que se lleva a cabo el estudio de viabilidad propiamente dicho, que involucra las variables correspondientes, y su posterior estimación de costos (variables económicas).

○ Evaluación de la Viabilidad:

Un modelo completo para la evaluación cuantitativa de viabilidad en proyectos Cloud Computing forma parte de una de las futuras líneas de investigación, pero en este trabajo se presentan los pasos generales del modelo. A tal efecto, se utiliza como base el modelo propuesto en [18], que establece un procedimiento para realizar dicha evaluación teniendo en cuenta un conjunto de características valoradas mediante cinco valores lingüísticos que son evaluadas aplicando el principio de los Sistemas Expertos Difusos [19]. Dichas características pueden ser clasificadas de acuerdo a las condiciones que el proyecto debe cumplir para ser considerado viable; las cuales son:

- Condiciones que determinan la *Plausibilidad* del proyecto (referidas a aquellas que hacen posible llevar a cabo la migración): si los datos y funcionalidades a migrar tienen importancia para el negocio y son fundamentales para cumplir sus objetivos; si existen personas en el equipo de trabajo con un mínimo conocimiento sobre tecnologías basadas en la Nube; y si los volúmenes de información de la organización superan o podrían superar la capacidad instalada.
- Condiciones que determinan la *Adecuación* del proyecto (que evalúan si la solución elegida es la apropiada): si existe un parámetro o función de crecimiento del volumen de datos y aplicaciones; y si existen obstáculos para la adquisición y mantenimiento de servidores propios.
- Condiciones que determinan el *Éxito* del proyecto: si la información a migrar se encuentra estandarizada; si los repositorios de datos están implementados con tecnologías que permiten su fácil manipulación; si no existen barreras técnicas/físicas (ancho de banda, por ejemplo), ni burocráticas (todos los niveles de la organización están de acuerdo con el cambio).

○ Estimación de Costos:

Luego de decidir que el proyecto es viable, se debe llevar a cabo una estimación temprana de costos. Cuando se contrata un servicio de computación en la Nube, el oferente proporciona una herramienta de análisis de

presupuestos. Por ejemplo, tanto la herramienta Windows Azure TCO de Microsoft [20] como la RDS Cost Comparison Calculator de Amazon [21] permiten estimar los costos operacionales de los servicios que ofrecen las empresas mencionadas y determinar los beneficios económicos generados debido a la diferencia entre el mantenimiento de un centro de datos privado y los precios de los servicios en la Nube. Estas herramientas se basan en el análisis de tres grandes factores que define el usuario: el ancho de banda a consumir, la cantidad de horas de uso de CPU´s, y el espacio de almacenamiento en discos.

Por lo tanto, la mayor parte de la literatura especializada ofrece algún tipo de métrica que permite cuantificar los costos de llevar a cabo un proceso de migración de datos a la Nube. En [2], por ejemplo, se propone la ecuación (1) que permite comparar los costos de un Data Center propio contra uno implementado en la Nube.

$$HrUsrCloud \times (I - CoCloud) \geq HrUsrDC \times (I - \frac{CoDC}{Util})$$

Ecuación 1. Comparación Costos

Siendo:

HrUsrCloud = Horas de usuario que utiliza el servicio en la Nube.

I = Ingreso esperado, resultado de implementar una tecnología en el negocio, comparado con la no implementación de la misma.

CoCloud = Costos anuales del servicio de virtualización Cloud Computing (precio del mercado promedio).

HrUsrDC = Horas de usuario que utiliza el servicio del Data Center.

CoDC = Costos de mantenimiento del Datacenter. Incluyen, pago a empleados, servicios de actualización y reparación, refrigeración y mantenimiento del ambiente climatizado, pago a empleado/s clasificados, entre otras cosas.

Util = Utilización (en horas de uso) del hardware, ya que no en todo momento se hace usufructo de la misma forma. Esto permite visualizar un promedio de los picos de demanda, en el tráfico de datos.

En la parte izquierda de la fórmula (1) se obtiene como resultado la ganancia esperada del uso de Cloud Computing (considerando el ingreso obtenido, el costo del servicio y las horas utilizadas); mientras que en la parte derecha se desarrolla el mismo cálculo para un centro de datos de una determinada capacidad (factorizándolo por el promedio de utilización para incluir los períodos en los que la carga de trabajo no presenta picos). Entonces si el valor del lado derecho (costo del centro de datos) es mayor valor al izquierdo (costo de uso de la Nube), significa que existe la oportunidad de generar mayores beneficios adoptando dicha tecnología.

Aunque la métrica analizada ofrece una aproximación general de la Tasa de Retorno de la Inversión, deja de lado las características propias de cada proveedor del servicio.

Además, en la fórmula anterior no se consideran los costos de formación de usuarios y los costos operativos propiamente dichos. Por lo tanto [14] agrega estas variables y propone la ecuación (2) para estimar el costo de adquisición de Software como Servicio (SaaS).

$$CoTotal = N \times CoCloud + CoOp + CUsr$$

Ecuación 2. Costo de Adquisición de SaaS

Siendo:

CoTotal = Costo de adquisición total del SaaS.

N = Cantidad de Usuarios

CoOp = Costos Operativos. Que involucran: migración de datos propiamente dicha, certificaciones y auditorías que permiten llegar a un nivel determinado en cuanto a la normalización de los datos persistidos.

CUsr = Costos de formación de usuarios. Que involucran: capacitación para la instalación, uso, y mantenimiento diario del servicio. También incluyen aquellos referidos a la formación para el manejo de las plataformas particulares de cada proveedor.

Esta última ecuación toma en cuenta factores que la primera deja de lado, pero no permite comparar con los costos de no implementar Cloud Computing.

En conclusión, es necesario aplicar ambas ecuaciones para determinar la conveniencia económica del proyecto. Con respecto a la ecuación (1), si el miembro en el que intervienen las variables del Data Center dan valores mayores (ingresos mayores), no pasará el análisis económico. Así como tampoco será conveniente, si el costo total de adquisición de SaaS, resultado de aplicar la ecuación (2), es mucho mayor que el ingreso obtenido en la ecuación (1).

3.4 Definición de un SLA genérico

Llegada esta etapa, la organización interesada debería poder determinar si es conveniente para ella la adopción de una tecnología Cloud Computing en general, y la adquisición de SaaS en particular. Si cumple con los requisitos propuestos en las secciones 3.1 y 3.2, el estudio de viabilidad de la sección 3.3 da positivo y la estimación económica temprana de la sección 3.4 arroja resultados favorables, la migración de datos puede ser llevada a cabo. Si cualquiera de dichas evaluaciones o métricas dice lo contrario, es conveniente volver a analizar la situación actual de la organización y plantearse una posible reestructuración que contemple las características y condiciones propuestas en este trabajo, para que las métricas del análisis TEO resulten positivas.

Pero para aquellas que llegaron hasta esta etapa con un horizonte que favorece la adopción de computación en la Nube, se recomienda altamente la realización de un documento donde se dejen plasmados los requisitos que

la PyME tendrá con cualquier proveedor. Estos requisitos se necesitan conocer antes de contratar a cualquier proveedor de Cloud Computing. Es decir, se debe confeccionar un documento semejante a un Acuerdo del Nivel de Servicios (en inglés, Service Level Agreement o SLA), pero genérico, es decir que no dependa de ningún proveedor en particular. Por lo tanto esta etapa se encuentra antes que la selección del proveedor, ya que la PyME deberá dejar asentados sus requisitos técnicos, operativos y económicos.

Las empresas de mayor tamaño suelen poseer más experiencia en la documentación y protocolos requeridos de este tipo de proceso. Como las PyMEs, proporcionalmente, suelen derivar menos funciones a terceros, es necesario realizar un análisis riguroso del contenido del SLA [22]. El SLA es el documento que describe la relación entre el proveedor y el usuario. Esta descripción pretende definir objetiva y cuantitativamente las condiciones del servicio. Es esencial a la hora de interactuar con un proveedor por contener una definición completa y precisa sobre cada servicio brindado y las responsabilidades de cada parte.

Entre los factores que el consumidor necesita considerar, en [23] se destacan ciertos aspectos contractuales:

- *Definición del Nivel de Objetivos del Negocio*: el usuario debería saber por qué quiere migrar su información, antes de definir qué migrar. Se deben elegir proveedores y servicios en base a los objetivos de la organización.
- *Definición de Responsabilidades tanto del proveedor como del consumidor*: las responsabilidades del consumidor deben incluir las restricciones del tipo de información que puede a ser alojada, límites en el uso del sistema, o simplemente, tener una licencia válida de cualquier aplicación que corre en los servidores del proveedor. En cambio, el proveedor debe indicar el porcentaje de tiempo que el servicio estará disponible, las compensaciones que brindará ante fallas en el rendimiento del servicio, las políticas de preservación de datos ante eventos que conlleven a la terminación del servicio, y la responsabilidad de custodiar legalmente la información de los clientes.
- *Definición de Limitaciones*: el proveedor debe restringir su responsabilidad ante las fallas ocasionadas durante los cortes programados de servicios, las violaciones de seguridad que no son abarcadas en las políticas de protección brindadas por el proveedor, asimismo se reservan el derecho de cambiar el acuerdo de servicio y modificar los servicios, notificando al consumidor en un tiempo determinado.
- *Definición de Obligaciones*: se exige a los consumidores de abstenerse de almacenar contenidos ilegales en los componentes de hardware del proveedor, respetar las licencias del software y comprometerse a cumplir con los pagos de los servicios consumidos.

Puesto que no existe un estándar para la presentación de SLA para Cloud Computing, cada proveedor utiliza su propio criterio para la redacción de estos acuerdos y como consecuencia el proceso de evaluación de acuerdos de servicios es una tarea compleja para los usuarios.

En busca de estandarización, algunas investigaciones han centrado su atención en los protocolos existentes basados en Arquitectura basada en Servicio (SOA) [24], por ejemplo el Web Service Level Agreement (WSLA), Web Services Agreement Specification (WS-Agreement) [25]. Como estos estándares fueron creados específicamente para SOA, no soportan los modelos de costos y no son aplicables a todos los modelos de servicios de computación en la Nube.

3.5 Selección del Proveedor

Finalmente, se debe realizar el análisis de infraestructura y los modelos de arquitectura que cada proveedor ofrece. Con cada tipo de entrega de servicios, se debe estudiar el modelo de costos del proveedor. A tal efecto, se debe tener en cuenta los tres grandes modelos de precios que se utilizan actualmente en el mercado.

- *Precio por nivel*: el precio es fijado según el nivel de abstracción de servicio y cada nivel posee sus propias especificaciones.
- *Precio por unidad*: generalmente se refiere al precio de la transferencia a través de la red, el procesamiento de datos y la utilización de la memoria disponible.
- *Precio por subscripción*: los usuarios pagan un importe periódicamente que les permite utilizar los servicios del proveedor, sin la necesidad de llevar la contabilidad de lo que realmente han utilizado.

Además, se debe tener en cuenta el tipo de virtualización que las organizaciones proveen. Espino Barrios [15] distingue las siguientes con sus respectivas características:

- *Computación Virtualizada y Distribuida*: ayuda a proveer un escenario de ejecución para aplicaciones en el tope de infraestructuras interconectadas. Este sistema distribuido utiliza máquinas virtuales como bloques para la construcción de escenarios de ejecución. Además, el ambiente de ejecución es una red de máquinas virtuales creadas para satisfacer los requerimientos de cierta aplicación, de esta forma se ejecuta aisladamente de otros ambientes.

- *Fábrica de Servicios de Negocios Virtuales (VBSF)*: como un modelo para virtualizaciones heterogéneas y abstracción de servicios, de recursos y de infraestructura. Este modelo es aplicable a servicios y ambientes de infraestructura que trascienden entre Private y Public Clouds.
- *Contrato de Máquina Virtual (VMC)*: modelo basado en contratos, que propone una plataforma independiente que automatiza las comunicaciones y la administración de ciertos requerimientos. Consiste en una extensión del Open Virtual Machine Format (OVF) [26].
- *Diverter*: el cual es un modelo basado en red para infraestructuras virtualizadas. Este modelo busca alta flexibilidad, larga escalabilidad con eficiencia, sistemas de conexión distribuidos.

Por otra parte, entre los factores a considerar para elegir un proveedor de servicios no se pueden dejar de lado las cuestiones de seguridad. Uno de los principales desafíos de Cloud Computing es la seguridad, debido a que los datos son los activos más valiosos de las organizaciones. Se debe elegir un proveedor que logre cumplir, por lo menos, con los parámetros listados a continuación:

- Cifrar los datos para evitar posibles penetraciones de intrusos en el sistema.
- Cifrar los datos en tránsito, asumiendo que los datos pasarán por una red pública.
- Requerir autenticación entre aplicaciones.
- Poner atención a la criptografía y estar actualizados en algoritmos de cifrado.
- Manejar de una manera segura los accesos de los usuarios.

Finalmente, el almacenamiento que es esencial en sistemas distribuidos y en Cloud Computing, y donde se propone un esquema distribuido que es flexible y efectivo, que tiene soporte dinámico de datos incluyendo agregado, actualizado y borrado. Se centra en la verificación de datos a través de un integrado sistema de almacenamiento garantizando la identificación del servidor que este causando problemas de comportamiento.

4 Caso de Estudio

Para ilustrar la implementación del proceso propuesto en este trabajo, se utilizan dos donde se describe los pasos y problemas encontrados al realizar la migración a arquitecturas Cloud Computing propuesto. El primer caso descripto en [27] para un proyecto finalizado exitosamente (sección 4.1); mientras que el segundo corresponde a un ejemplo planteado en [2] sobre las características que pueden llevar al fracaso de un proyecto de migración (sección 4.2).

4.1 Caso finalizado Exitosamente

El primer caso descripto corresponde al proyecto real de migración de información a Cloud Computing realizado por el Instituto Argentino de Responsabilidad Social Empresaria (IARSE) y que se encuentra documentado en [27]. Aunque este caso fue estudiado también utilizado en [4], en este nuevo trabajo se agrega un análisis más detallado de cada etapa propuesto.

El objetivo que perseguía el IARSE era afrontar en tiempo y forma la comunicación entre sus empresas miembros [28]. La mayor parte de ellas son pequeñas y medianas empresas, pero sus volúmenes de información varían dependiendo de tendencias estacionales y otros factores. Entre las posibles soluciones se encontraba la adopción de SaaS, o la compra de una aplicación y la consecuente compra de hardware para almacenar el tráfico de datos que la misma generaría.

Teniendo en cuenta la ubicación geográfica, la infraestructura de hardware de la organización en ese momento (como lo indica la etapa 3.1) y que los usuarios de la aplicación a implementar se encuentran a lo largo de toda la Argentina, se decidió analizar la adopción de Cloud Computing, ya que se buscaba una distribución rápida y bajo demanda de los servicios. Un punto clave en esta primera aproximación al dominio, es que tanto el IARSE como sus empresas socias, contaban con certificaciones que garantizaban al menos un mínimo grado de estandarización en sus procedimientos y sobre la información que generaban y recibían. Este es un punto positivo según lo mencionado en la sección 3.1.

Debido a que la organización no contaba con una aplicación propia para compartir información entre los usuarios, se llegó a la conclusión que debía adquirir una externa. La misma trataría de una plataforma colaborativa de intercambio de información en tiempo real. Por lo que los datos que administraría no eran de alto riesgo, y podrían ser alojados en servidores públicos. Entonces, se logró detectar la necesidad de alojar una aplicación con las características definidas en la sección 3.2 de este trabajo

A través del cálculo de costos y, antes de elegir un proveedor determinado, se estimaron los costos de forma temprana. La aplicación de la ecuación (1) presentada en la sección 3.3 hubiera arrojado valores mayores para el miembro izquierdo de la ecuación; es decir, era económicamente más conveniente la adopción de un entorno basado en la Nube. Como resultado, se identificó que la migración a servidores virtualizados representaba, aproximadamente, un 200% de ahorro con respecto a la compra de hardware propio. No se contaba con personal especializado en el uso de esta nueva tecnología, por lo

que la ecuación (2) de la sección 3.3 aumenta el riesgo de migración, pero esto no impide que se lleve a cabo, solo aumenta los costos de implementación.

Los directivos de IARSE decidieron elaborar una lista de servicios y funcionalidades que la aplicación y el oferente de servicios debía brindar; lo cual se asemeja a la creación del modelo genérico del nivel de servicios propuesto en la etapa 3.4.

Por último, luego del análisis de varias alternativas en base al costo, el valor agregado por servicios y los modelos de arquitectura ofrecidos por los distintos proveedores, se decidió la adopción de la tecnología LotusLive de IBM [29]. LotusLive es una herramienta de software de colaboración en modalidad Cloud Computing que le permite a IARSE utilizar aplicaciones para Comunidad, Web Meetings y Foros de Discusión. Como la misma está alojada en bases de datos externas, no requiere gastos adicionales en lo que respecta a su mantenimiento. Esta solución resuelve la problemática de comunicación de la organización, basándose en SaaS y la virtualización de servidores. Es decir que el servicio ofrecido encajó perfectamente en una de las clasificaciones propuestas en la sección 3.5 sobre el tipo de virtualización ofrecido, específicamente, se trata de la categoría 'Computación Virtualizada y Distribuida'.

Actualmente, IARSE sigue utilizando la aplicación después de 3 años de su implementación y de la adopción de Cloud Computing.

Vale aclarar que, en el caso descripto anteriormente, el proceso no se implementa de forma rigurosa. Por ejemplo, no se formalizó la documentación referida a la estandarización de procedimientos. Además, al no disponer de recursos humanos especializados en el uso de la herramienta adquirida, los costos de capacitación se elevan y eso aumenta el riesgo a la hora de evaluar la migración a la Nube. A pesar de esto, este caso de éxito enfocó su estrategia de adopción de una manera muy similar a la que se propone en este trabajo.

4.2 Caso con Fracaso

El objetivo de este segundo caso es demostrar la utilidad del análisis técnico y operativo que se describe en este trabajo mediante adaptación del ejemplo propuesto por [2]. En dicho ejemplo se observa cuán importante es analizar los factores operativos (volúmenes de información producida y velocidad de transferencia, entre otros), para determinar la viabilidad de un proyecto de este carácter. En el ejemplo se plantea el estudio de la migración a la Nube de un laboratorio biológico. Para ello, la organización pretende evaluar si es conveniente o no la adopción de Cloud Computing, tomando como prioridad la velocidad con que se obtendrán los resultados.

Para esta situación ficticia, se determinan los parámetros de comparación en base a los servicios prestados por el proveedor Amazon. En este sentido, la velocidad de procesamiento se compara con la de las máquinas EC2 de Amazon Web Services (AWS).

La descripción establece que, el laboratorio genera 500 GB de nueva información por cada experimento que lleva a cabo en sus instalaciones. Además, cuenta con el equivalente a 20 instancias locales de una EC2, anteriormente mencionada. Cada instancia de una EC2 toma 2 horas por GB para procesar la información. Entonces, el tiempo para evaluar el experimento será de $500 \times 2 / 20$ es decir, que le toma 50 horas de procesamiento de forma local, en el caso de no optar por la tecnología Cloud Computing.

Otra alternativa podría haber sido procesar los 500 GB de datos en solo una hora, con 1.000 instancias de EC2 gracias a los servicios que AWS ofrece. En esta adaptación del ejemplo se deja de lado el cálculo de los costos debido a que sólo se pretende mostrar las limitaciones que las variables operativas (velocidades de transferencia y procesamiento) pueden presentar a la hora de elegir un proveedor en la Nube.

En este ejemplo se considera una velocidad de transferencia entre el laboratorio y AWS promedio de 20 Mbits/segundo. Entonces, el tiempo de transferencia total se calcula de la siguiente manera:

a) Se convierten las unidades de GB a Mbits mediante la multiplicación:

$$500 \text{ GB} \times 1000 \text{ MB/ GB} \times 8 \text{ bits/Byte}$$

para obtener un valor de 4.000.000 Mbits.

b) Al valor anterior se lo divide por la velocidad obteniendo 200.000 segundos (es decir más de 55 horas) de tiempo en transferencia.

c) A este último resultado, se le debe adicionar una hora más, que es lo que tarda en procesarse la información en las 1.000 instancias de EC2.

De esta manera, en este caso realizar el proceso en forma local con hardware propio insumiría aproximadamente 50 horas; mientas que realizarlo con AWS (con 1.000 instancias de EC2) llevaría unas 56 horas debido al tiempo de transferencia de la información. Como consecuencia de estos resultados, se afirmar que para este laboratorio, que prioriza el tiempo para obtener los resultados, no es conveniente migrar a una arquitectura Cloud Computing.

Por otra parte, se considera de interés destacar que este ejemplo no sólo demuestra las evaluaciones del análisis técnico y operativo (realizado en la evaluación de la viabilidad de la etapa 3 del proceso propuesto), sino que también incluye aspectos de la integración de la arquitectura actual con el entorno Cloud Computing

(etapa 1). Es en esta primer etapa donde se debe evaluar si el entorno donde la PyME opera facilita o no la migración a la Nube. Como se puede notar en este ejemplo, si las velocidades de banda ancha no son suficientes para transferir los volúmenes de información necesarios, no vale la pena continuar con el proceso de migración a esta tecnología.

Finalmente, como conclusión, se desprende que es fundamental medir, comparar y evaluar los factores técnicos y operativos, antes de definir al proyecto como viable y pasar a un análisis económico concreto.

5 Conclusiones

En el presente trabajo se ha logrado desarrollar un proceso que permite evaluar las condiciones y características de una PyME para realizar la migración de sus datos, y ciertas de sus aplicaciones, a un modelo basado en la Nube. Asimismo, se proponen procedimientos para asistir la toma de decisiones, teniendo en cuenta aspectos técnicos, económicos y organizacionales. Para ello, se presentan cinco etapas que consideran no sólo los aspectos técnicos o económicos sino que se realiza un análisis integral de la estructura organizacional. Como se puede observar, el estudio de viabilidad y la estimación económica representan sólo uno de los factores que hay que analizar antes de implementar Cloud Computing. Es indispensable tener en cuenta las características propias de las PyMEs, la problemática y contexto propio de cada una para llevar a cabo un estudio más maduro si se piensa adoptar este nuevo paradigma.

Por último, se ha analizado dos casos de adopción de esta tecnología donde se puede observar que uno ha llegado al éxito del proyecto por haber aplicado etapas similares a las propuestas en este proceso; mientras que el otro ha fracasado por no hacerlo.

Como futuras líneas se trabajará en el desarrollo y la validación de la primera parte del análisis TEO, es decir del modelo de viabilidad completo. Además, se buscarán más casos de estudio u otras técnicas para validar el proceso presentado en este trabajo.

6 Referencias

[1] Mell, P. & Grance, T. The NIST Definition of Cloud Computing. Special Publication, National Institute of Standards and Technology, 2011.

[2] Armbrust, M. & Fox, A. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.

[3] Khajeh-Hosseini, A., Greenwood, D., Smith, J., Sommerville, I. The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise, 2010.

[4] Bernal, L., Vegega, C., Pytel, P., Pollo-Cattaneo, M. F.. Propuesta de una Metodología para el Análisis de Adopción de Cloud Computing en PyMEs. Proceedings XIX Congreso Argentino de Ciencias de la Computación, 2013, ID 5731. ISBN 978-987-23963-1-2.

[5] Pressman, R. Software Engineering: A Practitioner's Approach. Editorial Mc Graw Hill. 2004.

[6] Sapag, N. Preparación y evaluación de proyectos. 4ª Edición. McGraw Hill. 2000.

[7] Cardozo, E., Velásquez de Naime, Y., Rodríguez Monroy,C. El concepto y la clasificación de PyME en América Latina. En: Global Conference on Business and Finance Proceedings, 2012.

[8] Fundación Observatorio PyME Informe Especial: Definiciones de PyME en Argentina y el resto del mundo, 2013. http://goo.gl/0VrU1

[9] Políticas de apoyo a las micro, pequeñas y medianas empresas del MERCOSUR – Etapa II. En: XXXII GMC, MERCOSUR/GMC/RES. N° 59/98 (1998)

[10] Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M. A Break in the Clouds: Towards a Cloud Definition. Nota Editorial, ACM SIGCOMM Computer Communication Review, 2009, Vol. 39, N. 1.

[11] VMware, Inc. El momento adecuado para adoptar la virtualización de escritorios: Siete indicadores clave, 2013. http://goo.gl/W9Qxd

[12] Fourcade, G. Seis variables para analizar antes de saltar a la nube. IBM, (S/A) http://goo.gl/nXotK

[13] Microsoft Corporation The Economics of the Cloud, S/A. http://goo.gl/B1FpD

[14] Ercolani, G. Análisis del potencial del Cloud Computing para la PYMES. // Cuadernos de Gestión de Información, 2012 p40-55.ISSN 2253-8429

[15] Espino Barrios, L. Cloud Computing como una red de servicios. Reporte Técnico, Instituto Tecnológico de Costa Rica, 2009.

[16] Orlandi, P. Las PyMEs y su rol en el Comercio Internacional. White Paper Series del Centro de Estudios para el Desarrollo Exportador – CEDEX, S/A.

[17] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. J Network Comput Appl (2010), doi:10.1016/j.jnca.2010.07.006

[18] P Pytel, H Amatriain, P Britos, R García Martínez. Estudio del Modelo para Evaluar la Viabilidad de Proyectos de Explotación de Información - Proceedings IX Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. Lima, Perú - 2012.

[19] Jang, J.S.R. Fuzzy inference systems. Upper Saddle River, NJ: Prentice-Hall, 1997.

[20] Microsoft Corporation Plataforma Windows Azure: TCO and ROI Calculator, S/A http://goo.gl/26Apo

[21] Amazon Web Services, RDS Cost Comparison Calculator, S/A http://goo.gl/pFiWU

[22] Cloud Computing Use Case Discussion Group, Cloud Computing Use Cases – Version 4.0. White Paper, S/A.

[23] Zalazar, A. S., Gonnet, S., Leone, H. Aspectos Contractuales de Cloud Computing. Reporte Técnico, INGAR (UTN-CONICET), 2013.

[24] Tsai, W. T., Sun, X., & Balasooriya, J. Service-oriented cloud computing architecture. In Information Technology: New Generations (ITNG), Seventh International Conference on (pp. 684-689). IEEE, 2010, April.

[25] Keller, A., & Ludwig, H. The WSLA framework: Specifying and monitoring service level agreements for web services. Journal of Network and Systems Management, 2003, 11(1), 57-81.

[26] A Vouk, M. Cloud computing–issues, research and implementations. CIT. Journal of Computing and Information Technology, 2008, 16(4), 235-246.

[27] IBM IARSE optimiza su comunicación y amplía el debate sobre RSE en Argentina a través de una plataforma colaborativa cloud de IBM, S/A. http://goo.gl/DXCCH

[28] Rolando De Serra, A. IARSE - Construyendo un nuevo horizonte de RSE en Argentina (c.01 de lo periférico a lo estratégico), 2012. http://goo.gl/EOgrN

[29] IBM Software como servicio: LotusLive, S/A. http://goo.gl/5Bgop

Datos de Contacto:

Bernal Tomadoni, Luciano. *GEMIS. Universidad Tecnológica Nacional, Regional Buenos Aires. Medrano 951 (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina.*
bernal.luciano@gmail.com

María Florencia Pollo-Cattaneo. *GEMIS. Universidad Tecnológica Nacional, Regional Buenos Aires. Medrano 951 (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina.*
fpollo@posgrado.frba.utn.edu.ar

Pablo Pytel. *GEMIS. Universidad Tecnológica Nacional, Regional Buenos Aires. Medrano 951 (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina. ppytel@gmail.com*

SPRABI: Una Herramienta para la Planificación, Control y Evaluación de Proyectos

Edgardo Bernardis, Germán Montejano, Luis Roqué, Hernán Bernardis, Mario Berón
Facultad de Ciencias Físico-Matemáticas y Naturales
Universidad Nacional de San Luis
Ejército de los Andes 950, San Luis, Argentina
{ebernardis, gmonte, hbernardis, mberon}@unsl.edu.ar, araroq@yahoo.com

Resumen

Hoy en día se manejan extensas cantidades de información, siendo esto un problema complejo al momento de la toma de decisiones por parte de personas o empresas. Para dicho fin, existen diferentes metodologías y herramientas que ayudan a la hora de manejar grandes cantidades de información que posibilitan extraer los aspectos más importantes que permiten tomar decisiones respecto del plan que se lleva a cabo. Dichas metodologías y herramientas se basan en controlar los tiempos y recursos para la gestión del plan; siendo un factor restrictivo a la hora de la toma de decisiones. Hay un gran número de elementos, tanto internos como externos, que influyen y afectan el desempeño de un proyecto, por tal motivo, se hace necesario incluirlos y controlarlos a fin de mejorar la gestión y la toma de decisiones en los proyectos. Por otro lado, las herramientas existentes no son lo suficientemente flexibles al momento de controlar un proyecto a través de indicadores y, además, que permitan realizar las acciones correctivas pertinentes en la ejecución de dicho proyecto. Por esta razón, se desarrolló un modelo y su correspondiente prototipo, funcionando operativamente, que subsane estas restricciones. En este artículo se presenta SPRABI, una herramienta que permite controlar y administrar el avance de cualquier tipo de proyecto mediante el uso de indicadores, y en base a estos, sugerir las acciones correctivas correspondientes según sea el caso.

1. Introducción

En los últimos años, se ha observado que los problemas en los proyectos de las organizaciones (empresas, el estado, ONGs empresariales, etc.) estaban dados, en la mayoría de los casos, por una creencia errónea de que lo único que hace falta para tener éxito es contar con una estrategia correcta, cuando en realidad el problema no es una mala estrategia sino la mala aplicación de la misma.

La administración y control del avance de proyectos es de máxima utilidad en aquellas organizaciones que quieran definir una estrategia y llevarla a cabo con un seguimiento sistemático.

El control de gestión requiere de una interpretación amplia en torno a los elementos básicos que intervienen en el proceso, por un lado se tiene que el control parte desde un objetivo definido dentro de parámetros de alcance de logros, y por el otro, el control exige técnicas específicas para llevarlo a cabo de una manera efectiva dentro de un contexto organizacional concreto. Lo primero lo define un determinado marco de planificación y lo segundo la sistematización operativa del control [8, 20].

La planificación estratégica como sistema, contiene un determinado número de procesos que hacen de ella una actividad dinámica, flexible y continua. Esta complejidad no viene dada por las dificultades propias del proceso en torno a la aplicación de sus métodos y técnicas, sino más bien corresponde a la dinámica de la realidad que desborda los más elaborados análisis del entorno y hacen de ésta, un instrumento racionalizador de las aspiraciones organizacionales en un mundo de incertidumbres constantes.

La planificación estratégica asume un rol importante en la construcción de escenarios factibles dentro de cierta certidumbre esperada. Para ello, establece un conjunto de procesos integrados capaces de sistematizar acciones que permitan lograr dentro de un contexto específico, objetivos y metas productivas.

En la actualidad, resulta crítico la toma de decisiones basadas en información relevante, exacta y actual. Dada la vasta cantidad de información que se genera hoy en día, es necesario contar con una herramienta que permita manipular tal cantidad de información y que la presente en una forma sintetizada y con extrema facilidad de uso.

Si bien los Sistemas de Información existen desde hace

varias décadas, su incursión en las organizaciones, desde el punto de vista estratégico, y su subsecuente establecimiento como ventaja competitiva en las mismas es algo más reciente. Los Sistemas de Información se usan para recoger y almacenar información, y para producir reportes específicos para los distintos integrantes y directivos de la organización. Sin embargo, los directivos sub-utilizan estos sistemas directamente y piensan que la información les resulta de poco provecho si no tienen la posibilidad de explorar detalles ocultos [18].

A la fecha, existen diferentes técnicas para el control de tiempos y recursos (Pert, Gantt, etc.); metodologías (PMBOK [8], BSC [15, 14, 13], CMMI [6, 22, 5], ISO 9001:2008 [9, 24, 1], RUP [10, 23, 17], etc.) e incluso herramientas de software para controlar y administrar proyectos. Todas ellas presentan ciertas deficiencias a la hora de satisfacer plenamente las necesidades de medición que se requieren a la hora de administrar y controlar un proyecto. Son complejas de utilizar, poco flexibles, no incluyen las partes interesadas (stakeholders), tanto internas como externas, etc.

El artículo está organizado como sigue. La sección 1 introduce en la temática sobre administración de proyectos y luego plantea el problema que se aborda. La sección 2 expone el marco teórico junto con definiciones claves respecto de la temática planteada. Como paso siguiente se profundiza en el concepto de administración de proyectos y las partes asociadas que derivan del mismo. Se continúa con la importancia y el rol fundamental que cumplen los indicadores en la administración de proyectos. Por último, se habla sobre el modelo de la estructura que se plantea para dar solución al problema planteado. En la sección 3 se especifican los aspectos involucrados en el desarrollo de *SPRABI* (System for Project Administration by Indicators), desde el punto de vista arquitectural. La sección 4 muestra el uso y desempeño de *SPRABI* en la práctica. Para esto, se utiliza un caso de estudio que surge del dominio de una empresa privada proveedora de servicios públicos de energía. Finalmente, la sección 5 presenta las conclusiones de este artículo.

2. Marco Teórico

Cuando una organización pretende lograr ciertos objetivos específicos, necesita organizar ciertas actividades y tareas para lograrlo; para lo cual, define un proyecto. Un proyecto [8, 20] es *un conjunto de actividades interrelacionadas y coordinadas, que insumen recursos y son desarrolladas en un lapso de tiempo por una entidad para alcanzar un objetivo*.

Habitualmente son muchos y variados los stakeholders (interesados) en los diferentes aspectos de un proyecto, razón por la cual es necesario identificarlos e identificar sus intereses y/o metas. Comprender el concepto de Stakeholder suele ser complejo [3, 7], pero resumiendo se puede decir que un Stakeholder puede ser considerado como *cualquier individuo o grupo que puede afectar o es afectado por las acciones, decisiones, políticas, prácticas o metas de la organización*. Esta definición es muy amplia, pero en este amplio concepto, la organización o el que toma las decisiones, tiene más probabilidades de explorar sus responsabilidades sociales y éticas completamente, que cuando se utiliza una definición más restringida.

Por otro lado, estos stakeholders necesitan monitorear continuamente la evolución del proyecto mediante indicadores, con el objetivo de evaluar y detectar desviaciones en el proyecto que afecten a sus intereses y poder recomendar las correcciones adecuadas. Todo el monitoreo y control que se hace sobre un proyecto hace a la administración de proyectos.

2.1. La Administración de Proyectos

Hay empresas buenas y malas y todas ellas luchan contra proyectos que, dadas sus características, son simples y otros una pesadilla [21, 16]. Sin embargo, si se hiciera una autopsia de cada proyecto, sería muy probable que se encuentre un denominador común constante: una débil gestión.

La dirección de proyectos es la *aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto con el fin de culminar todo el trabajo requerido para desarrollarlo y cumplir con el/los objetivos, dentro de los límites de tiempo y costos definidos* [18]. Todo esto implica realizar acciones como planear, organizar, asegurar, administrar, dirigir y controlar los recursos.

La dirección de un proyecto incluye:

- Identificar los requisitos
- Establecer unos objetivos claros y posibles de realizar
- Equilibrar las demandas concurrentes de calidad, alcance, tiempo y costes
- Adaptar las especificaciones, los planes y el enfoque a las diversas inquietudes y expectativas de los diferentes interesados.

La gestión de un proyecto comienza con un conjunto de actividades que globalmente se denominan planificación del proyecto. Antes de que el proyecto comience, se debe realizar una estimación del trabajo a realizar, de los recursos necesarios y del tiempo que transcurrirá desde el comienzo hasta el final de su realización. Siempre que se estima, se está mirando hacia el futuro y aceptando resignado cierto grado de incertidumbre. Aunque la estimación es más un arte que una ciencia, es una actividad importante que no debe llevarse a cabo de forma descuidada.

La estimación comienza con una descripción del ámbito del producto o servicio. Hasta que no se delimite el ámbito no es posible realizar una estimación con sentido. Al inicio de un proyecto se requieren estimaciones cuantitativas y un plan organizado, pero no se dispone de información sólida. Un análisis detallado de los requisitos proporcionaría la información necesaria para las estimaciones, pero el análisis a menudo lleva semanas o meses. Aún peor, los requisitos pueden ser fluidos, cambiando regularmente a medida que progresa el proyecto. Y, aún así, se necesita un plan de manera inmediata. Por lo tanto, se debe examinar el problema a resolver al inicio del proyecto y por lo menos establecer el ámbito y delimitarlo.

Para desarrollar un plan de proyecto razonable, se tiene que descomponer funcionalmente el problema a resolver. Los seres humanos tienden a aplicar la estrategia de divide y vencerás cuando se enfrentan a problemas complejos. Dicho de manera sencilla, un problema complejo se parte en problemas más pequeños que resultan más manejables. Ésta es la estrategia que se aplica al inicio de la planificación del proyecto.

Es necesario decidir qué modelo de proceso es el más adecuado para el proyecto. Cuando se selecciona un modelo de proceso, se define entonces un plan de proyecto preliminar basado en un conjunto de actividades estructurales. Una vez establecido el plan preliminar, empieza la descomposición del proceso. Es decir, se debe crear un plan completo reflejando las tareas requeridas para cubrir las actividades.

Una vez que se tiene el plan y se comienza el desarrollo del proyecto, es necesario hacer un seguimiento del mismo para evaluar la ejecución real del proyecto respecto de la planificada, realizar mediciones, reunir y analizar métricas del proyecto. Este seguimiento del progreso mientras se está llevando a cabo el trabajo permite tomar decisiones inteligentes y una vez finalizado el proyecto, realizar un análisis *Postmortem* lo que servirá para extraer sabias lecciones.

La medición nos permite tener una visión más profunda proporcionando un mecanismo para la evaluación objetiva. Lord Kelvin en una ocasión dijo: “Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente, puede ser el comienzo del conocimiento, pero en sus pensamientos, apenas está avanzando hacia el escenario de la ciencia”.

Las métricas del producto son una medida cuantitativa que permite tener una visión profunda de la eficacia del proceso y de los proyectos que dirigen utilizando el proceso como un framework (marco de trabajo). Se reúnen los datos básicos de calidad y productividad; estos datos son entonces analizados, comparados con promedios anteriores, y evaluados para determinar las mejoras en la calidad y productividad. Las métricas son también utilizadas para señalar áreas con problemas de manera que se puedan desarrollar las correcciones y mejorar el proceso.

La importancia de lo anterior radica en que, si no se mide, sólo se podrá juzgar basándose en una evaluación subjetiva. Mediante la medición, se pueden señalar las tendencias (buenas o malas), realizar mejores estimaciones, llevar a cabo una verdadera mejora sobre el tiempo.

Hay cuatro razones para medir los procesos, los productos y los recursos:

- Caracterizar: sirve para comprender mejor los procesos, los productos y servicios, los recursos y los entornos para establecer las líneas base para las comparaciones con evaluaciones futuras.
- Evaluar: posibilita determinar el estado con respecto al diseño. Las medidas utilizadas son los sensores que posibilitan conocer cuándo los proyectos y procesos están perdiendo la pista, de modo que se puedan poner bajo control. También se evalúa para valorar la consecución de los objetivos de calidad y para evaluar el impacto de la tecnología y las mejoras del proceso en los productos y procesos.
- Predecir: permite poder planificar. Realizar mediciones para la predicción implica aumentar la comprensión de las relaciones entre los procesos y los productos y la construcción de modelos de estas relaciones, por lo que los valores que se observan para algunos atributos pueden ser utilizados para predecir otros. Se hace esto porque se quiere establecer objetivos alcanzables para el coste, planificación y calidad, de manera que se puedan aplicar los recursos apropiados. Las medidas de predicción también son la base para la extrapolación de tendencias, con lo que la estimación para el coste, tiempo y calidad se puede actualizar basándose en la evidencia actual. Las proyecciones y las estimaciones basadas en datos históricos también ayudan a analizar riesgos y a realizar intercambios diseño/coste.
- Mejorar: se mide para mejorar cuando se recoge la información cuantitativa que ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

Se recopilan medidas y se desarrollan métricas para obtener indicadores. Un indicador es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso, del proyecto, o del producto o servicio en sí. Permite al gestor de proyectos tener una visión más amplia y tomar decisiones fundamentadas; permitiendo ajustar el producto y/o servicio, el proyecto o el proceso para que las cosas salgan mejor.

Los datos de las métricas que indican un área de problemas no se deben considerar negativas; estos datos son meramente indicadores de mejora de proceso. Las métricas e indicadores se utilizan para propósitos estratégicos.

2.2. El Rol de los Indicadores en la Evaluación de Proyectos

Un indicador se define como *la relación entre las variables cuantitativas o cualitativas, que permite observar la situación y las tendencias de cambio generadas en el objeto o fenómeno observado, respecto a los objetivos y metas previstas e influencias esperadas* [11].

Los indicadores empleados en proyectos son medidas específicas, explícitas y objetivamente verificables que buscan dar cuenta de los cambios que surgen a medida que el proyecto es ejecutado. Los Stakeholders del proyecto tienen diferentes intereses (metas) que, entre otras cosas, necesitan verificar de manera continua que el proyecto evoluciona de manera favorable a sus intereses; caso contrario, poder sugerir y realizar correcciones al detectar desviaciones; esto es posible mediante un conjunto de indicadores que controlan el grado de cumplimiento de objetivos y resultados.

Un indicador debe contemplar y tener en cuenta las siguientes fases para su elaboración:

1. Contar con objetivos y estrategias: Los objetivos deben cuantificarse para poder asignarle indicadores, generalmente un objetivo-meta cuenta con los siguientes patrones:
   - Poseer atributo (la meta).
   - Escala (unidad de medida).
   - Status (valor actual).
   - Umbral (valor a lograr).
   - Horizonte (período de logro).
   - Fecha de inicio (del horizonte).
   - Fecha de fin.
   - El responsable directo de llevar a cabo el logro de la meta.
2. Definir los indicadores para los factores claves de éxito: A cada factor de éxito se definirán los respectivos indicadores.
3. Determinar para cada indicador el estado, umbral y rango de gestión.
4. Diseñar la medición: Consiste en definir la fuente de información, la frecuencia de la medición, la presentación, la tabulación, el análisis y el responsable del proceso.
5. Medir, probar y ajustar el sistema de indicadores.

Una vez definidos los objetivos de cada perspectiva, se definen los indicadores para medir el éxito o fracaso del logro de los objetivos de un proyecto. Estos indicadores permiten medir la estrategia, con lo cual se logra:

- Confirmar las relaciones causa-efecto definidas entre objetivos: esto es importante ya que en el caso que dicha relación no se cumpla se debería replantear la estrategia para evitar errores posteriores.
- Confirmar o desaprobar la habilidad de la entidad de la organización para alcanzar lo que había planteado: esto permitirá realizar acciones correctivas en el momento adecuado para contribuir al éxito de la aplicación de la estrategia.
- Contribuir al compromiso de los integrantes con la estrategia: comunicar el éxito o fracaso del alcance de los objetivos, ayuda a enfocar a los integrantes de la organización con los temas que interesan.

Hay dos tipos de indicadores:

- Resultado: reflejan la eficiencia de la organización en el alcance de los objetivos.
- Desempeño: reflejan el progreso de las acciones que propician el logro de los objetivos.

Debe existir una relación armoniosa entre indicadores de resultado y desempeño para cada objetivo.

La definición de las características de los indicadores es vital para el éxito de la obtención de los objetivos. Un buen indicador debe:

- Ser operacional: enfocado a la acción y que suministre información útil para el control.
- Medirse contra un estándar de logro: definir los valores esperados para cada indicador.
- Ser confiable: que pueda conseguirse y evaluar fenómenos de tipo cualitativo y cuantitativo.
- Medirse fácilmente.

2.3. Modelo de la Estructura

Tal como se expresó anteriormente, las técnicas más conocidas para el control de proyectos se basan en el control de tiempos, y en casos de perfeccionamiento de éstas, usan el control de recursos; pero ninguna de ellas satisface plenamente las necesidades de medición que se requieren a la hora de administrar y controlar un proyecto. Es por ello, que

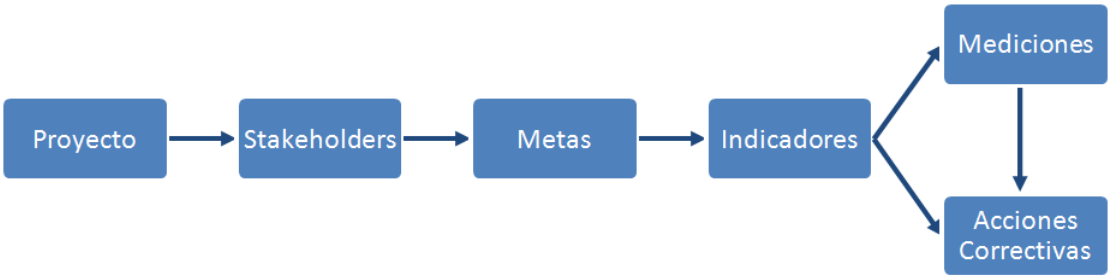


Figura 1. Estructura que utiliza la herramienta

se propone desarrollar un modelo con su posterior prototipo (sistema), que permita administrar y controlar proyectos de una manera simple e intuitiva. Dicho procedimiento estructurado y ordenado, posibilita realizar el seguimiento de los proyectos, su administración y en base a los resultados brindados por la misma, realizar las acciones correctivas, en caso de ser necesario, para alcanzar los objetivos planteados en los proyectos.

Para lograr lo planteado anteriormente, es necesario definir una estructura que soporte toda la información necesaria para realizar la administración y luego el seguimiento y control de proyectos. En este caso, la estructura definida tiene una relación jerárquica que se muestra en la Figura 1.

Esta organización, dada la generalidad que plantea, permite definir cualquier tipo de proyecto; los stakeholders necesarios, siendo estos de cualquier naturaleza, y controlar los objetivos a cumplir con la definición de indicadores que no se restringen a un entorno en particular.

Para poder controlar y hacer un seguimiento de lo que se planea en un principio para un proyecto, respecto de lo que realmente ocurre luego en la ejecución del mismo, es necesario dividir en dos etapas el modelo. La primera etapa se denomina *Plan de Proyecto* y comprende los pasos de planificar y construir la estrategia en base a la estructura planteada anteriormente. A partir de allí, comienza la etapa que se denomina *Ejecución del Proyecto*, la cual consiste en la evaluación sistemática del cumplimiento de las metas en base al control que hacen los indicadores en base a las mediciones que se obtienen en el avance o ejecución del proyecto.

En el modelo propuesto, la primera parte llamada Plan de Proyecto, responde a la pregunta: ¿Cómo debería llevarse a cabo la estrategia de manera tal que el proyecto sea exitoso?. En la segunda parte del modelo propuesto, llamado Ejecución del Proyecto, responde a la pregunta: ¿Los valores reales de los indicadores están reflejando que la estrategia se está llevando a cabo según lo planificado para que el proyecto sea exitoso? [18].

En este modelo; cada stakeholder, por ejemplo el $stakeholder_i$, puede ser una instancia de un cliente, un accionista, un empleado, un proveedor, alguien de la comunidad o, principalmente, una unidad de negocio o servicio soporte (compartido) de la organización. No necesariamente es una persona, sino que puede ser un equipo de personas trabajando en la misma dirección y produciendo sinergia para alcanzar una meta.

Respecto de los aspectos referidos a los indicadores; en el modelo, a cada indicador *j* se lo planifica con un valor ideal V_{ji} en un momento dado del avance de la ejecución del proyecto t_i, con un rango de aceptabilidad, es decir, valores de desviación d_{ji-k} y d_{ji+l}. En un momento dado t_i, el indicador brindará el valor real actual AV_{ji} obtenido y se evaluará si está dentro del rango de aceptación respecto de su valor ideal.

A partir de este valor, el stakeholder sabrá si necesita tomar acciones correctivas para ajustarse al valor deseado en la próxima medición del indicador o si estando en el camino correcto, pero viendo una desviación en más o en menos, le va dando una pista de que este indicador comienza a desviarse, es decir, si bien en el momento actual está dentro de los límites aceptables; por la tendencia que está teniendo, probablemente en las próximas mediciones quede fuera del rango de aceptación.

3. SPRABI

En esta sección se describe *SPRABI* (System for Project Administration by Indicators) una herramienta accesible desde la web cuyo principal objetivo es asistir al interesado en la administración de proyectos.

SPRABI hereda las características de los patrones arquitecturales JEE [12, 19] y MVC [2]. Es por ello que SPRABI está dividido arquitecturalmente en las siguientes grandes componentes:

- Capa Cliente
- Capa Web
- Capa de Negocio

- Capa de Servicios de Información

Cada una de estas capas (ver Figura 2) se explica en mayor detalle en los ítems siguientes.

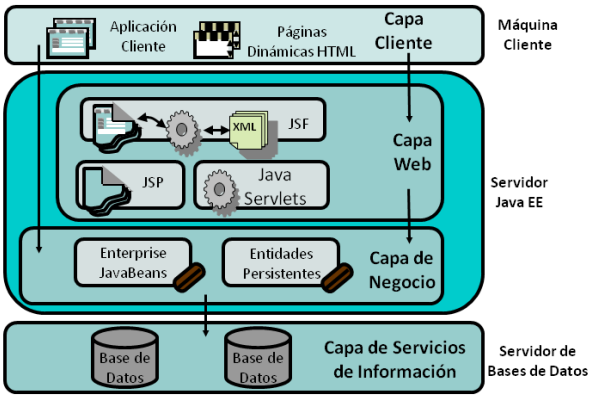


Figura 2. Arquitectura JEE

Capa Cliente: La mencionada capa está compuesta por la página web que interactúa con los usuarios. Se encuentra dividida lógicamente en tres partes. La primera corresponde al Plan de Proyecto, la cuál permite cargar toda la información necesaria que sirve para planificar y construir la estrategia del proyecto. La segunda corresponde a las Mediciones, es aquí donde se posibilita realizar las cargas de valores (mediciones) a los indicadores, lo que permite en la etapa de Ejecución del Proyecto, determinar cómo es el avance del mismo, respecto de lo que se planteo en la etapa del Plan del Proyecto. Por último, la Ejecución del Proyecto, muestra de diferentes formas (gráficas) la evaluación sistemática del cumplimiento de las metas según el control que hacen los indicadores en base a las mediciones que se obtienen en el avance o ejecución del proyecto. A si mismo, según corresponda, muestra las acciones correctivas a realizar en caso de que la evaluación de un indicador así lo indique.

Capa Web: dentro de esta capa se tiene a Main.jsf que posee todo el código fuente necesario para generar la página web antes mencionada. También se cuenta con Backing Beans, estos componentes definen propiedades y funciones para los componentes de una página. En este caso son los encargados de llevar adelante ciertas tareas de control en la carga y muestra de datos en la pagina web. En el caso de SPRABI se tienen los siguientes Backing Beans:

- Borrar.java: permite eliminar la información según la tabla en la que se este trabajando.
- Control.java: realiza los controles de los rangos de las fechas.
- Gauge.java: administra los valores para los gauges (relojes).

Capa de Negocio: esta capa cuenta con Enterprise Java Beans (EJB) y Entidades Persistentes. Dentro de los EJB se tiene un SessionEJBBean.java, en este caso se trata de un Session Bean Stateless. Los Session Beans encapsulan la lógica de negocio que puede ser invocada, mediante programación, por parte del cliente. Un Session Bean Stateless (sin estado) no mantiene un estado conversacional con el cliente.

Las Entidades Persistentes son clases equivalentes a cada una de las tablas de la Base de Datos. Estas clases poseen los métodos necesarios para obtener y/o modificar los valores de sus respectivos atributos. Para SPRABI se tienen las siguientes clases:

- AccionCorrectiva.java
- Indicador.java
- Medicion.java
- Meta.java
- Proyecto.java
- Stakeholder.Java

Capa de Servicios de Información: en esta capa se encuentra la Base de Datos XE, la cuál soporta los requerimientos de persistencia de información de SPRABI. Estos requerimientos de persistencia fueron derivados a partir de un Modelo de Clases del Dominio especificado mediante un Diagrama de Clases de UML, como se muestra en la Figura 3.

Este modelo, que surge a partir del dominio y los requerimientos de información de SPRABI, contempla lo que se considera necesario para poder contrastar el Plan de Proyecto respecto de la Ejecución del Proyecto. Es importante notar que se genera una estructura jerárquica del tipo padre-hijo y en donde éste último no puede existir por si mismo si desaparece el primero.

En un principio es necesario guardar la información propia del proyecto (clase Proyecto) en si mismo. Luego la información referida a las partes interesadas en el proyecto, en este caso los stakeholders (clase Stakeholder). El interés que tienen los stakeholders, en el proyecto, se ven reflejados en los objetivos o metas a cumplir (clase Meta). Para conocer si una meta u objetivo se cumple es necesario medirlo, para ello se necesitan de los indicadores (clase Indicador). Almacenar los distintos rangos o umbrales de medición

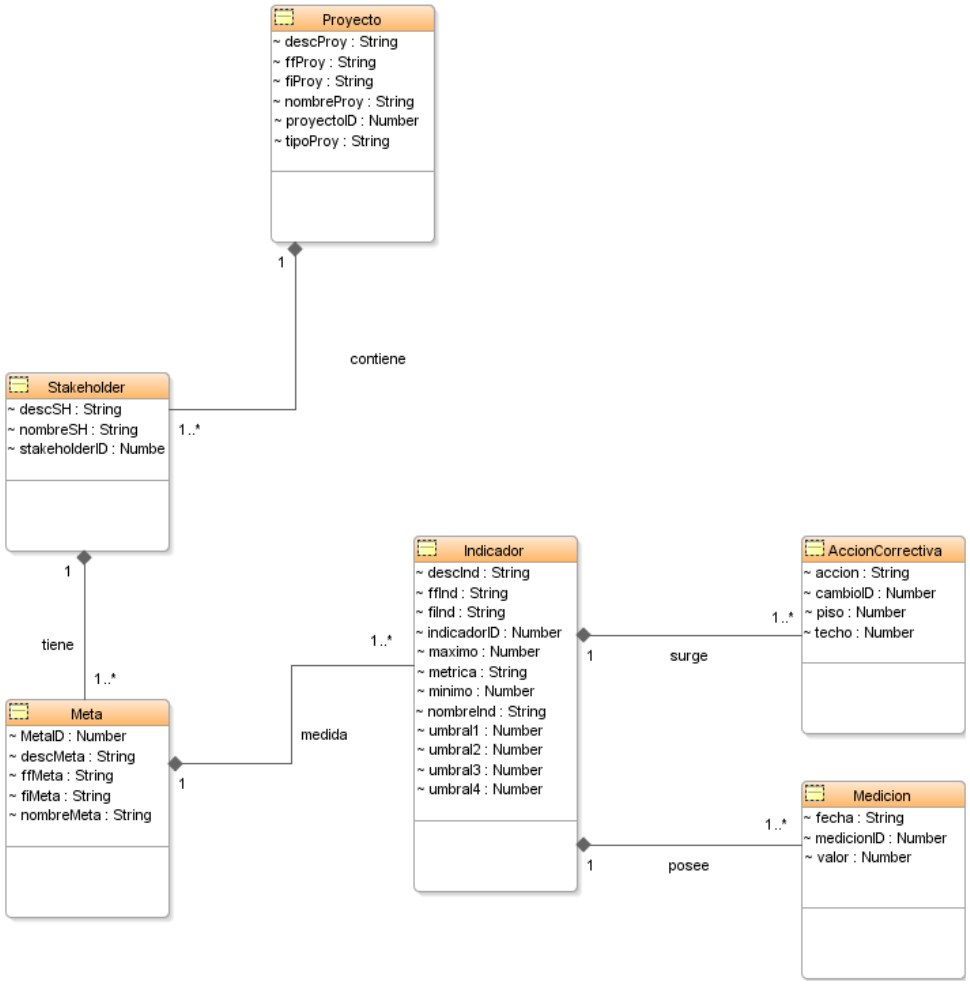


Figura 3. Modelo de clases del dominio de SPRABI

de valores, que tiene un indicador, determinará el cumplimiento o no de un objetivo. Cada indicador tiene sus propias mediciones (clase Medición), estos valores sirven para determinar, en un momento dado (tiempo), en que rango de aceptación cae el valor y por lo tanto, el cumplimiento o no de los objetivos según lo planificado. Por último, cada indicador tiene sus propias acciones correctivas (clase AccionCorrectiva), estas son todas aquellas acciones que se tienen que llevar a cabo, en caso de que las mediciones para un indicador dado no caigan en el rango deseado.

4. Caso de Estudio

Es fundamental presentar un caso de estudio que exhiba y valide la funcionalidad y utilidad de SPRABI, como así también, de los conceptos presentados. El caso de estudio aquí exhibido, surge de un dominio específico de una empresa privada proveedora de servicios públicos de energía. Esta realidad se plantea y desarrolla en la tesis de maestría denominada *Definición de un patrón de diseño de un Balanced Scorecard aplicado a empresas argentinas de servicio público eléctrico* [4].

La finalidad de este capítulo es mostrar cómo se comporta y adapta SPRABI a la información ingresada, siendo los resultados obtenidos acordes a lo esperado. Además, SPRABI se utiliza como medio para presentar y probar diversos temas de la mencionada tesis de maestría. Para mayor información y/o detalle, referirse a dicho trabajo.

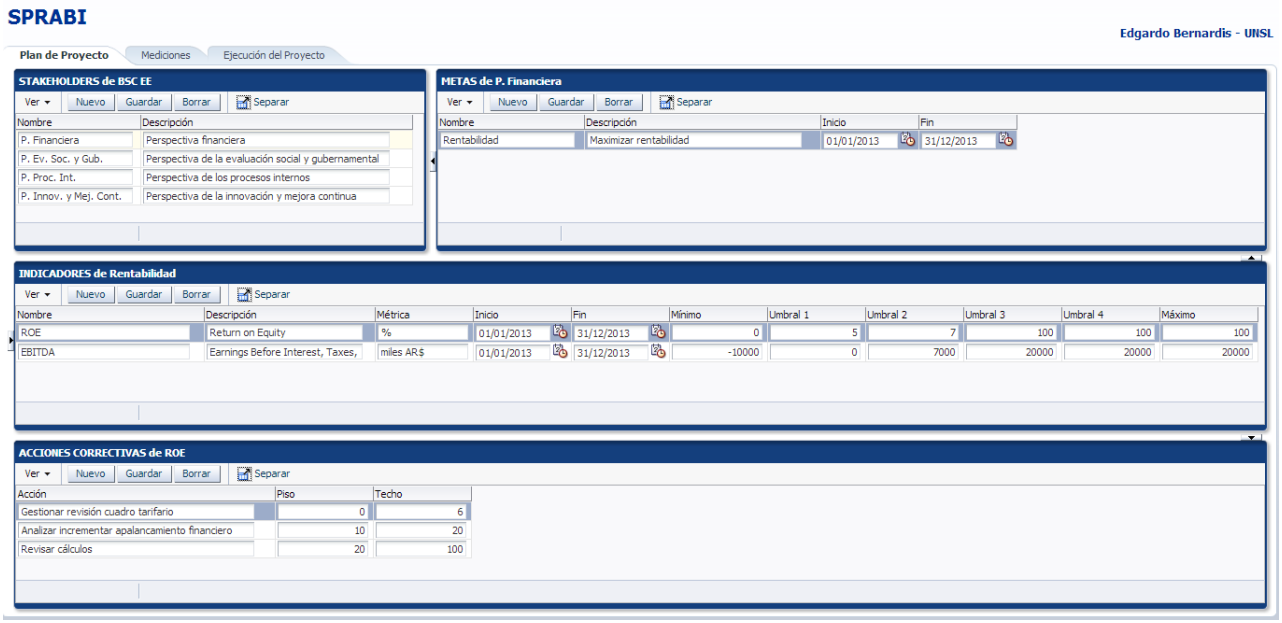


Figura 4. Información dependiente del stakeholder Perspectiva Financiera

4.1. Especificación del Dominio

El caso de estudio de este trabajo está fundado en experiencias reales de implementación de la técnica Balanced Scorecard en empresas distribuidoras de electricidad de la República Argentina. Concretamente: EDESAL S.A. (San Luis), EDELAR S.A. (La Rioja) y EDESA S.A. (Salta).

El alcance de las respectivas implantaciones del Balanced Scorecard incluyó las áreas de explotación , comercial, administración y finanzas, recursos humanos y tecnología informática. Metodológicamente, el proyecto comprendió el desarrollo de una serie de actividades, por parte de analistas funcionales asignados por cada área de las respectivas empresas y un equipo de técnicos de una compañía transnacional especializada en este tipo de implantaciones. El desarrollo de tales actividades se plasmó en un diseño conceptual del Balanced Scorecard enfocado en los siguientes principios: ser de utilidad para gestionar las respectivas áreas, tener en cuenta los datos disponibles en la empresa, estar relacionado con los objetivos estratégicos de la compañía y satisfacer sus necesidades de información. Luego de superada esta fase del proyecto, la siguiente prosiguió con las etapas propias de la implantación de una herramienta de software.

4.2. Ejecución de SPRABI

El funcionamiento de SPRABI se divide en dos partes, una parte de carga de datos y otra de ejecución y muestra de resultados. Primero se procede a cargar los datos requeridos por el sistema, correspondientes a la parte del Plan del Proyecto, los cuales se muestran en la pantalla exhibida en la Figura 4. El orden de ingreso de la información se realiza siguiendo la estructura jerárquica del modelo propuesto. Para cargar, modificar y/o eliminar la información, se optó por dotar a cada tabla de información de tres botones, los mismos son:

- Botón Nuevo: crea una nueva fila vacía en la tabla para poder ingresar la información correspondiente.
- Botón Guardar: almacena la información nueva o una modificación de la misma.
- Botón Borrar: elimina la fila seleccionada en la tabla.

Lo primero que se ingresa es la información respectiva al proyecto en si, en este caso al proyecto se lo denomina BSC EE, se lo se describe como un Balanced Scorecard para una Distribuidora de Electricidad y abarca el período comprendido entre el 01/01/2013 y el 31/12/2013.

Luego se procede con la carga de los stakeholders, en este proyecto se cuentan con cuatro stakeholders, estos son:

- Perspectiva Financiera.
- Perspectiva de la Evaluación Social y Gubernamental.
- Perspectiva de los Procesos Internos.
- Perspectiva de la Innovación y Mejora Continua.

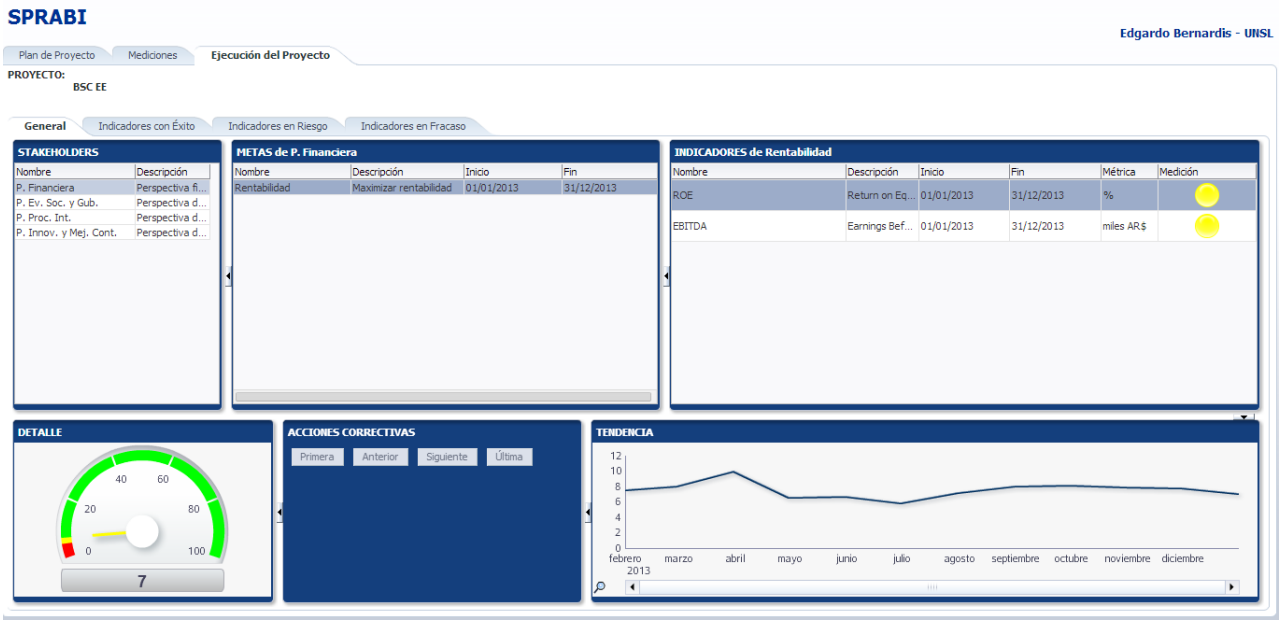


Figura 5. Evaluación de ROE

Siguiendo este proceso, se proceden a cargar todos los datos que jerárquicamente dependen del stakeholder Perspectiva Financiera. En la Figura 4 se observa como se ve en SPRABI toda la información dependiente del mencionado stakeholder, el cuál cuenta con una única meta que es la Rentabilidad. Dicha meta cuenta con dos indicadores: ROE (Return on Equity) y EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization). El indicador ROE revela la proporción que representa el margen de utilidades netas de la compañía con respecto al capital propio o patrimonio neto. Mientras que el indicador EBITDA permite obtener una idea clara del rendimiento estrictamente operativo de la empresa. De los indicadores antes mencionados, sólo el indicador ROE posee acciones correctivas.

Luego de haber cargado toda la información, se proceden a cargar las mediciones, que se tengan, para sus respectivos indicadores en la sección Mediciones, dicha sección cuenta con una pantalla similar a la del Plan de Proyecto, solo que se agrega una tabla específica para la carga de los valores de las mediciones.

Una vez que se tengan las primeras mediciones se puede proceder a la ejecución y muestra de resultados. Esto se lleva a cabo en la sección Ejecución del Proyecto, la cuál muestra cómo es el avance real del proyecto respecto del Plan del Proyecto original.

En la Figura 5 (en la pantalla Ejecución del Proyecto) se puede ver como son las últimas mediciones para los indicadores ROE y EBITDA. Para el primero, se observa que su valor está en riesgo, por lo tanto su color es amarillo. Si se presta atención a la Tendencia; se puede ver que tuvo una caída, ya que el mes anterior su valor era dentro de los parámetros óptimos. A pesar de que los valores no son perfectos, no se sugieren acciones correctivas para cambiar tal situación.

El segundo indicador, EBITDA, también se encuentra en riesgo; si se lo selecciona en la tabla Indicadores y examinando su Tendencia, ver Figura 6; se percibe como son muy variables los valores, para este indicador, a lo largo de todas las mediciones obtenidas. Sin embargo, a partir de septiembre se puede ver una mejora creciente tendiendo a valores esperados u óptimos.Tampoco este indicador tiene acciones correctivas para esta condición.



Figura 6. Evaluación de EBITDA

Por una cuestión de espacio, no es posible mostrar toda la información dependiente de los stakeholders restantes, pero si se muestra el resultado de la evaluación de todos los indicadores restantes en la parte de Ejecución del Proyecto.

En la Figura 7 se pueden observar la evaluación de los resultados de los indicadores restantes. Con estas evaluaciones, se puede ver y concluir que sólo un indicador se encuentra óptimo (color verde), es decir, el valor obtenido cae dentro de los valores esperados. Por el contrario, todos los

restantes están en riesgo (color amarillo) o en fracaso (color rojo). A pesar de esto, el proyecto no es malo, sólo algunos indicadores están lejos de los valores esperados y esto se debe a los altos estándares que se pretenden en dicho BSC.



Figura 7. Todos los indicadores

Para finalizar, y de manera breve, se expone un caso más sobre el funcionamiento de SPRABI, en la Figura 8 se muestra que el indicador Horas de Capacitación por Empleado está en color rojo, lo cuál indica que esta en fracaso con un valor no deseado para la empresa. Si se presta atención a la parte de Detalle, se ve que es una medición que está por debajo de lo óptimo. Para tal comportamiento variable, que se ve en la Tendencia, se proporciona una acción correctiva (para cierto rango de valores) que necesita ser llevada a cabo para contrarrestar este comportamiento y que fue previamente cargada en la etapa de Plan de Proyecto.



Figura 8. Evaluación Horas de Capacitación por Empleado

5. Conclusiones

Las conclusiones obtenidas respecto de este trabajo se enumeran a continuación:

- Se definió exitosamente una estructura que soporte toda la información necesaria para contrastar el plan del proyecto respecto de su ejecución:
  - Con indicadores flexibles, no acotados a un ámbito particular, pudiendo medir un amplio espectro de necesidades según se requieran.
  - Con la adición de acciones correctivas que permiten cambiar el rumbo del proyecto según sea necesario.
  - Con diferentes perspectivas según las partes interesadas (stakeholders) en monitorear el proyecto.
- Se comprobó la complejidad y el impacto decisivo que tiene una buena administración en el potencial éxito de un proyecto. El rol fundamental que cumplen los indicadores en la gestión, como las mediciones obtenidas para cada uno de ellos, afectan los intereses de los stakeholders en el cumplimiento y establecimiento de sus objetivos.
- Se logró diseñar un sistema que sea accesible a través de la web (Sistemas basados en la Web) teniendo en cuenta el crecimiento y la necesidad de este tipo de sistemas en la actualidad. Este tipo de sistemas permiten, entre otras cosas: la independencia del sistema operativo, trabajar con aplicaciones sin necesidad de distribuir e instalar software a miles de usuarios potenciales, ser portables, inexistencia de problemas de incompatibilidad.
- Se programó haciendo énfasis en la arquitectura; con el uso de distintos patrones arquitecturales se logró una construcción robusta, escalable, orientada a componentes, jerárquica, distribuida y heterogénea; con separación de interfaces y de manejo de interacción respecto de funciones y reglas de negocio, apta para procesamiento de requerimientos, con vínculo formal y garantía de consistencia entre interfaces y conocimiento de negocio.
- SPRABI posee una interfaz y diseño simple, intuitiva, sin sobrecarga de información, con menús y barras de fácil acceso; todo ello, permitiendo una lectura rápida de la información requerida.
- Se desarrolló con éxito una herramienta, que en la práctica, ha sido validada en dos ámbitos sumamente diferentes; una empresa de energía eléctrica y una municipalidad de la provincia.
- SPRABI es independiente de la metodología usada para controlar y administrar un proyecto; si se sigue la estructura de trabajo planteada, es aplicable a cualquier modelo de administración de proyecto tales como PMBOK, BSC, CMMI, ISO 9001:2008, RUP, etc.
- SPRABI es una herramienta que no se restringe a proyectos de software, sino que puede aplicarse a cualquier tipo de proyectos en general.

Referencias

[1] I. 9001:2008. http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm, 2014.

[2] R. R. H. S. P. S. M. Buschmann, Frank; Meunier. *Pattern-oriented software architecture. A system of patterns*, volume 1. John Wiley & Sons, 1996.

[3] A. B. Carroll and A. K. Buchholtz. *Business & society: Ethics and stakeholder management*. CengageBrain.com, 2011.

[4] G. Cecconato. Definición de un patrón de diseño de un balanced scorecard aplicado a empresas argentinas de servicio público eléctrico. Master's thesis, Universidad Nacional de San Luis, 2014.

[5] M. Chrissis, Mary Beth; Konrad and S. Shrum. *CMMI for development: guidelines for process integration and product improvement*. Pearson Education, 2011.

[6] CMMI. http://cmmiinstitute.com/, 2014.

[7] R. E. Freeman. *Strategic management: A stakeholder approach*. Cambridge University Press, 2010.

[8] P. M. Institute. A guide to the project management body of knowledge: Pmbok® guide. Project Management Institute, 2008.

[9] ISO. 9001: 2008 sistemas de gestión de la calidad-requisitos. 2008.

[10] G. Jacobson, Ivar; Booch and J. Rumbaugh. *El proceso unificado de desarrollo de software*. Addison Wesley Reading, 2000.

[11] B. Jaramillo and J. Mauricio. *Indicadores de gestión: herramientas para lograr la competitividad*. Bogotá: 3R editores, 1999.

[12] R. E. I. G. D. H. K. M. W. S. C. Jendrock, Eric; Cervera-Navarro. *The Java EE 6 Tutorial: Advanced Topics*, volume 4. Addison-Wesley, 2013.

[13] D. P. Kaplan, Robert S; Norton. Using the balanced scorecard as a strategic management system. *Harvard business review*, 74(1):75–85, 1996.

[14] R. S. Kaplan and D. P. Norton. *The Balanced Scorecard: Translating Strategy into Action*. Harvard Business Press, 1996.

[15] R. S. Kaplan and D. P. Norton. *Cómo utilizar el Cuadro de Mando Integral, para Implantar y Gestionar su Estrategia.* 2000.

[16] H. R. Kerzner. *Project management: a systems approach to planning, scheduling, and controlling*. Wiley, 2009.

[17] P. Kruchten. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004.

[18] G. A. Montejano. Especificación formal en rsl del balanced scorecard. Master's thesis, Universidad Nacional de San Luis, 2005.

[19] Oracle. *The Java EE 6 Tutorial*. Oracle, 2013.

[20] C. Parodi. El lenguaje de los proyectos. *Gerencia social. Diseño, monitoreo y evaluación de proyectos sociales*, 2001.

[21] R. S. Pressman. *Ingeniería del software un enfoque practico. Quinta edición.* McGraw-Hill Interamericana de España, 2002.

[22] SEI. Cmmi® for development, version 1.3. Technical report, Carnegie Mellon University, 2010.

[23] J. W. Traa. Rational unified process vs. microsoft solutions framework: A comparative study, 2006.

[24] R. Tricker. *ISO 9001:2008 for Small Businesses*. Elsevier, 2010.

Mecanismos Institucionales de transferencia tecnológica

Patricia Zachman[1], Walter López[1], Andrés Redchuk[2,1]

[1] Departamento de Ciencias Básicas y Aplicadas.
Universidad Nacional del Chaco Austral. Argentina.
[2] Facultad de Ingeniería.
Universidad Nacional de Lomas de Zamora. Argentina.

ppz@uncaus.edu.ar, walgus@uncaus.edu.ar, andres.redchuk@gmail.com

Abstract

En las últimas décadas, la Universidad necesita transferir los resultados de sus investigaciones para ser aplicados al entorno productivo, traduciéndolas en valor comercial. Ello da origen a la creación de incubadoras de empresas universitarias, convertidas en los últimos años en uno de los mecanismos de transferencia de tecnología de referencia.

El trabajo que se presenta aborda la creación de incubadoras de empresas universitarias bajo el paradigma de la Arquitectura Orientada a Servicios (SOA) con la intervención de estrategias del Modelado Procesos de Negocios (BPM), para establecer directrices que integren los conceptos de tecnología y de negocios en una arquitectura empresarial definida, sencilla y actualizada con la meta de innovar servicios a clientes y adaptarse ágilmente ante cambios, incluyendo la reacción temprana ante la competitividad.

Mientras que SOA, ofrece un marco de trabajo de negocios para diseñar, desarrollar, integrar y consolidar sistemas heredados y distribuidos, BPM proporciona conceptos, métodos, técnicas y herramientas de software que permite el análisis y la detección de problemas y oportunidades que, diseñadas electrónicamente, pueden ser ejecutadas y mejoradas continuamente.

La contextualización de la propuesta se circunscribe a la Universidad Nacional del Chaco Austral, en la ciudad de Sáenz Peña, Chaco.

Palabras Clave: *Spin off, incubadoras, Pertinencia Sustentable, Arquitectura orientada a servicios, Modelización de Procesos de Negocio.*

1. Introducción

Un componente importante en el desarrollo de la educación superior universitaria es el énfasis en el vínculo de las siguientes dimensiones: las demandas del mercado laboral-empresarial-profesional, los requerimientos de la sociedad, y la gestión de la autorrealización humana. La Universidad, el sector productivo y los gobiernos regionales son los actores principales en la dinámica de la relación Universidad - Mercado – Estado. Desde un enfoque contextual, esta relación significa, a nivel local centrochaqueño, poder acercar a los docentes investigadores a las realidades de las necesidades de cada sector productivo y encontrar, por medio de sus proyectos articulados de investigación, soluciones prácticas e innovadoras, que puedan mejorar la productividad, la competitividad, la satisfacción y una mejor calidad de vida.

La Universidad Nacional del Chaco Austral (UNCAUS) se alinea en tres funciones: producción de conocimiento, transmisión del conocimiento y transferencia del conocimiento, esta tercera misión es la estrategia que este proyecto propone.

Resulta ser una universidad joven, nacida en el 2009, pero con un caudal heredado de investigaciones, producciones y servicios destinados a la provincia.

A partir de esta experiencia, pretende, con el conocimiento útil, apropiarlo y validarlo en los usuarios finales a través del emprendedurismo universitario, en otros términos repensar el conocimiento generado desde la inteligencia de negocios y la actualidad en la prestación tecnológica de servicios.

En la literatura académica, se habla de Spin-Offs cuando se crea una nueva empresa a partir de un grupo de investigación universitario, cuando un empleado deja su compañía para emprender un nuevo negocio o cuando una empresa se divide en diferentes partes [11]. Shane [10], define un Spin-Offs académico como "las empresas de alta tecnología, cuya actividad principal se basa en la valorización comercial de los resultados de una investigación científica y tecnológica". Más recientemente, Conti et al [3], las define como "aquellas empresas que germinan de una universidad, donde un grupo de investigadores componen la unidad empresarial con miras a la explotación de los conocimientos y resultados de la investigación desarrollada en propia Universidad".

En primer lugar se trata de una entidad distinta e independiente de la universidad de origen, cuyo fin es

obtener beneficios de la realización de su actividad. El conocimiento transferido a la empresa no tiene por qué ser exclusivamente innovaciones tecnológicas o patentes, sino que también puede considerarse el conocimiento tácito acumulado por el individuo durante sus actividades académicas [9].

Utilizar un mismo concepto para describir diferentes realidades impide comparar los resultados y avanzar en el conocimiento de la actividad emprendedora de las universidades. Por ello, a la hora de abordar el fenómeno de las Spin-Off académicas se hace preciso identificar sus características definitorias. Así, dado que en este trabajo pretende vincular la actividad investigadora y las iniciativas empresariales centrochaqueñas, se considerarán spin-off aquellas empresas promovidas por académicos para explotar comercialmente los resultados de sus investigaciones, bien con propósitos industriales, o bien orientados a la prestación de servicios. Siguiendo a Aceytuno y Cáceres [1] no es relevante que los investigadores formen parte de la spin-off creada, ni que exista apoyo financiero de la universidad de origen, ya que el objetivo de estas empresas es contribuir al desarrollo económico a través de la transferencia de conocimiento.

Las Spin-off se instalan en un parque científico y tecnológico, donde a su vez surgen incubadoras. Se trata de una macroestructura, "el parque", que dentro alberga la mesoestructura "la incubadora", que a su vez alberga varias microestructuras: "las Spin-off o Start up". Un parque/polo científico y tecnológico es la infraestructura técnica y de laboratorios y es este el enfoque que prevalece al hablar de una Spin-off Estos parques ofrecen instalaciones y servicios comunes con el objetivo de acoger a los investigadores-emprendedores.

Por ello nacen las incubadoras de empresas, viveros de empresas, centros de promoción empresarial, centros de nuevas empresas que consisten básicamente en espacios físicos acondicionados para albergar actividades empresariales o industriales en etapa de diseño, prototipos e inicio formal de producción o servicios, al cual se agrega la asistencia técnica y el acompañamiento necesario para llegar a constituirse en empresa.

Dentro de nuestro medio científico - tecnológico, el perfil clásico del emprendedor de países desarrollados es difícil de encontrar. Ello se denota por factores como:

- *Déficit de carácter emprendedor: Las instituciones científicas, por lo general, no han desarrollado una política de promoción del emprendedor ni de las características de una universidad emprendedora. No existe cultura social relacionada con el riesgo y que acepte el fracaso habitual en operaciones tipo Spin-Offs de alto riesgo comercial. Desde el punto de vista personal, para la mayoría de los investigadores, la creación de una empresa a partir de sus conocimientos y resultados no sólo está alejada de sus intereses, sino que tampoco se promueve como una alternativa para encaminar a un futuro profesional o a estudiantes graduados.*

- *Déficit organizativo: La función genérica de apoyo a la investigación, no contempla directamente la necesidad o conveniencia de creación de los Spin-Offs, si bien algunas universidades han iniciado programas experimentales que incluyen incubadoras de empresas y servicios básicos de apoyo.*

- *Déficit de experiencia del equipo promotor de la idea: generalmente los promotores de nuevas ideas con potencialidades comerciales no presentan conocimientos y experiencia directiva. El número de emprendedores con conocimientos empresariales y una formación sólida científica y tecnológica, es insuficiente para hacer frente a las potenciales ideas que en un futuro debieran llenar las Spin-Offs.*

- *Déficit de capital social o de recursos económicos: El ciclo de financiamiento está poco desarrollado, con poca experiencia en algunos de sus instrumentos. Hay poca tradición en capital de riesgo, inexistencia de inversionistas, así como pocos inversores especializados en Spin-Offs de alto riesgo.*

- *Déficit en la sostenibilidad de empresas que surgen de incubadoras. Modelos de desarrollo no consolidados implican la revisión y ajuste continuo de las variables de superviviencia empresarial*

Paradójicamente, el tema de las Spin-Offs no es desconocido, es citado como estrategia en muchos planes de desarrollo económico o políticas de ciencia y tecnología, no solo en países como Colombia, Perú, Ecuador y Chile; sino también como parte de Planes Estratégicos Educativos en Argentina. Ello permite inferir que existe sensibilización hacia la importancia que implica la generación de las Spin-Offs para el desarrollo económico y tecnológico de los países, pero faltan elementos impulsores y sustentadores para lograr resultados satisfactorios.

Es importante resaltar también, que existe poca investigación local en innovación y emprendimiento. Bajo este panorama, la Universidad Nacional del Chaco Austral se apresta a abordar la creación de un modelo de Spin-Offs con enfoque SOA desde BPM, como un mecanismo de transferencia tecnológica al mercado chaqueño.

2. El paradigma Spin off en el contexto del Chaco

La evolución del concepto de spin-off encuentra a la UNCAUS en un proceso incipiente de creación de

incubadoras vinculadas a los procesos de desarrollo económico.

El paradigma spin off - UNCAus agrupa esencialmente de dos partes (o módulos). La primera corresponde a la Generación de Empresas y la segunda al Crecimiento de las mismas. El módulo de Generación de Empresas es el punto de partida e implica la convergencia de ideas y de personas. El Módulo de Generación de Empresas es un intento para dar algo de orden y de estructura al proceso de despegue, al agrupar a las ideas y personas de una manera organizada, dejando al mismo tiempo libertad para la creatividad y la suerte. Una vez formada, la empresa avanza por un Módulo de Crecimiento Empresarial. Este proceso es más estructurado y la empresa tiene que atravesar determinadas etapas de crecimiento que responden a diferentes grados de madurez y necesidad de apoyo. En esta forma, se estimula a la empresa a que alcance todo su potencial. Si esto no parece posible, entonces se consideran otras opciones, tal como encontrar un socio estratégico o eventualmente cerrar la empresa.

Los ingredientes básicos de cualquier empresa son ideas y personas. Las opiniones difieren sobre cuál es la más importante, pero indudablemente ambas son necesarias si se quiere establecer una empresa exitosa. La metodología del Módulo de Inicio de la Empresa consiste en tratar a las ideas y a las personas como dos áreas distintas para luego agruparlas en un proceso conjunto en una tercera área.

Una vez iniciada la empresa pasa por una serie de etapas. Este modelo define cuatro etapas que se relacionan directamente al progreso del producto o servicio ofrecido por la empresa. Estas etapas son descritas en término de los estados de crecimiento biológicos que se encuentran en el mundo natural:

a) *Etapa Embrionaria: Durante esta etapa el equipo involucrado tiene un prototipo que puede ser mostrado a posibles clientes e inversionistas. Con el fin de mantener bajos los costos generales y estar cerca de los creadores de la idea, esta etapa se realiza en el laboratorio universitario.*

b) La Etapa de Nutrición: En esta etapa el prototipo se convierte en un producto vendible y mientras esto ocurre, la empresa puede ir recibiendo órdenes de compra cuando los clientes estén preparados para aceptar que todavía se está probando el producto, o se está desarrollando.

c) La Etapa de Despegue: Aquí la empresa avanza desde su primer producto y comienza a desarrollar una gama de productos o servicios.

d) Etapa de Madurez: La empresa tiene un gran potencial pero se debe tener cuidado para lograr desarrollarla completamente. Muy a menudo la compañía no logra desarrollar todo su potencial porque no es capaz de lograr cambiar o madurar su estilo gerencial.

El proceso de generar y desarrollar nuevas empresas requiere una infraestructura de apoyo para que funcione. La infraestructura de apoyo sirve de base y sostén a este proceso proveyendo las instalaciones y recursos necesarios. Se puede dividir en infraestructura de apoyo primaria y secundaria. La primaria incluye las instalaciones de operación y el soporte financiero y la secundaria las facilidades locales, los recursos de la empresa y las redes de empresa.

Respecto de la infraestructura de apoyo primaria, se han previsto:

- *Instalaciones de operación. Incluyen los laboratorios de redes e investigación de la Universidad, que soportan a las empresas en su crecimiento.*
- *Soporte financiero: Comprende el apoyo financiero provisto para las empresas como individualidades.*

Respecto de la infraestructura de apoyo secundaria, se consideran:

- *Condiciones de Operación; este aspecto se refiere a una infraestructura de comunicaciones mínima.*
- *Recursos de soporte empresariales: Son parte importante de la infraestructura porque afectan la facilidad con que puede operar una compañía que se está iniciando. Como en este caso se consideran recursos universitarios, se menciona personal de apoyo técnico, asistencia profesional en las áreas de gerencia y mercadeo. Todo esto forma un cuerpo de recursos que toda empresa nueva necesita.*
- *Redes Empresariales: Estos son los elementos de apoyo más importantes y menos costosos. Las redes informales son desarrolladas a través de la universidad.*

Se menciona, también, el conjunto de variables sobre el medio ambiente de operación:

- *Situación del mercado: Esto contiene varios aspectos, por ejemplo: las dificultades de entrada al mercado presentan problemas particulares para las empresas de tecnología que se inicia porque o bien tendrá que crear su propio mercado o convencer a un mercado tradicional existente.*
- *Factores económicos: La nueva compañía tiene que aprender a vivir con la inflación, las tasas de interés, las normas para obtener préstamos de capital y el clima económico en general.*
- *Estructura legal: Las nuevas empresas aprenden a trabajar de acuerdo a las normas legales del mercado al que ingresan. La posición sobre los derechos de propiedad intelectual en las universidades constituye un importante ejemplo ya que las restricciones son generalmente*

demasiado grandes y existe poco incentivo para que el académico piense en explotar el conocimiento desarrollado.

- *Circunstancias políticas: Éstas incluyen los factores como la estabilidad política de la región (en corto plazo), y las políticas nacionales.*
- *Aspecto socio-cultural: incluye la actitud hacia la empresa y la relación empresarial, porque esto afecta las habilidades de las personas para ver oportunidades comerciales y la facilidad con que ellos pueden aprovecharlas.*

3. Panorama argentino

Si bien no hay información censada sobre la cantidad de spin-offs con sede universitaria en Argentina [2], se ejemplifican a continuación algunas de ellas (Tabla 1)

Empresa	*Lugar de Nacimiento*	*Actividad*
Homoderivados	*UNC*	*Medicamentos*
METAS SRL	*UNC*	*Servicios Tecnológicos para empresas alimenticias, metalúrgicas y energéticas*
CELINT SA	*UNL*	*Producción de celulosa microcristalina grado farmacopea*
Res SCREEN SRL	*UNL*	*Detección de antibióticos en la leche.*
Spinlock S.R.L	*UNC*	*Detectores de explosivos por resonancia cuadrupolar*
INVAP	*CNEA. Instituto Balseiro.*	*Realización de proyectos espaciales*
Zelltek SRL	*UNL*	*Biotecnología*

Tabla 1: Spin Offs en Argentina

Al introducirse en el tema del Spin-off universitario, se observa un campo amplio, heterogéneo, que lo caracteriza con una diversidad de conceptos utilizados para nombrar un hecho también diverso, lo cual ejemplifica los múltiples enfoques que puede adquirir la instauración de este negocio.

4. SOA y BPM

La presentación de la Universidad Emprendedora ha transcendido repercutiendo en las acciones estratégicas de estas instituciones en todo el territorio nacional: aumento del número de licencias, patentes, inversión en incubadoras y participación activa en parques científicos y tecnológicos, así como aumento de servicios y acciones de apoyo para fomentar la cultura emprendedora y animar la transferencia de tecnología. A este contexto se le une la nueva realidad de financiación por indicadores de las Universidades que refuerza el compromiso y favorece la orientación a resultados, todo ello con el imperativo de soportar sus estructuras de investigación acercándose al tejido productivo y poniendo a su disposición sus capacidades científico-técnicas. [8]. La interiorización de modelos de apoyo a la creación de Empresas de Base Tecnológica y, de forma especial de las Spin-Offs, se ha convertido en un desafío para las políticas universitarias. Esta cuestión es común a todos los sistemas universitarios, incluidos aquellos menos consolidados.

El éxito o supervivencia de una spin off depende del aprovechamiento eficiente de sus recursos y capacidades para competir con éxito con otras empresas. (Figura 1).



Figura 1: Pilares de la Supervivencia Empresarial de las Spin Off

Muchas de las tareas están orientadas hacia el área de sistemas, volviéndose pieza clave tanto las estructuras de Hardware como de Software. Lo que se necesita es actualizar el paradigma de procesos e implementar una integración y orientación de todos los datos y transacciones.

En conformidad con los últimos avances tecnológicos en el área de Software surge un paradigma orientado a servicios, llamado SOA (Service Oriented Architecture). Una Arquitectura Orientada a Servicios (SOA), es un concepto de diseño de arquitectura que trata de agrupar aplicaciones de software (TI) para integrarlas en un mismo lenguaje (servicios: flexibles, seguros y funcionales bajo estándares) para el uso eficiente empresarial de todos los usuarios y procesos alineados a los objetivos organizacionales. Su objetivo es establecer la integración de aplicaciones independientes con lenguajes de programación, arquitectura y plataformas diferentes, de manera que desde la web puedan tomarse las diferentes funcionalidades convertidas en servicios, para que estos sean totalmente versátiles y flexibles, permitiendo la reutilización de procesos en el entorno organizacional; básicamente alinear los procesos de negocio con los procesos de tecnologías de información, que a grandes rasgos pro ser reutilizables a corto y mediano plazo, pueden convertirse en un ahorro significativo, hablando monetaria y procedimentalmente.

Desde el punto de vista de negocio, se define SOA como un conjunto de componentes informáticos que se integran de forma flexible para configurar distintos procesos de negocio [6]; desde una

perspectiva técnica, estas arquitecturas constan de servicios que se pueden invocar para realizar operaciones específicas.

Dentro del marco conceptual de SOA, encajan distintas tecnologías y protocolos, como los Servicios Web. Los Servicios Web pueden definirse como aplicaciones de software que se descubren, describen y resultan accesibles mediante mensajes escritos en XML (eXtensible Markup Language) o en otros protocolos propios de las tecnologías Web [6]. Su función es habilitar el intercambio de datos entre aplicaciones que se han desarrollado en lenguajes de programación distintos y que se ejecutan en diversas plataformas informáticas; asimismo, favorecen la integración multicanal de las empresas.

Los servicios o aplicaciones de negocios son módulos informáticos bien definidos que ofrecen por sí mismos una funcionalidad de negocio y cuyo funcionamiento resulta dependiente del estado y de las características del resto de servicios con los que se relaciona [6].

Las funcionalidades individuales realizadas por cada servicio se deben orquestar para implementar los procesos de negocio específicos de cada organización; para ello, se recurre a los modelos de procesos. Las principales características funcionales de las soluciones SOA son las siguientes:

- *Infraestructura tecnológica formada por servicios independientes: los distintos servicios se unen en forma de red para componer los procesos de negocio de la organización y pueden ser utilizados por el resto de aplicaciones denominadas funciones de negocio mediante múltiples servicios ([12]).*

- *Posibilidad de reutilizar los servicios: muchos de los servicios se pueden reutilizar en distintos procesos de negocio porque realizan funciones estándares; de manera que la implementación de nuevos procesos sólo requiere el desarrollo de las funcionalidades propias del mismo, lo que simplifica los desarrollos informáticos de una empresa [12].*

Estas características funcionales son posibles gracias a un conjunto de características lógicas y técnicas propias de este tipo de soluciones:

- *Bajo nivel de acoplamiento de los servicios (Loose Coupling): cada servicio debe ser independiente del resto de servicios conectados a la misma red. De este modo, se facilitan las modificaciones tanto del código de un servicio como de la configuración de todos ellos para dar lugar a los procesos de negocio (Scott J. E. Vessey I., 2002)*

- *Independencia entre los servicios y las características técnicas de su implementación: para favorecer la independencia entre los distintos servicios, las funciones que éstos realizan deben ser independientes de las características técnicas de su implementación [5] Esto, junto al uso de protocolos estándares, permite que las soluciones SOA puedan operar con aplicaciones de cualquier tecnología.*

- *Uso de interfaces estándares: la utilización de interfaces estándares es necesaria para permitir que servicios de distintas tecnologías e, incluso, formatos de datos diferentes puedan intercambiar información sin tener que recurrir a interfaces ad hoc [5].*

Con la intención de evitar riesgos en la elección de los casos, se generaliza la mención de características, considerando además que SOA es un paradigma que se está desarrollando principalmente en los últimos años.

La propuesta BPM (Business Process Management o Gestión de Procesos de Negocios) es que provee un conjunto de metodologías para el análisis, comprensión y documentación de los procesos de negocios. Ayuda a los directores de una empresa u organización a adaptarse a los cambios continuos de los procesos de negocios, mientras que SOA ayuda a los arquitectos de software a crear sistemas empresariales que se adapten a los continuos cambios en los requerimientos.

Los gestores de procesos (Bussiness Process Management-BPM) son programas informáticos que se encargan de decidir en cada momento qué servicio debe invocarse para que realice su función; esto lo hace basándose en la descripción del proceso de negocio que le ha sido programado [12]. Como los procesos de negocios son realizados en espacios de información tecnológica compleja, la integración de los sistemas de información existentes se convierte en una base importante para la implementación técnica de los procesos de negocio. BPM es entonces una estrategia para gestionar y mejorar el rendimiento de un negocio optimizando sus procesos a través de la modelización, ejecución y medida de rendimiento dentro de un ciclo de mejora continua. Es una disciplina que va mucho más allá de la arquitectura de software que la implementa [7] [6] [4].

Para poder implementarlas en el modelo empresarial de Spin Off Académico, se debe contar con una arquitectura empresarial totalmente soportable, con buenas bases, tanto tecnológicas como de negocios. Ya no es permitido orientar las organizaciones hacia solo manejo de datos y transacciones manuales, el enfoque debe ser tomado desde el punto de vista de los servicios. SOA y BPM son poderosos componentes para apoyar en todos los procesos de negocio. Mientas que SOA, proporciona sistemas livianos y adaptados que brinde apoyo a los procesos de negocio, BPM enmarca los procesos de negocio y los comunica con los sistemas de información permitiendo el análisis y diseño, implementación y prueba de unidades, integración y pruebas de los procesos del sistema y por último la operación, puesta en marcha y mantenimiento, sin

olvidar la generación de vinculación de todos los servicios como procesos legítimos de negocio. Todo ello bajo una reformulación de roles y responsabilidades en la definición, especificación e implementación de los proyectos.

5. Conclusiones y Discusión

Las Spin-Offs Universitarias se pueden considerar como herramienta de conversión de la investigación científica en valor empresarial y comercial, teniendo en cuenta:

1) El papel de la Universidad en las economías locales y de actual crecimiento;

2) La importancia de políticas de estímulo de emprendimiento en la Universidad como herramienta de conversión de la investigación científica en valor empresarial y comercial,

3) La relativa imprecisión terminológica en el uso del concepto Spin-Offs, especialmente en el marco universitario.

Es importante tener en cuenta que el concepto de "cloud computing" y virtualización, representa sin lugar a dudas un punto de inflexión para el valor que posee la computación en red. La idea subyacente en "cloud computing" es que aporta un nivel superior de eficiencia para distribuir y desplegar recursos de IT bajo demanda. La incorporación de conceptos como virtualización, despliegue bajo demanda, distribución de recursos por Internet y software open source, cambian el enfoque no sólo para el desarrollo de aplicaciones, sino también para su despliegue, mantenimiento y fundamentalmente su interoperabilidad.

De esta manera, se advierte en el campo de las redes sociales con Web 2.0 y "cloud computing" un terreno propicio para la aplicación de los conceptos de BPM en términos de aportar la coordinación necesaria a las actividades de una red social y de SOA como un marco de referencia para poner a disposición los servicios remotos de la "nube". Esto constituye un nuevo desafío dentro del área de las Spin-Offs universitarias.

Se puede concluir que la UNCAUS carece de estrategias y métodos planificados empresarialmente de tipo logístico a mediano y largo plazo, en lo que respecta a transferir investigación al mercado: determinar el tamaño del mercado, convenir el aporte que generará al cliente, definir la capacidad para atender al mercado, obtener recursos financieros, entre otros. Esta situación no facilita a la institución planificar un proyecto de Spin-Offs con enfoque I+D+i, con dificultades para poder advertir oportunamente las necesidades del mercado y la relación demanda – oferta.

La delimitación del campo de actividad de nuevas empresas subyacentes de la investigación, así como el grado de elaboración estrategias de inteligencia de negocios tiene una escasa concientización y comprensión académica.

Las competencias emprendedoras a nivel Universidad, con sus experiencias, sus motivaciones, que juegan un papel fundamental en la orientación y devenir de la empresa, se encuentran poco desarrolladas.

6. Referencias

[1] ACEYTUNO M.T, Y. C., F.R. 2009. Elementos para elaboración de un marco de análisis para el fenómeno de las Spin-Offss universitarias Revista de Economía Mundial, Vol. 23, p. 23-52.

[2] ALAFEC, X. A. G. D. 2009. Los desfíos de las Spin-Offs Universitarias.

[3] CONTI G., G. M., PICCALUGA A., 2011 La gestione del trasferimento tecnologico. Strategie, Modelli e Strumenti. Springer, Milano.

[4] GARMENDIA BERAZA J M., 2012. Los Programas De Apoyo A La Creación De Spin-Offss Académicas En Las Universidades Españolas: Una Comparación Internacional. Universidad del País Vasco. Facultad de Ciencias Económicas y Empresariales. Departamento de Economía Financiera II. ISBN 978-84-9860-422-1.

[5] OASIS COMMITTEE SPECIFICATION 2006. Reference model for Service Oriented Architecture 1.0. . McKenzie y otros (eds.). OASIS Open.

[6] PAPAZOGLU M P. 2007. Service Oriented Architectures: Approaches, Technologies and Research Issues . VLDB Journal.

[7] PIRNAY, F., SURLEMONT, B., Y NLEMVO, F. 2003. Toward a typology of university Spin-Offss. Small Business Economics, Vol. 21, núm. 4, p. 355.

[8] PLESCHAK F. 2003. Entwicklungstendenzen des Technologie transfers und Anforderungen an seine Ausgestaltung. . Technologie transfer, Anforderungen und Entwicklungstendenzen, Karlsruhe., pp 1 -16.

[10] SHANE S. 2004 Academic entrepreneurship: University Spin-Offss and wealth creation. Cheltenham, UK: Edward Elgar.

[11] WALLIN M.W. 2012. The bibliometric structure of Spin-Offs literature. Innovation: Management, policy & practice, Vol. 2, p. 162-177.

[12] ZACHMAN P, LOPEZ W & REDCHUK A. 2014, Abordaje de Spin off universitario desde BPM y SOA pertinentes con el Desarrollo Local, 2do Seminario Internacional para la Investigación en Administración y Negocios, Puerto Vallarta, México.

Metodologías y Guías de Gestión de Proyectos de TICs en el Sector Público: Enfoque Tradicional vs. Enfoque Ágil

Patricia R. Cristaldo[1], Luciana C. Ballejos[2], Mariel A. Ale[2]
(1) Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay
(2) CIDISI – Universidad Tecnológica Nacional, Facultad Regional Santa Fe

Resumen

En los últimos años la adopción de las Tecnologías de la Información y la Comunicación (TICs) se ha convertido en un motor de cambio y fuente de ventajas competitivas en las organizaciones del sector público. Debido a esto, la correcta gestión de este tipo de proyectos es un elemento esencial para mejorar la probabilidad de alcanzar los objetivos tanto de los procesos como de las personas implicadas en tal gestión de manera eficaz, a través de una metodología que contemple las buenas prácticas del área. En este artículo se presenta un análisis comparativo de distintas metodologías de gestión de proyectos, cada una de ellas con enfoques y características particulares, en relación a requisitos que las mismas deberían cumplir para la gestión de proyectos de TICs en el sector público. Además de analizarse y fundamentarse la necesidad de combinar los enfoques, generando un "híbrido", para el caso de la gestión de proyectos de TICs en éste entorno específico.

Palabras Clave
Gestión de proyectos de TICs, metodologías de gestión de proyectos, sector público.

Introducción

La gestión de proyectos dentro del nuevo paradigma de la *gestión pública* hace énfasis en mejorar la eficiencia de sus procesos, focalizándose en la excelencia y orientación al ciudadano, transparencia y responsabilidad [1, 2, 3]. Dentro de este contexto las TICs tienen un rol importante, ya que son consideradas fuente de la trasformación y modernización del estado. Por tal motivo, los especialistas proponen avanzar en los programas de modernización de la gestión pública y en los esfuerzos de incorporación de tales tecnologías [4,5]. Para esto, es necesario ejecutar los proyectos tecnológicos con la utilización de nuevos instrumentos de gestión y control.

Como respuesta a esta situación, diferentes guías y metodologías de gestión de proyectos han sido propuestas: las metodologías PRINCE2, METRICAV3 y APM, y las normativas PMBOK y ATERN. Las mismas deben aportar considerables ventajas a la gestión de proyectos, facilitando las tareas de planificación, control y seguimiento, optimización de recursos disponibles, evaluación de resultados, comunicación efectiva con todos los interesados, entre otras. Según la filosofía de desarrollo de proyectos, tanto las guías como las metodologías se pueden clasificar en dos grupos o enfoques: *tradicionales*, que se basan en una fuerte planificación durante todo el desarrollo y un ciclo de vida más tradicional [6, 7, 8, 9, 10], y *ágiles*, en las que el desarrollo es incremental, cooperativo, sencillo y adaptable [10, 11, 12, 13, 14]. Por otra parte, si se consideran los proyectos vinculados con las TICs que se ejecutan en el sector público, los mismos están condicionados a particularidades que determinan fuertemente la ejecución de este tipo de proyectos [4, 5]:

Gobierno: involucra aspectos tales como la existencia de normas jurídicas específicas que determinan las actividades y los tiempos políticos.

Tecnología: involucra aspectos relacionados con los cambios tecnológicos, que se trasforman a una velocidad diferente a los cambios en la administración pública.

Gestión: se incluyen las limitaciones propias de las personas encargadas de gestionar y administrar los proyectos

tecnológicos, conjuntamente con una escasa gestión de proveedores y otras partes interesadas.

En este artículo se realiza un análisis comparativo de las guías y las metodologías más usadas, a los fines de determinar la adecuación de alguna de ellas para la gestión de proyectos de TICs en el entorno específico de la gestión pública.

Elementos del Trabajo y metodología

De las problemáticas descriptas precedentemente, se han formulado cuatro perspectivas, a partir de los aportes de autores [15, 16], proporcionando un marco y una estructura que se espera que cumpla una metodología que gestione proyectos de TICs en el sector público. Las perspectivas son:

Formación y Crecimiento: esta perspectiva incluye todos aquellos aspectos relacionados con la gestión de los recursos humanos necesarios para poder ejecutar un proyecto de TICS, habilidades, motivaciones, entre otros. Los requisitos involucrados en esta perspectiva son los inductores necesarios para el desarrollo de equipos de trabajo.

Procesos internos: esta perspectiva incluye todos aquellos aspectos relacionados con la gestión de los procesos del proyecto, que el gerente del mismo deberá acentuar de manera tal de lograr la satisfacción de los interesados. Las autoridades y el jefe de proyecto, deben identificar los procesos críticos en los que se debe sobresalir con excelencia si es que quieren satisfacer los objetivos que ellos mismos se propongan y los objetivos de los demás interesados en el proyecto.

Interesados: esta perspectiva incluye todos aquellos aspectos vinculados con las problemáticas propias del sector gobierno. Es decir, es de vital importancia para el éxito de un proyecto TICs en el sector público, el grado de compromiso de las autoridades para con el mismo. Se trata de identificar claramente los interesados potenciales y la satisfacción de sus necesidades.

Técnicas: esta perspectiva incluye todos aquellos aspectos relacionados con las técnicas de ingeniería vinculadas con el proyecto, tales como: diseño, documentación, pruebas, tecnología, entre otros.

La Tabla 1, resume los principales aportes de cada metodología o guía de gestión de proyectos, en relación a una serie de requisitos prioritarios contenidos en las perspectivas consideradas precedentemente. Los requisitos se determinaron a partir de las razones de fracaso y los factores críticos de éxito en la gestión de proyectos de TICs, propuestos por la bibliografía y buenas prácticas profesionales [15, 17, 18, 19, 9]. Ellos se utilizan para realizar la comparación de características y restricciones de las metodologías y guías de gestión de proyectos, ya que se corresponden directamente a las problemáticas que condicionan fuertemente la ejecución de los proyectos de TICs en el dominio analizado. Es decir, los requisitos que se plantean, son los mínimos y necesarios que deberán ser considerados por una metodología, de manera tal de lograr el éxito de los proyectos de TICs en el sector público. La Tabla 1, muestra que si bien, la mayoría de los requisitos son considerados por las metodologías y guías analizadas, los mismos no están todos presentes en una única metodología de gestión de proyectos.

Tabla 1.Principales aportes de las metodologías y guías de gestión de proyectos en el sector público

Perspectivas	Factores de éxito	Requisitos	Enfoques				
			Tradicional			Ágil	
			PRINCE2	METRIC	PMBOK	APM	ATERN
Formación y Crecimiento	Dinámicas de equipo en la gestión de proyecto	Grado de cooperación y comunicación	X	X		X	X
	Compromiso, colaboración y disponibilidad de los miembros del equipo e interesados	Construcción y madurez del equipo/interesados.	X		X		X
		Desarrollar al interesado y a los miembros del equipo			X		
	Conductas de apoyo emocional entre los miembros del equipo	Compenetración en el equipo/moral.	X			X	
	Recursos especializados según el rol que ocupa cada miembro del equipo de proyecto	Competencias y habilidades de los miembros del equipo.	X	X	X		X
	Gestión eficaz del director del proyecto	Habilidades de gestión del director del proyecto.	X	X	X		
		Grado de satisfacción de las autoridades, el equipo de proyecto, los contratistas y demás partes interesadas	X	X		X	X
		Grado de poder de convicción a las autoridades.			X		
		Persona con perfil técnicamente competente y experimentado.	X	X	X		
		Generar espacio para el intercambio de experiencias.			X		
		Disposiciones logísticas ágiles			X		X
	Coordinación y competencias de los miembros del equipo	Definición de responsabilidades del equipo/interesados.	X	X	X	X	
Procesos Internos	Definición de requerimientos, alcance y limitaciones	Precisión y claridad en la definición del alcance y las responsabilidades.	X	X	X		X
	Objetivos claros de negocios	Elaboración de una estructura objetiva.	X		X	X	X
	Políticas y procedimientos de gestión de proyectos	Definición de políticas y procedimientos de gestión de proyectos.	X	X		X	X
	Planificación adecuada	Creación jerárquica de las actividades o tareas.	X	X	X		X
		Cumplir con la calendarización, secuenciación y control de las actividades o tareas.	X	X		X	X
	Optimización	Identificación de los procesos críticos.	X			X	X
	Prácticas ágiles en los procesos	Implementación de flexibilidad en los procesos	X		X	X	X
		Prácticas de logísticas ágiles					
Interesados	Apoyo de la organización	Grado de compromiso de las autoridades con el proyecto.	X	X	X		
		Grado de involucramiento y participación de las autoridades, en el control del cumplimiento de los objetivos			X		
		Participación de autoridades en la definición del proyecto.			X		
		Respaldo de las autoridades al equipo de trabajo.	X				
	Participación de otros interesados	Grado de compromiso y participación de los interesados claves.	X		X		
	Igual percepción entre los interesados y los profesionales de SI	Precisión y claridad en la definición del alcance y las responsabilidades.	X	X			X
		Grado de aceptación del producto final entre	X	X	X		

		los interesados y las autoridades.					
	Cultura de la comunicación	Flujo de comunicación entre las autoridades y el gerente de proyecto.	X			X	X
	Definición clara de los interesados y su vinculación con el proyecto	Definición y descripción de los interesados	X		X	X	
		Identificación de los principales interesados y apoyo a sus necesidades, intereses y capacidades	X	X	X	X	
	Contratistas o patrocinadores comprometidos	Grado de compromiso y satisfacción de los contratistas y patrocinadores	X		X	X	
Técnicas	Gestión de la comunicación	Desarrollo de técnicas de comunicación.			X		
		Reuniones periódicas, claras y concisas.	X		X		X
		Elaborar flujo de comunicación entre los miembros del equipo, el gerente de proyecto y los interesados.	X		X	X	
	Seguimiento y control del proyecto	Técnicas estrictas de seguimiento y control del proyecto.	X	X	X		
		Mecanismos que permitan medir, analizar y mitigar retrasos.	X	X	X		
	Gestión de riesgos	Identificación de riesgos e implementación de dispositivos de gestión de riesgos.	X	X		X	X
	Gestión de la contratación	Entrega de las obligaciones contractuales	X	X		X	
	Tecnología y herramientas adecuadas	Identificación de tecnologías y herramientas aplicables y adaptadas a las necesidades del proyecto y la organización.	X			X	
	Documentación	Técnicas de difusión de políticas de documentación, organización y coordinación del equipo.	X		X		
		Definir políticas que regulen la creación de la documentación y estimulen su consulta.					
		Documentación justa, ni excesiva ni insuficiente.					X

Resultados

Del análisis comparativo que muestra la Tabla 1, podemos afirmar que no hay una metodología que se corresponda completamente a un enfoque en particular y, aún más, que se ajuste totalmente a los requisitos establecidos como prioritarios para una correcta gestión de proyectos de TICs en el sector público. Es decir, evidentemente los requisitos planteados en la Tabla 1, en su mayoría son cumplidos por las metodologías y guías analizadas, pero no en su totalidad. Esto implica que es necesario generar una metodología que corresponda a un enfoque "híbrido", es decir integrando aspectos, características, fortalezas y las mejores prácticas de las metodologías de enfoque tradicional y ágil. Por otra parte, la razón de combinar ambos enfoques es asegurar que todos los requisitos sean considerados por una misma metodología de enfoque "híbrido".

Conclusiones

En el presente artículo se analizaron, las particularidades predominantes del enfoque tradicional y del enfoque ágil de gestión de proyectos, las características del sector público, además de las perspectivas formuladas a partir de dichas características. Luego en la Tabla 1, se presentó el grado de adecuación de las metodologías y guías consideradas, en relación a las perspectivas de gestión de proyectos de TICs en el sector público, derivadas a partir de las particularidades en dicho sector: gobierno, tecnología y gestión, para el cual se establecieron por cada perspectiva una serie de requisitos prioritarios. Del análisis realizado puede concluirse que una metodología no es suficiente y que no debería existir la aplicación de varias metodologías dentro del contexto de la organización ya que conviene que todos los interesados en la

gestión de diversos proyectos de TICs, estén alineados a una misma metodología. Para el caso particular de la gestión de proyectos de TICs en el sector público, es necesaria formular una propuesta metodológica de enfoque "híbrido" a un entono específico de la gestión pública, que armonice los procesos propios, complejos y rígidos, que hacen hincapié en el cumplimiento de procedimientos y normativas, propios de una Administración Pública, con la dinámica y adaptabilidad que ofrece el enfoque ágil de gestión de proyectos, y que caracterizan particularmente a los proyectos de TICs.

Como consecuencia de este estudio, ha surgido como trabajo futuro la formulación de una propuesta metodológica de gestión de proyectos de aplicación a un entorno particular, que integre los puntos fuertes de ambos enfoques, tradicional y ágil. Es decir, dicha propuesta metodológica debe cumplir con todos los requisitos planteados en la Tabla 1, basándose en las metodologías y buenas prácticas en gestión de proyectos estudiadas precedentemente.

6. Referencias

[1] Batley, R., Larbi, G.A. *The changing role of government: The reform of public services in developing countries.* Palgrave Macmillan, Basingstoke, New York. 2004.

[2] Currie W. L., Guah M.W. *Conflicting institutional logics: A national programme for IT in the organisational field of healthcare.* Journal of Information Technology, 22 (3), pp. 235–247, 2007.

[3] Gruening, G. *Origins and theoretical basis of new public management* International Public Management Journal, Artículo, pp. 1–25, 2001.

[4] Packer, Abel. *El rol de las tecnologías de la información y la comunicación TIC en las experiencias de institucionalización de la política social*. CEPAL, Santiago, 2005.

[5] Baranskaya, Anna. *Project management in public administration of transitional countries.* Moscow State University, School of Public Administration, Moscow, Russia, Manuscript,2010.

[6] Andersen, E. S. *Perspectives on projects.* Proceedings of the PMI Research Conference 2006, Canada, 2006.

[7] Boehm, B., Turner, R. *Management challenges to implement agile processes in traditional development organizations. IEEE software*, 22(5):30–39, 2005.

[8] Leffingwell, D. S*caling software agility: Best practices for large enterprises*. Boston, MA: Addison–Wesley, Pearson Education Inc., 2007.

[9] Spundak, M. *Mixed agile/traditional project management methodology – reality or illusion?* Procedia - Social and Behavioral Sciences 119, 939 – 948, 2014.

[10] Williams, T. *Assessing and moving on from the dominant project management discourse in the light of project overruns*. IEEE Transactions on Engineering Management, 52(4), 497–508, 2005.

[11] Conforto, E. C. & Amaral, D. C. *Evaluating an agile method for planning and controlling innovative projects.* Project Management Journal, 33(4), 4–14, 2008.

[12] *Manifesto for Agile Software Development* (2001). http://www.agilemanifesto.org

[13] Hass, K.B. *The blending of traditional and agile project management*. PM world today. Vol. 9, No. 5, pp. 1-8. 2007.

[14] Highsmith, J. *Agile project management: creating innovative products*. 2nd ed. Boston, MA: Addison-Wesley. 432 p., 2010.

[15] Chow, T. Chao. D. *A survey of critical success factors in agile software projects.* Journal of systems and software, 81(6):961–971, Jun. Available: Science Direct, 2008.

[16] Lehtinen, T., Mäntylä, M., Vanhanen, J., Itkonen, J., Lassenius, C. *Perceived causes of software project failures – An analysis of their relationships.* Information and Software Technology 56, 623–643, 2014.

[17] Cserháti, G., Szabó, L. *The relationship between success criteria and success factors in organisational event projects.* International Journal of Project Management 32, 613–624, 2014.

[18] Elkadi, H. *Success and failure factors for e-government projects: Acase from Egypt.* Egyptian Informatics Journal, 14, 165–173, 2013.

[19] Patanakul, P. *Managing large-scale IS/IT projects in the public sector: Problems and causes leading to poor performance.* Journal of High Technology Management Research 25, 21–35, 2014.

Datos de Contacto:

Patricia Cristaldo. Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay, Dpto. Sistemas de Información Ing. Pereira 676, Concepción del Uruguay, Entre Ríos, Argentina. cristaldop@frcu.utn.edu.ar.

Gestión de los Stakeholders en un Proyecto destinado a que la ONU pueda dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones

Uzal, R., Riesco, D., Montejano, G., Sanchez, A., Agüero, W., Baieli,C.
Universidad Nacional de San Luis, Facultad de Ciencias Físico, Matemáticas y Naturales

Abstract
Este trabajo contribuye, parcialmente, a la elaboración de las bases de un Proyecto que posibilite que organismos como las Naciones Unidas o la Unión Internacional de Telecomunicaciones puedan, si resultare necesario, dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones. Estos actos adquieren una gravedad inusitada en el caso de Ciber Ataques a la Infraestructura Crítica del país agredido y en escenarios de Ciber Espionaje masivo. Las distintas Partes Interesadas se han posicionado, respecto de este tema, según puntos de vista sumamente disímiles. Es por ello que, la mayor parte de los Factores Críticos de Éxito de un Proyecto de este tipo están incluidos en la Gestión de los Stakeholders con la salvedad de que, en este caso particular, se trata de un contexto eminentemente Multistaheholder. Investigadores de la Universidad Nacional de San Luis han venido aportando significativamente en los aspectos tecnológicos del emprendimiento en cuestión. En esta presentación, el peso lo está en aportar a un Gerenciamiento adecuado de la interacción de las Múltiples Partes Interesadas. Un consistente y representativo Mapeo de los Stakeholders es precondición para tomar decisiones adecuadas en Proyectos como el que está siendo mencionado.

Palabras clave: Defensa Cibernética, Ciber Agresiones, Ciber Espionaje, Naciones Unidas, Unión Internacional de Telecomunicaciones, Cyber Defense, Cyber Attacks, Cyber Espionage, United Nations, International Communication Union.

Introducción

Los conflictos en el Ciberespacio entre estados naciones constituyen el más importante desafío de la Defensa Nacional. Esto se verifica en prácticamente en todos los países del mundo (1) (2) (3).

Los conflictos en el Ciberespacio entre estados naciones evidencian una clara tendencia a incrementarse cuali-cuantitativamente (4) (5) (6).

Debido a una estrategia "anti control" de la proliferación de Ciber Armas, adoptada y mantenida por estados naciones que han realizado enormes inversiones en "Cyber Warfare", la única opción hoy vigente, para el común denominador de los países, es ejercer el derecho enunciado en el Artículo 51 de la Carta de las Naciones Unidas: Derecho a la autodefensa de los estados naciones (7) (8).

El contexto actual, sin acuerdos y sin controles, es el más propicio para que se produzca una importante proliferación de Ciber Armas muy difícil de ser controlada.

La proliferación de Armas Cibernéticas es potencialmente más perniciosa que la proliferación de Armas Nucleares (3) - ver opinión de los juristas- (9) (10).

El denominado "Problema de la Atribución" de las Ciber Agresiones entre estados naciones posee soluciones diversas (11) (12) (13) (14) (15). Sin embargo hay que reconocerles cierto éxito a quienes han difundido y sustentado, campañas comunicacionales mediante, el mito de la "no solución" del "Problema de la Atribución".

Resulta razonable, conveniente y, además factible, que organismos internacionales tales como las Naciones Unidas y la Unión Internacional de Telecomunicaciones (16) estén en condiciones de dilucidar

responsabilidades en el caso de Ciber Agresiones entre estados naciones.
Los ya mencionados estados naciones que han realizado enormes erogaciones en "Cyber Warfare" y que, actualmente, se oponen a la intervención de organismos internacionales tales como las Naciones Unidas y la Unión Internacional de Telecomunicaciones en casos de Ciber Conflictos, seguramente cambiarán sus opiniones si la proliferación y uso de Armas Cibernéticas mantiene la tendencia actual (17) (18) (19).
Este trabajo sustenta que es razonable y muy conveniente aportar en el sentido de que las Naciones Unidas / Unión Internacional de Telecomunicaciones adquieran las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones. Estos aportes deberían ir generándose a pesar de la temporaria oposición, explícita y/o implícita, de los estados naciones que, como se expresó, en la actualidad están erogando enormes sumas de dinero en Defensa Cibernética y en Espionaje Cibernético (20) (21) (22).

Un Proyecto que confiera a las Naciones Unidas las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones deberá, necesariamente, incluir un muy relevante capítulo relacionado con la "Gestión de los Stakeholders del Proyecto".
Si bien los estándares de Gestión de Proyectos de más amplia aceptación global han incluido al Área de Conocimiento "Gestión de los Stakeholders del Proyecto", el tratamiento dado a dicho tema es, en general, insuficiente tal como para ser aplicado a un proyecto en el cual, la casi la totalidad de sus Factores Críticos de Éxito residan, casualmente, en el ámbito de la "Gestión de los Stakeholders del Proyecto". Es más, el Proyecto al que se hace referencia en este aporte es un claro ejemplo de contexto "Multistakeholder" (23) (24) (25).

Un grupo de Investigadores de la Universidad Nacional de San Luis han venido trabajando en el sentido de aportar a la elaboración de las bases de un Proyecto que confiera a las Naciones Unidas / Unión Internacional de Telecomunicaciones, como ya se citó, las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones, Se estima que esta presentación constituye un subconjunto relevante de dichas "bases", específicamente, en el ámbito de la "Gestión de los Stakeholders". Se destaca que, en otras presentaciones, se han venido reportando aspectos instru-mentales, eminentemente tecnológicos, de este mismo Proyecto en gestación (26) (27) (28).

Sintetizando, la estructura de este trabajo es la siguiente:

- El estilo de Gerenciamiento del Proyecto al que se aporta.
- El estándar "Project Management Body of Knowledge" V 5 del Project Management Institute y la Gestión de los Stakeholders de un proyecto.
- El núcleo de la propuesta contenida en este trabajo
- Mapeo de los Stakeholders: Su finalidad
- Aportes conceptuales e instrumentales: Recapitulación
- Aportes de opiniones de doctorandos en Ingeniería Informática de la Universidad Nacional de San Luis
- Pasos subsiguientes en esta línea de trabajo
- Conclusiones

Para finalizar esta Introducción se destaca que, si bien en todos los trabajos de este tipo, las Referencias son sumamente importantes, en esta caso particular lo autores recomiendan, enfáticamente, un detallado análisis de dichas Referencias. Se estima que

ese análisis brindará un muy buen panorama general a los interesados en el tema.

El enfoque del Gerenciamiento del proyecto objeto de este trabajo

La Gestión de la Calidad Total (TQM es la abreviatura en inglés de Total Quality Management) es un enfoque de gestión de amplia aceptación global (32) (33), estandarizada (34) (35) y de probada efectividad (36) (37) y que, según la experiencia de los autores en Gestión de Proyectos (38) (39), ajusta para ser utilizada en un Proyecto destinado a que las Naciones Unidas puedan dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones.

Una de las características de TQM es que los Clientes son en realidad quienes definen las especificaciones de los Productos y/o Servicios. También son los Clientes quienes evalúan el ajuste de Productos y/o Servicios a las citadas especificaciones.

Como se citó, se entiende por Stakeholder, en una acepción amplia, a cualquier grupo o individuo identificables que pueda afectar el logro de los objetivos de una organización o que es afectado por el logro de los objetivos de una organización (grupos de interés público, grupos de protesta, agencias gubernamentales, asociaciones de comercio, competidores, sindicatos, clientes y segmentos de clientes, accionistas y otros); en una acepción restringida, Stakeholder es cualquier grupo o individuos identificables respecto de los cuales la organización es dependiente para su supervivencia (empleados, segmentos de clientes, ciertos proveedores, agencias gubernamentales clave, accionistas, ciertas instituciones financieras, y otros).

Se ha verificado que, en emprendimientos en el Ciberespacio, los múltiples Stakeholders requieren una dedicación similar a Clientes. Si bien Clientes es un subconjunto de Stakeholders, es muy difícil distinguir entre unos y otros cuando el Ciberespacio es el contexto en el cual se está trabajando.

En la Figura 1 se sintetiza lo expresado; los Multi Stakeholders como diseñadores de Productos / Servicios y también como evaluadores de dichos Productos / Servicios.

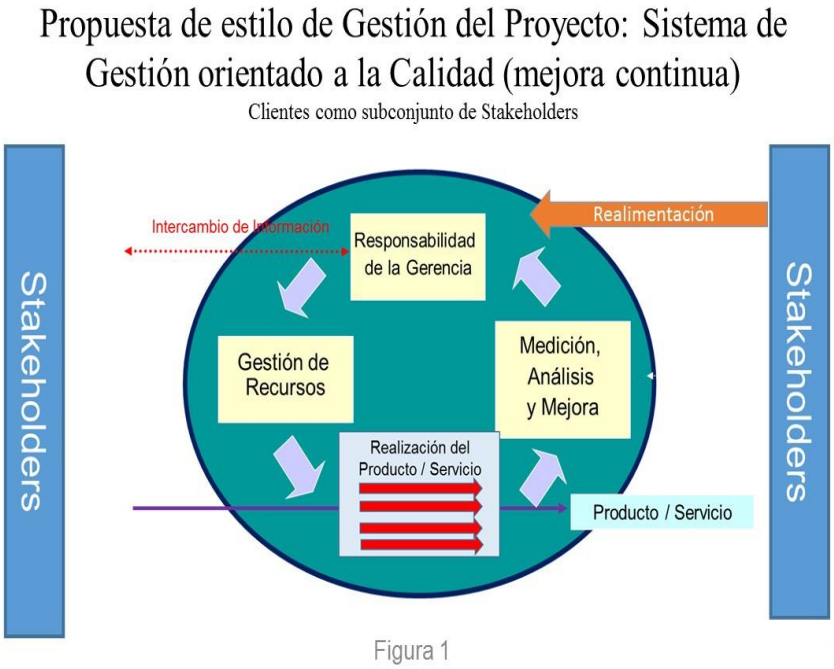


Figura 1

El estándar Project Management Body of Knowledge V 5 del Project Management Institute y la Gestión de los Stakeholders de un proyecto.

La experiencia de los autores ya citada da sustento a la propuesta de utilización, en el proyecto objeto de esta presentación, de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos del Project Management Institute (Versión 5).

La organización de los Grupos de Procesos de la Guía del Cuerpo de Conocimientos para al Gestión de Proyectos permite un esquema iterativo (aproximaciones sucesivas) e interactivo (validación cuasi permanente)

En términos prácticos la organización de los Grupos de Procesos de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos proporciona un ciclo de vida "en espiral" del proyecto modelado.

En la Figura 2 se observan aspectos conceptuales e instrumentales de la PMBOK del PMI.

Una buena representación de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos (PMBOK) del Project Management Institute lo es la Matriz Área de Conocimiento / Grupo de Procesos

La Versión 5 de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos (PMBOK) del Project Management Institute agregó el Área de Conocimiento "Gestión de los Stakeholders del Proyecto"

De acuerdo con la PMBOK V5 del PMI, la gestión de los Stakeholders comprende, tal como se lo observó en la Matriz Grupo de Procesos / Áreas de Conocimiento:

- Identificar los Stakeholders del emprendimientos
- Planear la Gestión de los Stakeholders
- Gestionar el compromiso de los Stakeholders
- Controlar el nivel de compromiso de los Stakeholders

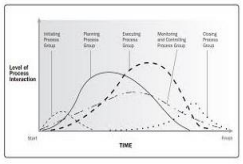
(se puede ver la última fila resaltada de la Matriz desplegada en la Figura 3)

Aspectos generales de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos (PMBOK) del Project Management Institute

Relación entre los Grupos de Procesos
- Procesos de Iniciación
- Procesos de Planeamiento
- Procesos de Ejecución
- Procesos de Control
- Procesos de Cierre



Interrelación entre los Grupos de Procesos en función del tiempo



En la propuesta del PMI subyace un Modelo de Ciclo de Vida en Espiral



Figura 2

10

Matriz Grupo de Procesos / Áreas de Conocimiento de la PMBOK del PMI

	iniciación	Planificación	Ejecución	Monitoreo	Cierre
Integración	Elaborar el documento inicial del Proyecto	Desarrollar el Plan del Proyecto	Gestionar las tareas del Proyecto	Monitoreo y control del Proyecto	Cierre de las tareas del Proyecto
Alcance		Planear el alcance		Validad el Alcance Controlar el Alcance	
Tiempo		Planear la elaboración del cronograma Definir las Tareas Definir la Secuencia Asignar Recursos a Tareas Estimar duración de las Tareas Elaboran el Cronograma		Control programático del Proyecto	
Costo		Planear la Gestión de Costos Estimar los costos Elaborar el Presupuesto		Control Presupuestario	
Calidad		Planear la Calidad	Ejecutar el Aseguramiento de Calidad	Control de la Calidad	
Recursos Humanos		Planear los Recursos Humanos	Convocar al equipo Desarrollar el equipo Gestionar el equipo del Proyecto		
Comunicación		Planear la Comunicación	Gestionar las Comunicaciones	Control de la Comunicación	
Riesgo				Control de los Riesgos del Proyecto	
Abastecimient		Planear el Abastecimiento	Coordinar el Abastecimiento	Control del Abastecimiento del Proyecto	Cierre del Abastecimiento
Stakeholders	Identificar los Stakeholders del emprendimientos	Planear la Gestión de los Stakeholders	Gestionar el compromiso de los Stakeholders	Controlar el nivel de compromiso de los Stakeholders	

Figura 3

El núcleo de la propuesta contenida en este trabajo

La Figura 4 modela gráficamente la "Gestión de los Stakeholders:

Una visión conceptual de "Gestión de los Stakeholders"

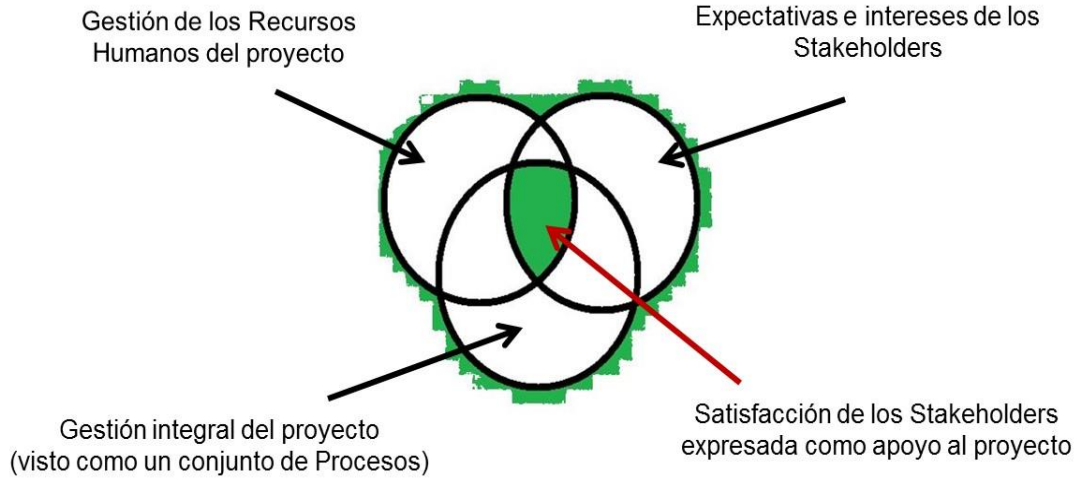


Figura 4

13

El mencionado "núcleo" de la propuesta está integrado por:

- Una visión integral de la Gestión de los Stakeholders
- El estilo de Gestión del Proyecto: Sistema de Gestión orientado a la Calidad (mejora continua)
- La Gestión del Proyecto según un Enfoque Sistémico hacia la Calidad Total
- Identificación de los Stakeholders Clave al Iniciarse el Proyecto
- Modelando un esquema Multistakeholder
- Mapeo de los Stakeholders: Su finalidad
- La representación de las Relaciones entre Stakeholders
- Matriz "inventario" de los Stakeholders del Proyecto

Un enfoque Sistémico en un contexto orientado a la Calidad Total se modela en la Figura 5.

Por otro lado, la Figura 6, muestra un esquema que ayuda a identificar a los stakeholders clave al iniciarse el Proyecto. Se están utilizando, en esta parte del trabajo, conceptos e instrumentos que con anterioridad habían sido empleados por el Gobierno Alemán (42)

La Gestión del Proyecto según un Enfoque Sistémico hacia la Calidad Total
(Proyecto como conjunto de Procesos)

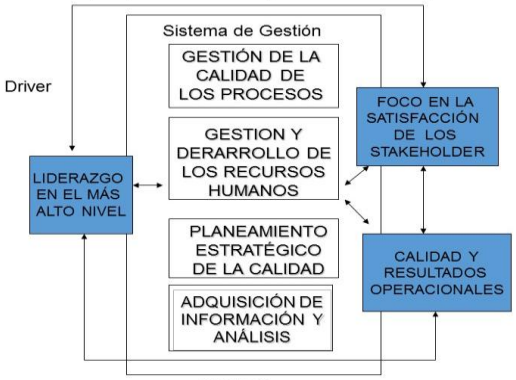


Figura 5

14

Identificación de los Stakeholders Clave al Iniciarse el Proyecto

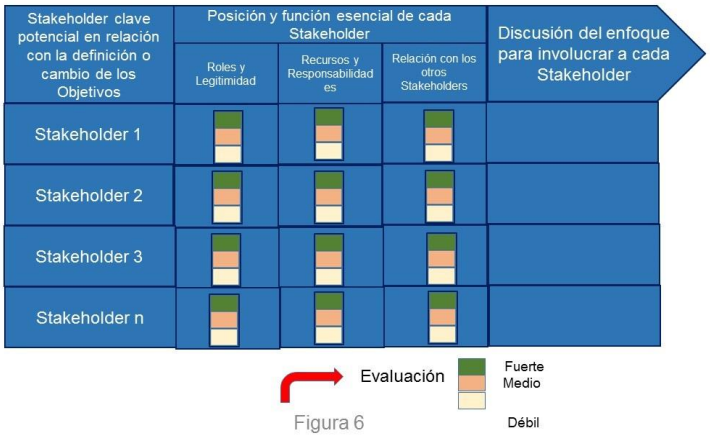
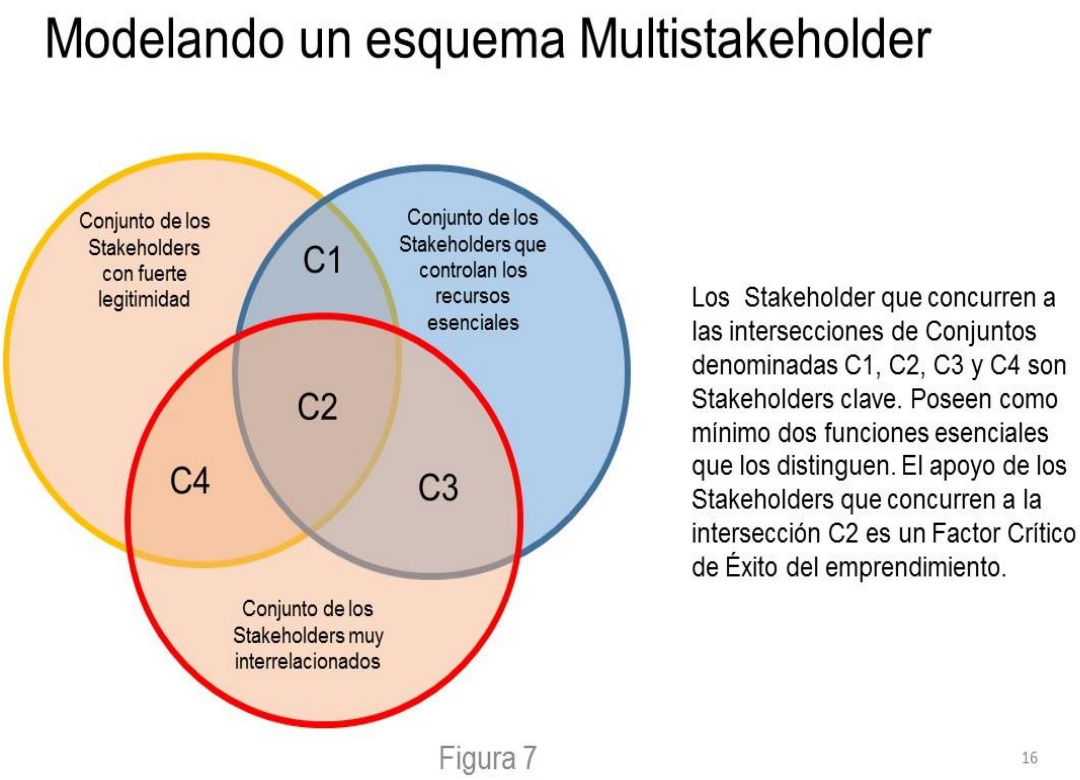


Figura 6

15

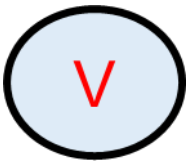
El esquema anterior se complementa con lo modelado en el contenido de la Figura 7:



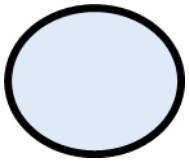
Mapeo de los Stakeholders: Su finalidad

- Visualizar los stakeholders cuya consideración es relevante a los fines del emprendimiento / proyecto.
- Caracterizar las relaciones entre stakeholders y la "red" que queda conformada.
- Localizar los aspectos de interés y los objetivos de cambio en el ámbito de los stakeholders

Los símbolos utilizados en el mapeo:
Como entorno para suministrar ejemplos, se utilizará a la organización y gestión del desarrollo de Netmundial (Sao Paulo, 23/24 de Abril de 2014). El denominado "NETmundial Multistakeholder Statement" (41) es la constancia que logró Brasil en la convocatoria, contiene críticas al espionaje "masivo y arbitrario" y, por otro lado, logró poco consenso en temas clave tale como "neutralidad de La Red". Veamos entonces el uso de la simbología propuesta aplicada a NETmundial



Stakeholder primario con derecho a veto: Brasil, por ser el organizador – anfitrión, en los hechos tuvo este status. EEUU también; sin el visto bueno de EEUU, ICANN / IANA no hubiesen estado presentes en NETmundial. Rusia confundió su rol. Creyó tener derecho a veto y los hechos demostraron lo contrario. Sin el apoyo explícito de las Naciones Unidas NETmundial hubiese carecido de sentido. Las Naciones Unidas tuvieron esta calificación en NETmundial.



Stakeholder primarios son aquellos que tienen una influencia directa sobre el desarrollo del emprendimiento / proyecto. El apoyo de Alemania, Suecia y Francia, por ejemplo, desempeñaron un rol esencial al apoyar las iniciativas de Brasil respecto de Ciber Espionaje Norteamericano



Stakeholder secundario el cual no está directa o evidentemente involucrados pero podrían llegar a influir en el emprendimiento / proyecto. Argentina, Perú, Cuba, India y otras decenas de países, durante NETmundial, tuvieron un rol de Stakeholders secundarios

Líneas llenas: Simbolizan relaciones fuertes en términos de intercambio de información, frecuencia de contactos, compatibilidades de intereses, coordinación, confianza mutua. Relaciones entre países integrantes de Mercosur o de UNASUR son ejemplos de este tipo de vinculaciones.

Líneas punteadas: Simbolizan relaciones débiles o informales. Un signo de pregunta se agrega si la relación no está del todo definida. En el contexto de NETmundial la relación entre Argentina y China puede ser tomadas como ejemplo de esta categoría.



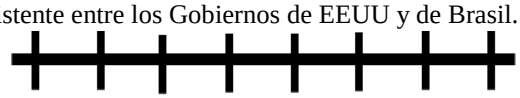
Línea paralelas: Simbolizan alianzas y esquemas colaborativos formalizados contractualmente o sólidamente institucionalizados. Un ejemplo de este tipo de vínculo sería el existente entre los países integrantes de “Five Eyes”



Flechas: Simbolizan la dirección de la relación dominante. Un ejemplo de este tipo de relación sería la existente desde el Gobierno de EEUU hacia ICANN



Líneas llenas cruzadas por un rayo o relámpago: Significan relaciones tensas, con conflictos de interés. Un ejemplo en el contexto de NETmundial sería la existente entre los Gobiernos de EEUU y de Brasil.



Líneas cortas cruzando una línea sólida: Simbolizan relaciones interrumpidas, temporaria o definitivamente. La situación planteada, en NETmundial, entre Rusia, Cuba e India con los organizadores del evento es un ejemplo de este tipo de vínculo

Tipos de Mapeo

Mapeo según un esquema del tipo “capas de cebolla”, tal como se ejemplifica en la Figura 8

El Mapeo de Stakeholders según “Capas de Cebolla”

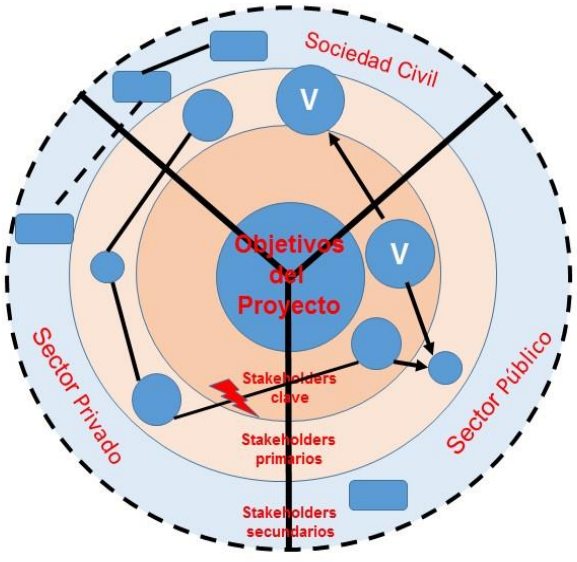


Figura 8

20

Alternativamente se dispone del esquema del tipo “arco iris”.Un ejemplo de este tipo de Mapeo se exhibe en la Figura 9

El Mapeo de Stakeholders tipo Arco Iris

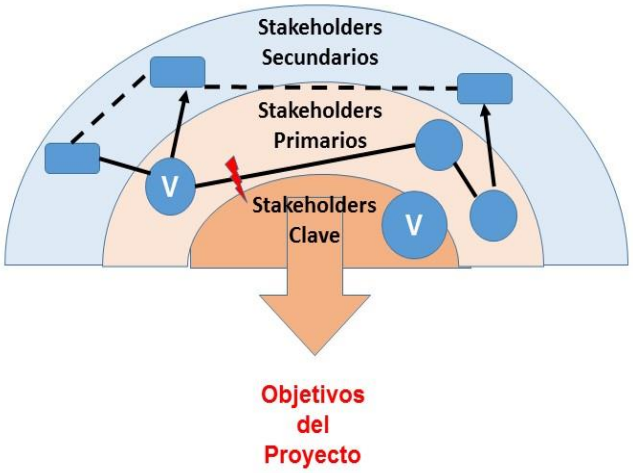


Figura 9

21

Sea cual fuere la variante de Mapeo que se elija, siempre es muy conveniente contar con un “inventario” de los Stakeholders del Proyecto. Se sugiere un formato matricial al respecto en la Figura 10.

Matriz “inventario” de los Stakeholders del Proyecto

Nombre del Stakeholder y rol fundamental	Objetivos estratégicos detectados	Grado de influencia y alcance de la misma	Alianzas con otros stakeholders en téminos de A B C D
Stakeholder 1			
Stakeholder 2			
Stakeholder 3			
Stakeholder n			

A Interdependencia regulada institucionalmente
B Intercambio de Información en proceso
C Acción coordinada
D Co-producción utilizando recursos comunes

Esta matriz complementa el Mapeo concretado según un enfoque “Capa de Cebollas” o “Arco Iris. El uso periódico de esta matriz durante el proyecto contribuirá a la Gestión de Stakeholders, específicamente en el Monitoreo / Control de la evolución de la relación de los Stakeholder con el proyecto

Figura 10

22

Aportes conceptuales e instrumentales: Recapitulación

- Se ha propuesto una visión integral de la Gestión de los Stakeholders
- Se detallaron de manera adecuada los aspectos generales mencionados en la Guía del Cuerpo de Conocimientos de la Gestión de Proyectos (Versión 5) del Project Management Institute
- Se ha presentado un estilo especialmente adecuado para la Gestión de un Proyecto con un fuerte "peso" Multistakeholder: Sistema de Gestión orientado a la Calidad (mejora continua)
- Se proporcionó un modelo respecto de la esencia de la Gestión de un Proyecto, con un fuerte "peso" Multistakeholder, según un Enfoque Sistémico orientado hacia la Calidad Total
- Se dio al enfoque Multistakeholder un tratamiento eminentemente sistémico.
- Se han suministrado conceptos e instrumentos para la identificación y catalogación de los Stakeholders Clave al Iniciarse el Proyecto
- Se instanció al caso tratado en esta presentación el enfoque del Dr Arthur Zimmermann y la Dra Claudia Maennling, editado en 2007 por
- Gabriele Geißler del Strategic Corporate Development Department, Policy and Strategy del Gobierno Alemán (42)
- Se ha propuesto un Modelo del significado de un esquema Multistakeholder
- Se han propuesto esquemas alternativos para el Mapeo de los Stakeholders
- Se han propuesto esquemas de catalogación y representación de las Relaciones entre Stakeholders
- Se ha propuesto una Matriz para modelar el "inventario" de los Stakeholders del Proyecto y para concretar el seguimiento del Proyecto.
- Se suministraron ejemplos referidos a NETmundial
- Se suministró un enfoque integral para dar soporte al aspecto más sensitivo de un futuro Proyecto destinado a que las Naciones Unidas puedan dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones

Aportes de opiniones de doctorandos en Ingeniería Informática de la Universidad Nacional de San Luis

Intervinieron los doctorandos Abdelahad, Corina; Agüero, Walter; Baieli, Claudio; Baigorria, Lorena; Bast, Silvia; Farías, Andrés; García, Pablo; Miranda, Enrique; Peralta, Mario; Salgado, Carlos.
Este equipo está llevando a cabo un minucioso análisis de los stakeholders vinculados a un proyecto del tipo como el que habría que encarar para que "La ONU pueda dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones".
En posteriores presentaciones se abundará en detalles sobre el avance del proyecto. A efectos representativos, como Figura 11, se muestra uno de los borradores (incompleto y no validado) utilizados en las discusiones entre doctorandos de la UNSL.

Pasos subsiguientes en esta línea de trabajo

Dichos pasos subsiguientes se dividen en:

- Aspectos Técnico Informáticos que comprenden detección de Ciber Agresiones mediante Análisis de Flujo de Redes e Ingeniería Reversa de Ciber Armas o instrumentos de Cíber Espionaje.

- Aspectos relacionados con la Gestión de Proyectos de Alta Complejidad en un contexto Multiskateholder.

Ejemplo de borrador de Mapeo de Stakeholders preliminar (incompleto y no validado) utilizado en las discusiones entre los doctorandos de la UNSL

1 Naciones Unidas
2 Unión Internacional de Comunicaciones
3 ICANN
4 IANA
5 EEUU
6 Rusia
7 China
8 Corea del Norte
9 Reino Unido
10 Five Eyes
11 Brasil
12 Francia
13 Alemania
14 Irán
15 Argentina
16 Ecuador
17 India
18 Cuba

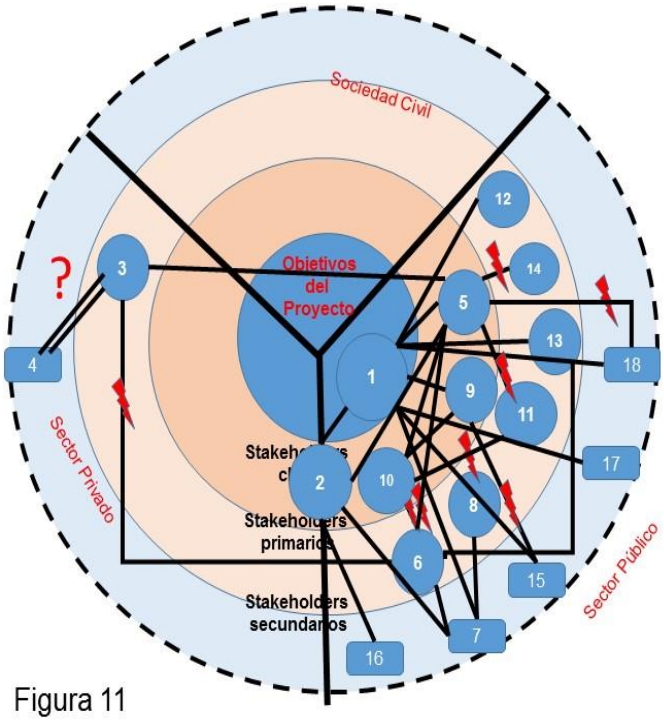


Figura 11

En lo que hace los aspectos Técnico Informáticos, dos alumnos de la Maestría en Ingeniería de Software de la Universidad Nacional de San Luis, concluirán, en los próximos meses, sus Tesis de Maestría en los laboratorios de la Universidad Federal de Minas Gerais – Belo Horizonte . Brasil. Las dos Tesis constituyen aportes de sustanciales en el campo de la detección de Ciber Agresiones.

Por otro lado, en lo que hace a la Gestión de Proyectos de Alta Complejidad en un contexto Multiskateholder, los doctorandos en Ingeniería Informática de la Universidad Nacional de San Luis, continuarán aportando de manera de llegar a contar un Mapeo de los Stakeholders que realmente llegue a ser de utilidad a organismos como la Organización de las Naciones Unidas o la Unión Internacional de Comunicaciones.

La idea es permanecer en contacto con organismos como los mencionados de manera de poder aspirar a realizar un aporte con impacto global.

Conclusiones

- Es posible y, además muy conveniente, que las Naciones Unidas y la Unión Internacional de Comunicaciones pudiesen dirimir responsabilidades en el caso de registrarse Ciber Agresiones entre estados naciones. La actual situación de “cuasi todos contra todos” no es buena para estado nación alguno. “Poder Militar” es un concepto eminentemente asimétrico en Defensa Cibernética. Los países

"convencionalmente" más poderosos deberían ser los que mayor soporte otorgaran a un emprendimiento como el que se menciona.

- Los autores están convencidos de que, pese a la actual oposición de países que han efectuado inversiones enormes en Cyber Warfare, en algún momento habrá que encarar un Proyecto que confiera a las Naciones Unidas las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones. Es bueno ir preparándose para ese momento.
- A pesar de la relevancia tecnológica de un proyecto como el citado, la casi totalidad de los Factores Críticos de Éxito del emprendimiento están relacionados con la Gestión de los Stakeholders (con orientación Multistakeholders)
- Un estilo de Gestión del tipo TQM se evidencia como el más adecuado en un Proyecto que confiera a las Naciones Unidas las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones.
- La Guía del Cuerpo de Conocimientos para la Gestión de Proyectos (PMBOK) del Project Management Institute (Versión 5) proporciona un excelente marco de referencia para Gerenciar un Proyecto de este tipo; sin embargo es necesario complementar los aspectos relacionados con Gestión de Stakeholders. Este trabajo apunta en esa dirección.
- Casualmente se exponen en esta presentación conceptos y herramientas para potenciar los contenidos de la Guía del Cuerpo de Conocimientos para la Gestión de Proyectos (PMBOK)
- Dichos conceptos y herramientas adicionales al PMBOK serán utilizados por los autores en los próximos pasos de la elaboración de las bases de un Proyecto que confiera a las Naciones Unidas las incumbencias y los instrumentos para dirimir responsabilidades en casos de Ciber Agresiones entre estados naciones. Dicha elaboración constituye un objetivo altamente prioritario en la Línea de Investigación en la que se desempeñan los autores.

Datos de Contacto:
Roberto Uzal, Daniel Riesco, Alberto Sanchez. Universidad Nacional de San Luis. Ejercito de los Andes 950 Bloque 2 Primer piso. Informática.
{ruzal | driesco | gmonte | alfanego}@unsl.edu.ar
Walter Agüero. waguero@sldigital.com.ar
Claudio Baieli. cbaieli@hotmail.com

Referencias

(1) http://www.cambridge.org/us/academic/subjects/law/humanitarian-law/tallinn-manual-international-law-applicable-cyber-warfare,Tallinn Manual on the International Law Applicable to Cyber Warfare, Cambridge University Press, 2013
(2) http://www.ccdcoe.org/publications/books/Peacetime-Regime.pdf Peacetime Regime for State Activities in Cyberspace, Cooperative Cyber Defense Centre of Excellence, NATO, Tallinn, 2013
(3) https://www.youtube.com/watch?v=vrRj-kRofRg , video de la Televisión Iraní acerca del ataque a la plataformas petroleras mediante la Ciber Arma Flame
(4) P.W. Singer, Allan Friedman, "Cybersecurity and Cyberwar: What Everyone Needs to Know", Oxford University Press, 2014
(5) Joint Forces Staff College Joint Advanced Warfighting School, "The Cyber Military Revolution and the Need for a New Framework of War" Washington DC, 2014
(6) https://www.youtube.com/watch?v=wFheOjvxoP0 , State of the Union 2013: El Presidente B. Obama trata el tema de las crecientes Ciber Amenazas
(7) R. Uzal et al, "Informe al Ministro de Educación de Argentina sobre NETmundial –

23 / 24 de abril Sao Paulo – Brasil". Pedir copia del informe, si existe interés, a ruzal@unsl.edu.ar

(8) R. Uzal et al, "Informe al Secretario de Comunicaciones de Argentina sobre NETmundial – 23 / 24 de abril Sao Paulo – Brasil". Pedir copia del informe, si existe interés, a ruzal@unsl.edu.ar

(9) http://thediplomat.com/2011/10/the-cyber-proliferation-threat/ The Cyber Proliferation Threat, "The Diplomat", 2011

(10) http://armscontrolnow.org/2013/05/30/is-there-a-place-for-nuclear-deterrence-in-cyberspace/ Timothy Farnsworth, "Is There A Place For Nuclear Deterrence in Cyberspace?" Mayo 30, 2013

(11) Uzal, Roberto y otros, "Conflictos en el Ciberespacio entre estados naciones", JAIIO, 2014

(12) https://www.dropbox.com/s/2cldzph3nydmpzu/ApunteCiber14.pdf Simposio de Ciber Defensa llevado a cabo en la Facultad de Ingeniería del Ejército Argentino el 21 y 22 de mayo de 2014

(13) http://resources.infosecinstitute.com/attribution-problem-in-cyber-attacks/

(14) https://iseclab.org/papers/disclosure.pdf

(15) http://www.ccdcoe.org/publications/2012proceedings/2_5_Hare_TheSignificanceOfAttribution.pdf

(16) http://www.itu.int/es/about/Pages/default.aspx Sitio Web de la Unión Internacional de Telecomunicaciones

(17) http://www.washingtontimes.com/topics/us-cyber-command/

(18) http://www.washingtontimes.com/topics/us-cyber-command (Cyber Pearl Harbor)

(19) http://www.nytimes.com/2012/10/12/world/panetta-warns-of-dire-threat-of-cyberattack.html?pagewanted=all&_r=0 (advertencia del Secretario de Defensa de EEUU)

(20) http://www.cnbc.com/id/101411741 Reuters: "US unveils measures to avoid 'cyber Pearl Harbor"

(21) http://www.cfr.org/cybersecurity/threat-cyber-pearl-harbor-potent-some-have-suggested/p30863 Council on Foreign Relations: "Is the threat of a "cyber Pearl Harbor" as potent as some have suggested?"

(22) http://www.telegraph.co.uk/news/worldnews/northamerica/usa/9604794/US-at-risk-of-cyber-Pearl-Harbor-Leon-Panetta-warns.html The Telegraph: "US at risk of cyber-Pearl Harbor"

(23) http://icannwiki.com/index.php/Multistakeholder_Model ICANN: ¿Qué es un Modelo Multistakeholder?

(24) http://www.netmundial.org/ Ejemplo de contexto Multistakeholder: NETmundial

(25) http://www.apc.org/en/glossary/term/275 APC: Otra propuesta de significado de Multistakeholder

(26) Uzal, R. "Análisis de Flujo de Redes para la detección de Amenazas Cibernéticas" – Simposio desarrollado en la Facultad de Ingeniería del Ejército Argentino (02 de abril de 2014) - solicitar apuntes y transparencias a ruzal@unsl.edu.ar

(27) http://content.netmundial.br/files/61.pdf Contenidos de NETmundial

(28) http://43jaiio.sadio.org.ar/sites/default/files/Programa%2043%20JAIIO.pdf Programa de las Jornadas Argentinas de Informática 2014 – Simposio Informática y Estado – "Conflictos en el Ciber Espacio entre estados naciones: Potenciales aportes para la eventual actuación de las Naciones Unidas", Roberto Uzal, Daniel Riesco, Germán Montejano, Claudio Baieli, Walter Agüero (Universidad Nacional de San Luis)

(29) http://www.pmi.org/PMBOK-Guide-and-Standards.aspx Sitio del Project Management Institute – ver Project Management Body of Knowledge V 5

(30) http://www.businessdictionary.com/definition/stakeholder-management.html Definición de "Stakeholders Management"

(31) http://www.pmi.org/Knowledge-Center/Knowledge-Shelf/Stakeholder-Management.aspx "Stakeholders Management" según el Project Management Institute

(32) http://www.businessdictionary.com/definition/total-quality-management-TQM.html (definición en el Diccionario de Negocios)

(33) http://www.tuobra.unam.mx/publicadas/040119150618-Gesti_oa.html (enfoque de la UNAM)

(34) http://www.iso.org/iso/iso_9000 (Web oficial de los estándares ISO)

(35) http://www.iram.org.ar/ IRAM – ISO en Argentina – Sitio Web

(36) http://www.sedici.unlp.edu.ar/handle/10915/18267/browse?authority=650&type=author

(37) http://www.researchgate.net/profile/Roberto_Uzal

(38) http://www.pminuevocuyo.org/presentaciones/100916%20Uzal.pdf

(39) http://www.pminuevocuyo.org/presentaciones/100605uzal.pdf
(40) http://sedici.unlp.edu.ar/bitstream/handle/10915/22234/Documento_completo.PDF?sequence=1
(41) http://netmundial.br/es/netmundial-multistakeholder-statement/
(42) Gobierno Alemán - Multi-stakeholder management: Tools for Stakeholder Analysis: 10 building blocks for designing participatory systems of cooperation https://www.k4health.org/toolkits/km/multi-stakeholder-managementtools-stakeholder-analysis-10-building-blocks-designing

Una Plataforma para Facilitar la Comprensión del Vocabulario y las Reglas de Negocio en el Desarrollo de Software

Ing. Guillermo Pantaleo
Profesor Adjunto, Dto. Computación, Facultad de Ingeniería, UBA
gpantaleo@fi.uba.ar
Ing. Ignacio Vazquez
Tekii SRL
ivasquez@gmail.com

Abstract

En este trabajo se propone un proceso de construcción del Modelo de Dominio del software de un sistema en desarrollo y las herramientas con las cuales realizarlo, en adelante nombrada "plataforma". La misma permite acortar y facilitar las tareas de construcción del modelo de objetos que representa el negocio sobre el cual actuará la aplicación en desarrollo, en adelante nombrado como Modelo de Dominio. El objetivo de esta plataforma es acelerar el aprendizaje del negocio, achicando así el riesgo más importante de un proyecto de desarrollo de software. Fue pensada para que un analista genere la entrada escribiendo texto estructurado guiado por un editor que realiza sugerencias y coloreado del texto, según el estándar SBVR (Semantics of Business Vocabulary and Business Rules). Este componente constituye el DSL (Domain Specific Lenguage) de la plataforma. La misma procesa el texto de entrada, lo valida contra el estándar y genera modelos de clases, patrones y dinámica de validación de las reglas en forma automática. La salida provista es la visualización de los modelos, constituyendo esta funcionalidad una herramienta poderosa de análisis. Un validador lógico implementado en Prolog valida el modelo generado. Se presentan los criterios con los cuales fueron concebidas y diseñadas cada parte del sistema implementado y se muestra el uso de la plataforma.

1. Introducción

Una cuestión clave en el proceso de desarrollo de software es la separación de, los objetos y las actividades, asignadas a la Aplicación y al Modelo de Dominio. La misma es de fundamental importancia debido a que los requerimientos son diferentes y provienen de fuentes distintas; además del diferente contexto y grado de re uso de sus componentes. En el primer caso son funcionales y expresan los deseos y necesidades de los usuarios a partir de la información presentada en interfaces de usuario, la navegación de las mismas y las tareas de negocio que la Aplicación permite llevar a cabo. Los otros son de negocio y provienen del negocio mismo, expresan conceptos del dominio del problema representados por entidades, sus relaciones y su dinámica.

Este Modelo de Dominio más los test que lo prueban está orientado a dar respuesta a la primera de las preocupaciones de un grupo de desarrollo; entender el negocio. Por esta razón un componente importantísimo es la representación de las reglas de negocio en dicho modelo. En [1] Jill Nicola y sus colaboradores presentaron un catálogo de patrones de análisis para la construcción de estos modelos. Estos patrones están compuestos por pares de clases relacionadas que colaboran en la validación de las reglas de negocio; de allí que también son conocidos como Patrones de Colaboración. Además de estos patrones, presentaron una estrategia de asignación de las reglas a las clases que los componen y una clasificación de reglas de negocio en categorías. La plataforma que se presenta en este artículo utiliza a dicho trabajo como referencia en lo que respecta a la estructura del Modelo de Dominio y su dinámica. Es decir un modelo de objetos que representa las entidades del dominio y que tiene la validación de las reglas de negocio asignadas; el cual será utilizado como parte del sistema en desarrollo para la representación del negocio [4]. Eric Evans se refiere a este concepto en su libro Domain Driven Design [17] en el cual además propone un proceso de desarrollo asociado, el cual está conducido por el conocimiento de este negocio. La comprensión del dominio y reglas de negocio del software de un sistema en desarrollo se obtienen a partir del relevamiento realizado por un analista en lenguaje natural. La ambigüedad inherente al lenguaje natural se trató de acotar y controlar con la utilización del estándar SBVR

[6], el cual aporta una estructura de significado que se vale de lógica de primer orden y lenguaje natural controlado. En este trabajo, para editar el texto que expresa el vocabulario del negocio y sus reglas se construyó un editor que hace sugerencias y autocompletado a la vez que consulta a un parser cuya gramática se construyó a partir de expresiones elaboradas según las definiciones de SBVR.

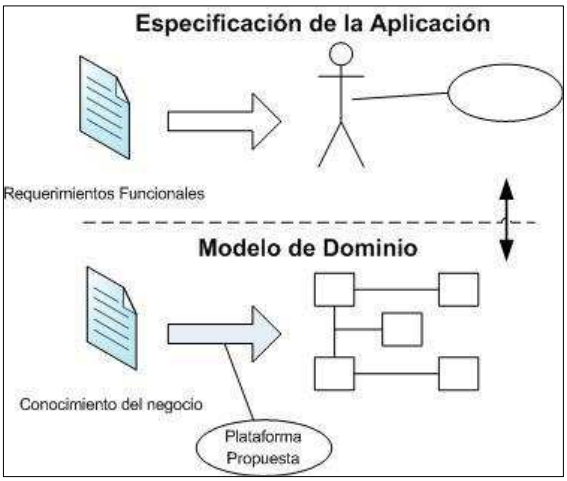


Figura 1 – Esquema donde se observan los componentes de la Aplicación y del Dominio de un sistema en desarrollo y el contexto de funcionamiento de la plataforma propuesta.

En la figura 1 se muestra esquemáticamente el contexto de funcionamiento de la plataforma propuesta.
En la próxima sección se presenta la plataforma implementada en relación con trabajos ya publicados y en las secciones siguientes se describe cada uno de los módulos que la componen. Nuestra plataforma tiene componentes de Programación Generativa [7]; en este caso el Espacio del Problema está constituido por el vocabulario y la reglas de negocio que son procesadas por el parser de SBVR el cual constituye el DSL para este caso. El Espacio de la Solución está compuesto por el visualizador y las vistas de los modelos asociados que son generados por el componente de Conocimiento de Configuración representado por el generador del Modelo de Dominio. Si bien desde el punto de vista estructural este mapeo con la disciplina es correcto, en nuestro caso no hemos realizado un Análisis del Dominio ni Modelos de Características [7] a efectos de determinar los posibles dominios y sus alcances, sobre los cuales nuestra plataforma permite realizar un trabajo de análisis de requerimientos para el desarrollo de sistemas. Sin embargo los componentes generados son reutilizables y aplicables a una gran cantidad de problemas diferentes y es una solución al problema originalmente planteado: contar con una herramienta de ayuda para el análisis de requerimientos en el desarrollo del software de un sistema. A efectos de generalizarla para su utilización en el desarrollo de una familia de sistemas se debería extender el trabajo con los componentes mencionados.

2. Plataforma Implementada

El sistema implementado consta de cuatro módulos bien diferenciados que son:

- Editor de SBVR
- Parser de SBVR
- Generador del Modelo de Dominio
- Visualizador
- Validador

En la figura 2 se muestra un diagrama esquemático del sistema que se describe a continuación. Ha habido diferentes propuestas de generar modelos de tipo UML a partir de lenguaje natural en forma previa a la utilización de SBVR [13, 14]. A partir de la aparición del estándar se han realizado distintos trabajos orientados a la generación de modelos a partir de lenguaje natural utilizándolo como marco de especificación. Algunos de ellos propusieron sistemas cuya entrada son requerimientos de software expresados en lenguaje natural cuyo procesamiento detecta construcciones compatibles con el estándar; generando modelos de clases en UML [12, 16]. En [8] además se procesan las reglas y se las traslada a diagramas de actividad. También se generan Object Constraint Language (OCL) a efectos de especificar requerimientos no funcionales que acompañan a los modelos UML [15]. Otras propuestas [11], se valen de un editor que conduce la edición del texto que describe el negocio, filtrando las entradas no compatibles con el solo propósito de producir una documentación del vocabulario y las reglas según el estándar. Nuestra propuesta utiliza un editor del estilo publicado en [11] a efectos de alimentar a la plataforma con una entrada SBVR compatible que de todas formas es validada por un parser. La salida del parser es introducida en el generador del Modelo de Dominio cuya salida son clases relacionadas según los Patrones de Colaboración [1] con las reglas de negocio asignadas a dichas clases además de diagramas que muestran la propagación de las validaciones de las reglas a través de las colaboraciones. Un validador lógico valida la estructura y comportamiento del modelo. Un módulo de visualización muestra en forma gráfica todas estas salidas.
En la sección que sigue se presenta el editor que forma parte de la plataforma.

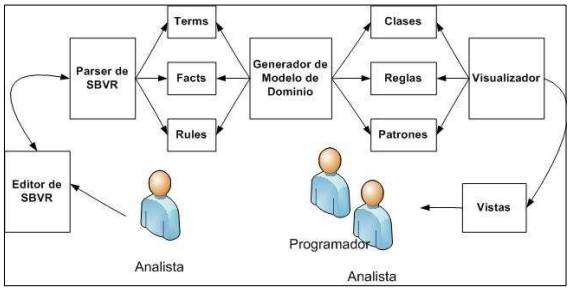


Figura 2 – Diagrama esquemático de la plataforma implementada donde se observan los diferentes componentes y su interacción.

3. Editor de SBVR

Este componente está orientado a resolver el problema de la ambigüedad inherente del lenguaje natural mediante el cual los analistas expresan los requerimientos del negocio cuando lo relevan. Hace algunos años fue presentado el framework SBVR [6] con el cual los expertos de negocios pueden comprender, validar y expresar vocabulario y reglas. La función del editor propuesto es facilitar al analista la escritura según este estándar. Para lograrlo trabaja en conexión con el parser, validando y proponiendo posibles entradas basadas en el texto que está siendo editado. Esta funcionalidad se implementó utilizando técnicas conocidas como autocompletado. Presentamos a continuación el formato de las reglas según SBVR, las pautas para el diseño, el criterio de selección de la tecnología utilizada, la comunicación con el parser y ejemplos del funcionamiento.

3.1. Formato de Reglas

El estándar SBVR es una estructura de tipo declarativa para expresar el conocimiento de negocios en lenguaje natural controlado (CNL - Controlled Natural Language) [19]. Está compuesta de nombres (terms), verbos, hechos (fact type) y reglas (rules). Para más detalle se puede consultar [6]. Las reglas se componen a partir de un hecho previamente definido que a su vez se compone de nombres previamente definidos. En la figura 3 se muestra esquemáticamente la composición de una regla.
El estándar incluye un código de colores que el editor debe respetar así como el tipo de letra y adornos de subrayado. En la figura 4 se muestra un ejemplo de una regla expresada según SBVR.

3.2. Pautas para el diseño

El editor tiene como característica principal brindar asistencia al usuario durante la redacción de documentos conforme al estándar SBVR. Sólo es una herramienta que facilita al analista la expresión de las reglas de negocio en un lenguaje conciso, con el propósito de evitar la mayor cantidad de ambigüedades posibles. Por estas razones, gran parte del esfuerzo de desarrollo fue dedicado a la correcta identificación y coloreado de la sintaxis de SBVR y a la funcionalidad de auto-completado.

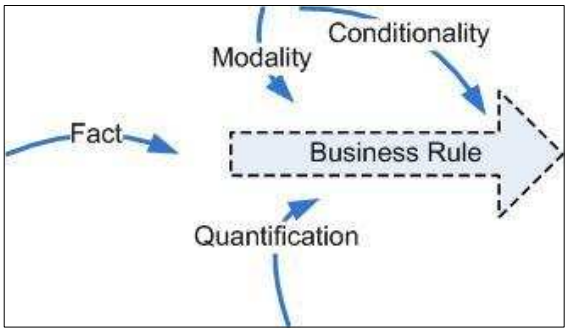


Figura 3 – Composición de las reglas de negocio utilizando SBVR; en estas construcciones se basa el parser para detectar y validar la entrada desde el editor.

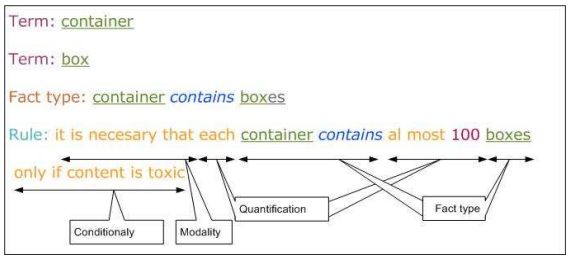


Figura 4 – Ejemplo de regla de negocio expresada según SBVR

3.3. Selección de la plataforma de edición

El editor desarrollado está categorizado como un "editor avanzado de texto" porque ofrece una gran cantidad de funcionalidades a diferencia de otros que sólo presentan un área en blanco para escribir contenido. Por esto es mucho más cercano a los editores utilizados para programación que a los productos estándar de oficina. Dentro del conjunto de editores avanzados de texto pueden diferenciarse dos grandes grupos. En el primero se encuentran aquellos programas instalables en las computadoras de los usuarios (que podemos llamar programas de tipo "estándar"), mientras que en el segundo se ubican distintas aplicaciones ejecutables desde un navegador web, cuyos datos pueden provenir de un servidor, de un servicio alojado en alguna "nube" o de archivos locales. Como ejemplos del primer grupo se

encuentran desde editores como Vim, Emacs o Sublime Text hasta entornos de desarrollo integrados (IDE según sus siglas en inglés) como Eclipse o NetBeans. Todos ellos presentan ventajas como la integración con el sistema operativo y facilidad para instalar distintos complementos que agregan funcionalidades según las necesidades propias de cada usuario. Dentro del segundo grupo aparecen distintas aplicaciones relativamente nuevas, tales como Rulemotion [11], Adobe Brackets, CodeMirror (proyecto sobre el que está basado Rulemotion) y Ace [20]. El desarrollo de complementos para validaciones, en base a métodos consultados por medio de servicios web o cualquier otro tipo de servicio en un servidor resulta más sencillo en este caso dado que los editores se ejecutan dentro de un contexto de tipo cliente/servidor.
El requerimiento principal al momento de decidir qué editor utilizar como base del proyecto fue la necesidad de acceder a servicios de validaciones publicados por un parser desarrollado en Java. Luego de evaluar las capacidades, ventajas y desventajas de cada opción, se resolvió optar por Ace por las siguientes razones:

- Es un proyecto Open Source.
- Hay mucha participación y actividad de la comunidad que lo mantiene, lo que facilita obtener soluciones para los diferentes problemas que se puedan encontrar.
- Es el más completo de todos los editores web evaluados al momento ya que no está orientado sólo a desarrolladores de front-end o diseñadores web (como Adobe Brackets).
- Puede incorporarse a cualquier aplicación web, lo que simplifica la interacción con el parser.

Para poder analizar el texto que ingresa el usuario, Ace requiere la definición de un Modo [21] que establezca las reglas del lenguaje a utilizar. El Modo de un lenguaje indica cómo se debe iterar el texto para realizar el coloreado de la sintaxis y contiene las reglas de auto-completado.

3.4. Comunicación con el parser

La integración del editor y el parser requirió la definición de una API (Interfaz de Programación de Aplicación) que se muestra en la figura 5.

```
interface ParserService

//Método que recibe un texto a validar
int linesParsing(String lines);

//Colecciones construídas a partir del texto  //validad
Map<String, String> getTerms();
Map<String, String> getFacts();
Map<String, String> getVerbs();
Map<String, String> getRules();
Map<String, String> getErrors();

//Colecciones de elementos fijos
List<String> getModalities();
List<String> getQuantifications();
List<String> getConditions();
List<String> getRelations();
List<String> getSpecialFacts();
```

Figura 5 - Interfaz para consulta del parser

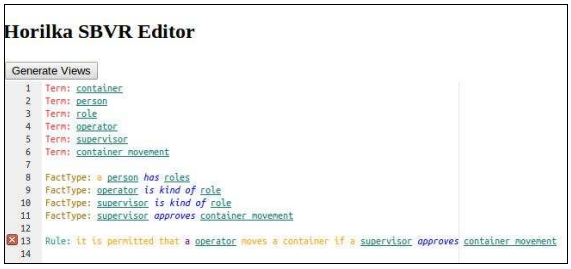


Figura 7 – Pantalla del editor mostrando segmentos de texto reconocido de forma correcta y un error debido a la falta de la definición de un hecho utilizado en la regla editada.

Permite enviar un texto desde el editor al parser, para que este último lo analice y responda enviando las colecciones reconocidas como texto SBVR válido y los errores identificados. El parser también devuelve los componentes estáticos del estándar SBVR como listas de elementos (modificadores, cuantificadores, condicionales, etc.). Por otra parte, los errores y las colecciones variables, o sea aquellas que se construyen a partir de lo escrito por el usuario, se devuelven como mapas de "clave => valor" en los que la clave es el número de línea del texto consultado y el valor es o el atributo válido o el error detectado.
Ace sufre de las limitaciones de toda aplicación web implementada en HTML y JavaScript: todo el código JavaScript se ejecuta a intervalos en un único proceso (o hilo de ejecución).

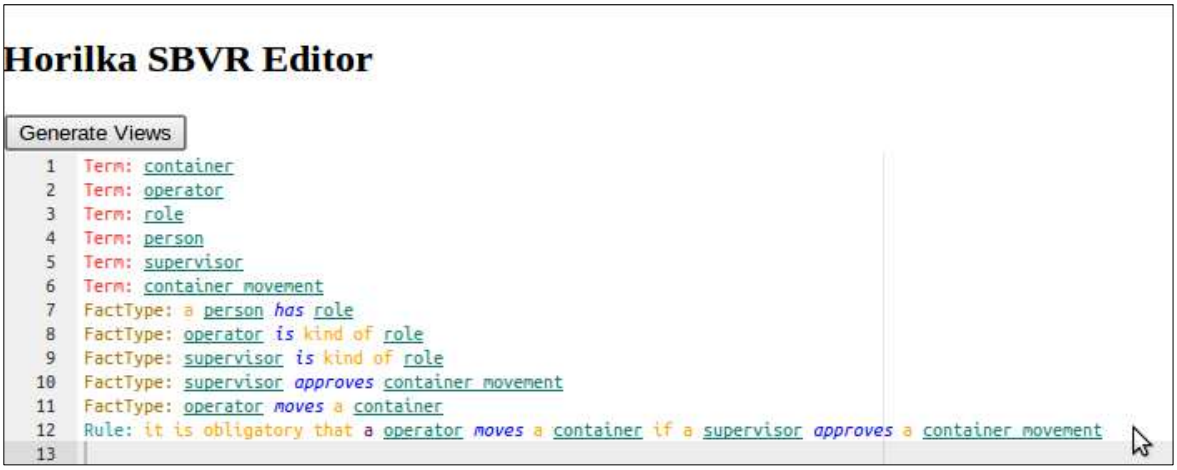


Figura 6 – Pantalla del editor mostrando segmentos de texto reconocido de forma correcta y formateado según el estándar SBVR.

Para reducir la carga de trabajo del proceso principal del navegador, Ace propone el uso de Web Workers [22] que realicen tareas a intervalos regulares de tiempo. Los Web Workers son una tecnología reciente en los navegadores web que consiste en generar hilos de ejecución que corren en segundo plano. De esta manera el proceso principal queda disponible para atender los eventos del usuario y la aplicación puede aprovechar los entornos de hardware paralelizados. En nuestro caso de uso, a medida que el analista ingresa un texto, el editor lo envía al parser por medio del Web Worker y al recibir la respuesta puede generar realimentación visual para el usuario con el fin de indicarle posibles errores.

3.5. Pruebas de funcionamiento

La figura 6 muestra un caso de texto SBVR reconocido por el editor y el parser de manera correcta. Los nombres, hechos y reglas se colorean según el estándar [6].
Para simular un caso de error en el análisis realizado por el parser alcanza con eliminar términos del texto como se muestra en la figura 7. De esta manera, el parser falla porque encuentra hechos que utilizan nombres que no fueron definidos con anterioridad. Los errores del parser se reflejan en la interfaz del editor como marcas junto al número de línea correspondiente. Si el usuario coloca el puntero sobre las marcas podrá ver un mensaje de error devuelto por el parser. Además, el resto del texto deja de colorearse de forma correcta porque dejan de identificarse combinaciones válidas.

3.6. Sugerencias y autocompletado

El procesamiento de sugerencias se inicia cuando el usuario ingresa la combinación de teclas Ctrl+Espacio.

La lista de opciones que se ofrecen como alternativas se completa teniendo en cuenta el contexto de la línea que se está escribiendo. A continuación se describen las reglas que componen la heurística utilizada para el auto-completado:

- Si la línea se encuentra en blanco se sugiere alguna de las tres posibles etiquetas: “Term:”, “FactType:” y “Rule:”. Como no es posible formular un hecho si no fue definido antes un término, o una regla sin un hecho previo, el editor sólo sugiere “FactType:” y “Rule:” si y sólo si ya fue definido con antelación un término o un hecho.

- Si comienza con la etiqueta “Term:” no se sugiere ninguna opción, el usuario debe ingresar un nuevo término.

- Si comienza con la etiqueta “FactType:”
  - y no hay nada luego de la etiqueta, sugiere un término
  - y la última palabra es un verbo, sugiere un término.

- Si comienza con la etiqueta “Rule:”:
  - y no hay nada luego de la etiqueta, sugiere una “modalidad”.
  - y la última frase es una modalidad, sugiere una “cuantificación”.
  - y la última frase es una cuantificación, sugiere un “FactType”.

A continuación se presenta el diseño del parser consultado por el editor y los criterios que lo soportan.

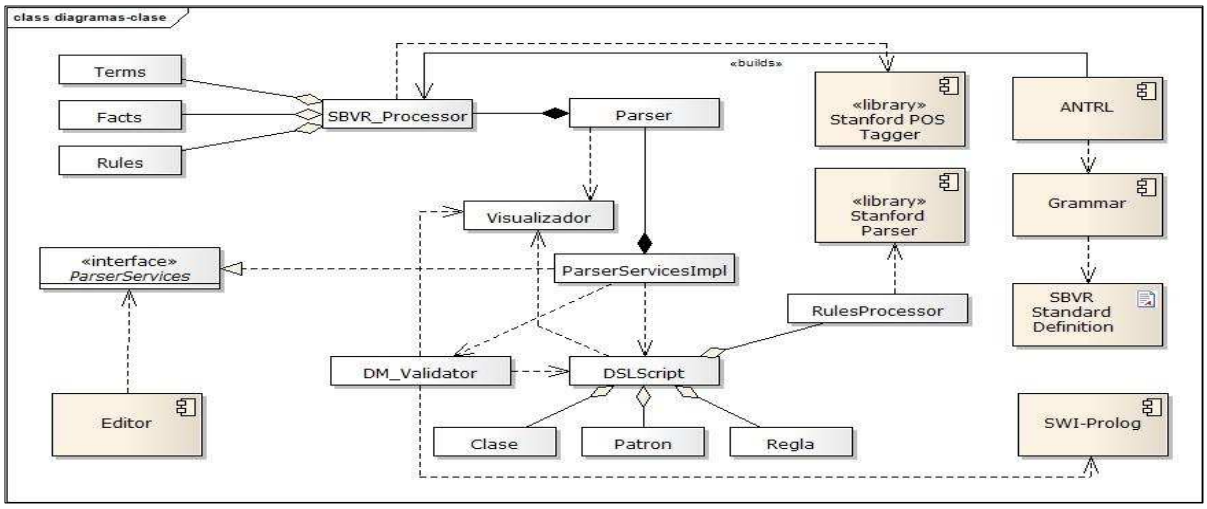


Figura 8 – Diagrama UML donde se observan los diferentes componentes que forman el Parser de SBVR, el constructor del Modelo de Dominio (DSLScript) y el generador de datos para el Visualizador.

4. Parser de SBVR

Se implementó un parser que memoriza la secuencia de frases de entrada ingresada por el analista. Los nombres, hechos y reglas detectadas como válidas en cada línea son incorporados a diccionarios que son utilizados en el procesamiento de las subsecuentes líneas de entrada.
Este componente de la plataforma es un traslador basado en reglas [2]. Esta arquitectura fue seleccionada frente al basado en la sintaxis ya que su complejidad es alta para que el parser resuelva la entrada a medida que accede a ella; estas reglas determinan qué información del contexto es utilizada y la forma de hacerlo. Debido a que la validación realizada depende de entradas anteriores este parser fue implementado utilizando predicados para la implementación de las reglas que se alimentan a su vez de la información previa memorizada en los diccionarios [2]. Estos diccionarios y su administración fueron implementados en lenguaje Java y constituyen la tabla de símbolos que almacena los datos para el posterior análisis semántico. Esta estrategia es necesaria porque las reglas dependen de los hechos previamente definidos que a su vez dependen de los nombres previamente registrados. Para desambiguar las secuencias de texto de entrada se utilizó la técnica de back tracking. La herramienta utilizada para la especificación de la gramática y la generación del parser es ANTLR [5].

4.1. Análisis Sintáctico

Las frases que componen los hechos fueron filtradas por una implementación en lenguaje Java del Part-Of-Speech Tagger (POS Tagger) [11] a efectos de detectar los verbos de estas sentencias, que como se verá más adelante se utilizan en la detección y asignación de métodos a las clases del Modelo de Dominio. Otra tarea que se lleva adelante en este procesamiento sintáctico es la detección de singulares y plurales de las palabras que definen los nombres y la generación de los correspondientes plurales y singulares. En la mayor parte del código Java escrito para la administración de las colecciones se utilizaron técnicas de búsqueda implementadas con expresiones regulares. En la figura 8 se muestran los componentes que conforman el Parser de SBVR, el generador del Modelo de Dominio, el Validador y el generador de datos para el Visualizador. El SBVR_Processor funciona como un Embedded Helper [4] en el parser, proveyendo la dinámica y las colecciones para la memorización de los datos detectados y validados que fueron ingresados por el usuario desde el editor. La integración de estas técnicas para la generación de estos componentes del software de la plataforma es un típico ejemplo de DSL [4]. Las reglas filtradas y validadas son mostradas por la plataforma como gráficos donde puede analizarse cada componente y sus dependencias. En la figura 9 se muestra un ejemplo de la salida producida por el sistema. Todas las colecciones memorizadas por el parser son utilizadas por el módulo que genera los modelos que se presenta en la siguiente sección.

5. Generador del Modelo de Dominio

En este módulo se realizan las siguientes tareas: detección de clases y asignación de atributos y métodos, detección de patrones de colaboración, y la clasificación de reglas en tipos y asignación a las clases que las

validan. Con el objetivo de construir un Modelo de Dominio a partir del vocabulario y las reglas de negocio se introdujeron en el procesamiento asociado al parser una serie de patrones de búsqueda que facilitaron la identificación de relaciones entre las entidades que conforman el modelo, además de propiedades, tipos y estados de cada una de ellas.

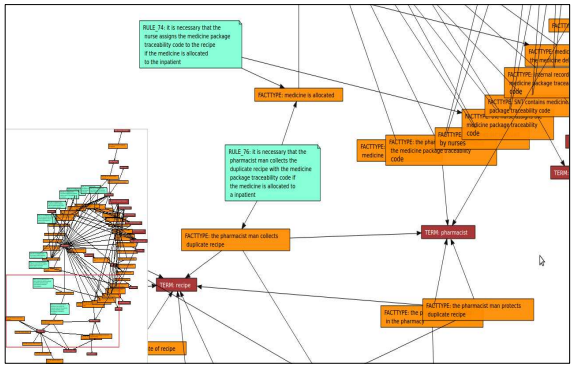


Figura 9 – Reglas, hechos y nombres relacionados de un caso de prueba mostrado por el Visualizador. Las reglas aparecen como notas en color turquesa, los nombres y hechos como rectángulos marrones y anaranjados respectivamente.

Este Modelo de Dominio se construye como un modelo de objetos por lo que a las entidades las llamamos clases en adelante. Las relaciones de interés dentro del modelo son de tipo composición, agregación y asociación con cardinalidad variada. Para facilitar su detección, la gramática definida para el parser así como las sugerencias realizadas por el editor al usuario, incluyó entre las reglas las expresiones del listado que sigue:

PROPERTY:
'property of';
STATE:
'state of';
TYPE:
'type of';
RELATION:
'designed'|
'part of'|
'includes'|
'included in'|
'contains'|
'contained in'|
'member of'|
'category of'|
'kind of' ;

Las clases que modelan los eventos y transacciones de negocio se detectaron a partir de las acciones que los roles realizan sobre otras entidades, utilizando los verbos de las diferentes frases asociadas.
La detección de los patrones de colaboración se realizó a partir del tipo de relación entre las clases; la asignación de las reglas de negocio a las clases para su validación se implementó utilizando las estrategias delineadas en [1]. En la figura 10 se observa un diagrama de clases y los patrones de colaboración correspondientes y en la figura 12 el mismo diagrama con las reglas vinculadas a las clases responsables de su validación. Todas estas gráficas fueron generadas por el Visualizador.

5.1. Detección de clases y asignación de atributos y métodos

Los nombres registrados son instanciados como clases en el Modelo de Dominio. Las propiedades indicadas con el patrón de entrada 'property of' en los hechos correspondientes son registrados como atributos de las clases; las entradas con el patrón 'state of' son registrados como estados de las clases correspondientes y las 'type of' como tipos.

5.2. Detección de patrones de colaboración

Los patrones son detectados a partir del tipo de relación entre las clases ya detectadas. A partir de la relación 'includes' se instancia un patrón 'Container-Content', de 'part of' se instancia un patrón 'Whole-Part', de 'member of' o 'category of' se instancia un patrón 'Group-Member' y de 'kind of' se instancia un patrón 'Item-Specific Item'.
Para la instanciación 'Actor-Role' se utilizan las entradas definidas como 'is designed´ acompañando a un nombre 'Role'. Los patrones que involucran transacciones se detectan a partir de las acciones realizadas por los diferentes roles sobre distintos objetos de negocio. Estos son 'Transaction-Role', 'Transaction-Place' y 'Transaction- Specific Item'. Todos estos patrones de colaboración fueron propuestos por Nicola en [1].

5.3. Clasificación de reglas en tipos y asignación a las clases que las validan

Una vez parseadas y validadas según el estándar SBVR, las reglas deben ser clasificadas por su tipo y asignadas a las clases responsables de su validación. En este proceso se deben determinar los siguientes datos:

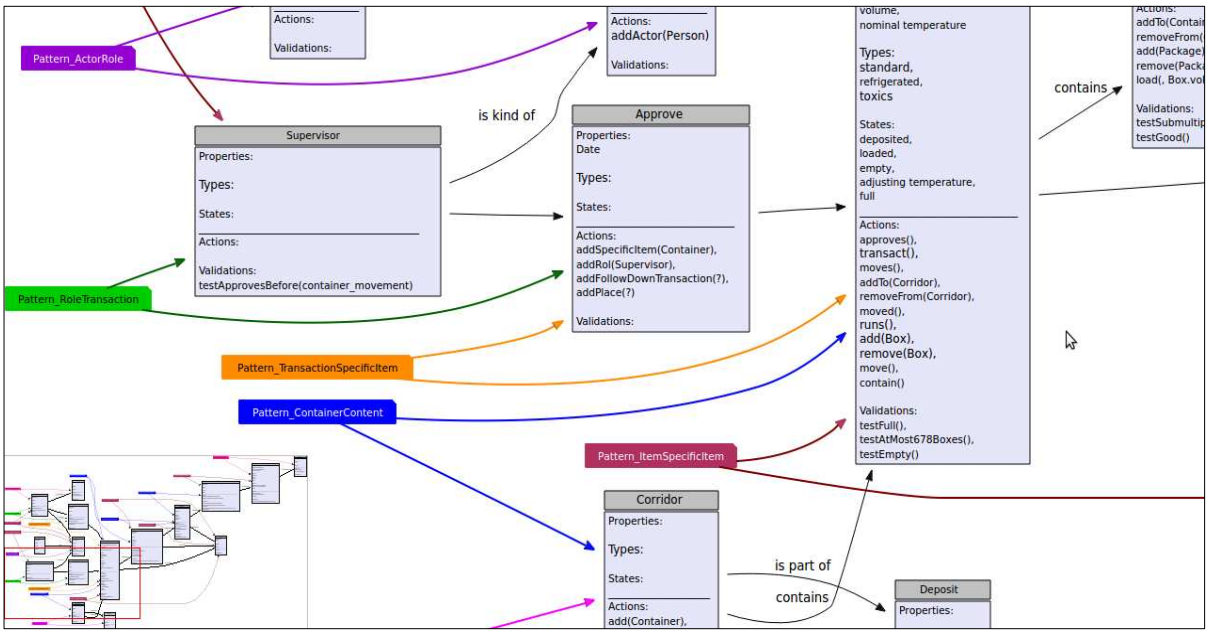


Figura 10 – Clases del modelo de dominio generado y los patrones de colaboración asociados, mostrados por el Visualizador.

Tipo de regla: donde los tipos son los definidos en [1] como Tipo, Multiplicidad, Propiedad, Estado y Conflicto.
Alcance de la regla: el alcance puede ser Auto o Colaboración, según se deba realizar la validación por una sola de las clases del par correspondiente al patrón o por ambas.
Cuándo validadarla: esto indica el lugar del código dónde será invocada la validación de la regla.
Decisión de la Validación: determina si la regla se cumple o viola y en consecuencia si se permite o prohibe realizar la acción en curso, la cual fue iniciada por un evento de negocio y cuyo desarrollo depende del cumplimiento de una o más reglas de negocio.
Condiciones de la validación: esto determina si la comparación que se realiza en la validación es por menor, igual o mayor; por sí o por no; además de cuáles variables participan de las comparaciones.
Parámetros: estos representan los valores de la información necesaria para armar la sentencia de validación.
Clases validadoras: nombres de las clases participantes en la validación.
Toda esta información se obtiene de un análisis sintáctico / semántico de la sentencia que conforma la regla de negocio. Este análisis se realiza utilizando el parser Stanford Dependencies Parser del Stanford Natural Language Processing Group [10]. Se seleccionó esta herramienta ya que está basada en la teoría de "dependencia entre palabras" la cual consideramos más apropiada para nuestro contexto. El análisis a partir de frases "constitutivas" agrega un grado de complejidad que se decidió evitar. El algoritmo utilizado en el procesamiento de las reglas es el que se describe a continuación. Los pasos del algoritmo implementado son:

1. Detección de la raíz de la sentencia (decisión de la regla)
2. Detección de la acción

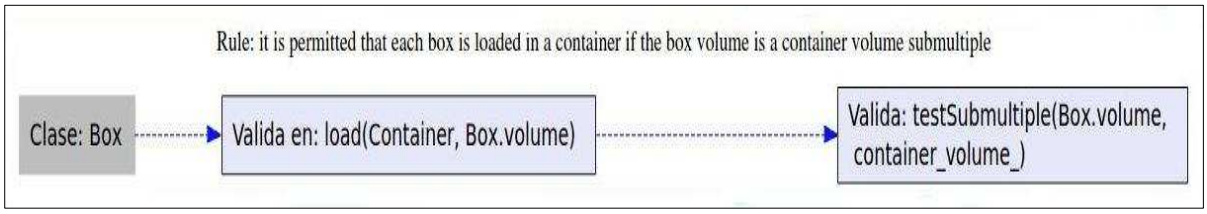


Figura 11 – Secuencia de invocación de una regla, mostrada por el Visualizador.

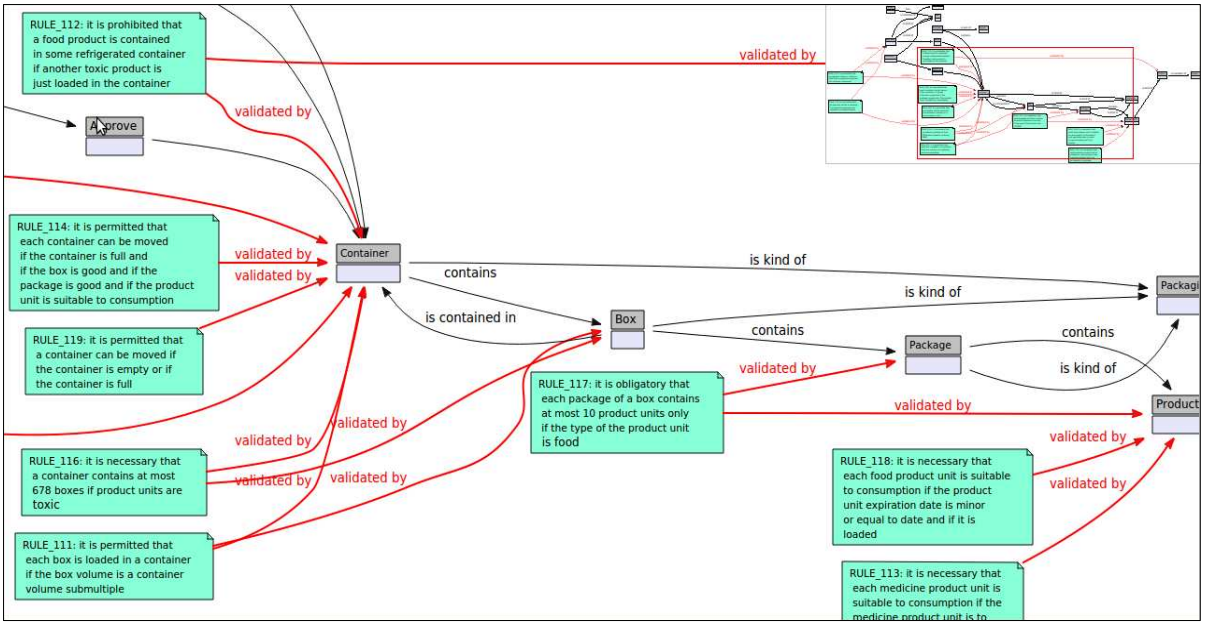


Figura 12 – Clases y las reglas asignadas, generadas y mostradas por el Visualizador.

3. Detección de las clases del patrón de colaboración
4. Detección de las condiciones de la regla
5. Detección de las variables de las condiciones
6. Asignación de las reglas a las clases que las validan

La implementación de estos pasos se basa en el procesamiento de la salida generada por el parser Stanford Dependencies Parser [10] que determina la relación entre cada par de las palabras que forman el texto de la regla. En el texto que sigue se muestra un ejemplo de esta información en el formato en que la genera el parser para una regla de negocio ejemplo y el resultado del procesamiento que consiste en métodos, para las acciones y las validaciones, asignados a las clases.

Rule: it is permitted that each box is loaded in a container if the box volume is a container volume submultiple

nsubjpass(permitted-3, it-1)
auxpass(permitted-3, is-2)
root(ROOT-0, permitted-3)
mark(loaded-8, that-4)
det(box-6, each-5)
nsubjpass(loaded-8, box-6)
auxpass(loaded-8, is-7)
ccomp(permitted-3, loaded-8)
prep(loaded-8, in-9)
det(container-11, a-10)
pobj(in-9, container-11)
mark(submultiple-20, if-12)
det(volume-15, the-13)
nn(volume-15, box-14)
nsubj(submultiple-20, volume-15)
cop(submultiple-20, is-16)
det(submultiple-20, a-17)
nn(submultiple-20, container-18)
nn(submultiple-20, volume-19)
advcl(loaded-8, submultiple-20)

La secuencia de invocación generada para esta regla se muestra en la figura 11. En la siguiente sección se describe resumidamente el visualizador que permite explotar toda la información obtenida.

6. Visualizador

El visualizador fue construido utilizando como referencia el estándar Dot de visualización de información [18]. En este diseño los gráficos compuestos como Clases + Reglas y Clases + Patrones se construyen con decoradores que operan sobre las clases generadoras de archivos con formato Dot. Este es un aspecto en el cual se continuará trabajando, se piensa incorporar formas de visualización más elaboradas para estas vistas.

7. Validador

La lógica de primer orden, que sustenta al esquema de SBVR y que está subyacente en la gramática del parser, valida los hechos y las reglas de negocio ingresadas como texto controlado utilizando el editor. Con el objetivo de dotar a la plataforma de un mecanismo de validación, para el Modelo de Dominio generado, se decidió expresar en forma declarativa los invariantes del modelo a través de definiciones y reglas lógicas en lenguaje Prolog. A partir de consultas ejecutadas

orientadas a validar el modelo se obtienen resultados válidos e inválidos cuando se las aplica a una base de datos cuyas instancias provienen del modelo generado. La ubicación de este componente ya se mostró en la figura 8; la interfaz entre Prolog y Java [23] implementada permite realizar la validación como un paso más del procesamiento de los datos en forma previa a la visualización. Uno de los invariantes expresados es que una transacción debe ser realizada por un rol, asignado a un actor, sobre un determinado ítem y en un determinado lugar. Cuando el texto ingresado no provee la información que cumpla con este invariante, una regla lógica de los invariantes falla indicando esto, lo cual es mostrado en la vista correspondiente. La falla puede expresar por ejemplo que no se definió el actor para el rol o que no se indicó el lugar donde se realiza la transacción. Estas advertencias guían al analista que puede completar el texto ingresado o desestimarla según sea el problema que está abordando.

La salida generada por este "compilador" del Modelo de Dominio es también mostrada por el Visualizador como las otras salidas ya descriptas. En la figura 13 se presenta la vista provista por el Visualizador para un caso en el cual se introdujeron un par de inconsistencias del estilo mencionadas anteriormente.

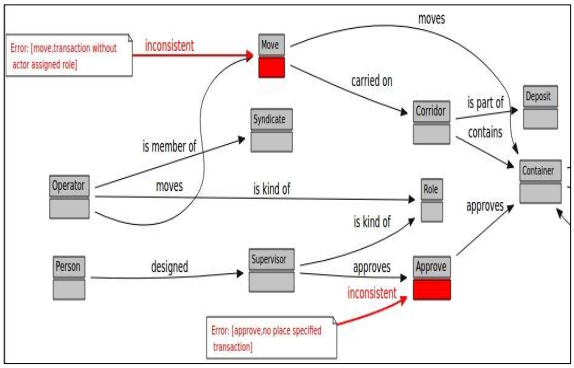


Figura 13 – Vista provista por el Visualizador para los errores detectados por el Validador.

En el párrafo que sigue se muestra parte del código Prolog. Se definieron los patrones de colaboración

```
actorRole.
roleTransaction.
transactionPlace.
transactionSpecificItem.
wholePart.
containerContent.
groupMember.
itemSpecificItem.
```

```
patternCommand(executes).
patternReference(role).
patternPar(transaction).
colaborationPattern(roleTransaction) :- executes, role, transaction.

patternCommand(transactionIn).
patternReference(transaction).
patternPar(place).
colaborationPattern(transactionPlace) :- transactionIn, transaction, place.

patternCommand(transactionOn).
patternReference(transaction).
patternPar(specificItem).
colaborationPattern(transactionSpecificItem) :- transactionOn, transaction, specificItem.
. . .

roleTransaction(R, T):- role(R), transaction(T), executes(R, T).
transactionPlace(T, L):- transaction(T), place(L), transactionIn(T, L).
transactionSpecificItem(T, IE):- transaction(T), specificItem(IE), transactionOn(T, IE).
. . .
```

se validó que la transacción fue iniciada por un rol asignado a un actor para la ejecución y que se desarrolla en un determinado lugar

```
validateTransaction(T, IE):-
(transactionSpecificItem(T, IE);
reportError([T,
'transaction not defined for the specific item'])),
((roleTransaction(R, T), actorRole(_, R));
reportError([T, 'transaction without actor assigned role'])),
(transactionPlace(T, _);
reportError([T, 'no place specified transaction'])).
```

Se definieron los tipos de reglas y la estrategia de asignación a las clases

```
ruleType(type).
ruleType(multiplicity).
ruleType(property).
ruleType(state).
ruleType(conflict).

ruleAssign(transactionPlace, [T, P], type, X):- place(P), transaction(T),
transactionIn(T, P), X = [P], !.
```

```
ruleAssign(transactionPlace, [T, P], multiplicity, X):- place(P), transaction(T), transactionIn(T, P), append([T], [P], X), !.

ruleAssign(transactionPlace, [T, P], property, X):- place(P), transaction(T), transactionIn(T, P), X = [P], !.
ruleAssign(transactionPlace, [T, P], state, X):- place(P), transaction(T), transactionIn(T, P), X = [P], !.
ruleAssign(transactionPlace, [T, P], conflict, X):- place(P), transaction(T), transactionIn(T, P), X = [P], !.
. . .
```

Se asignan las reglas a las clases según la estrategia definida

```
assignRulesToPatternClasses(Pattern, ClassesDetect, RuleType, ClassesVal, ID) :-
ruleToAssign(Pattern, ClassesDetect, RuleType, ID),
ruleType(RuleType), ruleAssign(Pattern, ClassesDetect, RuleType, ClassesVal),
 write('Assign: '), write([ID, Pattern, RuleType, ClassesVal]), nl.
```

8. Conclusión

Se presentó un proceso de documentación y análisis del vocabulario y reglas de negocio del software de un sistema en desarrollo. Este proceso se implementó sobre una herramienta que facilita el trabajo del analista durante el relevamiento y el análisis del dominio con el resto del grupo de desarrollo. Fue diseñado teniendo en cuenta el estándar SBVR y produce múltiples salidas gráficas orientadas a la visualización como mecanismo de comprensión rápida. Al momento de escribir este artículo se trabaja en la integración con una implementación de los objetos del Modelo de Dominio publicada en [3], la cual incluirá la generación automática del código del dominio y de los test de prueba de la validación de las reglas. Si bien la estrategia planteada como solución es un ejemplo de Programación Generativa [7], la extensión mencionada de la plataforma propuesta fue pensada como una herramienta de ayuda para facilitar la comprensión de los dominios en proyectos de desarrollo de software y no está orientada a la generación de código con el objetivo de acelerar tiempos de desarrollo. También se trabaja en la generación automática de pruebas de la validación de las reglas que incorporadas en un lenguaje específico de pruebas servirá como herramienta de aprendizaje interactivo del dominio de sistemas en desarrollo. El foco de la reutilización está puesto en los componentes que proveen la facilitación del análisis y comprensión de un problema. Debido al desarrollo avanzado del estudio de estadísticas de habla inglesa es que se trabajó en este idioma, sería deseable trasladar este desarrollo al idioma español para lo cual se debería contar con versiones de los estándares de SBVR, POSTagger y Parser de Lenguaje Natural.

Agradecimientos
Guillermo Pantaleo agradece a Tekii srl por el soporte parcial de este trabajo en sus inicios.

Referencias

[1] Jill Nicola y otros, S*treamlinned object modeling, patterns, rules and implementations*, Prentice Hall PTR, 2002.
[2] Terence Parr, *Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages*, Pragmatic Bookshelf, 2010.
[3] Guillermo Pantaleo, *Segregating and integrating views of Domain Objects using DSL and C ++ policy*, enviado para su publicación a Systems and Software, Elsevier, 2013.
[4] Martin Fowler, *Domain Specific Languages*, Adison Wesley, 2011.
[5] ANTLR, *Another Tool for Language Recognition*, http://www.antlr.org
[6] SBVR - *Semantics Of Business Vocabulary And Business Rules*, http://www.omg.org/spec/SBVR/1.0/
[7] Krysztof Czarnecki and Ulrich Eisenecker , *Generative Programming: Methods, Tools, and Applications*, 2000.
[8] Amit Raj, T. V. Prabhakar, Stan Hendryx , *Transformation of SBVR Business Design to UML Models*, ISEC'08, February 19-22, 2008, Hyderabad, India.
[9] Part-Of-Speech Tagger (POS Tagger), [http://nlp.stanford.edu/downloads/tagger.shtml]
[10] Stanford Natural Language Processing Group, http://nlp.stanford.edu/software/stanford-dependencies.shtml
[11] Alexandros Marinos, Pagan Gazzard, Paul Krause, *An SBVR Editor with Highlighting and Auto-completion*, CEUR Workshop Proceedings, volume 799, Issue 13, 2011.
[12] Imran Sarwar Bajwa, M. Abbas Choudhary, *From Natural Language Software Specifications to UML Class Models*, Lecture Notes in Business Information Processing (NLBIP-102), pages 224-237, Springer-Verlag Berlin Heidelberg, 2012.

[13] G.S. Anandha Mala and G.V. Uma, *Automatic Construction of Object Oriented Design Models [UML Diagrams] from Natural Language Requirements Specification*, Proceeding PRICAI'06 Proceedings of the 9th Pacific Rim international conference on Artificial intelligence, pages 1155-1159, Springer-Verlag Berlin, Heidelberg, 2006.

[14] H.M. Harmain, R.Gaizauskas, *CM Builder: A automated NL based CASE tool*, Proceeding ASE '00 Proceedings of the 15th IEEE international conference on Automated software engineering, page 45, IEEE Computer Society Washington, DC, USA, 2000

[15] Imran Sarwar Bajwa, Behzad Bordbar, Mark G. Lee , *OCL Constraints Generation from Natural Language Specification*, 2010 14th IEEE International Enterprise Distributed Object Computing Conference

[16] Hina Afreena, Imran Sarwar Bajwab, *Generating UML Class Models from SBVR Software Requirements Specifications , 23rd Benelux Conference on Artificial Intelligence BNAIC 2011, pages 23-32, Gent, Belgium, 2011*.

[17] Eric Evans*, Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison Wesley, 2006.

[18] Graphviz, *Gráficos en formato Dot,* http://www.graphviz.org/

[19] O'Brien, Sharon*, Controlling Controlled English - An Analysis of Several Controlled Language Rule Sets,* Proceedings of EAMT-CLAW, 2003.

[20] Ace - *The High Performance Core Editor For The Web*, [http://ace.c9.io/#nav=about]

[21] Ace - *Creating a syntax highlighter*, [http://ace.c9.io/#nav=higlighter]

[22] Mozilla Developer Network, *Using Web Workers,* [https://developer.mozilla.org/en-US/docs/Web/Guide/Performance/Using_web_workers]

[23] SWI-Prolog, *Plataforma Prolog*, http://www.swi-prolog.org/

Sintonización Estática de Parámetros Operacionales para un Modelo Paralelo de Evolución Diferencial

María Laura Tardivo (*,+,†), Paola Caymes Scutari (+,†),
Miguel Méndez Garabetti (+,†), Germán Bianchini (+)

(*) Departamento de Computación,
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Argentina, X5804BYA
lauratardivo@dc.exa.unrc.edu.ar

(+) Laboratorio de Investigación en Cómputo Paralelo/Distribuido (LICPaD)
Departamento de Ingeniería en Sistemas de Información, Facultad Regional Mendoza
Universidad Tecnológica Nacional
Mendoza, Argentina, M5502AJE
{pcaymesscutari,gbianchini}@frm.utn.edu.ar
miguelmendezgarabetti@gmail.com

(†) Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Resumen

La calidad de las aproximaciones obtenidas con técnicas metaheurísticas está estrechamente relacionada con los valores de los parámetros de configuración del algoritmo respecto de la función a optimizar. Para calibrar estos parámetros usualmente se hace necesario llevar a cabo múltiples pruebas con diferentes combinaciones de valores, hasta alcanzar el resultado deseado. Sin embargo, la cantidad de ejecuciones que se deben realizar se traduce a tiempos de cómputo elevados. En este tipo de situaciones, el empleo de alternativas paralelas de estos métodos puede ser de gran utilidad. En este trabajo se presenta un modelo paralelo basado en islas que ha sido aplicado a la metaheurística Evolución Diferencial. El objetivo de este trabajo es lograr calibrar los parámetros del algoritmo utilizando diferentes tipos de calibración estática, intentando obtener patrones de comportamiento que ayuden a identificar los valores para los cuales se obtiene mejor desempeño del algoritmo.

1. Introducción

En el campo de las ciencias computacionales se han desarrollado diferentes técnicas que intentan dar solución a ciertos problemas que no pueden ser resueltos de manera exacta en un tiempo razonable. Estas técnicas, denominadas metaheurísticas, han ganado popularidad durante los últimos años y han sido ampliamente utilizadas tanto en la ciencia como en la industria, especialmente aplicado a problemas de optimización [12,23,27]. Un problema de optimización es definido en [12] como un par (S, f), donde S representa el conjunto de posibles soluciones y $f: S \rightarrow \mathbb{R}$ es la función objetivo a optimizar, la cual asigna a cada solución $s \in S$ del espacio de búsqueda un número real indicando su desempeño. Por ende, el principal objetivo en resolver un problema de optimización es encontrar una solución $s^* \in S$, denominado *óptimo global*, que posee el mejor valor en la evaluación de la función objetivo respecto de todas las posibles soluciones del espacio de búsqueda. Sin embargo, encontrar el óptimo global puede requerir evaluar el espacio de búsqueda completo, lo cual se traduce a tiempos incalculables para algunos problemas.

Es por esto que obtener soluciones aproximadas al óptimo global puede resultar una alternativa. Las metaheurísticas intentan resolver instancias de estos problemas, explorando el amplio espacio de soluciones que esas instancias pueden admitir.

Existen varias clasificaciones para las metaheurísticas, una de las más utilizadas las suele dividir en dos categorías de acuerdo a la cantidad de soluciones que emplean en el proceso de búsqueda: metaheurísticas de trayectoria, o de una sola solución y metaheurísticas poblacionales. Este trabajo se centra en las metaheurísticas basadas en poblaciones, es decir, aquellas que operan con un conjunto de soluciones, denominado población, representado por individuos que interactúan entre sí para llevar a cabo el proceso de búsqueda.

Por su naturaleza, las metaheurísticas basadas en población son propensas a la paralelización, dado que las operaciones que realizan pueden ser llevadas a cabo en paralelo [12]. En la literatura se han propuesto diferentes alternativas paralelas, dependiendo del tipo de recurso disponible y del objetivo a lograr [10,12,19]. Uno de los modelos más difundidos es el modelo de isla, que generalmente se funda en un esquema "master-worker" [7], en donde un proceso master inicializa la población y delega a los procesos workers la evaluación de la función objetivo de cada individuo. Este tipo de esquema no afecta el comportamiento original de la metaheurística. Existen otras aproximaciones en donde la población inicial es distribuida a los procesos workers, para que estos realicen el proceso evolutivo de su porción de la población. Si bien las subpoblaciones evolucionan de manera separada, existe una interacción entre ellas en la cual cooperan intercambiando información de manera tal de lograr entre todos el objetivo global de optimización. Este tipo de esquema rompe con la estructura inicial de la metaheurística. Sin embargo, se ha demostrado que el uso de estas alternativas paralelas puede incrementar la calidad de las soluciones encontradas respecto de la versión secuencial [18], incluso utilizando un solo procesador [1]. En este trabajo nos centramos en modelos paralelos basados en el concepto de isla y migración entre las poblaciones, bajo entornos de ejecución con memoria distribuida.

Cada método, sea secuencial o paralelo, involucra una serie de parámetros que son utilizados en la configuración inicial de la población, de los parámetros propios del modelo paralelo o durante el proceso de búsqueda en sí, de manera tal de adaptar el problema a un determinado contexto. Una selección adecuada de dichos parámetros puede ser fundamental para obtención de resultados aceptables. Sin embargo, esta tarea no es trivial. Generalmente se comienza realizando diferentes pruebas, variando los valores de uno o más parámetros hasta encontrar el valor más conveniente para cada uno. Este tipo de calibración suele denominarse como "off-line", calibración estática [12] o sintonización estática, y consiste en fijar los valores de los parámetros previos a la ejecución de la metaheurística. Existe otro tipo de calibración, denominada "online", calibración dinámica o sintonización dinámica, en la cual los parámetros son controlados y actualizados dinámicamente o de manera adaptativa durante la ejecución de la metaheurística. En este trabajo presentamos la sintonización estática de un modelo paralelo basado en islas, que utiliza como metaheurística al algoritmo Evolución Diferencial (Differential Evolution o DE). Dicho modelo, denominado CIBM-DE, ha demostrado obtener buenos resultados en cuanto a la calidad de las soluciones encontradas [15,16]. Nuestro interés radica en calibrar tres parámetros claves de DE, estos son la probabilidad de cruzamiento (*Cr*), el factor de mutación (*F*), y el tipo de cruzamiento *(Ct)*. La calibración consiste en configurar las diferentes islas del modelo paralelo con ciertos valores de dichos parámetros, de manera tal de poder obtener patrones de comportamiento del modelo bajo los escenarios planteados.

Este trabajo se organiza de la siguiente manera. La sección 2 describe brevemente la metaheurística Evolución Diferencial, incluyendo el esquema de procesamiento y algunos modelos paralelos del estado del arte. La sección 3 presenta el modelo CIBM-DE, detallando sus características principales, las ventajas y desventajas de su uso y los parámetros que son de interés en este trabajo. La sección 4 muestra los casos de test que fueron definidos para realizar la experimentación, describe la configuración del modelo y del entorno de ejecución, y enuncia los resultados obtenidos. Finalmente, la sección 4 resume las conclusiones obtenidas y los trabajos a futuro.

2. La metaheurística Evolución Diferencial

En la actualidad existen numerosas metaheurísticas poblacionales que han sido ampliamente utilizadas como métodos de optimización. Entre ellas, el algoritmo Evolución Diferencial [10] ha estimulado el interés de científicos computacionales y usuarios de otras ramas de la ciencia e industria [19].

De acuerdo a la definición original dada por los autores en [10], DE comienza generando una población inicial de individuos con valores aleatorios dentro de los rangos del problema en cuestión. Cada individuo está representado por un vector de números reales y codifica una posible solución del problema.

En cada generación, se recorre la población y por cada individuo X_i se seleccionan aleatoriamente otros individuos que serán utilizados por un operador de mutación para generar un nuevo individuo $\boldsymbol{V_i}$,

denominado “vector mutante”. El vector mutante se genera utilizando la fórmula (1):

$$V_i = X_t + F * (X_r + X_s) \quad (1)$$

en donde $F \in (0,1+)$ es un número real positivo definido por el usuario. Si bien no hay límite sobre el valor de *F*, generalmente se eligen valores menores que 1.0. Los vectores X_t, X_r y X_s son individuos distintos entre sí y entre el individuo X_i corriente. La fórmula (1) es denominada DE/rand/1. Sin embargo, se han propuesto otras variantes de la regla de mutación. Una de las más populares, denominada DE/best/1, reemplaza en la fórmula (1) al individuo X_t por el mejor individuo de la población, es decir, aquel que tiene el mejor valor respecto de la función objetivo.

Luego del proceso de mutación, DE integra el individuo corriente X_i y el individuo mutante V_i en un operador de recombinación o cruzamiento, para formar el “vector trial” $\boldsymbol{U_i}$ utilizando el esquema de la fórmula (2):

$$u_{i,g+1}^{j} = \begin{cases} v_{i,g+1}^{j} & si\ rand_j \le Cr\ o\ j=k \\ x_{i,g}^{j} & otro\ caso \end{cases} \quad (2)$$

donde $j \in [1,D]$, $rand_j$ es un número aleatorio entre 0 y 1, $Cr \in (0,1)$ es la probabilidad de cruzamiento y constituye un parámetro importante del algoritmo, puesto que determina las porciones del individuo mutante y del individuo original que formarán parte del nuevo individuo generado en esta etapa, *k* es un número aleatorio y es utilizado para garantizar que el vector trial no sea exactamente igual al individuo corriente.

Existen dos tipos de cruzamiento, ambos utilizan la expresión de la figura 2, pero difieren en el modo en el que se aplica. El cruzamiento *binomial* itera sobre todas las componentes del individuo, copiando el j-esimo elemento del vector mutante V_i al correspondiente elemento del vector trial U_i si y solo si $rand_j <= Cr$ o j=k. En caso contrario, es copiado del vector corriente X_i. En cambio, el cruzamiento *exponencial* hereda las componentes del vector mutante V_i hacia el vector trial U_i comenzando por una posición aleatoria, y hasta que se cumpla la condición $rand_j <= Cr$. El resto de los valores del vector trial son heredados directamente del vector corriente X_i.

Finalmente, DE selecciona el mejor candidato entre el individuo corriente X_i y el vector trial U_i. El mejor candidato será el que sobreviva a la siguiente generación, es decir, U_i reemplaza a X_i si y solo si $f(U_i)<f(X_i)$ (para problemas de minimización), caso contrario, no ocurre reemplazo o selección.

Por su simplicidad, fácil codificación y robustez, DE ha sido utilizado para resolver múltiples problemas de optimización, por ejemplo, en el ámbito de la ingeniería electrónica [3,20], procesamiento de imágenes [11], ingeniería civil [28], entre otras. También se ha investigado la utilización de Evolución Diferencial en conjunto con otros métodos, formando algoritmos híbridos. Entre ellos podemos mencionar aquellos que combinan DE con métodos de trayectoria ([2,9,14]).

2.1. Modelos paralelos para DE

Si bien DE tiene un gran potencial, su desempeño puede ser mejorado agregando modificaciones a la estructura de procesamiento original del algoritmo con el fin de obtener mejoras significativas. Una forma de lograr este objetivo es integrar diferentes poblaciones o subpoblaciones de manera tal que evolucionen en paralelo. Bajo esta perspectiva, y con el desarrollo de modelos paralelos híbridos, surge el concepto de “isla” para representar una población o subpoblación de individuos que son evolucionados por una o más unidades de procesamiento [12]. Periodicamente ciertos individuos migran de una isla a otra y reemplazan a otros individuos en la isla destino. Este proceso suele ser denominado “etapa de migración”. Finalmente se selecciona de todas las islas el individuo de mejor desempeño.

En la literatura se han propuesto diversos modelos distribuidos paralelos, muchos de ellos utilizan el concepto de “isla” y se encuadran en modelos de memoria distribuida. En [17] se propone un esquema de procesamiento paralelo para DE, en donde la población inicial se subdivide en subpoblaciones y se asigna cada subpoblación a un nodo de procesamiento diferente. La topología de interconexión entre los nodos es de estrella: el proceso master coordina las comunicaciones entre los workers con una tasa de migración aleatoria que es controlada por un parámetro del algoritmo. El mejor individuo es enviado desde cada worker al master y este determina si se debe enviar o no dicho individuo hacia el proceso worker siguiente en la topología respecto del worker emisor. En [4] se describe una versión basada en el algoritmo propuesto en [17], en donde la población inicial es subdividida en subpoblaciones y es asignada a una unidad de procesamiento. Cada unidad realiza el proceso evolutivo hasta la etapa de migración, en donde se envía un grupo de individuos hacia un procesador destino siguiendo una topología de anillo.

En [24] se propone una versión paralela aplicada a la solución del problema de la frontera de Pareto. En este algoritmo, se utiliza un esquema de isla, y los individuos de las subpoblaciones migran con cierta probabilidad a una posición aleatoria de una población aleatoriamente seleccionada. Con una lógica similar, en [6] se propone una versión distribuida para DE, en la cual el proceso de migración selecciona el mejor individuo de la población

fuente para reemplazar al individuo más viejo de la población destino. En los ensayos, los autores realizaron pruebas con diferentes tasas de migración, e identificaron la mejor de ellas.

En [18] se presenta una versión distribuida para DE en donde la población inicial es subdividida en subpoblaciones, las cuales están agrupadas en dos categorías: de exploración y de explotación. Las primeras tienen tamaño constante y están organizadas bajo una topología de anillo, en donde se migra el individuo de mejor desempeño. Las segundas tienen tamaño dinámico y potencian la intensificación, no incluyen el operador de migración entre ellas y su objetivo es detectar de manera rápida soluciones de calidad. En cierta etapa del algoritmo, las subpoblaciones de explotación envían el mejor individuo encontrado a una población explorativa seleccionada de manera aleatoria. Este individuo es integrado en la población destino, posibilitando la exploración directa hacia mejores soluciones.

En los modelos mencionados anteriormente la población inicial es subdividida en subpoblaciones de manera tal de asignar cada una de ellas a una unidad de procesamiento, logrando una reducción del problema inicial y, por ende, del tiempo de ejecución respecto de la versión secuencial. El modelo utilizado en este trabajo tiene como objetivo abarcar un espacio de búsqueda mayor, con el objetivo de encontrar mejores soluciones. Para ello, cada worker posee una población completa, inicializada con una semilla diferente y realiza el proceso evolutivo sobre su población. Periódicamente los workers se comunican entre sí para intercambiar individuos de sus poblaciones. A continuación se describe en más detalle dicho modelo, denominado CIBM-DE.

3. CIBM-DE

En esta sección presentamos CIBM-DE (Clasical Island Based Model for Differential Evolution), un modelo paralelo basado en islas para la metaheurística Evolución Diferencial [16]. El modelo se implementa bajo un esquema master-worker, donde los workers se comunican a través de una topología de anillo. Una isla del modelo está integrada por una población de individuos y un proceso worker, quien realiza las diferentes etapas evolutivas sobre la población de su isla. En CIBM-DE el proceso master adopta un rol de control, su participación central será al final del proceso, para determinar la solución global.

Cada población de las islas es inicializada con una semilla diferente. Esto significa que cada isla administra un espacio de búsqueda diferente del problema bajo estudio, lo que puede favorecer la exploración de nuevas regiones no visitadas aún. Periódicamente, los workers se comunican entre sí para intercambiar un grupo de individuos de sus poblaciones. Los individuos a enviar están integrados por el mejor individuo de la población (es decir, aquel cuya evaluación de la función objetivo posee el mejor valor) y el resto está constituido por individuos seleccionados al azar. Cuando un worker recibe un grupo de individuos para integrar a la población, aplica una política de reemplazo semi-elitista, es decir, se reemplazan los peores individuos de la población siempre y cuando estos posean peor desempeño que los individuos que arriban.

La figura 1 ilustra el modelo CIMB-DE. Cada entidad está ubicada en un nodo de cómputo diferente. Los workers evolucionan la población hasta alcanzar un cierto número de evaluaciones de la función objetivo o hasta llegar a cierta cantidad prefijada de generaciones. Finalmente, cada worker selecciona el mejor individuo de la población, y lo envía al proceso master, quien clasifica el mejor de todos los encontrados en cada isla.

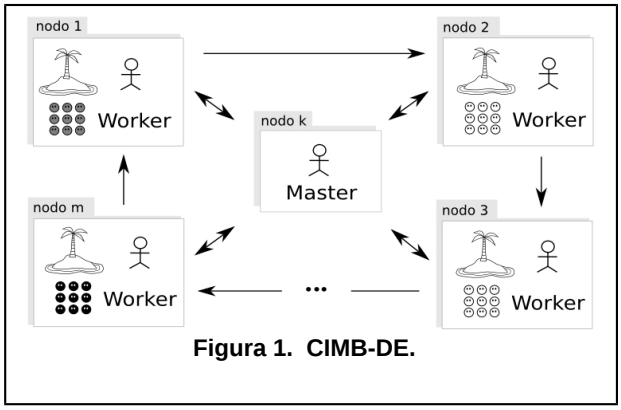


Figura 1. CIMB-DE.

Este modelo ha demostrado obtener un buen desempeño en cuanto a la calidad de las soluciones obtenidas respecto de la versión secuencial. Sin embargo, el tiempo de ejecución crece en función del aumento de la cantidad de islas. Esto se debe a que el modelo equivale a correr en paralelo múltiples instancias de DE, es decir, no hay una subdivisión de la población inicial. Además, se suma el tiempo necesario para el esquema de comunicación entre las diferentes islas en el proceso de migración y una comunicación final para la determinación del mejor individuo global. De cualquier manera, el tiempo de cómputo adicional puede considerarse despreciable en relación al tiempo total de ejecución del proceso evolutivo en sí [15]. En este trabajo hemos seleccionado CIBM-DE como modelo paralelo principalmente por su buen desempeño respecto de la calidad de las soluciones encontradas.

3.1. Los parámetros de interés

En los últimos años se han propuesto variaciones y

métodos que intentan mejorar el rendimiento de DE utilizando técnicas de sintonización. Entre ellos podemos mencionar aquellos que intentan adaptar los parámetros de manera auto ajustable [21,25,26]; otros utilizan diferentes mecanismos para optimizar la selección de los individuos que se utilizan en la etapa de mutación y cruzamiento [5], y algunos combinan estos dos últimos [22].

El interés de este trabajo radica en calibrar el modelo CIBM-DE en tres niveles de sintonización. Por un lado, se realiza una *calibración a nivel de parámetros individuales*, este tipo de sintonización se efectúa de manera inicial y consiste en determinar el valor de cada parámetro para el cual el algorimto posee mejor desempeño respecto de las funciones en estudio. Por otro lado, se realiza una *calibración combinada de parámetros*. Este tipo de sintonización consiste en establecer ciertos valores para los parámetros, que pueden o no ser los obtenidos de la calibración individual, y combinarlos de manera tal de poder analizar su interrelación bajo el modelo. Finalmente, se realiza una *calibración respecto del volumen de entidades del modelo* que adoptan cierta configuración, estableciendo una serie de valores particulares para las islas que lo integran o para un grupo de ellas.

Los parámetros que en este trabajo hemos seleccionado son utilizados en el proceso de búsqueda y son determinantes de la calidad de las soluciones encontradas. Ellos son la probabilidad de cruzamiento (*Cr)*, el factor de mutación (*F*) y el tipo de cruzamiento (*Ct).* Los parámetros *Cr* y *Ct* se utilizan en la etapa de cruzamiento para determinar las porciones del individuo que son copiadas desde el vector mutante hacia el vector trial. Además, el parámetro *Ct* tiene gran influencia sobre el tiempo de ejecución del algoritmo. Si el tipo de cruzamiento seleccionado es *binomial*, se recorren todas las componentes de cada individuo de la población en la etapa de cruzamiento, mientras que si se utiliza la configuración *exponencial* se recorre una cierta porción de cada individuo (se comienza por una posición aleatoria y se recorre el vector mientras no se cumpla que $rand_j < Cr$). En el peor de los casos, puede suceder que utilizando cruzamiento exponencial se recorran todas las componentes del individuo, como en el tipo de cruzamiento binomial, esto sucede cuando se recorre todo el individuo y nunca se cumple la condición $rand_j < Cr$. Sin embargo, existe mayor probabilidad que el tipo de cruzamiento exponencial produzca un tiempo promedio de procesamiento menor que el binomial.

El factor de mutación *F* se utiliza como factor de escala y controla la amplitud de la diferencia de vectores que se calcula en el proceso de mutación. Es utilizado para evitar el estancamiento del algoritmo, es decir, lograr que el algoritmo siempre converja hacia una solución (aunque no sea la óptima).

Inicialmente se realizó la ejecución del modelo CIBM-DE con valores obtenidos de la literatura: *Cr*=0.3, *F*=0.5 y *Ct*=Binomial. Para poder establecer valores de referencia que mejoren la calidad de la aproximación se realizó la *calibración de parámetros individuales* partiendo de dichos valores de la literatura. Esta etapa previa de calibración consistió en fijar el valor de dos de los parámetros y e ir modificando el tercer parámetro en cada ejecución, realizando pequeñas variaciones sobre su valor (tanto en valores superiores o inferiores al valor inicial) hasta obtener soluciones de mejor calidad. Este estudio determinó que utilizar *Cr* con valor 0.45 obtiene mejores resultados en cuanto a calidad de error, mientras que el valor *F*=0.5 y cruzamiento binomial son los que obtuvieron mejor desempeño. A partir de estos datos se establecieron los casos de test para los otros dos tipos de calibración, definiendo una serie de combinaciones para estos tres parámetros, fijando una cantidad de islas del modelo con ciertos valores y variando con determinado criterio los valores en el resto de las islas.

4. Casos de test y resultados obtenidos

A continuación describiremos los casos de test, cada uno establece un criterio de selección para los valores de los tres parámetros y corresponden a la *calibración combinada* y la *calibración del volumen de entidades del modelo*:

Caso 1: Todas las islas son configuradas con los mismos valores para *Cr y F (Cr=0.3* y *F=0.5).* El tipo de cruzamiento utilizado es *binomial.*

Caso 2: Todas las islas son configuradas con los mismos valores para *Cr y F (Cr=0.45* y *F=0.5).* El tipo de cruzamiento utilizado es *binomial.*

Caso 3: Todas las islas usan el valor de referencia para F, es decir F=0.5, pero utilizan valores aleatorios dentro de un rango cercano al valor de referencia 0.3 para *Cr* en el orden de más menos dos puntos, es decir $Cr \in [0.1, 0.5]$. El tipo de cruzamiento utilizado es *binomial.*

Caso 4: Todas las islas usan el valor de referencia para F, es decir F=0.5, pero utilizan valores aleatorios dentro de un rango cercano al valor de referencia 0.45 para *Cr* en el orden de más menos dos puntos, es decir $Cr \in [0.25, 0.65]$. El tipo de cruzamiento utilizado es *binomial.*

Caso 5: Todas las islas usan valores aleatorios para *Cr* y *F*, dentro de un rango acotado, cercano al valor de referencia en el orden de más menos dos puntos. El valor de referencia para Cr es 0.3 y para F es 0.5. Es decir $Cr \in [0.1, 0.5]$ y $F \in [0.2, 0.7]$. El tipo de cruzamiento es *binomial.*

Caso 6: Todas las islas usan valores aleatorios para *Cr* y *F*, dentro de un rango acotado, cercano al valor de

referencia en el orden de más menos dos puntos. El valor de referencia para Cr es 0.45 y para F es 0.5. Es decir $Cr \in [0.25, 0.65]$ y $F \in [0.2, 0.7]$. El tipo de cruzamiento es *binomial.*

Caso 7: Todas las islas usan los valores de referencia *Cr=0.3* y *F=0.5*. Respecto al tipo de cruzamiento, la mitad de las islas utiliza cruzamiento *exponencial* y la otra mitad utiliza cruzamiento *binomial.*

Caso 8: Todas las islas usan los valores de referencia *Cr=0.45* y *F=0.5*. Respecto al tipo de cruzamiento, la mitad de las islas utiliza cruzamiento *exponencial* y la otra mitad utiliza cruzamiento *binomial.*

Caso 9: Todas las islas usan valores aleatorios en el rango (0,1) para *Cr* y *F*. El tipo de cruzamiento es *binomial.*

Caso 10: Todas las islas usan valores aleatorios en el rango (0,1) para *Cr* y *F*. El tipo de cruzamiento es *exponencial.*

Caso 11: La mitad de las islas usan los valores de referencia para *Cr=0.3* y *F=0.5*, y la otra mitad utilizará valores aleatorios en el rango (0,1). El tipo de cruzamiento es *exponencial.*

Caso 12: La mitad de las islas usan los valores de referencia para *Cr=0.45* y *F=0.5*, y la otra mitad utilizará valores aleatorios en el rango (0,1). El tipo de cruzamiento es *exponencial.*

Caso 13: Todas las islas son configuradas con los valores de referencia *Cr=0.3* y *F=0.5.* El tipo de cruzamiento utilizado es *exponencial.*

Caso 14: Todas las islas son configuradas con los valores de referencia *Cr=0.45* y *F=0.5.* El tipo de cruzamiento utilizado es *exponencial.*

La Tabla 1 resume los casos de test. En ella se puede observar que se discriminó la primera y segunda mitad de las islas intervinientes, para mayor claridad de cada caso de test. Cuando se utilizan valores en rango de más menos dos puntos de cierto valor de referencia, la tabla utiliza la abreviatura "rango(N)", siendo N el valor de referencia. Si se utilizan valores aleatorios en el rango (0,1) la tabla utiliza la abreviatura "aleat.".

Se puede observar que algunos de los casos consideran valores aleatorios ya sea en un rango cercano a los valores de referencia (casos 3, 4, 5 y 6), o cualquier valor en el rango (0,1) (casos 9, 10, 11 y 12). La utilización de diferentes valores para *F* en una misma ejecución puede ampliar el espectro de los vectores que resultan de la diferencia que se realiza en la etapa de mutación. Por lo tanto, hacer que el factor de escala F tome diferentes valores puede ser una manera para incrementar el conjunto de potenciales vectores trial y minimizar el riesgo de estancamiento.

Tabla 1. Casos de test.

Caso	*1ra mitad de islas*			*2da mitad de islas*		
	Cr	*F*	*Ct*	*Cr*	*F*	*Ct*
1	0.3	0.5	bin	0.3	0.5	bin
2	0.45	0.5	bin	0.45	0.5	bin
3	rango (0.3)	0.5	bin	rango (0.3)	0.5	bin
4	rango (0.45)	0.5	bin	rango (0.45)	0.5	bin
5	rango (0.3)	rango (0.5)	bin	rango (0.3)	rango (0.5)	bin
6	rango (0.45)	rango (0.5)	bin	rango (0.45)	rango (0.5)	bin
7	0.3	0.5	bin	0.3	0.5	exp
8	0.45	0.5	bin	0.45	0.5	exp
9	aleat.	aleat.	bin	aleat.	aleat.	bin
10	aleat.	aleat.	exp	aleat.	aleat.	exp
11	0.3	0.5	exp	aleat.	aleat.	exp
12	0.45	0.5	exp	aleat.	aleat.	exp
13	0.3	0.5	exp	0.3	0.5	exp
14	0.45	0.5	exp	0.45	0.5	exp

Como mencionamos anteriormente, el tipo de cruzamiento está estrechamente relacionado con el tiempo de ejecución del algoritmo, por lo tanto uno de los puntos de análisis se centrará en la observación del tiempo de ejecución del modelo respecto de las diferentes configuraciones para las islas del factor *Ct*.

El desempeño del algoritmo fue probado con dos funciones escalables obtenidas de [13], habiendo seleccionado una función unimodal y otra multimodal. Se realizaron 30 ejecuciones con diferentes valores de semilla para cada función y para cada caso de test. Dado el volumen de la información experimental para todos los casos de test considerados (y el costo de coordinar su ejecución en un entorno compartido), se deja el estudio detallado de las demás funciones como trabajos futuros. No obstante, experimentaciones previas sobre gran parte del conjunto de funciones han demostrado que el comportamiento general del modelo CIBM-DE resulta consistente a todas ellas.

El tamaño de los problemas considerados es de 1000 y la población está integrada por 400 individuos. Las funciones utilizadas en las pruebas fueron Shifted Sphere (unimodal, con rango de búsqueda [-100,100]) y Shifted Rosenbrock (multimodal, con rango de búsqueda [-100,100]).

El modelo paralelo fue configurado con una tasa de migración del 15% de los individuos de la población cada 500 generaciones. Las pruebas realizadas fueron ejecutadas con diferentes configuraciones para las islas del modelo, lo que implica utilizar diferente cantidad de unidades de procesamiento. Los experimentos incluyeron 2, 4, 8 y 16 procesadores dedicados a los workers del modelo y un procesador extra para el master. Todas las

ejecuciones se llevaron a cabo en un cluster en el cual se utilizaron 5 nodos Intel Q9550 Quad Core de 2.83GHz y memoria RAM de 4GB DDR3 1333Mz. Todos los nodos están conectados a través de una red Ethernet de 1Gb/s. El software de base del cluster incluye el sistema operativo Debian 5 de 64 bits. En la codificación, utilizamos la librería de comunicación bajo el modelo de pasaje de mensajes MPICH [8].

por cada unidad de procesamiento. Se puede observar de la figura 2 que para la función Sphere se obtienen mejores resultados con el caso 2, es decir, cuando todas las islas toman valores de parámetro *Cr*=0.45 y *F*=0.5 (obtenidos en la *calibración de parámetros individuales*).

También se obtienen buenos resultados cuando se utilizan valores aleatorios cercanos a estos valores de referencia (casos 3 y 4).

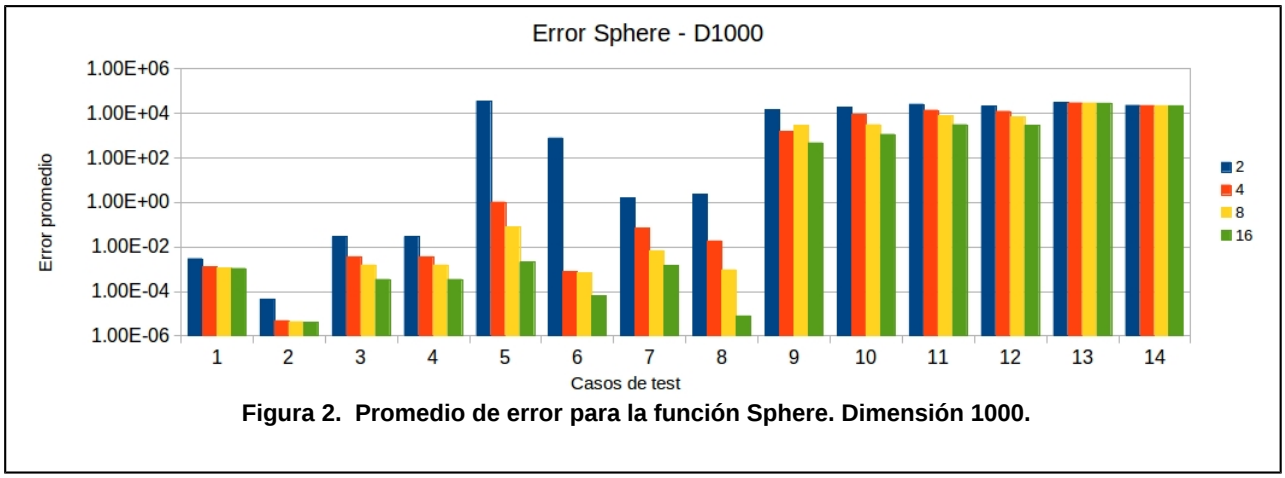


Figura 2. Promedio de error para la función Sphere. Dimensión 1000.

El error es definido como la diferencia entre el valor conocido del óptimo global y el valor obtenido por el algoritmo. Si el error es cero, significa que hemos encontrado el óptimo global. Para los problemas considerados, los mejores resultados son aquellos que están cercanos al valor cero.

Además, se puede observar que a medida que escalamos en cuanto a unidades de procesamiento, el modelo mejora la calidad de las soluciones, independientemente del caso de test que estemos considerando.

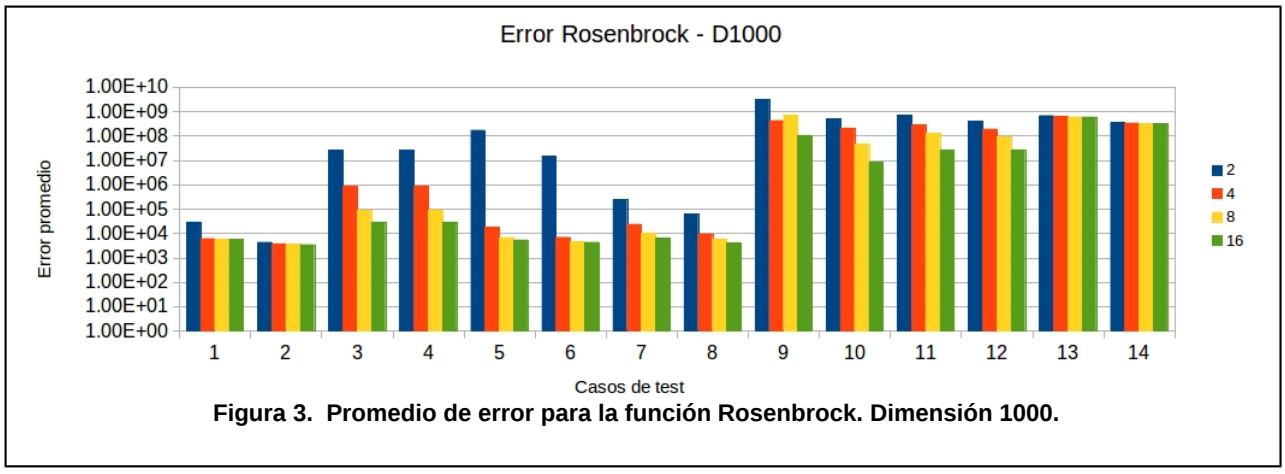


Figura 3. Promedio de error para la función Rosenbrock. Dimensión 1000.

Las figuras 2 y 3 muestran los resultados de error obtenidos con cada caso, para cada función. Cada grupo de barras corresponde a una prueba de escalabilidad de 2, 4, 8 y 16 procesadores, lo que equivale a tener una isla

Con esto corroboramos una vez más la propiedad de CIBM-DE de lograr mejoras en la calidad de las soluciones a medida que se incrementa la cantidad de islas del modelo, puesto que cada isla representa una

porción diferente del espacio de búsqueda. Podemos observar también que, en general, se obtienen mejores resultados utilizando un tipo de cruzamiento binomial (casos 1, 2, 3, 4, 5 y 6), o al configurar algunas islas con cruzamiento exponencial y otras con binomial (casos 7 y 8). Por otro lado, utilizar valores aleatorios en el rango (0,1) no es una buena elección para los parámetros *Cr* y *F*, puesto que los resultados que se obtienen no son los mejores (casos 9, 10, 11 y 12).

En contraparte, la utilización del tipo de cruzamiento exponencial logra una reducción de aproximadamente el 50% del tiempo de ejecución. Los casos 7 y 8 que utilizan ambos tipos de cruzamiento, obtienen, en su mayoría, valores de tiempo inferiores que el binomial. Esto es esperable dado que en el peor escenario posible el tiempo se ajusta al de aquellas islas que utilizan el tipo de cruzamiento binomial, que es el más lento.

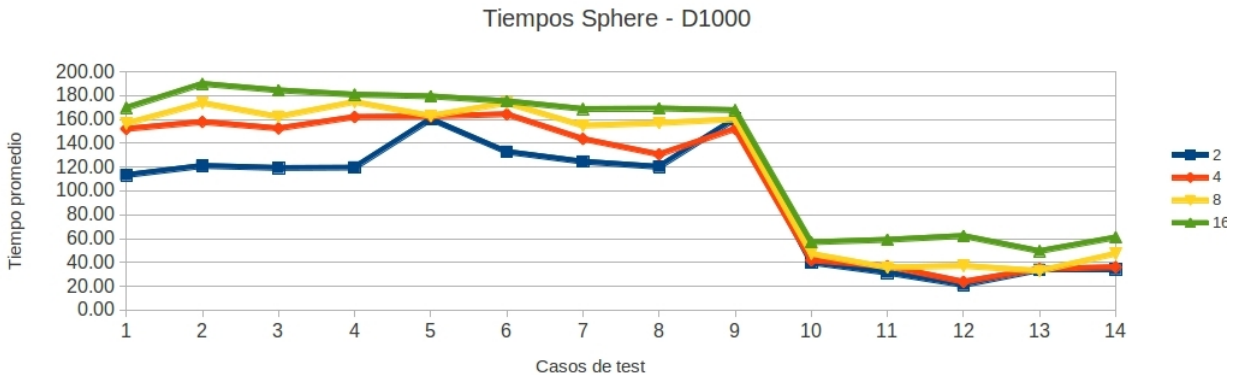


Figura 4. Promedio de tiempo para la función Sphere. Dimensión 1000.

La figura 3 muestra los resultados obtenidos con la función Rosenbrock. Podemos observar que se encuentran resultados similares a Sphere en cuanto a calidad de error.

Las figuras 4 y 5 muestran los resultados obtenidos en cuanto al tiempo de ejecución de las funciones de test para cada caso y para cada ejecución en escalabilidad. Se puede observar que el tiempo de ejecución es mucho mayor en los casos en donde se utiliza tipo de cruzamiento binomial.

Como observamos en las gráficas de error, la utilización de cruzamiento exponencial no produce los mejores resultados, sin embargo, en determinadas circunstancias puede ser necesario lograr una calidad de solución que no sea la mejor, pero a un costo temporal bajo. Haciendo un balance entre calidad y tiempo, podemos observar que existen dos escenarios.

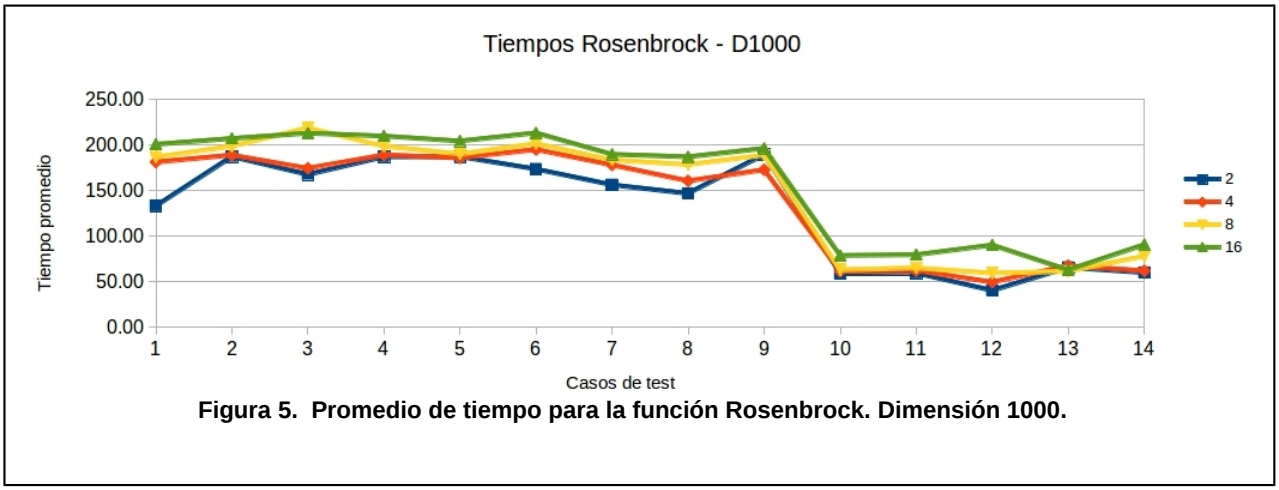


Figura 5. Promedio de tiempo para la función Rosenbrock. Dimensión 1000.

Si lo que se desea es lograr una reducción en el tiempo de ejecución, es recomendable utilizar el tipo de cruzamiento exponencial, o utilizar cierta cantidad de islas con este tipo de cruzamiento. Esto logra reducir el tiempo de ejecución del modelo, aunque la calidad de las soluciones obtenidas no es la mejor que se puede alcanzar. Si -por el contrario- se desea priorizar la calidad se recomienda utilizar tipo de cruzamiento binomial y establecer los valores para *Cr* y *F* dentro de un rango cercano a valores conocidos como efectivos.

Finalmente, es importante remarcar la utilidad de realizar una calibración de parámetros múltiple, puesto que si bien cada uno de manera individual es específico para un fin y cumple una función dentro del algoritmo, existe una relación intrínseca entre ellos que determina no solo el desempeño global de la metaheurística en cuanto a la calidad de la aproximación, sino también el tiempo global de ejecución. Un análisis combinado como el que se presenta en este trabajo ayuda a comprender esa relación, lo que permite tener un mejor control sobre este tipo de técnicas estocásticas paralelas.

5. Conclusiones

En este trabajo presentamos la sintonización de tres parámetros fundamentales del modelo CIMB-DE, un modelo paralelo basado en islas para la metaheurística Evolución Diferencial.

En la calibración, se realizaron diferentes pruebas que se basan en tres tipos principales de sintonización: *calibración a nivel de parámetros individuales*, *calibración combinada de parámetros* y *calibración respecto del volumen de entidades del modelo que adoptan cierta configuración*.

Del análisis se obtuvo que el modelo CIBM-DE obtiene mejores resultados utilizando los valores obtenidos en la *calibración a nivel de parámetros individuales*, es decir, configurando todas las islas con *Cr=0.45*, *F*=0.5 y *Ct*=binomial. A través del análisis realizado sobre la *calibración combinada* y la *calibración respecto del volumen de entidades que adoptan cierta configuración* pudimos observar que el tiempo de ejecución está íntimamente relacionado con el tipo de cruzamiento. El cruzamiento exponencial, si bien produce menor tiempo de ejecución promedio, no obtiene la mejor calidad de soluciones. Por este motivo, para lograr un balance entre calidad y tiempo se puede configurar ciertas islas con cruzamiento binomial y otras con cruzamiento exponencial, reduciendo los tiempos de ejecución y logrando una calidad de soluciones intermedia.

Por otro lado, CIBM-DE obtiene mejores soluciones a medida que se incrementa la cantidad de islas del modelo. Por ende, dependiendo de las unidades de procesamiento disponibles es posible incrementar aún más la calidad de las aproximaciones.

Como trabajo futuro se plantea estudiar y abordar las técnicas de sintonización dinámicas, de manera tal de lograr un entorno de ejecución paralelo/distribuido auto-ajustable para problemas de optimización. Además, se pretende aplicar el método a un problema específico del área de las Ciencias de la Tierra, como lo es la predicción de los incendios forestales.

6. Agradecimientos

Este trabajo ha sido financiado por UTN bajo el proyecto EIUTNME0002170, y por ANPCyT bajo el proyecto PRH PICT-2008-00242. Agradecemos también al Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la Universidad Nacional de San Luis por el soporte de hardware provisto para la realización de los experimentos.

7. Referencias

[1] Alba, E., Tomassini, M., "Parallelism and Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, 6, 5, Octubre 2002, pp. 443–462.

[2] Ali, M., Pant, M., Nagar, A., "Two Local Search Strategies for Differential Evolution". *IEEE Fifth International Conf. on Bio-Inspired Computing: Theories and Applications*, 2010, pp. 1429-1435.

[3] Amjady, N., Keynia, F., Zareipour, H., "Short-Term Load Forecast of Microgrids by a New Bilevel Prediction Strategy", *IEEE Transactions on Smart Grid*, 1, 3, Diciembre 2010, pp. 286-294.

[4] Apolloni, J., Leguizamón, G., García Nieto, J., Alba, E., "Island Based Distributed Differential Evolution: An Experiment Study on Hybrid Testbeds", *Eighth International Conference on Hybrid Intelligent Systems*, 2008, pp. 696-701.

[5] García-Martínez, C., Rodríguez, F., Lozano, M., "Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation", *Soft Computing*, 15, 11, 2011, pp. 2109-2126.

[6] Kozlov, K., Samsonov A., "New Migration Scheme for Parallel Differential Evolution", *Proc. Int. Conf. on Bioinf. of Genome Reg. and Structure*, 2006, pp. 141-144.

[7] Mattson, T., Sanders, B., Massingill, B., *Patterns for Parallel Programming*, Addison-Wesley Professional, chapter 5, 2004, pp. 143-152.

[8] MPICH Message Passing Interface, http://www.mpich.org/

[9] Noman, N., Iba, H., "Accelerating Differential Evolution Using an Adaptive Local Search", *IEEE Transactions on Evolutionary Computation*, 2008., pp. 107-125.

[10] Price, K., Storn, R., Lampinen, J., *Differential Evolution:*

A Practical Approach to Global Optimization, Springer, 2005.

[11] Saraswat, M., Arya, K.V., Sharma, H., "Leukocyte segmentation in tissue images using differential evolution algorithm", *Journal of Swarm and Evolutionary Computation*, 11, Febrero 2013, pp. 46-54.

[12] Talbi, E., *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[13] Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M., Yang, Z., "Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization", *Nature Inspired Computation and Applications Laboratory*, 2007.

[14] Tardivo, L., Cagninga, L., Leguizamón, G., "A Hybrid Metaheuristic Based on Differential Evolution and Local Search with Quadratic Interpolation", *Congreso Argentino de Computación*, 2012.

[15] Tardivo, M. L., Caymes-Scutari, P., Méndez-Garabetti, M., Bianchini, G., "Parameters Calibration for Parallel Differential Evolution Based on Islands", *XIX Congreso Argentino de Ciencias de la Computación*, Octubre 2013, pp. 296-305.

[16] Tardivo, M. L., Caymes-Scutari, P., Méndez-Garabetti, M., Bianchini,G., "Two Models for Parallel Differential Evolution", *Proc. of the High Performance Computing Latin American Symposium*, Julio 2013, pp. 26-36.

[17] Tasoulis, D., Pavlidis, N., Plagianakos, V., Vrahatis, M., "Parallel Differential Evolution", *Proc. of the Congress on Evolutionary Computation*, 2, Junio 2004, pp. 2023-2029.

[18] Weber, M., Neri., F., Tirronen, V., "Distributed differential evolution with explorative–exploitative population families", *Journal of Genetic Programming and Evolvable Machines*, 10, 4, 2009, pp. 343-371.

[19] Weber, M., *Parallel Global Optimization. Structuring Populations in Differential Evolution*. Jyväskylä University Printing House, 2010.

[20] Yang, G., "Battery parameterisation based on differential evolution via a boundary evolution strategy", *Journal of Power Sources*, 245, Julio 2014, pp. 583-593.

[21] Yang, Z., Tang, K., Yao, X., "Self-adaptative Differential Evolution with Neighborhood Search", *IEEE Congress on Evolutionary Computation*, 2008, pp. 1110-1116.

[22] Yang, Z., Tang, K.,Yao, X., "Scalability of generalized adaptative differential evolution for large-scale continuous optimization", *Soft Computing*, 15, 11, 2011, pp. 2141-2155.

[23] Ye, L., Cai, Q., Zhang, M., Tan,L., "Real-time observation, early warning and forecasting phytoplankton blooms by integrating in situ automated online sondes and hybrid evolutionary algorithms", *Journal of Ecological Informatics*, 22, Julio 2014, pp. 44-51.

[24] Zaharie, D., Petcu, D., "Adaptive Pareto Differential Evolution and Its Parallelization", *Proc. of the 5th Int. Conf. Parallel Processing and Applied Mathematics*, 3019, 2004, pp. 261-268.

[25] Zamuda, A., Brest, J., Bokovie, B., Zumer, V., "Large Scale Global Optimization Using Differential Evolution With Self Adaptation and Cooperative Co-evolution", *IEEE Congress on Evolutionary Computation*, 2008, pp. 1110-1116.

[26] Zhao, S., Suganthan, P., Das, S., "Self-adaptative differential evolution with multi-trajectory search for large-scale optimization", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 15, 11, Septiembre 2010, pp. 1-11.

[27] Zhao, Z., Wang, J., Zhao, J., Su, Z., "Using a Grey model optimized by Differential Evolution algorithm to forecast the per capita annual net income of rural households in China", *International Journal of Management Science*, 40, 5, Octubre 2012, pp. 525-532.

[28] Zhiqiang, W., Xin, Z., Wenxia, L., Boliang, L., "Substation Planning Based on Geographic Information and Differential Evolution Algorithm", Int. Conf. on Sustainable Power Generation and Supply, Abril 2009, pp. 1-7.

Análisis de consistencia de Casos de Uso con simuladores de autómatas finitos.

Marcelo Martín Marciszack, Oscar Carlos Medina, Claudia Castro, Enrique Humberto Moyano

GIDTSI, Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Maestro López esq. Av. Cruz Roja Argentina, Ciudad Universitaria – (5016) Córdoba

{marciszack, oscarcmedina, ingclaudiacastro, enriquemoyano}@gmail.com

Abstract
El objetivo del trabajo es obtener un método automatizado de análisis de consistencia de Casos de Uso. Para lo cual se construyó una aplicación web denominada SIAR (Sistema Integral de Administración de Requerimientos) que gestiona los requerimientos funcionales de un sistema de información según los lineamientos de UML (Lenguaje Unificado de Modelado).
Los casos de uso son una herramienta de generación y análisis de requisitos de sistemas. La finalidad principal de SIAR es la administración de casos de uso con una herramienta informática que agilice su registración, normalice su contenido y posibilite implementar validaciones funcionales, como por ejemplo un método automatizado de análisis de consistencia de casos de uso. Con este fin, el sistema genera un grafo con la transición de estados de cada caso de uso, expresado en el protocolo XPDL (Lenguaje de Definición de Flujo de Trabajo), que es analizado en un simulador de autómata finito determinista para verificar la cohesión de los escenarios en él definidos.

Palabras Clave
Caso de uso, Análisis de requerimientos, Autómata finito determinista, UML, XPDL.

Introducción

En la actualidad UML es reconocido como el estándar para el modelado de proyectos de software orientado a objetos.
Los componentes principales de esta metodología son los Casos de Uso que especifican el comportamiento deseado del sistema. Por definición el caso de uso "especifica una secuencia de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un particular actor" [1].

Una vez generado un caso de uso, es necesario comprobar y asegurar su validez. La verificación de consistencia de la secuencia de acciones descripta en el caso de uso es una de las tareas que permiten su validación.
Es deseable que esta verificación pueda realizarse de manera automatizada para lo cual se podría trabajar con autómatas finitos deterministas, ya que un autómata finito es un conjunto de estados y un control que se mueve de un estado a otro en respuesta a entradas externas [2]. Se llama determinista al autómata que puede estar únicamente en un estado en un momento determinado. El desafío es transformar el caso de uso en un autómata finito determinista para validar su cohesión secuencial.
Partiendo de estas premisas, el aporte que pretende hacer el presente análisis consiste en dar respuesta a la siguiente pregunta:
¿Cómo se definiría un método, basado en simuladores de autómatas finitos deterministas, que verifique la consistencia de la secuencia de acciones definida en un caso de uso, incluyendo variantes?.

Elementos del Trabajo y metodología

A continuación se presenta la metodología con una propuesta de tres fases:

a) Normalizar la registración del requerimiento
b) Transformar el caso de uso en una máquina de estados

c) Verificar la consistencia secuencial de los escenarios del caso de uso

a) Normalizar la registración del requerimiento:
Un requerimiento es una característica de diseño, una propiedad o un comportamiento de un sistema. Cuando se enuncian los requerimientos de un sistema se está estableciendo un contrato entre los elementos externos al sistema y el propio sistema, que establece lo que se espera que haga el sistema. Los requerimientos se pueden expresar de varias formas, desde texto sin estructura hasta expresiones en un lenguaje formal, pasando por cualquier otra forma intermedia. La mayoría de los requisitos funcionales de un sistema, si no todos, se pueden expresar con casos de usos [1].
Como el modelo de requerimientos tiene por objetivo la captura de requerimientos funcionales [3], surgió la necesidad de una herramienta propia que se llamó SIAR (Sistema Integral de Administración de Requerimientos), con el fin de brindar soporte a la construcción del modelo de requerimientos de un proyecto de software según el estándar UML. El desarrollo se programó con la plataforma GeneXus versión X evolution II por su alta eficiencia y flexibilidad para gestionar cambios, y la disponibilidad de licencias para uso académico de acuerdo al programa universitario de extensión celebrado con la empresa Artech.
SIAR es una aplicación web que permite registrar en forma normalizada los casos de uso y cuya primera versión comprende el siguiente alcance:

- Administración de los atributos de un proyecto de sistemas y sus versiones.

- Gestión de los alcances de cada versión del proyecto y los casos de uso asignados.

- Administración de los artefactos de un caso de uso, incluyendo actores, pre-condiciones, post-condiciones, escenario principal y escenarios alternativos, y su versionado.

- Clasificación, priorización y trazabilidad de los casos de uso.

- Visualización de consultas y generación de reportes en distintos formatos, inclusive XPDL, para comunicarse con otras aplicaciones.

- Diseño y validación del Modelo Conceptual.

- Gestión de atributos de procesos de negocio, de actividades de negocio que los componen y los casos de uso asociados a estas actividades.

- Registración y consulta de un glosario por proyecto, con entradas y sinónimos, siguiendo las recomendaciones de LEL (Léxico Extendido del Lenguaje), que es una estrategia de modelado de requisitos basada en lenguaje natural [4], que tiene por objetivo "comprender el lenguaje del dominio del problema y luego estudiar la dinámica de éste" [10].



Figura 1: Versionado de Proyectos y Casos de Uso en SIAR

b) Transformar el caso de uso en una máquina de estados:

En segunda instancia se definió un conjunto de reglas de conversión del caso de uso en un grafo de estados para que sea la entrada a un simulador de autómata finito determinista.
Una vez completa la versión de un caso de uso, se genera un grafo de estados a partir de las siguientes definiciones:

- Todo caso de uso tiene al menos un escenario principal.

- Un caso de uso puede no tener ningún, o tener uno, o tener varios escenarios alternativos.

- Cada paso de un escenario de un caso de uso responde a un estado y es un nodo de la máquina de estados.

- Todo caso de uso tiene un paso inicial, y es el primer paso de todos los escenarios. Este paso es el estado/nodo inicial.

- Todo caso de uso tiene un paso final por aceptación y es el último paso del escenario principal. También puede ser el último paso de algún escenario alternativo. Este paso es el estado/nodo final por aceptación.

- Un caso de uso puede no tener ningún final por error, o tener uno o varios finales por error, en cuyo caso son el último paso de un escenario alternativo. Estos pasos se denominan estados/nodos finales por error.

- Todo caso de uso pasa de un estado/nodo a otro por medio de una función de transición, que es una tabla de transición de estados donde se muestra a qué estado se moverá un autómata finito dado, basándose en el estado actual y otras entradas.

- Partiendo de un estado/nodo origen se puede continuar en un único estado/nodo destino, o en dos nodos/estados destino alternativos dependiendo de una condición de bifurcación.

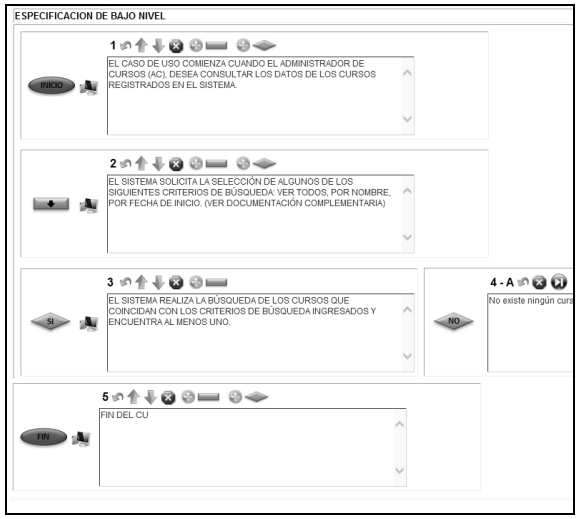


Figura 2: Bifurcación del escenario principal en la especificación de bajo nivel de un caso de uso.

- El grafo de estados asociado al caso de uso tiene un alfabeto de tres símbolos para indicar qué evento lo cambia de un estado/nodo a otro:
  - Símbolo "A" = Por medio de una Acción determinada.
  - Símbolo "S" = Cuando Si se cumple una condición que bifurca a un escenario alternativo.
  - Símbolo "N" = Cuando No se cumple una condición que bifurca a un escenario alternativo.

- Partiendo de un estado/nodo origen, en la función de transición puede estar asociado solamente uno de los símbolos: "A", "N" o "S". Con esto se cumple la condición necesaria de un autómata finito determinista.
  De esta manera, si la transición entre dos estados/nodos se da dentro de un mismo camino, se asocia el símbolo "A". Si en cambio interviene una bifurcación, la función de transición hacia el estado/nodo destino por cumplimiento de la condición de bifurcación, se asocia el símbolo "S". Por el otro camino de la bifurcación, se asocia el símbolo "N".

Tabla De Estados Casos de Uso: 1 - 1 - 1 - 1 CONSULTAR CURSOS Versión: 6

Estado / Paso Origen	Estado / Paso Destino	Transición	Estado Final por Error	Tipo
1	2	A		S
2	3	A		S
3	4	A		S
4	4 - A	N		C
4 - A	4 - A - 1	A		S
4 - A - 1	4 - A - 2	A	SI	S
4	5	S		C
5	6	A		S
6	6 - A	S		C
6 - A	6 - A - 1	A		S
6	7	N		C

Figura 3: Tabla de estados de un caso de uso. Transformación automática por SIAR.

- Una vez generado el grafo de estados se expresa en protocolo XPDL, por ser el más adecuado para intercambiar modelos de procesos entre distintas herramientas. Este protocolo de intercambio de datos está basado en XML y su sintaxis fue creada específicamente para representar el diagrama de flujo de un proceso.

```
<transicion
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="java:dominio.automatas.TransicionFinita">
    <simbolo>
        <simbolo>S</simbolo>
    </simbolo>
    <estado-inicio>
        <denominacion>4</denominacion>
    </estado-inicio>
    <estado-fin>
        <denominacion>5</denominacion>
    </estado-fin>
    <denominacion>f( 4, S) = 5</denominacion>
</transicion>
```

Figura 4: Fragmento de archivo XPDL generado por SIAR

El lenguaje XPDL "da soporte a la definición y a la importación / exportación de procesos, con el objetivo de que, aunque se modele un proceso en una aplicación, este modelo pueda ser usado por otras aplicaciones de modelado y/o por otras aplicaciones que trabajen en el entorno de ejecución" [5].

c) Verificar la consistencia secuencial de los escenarios del caso de uso:
Finalmente se ingresa el archivo XPDL, que representa al caso de uso como grafo de estados, al programa "Autómata Finito" que forma parte del conjunto de "Herramientas Prácticas para el Aprendizaje de Informática Teórica". Esta aplicación java "permite definir autómatas y gramáticas para la enseñanza y ejercitación de los alumnos de la asignatura Sintaxis y Semántica del Lenguaje que se dicta en la carrera de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional Facultad Regional Córdoba" [6].

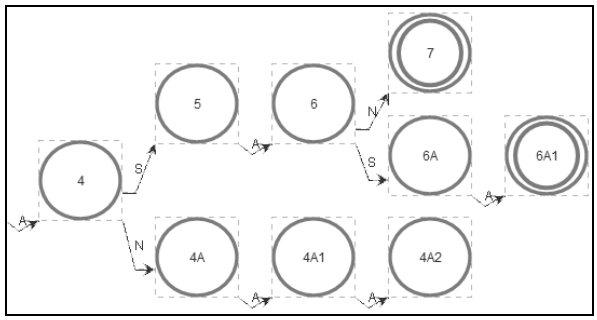


Figura 5: Fragmento del grafo de estados en el simulador de autómatas finitos

El simulador posibilita probar el grafo de estados y comprobar si es aceptado o rechazado. Si es aceptado, significa que la consistencia de los escenarios del caso de uso es correcta. Sino, el rechazo indica que en algunos de los caminos hay un error en la secuencia, pudiéndose visualizar el detalle para detectar la inconsistencia ya sean estados inconexos, inconsistencias o bucles infinitos. Luego, el error puede ser corregido en una nueva versión del caso de uso, generando su correspondiente grafo de estados automáticamente.
De esta manera se logra control y trazabilidad de los cambios de estados en todos los escenarios que conforman un caso de uso, el cual describe un requerimiento funcional.
Vale recordar a F. P. Brooks cuando dijo: "la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto" [7].

Resultados

El trabajo puesto en consideración en este artículo se originó dentro del proyecto de investigación "Validación de Requerimientos a través de Modelos Conceptuales" del GIDTSI (Grupo de Investigación, Desarrollo y Transferencia

de Sistemas de Información), dependiente del Departamento Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional, Facultad Regional Córdoba.
El proyecto "busca dar solución a uno de los principales problemas de la Ingeniería en Sistemas relacionado a la elicitación y especificación de requerimientos, que vincula las distintas etapas del proceso de desarrollo de software manteniendo la trazabilidad de los mismos hasta su validación e implementación" [8].
Para alcanzar esta meta, se han analizado y comprendido el procedimiento y el modelo de datos asistido por una aplicación web denominada SIAR que gestiona los requerimientos funcionales de un sistema de información según los lineamientos de UML y permite implementar un método automatizado para validar la cohesión de un caso de uso desde el punto de vista de la transición de estados definidos intrínsecamente en los pasos de su especificación funcional. Haciendo posible también enlazar este proyecto con un trabajo académico de la carrera Ingeniería en Sistemas de Información, del Grupo de Herramientas Didácticas de Informática Teórica que contribuyó con el simulador de autómatas finitos. La decisión de desarrollar una herramienta propia y no utilizar una de las ya existentes que soportan modelado de casos de uso en UML, fue tomada porque a futuro nos permite ampliar sus funcionalidades integrando otros programas y/o metodologías de acuerdo a los avances del proyecto y a las necesidades de comprobación de nuestra investigación.

Discusión

Así como es aceptado universalmente en la actualidad el uso de UML para el modelado de sistemas, también son reconocidos los errores comunes en la especificación de requerimientos utilizando casos de uso. Según J.A. Pow-Sang: "Lamentablemente, la bibliografía existente, muestra muchas formas de aplicar los casos de uso y no son pocas las veces en que su empleo es inadecuado. Algunas de las causas son: mala interpretación del estándar UML y secuencia incorrecta de actividades para la creación de casos de uso." [9].
Además "una especificación incorrecta de requerimientos puede ocasionar errores ocultos dentro del sistema" [10] que repercutan en efecto dominó sobre las fases posteriores del desarrollo de un sistema, provocando lo que Mizuno [11] denominó una "catarata de errores".
Aunque ya se sabe que ninguna metodología es infalible, no implica que por ello no sea perfectible. Por ejemplo, Brooks promueve hacer foco en resolver la complejidad esencial del proceso de software, no perdiendo tiempo en las complejidades accidentales, para lograr importantes mejoras, por él estimadas "mayor que diez veces en un período de diez años" [7]. Coincidiendo en esta línea de pensamiento, el proyecto de investigación que enmarca este trabajo, se ocupa de la *validación y trazabilidad de los requerimientos* de un sistema de información al momento de formular su Modelo Conceptual.
Se identificó como una de las validaciones factibles de implementar la consistencia de los escenarios definidos en un caso de uso. Para ello se reutilizó el conocimiento y el desarrollo informático de otro proyecto de la Facultad logrando un resultado positivo en tiempo y forma. La justificación de que el mencionado desarrollo esté basado en autómatas finitos deterministas es parte de las conclusiones de un trabajo anterior donde se expone que las propiedades de validación de estas máquinas abstractas "son consideradas completas en la validación de los mismos, ya que el único punto que no está presente en comparación de otras herramientas de model Checker, es el control de deadlock, o bloqueo mutuo, el cual no es necesario considerarlo ya que por la particularidad de la elección dentro de las máquinas Abstractas, como son los Autómatas Finitos, su definición, configuración y forma de funcionamiento no permiten que esta situación se plantee ya que las entradas son secuenciales y

procesadas únicamente con desplazamiento en un solo sentido." [12].
Sin embargo, es pertinente la observación de por qué no se investigó en líneas paralelas y complementarias software derivado de "Model Checking" [13]; ya que este mismo planteo se hizo el grupo de trabajo en forma interna y se lo incluyó como uno de los próximos alcances del proyecto. Motivó la decisión el trayecto requerido para abordar esta técnica en forma análoga, como ser el proceso propuesto por Shinkawa [14] para verificación de modelos formales de sistemas, y poder agregar a SIAR una salida adicional en lenguaje de ProMeLa (Proceso o Protocolo Meta Lenguaje) y una interfase con el "model checker" SPIN.
Otra limitación de alcance de este análisis que corresponde aclarar, es que aplica solamente en forma individual para cada caso de uso y no se revisa la consistencia "entre" los casos de uso de un mismo Modelo Conceptual.
Resumiendo, dado que "el proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo." [1], la validación del diseño es de primordial importancia para la construcción e implementación de un sistema. Por ello, se presenta este método de análisis simple y concreto, en pos de sumar un eslabón más a la validación del Modelo Conceptual que formaliza el diseño de un software, pues como dijo Edsger W. Dijkstra: "La simplicidad es un prerrequisito para la fiabilidad".

Conclusión

A la pregunta planteada en la Introducción *¿Cómo se definiría un método, basado en simuladores de autómatas finitos deterministas, que verifique la consistencia de los escenarios definidos en un caso de uso?*, se responde con la puesta a consideración este método automatizado, basado en el aplicativo SIAR que registra, normaliza y transforma un caso de uso a formato XPDL y un simulador de autómatas finitos, que permite verificar la consistencia secuencial de los distintos caminos del caso de uso.
El aporte que realiza SIAR al proyecto en el que fue concebido, es el de constituirse como plataforma de software integradora de las aplicaciones que se utilizan en cada una de las líneas de investigación.

Agradecimientos
A las respectivas familias, por su apoyo a nuestra labor académica, y a los colegas investigadores del grupo GIDTSI de U.T.N. F.R.C..

Referencias

[1] J. Rumbaugh, I. Jacobson, G.Booch. "The Unified Modelling Language Reference". Addisson Wesley 1999.
[2] Samarjit Chakraborty. "Formal Languages and Automata Theory-Regular Expressions and Finite Automata-". Computer Engineering and Networks Laboratory Swiss Federal Institute of Technology (ETH) Zurich. Marzo 2003.
[3] Ivar Jacobson y otros.Object Oriented Software Engineering. A Use Case Driven Approach. Addison Wesley, 1992.
[4] C. Leonardi, J.C.S. Leite, Gustavo Rossi. "Una estrategia de Modelado Conceptual de Objetos, basada en Modelos de requisitos en lenguaje natural". Tesis de Maestría Universidad Nacional de la Plata Año 2001.
[5] J. D. Pérez. "Notaciones y lenguajes de procesos. Una visión global". Tesis de Doctorado Universidad de Sevilla.
[6] Proyecto Construcción de Herramientas Didácticas para la enseñanza y ejercitación práctica en laboratorio de Informática Teórica en las Carrreras con Informática. "Manual de Usuario – Grupo de Herramientas Didácticas". Sitio web:http://www.profesores.frc.utn.edu.ar/sistemas/ssl/Marciszack/GHD/Main.htm.U.T.N. – F.R.C. 2009.
[7] Frederick P. Brooks. "No Silver Bullet. Essence and Accidents in Software Engineering". IEEE Computer. Abril 1987.
[8] Marcelo Marciszack, Ramiro Pérez y Claudia Castro. "Validación de Requerimientos a través de Modelos Conceptuales – Modelos y Transformaciones". WICC 2013.
[9] Pow-Sang, J.A., La Especificación de Requisitos con Casos de Uso: Buenas y Malas Prácticas, II Simposio Internacional de Sistemas de Información e Ing. de Software en la Sociedad del Conocimiento-

SISOFT 2003, Pontificia Universidad Católica del Perú, Lima-Perú, 200.
[10] Antonelli, Leandro. "Traceability en la elicitación y especificación de requerimientos". Tesis de Magister en Ingeniería de Software. Facultad de Informática. Universidad Nacional de La Plata. La Plata, Febrero 2003.
[11] Mizuno Y., "Software Quality Improvement", IEEE Computer, Vol. 16, No. 3, Marzo 1983.
[12] Manuel Perez Cota, Mario Groppo, Marcelo Marciszack. "Validación de Especificaciones Funcionales en el modelado de Esquemas Conceptuales a través de Máquinas Abstractas". CoNaIISI 2013. 1er. Congreso Nacional de Ingeniería Informática / Sistemas de Información. Córdoba, Argentina. Noviembre 2013.
[13] Edmund Clarke, Allen Emerson, and Joseph Sifakis. Model Checking: Algorithmic Verification and Debugging. ACM 2007 Turing Award.
[14] Yoshiyuki Shinkawa. "Model Checking for UML Use Cases". Faculty of Science and Technology, Ryukoku University. 2008.

Datos de Contacto:

Marcelo Martín Marciszack, marciszack@gmail.com – Oscar Carlos Medina, oscarcmedina@gmail.com – Claudia Castro, ingclaudiacastro@gmail.com – Enrique Humberto Moyano, enriquemoyano@gmail.com.
GIDTSI, Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información - Universidad Tecnológica Nacional - Facultad Regional Córdoba –Maestro López esq. Av. Cruz Roja Argentina, Ciudad Universitaria – (5016) Córdoba.

Perspectivas de Usabilidad para Aplicaciones Móviles: Un Caso Práctico de Evaluación de Facebook

Lucas Santos, Pablo Becker, Luis Olsina
Universidad Nacional de La Pampa, Facultad de Ingeniería,
General Pico, La Pampa, Argentina
santos.ls@live.com, [beckerp,olsinal]@ing.unlpam.edu.ar

Abstract
La cantidad de dispositivos y aplicaciones móviles ha ido creciendo exponencialmente, por lo que comprender la calidad de dichas aplicaciones se ha tornado últimamente un asunto de relevancia en la industria de software. Para este fin, es necesario comprender qué características de las aplicaciones móviles influyen en los usuarios para que utilicen satisfactoriamente determinadas aplicaciones. El presente trabajo está enfocado a especificar uno de estos aspectos claves: la Usabilidad. Se da una caracterización de la misma donde se especifica las distintas perspectivas de Usabilidad, tanto desde el punto de vista de calidad de sistema como de calidad de sistema en uso. Además, se resalta la importancia que tiene el contexto relacionado tanto a la aplicación como al sistema en uso, en proyectos de medición y evaluación. Como caso práctico, se evalúa la Usabilidad de la aplicación móvil Facebook empleando la estrategia GOCAME (Goal-Oriented, Context-Aware Measurement and Evaluation). Además, se ilustra las capacidades de método y proceso de esta estrategia a medida que se desarrolla la evaluación desde la perspectiva de calidad de sistema.

Palabras Clave
Usabilidad. Experiencia de Usuario. Evaluación. GOCAME. Aplicación Móvil. Facebook.

1. Introducción

En el presente mundo globalizado, la comunicación entre personas ya no requiere obligatoriamente de encuentros en lugares físicos, sino que con tan solo un clic es suficiente para poder tener encuentros virtuales. Y gracias a los avances tecnológicos de la información y de las comunicaciones, ya no estamos atados a nuestras PCs de escritorio dado que existen dispositivos como tablets, smartphones, entre otros, con el objetivo de brindarnos movilidad y conexión.

Recientemente, han surgido diferentes aplicaciones para poder satisfacer prácticamente cualquier necesidad que un usuario tenga. Dentro de estas aplicaciones se encuentran las tan populares redes sociales. Una de las redes sociales con mayor penetración en el mercado es Facebook. Esta surgió a inicios del 2004 y, poco a poco, fue ganando popularidad hasta convertirse en el segundo sitio más accedido en todo el mundo [1]. Facebook permite estar conectado con amigos, compartir noticias, subir fotos y hasta mostrar el estado de ánimo del usuario. Estas funcionalidades y servicios contribuyeron a que Facebook logre posicionarse como la principal red social.

Si tenemos en cuenta que la opinión de los usuarios es uno de los factores más importantes para que las aplicaciones triunfen o fracasen, hay que analizar qué es lo que influye sobre estos para que adopten o abandonen determinada aplicación. Aquí es donde entra en juego el concepto de las perspectivas (o vistas) de calidad, y más específicamente la de Calidad en Uso (Quality in Use, en inglés, abreviado de ahora en más QinU) de las aplicaciones (sistemas) en uso. En ISO 25010, QinU se define como "*el grado en que un producto o sistema puede ser usado por usuarios específicos para lograr metas específicas con efectividad, eficiencia, libre de riesgo y satisfacción en un contexto específico de uso*" [2]. En otras palabras, QinU representa lo que el usuario percibe y siente al utilizar la aplicación, cómo realiza las tareas de la aplicación y si logra completarlas correctamente en determinados contextos de uso.

Sin embargo, desarrollar aplicaciones que satisfagan un alto nivel de QinU no es una tarea sencilla. Una manera de alcanzarlo es satisfaciendo previamente un alto nivel de calidad de sistema, la cual representa a la perspectiva de calidad externa de la aplicación (*External Quality*, o EQ). Según ISO 25010, la EQ se define como el "*grado en que un producto de software permite que el comportamiento del sistema satisfaga las necesidades explícitas e implícitas cuando es utilizado bajo condiciones específicas*" [2]. En dicha norma, existen relaciones entre estas dos perspectivas de calidad. Por un lado tenemos la relación "influencia", la cual va desde EQ hacia QinU y significa que un alto nivel de EQ influye o contribuye a alcanzar un alto nivel de QinU. Por otro lado, tenemos la relación "depende de" que está dirigida desde QinU hacia EQ, la que indica que el nivel de satisfacción de QinU depende del nivel de EQ alcanzado por el sistema. Notar que "influencia" es una relación necesaria pero no suficiente.

Por otra parte, EQ está compuesta por aquellas características y atributos que pueden ser evaluados

sobre la aplicación en estado simulado de ejecución o aún operativo. Mientras que QinU está compuesta de aquellas características y atributos que son evaluados sobre la aplicación en uso siendo utilizada por usuarios finales en un contexto real de uso.

Según el estándar citado, EQ se compone de ocho características, como por ej. Seguridad, Usabilidad, Confiabilidad, entre otras. En el presente trabajo, nos concentramos en la característica Usabilidad, la cual se define en ISO 25010 como el "*grado en que un producto o sistema puede ser usado por usuarios específicos para cumplir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto especifico de uso*" [2]. Notar que esta definición de Usabilidad es similar a la dada para QinU, lo cual causa cierta confusión por pertenecer al mismo estándar. Con el fin de distinguir claramente estos dos conceptos, en este trabajo proponemos el uso de la definición previa de Usabilidad dada en ISO 9126, a saber "*la capacidad de un producto software de ser comprendido, aprendido, utilizado y atractivo para el usuario cuando es utilizado bajo condiciones específicas*" [3], con una ligera adaptación, como veremos más adelante.

Considerado a la perspectiva de QinU, hemos modelado características tales como Usabilidad en Uso (Actual Usability) y Experiencia de Usuario (User Experience, UX), extendiendo de esta manera el modelo ISO 25010. Particularmente, en trabajos previos [4] hemos desarrollado el marco de calidad denominado 2Q2U (*Quality, Quality in use, actual Usability and User experience*), el cual enlaza las vistas de calidad relacionando características de EQ con las de QinU. Estas relaciones muestran por ejemplo una correspondencia entre Usabilidad de EQ, con Usabilidad en Uso y UX de QinU. Por este motivo es que consideramos relevante que cuando se habla de Usabilidad se debe precisar desde qué perspectiva se focaliza (EQ? o QinU?) además de precisar si se consideran sub-características de Usabilidad, de Usabilidad en Uso y/o de UX. Estas características se abordarán con más detalle en secciones posteriores.

Por otra parte, cabe preguntarse: ¿cómo se obtiene el nivel de calidad de una aplicación para saber si satisface o no los indicadores de desempeño esperados? Una manera de comprender qué niveles de calidad ha alcanzado un ente concreto, es empleando una estrategia de medición y evaluación (M&E) apropiada. En este trabajo utilizamos a GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation strategy*) [4], la cual es una estrategia de M&E integrada, que sigue un enfoque orientado a objetivos, sensible al contexto y centrada en la organización. Está fundamentada en tres pilares fundamentales, a saber: 1) un marco conceptual de M&E centrado en ontologías, 2) especificaciones de las vistas del proceso, y 3) en especificaciones de métodos con soporte metodológico/tecnológico.

Es más, utilizando adecuadamente el marco de modelos de calidad 2Q2U y sus dos perspectivas, junto con una estrategia de evaluación, cambio y mejora continua, se pueden explorar las relaciones existentes entre las características de QinU y EQ. Esto se ha documentado en un caso de estudio en [5], en donde al realizar una evaluación de QinU los evaluadores identifican los problemas de Usabilidad en Uso y determinan aquellos atributos de Usabilidad (para la vista de EQ) que pueden estar relacionados con QinU. Luego de evaluar EQ, realizar recomendaciones e implementar cambios en la aplicación para resolver los problemas de EQ, se realiza una re-evaluación de QinU para determinar las mejoras obtenidas en los indicadores de desempeño.

Por último, otro aspecto relevante a considerar es el contexto. En ISO 25010, se incluye la característica Cobertura de Contexto únicamente en el modelo de QinU. Sin embargo, esta es una característica que debería estar representada independientemente de los modelos de calidad utilizados, como discutiremos más adelante.

En resumen, las contribuciones de esta investigación consisten en: i) caracterizar a la Usabilidad a partir de características y relaciones entre Usabilidad, Usabilidad en Uso, Experiencia de Usuario y Contexto; ii) diseñar los requerimientos no funcionales teniendo como foco a la Usabilidad para la aplicación móvil Facebook, además de diseñar e implementar la M&E usando la estrategia GOCAME; y iii) brindar recomendaciones a partir de la evaluación de Facebook con el objetivo de planificar cursos de acción con el fin de mejorarla.

El presente artículo se organiza de la siguiente manera: la Sección 2 discute trabajos relacionados sobre Usabilidad de aplicaciones móviles y web, además de trabajos sobre calidad en uso. La Sección 3 aborda aspectos relacionados a la caracterización de Usabilidad, Usabilidad en Uso y Experiencia de Usuario aplicables al dominio de sistemas móviles; además se discute la importancia de especificar el contexto. La Sección 4 describe un caso de estudio, utilizando la estrategia de M&E GOCAME, para comprender el nivel de Usabilidad (desde la perspectiva de EQ) alcanzado por la aplicación Facebook para dispositivos móviles. Finalmente, la Sección 5 resume las principales conclusiones y trabajos futuros.

2. Trabajos Relacionados

Una de las principales referencias en el área de calidad de producto/sistema de software es el estándar ISO 25010 [2] (que sustituye al estándar ISO 9126-1[3]). En este estándar se especifican dos perspectivas de la calidad junto a sus respectivos modelos, uno para producto o sistema y otro para sistema en uso. El modelo de calidad de sistema está compuesto por 8 características entre las cuales se encuentra Usabilidad; sin embargo, Usabilidad en

Uso y UX son características ausentes en el modelo de QinU de ISO.

Por otro lado, también existen trabajos sobre Usabilidad para aplicaciones móviles. Ejemplos de ellos son las guías sobre diseño de aplicaciones e interface de usuario de Apple [6] y Google [7]. Cabe mencionar que en estos trabajos no es explícita la relación entre los entes de las aplicaciones móviles y los conceptos de Usabilidad y UX en modelos de calidad. Tampoco se observa esta relación en otros trabajos sobre Usabilidad como en [8], ni en trabajos relacionados a calidad como [9] y [10]. Por ejemplo, Nielsen desarrolló en [8] una lista de características y checklists para aplicaciones móviles, que son deseables o necesarias en determinados contextos de uso, pero sin utilizar modelos o vistas de calidad. Por lo tanto, la capacidad de utilizar un marco de calidad y estrategias para correlacionar y aplicar estos conceptos, evaluarlos y finalmente mejorar una aplicación móvil es bastante limitada.

Con respecto a Usabilidad en aplicaciones móviles existen algunos trabajos que tratan de explicar distintos aspectos de una aplicación. Por ejemplo, cuando se trata de crear un menú o enlaces que tengan un diseño apropiado, [11] propone algunas reglas o pautas para el buen diseño de las opciones del menú, como: 1) etiquetas claras y fáciles de interpretar, 2) navegación consistente por el sitio, y 3) que provea predicción basada en las acciones que haya realizado el usuario. También aconsejan no mostrar listas largas de opciones para evitar o minimizar el esfuerzo del usuario y que la estructura del menú le facilite al usuario realizar las tareas con un mínimo de interacciones.

Un trabajo digno de mencionar es [12] donde se evalúa la Usabilidad de aplicaciones móviles de una manera analítica. La evaluación se realiza utilizando un modelo jerárquico y multinivel de factores de Usabilidad. Estos factores relacionados con vistas y entes, son medidos para dar un puntaje único mediante la utilización de checklists. El marco conceptual que utilizan en [12] para especificar los indicadores de Usabilidad [13] está basado en el modelo de medición de la ISO 15939 [14], en el cual sus términos están estructurados en un glosario en vez de ontologías.

Con respecto al contexto y contexto en uso, Zhang y Adipat [15], ponen de manifiesto algunos problemas que pueden afectar a la Usabilidad de las aplicaciones móviles, algunas de estas son el tamaño de la pantalla, la resolución de la pantalla, los métodos de inserción de datos. En nuestro trabajo tenemos en cuenta propiedades de contexto del dispositivo tales como tipo y modelo de dispositivo, tamaño de la pantalla, entre otros.

A continuación, se brinda una caracterización para Usabilidad y sus perspectivas en consideración de categoría de entes, y aspectos relacionados al contexto para aplicaciones móviles.

3. Caracterizando a la Usabilidad de una Aplicación Móvil

Los dispositivos móviles, como smartphones y tablets, ofrecen una pantalla pequeña comparada con una PC convencional aunque permiten a los usuarios utilizar aplicaciones (tanto nativas como web) en cualquier lugar, donde factores como la ubicación, la hora del día, el perfil del usuario y el rendimiento de la red tienen un impacto importante sobre la calidad percibida por el usuario, y por lo tanto de la experiencia de usuario.

En esta sección se analizarán los factores relevantes para comprender características como Usabilidad y UX para aplicaciones móviles. Con este fin, en la Figura 1 se muestra un esquema cuyos bloques establecen relaciones entre: i) categorías de ente, vistas de calidad y sus características, y propiedades medibles (cajas verdes y naranjas); ii) medición (caja celeste); y iii) evaluación (caja rosa). Primero, nos enfocaremos en i) para facilitar la comprensión acerca de los requerimientos no funcionales y luego poder especificar atributos de Usabilidad y UX para evaluaciones basadas en interfaces, tareas y percepciones.

La etiqueta “ente” (entity) en la caja superior de la Figura 1, representa a la categoría de ente objetivo (target) a ser evaluada. Esta se define como “*categoría de objeto que será caracterizada midiendo sus atributos*”, mientras que un atributo es “*una propiedad física o abstracta de una categoría de ente, que puede ser medida,*” [16]. Existen dos instancias de mayor nivel de abstracción de categorías de ente que son de interés para la evaluación de Usabilidad y UX, a saber: producto/sistema y sistema en uso. Al mismo tiempo, una categoría de ente puede estar compuesta por sub-entes, por ejemplo, una aplicación móvil está compuesta por objetos GUI (*Graphic User Interface*) básicos o avanzados, como se muestra en la Figura 1. Notar que estos objetos pueden contener botones, iconos, etiquetas, widgets, etc.

En el bloque superior de la Figura 1 también se observan las etiquetas Contexto y Contexto en Uso. El Contexto puede definirse como “*un tipo de ente especial que representa el estado de la situación de una categoría de ente, el cual es relevante para una necesidad de información particular*” [16]. Por lo tanto, un ente Sistema está caracterizado por un ente de Contexto que posee sub-entes y atributos relacionados al dispositivo, sistema operativo, etc., sobre el cual se realiza la evaluación, mientras que un ente Sistema en Uso está caracterizado por un ente de Contexto en Uso, el cual puede contener sub-entes contextuales de entorno, usuario y tareas.

Por otro lado, un ente (categoría de ente) y sus sub-entes asociados no pueden ser medidos y evaluados directamente, sino mediante las propiedades medibles asociadas a ellos, es decir los atributos de los entes y propiedades de contexto.

Estas propiedades deben ser seleccionadas cuidadosamente a partir de la vista de calidad (de sistema, de sistema en uso, etc.) que se trata evaluar. Para cada vista se utilizan los modelos de calidad que pueden estar personalizados para distintos tipos de entes. Estos modelos especifican requerimientos de calidad tanto de producto/sistema como de sistema en uso gracias a las características incluidas que pueden a su vez estar relacionadas a sub-características, las cuales combinan atributos. Como se muestra en la Figura 1, tanto EQ como QinU poseen características, tales como Usabilidad, Confiabilidad, Calidad Funcional, etc. para EQ y Experiencia de Usuario, Satisfacción y Usabilidad en Uso para QinU. Tanto los requerimientos de producto/sistema como los de sistema en uso, están modelados bajo una perspectiva de calidad, es decir, las vistas de EQ y QinU respectivamente.

Finalmente, dentro del bloque superior de la Figura 1, se pueden observar las relaciones "usa" (uses) y "relacionado a" (*related to*) que enlaza los conceptos de contexto (tanto contexto como contexto en uso) con el producto/sistema y sistema en uso. En el bloque inferior se observan unas relaciones entre EQ y QinU, que significan que la calidad de sistema "influencia" (*influences*) a la calidad en uso y, por el contrario, la calidad en uso "depende" (*depends on*) de la calidad de sistema.

Como indicamos en la Introducción, una vez que los modelos de EQ y QinU están instanciados, estas relaciones pueden ser exploradas a través de los atributos y características del ente concreto mediante la realización de evaluaciones. Por ejemplo, si utilizamos una estrategia de evaluación como en [5], podríamos explorar las relaciones entre los atributos de calidad de sistema y calidad de sistema en uso, para implementar mejoras. Esto permite conocer qué atributos de EQ afectan a la QinU y viceversa.

Siguiendo el esquema de la Figura 1, las siguientes sub-secciones van a examinar características de Usabilidad, Usabilidad en Uso, Experiencia de Usuario y Contexto focalizándonos en aplicaciones móviles.

3.1. Caracterizando a la Usabilidad

Usabilidad es una de las 9 características de calidad de sistema definidas en 2Q2U v2.0, y se la define como el "*grado en que un producto o sistema posee atributos que le permiten ser entendido, fácilmente aprendido, operado, protegido contra errores, atractivo y accesible por el usuario, cuando es utilizado bajo ciertas condiciones*" [17]. Notar que 2Q2U no toma la definición de Usabilidad dada en ISO 25010 como se indicó en la Introducción, sino que utiliza una definición adaptada de [3].

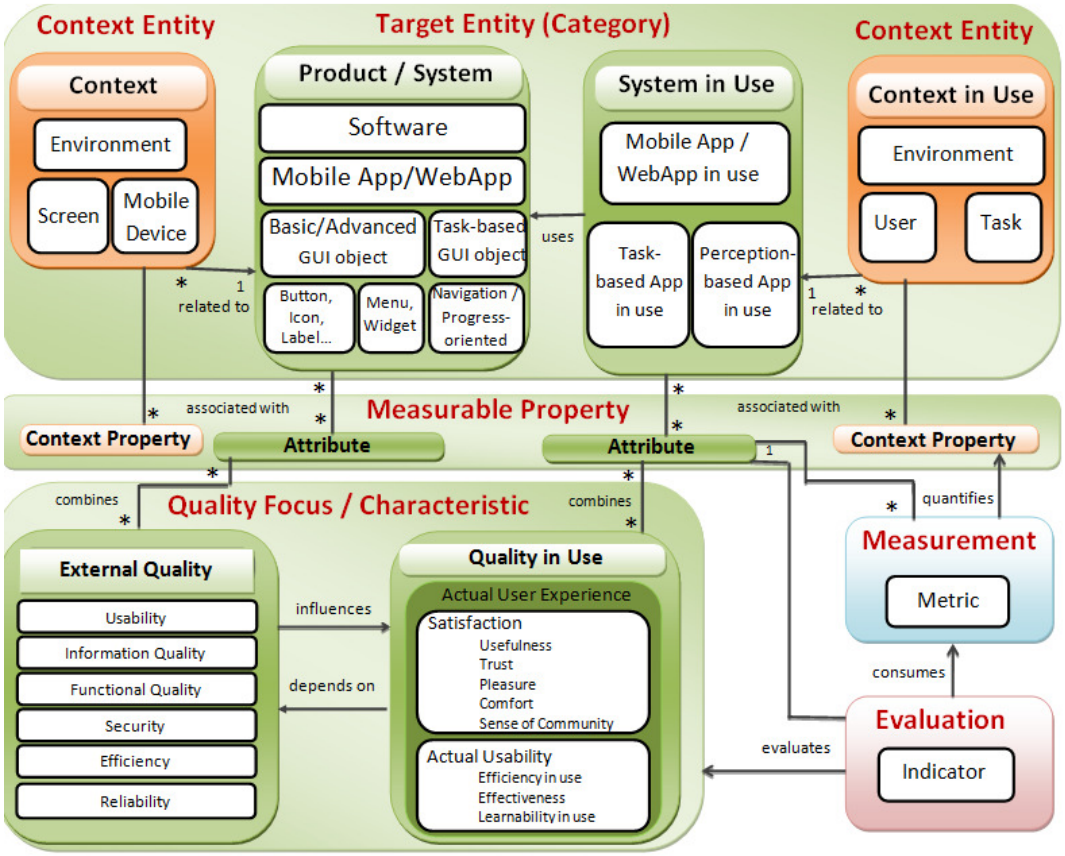


Figura 1. Esquema global en el que se establecen las principales relaciones entre los bloques que representan a: Categoría de Entes (Producto/Sistema, Sistema en uso y Contexto), Focos de Calidad (EQ y QinU), Propiedades Medibles, Medición (usando Métricas) y Evaluación (usando Indicadores).

El ente producto, al que hace referencia la definición, es un objeto que se encuentra en las primeras etapas del ciclo de vida del software (por ejemplo, diagramas, esquemas de interface, código, etc.), mientras que el ente sistema es un producto de software ejecutable (por ejemplo, aplicaciones móviles en etapa de testeo u operativa pero no en el contexto real de uso con usuarios reales) que puede incluir tanto software como hardware.

Podemos observar en [17] que un ente sistema (en nuestro caso, una aplicación móvil) está asociado con atributos, y estos se combinan a sub-características de Usabilidad tales como Comprensibilidad, Facilidad de Aprendizaje, Operabilidad, Protección contra Errores, Estética de la Interface de Usuario y Accesibilidad. En la primer columna de la Tabla 1 se pueden ver algunos atributos y sub-características de Operabilidad usados en el caso de estudio de Facebook, el cual será analizado en la Sección 4. Estos atributos se agrupan en sub-características y características (tal como se aprecia en la Tabla 1), y están asociados a sub-entes concretos como pantallas, barra de botones, etc.

Notar que por cuestión de espacio, en la Tabla 1 no se incluye la totalidad de características, sub-características y atributos que fueron utilizados en el caso de estudio. La tabla completa junto con todas las definiciones se encuentra en [17]. A seguir, definimos sólo algunos de los atributos y sub-características que se ilustrarán en la Sección 4.

La sub-característica Visibilidad se define como: "*Grado en que la aplicación permite facilitar el uso mediante la utilización de controles y textos que pueden ser vistos y distinguidos por el usuario con el fin de tomar decisiones apropiadas*". Para esta sub-característica hemos especificado dos sub-sub-características: Visibilidad Apropiada del Color (código 1.3.2.1 en la Tabla 1) y Tamaño Apropiado del Objeto GUI (código 1.3.2.2). La primera combina dos atributos, *Diferencia apropiada de brillo (1.3.2.1.1)* y *Diferencia apropiada de color (1.3.2.1.2)*. Mientras que la segunda está compuesta por los atributos *Tamaño apropiado del control (1.3.2.2.1)* y *Tamaño apropiado del texto (1.3.2.2.2)*.

Por ejemplo, podemos estudiar el comportamiento de 1.3.2.1 ya que contiene atributos que influencian a la legibilidad mientras los usuarios realizan las tareas de la aplicación. Hemos definido a Visibilidad Apropiada del Color como el "*grado en que los colores de los objetos GUI principales (textos, controles, widgets, etc.) comparados con el color de fondo (background) provee un suficiente contraste y visibilidad apropiada*". El atributo *1.3.2.1.1* se define como el "*grado en que el color de un objeto GUI (foreground) comparado con el color de fondo (background) provee una diferencia de brillo apropiada*". Y el segundo atributo (*1.3.2.1.2*) se define como el "*grado en que el color de un texto o control (foreground) comparado con el color de fondo (background) provee una diferencia de color apropiada*".

Los tipos de objetos GUI que se tienen en cuenta para estos atributos, suelen ser los textos de la aplicación, los botones, los campos de texto, entre otros. En la Figura 1 se pueden observar distintos sub-entes para GUIs básicas o avanzadas.

3.2. Caracterizando a la Calidad en Uso

La vista de QinU ayuda a comprender el impacto que tiene el sistema en uso sobre usuarios reales en contextos de uso reales, tal como un usuario enviando un mensaje a través de su dispositivo móvil mientras va caminando por la calle.

En la ISO 25010 se define un modelo de QinU el cual está compuesto por cinco características, a saber: Efectividad, Eficiencia, Satisfacción, Libre de Riesgo y Cobertura del Contexto. En [4] argumentamos sobre la necesidad de incluir la característica de mayor nivel (ver Figura 1) denominada Experiencia de Usuario (UX), la cual definimos como el "*grado en que un sistema en uso le permite a usuarios específicos lograr sus metas específicas con satisfacción, usabilidad en uso y libertad de riesgo en contextos específicos de uso*". Además, en [17] se pueden encontrar definiciones de características y sub-características ausentes en la ISO 25010 o que fueron redefinidas en 2Q2U v2.0.

UX es determinada por la satisfacción del usuario al lograr las metas de performance o "*do goals*" y las metas hedónicas o "*be goals*" [18]. Los "*do goals*" están relacionados con el usuario cuando este logra cumplir con sus metas de tarea con efectividad y eficiencia en uso (Usabilidad en Uso), mientras que los "*be goals*" están relacionados con las percepciones de satisfacción del usuario al utilizar el sistema real. La característica (o sub-característica) Satisfacción contiene a su vez sub-características relacionadas a la percepción subjetiva del usuario, como Utilidad, Confianza, Placer, Comodidad y Sentido de Comunidad [4].

Usabilidad está relacionada con la especificación y evaluación de sub-características y atributos del sistema que están asociados a la interface y sus interacciones, mientras que la Usabilidad en Uso está relacionada con la especificación y evaluación de sub-características y atributos de una aplicación en uso que están basados en tareas. A su vez, Satisfacción está relacionada con las sub-características y atributos evaluados mediante percepciones. Como las sub-características combinan atributos que están asociados a entes, hemos considerado, en la Figura 1, dos sub-entes típicos de las aplicaciones en uso, a saber: los basadas en tarea y los basados en percepción (Task-based, and Perception-based App in-use).

Tabla 1. Usabilidad, sub-características, atributos, pesos (W), operadores (Op), valor de indicadores elementales (EI), parciales (PI) y globales (GI) obtenidos realizando una evaluación en el caso de estudio de Facebook. Notar que esta tabla es un fragmento de la mostrada en [17].

Características y Atributos	W	Op	EI	PI/GI
1. Usabilidad		C-		***60.5***
1.1. Comprensibilidad	0.20	A		**76.1**
1.2. Facilidad de Aprendizaje	0.25	C--		**59.7**
1.3. Operabilidad	0.35	C--		**80.7**
1.3.1. Facilidad del Ingreso de Datos	0.20	A		**90**
1.3.1.1. Defaults	0.40		**100**	
1.3.1.2. Entrada obligatoria	0.20		**50**	
1.3.1.3. Widget apropiado	0.40		**100**	
1.3.2. Visibilidad	0.40	A		**81.5**
1.3.2.1. Visibilidad Apropiada del Color	0.50	C-+		**100**
1.3.2.1.1. Diferencia apropiada del brillo	0.50		**100**	
1.3.2.1.2. Diferencia apropiada del color	0.50		**100**	
1.3.2.2. Tamaño Apropiado del Objeto GUI	0.50	C--		**63**
1.3.2.2.1. Tamaño apropiado del control (widget)	0.40		**100**	
1.3.2.2.2. Tamaño apropiado del texto	0.60		**42.1**	
1.3.3. Consistencia	0.40	C--		**75.5**
1.3.3.1. Permanencia de los Controles	0.50	C--		**57.3**
1.3.3.1.1. Permanencia de los controles principales	0.80		**54.9**	
1.3.3.1.2. Permanencia de los controles contextuales	0.20		**67.4**	
1.3.3.2. Estabilidad de los controles	0.50		**95.5**	
1.4. Protección Contra Errores	0.10	A		**8.4**
1.5. Estética de la Interface de Usuario	0.10	A		**80.8**

Los sub-entes que están basados en tareas, pueden ser evaluados utilizando atributos de Eficiencia en uso, Eficacia, y Facilidad de Aprendizaje en uso (Learnability in use), mientras que los sub-entes que están basados en percepciones pueden ser evaluados utilizando cuestionarios de Satisfacción, considerando a los atributos como los ítems del cuestionario.

Como se evidenció en el caso de estudio realizado en [5], para obtener un alto nivel de performance en Usabilidad en Uso, es necesario satisfacer un alto nivel de Usabilidad de la aplicación o sistema. Para nuestro caso, si en una aplicación móvil tenemos botones que son para el usuario legibles, consistentes y en definitiva fáciles de operar, y además se ofrece en otros controles ayuda contextual, o se provee opciones predeterminadas en selecciones, o indicadores de campo obligatorio al rellenar formularios, etc. esto incrementaría el nivel de Usabilidad e influenciaría a que el usuario logre realizar los objetivos de las tareas con eficacia y eficiencia en uso. Por lo que Usabilidad puede influenciar a la Usabilidad en Uso y aún más a la Experiencia de Usuario.

Un aspecto a resaltar en la evaluación de sub-características de Usabilidad en Uso es que las tareas seleccionadas de la aplicación en uso tienen que ser evaluadas mientras están siendo realizadas por usuarios reales, en el contexto real de uso. En consecuencia, no es tan fácil lograr niveles altos de performance en Usabilidad en Uso, sino se tiene en cuenta varios aspectos clave de diseño que tengan impacto positivo sobre la Eficacia y la Eficiencia en realizar las tareas. Por lo tanto, una tarea tiene que ser desarrollada con el objetivo de ser utilizada en una aplicación móvil en cualquier contexto posible. Esta debe ser sencilla y con la menor cantidad de pasos posibles. Por ejemplo, si el usuario está realizando otra actividad (sea una reunión, caminando, etc.) mientras usa la aplicación en un contexto dado (de día o de noche, con luz artificial o natural), no dispone así de demasiado tiempo ni capacidad de atención.

A causa de estos factores, entre otros, las tareas en aplicaciones móviles no deberían ser demasiado largas, ya que de lo contrario potencian a que el usuario cometa errores y que no pueda completarlas correctamente.

3.3. Caracterizando al Contexto de una Aplicación Móvil

Contexto es un tipo especial de ente que representa el estado de situación del ente objetivo (Target Entity, en Figura 1) a ser evaluado, el cual es relevante para una necesidad de información. El

contexto para un proyecto de M&E de QinU es particularmente importante, así como la instanciación de los requerimientos de QinU debe ser hecho de manera consistente y en el mismo contexto para que las evaluaciones y las mejoras puedan ser realizadas y comparadas. Pero también es importante el contexto para la vista de EQ. En la Figura 1 mostramos el bloque superior izquierdo que representa a un ente contexto relacionado al ente sistema, el cual está caracterizado por sub-entes tales como entorno, dispositivo, pantalla, etc.

Como comentamos en [19] el contexto en uso de una aplicación móvil es muy diferente al de una aplicación web de escritorio. El tamaño de la pantalla, la resolución, etc., son claves a la hora de la evaluación de la aplicación. Además existen otros factores como el entorno en el que se encuentra el usuario, lo que conlleva a tener en cuenta el tiempo de uso, localización, etc. que tienen un impacto directo en la calidad de la experiencia del usuario.

Algunos de esos factores contextuales son: i) *La Actividad:* es lo que el usuario está haciendo mientras utiliza la aplicación; ii) *Momento del día:* es cuándo y en qué momento el usuario utiliza la aplicación; iii) *Desempeño de la Red:* esto se refiere a la velocidad con la que la aplicación envía y recibe datos.

En contraposición con el que podría denominarse contexto en uso, está el contexto relacionado al sistema. Este ente trata de especificar aquellos factores que tienen una influencia sobre la aplicación en estado de ejecución. Debido a los grandes avances tecnológicos, en la actualidad existe una gran variedad de dispositivos, los cuales son diferentes entre sí. Por lo que consideramos importante tener en cuenta este contexto a la hora de evaluar la EQ. Algunos de estos factores que influencian a la EQ son: i) *Tipo de aplicación móvil:* define el tipo de aplicación, si es una aplicación nativa –aquellas aplicaciones que están desarrolladas directamente para ser instaladas en el dispositivo-, o si es una aplicación móvil web –aquellas que son accedidas a través de un navegador; ii) *Versión-tipo de sistema operativo:* define el tipo y versión del sistema operativo que utiliza el dispositivo en el cual se realiza la evaluación; iii) *Tamaño de la pantalla:* el tamaño de la pantalla tiene una influencia directa sobre lo que se puede mostrar en una aplicación y cómo esta se debería adaptar a los distintos tamaños del mercado. En [8] se clasifica el tamaño de la pantalla dependiendo de la generación del dispositivo, a saber: "Regular cellphone" –aquellos teléfonos con pantallas pequeñas; "Mid-sized smartphone" –aquellos teléfonos con una pantalla de mediano tamaño (generalmente aquellos que poseen teclado y pantalla en el mismo dispositivo); "Full-sized smartphone" –aquellos teléfonos con una pantalla grande con tamaño similar al del dispositivo.

Estos entes y propiedades de contexto y/o contexto en uso deben registrarse en proyectos de M&E, dado que se relacionan al ente objetivo de la evaluación de EQ o de QinU. Y si bien, en nuestra estrategia, no son parte de los modelos instanciados de EQ o QinU, deberían ser registrados para alcanzar mayor consistencia y nivel de comparación en el análisis de datos y recomendaciones entre proyectos de M&E concretos.

4. Estrategia de M&E aplicada a Evaluar la Usabilidad de la Aplicación Móvil Facebook

El área de M&E y Análisis es de importancia para aquellas organizaciones que proveen productos o servicios. Este área ofrece información relevante para la toma de decisiones, por ejemplo, para comprender y/o mejorar la calidad de un producto o sistema. Para obtener resultados consistentes, comparables y reproducibles, es indispensable contar con una estrategia de M&E que tenga en cuenta tres capacidades: i) *una base conceptual de M&E*; ii) *la especificación del proceso de M&E*; y iii) *la especificación de métodos*.

En la actualidad existen algunas estrategias de M&E pero no todas poseen dichas capacidades integradas tal como ofrece GOCAME [4, 16]. A seguir, se describen las tres capacidades sobre las que se funda GOCAME, en tanto se ilustra un caso de estudio realizado sobre la aplicación Facebook para dispositivos móviles, con el fin de comprender el nivel de satisfacción de Usabilidad alcanzado, y proponer recomendaciones de cambio si las hubiere.

4.1. Base Conceptual de M&E

La base conceptual de M&E en GOCAME está definida en una ontología [20] de la cual surge el marco conceptual C-INCAMI (*Contextual–Information Need, Concept model, Attribute, Metric and Indicator*) [16]. Este marco está estructurado en seis componentes, a saber: a) Proyecto de M&E; b) Requerimientos No-Funcionales; c) Contexto; d) Medición; e) Evaluación; y f) Análisis y Recomendación.

Estos componentes contienen los términos y relaciones claves del dominio de M&E. Algunos de los términos son: Necesidad de Información, Contexto, Categoría de Ente, Atributo, Métrica, Indicador, entre otros. La Figura 2 muestra los principales términos y relaciones de los componentes b) y c). Así, "Sistema" y "Sistema en Uso" en Figura 1, son instancias de Categoría de Ente (Entity Category *name* en Figura 2). Al mismo tiempo, una Categoría de Ente puede contener sub-categorías como "Menú", "Botón", "Widget", entre otros. Un Ente representa a un objeto concreto como por ejemplo, "Aplicación móvil de Facebook" el cual pertenece a la categoría de ente "Sistema".

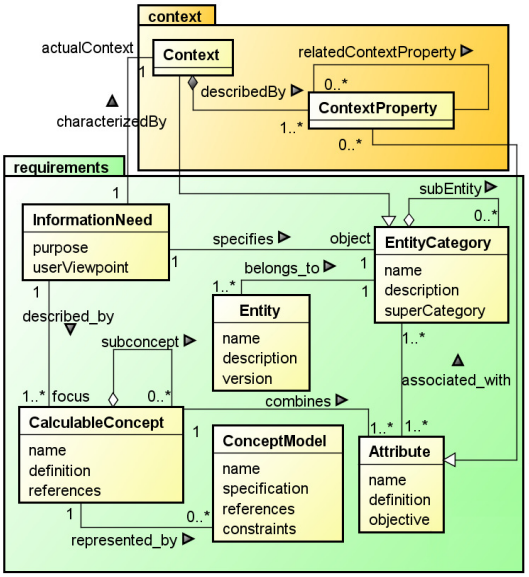


Figura 2. Principales términos y relaciones de los componentes de Contexto y Requerimientos no-funcionales.

El componente de requerimientos (Figura 2) especifica la Necesidad de Información para un proyecto de M&E, es decir, el propósito (por ej. "comprender", "mejorar") y el punto de vista del usuario ("usuario final", "desarrollador"). Esta Necesidad de Información tiene como foco un Concepto Calculable, es decir, características como "Usabilidad", "UX", etc. Además, un Modelo de Concepto para "Usabilidad" contiene sub-características como "Operabilidad" y puede estar instanciado con Atributos, formando un árbol de requerimientos tal como se muestra en la primer columna de la Tabla 1.

Por otra parte, Métrica es un término clave del componente de Medición. Este componente permite especificar Métricas Directas e Indirectas usadas para producir Medidas. Una Métrica es "*el procedimiento definido de medición o calculo y la escala*". Por cuestiones de espacio se omiten varias definiciones, tales como Indicador, etc. que pueden ser vistas en [16].

4.2. Proceso de M&E en GOCAME

En segundo lugar, GOCAME cuenta con un proceso de M&E bien establecido que garantiza repetitividad al realizar sus actividades [21]. Este proceso cuenta con 6 actividades principales, a saber: A1) *Definir los requerimientos no-funcionales;* A2) *Diseñar la medición;* A3) *Implementar la medición;* A4) *Diseñar la evaluación;* A5) *Implementar la evaluación;* y A6) *Analizar y recomendar*. A seguir se ilustran estas actividades usando el caso de estudio de Facebook.

4.2.1. Definir los requerimientos no-funcionales. En A1 se define la necesidad de información, el concepto foco a ser evaluado y sus sub-características, la categoría de ente y el contexto. En nuestro estudio, el propósito de la necesidad de información es "comprender" el nivel de satisfacción alcanzado para la vista de EQ, evaluando particularmente las fortalezas y debilidades de "Usabilidad", desde el punto de vista del "usuario final". La "aplicación móvil Facebook, Versión 3.8 al 23/10/2013 para Android" es el ente concreto y algunas propiedades de contexto son a) mobile device type, b) screen size y c) mobilephone generation, entre otras, cuyos valores son respectivamente: a) "mobilephone", b) "540x960px/ 4.3inches", y c) "full-sized smartphone".

Usabilidad se divide a su vez en varios conceptos calculables de menor nivel de abstracción (sub-características en Tabla 1) como "Comprensibilidad" (1.1), "Facilidad de Aprendizaje" (1.2), "Operabilidad" (1.3), "Protección contra Errores" (1.4) y "Estética de la Interface de Usuario" (1.5). A su vez, cado una de estos sub-conceptos se dividen en sub-características, como por ejemplo "Operabilidad" se divide en "Facilidad de Ingreso de Datos" (1.3.1), "Visibilidad" (1.3.2) y "Consistencia" (1.3.3), que combinan atributos asociados a los entes. Por ejemplo, "Visibilidad Apropiada del Color" (1.3.2.1) combina 2 atributos asociados a objetos de tipo "Color/Texto", tal cual se resalta en la Figura 3.

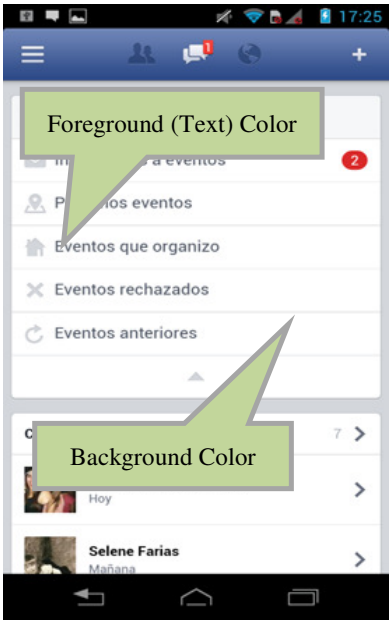


Figura 3. Una pantalla de Facebook utilizada en la medición de atributos de Visibilidad.

4.2.2. Diseñar la medición. En la actividad A2, el evaluador es el encargado de seleccionar desde un repositorio, las métricas adecuadas para cada atributo del árbol de requerimientos. En la Tabla 2 se muestra la especificación de la métrica "*Promedio de la Diferencia de Brillo Apropiada (ABDA)*" seleccionada para cuantificar al atributo 1.3.2.1.1.

Tabla 2. Especificación de las Métricas utilizadas para cuantificar el atributo 1.3.2.1.1.

Indirect Metric

Name: Average of Brightness Difference Appropriateness (ABDA).
Objective: Calculate the average of the level of brightness difference in the selected screens.
Author: Santos L.; **Version:** 1.0;
Quantified Attribute: *Brightness Difference Appropriateness* (1.3.2.1.1)

Calculation Procedure:

Formula: $\text{ABDA} = \frac{\sum_{i=1}^{n} BDLi}{n}$
for i=1 to n, where n is the number of selected screens (i.e., the set of selected application screens).
Numerical Scale:
Representation: Continuous; **Value Type:** Integer
Scale type: Interval
Unit/Name: Average of Brightness Difference;
Acronym: ABD.

Related Metric

Name: Brightness Difference Level (BDL)
Objective: Determine the level of brightness difference for the given screen.
Author: Santos L.; **Version:** 1.0
Tool: http://tools.cactusflower.org/analyzer/

Calculation Procedure: The agent (e.g. an expert o tool) inspects the screen and takes a Red value1, Green value1 and Blue value1 values of the foreground (text) color and a Red value2, Green value2 and Blue value2 values of the background color. (Note that it could be done using some tools like Photoshop or others tools used for image design).
Formula:
F = ((Red value1 * 299) + (Green value1 * 587) + (Blue value1 * 114)) / 1000

B = ((Red value2 * 299) + (Green value2 * 587) + (Blue value2 * 114)) / 1000

BDL = |F-B|
Reference: [http://tools.cactusflower.org/analyzer/].

Numerical Scale:
Representation: Continue; **Value Type:** Real
Scale type: Interval
Unit/Name: Brightness Difference; **Acronym:** BD

Esta métrica tiene como objetivo "*determinar en promedio cuán apropiada es la diferencia de brillo, para las pantallas seleccionadas de la aplicación*". Cabe mencionar que para el caso de estudio se diseñaron un total de 48 métricas, entre indirectas y directas.

4.2.3. Implementar la medición. Una vez seleccionadas todas las métricas, se puede ejecutar la actividad A3. Esta actividad es la encargada de producir las medidas, utilizando métricas, para cada uno de los atributos del árbol de requerimientos. Las mediciones fueron realizadas sobre la versión 3.8 de de Facebook para Android, entre el 26 y el 28 de Dic. del 2013. Particularmente, para el atributo *1.3.2.1.1*, el valor obtenido fue 178.9 aplicando el procedimiento de cálculo especificado en la Tabla 2.

4.2.4. Diseñar la evaluación. En la actividad A4 los participantes expertos se encargan de definir los indicadores elementales y globales y los niveles de aceptabilidad. Mientras que un indicador elemental evalúa el nivel de satisfacción de un requerimiento elemental, es decir, un atributo del árbol de requerimientos, un indicador parcial/global evalúa el nivel de satisfacción de requerimientos parciales como (sub-)características y requerimiento global (concepto foco) del árbol de requerimientos.

En el presente estudio se utilizaron tres rangos de aceptabilidad en escala porcentual, a saber: un valor entre 0-60 (rango insatisfactorio –rojo-) indica que se deben realizar acciones de cambio con alta prioridad; un valor entre 60-80 (rango marginal –amarillo-) indica que se debería realizar acciones de cambio; y un valor entre 80-100 (rango satisfactorio –verde-) indica que no es necesario ningún cambio. En la Tabla 3 se muestra la especificación del indicador elemental que interpreta el valor del atributo 1.3.2.1.1.

Tabla 3. Especificación del Indicador Elemental utilizado para evaluar al atributo 1.3.2.1.1.

Attribute: *Brightness Difference Appropriateness*
Coded: 1.3.2.1.1
Elementary Indicator:
Name: Performance Level of the Brightness Difference Appropriateness (P_BDA)
Author: Santos L.; **Version:**1.0;

Elementary Model:
Specification:the mapping is: P_BDA = 0 iff ABDA< 110; P_BDA = 75 iff ABDA >= 110 & ABDA < 135; P_BDA = 100 iff ABDA >= 135.

Decision Criterion:
[Three Acceptability Levels]
Name 1: Unsatisfactory; **Range:** if 0 ≤ P_BDA <60
Description: Indicates change actions must be taken with high priority
Name 2: Marginal; **Range:** if 60 ≤ P_BDA <80
Description: Indicates a need for improvement actions
Name 3: Satisfactory; **Range:** if 80 ≤ P_BDA ≤ 100
Description: Indicates no need for current actions

Numerical Scale:
Value Type: Real; **Scale Type:** Ratio
Unit/Name: Percentage; **Acronym:**%

Por otra parte, cabe preguntarse si por ejemplo para alcanzar un nivel de aceptabilidad satisfactorio para la sub-característica "Visibilidad Apropiada del Color" (1.3.2.1), no se debería satisfacer en simultáneo a los atributos 1.3.2.1.1 y 1.3.2.1.2. Es decir, debido a que estos dos atributos están relacionados, es necesario establecer un tipo de dependencia de conjunción entre ellos a la hora de calcular la evaluación parcial. En el presente caso de estudio, las relaciones de dependencia en el árbol de requerimientos la representamos utilizando un modelo de agregación lógico como lo es *Logic Scoring of Preference* (Puntaje Lógico de Preferencias o LSP) [22].

El modelo LSP de agregación de atributos, sub-características y características (y su procedimiento de cálculo) utiliza pesos y operadores lógicos. Los pesos establecen la importancia relativa de los elementos del árbol de requerimientos (ver segunda columna de la Tabla 1). En tanto que los operadores modelan relaciones de independencia (A), simultaneidad (C, u operadores de conjunción), y remplazabilidad (D, u operadores de disyunción). Por ejemplo, en la tercer columna de la Tabla 1, vemos que el operador de cuasi-conjunción C-+ es usado para producir la salida del indicador parcial de 1.3.2.1, cuyas entradas son los valores de indicador elemental de 1.3.2.1.1 y 1.3.2.1.2.

4.2.5. Implementar la evaluación. Una vez finalizadas las actividades A3 y A4 se puede realizar recién la actividad A5. En esta actividad se calculan los valores de los diferentes indicadores, tanto elementales como parciales y global. En el caso del atributo 1.3.2.1.1, el valor de indicador fue del 100 % alcanzando un nivel de aceptabilidad totalmente "Satisfactorio", lo que significa que este atributo no requiere acciones de mejora. Por el contrario, el atributo 1.3.2.2.2 denominado *Tamaño Apropiado del Texto* obtuvo el valor de 42.1 % cayendo en el nivel de aceptabilidad "Insatisfactorio". Esto significa que es necesario recomendar acciones de cambio urgentes para mejorar el desempeño de este atributo (recordando que un atributo representa un requerimiento no-funcional elemental a satisfacer).

4.2.6. Analizar y recomendar. En esta actividad (A6) se tiene en cuenta todos los artefactos y resultados producidos en las actividades anteriores (por ejemplo, especificación de requerimientos no-funcionales, especificación de métricas, especificación de indicadores, valores medidos y valores de indicador calculados) con el fin de analizarlos y generar recomendaciones. El documento de recomendación declara la potencial acción para mejorar aquellos atributos que alcanzaron niveles de aceptabilidad por debajo del nivel "Satisfactorio". En la Tabla 4, se detallan recomendaciones para algunos atributos de este caso de estudio, con sus respectivas prioridades.

Como se puede ver en la 5ta columna de la Tabla 1, Usabilidad alcanzó un nivel de aceptabilidad marginal (60.5%), lo que significa que existen sub-características que poseen niveles bajos de aceptabilidad y por lo tanto se requiere que se tomen acciones de mejora sobre los atributos que afectan negativamente a estas sub-características. Por ejemplo, Operabilidad (1.3) alcanzó un nivel del 80.7% de satisfacción, esto significa que hay atributos que fueron satisfactorios y otros que fueron marginales o insatisfactorios. Así, el concepto calculable Visibilidad Apropiada del Color (1.3.2.1) obtuvo 100% en su indicador parcial y, más precisamente 100% en sus dos atributos (1.3.2.1.1 y 1.3.2.1.2). Por el contrario, Permanencia de los Controles (1.3.3.1) obtuvo un valor de 57.3%, y sus dos atributos asociados (1.3.3.1.1 y 1.3.3.1.2) obtuvieron 54.9% y 67.4% respectivamente. Por lo tanto, se puede recomendar acciones de cambio.

Tabla 4. Extracto del artefacto "Informe de Recomendación" producido en la actividad Analizar y Recomendar. (Nota: el acrónimo A significa prioridad Alta, es decir, se recomienda actuar de un modo urgente. M significa Moderada, e implica que se debería actuar).

ID	Recomendación	Atributo	Prioridad
R1	**1.** Asegurar que en el conjunto de pantallas seleccionadas, el tipo de letra sea el apropiado para una correcta legibilidad. **2.** Asegurar que en el conjunto de pantallas seleccionadas, el tamaño de la letra sea el apropiado para una correcta visibilidad y legibilidad.	*Tamaño apropiado del texto (1.3.2.2.2)*	**A**
R2	**1.** Asegurar que en el conjunto de pantallas seleccionadas en los controles de entrada de los formularios cuyo ingreso de datos sea obligatorio tengan el soporte o indicador adecuado.	*Entrada obligatoria (1.3.1.2)*	**A**
R3	**1.** Asegurar que en la barra de controles principales todos los botones (controles) principales permanezcan consistentemente, en el conjunto de pantallas seleccionadas.	*Permanencia de los controles principales (1.3.3.1.1)*	**A**
R4	**1.** Asegurar que los controles contextuales permanezcan consistentemente en el conjunto de pantallas seleccionadas y que sean apropiadas al contexto.	*Permanencia de los controles contextuales (1.3.3.1.2)*	**M**

Además, si tenemos en cuenta las sub-características -que no se muestran en la Tabla 1, por razones de espacio- como Comprensibilidad (1.1), Facilidad de Aprendizaje (1.2) y Protección Contra Errores (1.4), las mismas lograron niveles de aceptabilidad marginal, para la primera, e insatisfactorio para las otras dos. Lo que significa que algunos de sus indicadores elementales alcanzaron un nivel bajo de satisfacción y, que por lo tanto, necesitan recomendaciones para dar soporte al diseño e implementación a un plan de acciones de cambio.

Para nuestro caso de estudio, estas acciones de cambio se pueden planificar a partir del documento de Informe de Recomendación. Estas recomendaciones con sus prioridades se detallan en la Tabla 4 considerando solamente a los atributos mostrados en la Tabla 1 cuyos indicadores elementales cayeron por debajo del nivel de aceptabilidad Satisfactorio. El informe con las cinco recomendaciones, junto a la documentación de la especificación de las métricas (una mostrada en la Tabla 2) sumado a los registros de la implementación de la medición, sirven de gran ayuda al momento de realizar la planificación de los cambios, ya que se puede establecer específicamente qué atributo o capacidad está débil o ausente en sub-entes concretos (por ejemplo pantallas específicas, barra de controles principales o campos de texto concretos, etc.) de la aplicación.

Por ejemplo, si observamos la recomendación R1.2 que se realiza en base a la baja performance del indicador elemental (42.1%) asociado al atributo *Tamaño del Texto Apropiado* (1.3.2.2.2) indica que hay que "*asegurar que en el conjunto de pantallas seleccionadas, el tamaño de la letra sea el apropiado para una correcta visibilidad y legibilidad*". Como consecuencia, se debe analizar y localizar en el registro de mediciones los ID de las pantallas en las cuales el texto no tenía un tamaño apropiado.

Posteriormente, se puede decidir desarrollar un plan de cambio para este atributo en esas pantallas concretas, usando alguna técnica de cambio específica sobre la versión actual del producto. Si los cambios fueran implementados sobre todas las pantallas involucradas para este atributo, el lector fácilmente puede conjeturar que el indicador elemental, luego de una re-evaluación, alcanzaría el 100% de satisfacción.

4.3. Métodos de M&E en GOCAME

Además de procesos, GOCAME dispone también de especificaciones de métodos de M&E bien establecidos. Generalmente, las especificaciones de proceso dicen qué se debe hacer, pero no cómo debe hacerse. Es decir, no especifican los métodos y técnicas (y herramientas que las soporten) que deben ser utilizados para llevar a cabo las actividades del proceso. Recordar que se denomina metodología al conjunto de métodos relacionados asignados a las distintas actividades, que permiten transformar entradas en salidas de un modo particular. Además, una misma descripción de actividad, puede ser realizada por uno o más métodos distintos.

La metodología utilizada por la estrategia GOCAME se denomina WebQEM [23], la cual permite llevar a cabo las principales actividades que intervienen en procesos de medición, evaluación, análisis y recomendación. WebQEM provee formas particulares (especificaciones) para representar a los requerimientos no-funcionales, a la medición, a la evaluación y a las recomendaciones. Por ejemplo, la metodología utiliza vistas y modelos de calidad centrados en características, sub-características y atributos para representar e instanciar requerimientos no funcionales, aunque también puede emplear la estructura de cuestionarios para representar e instanciar los requerimientos de calidad, como por ejemplo, sub-conceptos e ítems de cuestionario para evaluar percepciones de usuario en QinU, en particular, para Satisfacción.

Por otra parte, la descripción de la actividad de medición se realiza mediante especificaciones de métricas (que representa cómo transformar la entrada -atributo- en salida -medida-). La Tabla 2 muestra la especificación de una métrica. Por último, para la actividad de evaluación elemental se utilizan las especificaciones de indicador elemental (ver Tabla 3) mientras que para la actividad de evaluación global, diferentes métodos como el aditivo, LSP, etc., se pueden emplear para agregar y calcular los indicadores parciales y global.

5. Conclusiones y Trabajo Futuro

Como hemos indicado en la Introducción, las contribuciones del presente trabajo son, primero, la caracterización realizada para el concepto de Usabilidad a partir de características y relaciones entre Usabilidad y las vistas de calidad externa de sistema y calidad de sistema en uso. Es decir, hemos clarificado los conceptos de Usabilidad, Usabilidad en Uso, Experiencia de Usuario y Contexto, y discutimos además la falta de una clara caracterización de Usabilidad en los trabajos relacionados, lo cual quita sistematicidad y precisión a la hora de especificar requerimientos no funcionales para su ulterior evaluación.

Segundo, como caso práctico e ilustrativo, hemos diseñado los requerimientos no-funcionales para medir y evaluar la Usabilidad de la aplicación Facebook para dispositivos móviles (en particular, para smartphones), especificando algunas de sus sub-características y atributos desde la perspectiva de calidad de sistema o EQ. Además, hemos diseñado e implementado la medición (por medio de métricas) y la evaluación (por medio de indicadores) de Usabilidad, utilizando la estrategia GOCAME.

El presente estudio se realizó sobre Facebook como caso típico, dado que es una red social con un

marcado liderazgo a nivel regional e internacional [1]. Como esta aplicación pertenece a una entidad privada, no pudimos tener acceso a su diseño y código fuente, pero de haber contado con esa documentación, GOCAME puede dar soporte a planificar acciones de cambio. El plan de cambios puede estar basado en las recomendaciones mostradas en la Tabla 4, con el fin de implementarlas y mejorar las debilidades encontradas. A posteriori, podríamos realizar una re-evaluación de la nueva versión de la aplicación Facebook con los cambios realizados para verificar la ganancia en la mejora respecto a la Usabilidad de la versión previa. Una vez incrementado el nivel de aceptabilidad para Usabilidad, usando una estrategia como en [5], se podría evaluar el incremento de los niveles de aceptabilidad de Usabilidad en Uso y Satisfacción y, en definitiva, de UX. Además, se podría establecer un ciclo de mejora continuo a partir de evaluaciones de QinU/EQ/QinU.

Otro trabajo futuro que sí podemos realizar a corto plazo, es evaluar la nueva versión de la aplicación desarrollada por Facebook (versión 13.0 al 22/07/2014), para comparar las dos evaluaciones y analizar si los cambios realizados por la organización lograron mejorar, o no, la Usabilidad de la misma.

6. Referencias

[1] Information Geographies at the Oxford Internet Institute: "*Age of internet empires*". *http://geography.oii.ox.ac.uk/?page=age-of-internet-empires*, 2013. Recuperado en Julio, (2014).

[2] ISO/IEC 25010: "*Systems and softwareengineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models*", (2011).

[3] ISO/IEC 9126-1: "*Software Engineering - Product Quality - Part 1: Quality Model*", (2001).

[4] Olsina L., Lew P., Dieser A., and Rivera B.: "*Updating Quality Models for Evaluating New Generation Web Applications*". In Journal of Web Engineering, Special issue: Quality in new generation Web applications, Abrahão S., Cachero C., Cappiello C., Matera M. (Eds.), RintonPress, USA, 11 (3), pp. 209-246, (2012).

[5] Lew P., Olsina L., Becker P., and Zhang, L.: "*An Integrated Strategy to Understand and Manage Quality in Use for Web Applications*". Requirements Engineering Journal, Springer London, (17): 4, pp. 299-330, (2012).

[6] Apple iOS Human Interface Guidelines: *http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html*, recuperado en Julio, (2014).

[7] Google User Interface Guidelines: *http://developer.android.com/guide/practices/ui_guidelines/index.html*, recuperado en Julio, (2014).

[8] Nielsen J., and Budiu R.: "*Mobile Usability, New Riders*", Berkeley CA, (2012).

[9] Nayebi F., Desharnais J-M., and Abran A.: "*The state of the art of mobile application usability evaluation*". In 25th IEEE Canadian Conference on Electrical Computer Engineering, 2012, pp.1-4, (2012).

[10] Sohn T., Li K.A., Griswold W.G. and Holland J.: "*A diary study of mobile information needs*". In ACM, Conference CHI'08, Florence, Italy, pp. 433-442, (2008).

[11] Chittaro L., and Cin P. D.:"*Evaluating Interface Design Choices on WAP Phones: Navigation and Selection*". Personal and Ubiquitous Computing, 6, 237-244, (2002).

[12] Heo J., Ham D-H., Park S., Song C., and Chul W.: "*A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors*". Interacting with Computers, Elsevier.21 (4), pp. 263-275, (2009).

[13] Ham D-H., Heo J., Fossick P., Wong W., Park., Song C., and Bradley M.: "*Model-based Approaches to Quantifying the Usability of Mobile Phones*". LNCS 4551, Springer,Human-Computer Interaction. Interaction Platforms and Techniques,pp 288-297, (2007).

[14] ISO/IEC 15939: "*Software Engineering - Software Measurement Process*", (2002).

[15] Zhang D., Adipat B.: "*Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications*". International Journal of Human-Computer Interaction, 18(3), 293 – 308, (2005).

[16] Olsina L., Papa F., Molina H.: "*How to Measure and Evaluate Web Applications in a Consistent Way*". Chapter 13 in Web Engineering, Springer Book, Rossi, G., Pastor, O., Schwabe, D., Olsina, L., Eds.. pp. 385–420, (2008).

[17] Olsina L.; Santos L.; Lew P.: "*Evaluating Mobileapp Usability: A Holistic Quality Approach*", LNCS 8541, Springer, 14th Int'l Conference on Web Engineering (ICWE 2014), Toulouse, France, pp. 111–129, (2014).

[18] Hassenzahl M.: "*User Experience: towards an experiential perspective on product quality*". In: 20th Int'l Conference of the Assoc. Francophone d'IHM; Vol. 339, pp. 11-15, (2008).

[19] Lew P., and Olsina L.: "*Relating User Experience with MobileApp Quality Evaluation and Design*". LNCS 8295, Springer, Current Trends in Web Engineering, ICWE Int'l Workshops, Q.Z. Sheng and J. Kjeldskov (Eds.), Aalborg, Denmark, pp. 253–268, (2013).

[20] Olsina L., Martín M.: "*Ontology for Software Metrics and Indicators*", Journal of Web Engineering, Santiago de Chile, Rinton Press, US. 4, vol. 2, pp. 262-281, (2004).

[21] Becker P., Lew P., Olsina L.: "*Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy*", In: Advances in Software Engineering Journal, Academic Editor: Osamu Mizuno, Hindawi Publishing Corporation, Vol. 2012, 27 pg., DOI: 10.1155/2012/949746, (2012)

[22] Dujmovic J.: "*Continuous Preference Logic for System Evaluation*", IEEE Transactions on Fuzzy Systems, (15): 6, pp. 1082-1099, (2007).

[23] Olsina L., Rossi G.: "*Measuring Web Application Quality with WebQEM*.". IEEE Multimedia Magazine. (9): 4, pp. 20-29, (2002).

Herramientas de Análisis de Bordes para Reconocimiento y Clasificación de Objetos en Imágenes Digitales

Jorge Kamlofsky, María Lorena Bergamini

[1] CAETI - Universidad Abierta Interamericana

Av. Montes de Oca 725 – Buenos Aires – Argentina

{Jorge.Kamlofsky, Maria.Bergamini}@uai.edu.ar

Resumen

En este trabajo se presentan avances en las investigaciones orientadas al desarrollo de algoritmos para reconocimiento y clasificación de objetos en imágenes digitales. Basándose en una aproximación poligonal de curvas digitales, se construye un patrón que representa el cambio en el vector tangente y permite determinar características geométricas del objeto, como ser concavidad, convexidad, puntos angulosos, entre otros. Los puntos angulosos se toman como puntos de referencia para dividir la curva en partes significativas. La identificación de objetos se realiza por comparación de patrones, de acuerdo a una métrica apropiada; aplicada al patrón completo o a una o varias partes significativas. De esta manera se pueden detectar similaridades en siluetas de objetos en imágenes digitales, así como diferencias en ciertas partes significativas, permitiendo la clasificación de objetos del mismo tipo.

1. Introducción

El análisis de imágenes digitales para la identificación de objetos dentro de las mismas muestra un creciente auge, sobre todo por su aplicación a procesos automatizados de control. Ejemplos de estos procesos son control de calidad en la línea de fabricación, sistema de monitoreo de un predio para detección de intrusos, sistema automatizado de barreras en un estacionamiento privado, control de tránsito en determinada vía, etc. En estos ejemplos es una tarea central reconocer objetos y clasificarlos. Por ejemplo, en el control de calidad de una fábrica, el aporte que se espera del análisis de imágenes tomadas con una cámara en distintos puntos de la línea de fabricación consiste en determinar tempranamente fallas observables en su aspecto físico. Un producto que no respete la forma normalizada para la que fue diseñado debe descartarse.

En un estacionamiento privado que se gestiona automáticamente interesa por un lado tener información de la cantidad de espacios que están disponibles en cada momento, distinguir si el vehículo entrante es uno de los vehículos autorizados, controlar el tipo de vehículo que ingresa (y egresa), etc. Y todas esas tareas se pueden llevar a cabo a través del registro de cámaras dispuestas adecuadamente para tal fin.

El proceso de análisis de imágenes debe identificar los rasgos característicos del objeto que desee reconocer. Actualmente existe una gran cantidad de aplicaciones que usan visión artificial basada en topología digital para la eficiente identificación de objetos. Las estrategias usadas para la identificación se basan en análisis de bordes, o análisis de toda la región que comprende el objeto [10].

Un objeto digital en una imagen es un conjunto conexo de puntos en una grilla digital, y el borde del mismo puede verse como una o varias curvas cerradas simples.

El conjunto de puntos que constituyen el borde de una figura contiene mucha información acerca de la forma del mismo: convexidad, dimensiones, agujeros, puntos angulosos, bordes rectos, circularidad, etc.

El reconocimiento de formas puede hacerse por una comparación píxel a píxel entre dos imágenes, o utilizando ciertos atributos basados en características geométricas para medir la similitud entre dos formas. Claramente, la primera alternativa usa los recursos computacionales (memoria y procesador) en forma intensiva, mientras que la segunda opción aprovecha características distintivas de las formas, para identifi-

carlas con pocos datos, pero sin perder información importante del objeto analizado.

Una de las características geométricas más útiles para el estudio de formas es la curvatura. La definición de curvatura para curvas digitales ha dado lugar a diversos estimadores. Considerando a una curva digital como la digitalización de una curva suave continua, los métodos de estimación se basan en el cálculo aproximado de la derivada de la tangente, el módulo de la segunda derivada de una parametrización, o mediante la inversa del radio del círculo osculante [2]. Por su parte, otros autores proponen una estimación basada en optimización global [5] o en arcos maximales [8].

Varios estimadores se basan en medir segmentos digitales [9]. En general, la detección de segmentos de líneas rectas digitales es muy sensible a la presencia de ruido en la imagen. En [7] se trata el ruido en imágenes digitales mediante el concepto de segmentos borrosos. Este concepto se fundamenta en la noción de barra digital [6], que es una generalización de la definición de segmentos digitales. En [1] se introduce el concepto de línea digital de ancho α. Para detectar tales líneas es necesario conocer o calcular la cáscara convexa del conjunto de puntos, y medir su ancho vertical y horizontal.

En [4] se propone un sencillo algoritmo de aproximación de curvas digitales por medio de polígonos digitales. El método se basa en eliminar puntos internos de arcos de curvas que son aproximadamente rectos; más precisamente arcos que constituyen un segmento borroso. El conjunto de puntos en el arco tiene un ancho perpendicular menor que cierta tolerancia establecida por el implementador.

También en [4] se definió una representación de formas invariante frente a movimientos rígidos, a fin de ser aplicado para el reconocimiento confiable de objetos, en forma eficiente. El método se basa en aproximar inicialmente el borde de un objeto por un polígono y obtener luego una representación de la evolución de la curvatura a lo largo de todo el polígono. Dicha representación constituye un patrón de giro que caracteriza una forma. Basándose en el patrón de giro, se define una métrica de similitud entre formas y se comparan curvas digitales simples.

La representación de formas mediante el patrón de evolución de curvatura puede verse como la versión discreta de la evolución de la curvatura en figuras continuas planas.

Se requiere que el patrón de giro sea invariante frente a movimientos rígidos (traslación, rotación, simetrías) y también ante escalado uniforme. La invariancia frente a traslación, simetrías y escalado se cumple evidentemente. En [3] se propuso una primera solución para lograr invariancia frente a rotaciones, pero de eficacia limitada.

En este trabajo se propone en primer lugar una alternativa para la invariancia frente a rotaciones. Además, se describe el uso de tramos parciales del patrón para focalizar la comparación en ciertas áreas de las formas. Para ello se identifican puntos distintivos en el patrón, lo que permite partir el contorno de un objeto en partes significativas.

En la siguiente sección se describe el algoritmo de aproximación de curvas digitales por polígonos de vértices de coordenadas enteras y en la sección 3 se describe la construcción del patrón de giro normalizado, la identificación de puntos de referencia y el procedimiento para lograr que el patrón sea independiente del punto de inicio de la curva y, por lo tanto, invariante frente a rotaciones. La sección 4 describe cómo el patrón de giro permite identificar formas similares, definiendo una distancia. Se incluye una sección de resultados experimentales.

Finalmente, en la última sección se discuten los resultados obtenidos, y el alcance del algoritmo propuesto.

2. Poligonalización de curvas digitales planas

El método propuesto para el estudio de formas de figuras digitales comienza poligonalizando la curva de borde [4]. Este procedimiento simplifica la curva, haciendo que la cantidad de datos a manejar se reduzca notablemente, sin perder información relevante. Además, es capaz de ignorar los efectos del ruido en la imagen introducido por la captura o por el proceso de segmentación objeto-fondo, siempre que la profundidad del ruido sea menor que la tolerancia de aproximación.

El proceso de poligonalización elimina puntos de la curva que están aproximadamente alineados, es decir, que se asemejan a un segmento. Formalmente, un arco de curva conteniendo los puntos $\{p_s, p_{s+1}, p_{s+2}, \ldots, p_{s+r}\}$ se considera aproximadamente recto si la distancia de cada uno de los puntos al segmento continuo de extremos p_s y p_{s+r} es menor que cierta tolerancia preestablecida. Así, a partir de la curva digital $\{p_1, p_2, \ldots, p_k\}$ se obtiene un conjunto de vértices $\{v_1, v_2, \ldots, v_n\}$ del polígono.

En promedio, el polígono tiene entre el 5% y el 20% de la cantidad de puntos de la curva; dependiendo de la tolerancia y de la forma estudiada.

3. Patrón normalizado de giro

Para comparar polígonos y medir la similitud entre ellos, se define el patrón normalizado de giro. Para cada vértice v_i se define λ_i como la longitud normalizada de los lados del polígono desde el vértice v_1 al vértice v_{i+1}. Es decir,

$$\lambda_i = \sum_{j=1}^{i} \frac{long(v_j v_{j+1})}{L}$$

y $\lambda_0 = 0$; donde L es el perímetro del polígono.

Por otro lado, se define una estimación de la curvatura acumulada desde el punto inicial al vértice v_i como

$$\kappa_i = \sum_{j=1}^{i} \Delta\theta_j$$

donde $\Delta\theta_j$ es el ángulo orientado en el vértice v_j.

La orientación del ángulo en v_j se considera positiva si al recorrer los vértices en orden v_{i-1}, v_i,v_{i+1} se produce un giro en sentido antihorario; y se considera negativa si el giro es sentido horario.

Claramente, para el último vértice del polígono v_n, λ_n=1 y κ_n = -2π (recorriendo el borde de la figura en sentido horario).

La curva lineal por tramos en el plano λ–κ, determinada por los puntos $\beta_i = (\lambda_i, \kappa_i)$, con inicio en β_0 = (0, 0) es el *patrón normalizado de giro*. Como esta curva lineal por tramos no tiene segmentos verticales, también se puede pensar que define a κ como función de λ, obteniéndose la función $\kappa(\lambda)$ para todo λ en el intervalo [0,1].

El patrón así definido depende del orden en que se tomen los puntos del polígono, en particular, de cuál es el primer punto. Obviamente, al cambiar el punto de inicio del polígono, cambiará el patrón, pero no la información que contiene.

Si se tiene un patrón obtenido a partir del vértice v_1, y se quiere obtener el patrón del mismo polígono, pero a partir del vértice v_m, todos los puntos del patrón se desplazan de acuerdo a las relaciones:

Para $k = m, m+1, \ldots n$:

$$\beta_{k-m+1} = (\lambda_k - \lambda_{m-1}\ ,\ \kappa_k - \kappa_{m-1})$$

Para $k = 1, 2, \ldots, m-1$:

$$\beta_{k-m+n+1} = (\lambda_k - \lambda_{m-1} + 1,\ \kappa_k - \kappa_{m-1} - 2\pi)$$

con el punto de inicio β_0 = (0, 0).

El patrón de giro encierra toda la información de la silueta de la forma analizada. Las zonas crecientes del patrón representan las partes cóncavas de la curva, y las decrecientes las zonas convexas. De esta forma, los mínimos relativos del patrón representan puntos de inflexión. Estos puntos son puntos de referencia para segmentar la forma e identificar en el objeto analizado los tramos característicos. Esto es de gran utilidad para llevar a cabo clasificación de objetos de cierto tipo, que son en gran parte similares, y las disimilitudes se centran en una determinada zona. Tal es el caso de los distintos tipos de vehículos de cuatro ruedas, que se pueden clasificar en tricuerpo, monovolumen, furgoneta, etc. Las diferencias entre las vistas de perfil de los distintos tipos de automóviles se pueden centrar en la zona trasera, o en la zona delantera.

Por otra parte, los puntos de referencia (puntos mínimos del patrón) son útiles para determinar la orientación de la figura; o mejor dicho, comparar dos objetos independientemente de su orientación. La orientación con que se encuentre una figura en una imagen influye en el punto inicial de la curva de borde, y por lo tanto, en el punto inicial del patrón de giro. Haciendo la comparación independiente de la orientación, resulta independiente del punto inicial.

Para ello, se busca establecer una correspondencia entre los puntos de referencia de dos objetos.

Sean $X=\{x_1,x_2,\ldots,x_n\}$ e $Y=\{y_1, y_2, \ldots, y_m\}$ los conjuntos de puntos de referencia de dos objetos O_x y O_y. Estos son los valores de λ que son mínimos relativos en el patrón. Se establece una correspondencia entre los puntos de esos conjuntos, tomando la asignación que logre mejor correlación entre patrones, de acuerdo a una función de desviación.

Definimos la desviación de la correspondencia $x_j \sim y_k$ como

$$f(x_j,y_k) = \sum_{i=1}^{n} wx_{ijk} + \sum_{i=1}^{m} wy_{ikj}$$

donde $wx_{ijk} = \min_{h=1,\ldots,m} |\tilde{x}_{ij} - \tilde{y}_{hk}|$ y $wy_{ikj} = \min_{h=1,\ldots,n} |\tilde{x}_{hj} - \tilde{y}_{ik}|$, siendo $\tilde{x}_{ij} = x_i - x_j$ y $\tilde{y}_{hk} = y_h - y_k$; es decir, son las coordenadas centradas en (x_j,y_k).

Finalmente, se selecciona la correspondencia con menor desviación; y esos puntos se toman como inicio de los patrones correspondientes.

En la figura 1 se observan dos patrones de objetos a identificar. Corresponden a los perfiles de dos vehículos similares. Todos los puntos de referencia en cada patrón se muestran en gris, mientras que el par de puntos que ofrecen menor desviación se marcaron en negro.

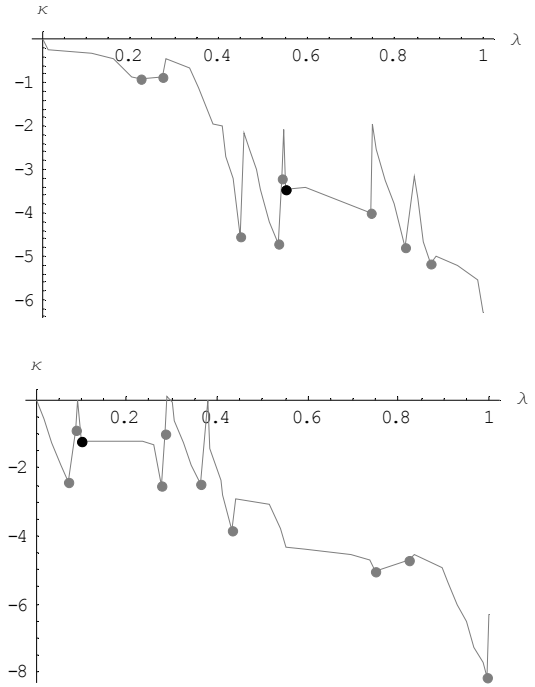


Figura 1: Dos patrones de objetos similares, y los puntos de referencia de menor desviación

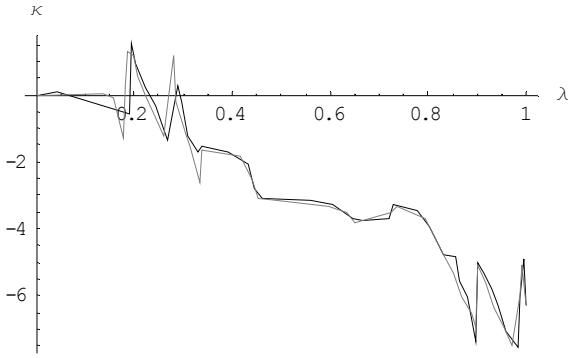


Figura 2: Los dos patrones de la figura 1, tomando como punto de inicio los puntos de referencia asociados

De esta forma, se puede tomar el inicio de los patrones en el punto obtenido al hacer la correlación entre los puntos de referencia. Los patrones con sus respectivos nuevos inicios resultan como en la figura 2. Se puede observar la analogía entre los patrones, hecho que es esperable, dado que corresponden a objetos similares.

4. Identificación de objetos usando el patrón de giro

Definimos la distancia entre dos patrones κ_1 y κ_2 como

$$D(\kappa_1, \kappa_2) = \int_0^1 | \kappa_1(\lambda) - \kappa_2(\lambda) | \, d\lambda$$

Dos patrones obtenidos a partir de distintas instancias de la misma figura, donde una se obtiene por traslación y/o escalado de la otra en la imagen digital, tienen distancia cero entre sí. Pero esto no ocurre en figuras relacionadas por otra transformación.

La rotación es una transformación rígida en un espacio euclídeo, pero no en el plano digital. Al rotar una curva digital se exige a los puntos caer en puntos grilla, introduciendo así error de discretización. Si se comparan instancias de una figura, afectadas por una rotación, el error de discretización afectará el patrón de giro, haciendo que la distancia así definida no sea cero.

Esta discrepancia entre distintas instancias del mismo objeto, ubicado en distintas posiciones, se hace más notable si se procede con imágenes reales, tomadas con scanners o cámaras, ya que la captura podría introducir ruido.

La clave para que el método de reconocimiento sea efectivo es que estas diferencias debido a la discretización son mínimas frente a diferencias debidas a cambios perceptibles en las formas.

Sea $\kappa(\lambda)$ el patrón de giro de una forma S que se quiere identificar. La identificación se realiza por comparación con patrones conocidos. Sean $\alpha_j(\lambda)$ patrones de formas candidatas a empatarse con la forma analizada. Entonces, se identifica la forma S con aquella cuyo patrón α_j diste de κ en menos de un valor umbral, es decir:

$$D(\kappa, \alpha_j) \leq \mu$$

La distancia definida también puede aplicarse a tramos parciales del patrón. La idea es analizar el tramo del borde comprendido entre dos puntos de referencia. Para hacer la comparación, se normaliza el tramo y se mide la distancia entre los patrones correspondientes.

Sean $\beta_j = (\lambda_j, \kappa_j)$ y $\beta_k = (\lambda_k, \kappa_k)$ los puntos de inflexión que delimitan la zona a estudiar en el borde de un objeto. Los puntos del patrón comprendidos entre β_j y β_k se normalizan con

$$Pi = \left(\frac{\lambda_i - \lambda_j}{\lambda_k - \lambda_j}, \kappa_i - \kappa_j \right)$$

Luego se mide la distancia entre el tramo considerado en las figuras que se están analizando.

Esto es aplicable al análisis y clasificación de objetos similares, que concentran las diferencias en una zona del borde, como es el caso de los distintos tipos de vehículos.

5. Aplicación experimental

Se implementó el método propuesto sobre un conjunto de imágenes de 19 vehículos de distintos tipos, y un conjunto de otras imágenes que contienen objetos digitales de diversas formas (formas abstractas, muebles, frutas, etc).

Se obtuvo que en general la distancia entre cada par de vehículos no supera el valor 1.2, siendo el promedio de estas las distancias 0.54. Para ejemplificar, vale mencionar que los vehículos cuyos patrones se muestran en la figura 2 distan 0.216.

Los pocos casos en que la distancia supera el valor de 1.2 son ocasiones en que el método de identificación de puntos de referencia establece la correspondencia entre puntos que no están en la misma posición en el borde del vehículo. El motivo principal de esto es el hecho de que ciertos objetos traen una cantidad considerable de ruido, lo que hace que aparezcan en los correspondientes patrones puntos de inflexión ficticios, que no representan puntos de inflexión en la forma.

La distancia entre un objeto-vehículo y cualquier otra forma, en nuestra experimentación, es siempre mayor que 2. La figura 3 muestra el patrón de un vehículo, junto con el patrón correspondiente a una forma abstracta, semejante a una flor. Puede observarse la disparidad de los patrones en toda su traza.

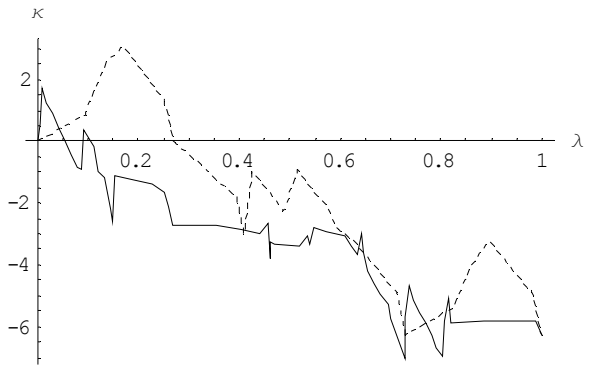


Figura 3. Patrón de un vehículo (línea llena) y de otro objeto (línea punteada).

Haciendo un análisis por sectores del borde del objeto digital, encontramos que la diferencia entre pares de patrones de vehículos, en ciertos casos, se concentra en una determinada zona. Por ejemplo, en la figura 4 se comparan patrones de un vehículo tipo furgoneta y un vehículo tricuerpo. La distancia global es 0.289. Se puede observar que la mayor diferencia se encuentra para valores de λ entre 0.2 y 0.3. Numéricamente, en ese intervalo se acumula el 30% de la distancia total. Esa zona de concentración de la diferencia corresponde a la parte trasera del vehículo, donde uno de los vehículos presenta baúl y el otro no.

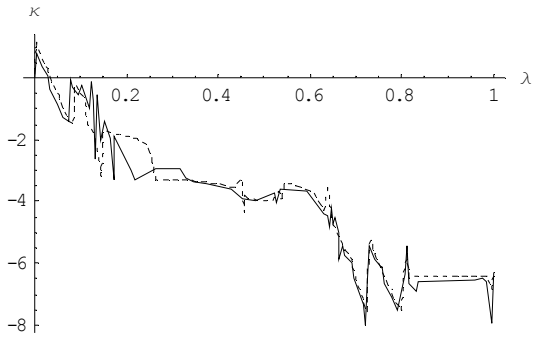


Figura 4. Patrones de dos vehículos: furgoneta (línea punteada) y tricuerpo (línea llena)

Otro ejemplo puede verse en la figura 5. Allí se muestran en conjunto los patrones de un vehículo bicuerpo y un vehículo tricuerpo. Se destaca la marcada diferencia en el tramo inicial de los patrones, nuevamente correspondientes a la parte trasera de los vehículos.

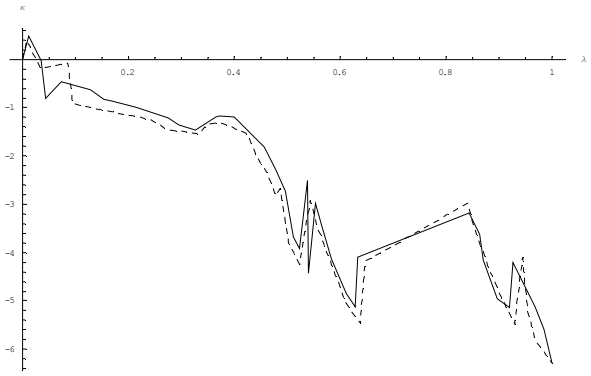


Figura 5. Patrones de dos vehículos: bicuerpo (línea punteada) y tricuerpo (línea llena)

Por otro lado, en la figura 6 se pueden observar los patrones de dos vehículos tipo furgoneta. La distancia entre ellos es 0.204, y esta distancia se distribuye regularmente a lo largo de todo el patrón.

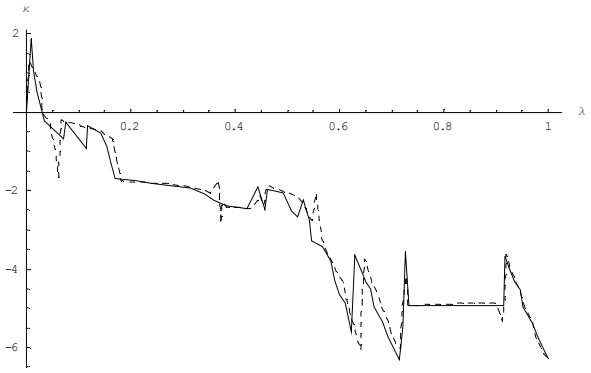


Figura 6. Patrones de dos vehículos tipo furgoneta

6. Discusión y conclusiones

El patrón de giro definido es una herramienta útil en la comparación de formas en una imagen digital. Al ser una aproximación de la curvatura acumulada de la curva borde del objeto, permite determinar puntos de inflexión y usarlos como referencia para dividir la curva en partes significativas, a fin de lograr la invariancia del patrón frente a rotaciones, es decir, que el patrón sea independiente del punto inicial de la curva digital.

Aunque en la mayoría de los casos de experimentación la localización de los puntos de referencia correspondientes fue exitosa, hubo casos en que esto falló, debido a la aparición de puntos de referencia ficticios, originados por ruido de alta profundidad. El proceso de poligonalización tiende a ignorar el efecto del ruido dependiendo de la tolerancia usada. A mayor tolerancia, menos efecto del ruido, pero la aproximación de aleja de la forma real. Existe entonces un compromiso entre la tolerancia usada y la incidencia del ruido en la eficacia del algoritmo.

El algoritmo propuesto es simple, de baja complejidad, y la tolerancia está determinada por el implemen-

tador, por lo que este procedimiento permite lograr la identificación rápida de objetos dentro de las imágenes, y la clasificación dentro de un conjunto de objetos de la misma especie. La posibilidad de descomponer el patrón en tramos significativos permite medir distancias parciales en determinadas zonas del borde de objetos.

Referencias

[1] Buzer L. *An Elementary Algorithm for Digital Line Recognition in the General Case*. Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science Volume 3429, 2005, pp 299-310

[2] Hermann S., Klette R.: *A comparative study on 2D curvature estimators*. Int. Conference on Computing: Theory and Applications (ICCTA'07), pp. 584-589 (2007).

[3] Kamlofsky, Jorge Alejandro y Bergamini, María Lorena. *Patrón de Evolución Discreta de Curvatura y Concavidad para Reconocimiento de Formas*. CONAIISI, (2013).

[4] Kamlofsky, Jorge Alejandro y Bergamini, María Lorena. *Patrón de Ángulo de Giro para Reconocimiento de Objetos en Imágenes Digitales* 14th Argentine Symposium on Technology – JAIIO 2013. ISSN 1850-2806 (2013)

[5] Kerautret B., Lachaud J.-O.: *Curvature Estimation along Noisy Digital Contours by Approximate Global Optimization*. Patt. Recogn. 42(10), 2265 – 2278 (2008)

[6] Kettle R., Rosenfeld A.: *Digital Straightness – a review.* Discrete Appl. Math. 139, 197-230 (2004).

[7] Nguyen T.P., Debled-Rennesson I.: *Curvature Estimation in Noisy Curves*. In: Kropatsch W., Kampel M., Hanbury A. (eds.) Computer Analysis of Images and Patterns, LNCS, vol. 4673, pp 474-481, Springer, Heidelberg (2007).

[8] Roussillon T., Lachaud J-O.: *Accurate Curvature Estimation Along Digital Contours With Maximal Digital Circular Arcs*. In: Aggarwal J., Barneva R., Brimkov V. Koroutchev K., Korutcheva E. (eds.) Combinatorial Image Analysis. LNCS, vol 6636, pp 43-55, Springer, Heidelberg (2011).

[9] Vieilleville , Lachaud *Revisiting digital straight segment recognition*, Discrete Geometry for Computer Imagery Lecture Notes in Computer Science Volume 4245, 2006, pp 355-366

[10] Zhang D., Lu G.: *Review of shape representation and description techniques*, Patt. Recogn., 37 (1),1-19 (2004).

The Role of Stories in Software Engineering

Manuel Imaz
BlendMind
Madrid Spain
imaz@mac.com

Abstract

We may consider two aspects of stories: learning to tell stories and telling stories to learn. We intend in this paper to use stories in order to enhance the practice of Software Engineering by learning some underlying facets of the mind. We witness the increasing use of stories by businessmen and politicians and so we may think it is something that belongs to less formal areas than Software Engineering's. Quite the opposite, there are many published studies that point to the use of stories in various areas of Software Engineering which can serve as excellent tools for software development. Yet another point is the frequently unknown cognitive foundations of stories as essential mind processes, which make them the suitable learning, support and work tools in various stages of software projects.

1. Introduction

The use of stories dates back at least to Greek antiquity, although we only have evidence of such use in Rhetoric from Roman writings. The *Rhetorica ad Herennium* was the most popular book on rhetoric during the Middle Ages and the Renaissance. An interesting use of stories was applied to memory and the book contains the first known description of the method of loci, a mnemonic technique. *Ad Herennium* also provides the first complete treatment of memoria (memorization of speeches). In fact, the method of loci or memory palace – because it is the visualization of a palace or house– is a method of memory enhancement, which uses visualization based on a basic story (the tour of the palace) to organize and recall information.

For those who may be hesitating on whether to address the issue of stories in Software Engineering we may answer that some people consider it as a social sciences area: the department of Informatics of the Ümea University, Sweden, belongs to the Faculty of Social Sciences. In their web, they tell that *A social sciences field, Informatics is strongly influenced by studies of technology and its use. Informatics is a field for people who wish to understand information technology, and for those who seek to learn how it is designed and used.*

According to Frances Yates, the clearest description of the process of recalling a speech or narration is that given by Quintilian:

The commonest, though not the only, type of mnemonic place system used was the architectural type...In order to form a series of places in memory, he says, a building is to be remembered, as spacious and varied a one as possible, the forecourt, the living room, bedrooms, and parlours, not omitting statues and other ornaments with which the rooms are decorated...as soon as the memory of the facts requires to be revived, all these places are visited in turn and the various deposits demanded of their custodians. ***We have to think of the ancient orator as moving in imagination through his memory building whilst he is making his speech, drawing from the memorised places the images he has placed on them****. The method ensures that the points are remembered in the right order, since the order is fixed by the sequence of places in the building.* [14] p. 3. Bolds included by me. MI

A memory building –a roman house– is represented in Figure 1.

More recently, we have seen some businessmen using stories in their speeches like Steve Jobs in his 2005 Stanford University commencement speech, based on three stories: the first one about *connecting the dots*, the



Figure 1. Going through a roman house in order to recall a narration.

second about *love and loss* and the third about *death*. Politicians as Barack Obama has been advised by his think tank to use stories in his speeches. During his campaign, Obama used stories, forgotting soporific speeches plenty of arguments based on rationality and batteries of dark statistics and emphasizing the emotional side of his messages.

However, as Goguen points out,

This culture [of using stories] *therefore calls for using methods from the social sciences, particularly during the requirements phase of large system development projects. A significant difficulty with this, however, is that few computing scientists know very much about the social sciences; indeed, many computing scientists have little sympathy with the social sciences, and prefer the relative certainties of hardware, software and dryware, to the ambiguities, conflicts and vagueness of wetware* [4] p. 4.

Due to this shortage of using methods from the social sciences and other fields like cognitive neuroscience – that is, the scientific study of the neural mechanisms underlying cognition– is that we consider useful to present here some evidences based on cognitive sciences. The aim of this article is to offer some evidences of the importance of stories based on cognitive neuroscience research (in 1.1), on cognitive semantics (in 2) and on the use of scenarios, user stories and use cases with decisive results in requirements engineering (in 1.2). As far as our information reaches, we do not know any literature about using stories based on the latest research in cognitive science as is the use of fMRI (functional magnetic resonance imaging) or on cognitive semantics. At the same time, we show how the use of stories in other areas of software engineering is as important like in requirements capture, so we recommend a widespread adoption of the use of stories in software development.

1.1. Cognitive foundations of stories

We know that meaning is a social phenomenon. In a social context, such as a development team, we have a knowledge sharing common to any group, community or society. But the important point is that such a team's communication be as efficient as possible.

When using stories we are producing a narrative meaning, indicating that something is a part of a whole and usually that something is the cause of something else. It is usually combined with human actions or events that affect human beings, The meaning of each event is produced by the part it plays in the whole story.

In relation with storytelling it is also employed the term *narrative intelligence*. When using a standard formal way of communication this seems to imply a one-way relationship of the type "I tell you that and you have to listen". Instead, when using a storytelling way we deploy a two-way, interactive relationship.

Some recent research, such as those by Hasson et al. [16], show a brain-to-brain coupling when a speaker tells a story to a listener. *Brain-to-brain coupling is analogous to a wireless communication system in which two brains are coupled via the transmission of a physical signal (light, sound, pressure or chemical compound) through the shared physical environment…On average, the listener's brain responses mirrored the speaker's brain responses with some temporal delays…However, the analysis also identified a subset of brain regions in which the responses in the listener's brain in fact preceded the responses in the speaker´s brain.* [16], p. 118.

Another experiment [17] shows how some brain's areas are activated when human beings are listening short stories. In the experiment, participants read brief stories ending in either a literal, metaphoric, or ironic statement. The authors used fMRI in an event related design to measure brain activation while the participants were reading the final, critical sentence. In Figure 2 we can see the images of the fMRI:

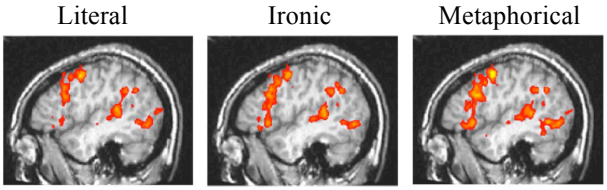


Figure 2. Three fMRI images of brain activity when reading literal, ironic and metaphorical stories.

1.2. Evidence-based software engineering

There is a current trend in software engineering that considers a method should be based on formal results or on evidence-based guidelines, in this case those of software engineering (EBSE – Evidence-based Software Engineering). When the guidelines are based on formal developments, the methods are nearly immediately adopted, as it was the case with structured programming. It is important to say *nearly immediately* because although the arguments were clear, programmers were reluctant to adopt the new rules as they were only a new way of *doing the same thing*. There were no reasons to continue programming in an old-fashioned style when it

was shown that all structures of languages were reduced to just three: sequence, selection and iteration, without the need to use the unnecessary and counterproductive go-to instruction.

A different case was the introduction of the object-oriented paradigm. Even if we had good reason to trust the goodness of the new structures of the OO style, there were no evidence-based reasons offered by developers groups that indicate the higher virtues of the new programming languages.

Many of the arguments were of philosophic or semantic styles, such as a match between the objects constructs and objects of the real life. And other arguments, more technological ones, such as objects being components and then allowing a more natural development, we have seen they were flawed arguments. These are reasons we have learned as a long historical lesson and not a conclusion derived from an experiment of a team, for example, of 25 people working during 3 months.

It is evident that some aspects of EBSE are essentially low risk and should be adopted as soon as possible such as the development and adoption of guidelines for systematic reviews or those related to software testing that appear to be promising candidates for EBSE.

However, there are also scientific problems that may be more difficult to address. The problem of evaluating technologies that rely on human skills means that our experiments will always be vulnerable to subject and experiment bias. There are approaches that can be used to reduce the scale of the problem. More difficult to resolve is the problem of the complexity of the software lifecycle [15]. Nonetheless, the use of stories has been showing its usefulness in various areas of software engineering over the last years.

Regarding the use of scenarios, user stories and use cases, there is abundant literature showing how these constructs have been used in requirements engineering with very good results. For example, use cases have proven to be one of the most widely adopted practices in software development. The equivalent of use cases in agile methods such as Scrum, are user stories, with a similar structure to use cases but less content. This is historical evidence that the stories, in its various forms, are a powerful tool for the development of requirements. In this paper we present other uses of stories: in strategic planning, in UX (User Experience) and Documentation.

From the extensive literature we can cite the following books [1], [18], and [19].

2. The Literary Mind

A human being is nothing but a story with a skin around it

Fred Allen

In the mid-nineties, Turner [13] and Fauconnier have developed some ground theories about stories and mental spaces. In particular, Turner has illustrated how an aspect of our mind –the literary side– provides us a knowledge that sometimes lacks a more scientific discipline [5].

The literary mind is not a separate kind of mind. It is our mind. The literary mind is the fundamental mind. Although cognitive science is associated with mechanical technologies like robots and computer instruments that seem unliterary, the central issues for cognitive science are in fact the issues of the literary mind.

***Story** is a basic principle of mind. Most of our experience, our knowledge, and our thinking is organized as stories. The mental scope of story is magnified by **projection** –one story helps us make sense of another. The projection of one story onto another is **parable**, a basic cognitive principle that shows up everywhere, from simple actions like telling time to complex literary creations like Proust`s **A la recherche du temps perdu**.* [13] p. V (bolds are italics in the original. MM).

Turner`s intention is to show how wrong the common view is and to replace it with a view of the mind that is more scientific, more accurate, more inclusive, and more interesting, a view that no longer misrepresents everyday thought and action as divorced from the literary mind. We constantly witness stories that involve actors engaged in bodily actions, that is, small stories of events in space: the wind blows clouds through the sky, a child throws a ball, a girl pours water from a glass.

When we watch someone sitting down into a chair, we see what physics cannot recognize: an animate agent performing an intentional act involving basic human-scale categories of events like sitting and objects like chair. So what we, as human, additionally see is the intention of sitting of the animate agent.

Another paper, published by Shaw et al. [10] states that stories are central to human intelligence and memory and explains how cognitive scientists describe the way we gradually acquire the ability to formulate plans through the stories we hear in childhood. It`s from stories that a child learns to ``imagine a course of action, imagine its effects on others, and decide whether or not to do it'' (Scientific American, October 1994).

Needless to say that stories also play an important role in learning. Researchers studying how high school students learn found that the story-based style of Time and Newsweek weekly publications was the best way to learn and remember. Another important fact has been pointed out: when some researchers translated textbooks into this format –Time and Newsweek story-based style– they found that students recalled up to three times more than with classical styles.

The recognition of how universal is the use of stories in our everyday life may be a difficult point. To such an extent that we are not aware of them:

We must adopt a scientific perspective to see why something we already know how to do without effort or conscious attention can pose an extremely difficult and important scientific puzzle. The capacity for recognizing and executing small spatial stories is –like the capacity to speak, to see color, or to distinguish sounds– an obvious and deceptively easy capacity. In fact, it presents the chief puzzle of cognitive science. How can five billion different human beings all recognize and execute small spatial stories? [13] p. 13}

2.1. Story-based Style of Writing

One of the few basic rules established by this style for writing and structuring any news story is the *inverted pyramid*; it means that the heaviest or most important information should be at the top –the beginning– of our story and the least important information should go at the bottom. And as we move from top to bottom, the information presented should gradually become less important.

The first paragraph – or *lede* (that`s how journalists spell it)– of any news story is also the most important. The lede must accomplish several things:

- a. give readers the main points of the story.
- b. get readers interested in reading the story.
- c. accomplish both “a” and “b” in as few words as possible.

Typically editors want ledes to be no longer than 35-40 words. Why so short? Readers want their news delivered quickly. A short lede does just that.

We observe that the usual structure of a report or presentation is based on lists, outlines and bullets, something that makes it difficult to understand and recall the text. Referring to these difficulties, Shaw explains that:

At one level it`s odd, but at another level it isn`t at all, since virtually all businesspeople plan using lists, outlines, and bullets. Cognitive psychologists have established that lists, in contrast, are remarkably hard to remember because of what is referred to as the recency and primacy effects: people mainly remember the first and the last items on a list but not the rest of it, and – more dangerous yet – their memory is guided by their interests. They remember what they like or find interesting; they do not recall the whole.

A good story (and a good strategic plan) defines relationships, a sequence of events, cause and effect, and a priority among items – and those elements are likely to be remembered as a complex whole. That likelihood, supported by a substantial amount of cognitive science, argues strongly for strategic planning through storytelling. [10] p. 4

In terms of literary analysis it is said that a narration has a content but, equally important, it has also a style. Analyzing the question from the point of view of Langacker in his Cognitive Grammar [6], it can be stated that:

In viewing a scene, what we actually see depends on how closely we examine it, what we choose to look at, which elements we pay most attention to, and where we view it from. The corresponding labels I will use, for broad classes of construal phenomena, are specificity, focusing, prominence, and perspective. They apply to concepts in any domain. [6] p. 55 (bolds in the original)

3. Stories in Strategic Planning

But, besides the previous concepts defined by Langacker, we need an additional one, that of **conceptual base**. ``As the basis for its meaning, an expression selects a certain body of conceptual content. Let us call this its conceptual base. Construed broadly, an expression's conceptual base is identified as its maximal scope in all domains".

In Figure 3, we have a conceptual base made up of several elements: a phone's digital device --or piece of software-- the market of such type of software, other companies with competing software products and so on. The point of view of the story is ours --our company-- and the focus of the scene may by continuously changing from an element to another of the conceptual base. For example, when describing the product from another company we can return to our product.

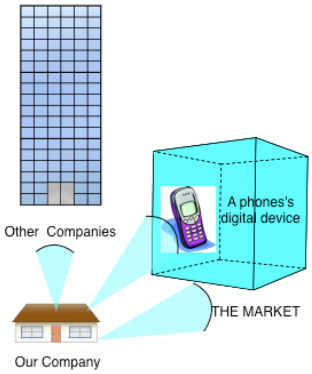


Figure 3. The conceptual content needed to build a strategic story.

This is the basis to build a strategic planning story, but the arguments are also applicable to any other area of Software Engineering were it is used a style based on lists, outlines, and bullets, that is, the PowerPoint style used in most scenarios: presentations, plans, learning and so on.

Some works have been published about the *bullet-point rhetoric* [3, 12] criticizing it and showing its deficiencies and dangers. For example:

This pedagogical void in the area of composition, this negligence with regard to discourse, has a simple explanation: the PowerPoint format favors a mode of communication in which words seem emptied of any substance. Merely consulting several presentations in the area of economics and finance, the kind organized for shareholders' meetings or for the presentation of the annual earnings report, is enough to take the measure of this linguistic impoverishment. [3] p. 50

In his paper, Shaw [10] points out that, even if in 3M --a multinational company that commercializes industrial, health care, electronics and consumer products– stories were a habit of mind they had so far –until 1998– yet they discarded storytelling when they did their strategic planning. Storytelling is the way that makes them see themselves and their business operations in complex, multidimensional forms and to discover opportunities for strategic change. We see that stories offer the opportunity –unlike the bullet-point rhetoric– of analyzing the business operations in a complex and multidimensional form.

Let`s see this difference with an example of some points extracted from a strategic plan for the introduction of a new software product in a software company. These might be the major objectives from the strategic plan:

- Increase market share by 10\%.
- Increase profits by 20\%.
- Increase new-product introductions to five a year.

This part of the strategic plan makes the plan to remain in an implicit form, preventing to see that it is a complex question and leaving an unexplained vision of the organization, the market, and the customer. It is not possible to deduce that vision from the bullet list. The list doesn't allow us to see which are the relationships between the objectives. In fact, many radically different strategies could be represented by these three simple points.

We may be guessing whether the improved marketing increases the market share, which results in increase profits, thus providing funds for increased new-product development

Market-Share➔ Profits ➔New-Product Development

Or maybe new-product development will result in both increased profits and market share at once:

↗ Market Share
New-Product Development
↘ Profits

Alternatively, perhaps windfall profits will let us just buy market share by stepping up advertising and new-product development:

Profits➔New-Product Development➔ Market Share

Writing the strategic planning as a story implies analyzing the industry's economics, its key success factors, and the forces that drive change, but also it must be introduced the *dramatic conflict*, for example: ``what challenges does the company face in this situation?" ``What critical issues stand as obstacles to success?"

As Shaw finally points out, when readers have read the complete narrative,

they will know how 3M intends to increase its market share from 40\% to 50\%. They will know which product

and process developments should, when combined, launch a new generation of fleet graphics. And they will be able to imagine that, given those new products and processes, as well as new training opportunities, sales representatives might well improve their performance by 50\%. [10] p. 8

4. Stories in Requirements Elicitation

The stories in requirements elicitation –the practice of collecting the requirements of a system from users, customers and other stakeholders– are stated from a narrator point of view –the developer– that observes each user who intends to interact with the system in order to get some results that are valuable for the users and customers.

Maybe the best known use of stories in software engineering are the user stories used in Agile and Lean development methods. Physics –unlike everyday stories– offers a representation of the world that leaves out agency, motive, intentionality, and a range of structure that is part of the conceptual equipment of everyone, including physicists. This way of describing reality could be called objectivist or neutral. But the essence of stories –in the sense we are employing here– such as described by Turner, is that they are tied to the human scale. For example:

sitting and chair are elements of story but not elements of physics. The fundamental units of physics exist at levels that are foreign to us –subatomic quarks, metrics of space-time, integrations from zero to infinity. Where physics offers an impenetrable but accurate physical description in the form of a wave equation, story offers Einstein sitting in a chair. [13] p. 14.

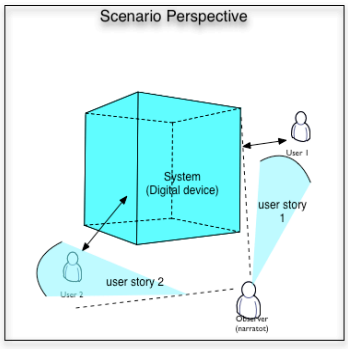


Figure 4. In user stories the attention is focused on the user and his interactions.

In requirements elicitation, the same idea of stories has been summarized in the Agile and Lean movement, to the point that Leffingwell [7] p. 56) points out that ``the team's backlog (unprocessed items) consists of all the work items the team has identified. In the meta-model, we generically call these work items stories". The author, before using the concept of user stories defines generically *stories* as a work item. A developer is observing some user stories in Figure 4.

The basic ideas of stories are resumed in the concept of *user stories*. A user story is a particular way of describing a functionality that will be valuable to either a user, developer or purchaser of a system or software. User stories are composed, in agile methods, of three aspects [1] p. 4:

- **written description** of the story used for planning and as a reminder
- **conversations** about the story that serve to flesh out the details of the story
- **tests** that convey and document details and that can be used to determine when a story is complete

These three aspects are also called Card, Conversation, and Confirmation. The Card may be the most visible manifestation of a user story, but it is not the most important.

It represents customer requirements rather than document them. This is the suitable way to think about user stories: while the card may contain the text of the story, the details are worked out in the Conversation and recorded in the Confirmation [1].

Agile and Lean methods like Scrum and Kanban use successfully user stories, as these are atomic units --when necessary, large stories are split into two or more stories of smaller size-- that will be developed in one iteration. This aspect makes them suitable for methods of development that need to be able to offer complete functionalities to users on each incremental iteration.

Some templates have been proposed to create user stories. According to a template each story can be written in the following format:

I as a (*role*) want (*function*) so that (*business value*)

We see, in the template, what Turner recognizes in any story: an animate agent –somebody in a role– performing an intentional act –a function– involving basic human-scale categories of events: the function results, which have a business value.

4.1. Stories and Personas

In *The Inmates Are Running the Asylum*, Alan Cooper states, `*To create a product that must satisfy a broad audience of users...you will have far greater success by designing for one single persona*' [2] p. 124. The idea of building a product with a single user, or a small selection of users, in mind seems to completely contradict the mind-set of our industry. At face value, it seems to suggest that if you limit the features and functions of the product you design to those that will satisfy just a few very specific people you will somehow build a successful product.

Personas are imaginary people who can help you build products that real people actually like to use. Personas are detailed descriptions of imaginary people with a photography, name, age, job, and so on. We believe that when you use data to create personas, and use personas in a thoughtful way during the product development process, you will increase product's usability, utility and general appeal.

In fact, we should add that personas help us understand users, not as part of a group or a quantifiable subsets within a given population, but as individuals with a history, goals, interests and a relationship to the product. They are *hypothetical archetypes* of actual users, such as shown in Figure 5.

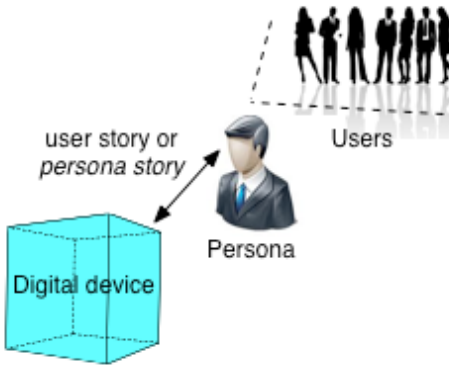


Figure 5. A *persona* represents the profile of a group of users.

5. Stories as Scenarios

We can call a scenario the description of a context or use situation. In general, we can say that scenarios describe people in their activities or work situations. In fact, scenarios are just stories. We can ask when scenarios may be used in place of user stories. And the answer may be that a scenario is a more general description –as a script (the scenario) describing an environment and including characters, each one with a set of stories appropriate when we are not sure about the possible uses of a new device.

But a scenario may be also useful in order to conceptualize a new device, as we need something concrete, even if we use it as a way of conceptualizing a different product. It is the case of the desktop interface, in which a real object is metaphorically translated to produce a digital device [5].

Some authors distinguish between conceptual and concrete scenarios and say the former scenarios are produced by abstracting from user stories, and that concrete scenarios are the result of design constraints on conceptual scenarios to arrive at concrete scenarios. But in the scenario below –*What shall we do now?*– we can observe that a conceptual scenario may be described before getting the user stories derived from it and have much wet information –in the sense established by Goguen [4] in his paper.

Goguen states that "a still closer examination of the major source of difficulty in large software projects which is their requirements shows that factors from the social world often lie at the root. Advocates of what I shall call the "wet culture" say that this must be taken seriously, because social, political and cultural factors ultimately determine the success of systems" [4] p. 4.

The Edinburgh Napier University was called, together



Figure 6. A scenario as a description of a context where a device will be used.

with other institutions, to undertake the design of a new device, the Home Information Center (HIC), a device intended to help perform some home tasks such as facilitating communication between family members, send orders to buy food at the supermarket and other shops, display lists of spectacles, buy cinema or theater tickets and so on. One of the scenarios described to design such a device goes as this:

Scenario: What shall we do now?

Jan and Pat are a couple in their mid thirties. Pat is a university lecturer in Cultural Studies and Jan is an accounts manager at Standard Life insurance. They live in the Stockbridge area of Edinburgh, Scotland in a two-bedroom flat overlooking the river. It is 12.00 noon on August 15th. Jan and Pat are sitting in their large, airy kitchen/dining room. The remains of pizza and mixed salad mingle with a pile of newspapers on the kitchen table...They decide that they would like to go to see one of the events that are happening as part of the Edinburgh Arts festival. Jan activates the HIC and chooses "Edinburgh Festival." The HIC connects to the different content providers who are registered as providing content about the festival [5] p. 158.

The advantage of scenarios, compared with user stories, is that scenarios –as a wet narration– offer the possibility of discovering new user stories. For example, someone could be guessing why the scenario describes the remains of pizza and bottles of beer, but rapidly we detect that Jan and Pat are users of food shipping shops and that such users stories should have to be included in the list of backlog items. See Figure 6.

The story of a real project for an important Spanish airlines company is another good example of scenario showing troubles. The system developed was intended for the cargo terminal but nobody knew the work conditions of users the system was implemented to. The implementation was based on traditional PCs with keyboards and mice. The environment of users was an industrial one, with lift trucks and other engines driven by people using gloves. The product developed was unfit for use, but a good scenario showing the real environment could have indicated how unsuitable was a classical device such as a classical PC and the need for new type of terminals in that context, like PDAs, for example.

6. Stories for UX (User Experience)

Although some may consider that UX, strictly, is not part of the software engineering discipline we include it in this article because both disciplines are closely related.

User experience (UX) involves a person's behaviors, attitudes, and emotions about using a particular product, system or service. User experience includes the practical, experiential, affective, meaningful and valuable aspects of human-computer interaction and product ownership.

We observe that the definition include such aspects as experiential and affective ones. That is why Don Norman, in his book *Emotional Design: Why We Love (or Hate) Everyday Things* has included the role emotion plays in consumer when he decides to purchase something. Human reaction to design exists on three levels: visceral (appearance), behavioral (how the item performs) and reflective. The reflective dimension is what the product evokes in the user in terms of self-image or individual satisfaction. The usual expression of *look and feel* also gives us the idea that we have an affective relationships with everyday objects, including digital devices.

That's why the use of stories in UX is entirely appropriate at all stages of the development, but it is clear that the stories used must be focused on describing or communicating some aspects of user experience. Covered stories are scenarios, user stories, stories for personas, storyboards, some (narrative) use cases and many other story forms that are part of different user experience methodologies [9].

An interesting question the authors point out is the triangular character of stories (see Figure 7), in the sense that the story becomes part of a dialogue between the storyteller and the audience, even if the audience is silent.

Another question is the difference they establish between a story and a narrative, even if in this article we make no difference between them.



Figure 7. The Story Triangle shows the connection between the storyteller and the audience.

You might have a description of a scene or a situation or environment. Sometimes events may be implied through context...Stories that are just narrative events –a series of statements of what ``happened''– aren't very interesting. They can be useful as a way to describe the details of an interaction. Use cases and flow charts are like that. [9] p. 28.

This interesting differentiation refers to the categories defined by Goguen: wet information furnished by stories, and dry information, when a *story* is impoverished –dried– and becomes a *use case*.

At a basic level, storytelling and user experience have common elements — like planning, research, and content creation — that can be utilized for effectively developing an experience. Storytelling offers a way for the team to really understand what they are building and the audience that they are creating it for. Stories allow for the most complex of ideas to be effectively conveyed to a variety of people. This designed product/experience can then offer meaning and emotion for its users.

7. Stories in Documentation

The creation of documentation (handbooks, manuals, presentations, etc.) involves a similar frame than user experience: planning, research, design and content creation. If we take into account recent proposals about visual design and the creation of books using touch sensitive images and animation, we can say that the differences between user experience and visual design for documentation is almost negligible. That is why a whole methodology is necessary for building a visual design.

There are many books about the field that propose such methods. For example, Sykes et al. [11] defines visual story process in which they describe four areas in the form of a matrix:

- a **content** area
- an **audience** area
- a **story** area
- a **tell** area

The *content* area is related to questions such as: Why, What, How and What if. Many presentations have too much content that is not relevant to the decision or change being proposed. If you want your sales team to search for new customers, your content must lead the audience to understand why you want them to act, and what they must do.

The *audience* area is related to questions such as: Who and learning and decision styles. You need to understand the people you're telling your story to. What do they need to know? How can you motivate them to respond in the way you intend? Whether you want your design team to develop a new logo, or your CEO to approve a new project, you need to consider the different people in the audience.

The *story* area is related to questions such as: Structure, character, sense of urgency and delivery plan. When you are clear about the content and your audience, you can focus on the story structure. We draw from centuries of practical experience of how to tell a compelling and interesting story. Using the format of a story, rather than simply presenting information, makes it easy for your audience to identify with your goals, remember your ideas, and agree with your suggestions.

The *tell* area is related to questions such as Design and test. Now you create the words and visuals to focus on the telling of the story. Work out how the story will be conveyed in different formats and test that it has the intended impact. Careful attention to the different ways a visual story can be told can make the difference between a clear story and a muddled message.

Finally, we are to focus on story area in order to see how the visual story map guides us through the process analyzing its structure, the character, the sense of urgency and the delivery plan:

Structure: Stories have a beginning, middle, and an end. They take the audience on a journey, and they use idioms, metaphors, rhetoric, and archetypal plotlines to help the audience connect with your ideas. Picking the right format and structure of a story helps to make it memorable and relevant.

Character: Your audience has to care in order to become truly engaged. Characters are critically important for humanizing your message and allowing members of your audience to see themselves, and their success, as being connected with the struggle of the characters in the story. The characters in your story must be meaningful to the audience, but that doesn't mean they all need to be people. Characters may be people, inanimate objects, and concepts — in short, anything that must change

Sense of urgency: There are many types of stories, but they all have one thing in common: the characters are placed in a situation where they must react. The sense of urgency in your story will help you to focus on the story elements that drive your characters, and your audience, toward the conclusion you want themto reach. Is there a

conflict to resolve, a puzzle to solve, or an obstacle to overcome? We build on lessons learned from top salespeople to capture urgency, to make your story compelling, and cause your audience to choose to act now.

Delivery plan: We are accustomed to stories being told in many different ways. Sometimes a story belongs at a team meeting, shared with peers, sometimes in a boardroom, and other times in a company cafeteria. Even business stories could be presented as movies or told around a campfire at an offsite retreat.

8. Conclusions

This paper aims to promote a larger use of stories in Software Engineering as we recognize the important role they play in almost all areas of the field. The need for such a promotion is based on the fact that software engineers may be reluctant to use methods they not consider formal enough to be employed in their projects, in particular methods from social sciences. We have mentioned, as an outstanding example, that the Informatics Department of the Ümea University belongs to the Faculty of Social Sciences, offering us a surprising point of view about our discipline. In opposition to the usual idea, they define Informatics as a social sciences field strongly influenced by studies of technology and its use.

In order to emphasize the importance of stories we present in this paper some recent cognitive sciences foundations by scholars of the discipline, like Mark Turner, Gilles Fauconnier and Rony Langacker. They give us enough arguments about the importance of stories in our everyday mental processes in either normal duties or scientific disciplines. This is why we must prevent the usual instinctive rejection to use methods based on social sciences, like stories, that risk to be considered by many as matters of rhetoric or literature in general.

The paper also has shown that stories are not reduced to user stories in agile methods but they also have an important role as scenarios –where the description of a general environment furnish opportunities to detect new user stories– as well as the user experience area or to develop documentation and presentations in general.

9. Acknowledgements

The author is grateful to Mauricio Milchberg for his revision of the manuscript and valuable comments.

10. References

[1] Cohn, M.: *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 2004.

[2] Cooper, A.: *The Inmates Are Running the Asylum*. New York: Macmillan, 1999.

[3] Frommer, F.: *How PowerPoint makes you stupid: The Faulty Causality, Sloppy Logic, Decontextualized Data, and Seductive Showmanship That Have Taken Over Our Thinking*. The New Press. New York. 2012.

[4] Goguen, J.: *The Dry and the Wet*. SRI International, MenloPark CA, USA. www-cse.ucsd.edu, 1992.

[5] Imaz, M., Benyon, D.: *Designing with Blends*. MIT Press, 2007.

[6] Langacker, R.: *Cognitive Grammar: A Basic Introduction*. Oxforf University Press, 2008.

[7] Leffingwell, D.: *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* Addison-Wesley Professional, 2011.

[8] By Pruit, J., and Adlin, T.: *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*. Elsevier Inc., 2006.

[9] Quesenbery, W., and Kevin, B.: *Storytelling for User Experience: Crafting Stories for Better Design*. Rosenfeld Media, Brooklyn. New York, 2010.

[10] Shaw, G., Brown, R., and Bromiley, P.: "Strategic Stories: How 3M Is Rewriting Business Planning". *Harvard Business Review*. May-June 1998.

[11] Sykes, M,. Malik, A., and West M.: *Stories That Move Mountains: Storytelling and Visual Design for Persuasive Presentations*. John Wiley & Sons Ltd., 2013.

[12] Tufte, E.: *The Cognitive Style of PowerPoint*. Graphics Press. http://www. edwardtufte.com/ ,2004.

[13] Turner, M.: *The Literary Mind*. Oxford University Press, Inc., 1996.

[14] Yates, F.: *The The Art of Memory*. ARK PAPERBACKS. An imprint of Routledge & Kegan Paul, 1984.

[15] Kitchenham, B., Dybå, T. and Jørgensen, M.: "Evidence-based Software Engineering", *Proceedings of the 26th International Conference on Software Engineering* (ICSE'04), 2004.

[16] Hasson, U. et al..: "Brain-to-brain coupling: a mechanism for creating and sharing a social world", *Trends in Cognitive Sciences, February 2012, Vol. 16, No. 2.* Elsevier, 2012.

[17] Eviatar, Z. Just, M.A.: "Brain correlates of discourse processing: An fMRI investigation of irony and conventional metaphor comprehension", *Neuropsychologia* 44 2348–2359, 2006.

[18] Carroll, J., *Scenario-Based Design, Envisioning Work and Technology in System Development*. John Wiley. 1995.

[19] Alexander, I. and Maiden, N. (Eds.) *Scenarios, Stories, Use Cases Through the Systems Development Life Cycle*. John Wiley & Sons, Ltd. 2004.

Generador de Código basado en GRAPHML

Cherencio, Guillermo Ruben
Universidad Tecnológica Nacional, Facultad Regional Delta

Abstract
GRAPHML es un formato de archivo para la representación de gráficos[1], permite la representación de gráficos dirigidos, no dirigidos, jerárquicos, referencias a fuentes externas, etc.. Entre las representaciones posibles, se encuentran los Diagramas de Flujo (flowcharts) como una forma de gráfico dirigido, que permite representar un algoritmo[2], el cual puede ser traducido en un programa de computación, expresado en un lenguaje de computación determinado. Por lo tanto, a partir de un archivo en formato graphml, que representa un diagrama de flujo, es posible obtener un programa de computación expresado en un lenguaje computacional determinado. El objetivo de este trabajo es el desarrollo de un software base que permita la inclusión de módulos de software, para que éstos actúen como generadores de código, a partir de la interpretación de diagramas de flujo expresados en graphml, para generar programas de computación en distintos lenguajes de programación. El software podría tener aplicación como una herramienta de apoyo en el proceso de aprendizaje de la programación de computadores. Todo el software se distribuye bajo licencia LGPL.

Palabras Clave
Flowchart Diagrama Flujo Programa Generador Código Java

Introducción

GraphML es un formato de archivo basado en XML[3] para la representación de gráficos. Un gráfico esta formado por nodos y líneas que los conectan; dentro de la sintaxis de GraphML[1] se distinguen 3 elementos principales: *graph, node, edge* que representan gráfico, nodo y línea respectivamente. Un diagrama de flujo es un gráfico dirigido en donde cada elemento (*node*) esta conectado por una linea (*edge*) la cual tiene un nodo origen y un nodo destino.

A partir de la consolidación de GraphML, han aparecido en el mercado una serie de editores que soportan este formato para la edición de distintos tipos de gráficos que pueden ser grabados y leídos en este formato. Uno de ellos es yEd Graph Editor de yWorks®[4], el cual permite realizar muchos tipos de gráficos, que pueden utilizarse en el diseño y documentación de sistemas de información. Esta herramienta tiene la particularidad de que puede utilizarse en distintas etapas del aprendizaje de la programación de computadores, partiendo de diagramas de flujo, en el aprendizaje de programación algorítmica y luego utilizando diagramas UML en el aprendizaje de la programación orientada a objetos. La licencia del producto permite su uso educativo y profesional, pero no es un producto de código abierto y no admite la programación de *plug-ins*. No obstante, y gracias a que lee y graba en formato graphML, motiva el presente desarrollo para crear una aplicación complementaria que genere código a partir de diagramas de flujo previamente creados con yEd Graph Editor® u otro editor que soporte graphML.

Se menciona este editor por la amplitud de diagramas que permite diseñar y cuya salida será tomada como entrada del software propuesto, en principio, para la generación de código a partir de diagramas de flujo, dejando para futuras ampliaciones

1 Ver http://graphml.graphdrawing.org
2 Secuencia finita de pasos para resolver un problema.
3 Ver http://www.w3.org/XML/
4 Ver http://www.yworks.com/en/products_yed_about.html

la utilización de otros diagramas con el mismo objetivo.

La motivación del trabajo parte de su posible aplicación pedagógica en la enseñanza de la programación algorítmica, en este sentido, es posible utilizar diagramas de Nassi-Shneiderman (NSD)[2] o diagramas de Chapin para la representación de un programa estructurado; este tipo de diagramas han sido ampliamente utilizados, la principal dificultad de su aplicación radica en la dificultad del alumno para percibir con claridad una estructura de selección dentro del contexto de otras estructuras repetitivas. Dificultad creciente en proporción directa con la extensión del algoritmo[5]. Esta dificultad pareciera ser menor en los diagramas de flujo, la estructura de selección queda perfectamente delimitada y es más perceptible por el alumno, no obstante, se requiere de cierta habilidad en la diagramación de estructuras repetitivas para no caer en diagramas no estructurados[6].

El software a desarrollar estará orientado a la interpretación de diagramas de flujo de datos con las características antes descriptas y diseñados con yEd Graph Editor®.

Elementos del Trabajo y metodología

Todo el trabajo se desarrolló en las siguientes etapas, tomando los lineamientos generales de Blanchette[3]:

1. Requerimientos
2. Análisis de archivos graphML.
3. Diseño e implementación del software base.
4. Diseño e implementación de generador ANSI C.
5. Implementación de diagramas de ejemplo y Testing.
6. Empaquetado, publicación y distribución.

1. Requerimientos

El software deberá interpretar el código graphML generado por el editor yEd Graph Editor® al grabar un diagrama de flujo (flowchart).

La interpretación del código graphML deberá hacerse a través de un intérprete que seleccione el usuario desde la interfaz gráfica del software (o bien se indique en línea de comandos), pues, un mismo diagrama de flujo puede ser interpretado por distintos intérpretes que generen código para distintos lenguajes de programación.

El alcance de este trabajo se limita al desarrollo del software base[7] para luego, sobre el mismo, implementar, al menos, un intérprete para generar código en lenguaje ANSI C[8], a ser ejecutado en plataforma Linux utilizando compilador GCC[9].

5 La observación esta basada únicamente en la experiencia personal de la enseñanza de diagramación lógica y programación algorítmica a alumnos de primer año universitarios y no universitarios del nivel superior durante más de 20 años. No se tiene referencia si hay estudios realizados en este sentido.

6 Idem anterior, además se utiliza un icono de yEd Graph Editor® para indicar el comienzo y fin de una estructura repetitiva, lo cual facilita la estructuración del diagrama.

7 Aplicación que implementa todo el código genérico de parsing de archivo graphML para luego dejar el resultado de este proceso disponible a los generadores de código que implemente el usuario.

8 La elección del lenguaje se debe a la necesidad que se tiene de generar código para este lenguaje, ya que es el lenguaje que actualmente se utiliza en la enseñanza de la programación algorítmica en las instituciones educativas vinculadas con el autor.

9 El compilador C/C++ para plataformas GNU/Linux de la Free Software Foundation, ver https://gcc.gnu.org

El software base deberá implementar una forma sencilla de incorporar nuevos generadores de código para distintos lenguajes que puedan ser desarrollados por terceros.

Cada generador de código deberá tener un manual en línea, consultable por el usuario desde la aplicación base, que indique la sintaxis que el usuario debe utilizar en el diagrama[10] para que el generador pueda interpretarlo y de esta forma, generar el código con la menor cantidad de errores sintácticos posibles.

El código generado deberá ser editado en forma interactiva por el usuario, a modo de poder realizar las correcciones que considere necesarias, previo a la compilación del mismo.

El software tendrá dos formas de ejecución: una ejecución interactiva, en donde el usuario interactúa a través de una interfaz gráfica y una forma de ejecución *batch*, en donde no hay interacción con el usuario, las acciones a realizar se indicarán por línea de comandos.

Cada generador de código puede tener sus propias restricciones en cuanto al compilador, intérprete, plataforma, arquitectura, etc. al que este orientado, no obstante, el software de base debería tener características multiplataforma, ya que los archivos con formato graphML podrían trasladarse de una plataforma a otra y el usuario final podría generar código aunque no tenga el compilador o intérprete instalado en su máquina. Esto se debe a la orientación pedagógica de este trabajo, en donde el usuario final, en su gran mayoría se tratará de alumnos que deben experimentar con el diagrama, a modo de explorar sus posibilidades y observar el código generado ante los distintos cambios que realicen sobre el mismo.

El software deberá emitir los mensajes de error correspondientes que pudieran ocurrir durante el proceso de generación de código o de interpretación/representación del diagrama.

El software deberá permitir al usuario copiar el código generado y/o guardarlo para su posterior uso.

2. Análisis de archivos graphML.

El archivo graphML generado por yEd Graph Editor® para representar un diagrama flujo, esta formado por una serie de *tags*[11] XML, entre los cuales son de interés para este desarrollo:

2.1 Tag graphml

Este tag es el inicial del archivo y a partir del mismo comenzará todo el proceso de *parsing*[12] del archivo graphML; sin la presencia de este tag, no se considerará un archivo graphML válido. Este tag contiene un tag *graph* que a su vez, contendrá el resto de tags del diagrama.

2.2 Tag node

El tag *graph* contendrá una serie de tags *node* que representan cada una de las figuras que componen el diagrama. Cada *node* tiene su propiedad *id* que lo identifica. Dentro de *node*, esta el tag *GenericNode* el cual tiene la propiedad *configuration* que indica un nombre de clase asociada con el

10 Por ejemplo, en el icono para el ingreso por teclado, el usuario deberá indicar -además del nombre de la variable a utilizar- su tipo de dato y/o otros datos adicionales que se requieran para la generación del código.

11 Secuencia de caracteres con el formato <nombre tag>contenido</nombre tag>. Ejemplo <html> .. </html>. Utilizados en lenguajes como html, xml, graphML.

12 Proceso computacional por el cual, la entrada se transforma en una estructura interna (árbol, lista, cola, etc.) que luego facilita su posterior uso.

node que sirve para interpretar el tipo de gráfico del nodo. Dentro del tag *GenericNode* se encuentra el tag *NodeLabel* que contiene el texto ingresado sobre dicha figura por parte del usuario.

2.3 Tag edge

El tag *graph* contendrá una serie de tags *edge* que representan cada una de las lineas que conectan las distintas figuras (nodos) del diagrama. Cada *edge* tiene la propiedad *id* que lo identifica, la propiedad *source* que indica el *id* del nodo origen de la línea y la propiedad *target* que indica el *id* del nodo destino de la línea. De esta forma, es posible conectar a cada nodo del diagrama.

A partir de las descripciones anteriores, un archivo graphML podría tener el siguiente formato (resumido):

```
<graphml ...>
 …
  <graph ..>
    <node id=”id0” ...>
      <data ..>
        <y:GenericNode
configuration="com.yworks.flowchart.start1">
          <y:NodeLabel ...>Texto del Nodo
          </y:NodeLabel>
        </y:GenericNode>
      </data>
    </node>
    ….
    <edge id="e0" source="id0" target="id1">
      <data ...>
        <y:PolyLineEdge>
          <y:EdgeLabel ...>Texto de la Linea
          </y:EdgeLabel>
        </y:PolyLineEdge>
      </data>
    </edge>
    ...
  </graph>
   ...
</graphml>
```

El archivo real muestra una complejidad mayor, se han omitido muchos tags y detalles, para facilitar la comprensión del formato y además no son relevantes para este desarrollo.

3. Diseño e implementación del software base.

Acorde a los requerimientos propuestos, el software se desarrollará en lenguaje Java para cumplir con el requerimiento de multiplataforma y también para facilitar la incorporación de nuevos generadores de código en tiempo de ejecución, haciendo uso de la tecnología Reflection[13] implementada en Java.

El proceso a realizar consta de dos etapas: una primera etapa es el procesamiento del código graphML que se resume a un *parsing* de XML, para ello, puede utilizarse el parser SAX, el cual es parte de la “Java API for XML Processing” (JAXP) [6]; la segunda etapa debería tomar el resultado del *parsing* XML, interpretarlo y generar el código en el lenguaje que corresponda.

La primera parte del proceso es genérica, mientras que la segunda parte del proceso es específica de cada generador de código.

La parte genérica del proceso, formará parte de lo que aquí llamamos software base y nos concentraremos en ella en esta sección.

Se debe representar el resultado del *parsing* XML en términos de objetos, para ello, se crearon las siguientes clases:

Clase	**Propiedad**	**Descripción**
Node	Id	Id del nodo que lo

13 Permite que un programa, en tiempo de ejecución, pueda autoexaminarse o examinar a otros programas, para cambiar su comportamiento, obtener información de metadata, etc.

(representa una figura del diagrama)		identifica
	next1	Referencia a Node por verdadero
	next2	Referencia a Node por falso
	previous	Referencia a Node anterior
	type	Propiedad configuration de tag GenericNode
	text	Texto del Nodo
Edge (representa una línea que conecta dos nodos del diagrama)	id	Id de la línea que la identifica
	source	Id del nodo que conecta en su origen
	target	Id del nodo que conecta en su destino
	text	Texto de la línea

Tabla 1.Clases Package model

A medida que se realiza el *parsing* XML de tags *node*, se crean objetos Node y éstos se almacenan en una lista cuya referencia estática es mantenida por la clase *Storage*. Esta clase también mantiene una referencia al primer nodo del diagrama. Cuando se encuentra un tag *edge*, se crea un objeto *Edge* y se utiliza la propiedad *source* y *target* para encontrar las referencias a los *nodos source* y *target* correspondientes, una vez ubicados los nodos, se procede a su vinculación de la siguiente forma:

-Si el texto del objeto *Edge* implica verdadero[14] (contiene alguno de los siguientes caracteres: “V” , ”v”, ”T”, ”t”, ”verdadero”, ”verdad”, ”true”) entonces se asigna el *nodo target* a la propiedad next1 del *nodo source*.

-Si el texto del objeto *Edge* implica falso[15] (contiene alguno de los siguientes caracteres: “F” , ”f”, ”M”, ”m”, ”mentira”, ”falso”, ”false”) entonces se asigna el *nodo target* a la propiedad next2 del *nodo source*.

-Si el texto del objeto *Edge* no implica ni verdadero ni falso[16] entonces se asigna el *nodo target* a la propiedad next1 del *nodo source*.

Al finalizar todo el proceso de *parsing*, la clase Storage cuenta con una referencia al primer nodo del diagrama y una lista doblemente enlazada que permite la navegabilidad de todos los nodos del diagrama. Esta lista es la salida de este proceso.

4. Diseño e implementación de generador ANSI C.

La aplicación cuenta con un archivo de propiedades en donde pueden *parametrizarse* varias características del software y su interfaz gráfica, entre ellas, se incorporaron 3 parámetros con el siguiente formato: *gen<nro>.name*, *gen<nro>.class*, *gen<nro>.param.file*; en donde *<nro>* es un número entero de 1 a N, que representa el número de generador de código. De esta forma, pueden agregarse todos los generadores de código que se desee. El primer parámetro es el nombre del generador, el segundo es el nombre completo de la clase Java que lo implementa y el tercero es el archivo de parámetros asociado con el mismo. Por ejemplo, para declarar el generador de código ANSI C, podemos indicar:

```
gen1.name=ANSI C
gen1.class=yEdgen.plugins.AnsiC
gen1.param.file=yEdgen/plugins/AnsiC.properties
```

14 Se asume que es una línea que parte de un nodo de selección.

15 Idem anterior.

16 Se asume que es una línea que conecta dos nodos del diagrama que no implica ninguna selección.

En este caso, la clase *yEdgen.plugins.AnsiC* es una clase que implementa la interfase *SourceGenerator*; esta interfase deberá ser implementada por todo generador de código. La interfase define los métodos *init()*, *start()* que serán ejecutados en ese orden y luego los métodos *getCode()* para obtener el código generado, *getErrors()* para obtener los mensajes de error de la generación de código y *getName()* que devuelve el nombre del generador de código. Todo generador de código deberá implementar la interfase *SourceGenerator*.

Cuando se inicia la interfaz gráfica de la aplicación, ésta verifica su archivo de parámetros y carga una lista desplegable con todos los generadores de código disponibles; de esta forma, el usuario puede seleccionar el generador de código que desea utilizar. Cuando se presiona el botón *"Start"* se ejecuta la primera parte del proceso descripta en el punto anterior y luego de ello se llama al método *init()* y luego *start()* implementados en la clase del generador de código, que previamente fue cargada utilizando *reflection*[7]. Por último, si la ejecución fue exitosa, se carga el contenido del área de texto del formulario principal con lo devuelto por el método *getCode()*.

4.1 Características del Generador ANSI C

Este generador ha sido implementado en forma experimental, para realizar una generación de código ANSI C básica, orientada a diagramas de flujos simples que representan algoritmos que combinan las estructuras de secuencia, selección y repetición[4].

Permite trabajar con los siguientes tipos de datos C: "char", "char *", "short", "int", "long", "float", "double". Resuelve la identación del código utilizando tabuladores. El ingreso de datos por consola se realiza utilizando la función *fgets()* para ingresar una cadena de caracteres y luego convierte dicha cadena al tipo de dato requerido.

Utiliza *printf()* para realizar la salida con formato de mensajes y variables. Asume un formato fijo para cada tipo de dato.

Implementa las estructuras repetitivas ANSI C: *do {... } while()*, *for()* y *while() {...}*.

4.1 Sintaxis

4.1.1 Inicio

Se utiliza el nodo de tipo *"Start 1"* para comenzar el diagrama y puede contener cualquier leyenda.

4.1.2 Fin

Se utiliza el nodo de tipo *"Terminator"* para finalizar el diagrama y puede contener cualquier leyenda.

4.1.3 Entrada de datos

Se utiliza el nodo de tipo *"Manual Input"* y en el texto del mismo, se debe utilizar la siguiente sintaxis:

```
prompt:var name:var type
```

Ejemplo:

Ingrese Nro:a:int

En este caso, el código generado va a mostrar la leyenda "Ingrese Nro" en la consola y permitirá el ingreso por teclado de la variable entera *a*.

4.1.4 Salida de datos

Se utiliza el nodo de tipo *"Display"* y en el texto del mismo se debe utilizar la siguiente sintaxis:

```
var1,var2,constante1,constant
e2,...
```

Ejemplo:

a,hola

En este caso, el código generador va a mostrar la variable *a* y luego el mensaje *hola*.

4.1.5 Proceso/Instrucción

Se utiliza el nodo de tipo *"Process"* y el texto del mismo puede incluir cualquier sentencia ANSI C. Incluso, soporta que aquí se realicen declaraciones de variables, por ejemplo:

```
int a;
```

también pueden inicializarse variables, llamar a funciones, asignaciones, etc., por ejemplo:

```
int a;a=4;
```

pueden ingresarse más de una instrucción separadas por punto y coma. No se puede declarar una variable e inicializarla al mismo tiempo, por ejemplo:

```
int a=4;
```

es una instrucción no válida.

4.1.6 Decisión

Se utiliza el nodo de tipo *"Decision"* y el texto del mismo puede ser cualquier expresión *booleana* válida expresada en ANSI C. Siempre salen dos líneas de este nodo: una indica la dirección a seguir "por verdadero" y otra "por falso". Estas líneas deben estar etiquetadas con las leyendas correspondientes, para que el sistema lo interprete como verdadero o falso[17]. Toda decisión siempre termina con un nodo de tipo "*Start 2*" (*endif*).

4.2 Parámetros

El generador de código ANSI C tiene una serie de parámetros indicados dentro del archivo *AnsiC.properties*:

Propiedad	**Valor**	**Descripción**
file.extension	.c	Extensión del programa a generar
file.extension.text	C Program	Nombre de la extensión del programa a generar
if.node	com.yworks.flowchart.decision	Clase yEd para nodo decisión
endif.node	com.yworks.flowchart.start2	Clase yEd para nodo fin decisión
input.node	com.yworks.flowchart.manualInput	Clase yEd para nodo ingreso por teclado
display.node	com.yworks.flowchart.display	Clase yEd para nodo de salida por pantalla
loop.node	com.yworks.flowchart.loopLimit	Clase yEd para nodo repetición
endloop.node	com.yworks.flowchart.loopLimitEnd	Clase yEd para nodo fin repetición
process.node	com.yworks.flowchart.process	Clase yEd para nodo proceso
includes	#include <stdio.h> ..	Cabeceeras de la librería C a incluir
main.prototipe	int main(int argc,char **argv)	Prototipo de la función main() a utilizar
help.text		Codigo html de ayuda

17 Los textos a utilizar se indicaron en la sección anterior.

start.text		Texto incial de programa C
end.text		Texto final de programa C

Tabla 2.Propiedades Generador ANSI C

4.3 Sustitución de Parámetros

Dentro del contenido de los parámetros *start.text* y *end.text* se pueden sustituir los valores de los parámetros, por ejemplo, si *start.text* contiene *$includes* durante la generación de código, *$includes* se reemplazará por el valor del parámetro *includes*.

Además de estas sustituciones de parámetros, también se agregan otros que pueden sustituirse:

Propiedad	**Descripción**
$filein	Nombre completo del archivo graphML de entrada.
$fileout	Nombre completo del archivo a generar
$generator	Nombre del generador de código
$now	Hora actual
$today	Fecha actual

Tabla 3.Parámetros sustituibles

5. Implementación de diagramas de ejemplo y Testing.

A medida que se avanzaba en el punto 4, también se hacía lo propio en este punto, ya que, se generaban diagramas de prueba con yEd Graph Editor® y se probaba la generación de código del mismo; de forma iterativa.

Se generaron una serie de diagramas simples que abarcan todas las posibilidades sintácticas básicas del generador de código. Los mismos se incluyen dentro de la carpeta *examples*.

Una vez generado el código, el mismo se compilaba con el compilador GCC y luego se ejecutaba para probar su correcto funcionamiento.

6. Empaquetado, publicación y distribución.

Todo el desarrollo se distribuye a través de SourceForge[18] bajo el nombre de proyecto *yEdGen* en donde se incluye todo el código con programas fuentes, documentación y diagramas de ejemplo.

El proyecto *yEdGen* se distribuye bajo licencia LGPL[19]. La versión compilada del proyecto permite su ejecución a partir de archivo *.jar*[20] Java.

Resultados

Se considera que el proyecto ha logrado alcanzar los objetivos propuestos. Si bien los diagramas que han sido utilizados hasta el momento en las pruebas de generación de código realizadas son de baja complejidad, han permitido hacer correcciones sobre el código y mejorar la calidad del mismo.

El software permite mostrar al usuario los errores encontrados en la generación de código, seguramente en esta área queda trabajo por realizar, pues se han generado errores siempre a partir de diagramas que han sido construidos siguiendo las reglas sintácticas del generador de código y se requiere la prueba del sistema con diagramas que no respetan la sintaxis o incluso, ni siquiera se trate de diagramas de flujo.

Las pruebas realizadas en Linux han sido satisfactorias, permitiendo la generación de código de todos los ejemplos

18 Ver http://sourceforge.net/
19 http://www.gnu.org/licenses/lgpl.html
20 Archivo comprimido Java Archive .jar

que se distribuyen junto con el software y su compilación.

Discusión

Se espera que la publicación del software permita un *feedback* por parte de usuarios que comiencen a usarlo. La interfase (si bien es *parametrizable*) esta en inglés para facilitar el acceso a una comunidad de usuarios mas amplia. En este sentido, también se espera que se creen nuevos generadores de código a partir del generador ANSI C aquí presentado.

Se espera que futuros desarrollos permitan la prueba del software propuesto con un número mayor de generadores de código instalados y una mayor variedad de diagramas.

Las pruebas se han realizado sobre plataforma Linux Debian amd 64 bits utilizando openjdk-7, se espera que la comunidad de usuarios permita ampliar la prueba a otras plataformas.

Se ha generado un código compacto, teniendo en cuenta la complejidad de la aplicación. Los tiempos de *parsing* y generación de código son aceptables, permiten un trabajo interactivo fluido con el usuario y los requerimientos de hardware son idénticos al requerimiento de Java JRE versión 7. Las pruebas de sucesivas ejecuciones no permiten apreciar una degradación en la performance del sistema.

Desde el punto de vista didáctico, bien podrían hacerse proyectos desde asignaturas tales como: Algoritmos y Estructuras de Datos, Paradigmas de la Programación, Sistemas y Semántica de los lenguajes, Sistemas Operativos, etc.

Conclusión

Se ha logrado concebir un software que puede ser una herramienta valiosa en el aprendizaje de la programación algorítmica, que suele presentar cierta dificultad en los alumnos ingresantes a la Ingeniería en Sistemas de Información.

El software también puede utilizarse en aplicaciones comerciales y su arquitectura permite su fácil extensión y crecimiento creando nuevos generadores de código para distintos lenguajes.

El software puede considerarse también un punto de partida para nuevos desarrollos de generación de código a partir de otros tipos de diagramas.

El código resultante es de escasa extensión, fue desarrollado en un entorno de desarrolllo BlueJ[21], lo cual facilita su *deployment* y distribución en múltiples plataformas. Cuenta además con el soporte de la plataforma de *SourceForge*.

Este trabajo abre nuevas lineas de investigación en cuanto a: mejoras que puedan introducirse al software, en cuanto a la creación de nuevos generadores de código, inclusión o extensión de la sintaxis actual para ampliar las capacidades del generador de código, la posibilidad de procesar otros tipos de diagramas, pruebas con archivos graphML generado con otros editores, mayor integración entre editores de archivos graphML y este software, mayor integración entre este software y los compiladores o intérpretes para los cuales se genera el código, posibilidad de ingeniera reversa, etc..

A partir de la arquitectura actual, la separación de clases acorde con su responsabilidad, etc., facilita la posibilidad de cambiar la interfase del sistema y

21 Ver http://www.bluej.org/ entorno de desarrollo Java muy simple, utilizado para el aprendizaje de este lenguaje.

orientarlo a una aplicación de tipo client-server o bien en una arquitectura n-tier[5] que permita su uso en un ambiente intranet o internet.

Agradecimientos
Al Ing. Luis H. Perna de la Universidad Tecnológica Nacional, Facultad Regional Delta, por motivarme a la presentación del presente trabajo.

Referencias

[1]U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, M.S. Marshall, GraphML Progress Report: Structural Layer Proposal., P*roc. 9th Intl. Symp. Graph Drawing (GD '01), LNCS 2265, © Springer-Verlag, 2002.*
[2]Ben Shneiderman, A short history of structured flowcharts (Nassi-Shneiderman Diagrams), Draft: May 27, 2003, disponible en http://www.cs.umd.edu/hcil/members/bshneiderman/nsd/
[3] Blanchette, Jasmin, The Little Manual of API Design, Trolltech, a Nokia company, June 19, 2008, disponible en http://www4.in.tum.de/~blanchet/api-design.pdf
[4] Bohm, Corrado, Giuseppe Jacopini, Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules, Communications of the ACM 9, 1966, pp. 366–371
[5] Sheriff Paul D, Fundamentals of N-Tier Architecture, PDSA, Inc., May 1, 2006.
[6] Trail: Java API for XML Processing (JAXP), Oracle Corp., Copyright 1995-2014, disponible en http://docs.oracle.com/javase/tutorial/jaxp/
[7] Trail: The Reflection API (The Java Tutorials), Oracle Corp., Copyright 1995-2014, disponible en http://docs.oracle.com/javase/tutorial/reflect/
.

Datos de Contacto:

Cherencio, Guillermo Ruben. Universidad Tecnológica Nacional, Facultad Regional Delta. San Martin 1171, CP 2804, Campana, Provincia de Buenos Aires, República Argentina. E-mail: grchere@yahoo.com.

Automatización de la Selección de Cloud Applications Invocadas por un Workflow

Marcela Daniele, Fabio Zorzan, Ariel Arsaute y Ariel Gonzalez
Departamento de Computación
Universidad Nacional de Río Cuarto
Ruta 36 km 601 CP X5804BYA, Río Cuarto, Córdoba, Argentina
{marcela, fzorzan, aarsaute, agonzalez}@dc.exa.unrc.edu.ar

Abstract

Las tecnologías de Workflows tienen como característica principal, automatizar procesos de negocio. La Workflow Management Coalition (WfMC) estandarizó esta automatización, definiendo un módulo especial para invocación de aplicaciones externas. En este marco, en trabajos anteriores se propuso optimizar la invocación de dichas aplicaciones a través de Servicios Web (SW). Así mismo, se propuso un proceso automático de selección de SW, en el cual se utiliza el lenguaje Relations de Query/View/Transformation (QVT). El objetivo de esta automatización es independizarse de la ubicación exacta de las aplicaciones y, en tiempo de ejecución, seleccionar la que más se adecúe a los requerimientos del usuario. Así, el presente trabajo propone avanzar en automatizar este proceso, concretamente automatizando la transformación de la especificación en Web Service Definition Language (WSDL) al modelo UML en formato Ecore, lo cual anteriormente se realizaba de forma manual. Para realizar dicha transformación se utiliza XML Stylesheet Language Transformations (XSLT).

Keywords: Workflow Management Systems, Web Services, WSDL, XSLT, QVT

1. Introducción

Cloud computing es un paradigma de computación que permite ofrecer servicios de computación por medio de Internet. Cloud computing incorpora la idea de ver al software como un servicio y, en este contexto, los servicios web juegan un papel fundamental.

El presente trabajo es una continuación de trabajos anteriores, donde los autores se han concentrado en la utilización de los servicios web (SW) en el marco de los Sistemas de Gestión de Workflow, y su necesidad de invocar aplicaciones externas en determinadas circunstancias.

El Modelo de Referencia de Workflow [1] estandariza la automatización de los procesos de negocio, en particular, define la Interfaz de las Aplicaciones Invocadas [2] para permitir la invocación de las aplicaciones desde un workflow. Esta interfaz, especifica la interacción con software existente que el workflow puede utilizar para la realización de ciertas actividades, teniendo en cuenta que, dicho software se puede encontrar en cualquier plataforma o lugar de la web.

En la actualidad, esta comunicación del workflow con las aplicaciones externas es estática, lo cual deriva en diversos problemas: cualquier acontecimiento producido en las aplicaciones externas se transferirá al workflow al momento de su invocación, como por ejemplo la no disponibilidad de la aplicación ya sea temporal o no, o una actualización de la misma que conlleve un cambio de ubicación, e incluso una modificación del formato de respuesta que hasta entonces era entendible para el workflow. Cualquiera de estos sucesos, provocará que el workflow posponga o aborte la invocación de la aplicación. En este marco, y tendiendo a mejorar esta selección e invocación, se ha propuesto el uso de SW.

Los SW proveen esencialmente un medio estándar de comunicación entre diferentes aplicaciones de software, ofreciendo la capacidad de acceder a servicios heterogéneos de forma unificada a través de Internet [3]. Con la utilización de los SW en la invocación de aplicaciones externas desde un workflow, el motor del mismo no necesita conocer la ubicación exacta de la aplicación a invocar, y las aplicaciones pueden variar su ubicación en la red sin provocar cambios en su invocación.

Actualmente, los SW son registrados a través del protocolo denominado UDDI [4] (Universal Description, Discovery and Integration), el cual presenta la dificultad de establecer una correspondencia entre los requerimientos del solicitante y las múltiples especificaciones de SW existentes en la actualidad que

proveen similares funcionalidades, aunque tienen diferente descripción sintáctica.

En [5] y [6], los autores se concentran en la utilización de los SW para invocar aplicaciones externas en los Sistemas de Gestión de Workflow mediante Técnicas de Transformación de Grafos, las cuales definen la contrastación de las especificaciones de los SW ofrecidos por los proveedores con las de los usuarios del workflow.

La técnica empleada proporciona una especificación semántica precisa de los SW a través de la representación de sus pre y postcondiciones como grafos con atributos. Si bien las herramientas basadas en transformaciones de grafos se encuentran teóricamente muy bien fundamentadas, y además, se ajustan coherentemente a los lenguajes gráficos de modelado como UML [7], resultan muy sensibles a la complejidad proveniente del carácter no determinista de la ordenación y estrategia de aplicación de reglas, a lo que debe agregarse el costo computacional del tratamiento de grafos y el evidente esfuerzo que conlleva la especificación y manipulación de transformaciones de modelos de gran tamaño.

En [8] se propone utilizar una propuesta alternativa para la contrastación de SW mediante la utilización de QVT [9]. QVT es un estándar definido por la OMG para trabajar con modelos de software. En particular se propone trabajar con el lenguaje declarativo denominado Relations. . El lenguaje QVT/Relations permite definir transformaciones de modelos que pueden ser utilizados para generar un modelo a partir de un modelo fuente o contrastar la correspondencia de dos modelos, esta última característica es la que se aplica en este trabajo. Además, en la actualidad existen herramientas maduras que implementan la especificación del lenguaje, por ejemplo MediniQVT [10].

En este marco, el presente trabajo propone avanzar en la automatización del proceso de selección de servicios web, en especial, automatiza el proceso que consiste en transformar la especificación WSDL de los requerimientos del usuario al modelo UML y, de la misma manera, con la especificación WSDL de los SW ofrecidos por los proveedores. Estas transformaciones a modelos UML, en trabajos previos se realizaban de forma manual. Para realizar dicha transformación se propone utilizar el lenguaje de Transformaciones XML Stylesheet Language Transformations (XSLT) [11].

El trabajo se estructura como sigue: en la sección 2 se describen brevemente los Sistemas de Gestión de Workflow, los SW y su utilización en la invocación de aplicaciones externas a un workflow. La sección 3 describe lenguaje de Transformaciones XML, y la sección 4 presenta el estándar QVT. La sección 5 muestra el esquema de contrastación de SW en el marco de los Sistemas de Gestión de Workflow. La sección 6 describe la automatización del proceso de contrastación con QVT. La sección 7 presenta un ejemplo de aplicación. Por último, se presentan las conclusiones de este trabajo.

2. Los Workflows y la invocación de Servicios Web

Los Workflows se relacionan con la automatización de los procedimientos de un Negocio donde los documentos, la información o tareas se pasan entre los participantes del mismo de acuerdo a un conjunto de reglas previamente establecidas.

Los Sistemas de Gestión de Workflow (WFMS por sus siglas en inglés) son sistemas de software que definen, crean y manejan la ejecución de Workflows.

La Workflow Management Coalition (WfMC) [1] ha estandarizado la automatización de los procesos de negocio, definiendo un marco genérico para la construcción de WfMS, permitiendo interoperabilidad entre ellos y con otras aplicaciones involucradas. En particular, define la Interfaz de las Aplicaciones Invocadas, que permite la invocación de aplicaciones externas, como por ejemplo servicios de e-mail, fax, administración de documentos, o aplicaciones de usuario.

Si bien en muchos WFMS se conocen a priori las aplicaciones que se desean utilizar y su ubicación, existen sistemas donde las aplicaciones requeridas por el motor están distribuidas en la red y no se conoce su ubicación. La tecnología implícita en los SW suministra una solución a este inconveniente planteado. Los SW proveen un enfoque basado en estándares de Internet para implementar componentes distribuidos, y ofrecen los mecanismos necesarios para la integración de aplicaciones usando protocolos estándares. Debido a la propagación y al bajo costo de Internet, los SW proveen una excelente solución a los problemas de interoperabilidad. El desarrollo de aplicaciones basadas en SW reduce costos de integración mejorando los niveles y calidad de los servicios.

3. El lenguaje de Transformaciones XML Stylesheet Language Transformations (XSLT)

Una hoja de estilo es una representación de las opciones de formato para un documento, normalmente almacenado fuera del documento mismo, por lo que el formato está separado del contenido.

Por ejemplo, una hoja de estilo para HTML puede especificar la fuente a utilizar en todas las cabeceras, y por ello, reemplaza un gran número de declaraciones de fuente en la página HTML.

XML XSL (Stylesheet Language), el lenguaje de hojas de estilo XML, estaba originalmente diseñado para generar HTML a partir de XML y es por ello una

extensión lógica de hojas de estilo HTML. El lenguaje incluye un mecanismo de transformación de propósito general, denominado XSLT (XSL Transformations, transformaciones de XSL), que se puede usar para transformar un documento XMl en otro documento XML, o a otros formatos como HTML. Las transformaciones XSLT son bastante potentes y en realidad XSLT puede incluso actuar como lenguaje de consulta. Las transformaciones XSLT se expresan como una serie de reglas recursivas, denominadas plantillas.

Aunque XSLT se puede usar como lenguaje de consulta, su sintaxis y semántica son bastante distintas de las de SQL. Una plantilla sencilla para XSLT consiste en una parte de coincidencia (match) y una parte de selección (select) (Figura 1).

```
<xsl:template match="/banco-2/cliente">
        <xsl:value-of select="nombre_cliente"/>
</xsl:template>
<xsl:template match="*"/>
```

Figura 1. Ejemplo de código XSLT

La instrucción xsl:template match contiene una expresión XPath que selecciona uno o más nodos. La primera plantilla busca coincidencias de elementos cliente que aparecen como hijos del elemento raíz banco-2. La instrucción xsl:value-of encerrada en la instrucción de coincidencia devuelve valores de los nodos en el resultado de la expresión XPath. La primera plantilla devuelve el valor del subelemento nombre_cliente. La segunda plantilla coincide con todos los nodos.

XSLT fue diseñado como leguaje de transformación, como parte del sistema de hojas de estilo XSL, que se usa para controlar el formato de los datos XML en HTML u otro lenguaje de impresión o visualización. Aunque diseñado para el formato, XSLT puede generar XML y expresar muchas consultas interesantes. Actualmente es el lenguaje más utilizado para la manipulación de datos XML.

4. Query/View/Transformation – QVT

El estándar QVT [9] fue definido por la OMG para trabajar con modelos de software. QVT consta de tres partes: consulta, vista, y transformaciones. El componente de consultas de QVT toma como entrada un modelo, y selecciona elementos específicos del mismo. Para la resolución de las consultas, se propone el uso de una versión extendida de OCL 2.0 [12]. Una vista puede verse como el resultado de una consulta sobre un modelo, ofreciendo como resultado un punto de vista de éste, restringiéndolo de acuerdo ciertas condiciones.

En esta sección, se describe brevemente el componente de transformaciones de QVT utilizado para definir transformaciones. Estas transformaciones describen relaciones entre un metamodelo fuente F y un metamodelo objetivo O, ambos metamodelos deben estar especificados en Meta Object Facility (MOF) [13]. Una vez definida la transformación, la misma se utiliza para obtener un modelo objetivo, que es una instancia del metamodelo O, a partir de un modelo fuente, que es una instancia del metamodelo F. También la transformación puede ser utilizada para chequear la correspondencia entre dos modelos.

La especificación de QVT 1.0 [9] tiene una naturaleza híbrida declarativa/imperativa. En este trabajo se aplica el lenguaje declarativo Relations.

4.1. El lenguaje Relations

El lenguaje Relations es una especificación declarativa de las relaciones entre metamodelos MOF.

Una transformación especifica un conjunto de relaciones que deben cumplir los elementos de los modelos involucrados. La ejecución de una transformación en una dirección particular se realiza seleccionando qué modelo será el objetivo de la transformación. La ejecución de una transformación intenta realizar las relaciones modificando el modelo objetivo.

Una relación puede ser declarada en modo sólo chequeo (check only) o forzado (enforced). Si un domino es marcado como sólo chequeo, cuando se ejecute la transformación, sólo será chequeado para ver si existe una correspondencia válida con otro dominio perteneciente al otro modelo; en cambio, si el modelo está marcado como forzado, cuando una relación no se satisface, los elementos del modelo pueden ser creados, borrados o eliminados en el modelo objetivo para hacer que la relación se satisfaga.

A continuación se presenta la herramienta que implementa la especificación QVT/Relations que se utilizó en esta investigación.

Eclipse Modeling Framework (EMF) [14] es un marco de modelado para la herramienta Eclipse, que es una plataforma de desarrollo de código abierto (open source) basada en Java. EMF toma un modelo definido en UML, un esquema XML (Extensible Markup Language) [15] o una interfaz Java, y genera las correspondientes clases de implementación.

Los modelos descritos en EMF se representan en un modelo interno llamado Ecore. EMF es la realización de IBM de MOF [13]. MOF es un estándar de la OMG para describir repositorios de metadatos. MOF es similar a Ecore en su capacidad de especificar clases con sus características de comportamiento y estructurales, herencia, paquetes y reflexión. Difieren en que MOF tiene complejidades adicionales para el ciclo de vida, estructuras de datos y tipos complejos de asociaciones.

La herramienta utilizada en el trabajo anterior [8] para la definición de las transformaciones se denomina MediniQVT [10], esta herramienta implementa la especificación QVT/Relations de la OMG en un poderoso motor QVT. MediniQVT está integrada a Eclipse y utiliza EMF para la representación de modelos. El proceso de selección de Aplicaciones Externas a un Workflow con Servicios Web.

5. Esquema general para la selección de SW utilizando Relations

A continuación se describe brevemente el esquema general para la invocación de SW desde un workflow. En la Figura 2 se observan los SW ofrecidos por los proveedores a través de UDDI, los SW solicitados en la invocación de aplicaciones externas y la correspondencia realizada por el motor workflow.

En el esquema propuesto, el componente principal es el motor del workflow, quien actúa como un agente de descubrimiento de servicios, basándose en los requerimientos funcionales y las características de calidad especificadas por el usuario. Para realizar la correspondencia y selección final, el motor realiza los siguientes pasos:

1.a) Especificación WSDL de los servicios web solicitados desde el Workflow y ofrecidos por proveedores de SW. La especificación de un SW solicitado contiene los requerimientos funcionales, como el tipo de aplicación, parámetros y requerimientos adicionales; y los requerimientos no funcionales, referidos a los atributos de calidad del servicio web. Las especificaciones de SW ofrecidos contienen información que se puede obtener del registro UDDI, referida principalmente a las características funcionales del servicio web. Estas especificaciones están definidas originalmente en un documento WSDL, que representa el metamodelo de los SW solicitados desde el motor del workflow y ofrecidos por proveedores de servicios web (ver Figuras 3, 4 y 5).

1.b) Obtención automática con XSLT de los modelos UML. A partir de las especificaciones WSDL de los servicios web solicitados y ofrecidos, se generan automáticamente los modelos UML(en formato Ecore), que serán la base de la contrastación en QVT/Relations.

2) Realización del proceso de contrastación. El motor workflow busca en el registro UDDI todos los SW que se corresponden con los requerimientos del solicitante y obtiene una lista preliminar. Durante este proceso de contrastación utiliza QVT para analizar la correspondencia de servicios, en base a las transformaciones definidas en este trabajo.

3) Invocación del SW elegido. Finalmente el motor workflow invoca el SW que resultó seleccionado.

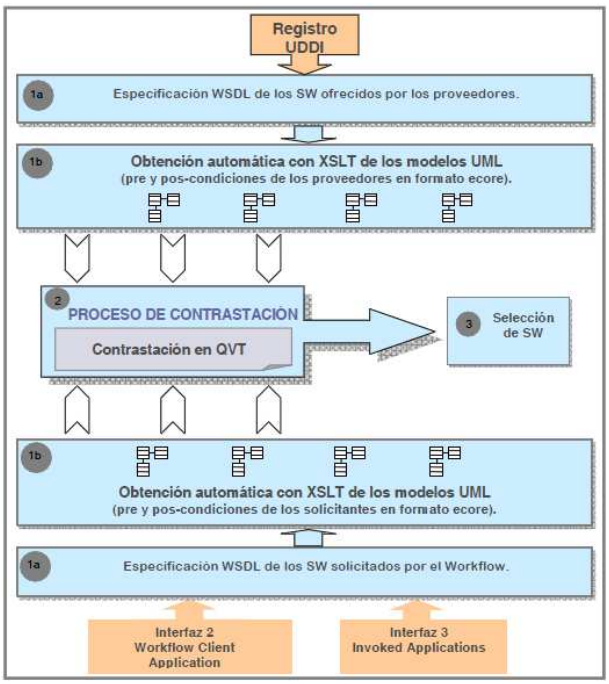


Figura 2. Esquema general para la selección de SW.

```
- <xsd:schema>
  - <xsd:element name="description">
    - <xsd:complexType>
      - <xsd:sequence>
        - <xsd:element name="Types">
          - <xsd:complexType>
            - <xsd:sequence>
              + <xsd:element name="InvokeApp"></xsd:element>
              + <xsd:element name="AppFound" minOccurs="0"
                             maxOccurs="unbounded"></xsd:element>
              + <xsd:element name="WFException" minOccurs="0"
                             maxOccurs="unbounded"></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figura 3. Encabezado de la especificación WSDL

En la siguiente sección se presenta una aproximación a la automatización del proceso de contrastación utilizando QVT/Relations.

6. Transformación XSLT de la especificación WSDL a formato Ecore

Para realizar la contrastación de servicios web, cada una de las especificaciones WSDL presentadas en la sección anterior se expresa como dos grafos con atributos, representando la precondición del SW, es decir, el estado del sistema antes de la ejecución del SW y la

postcondición, es decir, la situación después de satisfacerse la ejecución del SW.

```
- <xsd:element name="InvokeApp">
  - <xsd:complexType>
    - <xsd:sequence>
        <xsd:element name="appType" type="xsd:string"/>
      - <xsd:element name="wfInformation">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element name="procInstId" type="xsd:string"/>
              <xsd:element name="workItemId" type="xsd:string"/>
              <xsd:element name="appMode" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      - <xsd:element name="parameters" minOccurs="0" maxOccurs="unbounded">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="description" type="xsd:string"/>
              <xsd:element name="value" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      - <xsd:element name="aditionalReq" minOccurs="0" maxOccurs="unbounded">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="description" type="xsd:string"/>
              <xsd:element name="value" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      - <xsd:element name="QoSReq" minOccurs="0" maxOccurs="unbounded">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element name="QoSname" type="xsd:string"/>
              <xsd:element name="QoSunitType" type="xsd:string"/>
              <xsd:element name="QoSminValue" type="xsd:float"/>
              <xsd:element name="QoSmaxValue" type="xsd:float"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

Figura 4. Especificación WSDL del elemento InvokeApp.

```
- <xsd:element name="AppFound" minOccurs="0" maxOccurs="unbounded">
  - <xsd:complexType>
    - <xsd:sequence>
        <xsd:element name="appName" type="xsd:string"/>
        <xsd:element name="appLocation" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="WFException" minOccurs="0" maxOccurs="unbounded">
  - <xsd:complexType>
    - <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="message" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

Figura 5. Especificación WSDL de los elementos AppFound y WFException

Para ello se realiza una traducción de las especificaciones WSDL al formato Ecore, a través de la definición de una transformación XSLT. Las Figuras 6 y 7 muestran la transformación definida para el caso de la postcondición de un SW. Una vez realizada la traducción a formato Ecore (punto 1.b del esquema general de selección de SW presentado en la sección 5), es posible continuar con el proceso de contrastación de SW.

```
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
    <xsl:param name="a-param">default param value</xsl:param>
    <xsl:output encoding="iso-8859-1" />
  - <xsl:template match="description/Types/InvokeApp">
    - <PostCondicionServicioOfrecido:AppRequiredType xmi:version="2.0"
        xmlns:xmi="http://www.omg.org/XMI"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:PostCondicionServicioOfrecido="PostCondicionServicioOfrecido"
        xsi:schemaLocation="PostCondicionServicioOfrecido ..
                            .../Metamodelos/PostCondicionServicioOfrecido.ecore">
      + <appType>
      + <requeriments xsi:type="PostCondicionServicioOfrecido:AditionalReq">
      + <parameters xsi:type="PostCondicionServicioOfrecido:Parameter">
      + <requeriments xsi:type="PostCondicionServicioOfrecido:QoSReq">
      + <appFound xsi:type="PostCondicionServicioOfrecido:AppFound">
      </PostCondicionServicioOfrecido:AppRequiredType>
    </xsl:template>
  </xsl:stylesheet>
```

Figura 6. Especificación de la Transformación XSLT.

Ya en el proceso de contrastación de SW, se procede a comparar los grafos generados por el solicitante y el proveedor. Se formaliza usando relaciones de subgrafos. Si la precondición del proveedor es un subgrafo de la precondición del solicitante, entonces el proveedor provee toda la información necesaria para ejecutar el servicio web. Si la postcondición del solicitante es un subgrafo de la postcondición del proveedor, entonces el servicio genera todos los efectos esperados por el solicitante.

En [8], se definieron las siguientes dos transformaciones en QVT/Relations:

- ***Transformación 1***: la inclusión de la postcondición del servicio web solicitado en la postcondición del servicio web ofrecido por un proveedor.

- ***Transformación 2***: la inclusión de la precondición del servicio web ofrecido por el proveedor en la precondición del servicio web solicitado por el cliente.

Para poder definir ambas transformaciones se parte de los 4 metamodelos que especifican las pre y postcondiciones de los clientes y proveedores de los SW, obtenidos automáticamente a partir de las especificaciones WSDL, traducidas a formato Ecore.

En el caso de la primera transformación, la ejecución toma como parámetros un modelo que representa la postcondición de un servicio solicitado por el cliente, y un modelo que representa la postcondición de un servicio ofrecido por un proveedor, ambos basados en los metamodelos correspondientes. Lo mismo ocurre con la segunda transformación, pero analizando las precondiciones de los servicios web.

Debido a que la herramienta MediniQVT aún no soporta la ejecución de transformaciones en modo sólo chequeo, la transformación se ejecuta en modo forzado, y si los modelos objetivos no cambian, significa que la contrastación de las pre y postcondiciones fue satisfactoria. Al contrario, si alguno de los modelos objetivos cambia, quiere decir que las pre o postcondiciones no son compatibles.



Figura 7. Detalle de los elementos appType, aditionalReq, Parameter, QoSReq y appFound en la Trasnformación XSLT.

7. Ejemplo de Aplicación

En esta sección se retoma el ejemplo de aplicación planteado en [8], referido al uso de SW seleccionados automáticamente en el marco de las tareas involucradas en el proceso de desarrollo de software denominado OpenUP/Basic [16]. Particularmente, se plantea el uso en la generación automática de código a partir de un diseño de un sistema.

La Figura 8 muestra la especificación WSDL de la postcondición del servicio web ofrecido por el proveedor. Esta especificación representa un modelo particular, basado en el metamodelo definido en la sección 5.

De igual manera se define la postcondición del servicio web solicitado, y ambas precondiciones.

La Figura 9 presenta la transformación de WSDL a formato Ecore, de la postcondición del servicio web ofrecido por el proveedor. De igual manera se realizan las transformaciones a formato Ecore de las demás pre y postcondiciones involucradas.



Figura 8. Modelo de la postcondición del SW ofrecido por el proveedor (especificación WSDL).

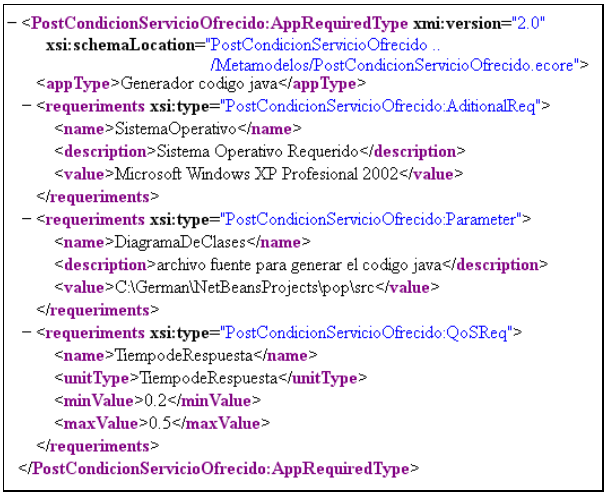


Figura 9. Resultado de la ejecución de la Transformación XSLT.

Una vez obtenidos los modelos que representan las postcondiciones del servicio solicitado por el cliente y la postcondición de los servicios ofrecidos por uno o más proveedores, se ejecuta la transformación definida para la contrastación de las postcondiciones. Como actualmente la herramienta MediniQVT no soporta la ejecución de transformaciones en modo sólo chequeo, y hasta que esta característica sea soportada por la herramienta, en este caso de estudio se ejecuta la transformación en modo forzado. Luego de la ejecución, se chequea el resultado obtenido (Figura 9) y si el modelo objetivo cambia, significa que la postcondición del servicio solicitado por

el cliente no está incluida en la postcondición del servicio ofrecido por el proveedor.

8. Conclusiones

En la actualidad la automatización de los procesos de negocio se ha convertido en un objetivo primordial para el crecimiento y desarrollo de las organizaciones, dado que produce una importante reducción de costos, incremento de eficiencia, menores oportunidades de cometer errores, y mejor control y calidad en beneficio de todos.

El uso de SW como parte de la Interfaz de las Aplicaciones Invocadas en los Sistemas de Gestión Workflow, provee importantes beneficios como la distribución transparente de las aplicaciones externas al workflow, permitiendo que las aplicaciones puedan cambiar su ubicación en la red sin que esto implique ninguna modificación en su invocación.

En este trabajo se incorporó una mejora al proceso de contrastación de SW definido en [8], la cual posibilita automatizar aún más dicho proceso, a través de la definición de transformaciones XSLT que permiten pasar de una especificación de SW en WSDL a una especificación de SW en formato Ecore, el cual es el formato utilizado por la herramienta de contrastación mediniQVT.

Con la transformación XSLT incorporada en este trabajo, se automatiza completamente el proceso de selección, haciendo más factible su implementación en los procesos de selección de SW por parte de clientes que lo solicitan.

La elaboración del caso de estudio muestra una aplicación de las transformaciones XSLT definidas, en particular la utilizada para el proceso de contrastación de la inclusión de la postcondición del servicio web solicitado, permitiendo de esta forma, automatizar la transformación de la especificación del SW en formato WSDL a formato Ecore (que en el trabajo previo era hecho de manera manual), y con esto automatizar completamente el proceso de contrastación para elegir el servicio web ofrecido que mejor se adapta a los requerimientos del cliente.

9. Referencias

[1] Workflow Management Coalition. *The Workflow Reference Model*. WfMC-TC00-1003. http://www.wfmc.org/standards/referencemodel.htm. Último acceso mayo 2014.

[2] Workflow Management Coalition. *Programming Interface 2&3 Specification.* WfMC-TC-1009.V2.0. http://www.wfmc.org/standards/publicdocuments.htm. Último acceso Diciembre 2013.

[3] World Wide Web Consortium. *Web Service Architecture*. http://www.w3.org/TR/ws-arch/. Último acceso Julio 2014.

[4] OASIS. UDDI Versión 3.0.2. http://uddi.org/pubs/uddi_v3.htm. Último acceso Abril 2014.

[5] Daniele M., Martellotto P., Riesco D. "Mejorando la Invocación de Aplicaciones desde un Workflow: Un caso de estudio". *XXXV Conferencia Latinoamericana de Informática - XXXV CLEI*. Pelotas, Brasil. 22 al 25/9 de 2009. Pág. 550 pp. ISBN 978-987-24611-0-2.

[6] Debnath N., Martellotto P., Daniele M., Riesco D., Montejano G.. Including non-functional requirements in the invocation of applications from the Workflow engine using Web Services. *19th International Conference on Software Engineering and Data Engineering - SEDE 2010*. San Francisco, CA USA. 16 al 18/6 de 2010. Páginas 117-120. ISBN 978-1-880843-77-2.

[7] Booch G., Rumbaugh J., Jacobson I., *The Unified Modeling Language.* Addison Wesley, Second Edition. 2005.

[8] Zorzan F., Martellotto P., Daniele M., Uva M., Arsaute A., Riesco D., Gonzalez A., Frutos M.. Refinamiento del Proceso de Invocación de Aplicaciones desde un Workflow por medio de QVT/Relations. *XXXVII Conferencia Latinoamericana de Informática - XXXVII CLEI*. Pontificia Universidad Católica del Ecuador. Quito, Ecuador. 10 al 14 de Octubre de 2011. Vol. I pp. 305-320. ISBN: 978-9942-9880-0-3.

[9] Object Management Group, "*Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*", OMG Document Number: formal/2008-04-03, Standard Document URL: http://www.omg.org/spec/QVT/1.0/PS/. Último acceso junio 20142.

[10] ikv++: medini QVT. http://www.ikv.de/,. Último acceso Junio 2014.

[11] Silberschatz, Korth, Sudarshan, "*Fundamentos de Base de Datos*". Quinta Edición. McGraw Hill Company, 2006.

[12] Object Management Group "*Object Constraint Language Version 2.0*". OMG Document Number: formal/06-05-01, Standard Document URL: http://www.omg.org/cgi-bin/doc?omg/03-06-01, Último acceso Julio 2014.

[13] Object Management Group "*Meta Object Facility (MOF) Core Specification" OMG Available Specification*. Versión 2.0. formal/06-01-01, http://www.omg.org/spec/MOF/2.0/PDF, Último acceso Mayo 2014.

[14] Eclipse Modeling Framework, URL: http://www.eclipse.org/modeling/emf/, último acceso Junio 2014.

[15] Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation 26 November 2008, URL: http://www.w3.org/TR/2008/REC-xml-20081126/. Último acceso mayo 2014.

[16] OpenUP/Basic. http://epf.eclipse.org/wikis/openupsp/index.htm Último acceso Julio 2014.

Estrategia Multipropósito de Medición y Evaluación Orientada a Cambio y Mejora Continua

Belén Rivera, Pablo Becker, Luis Olsina
Universidad Nacional de La Pampa, Facultad de Ingeniería,
General Pico, La Pampa, Argentina
[riveramb, beckerp, olsinal]@ing.unlpam.edu.ar

Abstract

Con las actividades de medición se obtienen datos cuantitativos a partir de métricas que cuantifican los atributos de un ente. Para obtener información de estos datos, el proceso de evaluación interpreta las medidas obtenidas usando indicadores. Con el fin de tomar decisiones respecto de los resultados obtenidos, es necesario brindar recomendaciones para planificar acciones. Si dichas recomendaciones implican cambiar para mejorar ciertos aspectos, será necesario volver a medir y evaluar para determinar si se alcanzó el nivel de satisfacción requerido. Frecuentemente, las actividades involucradas en procesos de cambio y mejora no están contempladas en estrategias de medición y evaluación. En este trabajo se presenta una estrategia de medición y evaluación que considera de manera integral, procesos de medición, evaluación, cambio y mejora. De esta manera, se robustece el proceso de medición y evaluación, dotándolo de características que conducen a la mejora continua.

Palabras Clave

Medición. Evaluación. Cambio. Mejora Continua Estrategias. GOCAME. SIQinU.

1. Introducción

Las organizaciones dedicadas al desarrollo de software deben promover programas de medición y evaluación (M&E) para gestionar la calidad de sus productos. Con el objetivo de asegurar la repetitividad y consistencia de los resultados, es necesario contar con un enfoque integral de M&E basado en al menos dos pilares, a saber, i) *un marco de modelado de calidad* [21], y ii) *estrategias de M&E* [6, 20].

El marco de modelado de calidad permite estructurar e instanciar modelos de calidad, dirigidos a diferentes categorías de entidad tales como recurso, proceso, producto, sistema, sistema en uso, servicio y proyecto. Además, es la base para la definición de los requerimientos no funcionales, estructurados en la forma de características, sub-características y atributos a ser evaluados.

Respecto a las estrategias integradas de M&E, estas deben satisfacer tres principios o capacidades simultáneamente [23], a saber: *un marco conceptual*, *una especificación (de las vistas) del proceso* y *una especificación de métodos*. El *marco conceptual* para el dominio de M&E tiene que estar construido sobre una base conceptual terminológicamente consistente (por ej. usando ontologías [17]), en donde se especifique de manera formal y explícita, los componentes, conceptos, relaciones y restricciones para el dominio en cuestión.

El segundo principio es la *especificación (de las vistas) del proceso*, el cual describe principalmente qué hacer. Un proceso puede ser especificado desde diferentes perspectivas o vistas [6], a saber: *Funcional*, en la cual se describen cuáles son las actividades planificadas y programadas junto a sus entradas y salidas; *de Comportamiento*, que especifica la dinámica del proceso mediante secuencias, paralelismos, iteraciones, etc.; *Organizacional*, que muestra qué roles intervienen en la realización de qué actividades; y *de Información*, que se centra en describir los productos de trabajo que se producen e insumen en las actividades. Un proceso de M&E especificado desde diferentes vistas facilita el entendimiento y la comunicación entre los interesados y además, asegura la repetitividad y reproducibilidad en la implementación de las actividades.

Por último, la *especificación de métodos* (y herramientas), permite llevar a cabo y automatizar, de manera parcial o total, las descripciones de las actividades. Los métodos son asignados de una forma flexible para realizar las actividades especificadas y pueden ser automatizados por herramientas.

Una estrategia de M&E conformada por un marco conceptual de M&E, la especificación del proceso de M&E y un soporte metodológico y tecnológico para realizar las actividades la denominamos *Estrategia*

Integrada de M&E [20, 23]. En este sentido, se han desarrollado en nuestras líneas de investigación, dos estrategias integradas de M&E: GOCAME *(Goal-Oriented Context-Aware Measurement and Evaluation)* [20, 21], y SIQinU (*Strategy for Understanding and Improving Quality in Use*) [15].

Básicamente, GOCAME es una estrategia multipropósito que sigue un enfoque orientado a objetivos, es sensible al contexto y está centrada en la necesidad de información de M&E en una organización. Puede ser usada para evaluar la calidad de entidades pertenecientes a diferentes categorías, tales como producto, sistema, sistema en uso, recursos, etc. Sin embargo, GOCAME no contempla acciones de cambio y de mejora de la entidad evaluada y/o de su contexto. Por lo tanto, esta estrategia no puede considerarse como una estrategia que permita cambios y mejoras conducidos por medición y evaluación.

Por otra parte, SIQinU es una estrategia de propósito específico que, a diferencia de GOCAME, sí permite introducir cambios a las entidades analizadas. La estrategia SIQinU se fundamenta en las relaciones existentes entre Calidad Externa y Calidad en Uso. Específicamente, la Calidad Externa "influye" a la Calidad en Uso y, a su vez, la Calidad en Uso "depende de" o está determinada por la Calidad Externa [10]. Teniendo presente estas relaciones, SIQinU busca mejorar, de manera incremental y continua, la Calidad en Uso de una aplicación a partir de la mejora de la Calidad Externa. Las acciones de cambio y mejora que contempla su proceso se inician a partir de los problemas detectados en la Calidad en Uso del sistema en uso. Por lo tanto, tampoco puede considerarse SIQinU como una estrategia multipropósito de medición, evaluación y cambio, porque no es instanciable para cualquier categoría de ente y foco de evaluación como la que ahora proponemos.

El presente trabajo tiene como objetivo definir una estrategia de M&E que sea multipropósito y que esté orientada a cambio y mejora, es decir, que además de permitir comprender la situación actual de cualquier ente a través de la medición y evaluación de sus atributos y características, analizar los resultados de la evaluación y dar recomendaciones, también permita continuar con un proceso de cambio que tienda a la mejora continua. La estrategia de medición, evaluación y cambio (ME&C) multipropósito que proponemos se denomina GOCAME^{+}.

En resumen, la contribución general de esta investigación consiste en: i) definir el proceso de ME&C para la nueva estrategia genérica y multipropósito GOCAME^{+}, presentando la vista funcional y de comportamiento de dicho proceso; y ii) ilustrar el proceso de ME&C propuesto con fragmentos de un caso de estudio de Usabilidad de Facebook realizado recientemente [22].

El artículo está organizado de la siguiente manera. En la Sección 2 se da un panorama de las estrategias GOCAME y SIQinU. Luego, en la Sección 3, se introduce el proceso de ME&C para GOCAME^{+}, que es la estrategia multipropósito orientada a cambio y mejora continua que se discute en este trabajo. En la sección 4 se presenta una prueba de concepto usando un caso de estudio ya realizado, con el fin de ilustrar el proceso de la nueva estrategia. En la Sección 5 se discute los trabajos relacionados a estrategias de M&E y a enfoques de gestión de la calidad y mejora continua. Finalmente, en la Sección 6, se esbozan las conclusiones y trabajos futuros.

2. Panorama de las Estrategias Integradas: GOCAME y SIQinU

GOCAME es una estrategia integrada a ser empleada en proyectos de M&E y contiene simultáneamente las tres capacidades previamente enunciadas, esto es, especificaciones de proceso, de método y marco conceptual.

El *marco conceptual* de GOCAME tiene su base terminológica definida a través de una ontología de M&E [17]. Esta ontología ha sido la base para la construcción del denominado marco conceptual C-INCAMI (*Contextual-Information Need, Concept Model, Attribute, Metric and Indicator*) [20], el cual permite la especificación de un modo robusto de metadatos y datos usados en las actividades y artefactos del proceso de M&E así como también favorece la instanciación de métodos y herramientas asignados a las actividades del proceso.

C-INCAMI está estructurado en seis componentes, a saber:

i. *de Proyecto*: permite especificar la información relativa a la gestión de los proyectos de M&E.
ii. *de Requerimientos No Funcionales*: permite especificar requerimientos no funcionales de alguna entidad para un propósito y un punto de vista particular. Los requerimientos no funcionales son representados por un modelo de concepto (por ejemplo, modelo de Calidad Externa) que incluye conceptos calculables (características), sub-conceptos y atributos. Los atributos son las propiedades medibles de la entidad bajo análisis.
iii. *de Contexto*: permite especificar el contexto del ente objetivo a ser evaluado mediante atributos o propiedades de contexto que sean relevantes a su situación.
iv. *de Medición*: permite que se especifiquen las métricas que cuantifican los atributos. Como resultado de una actividad de medición se obtienen medidas base y derivadas según sea directa o indirecta la medición.

v. *de Evaluación*: permite que se definan los indicadores elementales y derivados para interpretar los valores de los atributos y conceptos calculables del árbol de requerimientos no funcionales.
vi. *de Análisis y Recomendación*: permite que los datos e información obtenidos como resultado de la medición y evaluación sean analizados para dar recomendaciones.

Cabe acotar que en un trabajo reciente, los términos de los componentes *de Medición* y *de Evaluación* se han enriquecido semánticamente con estereotipos derivados de la ontología genérica de proceso desarrollada en [7].

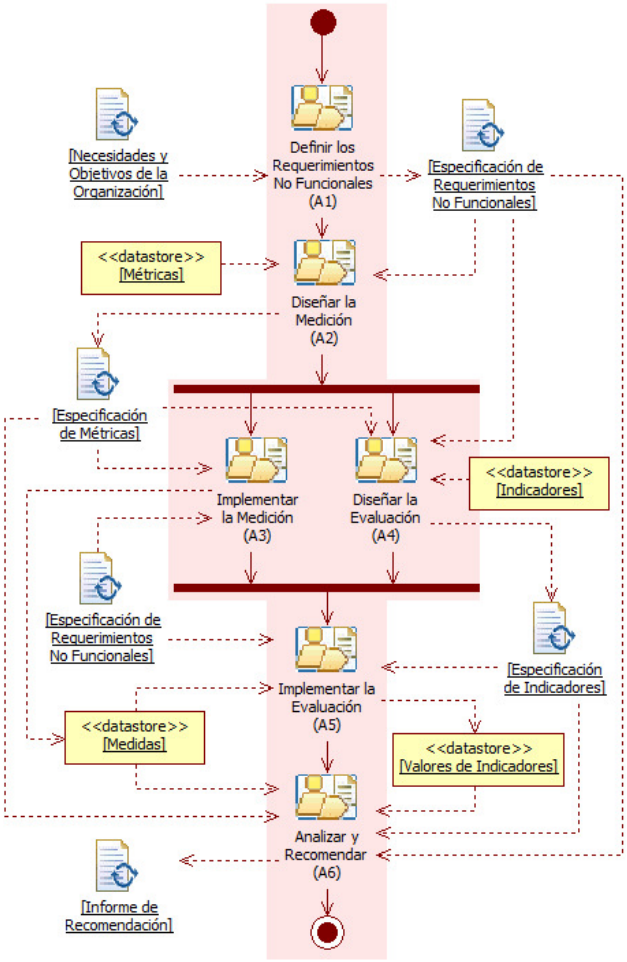


Figura 1. Vista funcional y de comportamiento del proceso de M&E en GOCAME usando el lenguaje de especificación SPEM

Respecto a la segunda capacidad de una estrategia integrada, GOCAME tiene definido su *proceso de M&E* de manera tal de poder asegurar repetitividad en la ejecución de las actividades y consistencia en los resultados. Su proceso de M&E consta de seis actividades principales, a saber:

i. *Definir los Requerimientos no Funcionales*
ii. *Diseñar la Medición*
iii. *Implementar la Medición*
iv. *Diseñar la Evaluación*
v. *Implementar la Evaluación*
vi. *Analizar y Recomendar*

La Figura 1 ilustra el proceso de GOCAME especificando mediante SPEM las vistas funcional y de comportamiento.

Por último, respecto a *métodos y herramientas,* GOCAME permite por ejemplo la especificación de métricas –es decir, una métrica como la especificación del método para realizar la actividad de medición- y de indicadores –como métodos utilizados durante la evaluación elemental y derivada (ver en detalle especificaciones de métricas e indicadores en [18]).

Por otra parte, SIQinU es una estrategia integrada para medir y evaluar Calidad en Uso de una aplicación web. Su propósito es detectar problemas provenientes en el uso de una aplicación, con usuarios reales ejecutando tareas habituales y, en base a ello, implementar acciones de mejora para solucionarlos. Los problemas encontrados usando la aplicación son vinculados con características y atributos medibles de Calidad Externa relacionados al ente evaluado. La aplicación se vuelve a medir y evaluar pero esta vez en consideración de características de Calidad Externa. Luego de la evaluación, se dan recomendaciones de mejora para aquellos indicadores elementales (atributos) con un bajo nivel de aceptabilidad. A partir de estas recomendaciones y mediante distintas técnicas, se llevan a cabo los cambios en la aplicación, generándose así una nueva versión –mejorada- de la aplicación. Esta nueva versión se vuelve a medir y evaluar en consideración del sistema en uso, para determinar si las mejoras realizadas en los atributos de Calidad Externa, mejoraron la calidad del sistema en uso cuando el usuario ejecuta las mismas tareas.

El proceso de SIQinU, mostrado en la Figura 2, comprende 6 fases. Notar que en dicha Figura, los nombres de las fases del proceso de SIQinU, se presentan en inglés tal cual aparecen en [15]. Las fases son:

Fase I (Ph I en la Figura): Especificar Requerimientos y Criterios de Evaluación para Calidad en Uso (*Specify Requirements and Evaluation Criteria for QinU*);

Fase II (Ph II): Ejecutar Evaluación de Calidad en Uso y Análisis (*Perform QinU Evaluation and Analysis*);

Fase III (Ph III): Derivar y Especificar Requerimientos y Criterios de Evaluación para Calidad Externa (*Derive/Specify Requirements and Evaluation Criteria for EQ*);

Fase IV (Ph IV): Ejecutar Evaluación para Calidad Externa y Proveer Análisis (*Perform EQ Evaluation and Analysis*);

Fase V (Ph V): Recomendar, Ejecutar Acciones de Mejora y Re-evaluar para Calidad Externa (*Recommend and Perform Improvement Actions for EQ*); y

Fase VI (Ph VI): Re-evaluar Calidad en Uso y Analizar Acciones de Mejora (*Re-evaluate Quality in Use and Analyze Improvement Actions*).

SIQinU es una estrategia derivada a partir de GOCAME. Por lo tanto, SIQinU reutiliza el marco C-INCAMI, así como también muchas de las actividades y métodos de GOCAME. Por ejemplo, las actividades A1, A2, A4 y A6 (según Figura 1) de GOCAME están incluidas en las fases I y III de SIQinU. A su vez, las actividades A3 y A5 están incluidas en las fases II y IV. Sin embargo, SIQinU cuenta con actividades específicas que no están incluidas en GOCAME. Por ejemplo, en la fase V hay actividades que permiten introducir cambios (mejoras) en la aplicación. Además, SIQinU define un proceso diferente (comparar Figuras 1 y 2), el cual incluye ciclos de cambio y de re-evaluación para conocer el nivel de mejora introducido. De igual manera, en SIQinU se utilizan métodos específicos, por ejemplo de cambio, no usados en GOCAME.

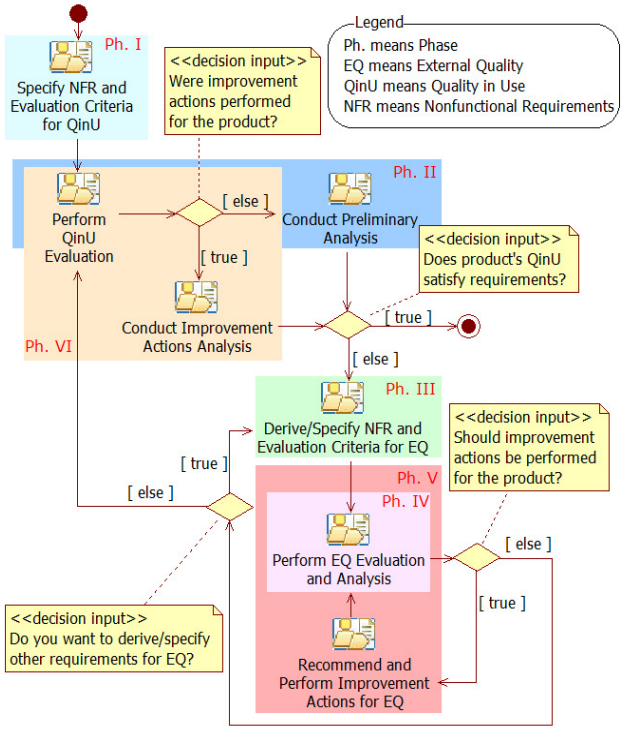


Figura 2. Vista funcional y de comportamiento del proceso de M&E en SIQinU. Notar que una fase agrupa procesos y actividades relacionadas

SIQinU ha sido diseñada específicamente con el propósito de "comprender" el nivel de Calidad en Uso de una aplicación web y "mejorarlo" a partir de cambios en la aplicación (no en la aplicación en uso). Por el contrario, GOCAME es una estrategia multipropósito, ya que se utiliza para "comprender", "predecir", "controlar", etc. diferentes focos/entes, como pueden ser Calidad Interna y Externa, Calidad en Uso, Calidad de Recurso, etc.; sin embargo, no considera ciclos de cambio y mejora.

La capacidad de GOCAME de poder ser utilizada para distintos propósitos, junto a la capacidad de SIQinU de considerar actividades de cambio como parte de su proceso de M&E son las dos características que ahora se contemplan de manera conjunta e integral para dar surgimiento a GOCAME$^+$.

3. GOCAME$^+$: Estrategia Multipropósito y Genérica orientada a Cambio y Mejora Continua

Toda organización que desee ser exitosa debe apuntar a la mejora continua de la calidad de sus productos/ servicios, procesos y recursos. Para ello, es preciso no sólo contar con actividades de M&E que permitan analizar y detectar debilidades y vulnerabilidades, sino también es necesario contar con actividades que permitan introducir cambios orientados a la mejora.

Para este fin surge la estrategia GOCAME$^+$, la cual es una estrategia multipropósito y genérica de ME&C que considera actividades de cambio y mejora continua. Al igual que las estrategias GOCAME y SIQinU introducidas en la Sección anterior, GOCAME$^+$ posee los 3 capacidades descriptas previamente. Este trabajo se centra en especificar las principales actividades del proceso de la nueva estrategia, mientras que la base conceptual de cambio será parte de trabajos futuros dado que se debe ampliar la actual base conceptual C-INCAMI con términos del dominio de cambio. El proceso genérico de ME&C de GOCAME$^+$ se presenta en la Figura 3, y consta de 8 actividades principales, a saber:

i. *Definir los Requerimientos no Funcionales*
ii. *Diseñar la Medición*
iii. *Implementar la Medición*
iv. *Diseñar la Evaluación*
v. *Implementar la Evaluación*
vi. *Analizar y Recomendar*
vii. *Diseñar las Acciones de Cambio*
viii. *Implementar las Acciones de Cambio*

GOCAME$^+$ incluye de algún modo el ciclo Planificar, Hacer, Verificar y Actuar (*Plan-Do-Check-Act* o PDCA) [9], que representa un enfoque básico para mejora continua donde se planifican acciones de mejora, se las lleva a cabo, luego se verifican (evalúan) los resultados obtenidos y por último se actúa sobre los problemas hallados, lo que puede implicar un nuevo ciclo PDCA.

La Figura 3 presenta el proceso general de GOCAME$^+$

distinguiendo con el sombreado, por un lado, las actividades de medición y evaluación y, por otro, las de cambio y mejora. El proceso comienza con la actividad Definir los Requerimientos No Funcionales (A1). Esta actividad permite establecer la necesidad de información, es decir, definir el propósito de la evaluación, el tipo de usuario, la entidad y el concepto foco a evaluar; especificar el contexto de la evaluación; y definir un árbol de requerimientos que tendrá como raíz al concepto foco, como nodos intermedios las (sub-)características relacionadas al foco, y como nodos hojas los atributos medibles asociados a la entidad seleccionada. Como salida de A1 se produce un documento denominado Especificación de los Requerimientos no Funcionales.

Una vez definidos los requerimientos no funcionales se procede a Diseñar la Medición (A2), cuyo propósito es el de identificar las métricas que se utilizarán para cuantificar los atributos definidos en el árbol de requerimientos. Las métricas, y sus metadatos asociados, son recuperadas desde un repositorio (identificado en la Figura 3 con el estereotipo *datastore*), las cuales han pasado por un proceso de discusión y consenso entre expertos [1]. La salida de esta actividad es el artefacto Especificación de Métricas. A continuación se realiza, en forma paralela o secuencial, sin importar el orden, el diseño de la evaluación y la implementación de la medición. La actividad Implementar la Medición (A3) consiste en utilizar las métricas seleccionadas en la actividad A2 para cuantificar los atributos sobre ente concreto. Si corresponde, se utilizan herramientas que automatizan los procedimientos de medición o cálculo. Como resultado se obtienen los valores (medidas) asociados a cada uno de los atributos que caracterizan a las entidades bajo análisis, los que se almacenan en un repositorio para su posterior análisis.

La actividad Diseñar la Evaluación (A4) implica definir, para cada atributo y concepto calculable del árbol de requerimientos, un indicador que lo evaluará. Para los atributos se definen indicadores elementales que especifican la manera de realizar el mapeo del valor medido del atributo a un valor que indique el nivel de satisfacción alcanzado. Luego, se deben identificar indicadores derivados para calcular el nivel de satisfacción alcanzado por los conceptos calculables. Como resultado de A4 se obtiene el artefacto Especificación de Indicadores.

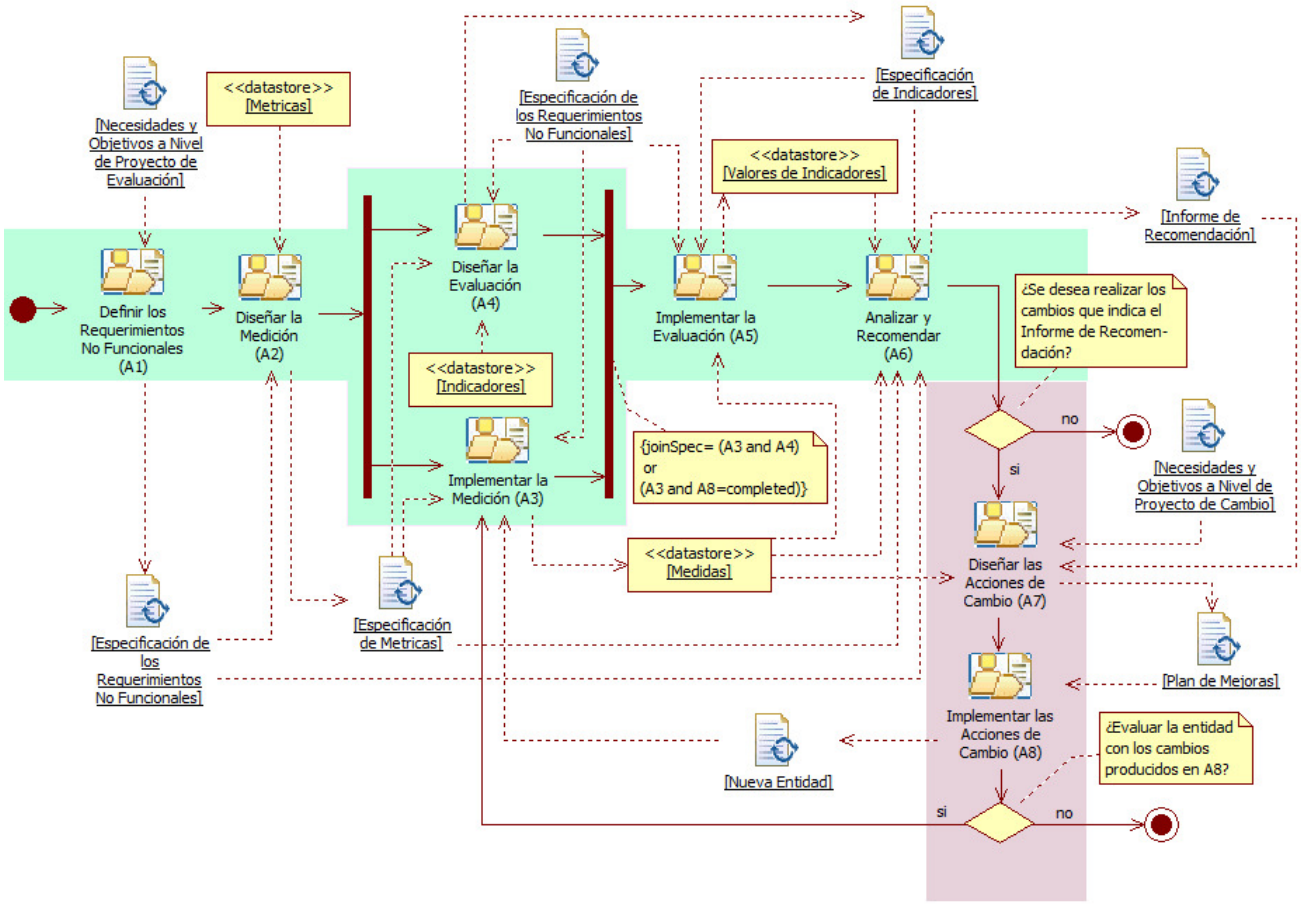


Figura 3: Vista funcional y de comportamiento del proceso de ME&C en GOCAME+

Finalizadas las actividades A3 y A4 se continúa con la actividad Implementar la Evaluación (A5). El objetivo de esta actividad es obtener los valores de los indicadores definidos en la actividad A4 con el fin de conocer el grado de satisfacción de cada uno de los requerimientos establecidos en A1 conforme a la necesidad de información. Primero se calculan los indicadores elementales para lo cual se realiza un mapeo de los valores obtenidos en la medición a un conjunto de valores, por lo general normalizados. Después se pueden calcular los indicadores derivados. Al finalizar esta actividad se obtienen los valores de los diferentes indicadores, tanto elementales como derivados, que servirán como entrada en la próxima actividad, Analizar y Recomendar (A6).

La actividad A6 permite realizar un análisis de los valores de indicadores obtenidos y en base a éstos, recomendar cambios a realizar a fin de lograr mejoras en la calidad de la entidad o en su contexto. La actividad requiere como insumos todos los artefactos y resultados producidos en las actividades anteriores, a saber: especificación de requerimientos no-funcionales, especificación de métricas, especificación de indicadores, valores medidos y valores de indicador calculados. Como resultado de esta actividad se obtiene el documento Informe de Recomendación, donde se detallan recomendaciones para aquellos atributos que no satisfacen cierto nivel de aceptabilidad. Las recomendaciones proporcionan sugerencias y prioridades para los atributos que deben mejorarse. En base al Informe de Recomendación, se debe analizar si se desea mejorar el nivel de satisfacción de los atributos cuyo valor de indicador obtuvo un bajo nivel de aceptabilidad o desempeño. En caso afirmativo, se lleva a cabo la actividad Diseñar las Acciones de Cambio (A7).

En la actividad A7, considerando el Informe de Recomendación y los objetivos de cambio, se produce el Plan de Mejoras en el cual se indican las acciones para cada recomendación y además se establece cómo se realizarán dichas acciones en pos de concretar los cambios. El cómo, indica cada uno de los métodos y/o técnicas que se emplearán para hacer efectivas las acciones de cambio. Los métodos y técnicas para cambiar la entidad y/o su contexto van desde reconfiguraciones parametrizadas, reprogramación del código y refactoring de interface (como se hizo en [19]), hasta el rediseño arquitectural. El Plan de Mejoras desde luego constituye un instrumento para el control y seguimiento de las diferentes acciones de cambio a desarrollar de manera organizada, priorizada y planificada.

Una vez culminada la actividad A7, Implementar las Acciones de Cambio (A8) es la próxima actividad. En base al documento Plan de Mejoras, se ejecutan los cambios correspondientes en el ente (y/o su contexto) evaluado, originando como salida una nueva versión de la entidad evaluada.

Una vez efectuados los cambios, es deseable medir y evaluar la nueva versión del ente para comprender la ganancia lograda en la mejora. Por lo tanto, luego de la actividad A8, se vuelven a realizar las actividades A3, A5 y A6, tal como se muestra en la Figura 3. Notar que en esta nueva M&E las actividades A1, A2 y A4 no se vuelven a realizar puesto que se miden y evalúan los atributos y conceptos calculables ya especificados utilizando las métricas e indicadores definidos inicialmente. De esta manera los resultados obtenidos en la nueva versión pueden ser comparados con la versión original. Al igual que se especificó anteriormente, al finalizar la actividad de Implementar la Evaluación, en la actividad Analizar y Recomendar se genera un nuevo documento de Informe de Recomendación, esta vez, para una versión del ente en consideración de los cambios sugeridos en el documento inicial. Finalizada entonces la re-evaluación y con el nuevo Informe de Recomendación, se determina cuáles atributos han aumentado su nivel de satisfacción gracias a las mejoras introducidas, y cuáles no, puesto que puede suceder que cambios en algunos atributos se traduzcan en un detrimento del nivel de satisfacción de otros que estaban en niveles óptimos. Si los cambios realizados no satisfacen, se vuelve a recomendar acciones de cambio, tendientes a mejorar otros atributos, a través de un nuevo documento Plan de Mejoras.

El proceso continúa cíclicamente hasta lograr una versión del ente (y su contexto) evaluado donde el nivel de satisfacción alcanzado sea el aceptable según lo establecido para dicho proyecto de ME&C.

4. Prueba de Concepto

Recientemente, hemos realizado un estudio de evaluación de Usabilidad y experiencia de usuario de la aplicación Facebook para dispositivos móviles con el fin de ayudar a los diseñadores a detectar problemas de Usabilidad en la aplicación y producir mejores soluciones de diseño [22]. En esta sección usaremos fragmentos del caso de estudio citado para mostrar las recomendaciones y planificar acciones de cambio que surgen luego de la evaluación, con el fin de ilustrar el proceso descripto para GOCAME^{+}, instanciado para el foco Calidad de Sistema.

La actividad A1 produce el documento Especificación de los Requerimientos No Funcionales, el cual se compone de las especificaciones de la necesidad de información y del árbol de requerimientos. En la Figura 4 se presenta la especificación de la necesidad de información de M&E generada y, en la 1^{era} columna de la Tabla 1 se aprecia, solo por razones de espacio, un extracto del árbol de requerimientos en el que se resaltan en itálica los atributos definidos para *Usability*, en particular para *Operability* de la aplicación Facebook.

Especificación de la Necesidad de Información

Propósito: Comprender
Punto de vista del usuario: Usuario final

Categoría de Entidad bajo análisis: Aplicación de Red Social
Entidad Concreta: Aplicación móvil Facebook, versión 3.8 para Android

Foco de la Evaluación: Usability
Conceptos Calculables (Sub-características): Understandability, Learnability, Operability, User Error Protection, y User Interface Aesthetics

Figura 4. Especificación de la Necesidad de Información de M&E

Para la actividad Diseñar la Medición (A2), se seleccionaron todas las métricas que iban a cuantificar a todos los atributos del árbol de requerimientos. Por ejemplo, el nombre de la métrica que cuantifica al atributo *Permanence of main controls* (Permanencia de controles principales) para la sub-característica *Permanence of controls,* es Porcentaje de Permanencia de Controles Principales cuyo objetivo es determinar el porcentaje de botones o controles, en la barra de controles principales, que se mantienen consistentemente en el conjunto apropiado de pantallas. (Para ver detalles de la especificación de esta métrica referirse a [22]).

Tabla 1. Fragmento del Árbol de Requerimientos (*EI* significa Indicador Elemental y *DI* significa Indicador Derivado)

	EI	DI
1 Usability		**60.5** 🟡
1.1 Understandability		**76.1** 🟡
1.2 Learnability		**59.7** 🔴
1.3 Operability		**80.7** 🟢
1.3.1 Data Entry Ease		90 🟢
1.3.1.1 Defaults	100 🟢	
1.3.1.2 Mandatory entry	50 🔴	
1.3.1.3 Widget appropriateness	100 🟢	
1.3.2 Visibility (Optical Legibility)		81.5 🟢
1.3.2.1 Color visibility appropriateness		100 🟢
1.3.2.1.1 Brightness difference appropriateness	100 🟢	
1.3.2.1.2 Color difference appropriateness	100 🟢	
1.3.2.2 GUI object size appropriateness		63 🟡
1.3.2.2.1 Control (widget) size appropriateness	100 🟢	
1.3.2.2.2 Text size appropriateness	42.1 🔴	
1.3.3 Consistency		75.5 🟡
1.3.3.1 Permanence of controls		57.3 🔴
1.3.3.1.1 Permanence of main controls	54.9 🔴	
1.3.3.1.2 Permanence of contextual controls	67.4 🟡	
1.3.3.2 Stability of controls	95.5 🟢	
1.4 User Error Protection		**8.4** 🔴
1.5 UI Aesthetics		**80.8** 🟢

Luego, la actividad Implementar la Medición (A3) permitió obtener los valores medidos de los atributos para la aplicación móvil Facebook. Para la métrica mencionada anteriormente, el resultado de la medición obtenida fue 54.9%. Es decir, alrededor del 55% de los controles o botones principales, mantienen su posición de manera consistente dentro de la barra de controles principales (ver Figura 5) para cualquier pantalla del conjunto de pantallas evaluadas de la aplicación. El registro de las mediciones (fecha, valor medido, responsable, ID de la métrica y atributo, etc.) es almacenado en un repositorio (ver objeto Medidas con estereotipo *datastore* en la Figura 3).

En cuanto a la actividad Diseñar la Evaluación (A4), se diseñaron indicadores elementales para interpretar la medida de cada atributo así como indicadores derivados con el fin de conocer el nivel de satisfacción alcanzado por cada (sub-)característica, como por ejemplo *Usability*, *Operability*, entre otros.



Figura 5. Dos pantallas de la aplicación móvil Facebook: se señala la barra de controles principales y la inconsistencia entre la ubicación del control contextual y el de chat

A partir de las medidas obtenidas en A3 y usando los indicadores producidos en A4 se llevó a cabo la actividad Implementar la Evaluación (A5), obteniendo los valores de indicadores elementales que se visualizan en la 2da columna de la Tabla 1, y de indicadores derivados, en la 3ra columna. En la Tabla 1 se utilizan distintos colores para identificar el grado de satisfacción alcanzado por cada atributo/sub-característica, siendo el rango de aceptabilidad considerado el siguiente: un valor de 0 a 60 representa un nivel no satisfactorio (semáforo en color rojo) e indica que de manera urgente se deben realizar acciones de cambio; valores comprendidos en el rango de 60-80 representan un nivel marginal (color amarillo) e

indica necesidad de acciones de cambio, y un valor en el rango de 80-100 representa un nivel satisfactorio (color verde) e indica un nivel óptimo, lo que no requiere recomendación de cambio.

Como se observa en la Tabla 1, tres atributos han obtenido un nivel de aceptabilidad no satisfactorio (rojo), a saber: *Mandatory entry (1.3.1.2), Text size appropriateness (1.3.2.2.2)* y *Permanence of main controls (1.3.3.1.1)*. Por lo tanto, se debe brindar recomendaciones y determinar acciones de cambio con alta prioridad para los mismos. Por otro lado, el atributo *Permanence of contextual controls (1.3.3.1.2)* obtuvo un valor de aceptabilidad marginal (amarillo), y por ende, también se requeriría realizar acciones de cambio.

La Figura 5 muestra dos pantallas de la aplicación en las que se evidencian cómo la barra de controles principales no mantiene los mismos controles (botón de chat, por ejemplo) para ambas pantallas, y además, el control contextual tampoco es consistente en su ubicación. Por lo tanto, el botón principal de chat permanece en la pantalla de la derecha como parte de la barra de controles principales, mientras que en la de la izquierda no se observa el control. (Notar que la barra de controles principales debía permanecer en 35 pantallas de la aplicación Facebook versión 3.8, conteniendo siempre a los 5 controles principales)

En la actividad Analizar y Recomendar (A6), el agente en cumplimiento del rol *Recommender* genera el documento Informe de Recomendación (ver Tabla 2). En este informe se especifican los atributos cuyos valores de indicadores no han alcanzado un nivel de aceptabilidad satisfactorio. Además, por cada uno de estos atributos se sugieren una o más recomendaciones que permitan luego llevar a cabo acciones que aumenten el nivel de satisfacción del atributo. El informe especifica también la prioridad del cambio, es decir, cuales recomendaciones deben ser consideradas con carácter de urgente (A, acción con prioridad alta) y cuales con carácter de urgencia moderada (M, prioridad moderada, o que se debería actuar). En la Tabla 2 se presentan las recomendaciones sugeridas para los atributos con un nivel de satisfacción inferior al óptimo (menos del 80%) y se señala la prioridad establecida (A ó M) para planificar y/o realizar las acciones.

En base al Informe de Recomendación, el agente en cumplimiento del rol *Maintenance Project Manager* produce en la actividad A7 el documento Plan de Mejoras que indica para cada recomendación documentada cuales son las acciones de cambio planificadas, las fuentes de los registros de medición que dan soporte al cambio, y cómo pueden llevarse a cabo en la aplicación. El "cómo" implica que para cada acción se debe enumerar el método y/o técnica que se empleará para hacer efectiva dicha acción de cambio en la actividad Implementar las Acciones de Mejora (A8).

A fin de ilustrar el contenido de un posible Plan de Mejoras para el caso Facebook, en la Tabla 3 se indican diferentes acciones de cambio (AC) a llevarse a cabo.

En base al documento Plan de Mejoras, el agente en cumplimiento del rol *Developer* realiza los cambios correspondientes en la aplicación, produciendo así una nueva versión de la misma. Las tareas involucradas en la actividad Implementar las Acciones de Cambio están relacionadas a las diferentes acciones sugeridas en el Plan de Mejoras y los métodos y/o técnicas a utilizar. Por ejemplo, para el atributo *Mandatory entry* (1.3.1.2), según las acciones de cambio AC3.1 y AC3.2 de la Tabla 3, se deben realizar tareas relacionadas a *programación* y a *refactoring de GUI*, lo cual requiere algún nivel de acceso al código fuente de la aplicación. Cabe decir, los cambios no han podido realizarse por no tener acceso al código fuente, diseños y objetos GUI de la aplicación Facebook por tratarse de un producto propietario. Es preciso aclarar que el estudio no fue encomendado por la empresa que gestiona Facebook, por lo tanto las acciones de cambio recomendadas están sujetas sólo a los requerimientos no funcionales establecidos en el estudio.

Una vez que los cambios son ejecutados, la nueva versión de la aplicación puede ser re-evaluada, realizando nuevamente las actividades A3, A5 y A6 de GOCAME^{+}, tal como especifica el flujo de actividades de la Figura 3.

La Figura 6 ilustra algunos de los resultados que se obtendrían al finalizar la actividad A8, luego de aplicar las acciones de cambio descriptas en la Tabla 3 sobre las pantallas de la Figura 5. Los resultados de la re-evaluación permiten determinar, en relación con los resultados de la evaluación previa, cuáles atributos han aumentado su nivel de satisfacción gracias a las mejoras introducidas y cuáles no.



Figura 6. Vista de dos pantallas de la aplicación móvil Facebook luego de implementar las acciones de cambio AC1.1 y AC2.1 según Tabla 3

Tabla 2. Extracto del artefacto "Informe de Recomendación" producido en la actividad Analizar y Recomendar. (Nota: el acrónimo A significa Alta, es decir, se recomienda actuar de un modo urgente. M significa Moderada, e implica que se debería actuar)

ID	Recomendación (R)	Atributo	Prioridad
R1	**1.** Asegurar que en el conjunto de pantallas seleccionadas, el tipo de letra sea el apropiado para una correcta legibilidad. **2.** Asegurar que en el conjunto de pantallas seleccionadas, el tamaño de la letra sea el apropiado para una correcta visibilidad y legibilidad.	*Text size appropriateness* (1.3.2.2.2)	A
R2	**1.** Asegurar que en el conjunto de pantallas seleccionadas en los controles de entrada de los formularios cuyo ingreso de datos sea obligatorio tengan el soporte o indicador adecuado.	*Mandatory entry* (1.3.1.2)	A
R3	**1.** Asegurar que en la barra de controles principales todos los botones (controles) principales permanezcan consistentemente, en el conjunto de pantallas seleccionadas.	*Permanence of main controls* (1.3.3.1.1)	A
R4	**1.** Asegurar que los controles contextuales permanezcan consistentemente en el conjunto de pantallas seleccionadas y que sean apropiadas al contexto.	*Permanence of contextual controls* (1.3.3.1.2)	M

Tabla 3. Extracto del artefacto "Plan de Mejoras" producido en la actividad Analizar y Recomendar

ID	Acción de Cambio (AC)	Fuente para la AC	Atributo	Método / Técnica
AC1	**1.** Agregar, consistentemente, los controles principales en la barra de controles principales de las pantallas con la debilidad detectada.	El ID de cada pantalla/botón principal con la debilidad detectada consta en el registro de Medidas/Mediciones para 1.3.3.1.1.	*Permanence of main controls* (1.3.3.1.1)	Refactoring de GUI
AC2	**1.** Agregar, consistentemente, los controles (botones) contextuales en las pantallas con la debilidad detectada.	El ID de cada pantalla/botón contextual con la debilidad detectada consta en el registro de Medidas/Mediciones para 1.3.3.1.2.	*Permanence of contextual controls* (1.3.3.1.2)	Refactoring de GUI
AC3	**1.** Agregar una función que verifique la existencia de datos de entrada en los campos obligatorios en los formularios que lo requieran. **2.** Agregar el indicador visual "*" en los campos de entrada obligatoria en los formularios que lo requieran.	El ID de cada formulario/ campo de entrada obligatoria consta en el registro de Medidas/Mediciones para 1.3.1.2	*Mandatory entry* (1.3.1.2)	Programación Refactoring de GUI

5. Trabajos Relacionados y Discusión

Los estudios recientes y literatura referidos a enfoques y estrategias de medición y evaluación es variada, abarcando tanto propuestas a nivel académico como a nivel industrial. Se han desarrollado en este sentido tanto enfoques estructurados de medición orientados a objetivos como a información, para guiar la recolección de datos y el posterior análisis de los mismos. Existen por otra parte metodologías y estándares internacionales que se orientan a la mejora continua de los procesos intervinientes en cualquier organización.

En esta Sección analizaremos resumidamente a dos estrategias de medición que consideramos ampliamente difundidas e implementadas analizando en ambas los tres pilares necesarios en una estrategia integrada, como así también discutiremos un enfoque orientado a la mejora continua, sin dejar de mencionar los estándares internacionales al respecto, como son CMMI y algunas normas ISO/IEC.

Uno de los enfoques de medición ampliamente aceptado en la industria que ayuda a la gestión de proyectos de desarrollo de software es *Practical Software Measurement* (PSM) [16]. Constituye un enfoque orientado a la información que describe un proceso de medición para software, y es parte de un programa integral de gestión y administración de proyectos de desarrollo de software. Describe cómo definir e integrar requerimientos de medición, recolectar y analizar los datos obtenidos en la medición e implementar la totalidad del proceso de medición en una organización. PSM sirvió como base para el desarrollo del estándar ISO/IEC 15939

[11] y, tras la aparición formal de dicho documento, PSM fue actualizado conforme al estándar.

De los tres pilares mencionados en la Introducción como necesarios en una estrategia integrada de M&E, PSM satisface los tres. Esto es, posee una base conceptual para relacionar los diferentes conceptos de la terminología de medición –a través de un Modelo de Información de Medición-; define un proceso de medición con actividades y tareas –mediante un Modelo de Procesos de Medición; y utiliza también métodos y herramientas para dar soporte del proceso. El proceso de medición que especifica contempla las siguientes actividades: Establecer y mantener el compromiso de medición (*Establish & Sustain Measurement Commitment)*, Planificar la medición *(Plan the Measurement Process),* Ejecutar la Medición *(Perform the Measurement Process),* Evaluar la medición *(Evaluate Measurement)* [11, 16]. Este proceso está basado en la secuencia de gestión para la mejora continua PDCA, que hemos señalado en la Sección 3.

En este sentido, el proceso de medición en PSM es un proceso iterativo en el cual las medidas son refinadas a la vez que las necesidades de información van cambiando y la organización implementa acciones de cambio. Sin embargo, en PSM no se describe un proceso de evaluación que, a través de indicadores, interprete las medidas obtenidas y un proceso de cambio orientado a mejorar la entidad medida y su contexto, a partir del análisis obtenido en la interpretación de la medición realizada.

Otro de los enfoques de medición más usados a nivel académico e industrial es *Goal Question Metric* (GQM) [4]. GQM guía a las organizaciones en la definición de objetivos (*goal*), en la especificación de datos a recolectar por medio de métricas y su ulterior análisis e interpretación. Usa implícitamente modelos que sirven a fines de interpretar los resultados de los datos según los objetivos definidos, en el contexto específico. La evolución de GQM dio origen a otra estrategia, de los mismos autores, denominada *GQM*$^+$*Strategies* [5], la cual constituye también un enfoque de medición pero que a diferencia de *GQM*, permite alinear objetivos a distintos niveles de una organización. De esta manera, provee un mecanismo para alinear objetivos de medición con objetivos estratégicos o de niveles superiores. Ambos enfoques satisfacen los 3 pilares que hemos enunciando para una estrategia integrada de M&E: tienen una base terminológica estructurada en un glosario en la que se definen los principales términos involucrados; definen explícitamente el proceso y los métodos y herramientas para llevar a cabo las actividades del proceso.

En [23] se presenta un estudio comparativo de estrategias integradas de M&E en el cual fueron evaluadas GOCAME y *GQM*$^+$*Strategies*. Dicho estudio permitió observar el nivel de satisfacción alcanzado por cada una de las estrategias analizadas teniendo en cuenta las tres capacidades de una estrategia integrada. Si bien ambas estrategia*s* satisfacen en gran medida las tres capacidades, no contemplan actividades de cambio tendientes a la mejora continua de un modo explícito, como se observa en el caso de GOCAME$^+$.

Otro trabajo relacionado es el Paradigma de Mejora de la Calidad, *Quality Improvement Paradigm* (QIP) [3], cuya premisa es que la mejora es un proceso continuo. Este enfoque ayuda a las organizaciones a poner en práctica un proceso de mejora continua tomando en consideración experiencias pasadas. QIP resulta beneficioso en organizaciones maduras, las cuales son conscientes de sus procesos y aprenden de experiencias pasadas. QIP utiliza GQM para la definición de metas y las métricas adecuadas que guiarán la ejecución del proceso y una herramienta denominada Factoría de Experiencias (*Experience Factory Organization*) [2]. Factoría de Experiencias es una estructura organizacional que da soporte a la reutilización de experiencias pasadas (consideradas aprendizaje) en una forma que pueda ser efectivamente entendida y modificada (a través de modelos), almacenadas en un repositorio de modelos de experiencias integradas (base de experiencias) que puedan ser accedidas/modificadas para satisfacer las necesidades del actual proyecto (reutilización). Si bien este paradigma está afianzado como un enfoque de mejora continua, no guía programas de medición, evaluación y cambio de manera integrada y robusta.

Por otra parte, existen normas ISO relacionadas a procesos de medición, evaluación y definición de requerimientos, entre otros. El estándar ISO 15939 [11] define un proceso de medición, el cual describe las actividades y tareas necesarias para definir, aplicar y mejorar la medición dentro de un proyecto en particular o dentro de la estructura de medición organizacional. El estándar ISO 15939 también provee definiciones de los términos de medición usados comúnmente dentro de la industria del software. Principalmente, se enfoca en los conceptos del proceso de medición, aunque también cubre algunos otros referidos a medidas y objetivos.

Otra norma ISO que es preciso mencionar es el estándar 14598 [12] referido al proceso de evaluación, el cual consta de seis documentos que proporcionan un marco de trabajo para evaluar la calidad de todos los tipos de productos de software e indica los requisitos para los métodos de medición y evaluación del proceso proporcionando además, una guía de los términos involucrados en evaluación. Por último, el estándar ISO 9126 [13] y su posterior actualización, el estándar 25010 [10], están referidos a la especificación de requerimientos a través de modelos de calidad. El estándar 25010 actualmente es parte de otros documentos que integran al proyecto SQuaRE (*System and Software Quality Requirements and Evaluation*).

Estos estándares ISO arriba mencionados, si bien contemplan la especificación de procesos relacionados con la definición de requerimientos no funcionales, con medición y con evaluación; y definen además los términos involucrados en cada proceso –mediante un glosario de términos-, no pueden considerarse como un enfoque general a ser usado en procesos de medición, evaluación y cambio de manera integral, sistemática e instanciable en estrategias. Por un lado, debido a que no existe una terminología común entre los estándares. Por otro lado, tampoco existe un proceso integrado que contemple tanto actividades de medición y evaluación, como de evolución y cambio.

Finalmente, una de las iniciativas referentes en el área de gestión de calidad y mejora continua es CMMI [8]. Este estándar *de facto* establece pautas para guiar la mejora de procesos en una organización. Define 22 áreas de proceso que se integran por un conjunto de prácticas que, llevadas a cabo de manera conjunta, buscan satisfacer un grupo de metas esenciales para la mejora del área. El enfoque de medición que utiliza CMMI es orientado a metas y se basa en las ideas de GQM y del proceso de medición del estándar ISO 15939. Sin embargo, CMMI no define explícitamente ningún modelo de proceso. Más bien se limita a fijar metas que deben alcanzarse para cada área de proceso y prácticas para alcanzar tales metas, pero sin establecer ningún proceso específico.

CMMI es ampliamente utilizado en la industria como guía que orienta a las organizaciones a la mejora continua de los procesos que en ella intervienen. No obstante, no podemos considerarlo como una estrategia integrada de ME&C puesto que carece de una base conceptual robusta, una representación de procesos más formalmente especificada, y de métodos y herramientas para llevar a cabo las prácticas/actividades; aunque este último aspecto no se contempla dado que no es objetivo en CMMI el de especificar el cómo de las prácticas específicas que enuncia para cada área de proceso.

6. Conclusiones y Trabajos Futuros

En este trabajo hemos presentado el proceso de la estrategia denominada GOCAME$^+$ que permite la mejora dirigida por medición y evaluación. Esta nueva estrategia basa su fundamento a partir de las oportunidades de mejora encontradas para las estrategias GOCAME y SIQinU o adaptaciones sugeridas en [23].

La nueva estrategia se considera *genérica* puesto que sirve para evaluar diferentes categorías de entidad y focos, aspectos que no son posibles en SIQinU ya que es específica para mejorar la calidad de un sistema en uso. GOCAME, en consideración de este aspecto, sí es genérica dado que permite personalizar las actividades involucradas en su proceso para distintos focos y categorías de entidad, según la necesidad de información establecida. En este sentido, la denominación de las actividades del proceso de ME&C en GOCAME$^+$ deben ser especializadas (o personalizadas) según el foco y entidad considerados.

Si bien para facilitar la comprensión en la prueba de concepto de la Sección 4, no especializamos los nombres de las actividades genéricas de GOCAME$^+$, el lector fácilmente entenderá que, por ejemplo, A1 (Definición de Requerimientos No Funcionales) se deberá especializar a Definición de Requerimientos No Funcionales para Calidad Externa, dado que el foco era Usabilidad (característica del modelo de Calidad Externa) para la categoría de entidad Sistema (en particular, Aplicación de Red Social). De la misma manera, habría que especializar los nombres genéricos de las 7 actividades restantes de GOCAME$^+$.

La estrategia propuesta es además *multipropósito*, es decir, sirve para comprender, predecir, mejorar, controlar, etc. distintas categorías de entes. Este aspecto está restringido en SIQinU a comprender y mejorar solamente.

Por otra parte, como hemos discutido en la Sección 2, la estrategia GOCAME no incluye actividades de cambio y mejora continua. En este sentido, en Trabajos Relacionados hemos analizado que los enfoques de medición y evaluación más conocidos y utilizados no contemplan actividades de medición, evaluación y cambio, de manera conjunta e integrada. Por lo que un enfoque integral que incluya medición, evaluación y cambio es necesario en cualquier organización que promueva programas que involucren dichas actividades como parte de la tarea de gestionar la calidad dentro de sus líneas de producción, mantenimiento y evolución de productos y servicios software.

De los tres pilares que incluye una estrategia de M&E integrada, este trabajo se ha centrado en la especificación del proceso de ME&C para GOCAME$^+$, considerando las vistas funcional y de comportamiento. Con el fin de ejemplificar dicho proceso, se utilizó un fragmento de un caso de estudio relacionado a la aplicación Facebook para smartphones.

Como trabajo futuro se desarrollará una base conceptual de evolución y cambio para GOCAME$^+$. Para este fin, es necesario ampliar la actual base conceptual de M&E de C-INCAMI para robustecerla con términos y relaciones pertenecientes al dominio de evolución y cambio. Las ontologías de cambios propuestas en [14, 24] serán sin dudas referencias valiosas al momento de investigar este dominio específico y su enlace con el de medición y evaluación.

7. Reconocimientos

Este trabajo está parcialmente soportado por el proyecto POIRE 2013-10 de la UNLPam, y por el plan plurianual 2013-2016 PROMINF/SPU, Ministerio de Educación, Argentina.

8. Referencias

[1] Baffini, M., Rivera, M., & Olsina, L. (2006) Sistema Colaborativo de Revisión de Métricas. 3th Engineering Workshop of Software Engineering and Data Bases, XII CACIC. San Luis, Argentina.

[2] Basili V., (1985), Quantitative Evaluation of Software Engineering Methodology, Proceedings of the First Pan Pacific Computer Conference, Melbourne, Australia.

[3] Basili V., (1989) Software Development: A Paradigm for the Future, Proceedings of the 13th Annual International Computer Software & Applications Conference (COMPSAC), Keynote Address, Orlando, FL.

[4] Basili V., Caldiera G., Rombach D. (1994) Goal, Question, Metric Paradigm, Encyclopedia of Software Engineering, vol. 1, J.J. Marciniak, ed., John Wiley & Sons, pp. 528-532.

[5] Basili V. et al (2007) GQM+Strategies: Aligning Business Strategies with Software Measurement, Proc. 1st Int'l Symp. Empirical Software Eng. and Measurement (ESEM 07), IEEE CS Press, pp. 488–490.

[6] Becker P., Lew P., Olsina L. (2012) Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy, In: Advances in Software Engineering Journal, Academic Editor: Osamu Mizuno, Hindawi Publishing Corporation, Vol. 2012, 27 pg., DOI:10.1155/2012/949746.

[7] Becker, P., Papa, F., Olsina, L. (2014) Process Conceptual Base for Enriching a Measurement and Evaluation Ontology, In proc. of XVII Conferencia Iberoamericana en Software Engineering (CIbSE'14), Pucón, Chile, pp. 53-66. ISBN: 978-956-236-247-4

[8] CMMI (Capability Maturity Model Integration)(2010), Version.1.3. CMU/SEI-2010-TR-033, USA.

[9] Deming E. (1986) Out of the Crisis. MIT Center for Advanced Engineering Study, MIT Press, Cambridge, MA.

[10] ISO/IEC (2011) ISO 25010. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models.

[11] ISO/IEC. (2007) ISO 15939. Systems and software Engineering - Measurement Process (2nd Edition).

[12] ISO/IEC (2001) 14598-5: Information Technology - Software Product Evaluation - Part 5: Process for evaluators.

[13] ISO/IEC. (2001) ISO 9126-1: Software Engineering Product Quality - Part 1: Quality Model.

[14] Kitchenham, B. A., Travassos, G. H., von Mayrhauser, A., Niessink, F., Schneidewind, N. F., Singer, J., Yang, H. (1999). Towards an ontology of software maintenance. Journal of Software Maintenance, 11(6), pp. 365-389.

[15] Lew P., Olsina L., Becker P., and Zhang, L. (2012) An Integrated Strategy to Understand and Manage Quality in Use for Web Applications. Requirements Engineering Journal, Springer London, (17): 4, pp. 299-330.

[16] McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., et al. (2001) Practical Software Measurement: Objective Information for Decision Makers. Addison-Wesley Professional. ISBN-13: 978-0-201-71516-3.

[17] Olsina, L., Martín, M. (2004) Ontology for Software Metrics and Indicators. Journal of Web Engineering, Rinton Press, US, 2(4), pp. 262-281, ISSN 1540-9589.

[18] Olsina L., Covella G., Dieser A. (2013) "Metrics and Indicators as Key Organizational Assets for ICT Security Assessment", Chapter 2 in *Emerging Trends in ICT Security*, Elsevier (Morgan Kaufmann), 1st Edition, Akhgar & Arabnia (Eds.), Print Book ISBN: 9780124114746, pp. 25-44.

[19] Olsina, L., Rossi, G., Garrido, A., Distante, D., & Canfora, G. (2008) Web applications refactoring and evaluation: a quality-oriented improvement approach. Journal of Web Engineering, Volume 7 Issue 4. Rinton Press , 258–280.

[20] Olsina, L., Papa, F., Molina, H. (2008) How to Measure and Evaluate Web Applications in a Consistent Way. Web Engineering: Modelling and Implementing Web Applications Springer HCIS, Rossi, Pastor, Schwabe, and Olsina (Eds), Ch. 13 pp. 385–420.

[21] Olsina L., Lew P., Dieser A., Rivera B. (2012) Updating Quality Models for Evaluating New Generation Web Applications, In: Journal of Web Engineering, Special issue: Quality in new generation Web applications. Rinton Press, US, 11 (3), pp. 209-246, ISSN 1540-9589.

[22] Olsina L.; Santos L.; Lew P., (2014) Evaluating Mobileapp Usability: A Holistic Quality Approach, In: 14th Int'l Conference on Web Engineering, ICWE 2014, Toulouse, France, Springer International Publishing Switzerland, S. Casteleyn, G. Rossi, and M. Winckler (Eds.): LNCS 8541, pp. 111-129.

[23] Papa, F., (2012) Toward the Improvement of a Measurement and Evaluation Strategy from a Comparative Study, In LNCS 7703, Springer: Current Trends in Web Engineering, ICWE Int'l Workshops, M. Grossniklauss and M. Wimmer (Eds.), pp. 189-203, ISBN 978-3-642-35622-3

[24] Ruíz, F., Vizcaíno, A., Piattini, M., & García, F. (2004) An ontology for the management of software maintenance projects. In International Journal of Software Engineering and Knowledge Engineering.

Aspectos para la definición de Reglas de Negocio

Castiblanco Sparano, Jose Luis
josesparano3@gmail.com

Baigorria, Lorena Soledad.
flbaigor@unsl.edu.ar

Departamento de Informática - Universidad Nacional de San Luis

Resumen

Las reglas de negocio cambian constantemente, generalmente estas definen las políticas, prácticas y procedimientos que se deben llevar a cabo para satisfacer los objetivos de una organización. Es menester que todas las aplicaciones de negocio apliquen dichas reglas ya que son fundamentales para la toma de decisiones. Si bien esto ocurre, las reglas quedan inmersas dentro del código de la aplicación junto con la funcionalidad básica, ocasionando inconsistencias en los procesos cada vez que se requiera realizar algún cambio. Es ahí donde juega un papel fundamental el hecho de separar las reglas de negocio del núcleo de la aplicación, para facilitar el trabajo de desarrolladores y expertos en el dominio, mantenimiento y respectiva administración. En este trabajo, se propone la Programación Orientada a Aspectos como eje fundamental para lograr dicho objetivo, brindándoles a los participantes del negocio un mecanismo intuitivo para definir, crear, administrar las reglas de negocio e integrarlas a la aplicación.

Palabras clave: AspectJ, Comercio Electrónico, Programación Orientada a Aspectos, Programación Orientada a Objetos, POA, POO, Reglas de Negocio, Separación de Concerns.

1. Introducción

En la actualidad, las organizaciones basan su funcionalidad en la toma de decisiones y éstas a su vez en la definición de un conjunto de reglas denominadas reglas de negocio. Estas reglas no son más que condiciones y hechos que permiten tomar decisiones, además influyen en el comportamiento colectivo de las organizaciones y sus integrantes, así como de los sistemas que la componen [1]. En las aplicaciones organizacionales, la lógica que satisface los objetivos definidos a través de las reglas de negocio no está completamente formalizada y éstas se encuentran inmersas dentro del código junto con la funcionalidad básica, por lo que al producirse cambios en el negocio, no se reflejan en las definiciones, generando inconsistencias a largo plazo [2]. Por tal motivo, cuando se requiere realizar algún cambio en las políticas del negocio - lo cual en el ámbito del comercio electrónico ocurre frecuentemente - los expertos del dominio en conjunto con los desarrolladores, necesitan identificar, ubicar, analizar dependencias, extraer, modificar e ingresar nuevamente las reglas de negocio; tarea complicada si se tiene en cuenta la complejidad y tamaño de la aplicación y que consume bastante tiempo en cuanto a desarrollo e impacto de los cambios deseados. Por tal motivo, se hace necesario automatizar la generación de las reglas de negocio, separarlas del código funcional básico de la aplicación y buscar en la medida de lo posible que sean administradas por el experto del dominio sin afectar el núcleo del sistema. De esta forma, se pueden tener aplicaciones dinámicas y flexibles, fáciles de mantener, que aporten significativamente a las necesidades de la organización y alcanzar la independencia de tareas, para que los desarrolladores se enfoquen en la integración de nuevas funcionalidades, mantenimiento y colaboración en el proceso de verificación de las nuevas reglas de negocio con los expertos del dominio.

2. Reglas de negocio

Las reglas de negocio constituyen un concepto fundamental al momento de definir los requisitos de

las aplicaciones organizacionales. Se definen las reglas de negocio como sentencias sobre la forma en que la organización realiza negocios. Asimismo reflejan las políticas del negocio que apuntan a la satisfacción del cliente y de los objetivos planteados, sin dejar de lado el uso de los recursos, leyes o convenciones de la organización. Posteriormente estas reglas se transforman en requisitos, lo que significa que pueden ser implementadas en un sistema de software como un requisito de ese sistema [3].

Las reglas de negocio expresan una política de negocio mediante un vocabulario formal y una serie de sentencias ***si-entonces.*** Describen, limitan o controlan un determinado aspecto del negocio [4]

Ejemplos básicos de reglas de negocio para un sistema de comercio electrónico son:

- Si un cliente no tiene órdenes válidas con un tiempo mínimo de 3 meses de antigüedad, entonces no es un cliente preferencial.
- Si un cliente es preferencial, entonces califica para un 20% de descuento.
- Un cliente no puede tener más de 10 órdenes abiertas al mismo tiempo.
- El total de la compra no puede ser mayor a USD $10.000.
- Si un cliente devuelve un producto en buen estado dentro de los 30 días de hacerse efectiva la compra, se retornará el 90% del valor del producto al cliente.
- Si es el día de aniversario de la tienda, se ofrecerá un 15% de descuento a todos los clientes y un 30% a aquellos pertenecientes a la categoría “cliente dorado”.

Una regla de negocio pertenece a la organización y no al sistema de información de la organización o a la aplicación de software asociada, por tanto, las reglas pertenecen al Universo de Discurso [5]. Por Universo de Discurso se entiende el contexto bajo el cual está definido, desarrollado y mantenido el software. Incluye a todos los participantes del negocio, tales como fuentes de información, personas, sectores, entre otros. Partiendo de este concepto, se hace indispensable la separación de las reglas de negocio de la funcionalidad básica de la aplicación, lo que facilita la respuesta rápida y eficaz ante los constantes cambios que presentan las organizaciones, específicamente aquellas dedicadas al comercio electrónico.

Una buena definición de las reglas de negocio garantiza una buena política de decisiones, ahora, si a esto se le añade independencia de la funcionalidad básica de la aplicación sobre la cual toma efecto, se garantiza la facilidad de cambio, adaptación rápida a los cambios presentados en el mercado, tiempos de respuesta más rápido para las tareas llevadas a cabo por los desarrolladores y expertos en el dominio. Esto es de gran importancia, ya que en el mundo del comercio electrónico la competencia es tan fuerte que los que mejor y más rápido se adapten a las necesidades variantes del mercado son quienes llevan la delantera.

Los sistemas de comercio electrónico necesitan reglas de negocio que estén separadas de los demás componentes, con el fin de que todos sepan de su existencia, externalizadas para que todos sepan qué son, rastreables hacia sus orígenes para que todos sepan de dónde vienen y debidamente orientadas al cambio para que todos sepan cómo mejorarlas. Cuando el negocio no emplea eficazmente las reglas de negocio, hace caso omiso de ellas o no las define para que sean adaptables, presenta dificultades relacionadas con el cambio, aprendizaje, crecimiento y consistencia con su comportamiento ubicándolo en una posición peligrosa dentro del mercado. Por tal motivo, hay que saber identificar cuáles son los escenarios donde la definición y adaptación de las reglas de negocio juegan un rol crítico dentro de los procesos organizacionales:

- El negocio necesita cambiar constantemente pero sus sistemas se convierten en una barrera.
- Se presentan nuevas políticas o leyes que permiten nuevas oportunidades de negocio.
- Hay una competencia virtual en la que los productos o servicios ofrecidos, están disponibles entre competidores que se

encuentran únicamente en la web.

Los expertos en el tema reafirman la importancia de las reglas de negocio en los sistemas, C.J. Date afirma que "*una nueva tecnología llamada reglas de negocio está comenzando a tener impacto en la industria IT, en la forma como desarrollamos y mantenemos las aplicaciones*" [6]. John E. Mann expresa una opinión similar, en la que afirma: "*internet está creando muchas cosas en las cuales el enfoque basado en reglas es el único que tiene sentido*" [7].

Los beneficios que se obtienen al utilizar correctamente un enfoque orientado a las reglas de negocio en un sistema, en este caso uno de comercio electrónico son variados. Entre estos se destacan:

- Simplicidad: es fácil de entender tanto para los expertos del negocio como para la parte técnica.
- Base teórica: existe bastante teoría y experiencia práctica con reglas desde el campo de la ingeniería de software.
- Pequeño número de conceptos no técnicos: el núcleo de este tipo de enfoque es la regla en sí y al haber pocos conceptos alrededor de ella se hace más fácil de comprender (decisiones, patrones, familias de reglas, clausulas).
- Independencia: se pueden definir las reglas independientemente de la tecnología utilizada en las aplicaciones.
- Facilidad en el desarrollo: la aplicación llama a la regla cuando necesita una decisión basada a partir de ciertas condiciones.
- Reutilización de las reglas: una regla puede ser utilizada por diferentes aplicaciones, por lo que se puede definir un conjunto de reglas que pueden ser utilizadas para diferentes propósitos.
- Diseño simplificado de sistemas: permite separar el flujo central del proceso de la ejecución de las reglas, creando un flujo para las reglas y otro para la secuencia de la ejecución del sistema.
- Reglas dinámicas: por la facilidad de cambiar o adaptar las reglas de negocio acorde a las necesidades.
- Rendimiento: los productos desarrollados para el manejo y administración de las reglas de negocio son hechos única y exclusivamente para esta tarea, lo que optimiza su rendimiento.
- Sistemas incrementales: Un sistema de reglas de negocio puede ser desarrollado fácilmente en piezas incrementales.

Otros beneficios en la parte técnica y optimización de las funciones designadas a expertos del dominio y desarrolladores:

- Los cambios no serán más disruptivos ni costosos.
- Los expertos en el dominio, estarán más cerca de las especificaciones del sistema.
- Las reglas de negocio están documentadas a través de un repositorio y no más escondidas en el código.
- Los sistemas con enfoque a las reglas de negocio evolucionan más rápido.
- Significa menos código de negocio en la aplicación, lo que implica menos probabilidades de errores.
- Menor tiempo de desarrollo de las aplicaciones, ya que se centra en la funcionalidad básica y no en las reglas de negocio.

El proceso de identificación de las reglas de negocio en un sistema depende de la complejidad del mismo, así como la forma cómo se hayan definido previamente las relaciones entre cada uno de sus componentes. Generalmente para identificar y determinar cuáles son las reglas de negocio que se van a integrar en un sistema se deben tener en cuenta las siguientes pautas:

- Identificar las decisiones detrás de los eventos del negocio.
- Descomponer las decisiones en reglas atómicas.
- Clasificar, nombrar las reglas y escribirlas utilizando alguna plantilla predefinida.

- Mejorar la calidad de las reglas determinando inconsistencias y/o redundancias.
- Clasificar las reglas acorde a patrones definidos.
- Determinar la dependencia entre las ejecuciones de las reglas.
- Preparar las reglas para el cambio.

3. Programación Orientada a Aspectos

Todo sistema de software hecho con calidad debe estar bien diseñado. La diferencia entre un buen y mal sistema es su capacidad para evolucionar a través del tiempo y adaptarse a los nuevos requerimientos sin alguna dificultad. Al combinar la Programación Orientada a Objetos (POO) con la Programación Orientada a Aspectos (POA), se obtiene una buena dirección que encamina a diseñadores y desarrolladores en el logro de sus objetivos.

En la práctica se suele dividir la aplicación en diversas capas lógicas (de presentación, negocio y datos). Sin embargo, existen ciertas funcionalidades que atraviesan dichas capas transversalmente. En la POA esto se conoce como ***crosscutting concerns.***

Los crosscutting concerns son entidades independientes que atraviesan transversalmente funcionalidades que no son consideradas "básicas" en una aplicación [8]. Por ejemplo la seguridad, transaccionalidad, logueo, manejo de excepciones, control de la concurrencia, entre otras.

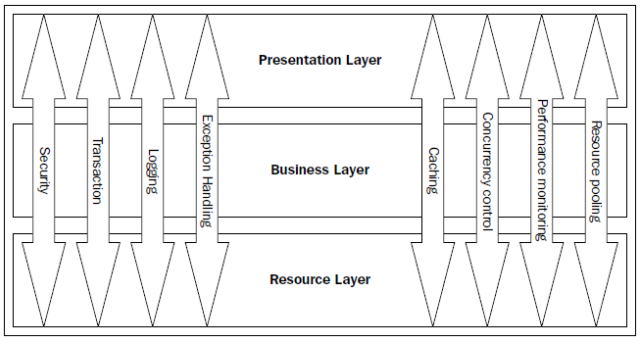


Figura 1. Crosscuting Concerns

Si se compara la POO con la POA, es fácil identificar una característica particular. En la POO la unidad básica es la Clase, mientras que en la POA es el *Aspecto,* el cual contiene la implementación de un crosscuting concern.

Kiczales define en [11]: "*Un aspecto es una unidad modular que se disemina por la estructura de otras unidades funcionales. Los aspectos existen tanto en la etapa de diseño como en la de implementación. Un aspecto de diseño es una unidad modular del diseño que se entremezcla en la estructura de otras partes del diseño. Un aspecto de implementación es una unidad modular del programa que aparece en otras unidades modulares del programa*".

La funcionalidad provista por las entidades en los aspectos será aplicada a los objetos a través de un tejedor o clases proxy. Otro punto a tener en cuenta es la lista de dificultades que se presentan con el enfoque de la POO y que pueden ser fácilmente resueltas utilizando un enfoque de POA. Estos inconvenientes radican principalmente en el código disperso y la falta de legibilidad del mismo, por ejemplo al tener en un mismo módulo logueos, autenticación, caché, manejo de excepciones, entre otras funcionalidades que no hacen parte del núcleo básico del sistema.

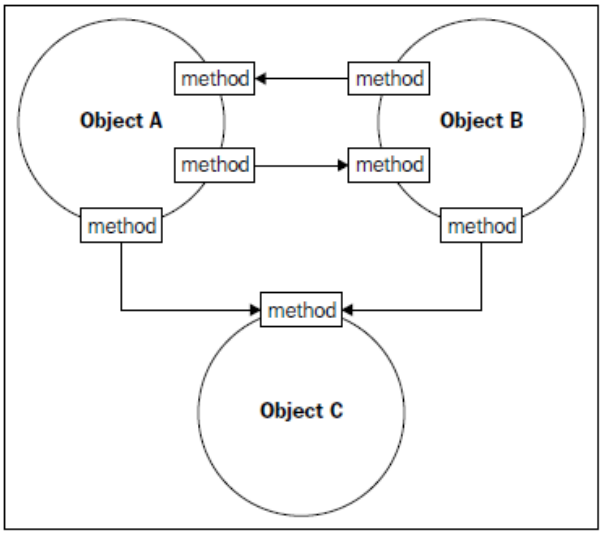


Figura 2. Interacción tradicional en la POO

En la POO los objetos se comunican por medio del pasaje de mensajes utilizando llamadas a métodos. En este enfoque los crosscuting concerns

están inmersos en las clases A, B y C, lo que conlleva a las desventajas descritas con anterioridad.

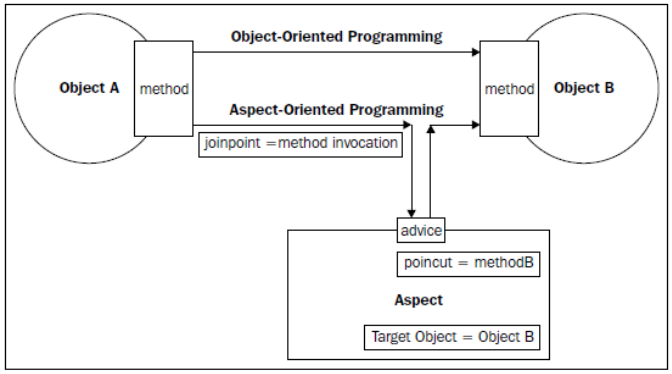


Figura 3. Flujo de invocación de un método en la POA

En el enfoque orientado a aspectos, ilustrado en la Figura 3, se aprecia cómo se extrae la funcionalidad de los crosscutting concerns de la implementación orientada a aspectos y se aplican aspectos donde realmente son indispensables. También se puede observar la inclusión de nuevos términos y conceptos, definidos en [10] y descritos a continuación.

Aspecto: Es el equivalente de la clase en la POO, es en pocas palabras la funcionalidad del crosscutting concern.

Joinpoint: Es el punto de aplicación del aspecto, es un punto de ejecución del programa. Por ejemplo la invocación de un constructor o el manejo de una excepción.

Advice: Es la acción que un aspecto lleva a cabo en un determinado joinpoint. Pueden tener la forma "around", "before" y "after".

Pointcut: Se define como un conjunto de joinpoints.

Target Object: Es el módulo u objeto sobre el cual será aplicado el aspecto.

Weaving: Es el enlace de acción entre los aspectos y objetos sobre los que se deben aplicar los advices.

En la Figura 3, el flujo se describe de la siguiente forma. Teniendo en cuenta que un joinpoint es el punto de aplicación del aspecto, y que en este caso el joinpoint es el **método B,** el aspecto ejecuta el crosscutting concern incluido en el advice cuando el método B es invocado en el **Target Object B**. Este tipo de interceptación pertenece a un advice de tipo ***before*** (se ejecuta el código del advice ***antes de***).

4. POA para definir reglas de negocio

Una aplicación de software involucra muchos y variados concerns. Por concern, se entiende aquella propiedad o área de interés en un sistema [9]. Los concerns pueden ser de diversas formas, por ejemplo aquellos de alto nivel como la seguridad, y calidad en el servicio, así como los de bajo nivel como por ejemplo el manejo de caché y buffer. De igual forma, existen concerns funcionales y no funcionales, en la primera categoría se pueden mencionar las reglas de negocio, punto fundamental en este artículo.

En cualquier negocio que quiera perdurar en el mercado, las reglas de negocio sin duda alguna cambiarán constantemente. Si estos negocios se apoyan en el uso de aplicaciones de software, los cambios en dichas reglas deben verse reflejados. Con el enfoque de la POO, las reglas de negocio están inmersas en su gran mayoría en el código central de la aplicación. De igual forma, el trabajo de administración de las reglas, queda sujeto al desarrollador, lo cual no es óptimo, ya que no es precisamente este actor del negocio quien maneja de mejor forma todo lo relacionado con la definición, integración y dependencias entre las reglas.

Si se tiene en cuenta que la POA facilita la separación de concerns y que en este caso, la separación de las reglas de negocio es vital para facilitar el mantenimiento de las aplicaciones de software, no cabe duda que emplear este enfoque sirve de gran ayuda para lograr uno de los principales objetivos de cualquier organización dedicada al comercio, en este caso puntual, al comercio electrónico: evolucionar rápidamente ante los cambios en el mercado, para satisfacer las necesidades de los clientes, sin ver afectada la productividad de sus empleados.

Estas son algunas de las desventajas que se encuentran en el enfoque de la POO, y que fácilmente se resolverían con el enfoque de la POA:

- El experto en el dominio no puede administrar de manera cómoda las reglas de negocio que considera fundamentales para el funcionamiento de la organización.
- Las reglas de negocio al final son integradas en la aplicación por el desarrollador, lo cual es muy riesgoso ya que pudo ocurrir alguna interpretación errónea sobre los requerimientos solicitados por el experto en el dominio.
- Cada vez que se requiera realizar un cambio en siquiera una regla de negocio, el desarrollador necesita suspender sus actividades relacionadas con el mantenimiento de la funcionalidad básica de la aplicación, lo que implica cierta pérdida de la productividad.
- El experto en el dominio pierde el rastro del número de reglas de negocio que participan en un momento determinado en la aplicación, por tanto no se puede llevar un análisis concreto de la efectividad de las decisiones tomadas.
- Si se define una regla de negocio que aplica en diferentes puntos de la funcionalidad básica, se puede incurrir en la repetición de código y dificultar la rastreabilidad de las mismas.
- Las reglas de negocio deben ser independientes de la tecnología utilizada, si se decide cambiar la tecnología de la aplicación, obligatoriamente, se deben rehacer todas y cada una de las reglas de negocio.

Teniendo en cuenta las desventajas descritas anteriormente, y aquellas que sin duda surgirán con el tiempo, es necesario plantear una forma diferente en el diseño e implementación de las aplicaciones relacionadas con el comercio electrónico para asegurar el fácil mantenimiento, adaptación rápida al cambio por parte de la aplicación y separación de tareas entre los dos participantes más importantes del negocio en este punto (experto en el dominio y desarrollador). Por tal motivo se propone utilizar todo el potencial de la POA, para lograr este objetivo.

A continuación, se ilustra el diseño arquitectónico propuesto para la definición, administración e integración de las reglas de negocio a una aplicación de comercio electrónico:

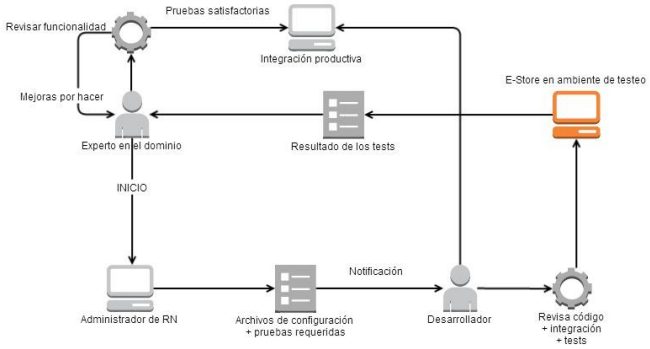


Figura 4. Diseño para la integración de RN usando POA.

El flujo es el siguiente:

- Inicia con el experto en el dominio, quien a través de una aplicación destinada a la administración de las reglas de negocio (creación, lectura, actualización, borrado) define o modifica las reglas que serán integradas a la aplicación.
- La aplicación generará los archivos de configuración necesarios, así como el código del aspecto en un lenguaje orientado a aspectos.
- El experto en el dominio, elaborará un documento donde se especificará cuáles son los casos de prueba requeridos para las nuevas reglas.
- Una vez definidas las reglas de negocio que se integrarán, los archivos de configuración, código y los respectivos casos de prueba a ser tenidos en cuenta, se notifica al desarrollador la existencia de las mismas.
- El desarrollador revisa los archivos recibidos, y procede a integrarlos a la aplicación de testeo previo al despliegue productivo por los riesgos implicados.
- El desarrollador procede a realizar los tests automáticos correspondientes para validar la

correcta funcionalidad de la aplicación y notifica al experto en el dominio el resultado de los mismos.

- El experto en el dominio procede a realizar sus respectivas pruebas en el ambiente de testeo y al verificar que la regla de negocio se refleja en el sistema y a su vez esté integrada, da el visto bueno al desarrollador para que las nuevas reglas sean integradas en la aplicación productiva. Si por el contrario no está conforme, procede a la creación/modificación de las reglas de negocio pertinentes e inicia nuevamente el ciclo.

5. Conclusiones y trabajo futuro

Este trabajo identifica un problema común y crítico en el uso de la POO para aplicaciones destinadas al comercio electrónico, la dificultad de separar los concerns ocasionando problemas a corto, mediano y largo plazo en la mantenibilidad del sistema. Entre los problemas más importantes a solucionar se encuentran el código repetido, la inmersión de las reglas de negocio dentro de la funcionalidad básica, la complejidad de realizar los cambios requeridos en el tiempo preciso, la designación de tareas a desarrolladores que pueden incurrir en errores al no tener un dominio completo del negocio, entre otros. Por tal motivo, se plantea la POA como alternativa para solucionar estos inconvenientes. Al emplear eficientemente todo el potencial que ofrece, se logrará sin duda desarrollar aplicaciones que apliquen de manera óptima las reglas de negocio definidas por el experto del dominio en la organización y que sean capaces de adaptarse rápidamente a los cambios que surgen en el mercado.

Separar y facilitar las tareas de los expertos del dominio y desarrolladores es un factor fundamental para aumentar la productividad, así como disminuir las probabilidades de implementar reglas de negocio erróneamente y evolucionar satisfactoriamente en un mercado que cambia de manera constante. Además, si se garantiza una independencia tecnológica que permita definir tranquilamente las reglas de negocio sin que se vea afectada la funcionalidad básica de la aplicación, se favorece la robustez y escalamiento de las aplicaciones.

La propuesta realizada en esta publicación consiste en un módulo administrativo que permita al experto en el dominio, a través de un mecanismo intuitivo, definir, consultar y modificar las reglas de negocio que considere importantes para el buen funcionamiento de la organización. Además, el uso de un sistema de comunicación empleando tecnologías actuales que le permitan al desarrollador conocer los requerimientos de cambios en las reglas de negocio, para que se evalúen, prueben e integren en la aplicación.

Entre las mejoras que se plantean como trabajo futuro está la posibilidad de integrar y testear de forma automática las reglas de negocio definidas por el experto en el dominio desde el módulo administrativo en la aplicación comercial (sitio web de comercio electrónico), con el fin de evitar el envío de los archivos al desarrollador, optimizando el tiempo en la implementación de los cambios y brindándole la posibilidad al desarrollador de enfocarse en las tareas relacionadas 100% con la funcionalidad básica de la aplicación. Otra mejora podría ser la elección de la tecnología empleada para definir los aspectos, se considera la viabilidad de generar un archivo genérico en formato XML (Extensible Markup Language) que se pueda transformar por ejemplo a AspectJ [12] o SpringAOP [13] haciendo uso de alguna aplicación externa o adicionando funcionalidad al administrador de reglas de negocio.

6. Referencias

[1] Halle, Barbara von, and G. Ronald. *Business rules applied: building better systems using the business rules approach*. John Wiley & Sons, Inc., 2001.

[2] Silveira, Luciano, and M. Sc Juan José Prada. "Diseño e implementación de un motor de reglas dinámicas usando especificaciones GeneXus." *Universidad de la República, Uruguay* (2010): 55-65.

[3] Rosca, Daniela, et al. "A decision making methodology in support of the business rules lifecycle." *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. IEEE, 1997.

[4] "Reglas de negocio. IBM." http://pic.dhe.ibm.com/infocenter/brjrules/v7r1/index.jsp?topic=%2Fcom.ibm.websphere.ilog.jrules.doc%2FContent%2FBusiness_Rules%2FDocumentation%2F_pubskel%2FJRules%2Fps_JRules_Global1384.html. Consultado el 11/05/2014.

[5] Leite, Julio CSP, and A. P. Oliveira. "A client oriented requirements baseline."*Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on*. IEEE, 1995.

[6] Date, Christopher John. *What not how: the business rules approach to application development*. Addison-Wesley Professional, 2000.

[7] Mann, John E. (April 2000). "Rules for E-Business." Patricia Seybold Group. Written for Versata Inc.

[8] Dessì, Massimiliano. *Spring 2.5 Aspect Oriented Programming*. Packt Publishing Ltd, 2009.

[9] Cibrán, María Agustina, Maja D'Hondt, and Viviane Jonckers. "Aspect-oriented programming for connecting business rules." *Proceedings of the 6th International Conference on Business Information Systems*. Vol. 6. No. 7. 2003.

[10] Ramnivas Laddad (2003). AspectJ in Action: Practical Aspect-Oriented Programming.

[11] "Domain-Specific Aspect Languages" http://www.dsal.cl/dsal/cool. Consultado el 19/08/2014.

[12] "AspectJ". https://eclipse.org/aspectj/ Consultado el 19/08/2014.

[13] Dessì, M. (2009). *Spring 2.5 Aspect Oriented Programming*. Packt Publishing Ltd.

[14] Figura 1, 2 y 3. Ramnivas Laddad (2003). AspectJ in Action: Practical Aspect-Oriented Programming.

Towards Counterexample-Guided Computation of Validated Stability Certificates for Hybrid Systems *

Eike Möhlmann and Oliver Theel
Carl von Ossietzky University of Oldenburg
Department of Computer Science
D-26111 Oldenburg, Germany
{*eike.moehlmann,theel*}*@informatik.uni-oldenburg.de*

Abstract

We propose a method to obtain trustable Lyapunov-based certificates of stability for hybrid systems. A hybrid system is a system exhibiting discrete-time as well as continuous-time behaviors, e.g. embedded systems within a physical environment. Stability is a property which ensures that a system starting in any possible state will reach a desired state and remain there. Such systems are particularly useful when a certain autonomous operation is required, e.g. keeping a certain temperature or speed of a chemical reaction or steering a vehicle over a predefined track. Stable hybrid systems are extremely valuable because if an error disturbes their normal operation, they automatically "steer back" to normal operation. Stability can be certified by finding a so-called Lyapunov function. The search for this kind of functions usually involves solving systems of constraints. The state-of-the-art in Lyapunov-based stability verification is to use numerical methods to solve systems of inequalities, which if solvable indicate stability. We propose to use Satisfiability-Modulo-Theory (SMT) methods to (a) validate the results of a numerical solver and (b) use counterexamples to guide the numercial solver towards a valid solution.

Keywords

Hybrid Systems; Automatic Verification; Stability; Lyapunov Theory; Optimization; Satisfiability-Modulo-Theory

1. Introduction

In this paper we present experiments on combining Satisfiability-Modulo-Theory (SMT) methods with – the current state-of-the-art – numerical solvers, for the search of Lyapunov functions. A *Lyapunov function* can be used to certify that a given hybrid system is indeed (asymptotically) stable. It can be seen as a "generalized metric" whose value indicates for a given system state the "distance" to the desired system state; the so-called *equilibrium point*. The equilibrium point is w.l.o.g. assumed to be the origin, i.e. 0, of the state space. Lyapunov functions have a special shape that guarantees that while the system evolves the "distance" of the system state to the equilibrium point decreases. Thus, any system for which a Lyapunov function can be found, eventually reaches the equilibrium point. For some systems, it is even possible to give a bound on the rate in which the system approaches the origin. In that case, one can compute an upper bound on the time required to reach a certain region around the equilibrium point. This is especially useful if one is interested in *region stability* [1].

The search for Lyapunov functions involves generating and solving sets of *conditioned constraints*, where each constraints has the form $\forall x \in P : 0 \preceq f(\rho, x)$. Here, "$0 \preceq f(\rho, x)$" is a linear constraint on polynomials involving some free parameters ρ, where "$\preceq$" is the positive semi-definiteness operator, and P is the region in which the constraint has to be satisfied. This is called Semidefinite Programming (SDP) [2].

Conditioned constraints cannot directly be given to an SDP solver (such as CSDP [3] or SDPA [4]) because such a solver expects unconditioned constraints. So, instead, one uses the so-called *S-Procedure* which constructs unconditioned constraints. These unconditioned constraints can then be handed over to an SDP solver.

The typically used SDP solver uses some kind of interior point methods and thus numerically approximates the solution. In practice, it is a problem that such a numerical solvers sometimes suffers from numerical inaccuracies. Therefore, the constraints may be *strengthened* by adding additional "gaps" to the inequalities. This makes them more robust against these numerical issues. However, such gaps can introduce contradictions. In [5], a heuristic is proposed, which counter-acts this problem. The heuristic analyzes the conditioned constraints before the unconditioned constraints are generated, and it removes implicit equalities by substituting free parameters in the other constraints.

While such a strengthening indeed helps to obtain solutions, one still needs to validate the result. In our tool, STABHYLI [6], we already use the computation of eigenvalues and principle minors to further ratify the validity

This work has been partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

of a solution. However, the computation of eigenvalues is usually done via numerical algorithms which again may suffer from numerical issues.

Therefore we propose the use of SMT solvers to validate candidate solutions. An SMT solver combines SAT solvers with theory solver to check the satisfiability (SAT) of first-order logic (FOL) formulae wrt. first-order theories. One instance of such a theory is nonlinear real arithmetic (NRA). A formula is satisfiable if there exists a model – a valuation for each free variable – such that using the model, the formula evaluates to $true$. In general FOL is semidecidable, i.e., there exists a procedure such that, given a model and a formula, the procedure eventually decides whether the model satisfies the formula. Due to Tarski [7], we know that NRA on the other hand is decidable with double exponential complexity.

Related Work

SMT solving has received quite some attention over the last years but, unfortunately, most available implementations are either restricted to linear arithmetic or do not support quantifiers. There is only a handful of SMT solvers which support NRA with quantifiers: Z3 [8], CVC3 [9], and CVC4 [10]. As stated above, SMT is already combining SAT solving with different theory solvers. Nevertheless, other combinations built on top of SMT solvers have been made. One combines a linear SMT solver with interval constraint propagation to solve nonlinear real arithmetic problems [11]. Another one combines convex programming with SMT solving to solve non-linear convex constraints [12]. Our approach and the mentioned methods have in common that we want to find a model for a formula. Gao et al [11] also use the result of a linear SMT solver to validate the answer of the interval constraints propagation method which is very similar to what we do, but they consider only quantifier-free formulas. In the search for a solution of the constraint systems, however, we have a single quantifier alternation that is an outer existential and an inner universal quantifier. Thus we cannot apply their method due to the lack of the quantifiers.

In contrast to safety properties, stability has not yet received that much attention wrt. automatic proving and therefor only a few tools are available. The tools known to the authors are the following:

- A tool by Podelski and Wagner which computes a sequence of snapshots and then tries to related the snapshots in decreasing sequence. If successful this certifies region stability, i.e., stability with respect to a region instead of a single equilibrium point [1].
- A tool by Oehlerking et al. implementing a powerful state space partitioning scheme to find Lyapunov functions for linear hybrid systems [13].
- RSOLVER [14] which computes Lyapunov-like functions for continuous system.
- A tool by Duggirala and Mitra that combines Lyapunov functions with searching for a well-foundedness relation for symmetric linear hybrid systems [15].

Finally various MATLAB toolboxes (YALMIP [16], SOSTOOLS [17]) that require a by-hand generation of constraint systems for the search of Lyapunov functions are available. These toolboxes do not automatic prove stability but assist in handling solvers. One such toolbox around SDP solvers is VSDP [18]. It can compute rigorous error bounds by correct handling of the rounding in floating point arithmetic and thereby validating the candidate solution. Nevertheless, it might happen that the error bounds are such that on one hand deciding whether the candidate solution is valid might be impossible. This means that the candidate needs to be rejected. On the other hand the error bounds are again computed numerically. This means it might happen that the bounds are too coarse such that former case results only from the fact of inexactness.

In conclusion, all Lyapunov-based tools use numerical/approximation methods. This means that the obtained candidate solution needs to be validated to ensure correctness.

The contributions of this paper are the following: 1) We present a method to validate stability certificates obtained by numerical approximation algorithms, 2) We sketch a very preliminary idea to guide a numerical solver away from bad candidate solutions.

The remainder of the paper is organized as follows: Section 2 gives the theoretical background and Section 3 describes how candidate solutions can be validated. In Section 4, we sketch the idea to "guide the numerical solver away from invalid solutions". Section 5 gives a simple example which leads to numerical issues in the search for Lyapunov function. These can be detected and fixed due the proposed method. A conclusion is given in Section 6.

2. Preliminaries

In this section, we define the hybrid system model, stability and the techniques required to certify stability of a hybrid system.

Definition 1: A **Hybrid Automaton** is a quintuple

$$\mathcal{H} = (\mathcal{V}, \mathcal{M}, \mathcal{T}, Flow, Inv) \text{ where}$$

- $\mathcal{V}$ is a finite set of *variables* and $\mathcal{S} = \mathbb{R}^{|\mathcal{V}|}$ is the corresponding *continuous state space*,
- $\mathcal{M}$ is a finite set of *modes*,
- $\mathcal{T}$ is a finite set of *transitions* (m_1, G, U, m_2) where
  - $m_1, m_2 \in \mathcal{M}$ are the *source and target mode* of the transition, respectively,
  - $G \subseteq \mathcal{S}$ is a *guard* which restricts the valuations of the variables for which this transition can be taken,
  - $U : \mathcal{S} \to \mathcal{S}$ is the *reset function* which might update some valuations of the variables,

- $Flow : \mathcal{M} \to [\mathcal{S} \to \mathcal{P}(\mathcal{S})]$ is the *flow function* which assigns a *flow* to every mode. A flow $f \subseteq \mathcal{S} \to \mathcal{P}(\mathcal{S})$ in turn assigns a closed subset of $\mathcal{S}$ to each $x \in \mathcal{S}$, which can be seen as the right hand side of a differential inclusion $\dot{x} \in f(x)$,
- $Inv : \mathcal{M} \to \mathcal{S}$ is the *invariant function* which assigns a closed subset of the continuous state space to each mode $m \in \mathcal{M}$, and therefore restricts valuations of the variables for which this mode can be active.

A *trajectory* of $\mathcal{H}$ is an infinite solution in form of a function $x(t)$ over time. Each solution has an associated (possibly infinite) sequence of modes visited by the trajectory. ◇

Intuitively, stability is a property expressing that all trajectories of the system eventually reach an equilibrium point of the sub-state space and stay in that point forever, given the absence of errors. For technical reasons, the equilibrium point is usually assumed to be the origin of the continuous state space, i.e. 0. This is not a restriction, since a system can always be shifted such that the equilibrium is in 0 via a coordinate transformation.

In the following, we refer to $x_{\downarrow\mathcal{V}'} \in \mathbb{R}^{|\mathcal{V}'|}$ as the *sub-vector* of a vector $x \in \mathbb{R}^{\mathcal{V}}$ containing only values of variables in $\mathcal{V}' \subseteq \mathcal{V}$.

Definition 2: **Global Asymptotic Stability with Respect to a Subset of Variables [19].**
Let $\mathcal{H} = (\mathcal{V}, \mathcal{M}, \mathcal{T}, Flow, Inv)$ be a hybrid automaton, and let $\mathcal{V}' \subseteq \mathcal{V}$ be the set of variables that are required to converge to the equilibrium point 0. A continuous-time dynamic system $\mathcal{H}$ is called *globally stable (GS) with respect to* $\mathcal{V}'$ if for all functions $x_{\downarrow\mathcal{V}'}(t)$,

$$\forall \epsilon > 0 : \exists \delta > 0 : \forall t \geq 0 : \\ ||x(0)|| < \delta \Rightarrow ||x_{\downarrow\mathcal{V}'}(t)|| < \epsilon.$$

$\mathcal{H}$ is called *globally attractive (GA) with respect to* $\mathcal{V}'$ if for all functions $x_{\downarrow\mathcal{V}'}(t)$,

$$\lim_{t\to\infty} x_{\downarrow\mathcal{V}'}(t) = 0, \text{ i.e.,} \\ \forall \epsilon > 0 : \exists t_0 \geq 0 : \forall t > t_0 : ||x_{\downarrow\mathcal{V}'}(t)|| < \epsilon,$$

where 0 is the origin of $\mathbb{R}^{|\mathcal{V}'|}$. If a system is both, globally stable with respect to $\mathcal{V}'$ and globally attractive with respect to $\mathcal{V}'$, then it is called *globally asymptotically stable (GAS) with respect to* $\mathcal{V}'$. ◇

Intuitively, GS is a boundedness condition, i.e. each trajectory starting δ-close to the origin will remain ϵ-close to the origin. GA ensures progress, i.e. for each ϵ-distance to the origin, there exists a point in time t_0 such that a trajectory always remains within this distance. By induction, it follows that every trajectory eventually approaches the origin. For a given hybrid system, this can be proven using Lyapunov Theory [20], which was originally restricted to continuous systems but has been lifted to hybrid systems.

We are handling polynomial hybrid systems. Thus, we assume that all guards, invariants, and flows are defined as expressions over polynomials. A *monomial* has the form $\prod_{v\in\mathcal{V}} v^{e_{v,j}}$ where all $e_{v,j} \in \mathbb{N}$. A weighted sum of such monomials is called a *polynomial* $g : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}$ and has the form $g(x) = \sum_j c_j \prod_{v\in\mathcal{V}} v^{e_{j,k}}$ where all $c_j \in \mathbb{R}$. Such a polynomial is called a *parameterized polynomial* if it has the form $f(\rho, x) = \sum_j c_j \rho_j \prod_{v\in\mathcal{V}} v^{e_{v,j}}$ where ρ_j is a *free parameter* and $\rho_j \prod_{v\in\mathcal{V}} v^{e_{v,j}}$ is called a *parameterized monomial*. Indeed, the parameter in a parameterized monomial is rather optional, but to increase readability and to shorten the formulas, we assume every summand in a parameterized polynomial to have a parameter – a "dummy" parameter might be used to represent a constant 1.

Theorem 1: **Discontinuous Lyapunov Functions for a Subset of Variables [19].**
Let $\mathcal{H} = (\mathcal{V}, \mathcal{M}, \mathcal{T}, Flow, Inv)$ be a hybrid automaton and let $\mathcal{V}' \subseteq \mathcal{V}$ be the set of variables that are required to converge. If for each $m \in \mathcal{M}$, there exists a set of variables $\mathcal{V}_m$ with $\mathcal{V}' \subseteq \mathcal{V}_m \subseteq \mathcal{V}$ and a continuously differentiable function $V_m : \mathcal{S} \to \mathbb{R}$ such that

1) for each $m \in \mathcal{M}$, there exist two class K^∞ functions α and β such that

$$\forall x \in Inv(m) : \quad \alpha(||x_{\downarrow\mathcal{V}_m}||) \preceq V_m(x) \\ \wedge\, V_m(x) \preceq \beta(||x_{\downarrow\mathcal{V}_m}||),$$

2) for each $m \in \mathcal{M}$, there exists a class K^∞ function γ such that

$$\forall x \in Inv(m) : \dot{V}_m(x) \preceq -\gamma(||x_{\downarrow\mathcal{V}_m}||)$$

for each $\dot{V}_m(x) \in \left\{ \left\langle \frac{dV_m(x)}{dx} \middle| f \right\rangle \mid f \in Flow(m) \right\}$,

3) for each $(m_1, G, U, m_2) \in \mathcal{T}$,

$$\forall x \in G : V_{m_2}(U(x)) \preceq V_{m_1}(x),$$

then $\mathcal{H}$ is globally asymptotically stable with respect to $\mathcal{V}'$ and V_m is called a *Local Lyapunov Function (LLF)* of mode m.

In Theorem 1, "$\left\langle \frac{dV(x)}{dx} \middle| f \right\rangle$" denotes the inner product between the gradient of a Lyapunov function V and a flow function f. Furthermore, each constraint has the form $0 \preceq c(x)$ where "$\preceq$" is a positive semi-definiteness operator which requires that $c(x)$ is non-negative almost everywhere and $c(0) = 0$.

Definition 3: A **Constraint** is called

- *unconditioned* iff it exhibits the form $0 \preceq f(x)$
- *conditioned* iff it exhibits the form $\forall x \in P : 0 \preceq f(x)$

where $f(x)$ is a parameterized polynomial, $P \subset \mathcal{S}$ is a subset of the continuous state space, and $\forall x \in S$ is called the condition.[1] ◇

Using the S-Procedure [2], a conditioned constraint can be transformed into an unconditioned constraint. The S-Procedure restricts a constraint to some region by exploiting

1. We assume the conditions to be expressed as a conjunction of equalities and inequalities over polynomials, i.e. $\bigwedge_i g(x) \triangle 0$ where $g(x)$ is a polynomial and $\triangle$ is a relation with $\triangle \in \{=, \geq, >\}$.

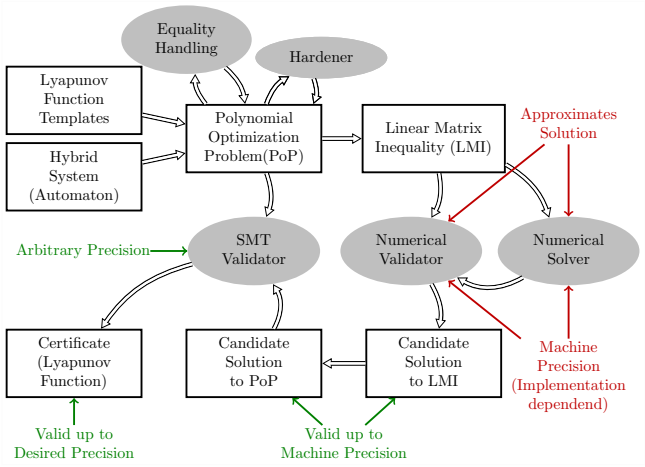


Figure 1: Overview on the Validating Solver Engine

the fact that finding a solution for

$$\left(\sum_i a_i \cdot g_i(x)\right) + \left(\sum_i b_i \cdot h_i(x)\right) \preceq g(x)$$

with $a_i \geq 0$ implies

$$\left(\bigwedge_i 0 \preceq g_i(x)\right) \bigwedge \left(\bigwedge_i 0 = h_i(x)\right) \Rightarrow 0 \preceq g(x).$$

Note that the parameters b_i of the equality conditions $0 = h_i(x)$ are not required to be non-negative.

To find a set of Lyapunov functions for a hybrid system, one also needs to supply the solver with *templates*, which are parameterized polynomials. Such a polynomial usually contains all monomials up to a particular degree that are constructible from a given set of variables. The following steps are performed using the templates: 1. Generate the constraint system described by Theorem 1, then 2. use the S-Procedure to cast the conditioned constraints into unconditioned ones, and then 3. try to solve them.[2] Upon success, we have a certificate of stability. Note that one cannot conclude non-stability from failing to solve the constraints since the described method is sound but incomplete: a higher degree template could still render the problem solvable.

3. Validating Solver Engine

In this section we describe the validating solver engine. This engine can be used to obtain validated stability certificates. Figure 1 gives an overview of the validating solver engine: 1. From the Lyapunov function templates and the hybrid system model, a linear polynomial optimization problem (PoP) is constructed. 2. An equality handler and a constraint hardener post-process the PoP. 3. The PoP is relaxed (via the S-Procedure and the Sums-of-Squares-decomposition) into a linear matrix inequality (LMI). 4. An SDP solver is used to obtain a LMI candidate solution. 5. The LMI candidate solution is validated (e.g., compute eigenvalues[3] and some principal minors). 6. If the LMI candidate solution passes this validation, then 7. it is taken as a candidate solution to the PoP. 8. This PoP candidate solution is then further validated by an SMT solver. If it is successfully validated, then it serves as a stability certificate.

The SDP solver and the numerical validator usually approximate a solution, i.e., they start with an initial valuation of the free parameters. Then they optimize the valuation in the direction which minimizes a given objective function as well as the error.[4] If (a) a certain accuracy is obtained, (b) a maximum number of steps is reached, or (c) the progress becomes too small, then the algorithm stops. Clearly, we can always increase the maximum number of steps and the desired accuracy but the closer the current valuation gets to an optimal solution the smaller the progress becomes. Therefore it is not always possible to reach the optimal solution.

The above described method can be automatized and the resulting LMIs can be solved using numerical solvers. Recall, that – unfortunately – this type of solvers do only approximate solutions and additionally suffer from numerical issues. Prior to applying the S-Procedure and the Sums-of-Squares decomposition, the constraint system has the following form:

$\exists \rho \in \mathbb{R}^m$ such that

$$\bigwedge_i \forall x \left(\bigwedge_j 0 \leq g_{(j,i)}(x)\right) \Rightarrow 0 \preceq f_i(\rho, x)$$

where $\rho = [\rho_1 \quad \dots \quad \rho_m]^T$ are the free parameters (stemming from the templates), $x \in \mathbb{R}^n$ are the system variables (stemming from the hybrid system), $g_{(j,i)}(x) : \mathbb{R}^n \mapsto \mathbb{R}$ are parameter-free polynomials describing a certain region, and $f_i(\rho, x) : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}$ are polynomials involving free parameters. Note that the goal is to find a valuation for ρ and all $f_i(\rho, x)$ are linear in ρ.

Example 1 shows a very simple version of such a constraint system.

Example 1:

$$\exists \rho \in \mathbb{R} : \forall x \in \mathbb{R} : true \Rightarrow 0 \preceq \rho \cdot x^2$$

For this example a numerical solver might return the candidate solution $\rho_{\mathsf{NUM}} = -1 \cdot 10^{-k}$, where k is sufficiently large. On one hand, this is in most cases sufficient and

2. Further steps are required, like translating the polynomial constraints into linear matrix inequalities (LMIs) which, in turn, can be solved using solvers for positive semi-definite problems. This type of translation can be done by using the Sums-of-Squares decomposition [21].

3. Note that computing eigenvalues has issues on its own since it is wellknown to be numerical ill-conditioned [22]

4. Roughly speaking the error determines the quality of the solution: the closer to 0, the better (in the sense of feasibility) the solution is.

allows to conclude feasibility of the above problem. On the other hand, reusing this candidate solution might render any successive calculation invalid. However, STABHYLI composes Lyapunov functions as conic combinations of other Lyapunov functions. This means that in order to guarantee validity of the composed Lyapunov function, we need to assure validity of the combined Lyapunov functions.

Validating Candidate Solutions

We tried to directly solve constraint systems obtained from applying the Lyapunov Theorem, using the SMT solvers Z3 [8], CVC3 [9], and CVC4 [10]. Unfortunately, none of the solvers is able to solve relevant instances – the solvers always returned `unknown`. However, we were able to use SMT solvers to successfully (in-)validate candidate solutions. This use of SMT solvers for validating solutions is beneficial since it can be done with exact arithmetic, i.e., using rationals with full precision. This allows us to conclude that a candidate solution is actually valid and further computation that reuse the candidate solution are also trustworthy.

Remark: If we want solve the constraint systems directly using an SMT solver then we can only use those solvers that support the logic `NRA` (non-linear real arithmetic) since the constraints involve polynomials. In contrast if we want to validate candidate solutions then we can use solvers that support `QF_NRA` (quantifier-free non-linear real arithmetic). Even though both logics are decidable in theory, the first one is computationally intractable in general.

To validate a candidate solution, we substitute the free parameters ρ in the constraint system by the candidate solution ρ_{NUM}. For Example 1, we obtain:

$$\forall x \in \mathbb{R} : \mathit{true} \Rightarrow 0 \preceq -1 \cdot 10^{-k} \cdot x^2. \tag{1}$$

To validate the candidate solution we simply iterate through all constraints and ask whether the negation is satisfiable. For Equation 1, we obtain:

$$\exists x \in \mathbb{R} : \mathit{true} \wedge -1 \cdot 10^{-k} \cdot x^2 < 0. \tag{2}$$

Then, an SMT solver reports one of the following:

case `unknown` We can not assure validity of the candidate solution – this might happen due to insufficient memory or computation time.
case `unsat` We know that the candidate solution is valid.
case `sat` We have to reject the invalid candidate solution.

For Equation 2, an SMT solver might return `sat` with the model $x_{\mathsf{SMT}} = 1$ and, indeed, x_{SMT} serves as a counterexample showing that ρ_{NUM} is an invalid solution.

Remark: In our experiments none of the solvers ever returned `unknown` while validating candidate solutions.

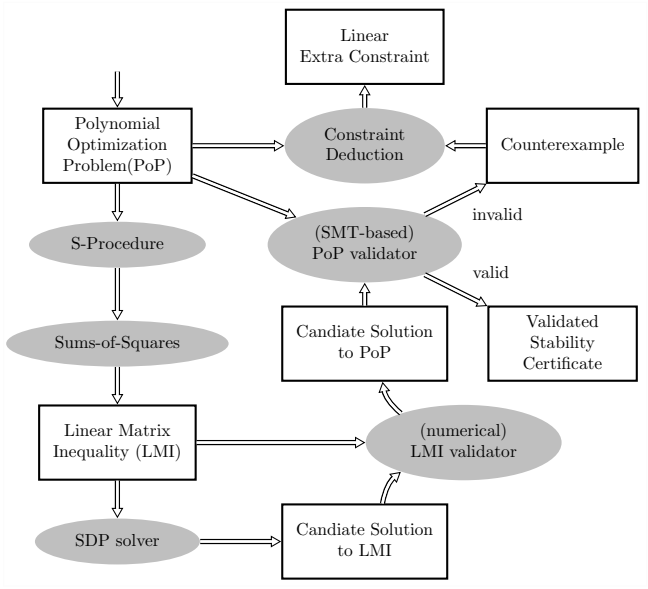


Figure 2: Overview of the Guiding Solver Engine

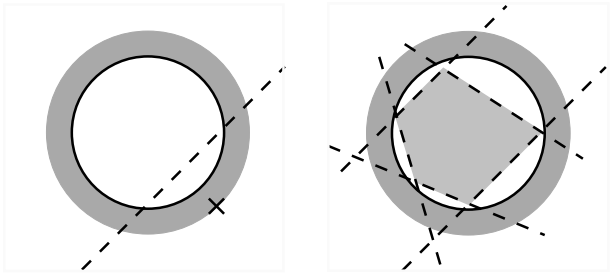
4. Guiding Solver Engine

In the previous section, we have shown how to use SMT solvers to validate solutions. Due to the nature of the numerical solvers (i.e., that they approximate solutions) we can expect many solutions returned by a numerical solver to be invalidated by the SMT solver. It even gets worse because of the way, the constraint systems are presented to the numerical solver. By default, the goal is to minimize $-\alpha$, β, and $-\gamma$ (see Theorem 1). In most cases this leads to a situation where the optimal feasible solution lies on the border of the solution space. That means that the optimization direction increases the attractiveness of invalid candidate solutions attractive.

To overcome this issue, we would like to gain knowledge from a counterexample. We propose to compute the error or residual at the point x_{SMT} and to extend the constraint system with an additional constraint.

Figure 2 gives an overview of the guiding (and validating) solver engine. It extends the proposed engine from Figure 1 by the constraint deduction with possibly exploiting a generated counterexample from the SMT-based validator. The idea is as follows: If the SMT solver returns `sat` and a model x_{SMT}, we construct an additional constraint that rules out the candidate solution ρ_{NUM}. The engine might also check for consistency and then be restarted using the extended constraint system. We call "adding of additional constraints" *guiding*, since it might not force the numerical solver to return valid solutions. Instead it may push the solver away from bad solutions.

Figure 3 sketches the idea. The inner circle represents the solution space while the gray outer ring represents the invalid candidate solutions that might be returned by the



(a) Solution space with a first counterexample

(b) Final solution space

Figure 3: Sketch of the solution space of a constraint system.

solver due to numerical issues and approximation inaccuracies. On the left side an "X" marks a possible candidate solution which is then successfully identified as invalid. The dashed line represents a constraint that is then added to rule out the invalid candidate solution – and hopefully other invalid candidate solutions as well. The right side figure represents a final result in which the original solution space is narrowed down to the most inner gray polygon due to four more constraints, that were added. Any solution that might be returned by the numerical solver that is within the gray polygon is valid. Even more, if again due to numerical issues, the solution lies "a bit outside" of the inner gray polygon but within the white circle, then this is not a problem since such a candidate would still satisfy the original constraint system.

Remark: Note, that the goal is to find solutions to the original constraint system. This fact allows us to exclude the added constraint from validation as they do not need to be satisfied since their sole purpose is to guide the numerical solver.

In the following, we explain how the additional constraints are constructed.

Guiding

By evaluating the all violated constraints $0 \leq f_i(\rho, x)$ using the counterexample x_{SMT} and the candidate solution ρ_{NUM}, we obtain

$$\mathsf{res}_i = f_i(\rho_{\mathsf{NUM}}, x_{\mathsf{SMT}}),$$

where $\mathsf{res}_i < 0$. Now, we can extend the constraint system from Example 1 by a constraint

$$|\mathsf{res}_i| \leq f_i(\rho, x_{\mathsf{SMT}}),$$

which is linear in ρ and does not contain quantifiers. In our running example, this leads to the constraint $|\mathsf{res}| \leq \rho \cdot x_{\mathsf{SMT}}^2$. The extended constraint system is as follows

Example 2:

$$\begin{aligned} \exists \rho \text{ such that } \quad & \forall x : true \Rightarrow 0 \preceq \rho \cdot x^2 \\ \wedge\ & |\mathsf{res}| \leq \rho \cdot x_{\mathsf{SMT}}^2 \end{aligned}$$

This extended constraint system can be handed back to the numerical solver, asking for a new ρ'_{NUM}. This will be again validated using the SMT solver. If we obtain a new counterexample x'_{SMT}, then we extend the problem again until either

- a valid solution is found,
- a maximum number of iterations has been reached, or
- the set of constraints with which we extended the original problem, contains a contradiction.

The last alternative might happen if the real solution space is empty or has a very small interior. Note that the additional constraints are all pure linear, unconditioned, and quantifier free. Thus linear programming (or again an SMT solver) might be used to further check contradiction-freeness, i.e., satisfiability of the additional constraints.

Such contradictions can be introduced because the constraints that we add are more restrictive than the original constraints. The rationale behind was the expectation that the numerical solver will achieve the same residual on the added constraint as on the original constraint. In this way the numerical solver will not return a solution to the extended constraint system. But – fortunately – that candidate satisfies the original constraint system. Thus we would have obtained a validatable solution.

Accelerated Guiding

We can accelerate this method further by alternatively extending the constraint system as follows:

Example 3:

$$\begin{aligned} & \exists \rho \text{ such that} \\ & \quad \forall x : true \Rightarrow 0 \preceq \rho \cdot x^2 \\ & \wedge\ \rho \cdot x_{\mathsf{SMT}}^2 \geq |\mathsf{res}| \cdot \mathsf{acc}, \end{aligned}$$

where $0 < \mathsf{acc}$ is an acceleration factor. Using $1 < \mathsf{acc}$ can lead to a validatable solution more quickly, but the drawback is that at the same time this narrows down the solution space faster. In the worst case, it might happen that the solution space becomes empty due to contradictions before a validatable solution is found. On the other hand, if $0 < \mathsf{acc} < 1$ is chosen, then the solution space is not narrowed down that fast. The drawback is that it might happen that there is nearly no increase in the quality of candidate solution returned by the numerical solver.

5. Example

Consider a sub-model of the hybrid system given in Figure 4 consisting of the two modes **Mode1** and **Mode2**, only.

$$f_i(x) = -c_i \cdot x$$

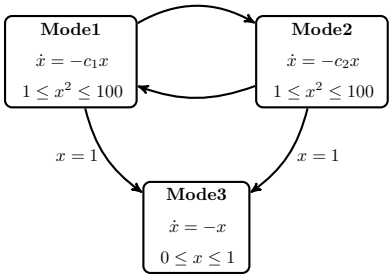


Figure 4: Bad Scaled Hybrid System

where $\mathsf{inv} = 1 \leq x^2 \wedge x^2 \leq 100$ is the invariant of both modes and arbitrary switching is allowed. Choosing the coefficients c_i to be badly scaled, e.g. $c_1 = 10^{-10}$ and $c_2 = 10^{10}$ together with using $V_3 \cdot x^2 + V_2 \cdot x + V_1$ as the template for both modes leads to the constraint system:

$$\begin{aligned}
\exists \alpha, \beta, \gamma : \\
&\mathsf{inv} \Rightarrow \alpha \cdot x^2 \preceq V_3 \cdot x^2 + V_2 \cdot x + V_1 \\
\wedge\ &\mathsf{inv} \Rightarrow V_3 \cdot x^2 + V_2 \cdot x + V_1 \preceq \beta \cdot x^2 \\
\wedge\ &\mathsf{inv} \Rightarrow \gamma \cdot x^2 \preceq 2c_1 V_3 \cdot x^2 + c_1 V_2 \cdot x \\
\wedge\ &\mathsf{inv} \Rightarrow \gamma \cdot x^2 \preceq 2c_2 V_3 \cdot x^2 + c_2 V_2 \cdot x \\
\wedge\ &0 < \alpha \\
\wedge\ &0 < \beta \\
\wedge\ &0 < \gamma.
\end{aligned}$$

One might easily see that $V_1 = V_2 = 0$, $V_3 = \alpha = \beta = 1$, $\gamma = 2c_1$ is a valid solution. Nevertheless, both SDPA and CSDP report "Lack of Progress" and thus the quality of the solution is unknown. Indeed, both solution have the problem that γ is slightly too large and CSDP additionally chooses V_3 and β slightly too small.

However, the SMT-base validator finds counterexamples. And after deducing two more constraints in case of SDPA and four more constraints in case of CSDP, both solvers – even though they still report "Lack of Progress" – return a valid solution.

Remark: The numerical validation is not helpful in this example since the eigenvalues are very close to 0. For the initial constraint system, i.e., without any additional constraints, computing the eigenvalues correctly indicates that the candidate solutions are not valid. In case of the final constraint system – the one obtained by guiding – computing eigenvalues falsely indicates that the solutions are invalid. Here using the SMT-based validator allows us to recover the solutions and lets us conclude stability of the hybrid system.

6. Conclusion

We have presented an approach to validate candidate solution obtained by numerically solvers. We have applied this approach in the process of solving constraint systems that arise in search for Lyapunov functions. The validation works by using SMT solvers to check satisfiability of the negated constraints. Such a validation is needed if the Lyapunov functions are reused, e.g. as barrier certificates proving unreachability of certain bad states, as a basis for composition, or as an estimator on the rate of convergence.

While this validation is already very helpful, one would also like to be able to gain knowledge from invalid candidate solutions. In Section 4, we have also shown a way to guide the numerical solver away from invalid candidate solutions. That way, it becomes more likely that "true solutions" are found. Future work includes to further analyze this approach and to give it a more theoretical basis. The relation to δ-satisfiability [23] is of special interest.

References

[1] A. Podelski and S. Wagner, "Region Stability Proofs for Hybrid Systems," in *Formal Modelling and Analysis of Timed Systems (FORMATS'07)*, ser. Lecture Nodes in Computer Science, vol. 4763. Springer, 2007, pp. 320–335.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.

[3] B. Borchers, "CSDP, a C Library for Semidefinite Programming." *Optimization Methods and Software*, vol. 10, pp. 613–623, 1999.

[4] K. Fujisawa, K. Nakata, M. Yamashita, and M. Fukuda, "SDPA Project : Solving Large-Scale Semidefinite Programs," *Journal of the Operations Research Society of Japan*, vol. 50, no. 4, pp. 278–298, Dec. 2007.

[5] E. Möhlmann and O. E. Theel, "Towards Automatic Detection of Implicit Equality Constraints in Stability Verification of Hybrid Systems," in *Proceedings of the 1th Congreso Nacional de Ingeniera Informática / Aplicaciones Informáticas y de Sistemas de Información, CoNaIISI 2013, November 21-22, 2013, Córdoba, Argentina*, M. M. Marciszack, R. M. Muñoz, and M. A. Groppo, Eds. Red de Carreras de Ingeniería Informática / Sistemas de Informatión (RIISIC), 2013.

[6] ——, "Stabhyli: A Tool for Automatic Stability Verification of Non-Linear Hybrid Systems," in *Proceedings of the 16th international conference on Hybrid systems: computation and control (HSCC'13)*, C. Belta and F. Ivancic, Eds. ACM, 2013, pp. 107–112.

[7] A. Tarski, "A Decision Method for Elementary Algebra and Geometry," *Bulletin of the American Mathematical Society*, vol. 59, 1951.

[8] L. De Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, ser. TACAS'08/ETAPS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 337–340.

[9] C. Barrett and C. Tinelli, "CVC3," in *Proceedings of the 19th International Conference on Computer Aided Verification (CAV'07)*, ser. Lecture Notes in Computer Science, W. Damm and H. Hermanns, Eds., vol. 4590. Springer-Verlag, Jul. 2007, pp. 298–302, berlin, Germany.

[10] C. Barrett, C. L. Conway, M. Deters, L. Hadarean, D. Jovanović, T. King, A. Reynolds, and C. Tinelli, "CVC4," in *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 171–177.

[11] S. Gao, M. Ganai, F. Ivancic, A. Gupta, S. Sankaranarayanan, and E. Clarke, "Integrating ICP and LRA solvers for deciding nonlinear real arithmetic problems," in *Formal Methods in Computer-Aided Design (FMCAD'10), 2010*, Oct 2010, pp. 81–89.

[12] P. Nuzzo, A. A. A. Puggelli, S. A. Seshia, and A. L. Sangiovanni-Vincentelli, "CalCS: SMT Solving for Nonlinear Convex Constraints," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-100, Jun 2010. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-100.html

[13] J. Oehlerking, H. Burchardt, and O. E. Theel, "Fully Automated Stability Verification for Piecewise Affine Systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Nodes in Computer Science, vol. 4416. Springer, 2007, pp. 741–745.

[14] S. Ratschan and Z. She, "Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-Like Functions," *SIAM J. Control and Optimization*, vol. 48, no. 7, pp. 4377–4394, 2010.

[15] P. S. Duggirala and S. Mitra, "Lyapunov abstractions for inevitability of hybrid systems," in *Proceedings of the 15th international conference on Hybrid systems: computation and control (HSCC'12)*. ACM, 2012, pp. 115–124.

[16] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the 13th Conference on Computer-Aided Control System Design (CACSD'04)*, Taipei, Taiwan, 2004.

[17] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, `http://arxiv.org/abs/1310.4716`, 2013.

[18] C. Jansson, D. Chaykin, and C. Keil, "Rigorous error bounds for the optimal value in semidefinite programming," *SIAM J. Numerical Analysis*, vol. 46, no. 1, pp. 180–200, 2007.

[19] J. Oehlerking, "Decomposition of Stability Proofs for Hybrid Systems," Ph.D. dissertation, Carl von Ossietzky University of Oldenburg, Department of Computer Science, Oldenburg, Germany, 2011.

[20] M. Lyapunov, "Problème général de la stabilité du movement," in *Ann. Fac. Sci. Toulouse, 9.* Université Paul Sabatier, 1907, pp. 203–474.

[21] S. Prajna and A. Papachristodoulou, "Analysis of switched and hybrid systems - beyond piecewise quadratic methods," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 4, June 2003, pp. 2779–2784 vol.4.

[22] J. Wilkinson, *Rounding Errors in Algebraic Processes*. Dover Publications, Incorporated, 1994.

[23] S. Gao, J. Avigad, and E. M. Clarke, "δ-Complete Decision Procedures for Satisfiability over the Reals," in *Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR'12)*, ser. Lecture Notes in Computer Science, B. Gramlich, D. Miller, and U. Sattler, Eds., vol. 7364. Springer, 2012, pp. 286–300.

Factores que Determinan la Adopción de la Trazabilidad de Requerimientos Un Análisis Cualitativo

Juan C. Vázquez, Leticia E. Constable, Brenda E. Meloni, Juan F. Giró
Departamento de Ingeniería en Sistemas de Información
Facultad Regional Córdoba, Universidad Tecnológica Nacional
Maestro López esq. Cruz Roja, 5016 Córdoba, Argentina
{jcjvazquez, leticiaconstable, bemeloni, juanfgiro}@gmail.com

Resumen

La Ingeniería de Software y los estándares internacionales, enfatizan los beneficios de contar con una efectiva trazabilidad de requerimientos en los procesos de desarrollo de sistemas y software. La escasez de evidencias de que las bondades indicadas se logran conseguir realmente en la Industria, motiva el estudio de cómo se usa la trazabilidad, de si se está efectivamente usando y de la conveniencia o no invertir recursos en desarrollarla. Para ello se han ideado modelos de predicción del éxito en su implementación. Los datos necesarios para la construcción y prueba de estos modelos, debieron ser obtenidos por medio de encuestas diseñadas a tal efecto, las que guiaron entrevistas efectuadas a diversos actores involucrados en la industria del software. Un análisis cualitativo de las respuestas obtenidas de los profesionales, brinda algunos indicios adicionales a los modelos de predicción de lo que puede estar pasando en la disciplina, respecto de este importante tema que impacta en la calidad de los productos generados.

1. Introducción

La trazabilidad de requerimientos en los proyectos de desarrollo de software y sistemas, ha sido una práctica recomendada desde que se instauró como disciplina la *Ingeniería de Software*, el intento de la industria y la academia de lidiar con la llamada *crisis del software* (proyectos excedidos en sus tiempos, en sus presupuestos y que en muchos casos no responden adecuadamente a sus objetivos) ya identificada en la década de 1960.

Los estándares iniciales de calidad CMM[1] (siglas de *Capability Maturity Model*) [1] desarrollados por el Instituto de Ingeniería de Software de la Universidad Carnegie-Mellon, en su primera versión de 1987 ya incluían la trazabilidad de requerimientos como una condición necesaria para alcanzar el nivel 2 de madurez, primer nivel con buenas prácticas establecidas.

Como lo señalaba el citado estándar (Fig. 1), el seguimiento y la supervisión de proyectos de software fueron considerados aspectos claves para lograr una

> *The purpose of Software Project Tracking and Oversight is to establish adequate visibility into actual progress so that management can take effective actions when the software project's performance deviates significantly from the software plans.*

Figura 1: Texto extraído del CMM original.

gestión oportuna y efectiva, ya que debería brindar información sobre desviaciones respecto del plan de trabajo. Como el trabajo a realizar por el proyecto está definido por la especificación de los requerimientos del sistema (lo que debe hacer, las propiedades que debe tener y las restricciones a las que está sujeto), lo anterior necesariamente involucra el seguimiento de estos requerimientos a través de todas las etapas de construcción.

En [2] se señala la importancia de efectuar la trazabilidad de requerimientos en un proceso de desarrollo, pensada como la habilidad de describir y seguir la vida de un requerimiento, bidireccionalmente a lo largo de todas las etapas del ciclo de vida del proyecto; no sólo como una herramienta útil de control de gestión durante el desarrollo del proyecto, sino como un dispositivo de reaseguramiento de la calidad del producto terminado, ya que garantiza la completitud (muestra que todos los requerimientos han sido tomados en cuenta), el cumplimiento de objetivos (permite demostrar que el sistema hace lo que debe) y facilita su posterior mantenimiento (al documentar en forma clara cómo y dónde se implementa cada requisito, sencillez en la determinación de impactos por cambios requeridos y

[1] CMM, que luego evolucionó a SW-CMM y se generalizó con CMMI, establece para las organizaciones una escala de cinco niveles de madurez referida a cómo realizan sus procesos: 1: Inicial, 2: Administrado, 3: Definido, 4: Cuantificado, 5: Optimizado.

certera medición de esfuerzos de desarrollo para realizar los mismos, entre otras cosas).

A pesar de lo expuesto y de los años transcurridos, la trazabilidad de requerimientos sigue siendo un desafío en los proyectos de desarrollo de software y sistemas [3][4]: en muchos de ellos se planea cómo hacerla y no se logra completar, en otros (los menos) se completa pero con mayor esfuerzo al originalmente previsto y en la mayoría, sólo se la plantea en forma parcial o, sencillamente, se decide no implementarla.

Para tomar contacto real con esta problemática, nuestro equipo de investigación diseñó, desarrolló y llevó a cabo una serie de encuestas sobre la trazabilidad de requerimientos, entrevistando a distintos actores de la industria del software (empresarios, líderes de proyecto, arquitectos, programadores, etc.) de la ciudad de Córdoba, Argentina, donde existe una importante cantidad de empresas de desarrollo de software de origen nacional y varias filiales de empresas extranjeras; también se consultó a estudiantes del último año de carreras de informática de distintas universidades locales y se concretó finalmente una encuesta en línea, exclusivamente para Directivos de una asociación que agrupa a gran cantidad de empresas del rubro y consultoras, radicadas en la provincia.

En base a la bibliografía existente (tanto libros como artículos especializados) y con los datos obtenidos de las primeras encuestas, se planteó la hipótesis de que el hecho de que la trazabilidad sea tan esquiva al momento de incluirla en un proyecto de desarrollo no es una casualidad; esto es *deben existir factores que condicionan el éxito de la trazabilidad de requerimientos en los proyectos de desarrollo de sistemas/software*, y bajo esta premisa, se propuso identificarlos y diseñar modelos de predicción de la conveniencia de invertir tiempo y dinero en su desarrollo y puesta en marcha.

La definición de estos factores y de las entidades a los cuales ellos se refieren, fue el núcleo de nuestra investigación durante tres años y motivó una sucesión de trabajos presentados con anterioridad [5–9], en los cuales se describen sucesivos modelos que fueron comparados y progresivamente mejorados, destinados a anticipar los resultados de la trazabilidad de requerimientos en los procesos de desarrollo.

En los mencionados trabajos se reconoció que los factores buscados estaban asociados a las dimensiones de tres entidades principales estrechamente relacionadas entre sí: *a*) el propio *producto software*, *b*) el *proyecto* y *c*) la *organización*. El proyecto responde a alguna metodología de desarrollo y la organización constituye el escenario en el que el proyecto es desarrollado hasta alcanzar el producto. *El cliente*, sujeto para el cual se desarrolla el sistema y quien lo utilizará, claramente es una entidad importante, pero no se lo distingue en los modelos definidos porque se lo considera implícitamente incluido en el producto a través de la especificación de requerimientos de sus necesidades.

Agrupados por entidad se muestran en las tablas 1 y 2, los ocho factores que fueron propuestos para predecir la conveniencia de realizar la trazabilidad de requerimientos en proyectos de desarrollo de software.

Tabla 1: Definición de entidades, factores, variables asociadas, descripciones y referencias.

Entidad	Factor	V	Descripción de la variable primitiva de referencia ***Vr***	Intervalo de ***Vr***
Producto	Tamaño	***t***	Puntos de Función ***PF***	100–1000
	Vigencia	***v***	Vida útil ***VU*** [años]	0,5 – 10
	Reutilización	***r***	Futura reutilización ***RE*** [%]	0 – 80
	Confiabilidad	***c***	Indicador confiabilidad ***CO*** (*)	0 – 5
Proyecto	Plazo	***p***	Duración proyecto ***DP*** [años]	0 – 5
	Equipo	***e***	Efectividad del equipo ***EF*** (**)	1 – 5
Organiza-ción	Madurez	***m***	Nivel de madurez ***CMMI***	1 – 5
	Dependencia	***d***	Nivel de autonomía ***NA*** (***)	1 – 5

Tablas 2: Definición de los indicadores ***CO*** (*), ***EF*** (**), Nivel de Madurez ***CMMI*** y ***NA*** (***)

Confiabilidad ***CO***	
0	No importante
1	Baja
2	Media
3	Alta
4	Muy alta
5	Absoluta

Efectividad ***EF***	
1	Pobre
2	Baja
3	Media
4	Alta
5	Muy alta

Madurez ***CMMI***	
1	Inicial
2	Gestionado
3	Definido
4	Cuant.Gestión
5	Optimizado

Autonomía ***NA***	
1	Independiente
2	Normas propias
3	Normas clientes
4	Nor. casa matriz
5	Combina 3 y 4

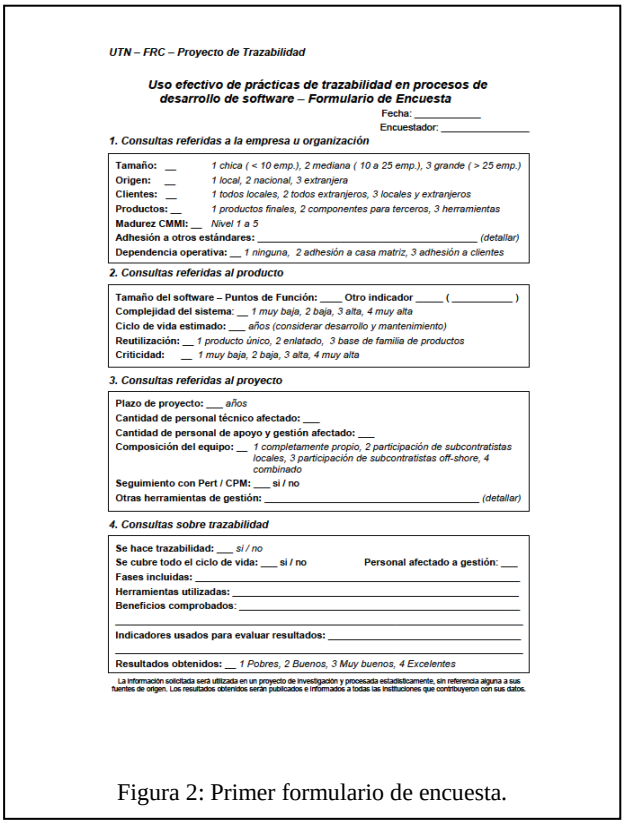
En [5] y [8] se presenta la justificación y el alcance de los factores propuestos junto con un detalle de las diversas métricas empleadas en los distintos modelos. La determinación de estos factores motivó la generación de nuevas encuestas que, en forma más específica, brindaran datos sobre los mismos. En los trabajos anteriores referenciados, se efectúa un *análisis cuantitativo* utilizando los datos reales relevados a través de las encuestas y sus combinaciones, como fuente de información para determinar la validez de los modelos; éstos van desde la representación de los datos en

diagramas radar, donde se define un perímetro discriminante que separa los proyectos en los que es conveniente implementar la trazabilidad de aquellos en los que no lo es, hasta el entrenamiento de una red neuronal artificial que, aprendiendo de los datos suministrados se transforma en una herramienta de decisión más precisa sobre la conveniencia de trazar o no requerimientos.

A diferencia del enfoque cuantitativo extensamente tratado en [5–9], el presente trabajo pretende reflexionar sobre las respuestas obtenidas en las distintas encuestas realizadas, haciendo un *análisis cualitativo*. En la sección 2 se describen los instrumentos de adquisición de datos (encuestas) y el por qué fueron diseñados, en la sección 3 se realiza el análisis cualitativo de las respuestas obtenidas y finalmente, se presentan las conclusiones y algunas recomendaciones, en la sección 4.

2. Instrumentos de Adquisición de Datos y Metodología.

Con el objetivo de tomar contacto con la realidad de la industria del software, se diseñó un primer formulario



UTN – FRC – Proyecto de Trazabilidad

Uso efectivo de prácticas de trazabilidad en procesos de desarrollo de software – Formulario de Encuesta

Fecha: _____
Encuestador: _____

1. Consultas referidas a la empresa u organización

Tamaño: __ *1 chica (< 10 emp.), 2 mediana (10 a 25 emp.), 3 grande (> 25 emp.)*
Origen: __ *1 local, 2 nacional, 3 extranjera*
Clientes: __ *1 todos locales, 2 todos extranjeros, 3 locales y extranjeros*
Productos: __ *1 productos finales, 2 componentes para terceros, 3 herramientas*
Madurez CMMI: __ *Nivel 1 a 5*
Adhesión a otros estándares: _____ *(detallar)*
Dependencia operativa: __ *1 ninguna, 2 adhesión a casa matriz, 3 adhesión a clientes*

2. Consultas referidas al producto

Tamaño del software – Puntos de Función: ____ **Otro indicador** _____ (_____)
Complejidad del sistema: __ *1 muy baja, 2 baja, 3 alta, 4 muy alta*
Ciclo de vida estimado: ___ *años (considerar desarrollo y mantenimiento)*
Reutilización: __ *1 producto único, 2 enlatado, 3 base de familia de productos*
Criticidad: __ *1 muy baja, 2 baja, 3 alta, 4 muy alta*

3. Consultas referidas al proyecto

Plazo de proyecto: ___ *años*
Cantidad de personal técnico afectado: ___
Cantidad de personal de apoyo y gestión afectado: ___
Composición del equipo: __ *1 completamente propio, 2 participación de subcontratistas locales, 3 participación de subcontratistas off-shore, 4 combinado*
Seguimiento con Pert / CPM: ___ **si / no**
Otras herramientas de gestión: _____ *(detallar)*

4. Consultas sobre trazabilidad

Se hace trazabilidad: ___ *si / no*
Se cubre todo el ciclo de vida: ___ **si / no** **Personal afectado a gestión:** ___
Fases incluidas: _____
Herramientas utilizadas: _____
Beneficios comprobados: _____
_____
Indicadores usados para evaluar resultados: _____
_____
Resultados obtenidos: __ *1 Pobres, 2 Buenos, 3 Muy buenos, 4 Excelentes*

La información solicitada será utilizada en un proyecto de investigación y procesada estadísticamente, sin referencia alguna a sus fuentes de origen. Los resultados obtenidos serán publicados e informados a todas las instituciones que contribuyeron con sus datos.

Figura 2: Primer formulario de encuesta.

de encuesta para guiar entrevistas personales con distintos actores; participaron de la misma profesores de las universidades locales que desarrollan su actividad profesional en el ámbito de la Informática, desarrolladores de software independientes, programadores, analistas funcionales, arquitectos de software y líderes de proyecto que trabajan en empresas de capital nacional y en filiales de empresas internacionales asentadas en nuestra región, investigadores y alumnos graduados de posgrado, etcétera, todos de la ciudad de Córdoba, Argentina.

Las entrevistas fueron llevadas a cabo personalmente por integrantes de nuestro equipo de investigación, en locaciones convenidas previamente con los entrevistados; algunas en nuestro laboratorio de la Universidad, otras en los lugares de trabajo de los encuestados y unas pocas, en lugares neutrales elegidos por los profesionales.

En la primera encuesta se consultan distintas cuestiones (Fig. 2) referidas a cuatro aspectos principales:

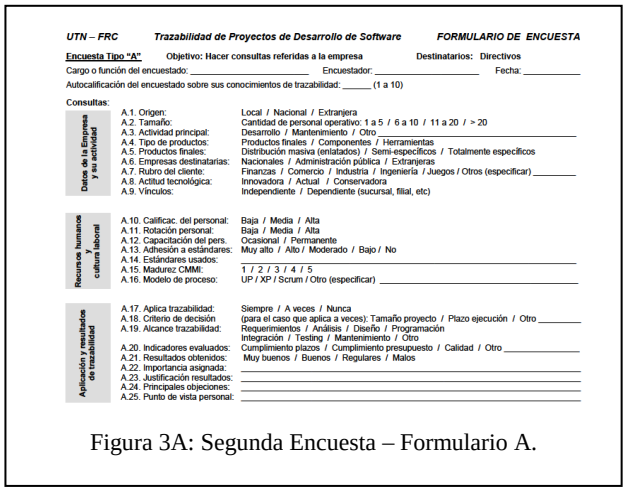
- Empresa u Organización
- Producto a construir por la organización
- Proyecto de desarrollo del producto
- Trazabilidad de requerimientos del producto

Las entrevistas se plantearon como charlas informales sobre la problemática de la trazabilidad de requerimientos; siempre al iniciar la conversación, el encuestador le informa al participante el objetivo de la investigación y el tratamiento a nivel estadístico al que se someterán los datos suministrados, asegurando el anonimato y la confidencialidad, tanto en lo referido a la organización a la cual pertenece como a su identidad personal. Paso seguido, se inicia la serie de consultas sobre cada uno de los ítems de la encuesta y es el encuestador quien registra en el formulario los datos tabulados y al dorso, comentarios generales vertidos por el encuestado.

Luego de entrevistar una buena cantidad de profesionales, que presentaban desconocimiento total o parcial respecto a la trazabilidad de requisitos, tuvimos que coincidir con [3] en que uno de los factores más relevantes que determina el uso o no de la trazabilidad, es el ¡conocimiento de su existencia!; esto es tan evidente que en general ni siquiera es mencionado en la bibliografía como un factor de importancia, pero en las primeras encuestas surgió como algo determinante.

Por otro lado, se obtuvieron en no pocas oportunidades dentro de una misma organización, respuestas muy distintas en cuanto al beneficio que brinda la trazabilidad, las herramientas que se utilizan para hacerla y los problemas que se identifican durante su implementación, según el entrevistado desempeñara un cargo de dirección, de liderazgo o gestión de proyectos, o trabajara como personal de desarrollo en tareas operativas.

Estas diferencias, y el desconocimiento evidenciado por varios entrevistados del concepto de trazabilidad, mostró la necesidad de efectuar una preselección de los encuestados y de refinar el formulario de encuesta, lo que llevó a diseñar distintas versiones (figuras 3A, 3B y 3C):



UTN – FRC *Trazabilidad de Proyectos de Desarrollo de Software* *FORMULARIO DE ENCUESTA*

Encuesta Tipo "A" **Objetivo: Hacer consultas referidas a la empresa** **Destinatarios: Directivos**

Cargo o función del encuestado: _____ Encuestador: _____ Fecha: _____

Autocalificación del encuestado sobre sus conocimientos de trazabilidad: _____ (1 a 10)

Consultas:

Datos de la Empresa y su actividad

A.1. Origen: Local / Nacional / Extranjera
A.2. Tamaño: Cantidad de personal operativo: 1 a 5 / 6 a 10 / 11 a 20 / > 20
A.3. Actividad principal: Desarrollo / Mantenimiento / Otro _____
A.4. Tipo de productos: Productos finales / Componentes / Herramientas
A.5. Productos finales: Distribución masiva (enlatados) / Semi-específicos / Totalmente específicos
A.6. Empresas destinatarias: Nacionales / Administración pública / Extranjeras
A.7. Rubro del cliente: Finanzas / Comercio / Industria / Ingeniería / Juegos / Otros (especificar) _____
A.8. Actitud tecnológica: Innovadora / Actual / Conservadora
A.9. Vínculos: Independiente / Dependiente (sucursal, filial, etc)

Recursos humanos y cultura laboral

A.10. Calificac. del personal: Baja / Media / Alta
A.11. Rotación personal: Baja / Media / Alta
A.12. Capacitación del pers. Ocasional / Permanente
A.13. Adhesión a estándares: Muy alto / Alto / Moderado / Bajo / No
A.14. Estándares usados: _____
A.15. Madurez CMMI: 1 / 2 / 3 / 4 / 5
A.16. Modelo de proceso: UP / XP / Scrum / Otro (especificar) _____

Aplicación y resultados de trazabilidad

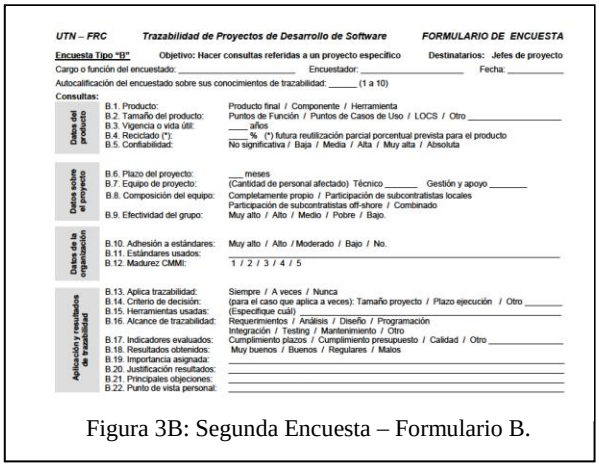
A.17. Aplica trazabilidad: Siempre / A veces / Nunca
A.18. Criterio de decisión (para el caso que aplica a veces): Tamaño proyecto / Plazo ejecución / Otro _____
A.19. Alcance trazabilidad: Requerimientos / Análisis / Diseño / Programación
Integración / Testing / Mantenimiento / Otro
A.20. Indicadores evaluados: Cumplimiento plazos / Cumplimiento presupuesto / Calidad / Otro _____
A.21. Resultados obtenidos: Muy buenos / Buenos / Regulares / Malos
A.22. Importancia asignada: _____
A.23. Justificación resultados: _____
A.24. Principales objeciones: _____
A.25. Punto de vista personal: _____

Figura 3A: Segunda Encuesta – Formulario A.

- Formulario A: para obtener datos generales de la organización, desde sus Directivos,
- Formulario B: para obtener datos específicos de un proyecto, desde su Jefe o Líder de Proyecto, y
- Formulario C: para entrevistar al personal técnico y operativo, sobre sus tareas de desarrollo.

Al utilizar esta segunda versión de las encuestas, ya se tenían identificadas las unidades de análisis (entidades) y los factores que se consideran en nuestros modelos como determinantes de la conveniencia de efectuar la trazabilidad de requerimientos en un proyecto de desarrollo; por ello los ítems a preguntar puede verse que son más específicos (Fig. 3A/B/C).

Con estos nuevos formularios, se realizaron nuevamente entrevistas a distintos actores de la industria del software con la misma metodología que se describió para las primeras encuestas, pero previamente se los seleccionó al preguntar si conocían y usaban trazabilidad de requerimientos, encuestando a aquellos de los que se obtenía una respuesta afirmativa. De la población encuestada, 24% resultaron propietarios o directivos de compañías de software, 26% líderes de proyecto y 25% analistas o programadores. Debieron ser descartadas por inconsistencias (y porque a pesar de declarar previamente hacer trazabilidad en sus proyectos, se vio que no era trazabilidad de requerimientos lo que se realizaba), el 25% de las encuestas



UTN – FRC *Trazabilidad de Proyectos de Desarrollo de Software* *FORMULARIO DE ENCUESTA*

Encuesta Tipo "B" **Objetivo: Hacer consultas referidas a un proyecto específico** **Destinatarios: Jefes de proyecto**

Cargo o función del encuestado: _____ Encuestador: _____ Fecha: _____

Autocalificación del encuestado sobre sus conocimientos de trazabilidad: _____ (1 a 10)

Consultas:

Datos del producto

B.1. Producto: Producto final / Componente / Herramienta
B.2. Tamaño del producto: Puntos de Función / Puntos de Casos de Uso / LOCS / Otro _____
B.3. Vigencia o vida útil: ____ años
B.4. Reciclado (*): ____% (*) futura reutilización parcial porcentual prevista para el producto
B.5. Confiabilidad: No significativa / Baja / Media / Alta / Muy alta / Absoluta

Datos sobre el proyecto

B.6. Plazo del proyecto: ____ meses
B.7. Equipo de proyecto: (Cantidad de personal afectado) Técnico _____ Gestión y apoyo _____
B.8. Composición del equipo: Completamente propio / Participación de subcontratistas locales
Participación de subcontratistas off-shore / Combinado
B.9. Efectividad del grupo: Muy alto / Alto / Medio / Pobre / Bajo.

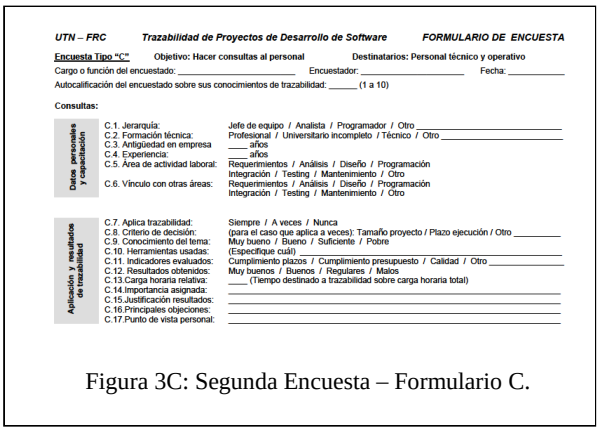
Datos de la organización

B.10. Adhesión a estándares: Muy alto / Alto / Moderado / Bajo / No.
B.11. Estándares usados: _____
B.12. Madurez CMMI: 1 / 2 / 3 / 4 / 5

Aplicación y resultados de trazabilidad

B.13. Aplica trazabilidad: Siempre / A veces / Nunca
B.14. Criterio de decisión: (para el caso que aplica a veces): Tamaño proyecto / Plazo ejecución / Otro _____
B.15. Herramientas usadas: (Especifique cuál) _____
B.16. Alcance de trazabilidad: Requerimientos / Análisis / Diseño / Programación
Integración / Testing / Mantenimiento / Otro
B.17. Indicadores evaluados: Cumplimiento plazos / Cumplimiento presupuesto / Calidad / Otro _____
B.18. Resultados obtenidos: Muy buenos / Buenos / Regulares / Malos
B.19. Importancia asignada: _____
B.20. Justificación resultados: _____
B.21. Principales objeciones: _____
B.22. Punto de vista personal: _____

Figura 3B: Segunda Encuesta – Formulario B.



UTN – FRC *Trazabilidad de Proyectos de Desarrollo de Software* *FORMULARIO DE ENCUESTA*

Encuesta Tipo "C" **Objetivo: Hacer consultas al personal** **Destinatarios: Personal técnico y operativo**

Cargo o función del encuestado: _____ Encuestador: _____ Fecha: _____

Autocalificación del encuestado sobre sus conocimientos de trazabilidad: _____ (1 a 10)

Consultas:

Datos personales y capacitación

C.1. Jerarquía: Jefe de equipo / Analista / Programador / Otro _____
C.2. Formación técnica: Profesional / Universitario incompleto / Técnico / Otro _____
C.3. Antigüedad en empresa ____ años
C.4. Experiencia: ____ años
C.5. Área de actividad laboral: Requerimientos / Análisis / Diseño / Programación
Integración / Testing / Mantenimiento / Otro
C.6. Vínculo con otras áreas: Requerimientos / Análisis / Diseño / Programación
Integración / Testing / Mantenimiento / Otro

Aplicación y resultados de trazabilidad

C.7. Aplica trazabilidad: Siempre / A veces / Nunca
C.8. Criterio de decisión: (para el caso que aplica a veces): Tamaño proyecto / Plazo ejecución / Otro _____
C.9. Conocimiento del tema: Muy bueno / Bueno / Suficiente / Pobre
C.10. Herramientas usadas: (Especifique cuál) _____
C.11. Indicadores evaluados: Cumplimiento plazos / Cumplimiento presupuesto / Calidad / Otro _____
C.12. Resultados obtenidos: Muy buenos / Buenos / Regulares / Malos
C.13.Carga horaria relativa: ____ (Tiempo destinado a trazabilidad sobre carga horaria total)
C.14.Importancia asignada: _____
C.15.Justificación resultados: _____
C.16.Principales objeciones: _____
C.17.Punto de vista personal: _____

Figura 3C: Segunda Encuesta – Formulario C.

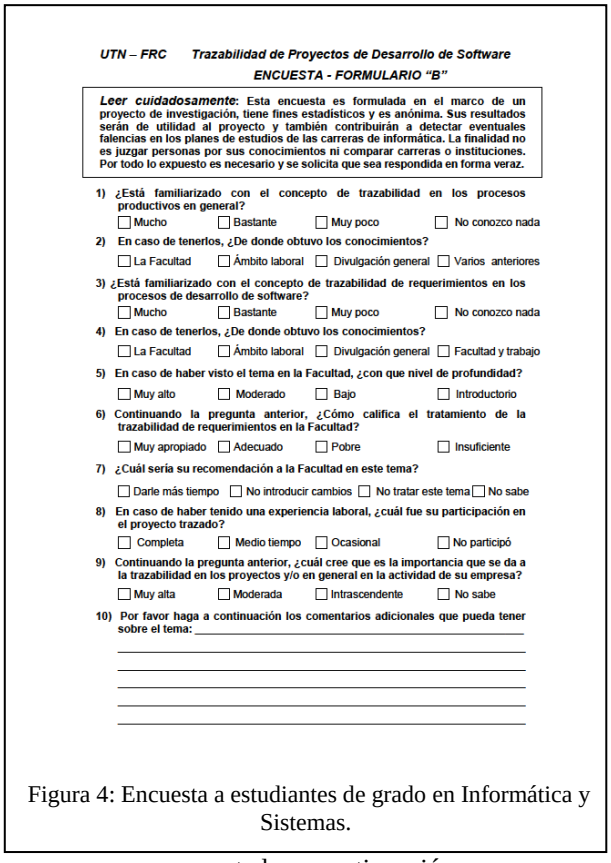
También, y habiéndose detectado como determinante que el conocimiento de la existencia del concepto de trazabilidad es decisivo a la hora de evaluarla como una herramienta a incluir en el proyecto, se pensó pertinente consultar a los futuros profesionales de sistemas y software. A partir de esta reflexión surge la iniciativa de conocer la preparación y opinión que los estudiantes de carreras informáticas tienen sobre el tema. Para ello se seleccionaron preguntas específicas, se diseñó un formulario (Fig. 4) y se propusieron como destinatarios a los alumnos que estaban cursando el último cuatrimestre de sus carreras en tres instituciones, dos de ellas Ingeniería en Sistemas y la tercera de Ingeniería Informática. Las tres instituciones tienen acreditadas sus carreras ante CONEAU, por lo que sus programas cubren los mismos contenidos mínimos.

Esta encuesta presenta nueve preguntas y se completa con un espacio que invita a hacer algún comentario o reflexión sobre el tema tratado. Se ejecutaron en el aula, en breves tiempos cedidos amablemente por nuestros colegas docentes de las tres Universidades.

En la realización de encuestas existe siempre la posibilidad de que no sean respondidas con la debida seriedad, o que se lo haga temiendo delatar ignorancia o desconocimiento, y se falte a la verdad. Por ello, previamente se les explicó a los estudiantes el objeto de la misma y se les aclaró que se respetaría el secreto

estadístico, por lo cual las encuestas no fueron nominadas en ninguna forma; se contestaron anónimamente en los formularios entregados.

Tal como puede observarse (Fig. 4), las preguntas pueden reunirse según su orientación y objetivos en 4



UTN – FRC *Trazabilidad de Proyectos de Desarrollo de Software*

ENCUESTA - FORMULARIO "B"

Leer cuidadosamente: Esta encuesta es formulada en el marco de un proyecto de investigación, tiene fines estadísticos y es anónima. Sus resultados serán de utilidad al proyecto y también contribuirán a detectar eventuales falencias en los planes de estudios de las carreras de informática. La finalidad no es juzgar personas por sus conocimientos ni comparar carreras o instituciones. Por todo lo expuesto es necesario y se solicita que sea respondida en forma veraz.

1) ¿Está familiarizado con el concepto de trazabilidad en los procesos productivos en general?
☐ Mucho ☐ Bastante ☐ Muy poco ☐ No conozco nada

2) En caso de tenerlos, ¿De donde obtuvo los conocimientos?
☐ La Facultad ☐ Ámbito laboral ☐ Divulgación general ☐ Varios anteriores

3) ¿Está familiarizado con el concepto de trazabilidad de requerimientos en los procesos de desarrollo de software?
☐ Mucho ☐ Bastante ☐ Muy poco ☐ No conozco nada

4) En caso de tenerlos, ¿De donde obtuvo los conocimientos?
☐ La Facultad ☐ Ámbito laboral ☐ Divulgación general ☐ Facultad y trabajo

5) En caso de haber visto el tema en la Facultad, ¿con que nivel de profundidad?
☐ Muy alto ☐ Moderado ☐ Bajo ☐ Introductorio

6) Continuando la pregunta anterior, ¿Cómo califica el tratamiento de la trazabilidad de requerimientos en la Facultad?
☐ Muy apropiado ☐ Adecuado ☐ Pobre ☐ Insuficiente

7) ¿Cuál sería su recomendación a la Facultad en este tema?
☐ Darle más tiempo ☐ No introducir cambios ☐ No tratar este tema ☐ No sabe

8) En caso de haber tenido una experiencia laboral, ¿cuál fue su participación en el proyecto trazado?
☐ Completa ☐ Medio tiempo ☐ Ocasional ☐ No participó

9) Continuando la pregunta anterior, ¿cuál cree que es la importancia que se da a la trazabilidad en los proyectos y/o en general en la actividad de su empresa?
☐ Muy alta ☐ Moderada ☐ Intrascendente ☐ No sabe

10) Por favor haga a continuación los comentarios adicionales que pueda tener sobre el tema: _____

Figura 4: Encuesta a estudiantes de grado en Informática y Sistemas.

grupos, que son comentados a continuación:

a) Preguntas 1 y 2: Conocimientos previos sobre la trazabilidad en general y el motivo o fuentes de donde fueron obtenidos, incluyendo el ámbito académico o una experiencia laboral.

b) Preguntas 3 y 4: Preguntas similares a las anteriores, pero específicamente referidas a la trazabilidad de requerimientos en procesos de desarrollo de software.

c) Preguntas 5, 6 y 7: Opinión del estudiante sobre la forma en que se trató el tema en su ámbito académico, considerando profundidad e interés despertado. Se incluye el pedido que haga una recomendación al respecto.

d) Preguntas 8 y 9: Consulta sobre si tuvo una experiencia laboral en este tema: su participación y la importancia que se le daba a la trazabilidad en esa empresa.

Como se anticipó, las preguntas referidas al ámbito académico fueron pensadas para evaluar el conocimiento recibido e interés despertado en los futuros ingenieros con respecto a la trazabilidad. Por otra parte, las preguntas referidas al ámbito laboral son pertinentes por dos motivos: *i*) todos los estudiantes deben pasar por una breve experiencia laboral obligatoria de 200 hs, denominada Práctica Profesional Supervisada (PPS) y *ii*) en las carreras afines a la Ingeniería de Sistemas la mayoría de los estudiantes realiza pasantías en empresas del medio en el último año. Esta circunstancia generó mucha expectativa en la inclusión de estas preguntas por la posibilidad de disponer de una mirada alternativa a la del ámbito laboral.

La población de encuestados se formó con estudiantes que cursaban el quinto año de la carrera y fue en todos los casos realizada poco antes de finalizado el último cuatrimestre. Cabe también agregar que los estudiantes no estaban advertidos sobre la encuesta, lo que impidió que se prepararan especialmente para responderla.

Finalmente, y con el objetivo de recabar la mayor cantidad de datos posible, se construyó una encuesta en línea utilizando herramientas de Google y fueron invitadas todas las empresas de la mayor asociación de empresas de software de la región por correo electrónico, a que la contesten; los resultados de esas encuestas son registrados por Google en una planilla de cálculo que luego sólo el equipo de investigación puede acceder. Las consultas presentes en esta encuesta electrónica, son similares a las que se encuentran en el Formulario A de la segunda encuesta (Fig. 3A).

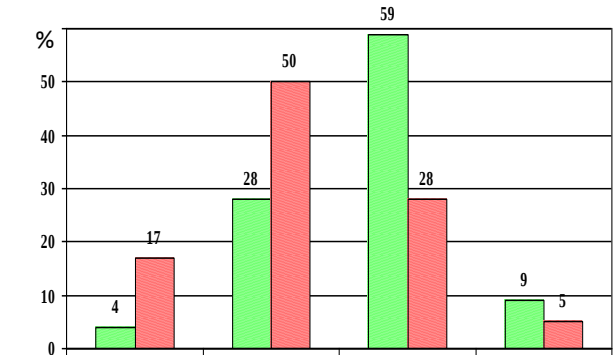
Tanto para la segunda encuesta, como con la encuesta a estudiantes y la última en línea, se enfatizó el carácter anónimo de las mismas y el manejo confidencial que se haría de los datos obtenidos, sólo utilizados en forma de estadísticas en los trabajos cuantitativos. Esto fue necesario porque durante la primera encuesta de prueba, nos dimos cuenta que es un tema muy sensible y hasta incómodo para los profesionales (y para sus organizaciones) explicar sus problemáticas internas y los particulares modos de hacer desarrollo que cada uno implementa.

3. Análisis Cualitativo de las Encuestas Realizadas

En primer lugar, y aunque algo ya se indicó anteriormente, nos resultó algo sorprendente el hecho que muchos de los profesionales entrevistados durante la primer encuesta, desconocieran el concepto de trazabilidad de requerimientos. La mayoría de estos profesionales tienen larga trayectoria en el desarrollo de sistemas y software, y poseen alguna titulación

universitaria en Informática. Aunque no se consultó en forma específica sobre el mayor nivel de estudios alcanzado, pudo determinarse a través de las conversaciones mantenidas con los entrevistados que, en general, una idea clara del concepto de trazabilidad de requerimientos y de su importancia en los proyectos de desarrollo, estaba incorporada en aquellos que habían realizado estudios de posgrado en Sistemas o en Software en los últimos años.

Utilizando datos del segundo juego de encuestas, debe indicarse que casi todos los directivos entrevistados contestaron que hacen trazabilidad de requerimientos *siempre*, salvo uno que indicó que en su organización la hacen a veces, dependiendo de qué tan bien estuvieran especificados los requerimientos (si bien su organización certifica ISO y no certifica CMMI, el encuestado estima que cumple el nivel 3); sin embargo, en el 70% de los casos se indicó que la trazabilidad era llevada a cabo en dos o tres etapas del desarrollo del proyecto, generalmente programación, pruebas y mantenimiento. Siendo que la trazabilidad de requerimientos debe ser bidireccional, desde los requerimientos elicitados del cliente hasta el código y los casos de prueba realizados, pasando por todas las etapas del proceso de desarrollo (especificación de requerimientos, análisis, diseño y definición de la arquitectura, codificación, pruebas, integración, implantación, mantenimiento, documentación, etc.), cualquiera sea la metodología de desarrollo utilizada, estas respuestas inducen a pensar que, en la mayoría de los casos, la Dirección piensa en trazabilidad como una herramienta de control de las etapas estrictamente relacionadas con la programación y no tiene gran precisión de cómo se hace trazabilidad en su organización y del grado en que se logra efectivamente.



	A: Mucho	B: Bastante	C: Muy	D: Nada
Preg.1	4,0	28,0	59,0	9,0
Preg.3	17,0	50,0	28,0	5,0

Figura 5: Representación de respuestas a las preguntas 1 y 3.

En el segmento de los líderes de proyecto, el 100% declara que aplica trazabilidad de los requerimientos *siempre*, aunque sólo el 40% dice que involucra en esto a todas las etapas del proceso y el otro 60% asegura utilizarlo fundamentalmente en tres etapas (no siempre las mismas) referidas en la mayoría de los casos a las actividades de programación y pruebas (el 30% de ellos trabajan en empresas con certificación CMMI nivel 3 y 5). Ven como indicadores evaluados por la trazabilidad, las horas de trabajo del equipo, el cumplimiento de plazos y presupuestos, la calidad del producto logrado, esto es, la están utilizando sobre todo como una herramienta de gestión de proyecto (en las etapas más técnicas-computacionales de desarrollo) que ordena el trabajo y los mantiene informados sobre el avance. En algún caso incluso se la considera una herramienta útil de versionado. Cabe recordar que éste fue (Fig. 1) el objetivo inicial que en los modelos CMM se le otorgó a la trazabilidad: seguimiento y control del proyecto. Los líderes también señalan la necesidad de la existencia de *un puesto* específico con al menos una persona exclusivamente asignada para llevar adelante el control de implementación de trazabilidad de requerimientos. Esta visión es consistente con lo indicado por los directivos de las organizaciones.

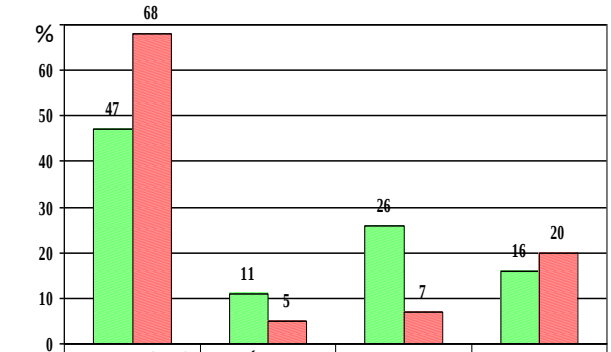
Por su lado, los operarios del proyecto (programadores, analistas, verificadores y validadores) encuestados, todos resultaron profesionales universitarios con un promedio de 4 años de experiencia; afirmaron que implementar la trazabilidad ocupa desde el 5% y hasta el 30% de su tiempo laboral; en general reconocen ampliamente su importancia y la vinculan con la gestión de las tareas que les ocupan, aunque se quejan del tiempo que les insume; algunos confiesan que es una tarea que se deja de lado cuando los tiempos apremian y que se revisa y completa, luego de entregar el sistema funcionando, en el mejor de los casos. Es de notar que muchos de ellos señalan que les *ordena* adecuadamente el trabajo, lo que coincide con el uso de herramienta de gestión que le asignan los líderes de proyecto y el 70% expresa que la herramienta utilizada (muy variadas, tanto adquiridas como propias) para hacer y controlar la trazabilidad es inadecuada, difícil de usar y que implica gran cantidad de documentación.

En cuanto a los estudiantes encuestados que están cursando su último año de carrera de Ingeniería en Sistemas o en Informática, los siguientes cuadros muestran las respuestas a las preguntas agrupadas según se indica en su título.

Como puede observarse en la Fig. 5, los alumnos que están bastantes o muy familiarizados con el concepto de trazabilidad en general alcanzan un 32%, mientras que este valor llega al 67% cuando se trata de la trazabilidad de requerimientos en el desarrollo de software.

En la Fig. 6 se agrupan las respuestas a las preguntas No. 2 y 4, referidas a donde se obtuvieron los conocimientos sobre el concepto de trazabilidad en

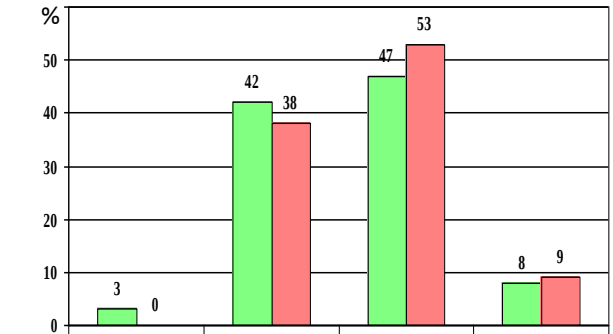
general y de la trazabilidad de requerimientos en particular.



	A: Facultad	B: Ámbito	C:	D: Varios
Preg.2	47	11	26	16
Preg.4	68	5	7	20

Figura 6: Representación de respuestas a las preguntas 2 y 4.

Las respuestas indican que en ambos casos (47% y 68%) el conocimiento sobre la trazabilidad proviene en su mayoría de la Universidad, mientras que el obtenido en el medio laboral es muy escaso (11% y 5%). La respuesta "D" incluye los dos ámbitos, Universidad y trabajo, por lo que podrían ser sumadas a ambos parciales, con lo que se tiene: Universidad: 63% y 88%, Trabajo: 27% y 25%. De estas respuestas se podría concluir que el tema es bien tratado en la Universidad, aunque una cantidad significativa de alumnos se familiarizaron con la trazabilidad en el desarrollo de software fuera de ella, en el ámbito laboral y divulgación (37% y 12%).



	A: Muy	B: Adecuado	C: Pobre	D:
Preg.5	3,0	42,0	47,0	8,0
Preg.6	0,0	38,0	53,0	9,0

Figura 7: Representación de respuestas a las preguntas 5 y 6.

En la Fig. 7 se reproducen las respuestas a las preguntas No. 5 y 6, en las que se consultó el nivel de profundidad con que se presentó el tema y si se lo hizo de manera adecuada. Esto último implica que las clases permitieron reconocer su importancia y despertaron suficiente interés por el mismo.

Las respuestas revelan poca conformidad de los estudiantes con la profundidad y forma en que fue tratado el tema en clase, lo que es evidenciado por el 53% y 62% de opiniones que corresponde a las calificaciones de "pobre" e "insuficiente".

Las respuestas a la pregunta No. 7, que consulta sobre recomendaciones que podrían hacer los estudiantes a la Facultad sobre el tratamiento de la trazabilidad en clase, son presentadas en forma separada por tener respuestas muy específicas.

Para ello se adopta un gráfico circular, en el que pone en evidencia que una gran mayoría (78%) considera que el tema merece que se le dedique más tiempo.

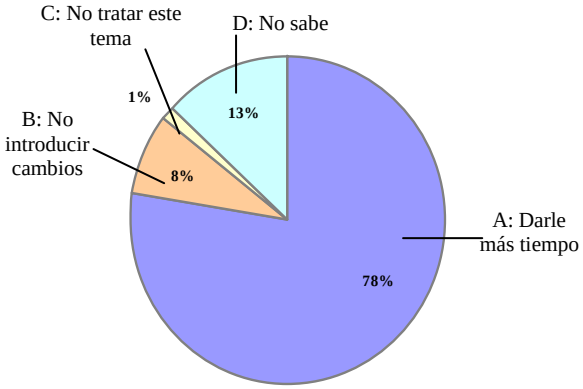


Figura 8: Representación de respuestas a la pregunta 7.

Por otra parte, los que recomiendan no introducir cambios representan el 8%, los que no tienen una opinión formada el 13% y solo un alumno (< 1 %) propuso no tratar el tema.

La pregunta No. 8 vuelve al ámbito laboral y consulta sobre la participación de estudiantes en proyectos trazables y el nivel de dedicación con que se lo hizo. Aquí nuevamente se recurrió a un diagrama circular, con resultados que son mostrados en la Fig. 9.

Al analizar estas respuestas es conveniente recordar a

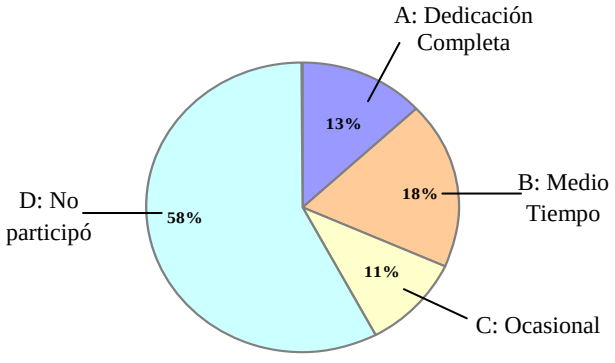


Figura 9: Representación de las respuestas a la pregunta 8.

esta altura de la carrera todos los alumnos tuvieron una

experiencia laboral, ya sea a través de pasantitas o de la práctica profesional supervisada (PPS). Ésta última es muy breve, con una extensión de 200 hs. Llama la atención los dos extremos: que haya un 58% de los encuestados que no tuvieron ningún contacto laboral con la trazabilidad y haya otro 31% que tuvieron una experiencia de semi-dedicación o dedicación completa en este tema. Si se agregan las experiencias ocasionales se llega a que el 42% de los estudiantes encontraron en la industria alguna oportunidad de tomar contacto con la trazabilidad, aunque sea puntual o muy breve.

De cualquier manera, que haya un 58% de los próximos egresados que no hayan tenido ninguna oportunidad de tomar contacto con la trazabilidad en la industria abre algunos interrogantes, que deberían esclarecerse a partir de algunas preguntas más específicas en futuras encuestas.

La última pregunta, la número 9, consulta sobre la importancia que considera que se brinda a la trazabilidad en el ámbito laboral que tuvo oportunidad de conocer. Los resultados obtenidos se muestran en el diagrama circular de la Fig. 10.

Aquí las respuestas muestran una correlación con las obtenidas para la pregunta anterior referida a la participación laboral en un proyecto trazable. La población de estudiantes que dijo haber tenido una dedicación completa (13%) prácticamente coincide con la

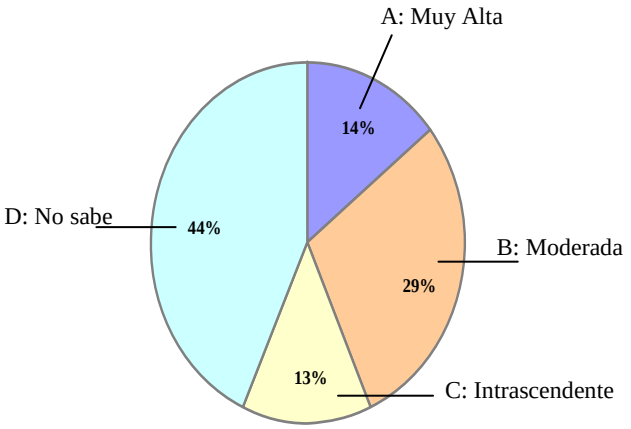


Figura 10: Representación de respuestas a la pregunta 9.

que le asigna mucha importancia a la trazabilidad en el ámbito laboral (14%) y los que tuvieron una dedicación parcial u ocasional ((18 + 11)%) coinciden con los que asignan moderada importancia a la trazabilidad (29%). Por último, el 58% que no participó laboralmente de un proyecto trazable coincide con los que dicen no saber sobre la importancia asignada a la trazabilidad (44%) o la consideran intrascendente (13%). Esta coincidencia merece un análisis más profundo para determinar si encierra alguna conclusión valida o simplemente invalida las respuestas obtenidas.

4. Conclusiones y Recomendaciones

La trazabilidad de requerimientos permanece como un desafío en los proyectos de desarrollo de software. En [3] se señala que es una actividad *optativa* en un proyecto, por lo cual su ejecución requiere de conocimientos y de una *decisión*; los modelos que se han diseñado y construido en nuestro proyecto, tratan de ayudar a tomarla con base en datos objetivos de su conveniencia.

En las empresas relevadas, se la emplea como herramienta de gestión de proyecto con énfasis en las etapas relacionadas con el código, lo que hace que no se obtengan todos los beneficios previstos por la teoría y los estándares. Se la confunde en muchos casos con herramientas de versionado.

Las encuestas a las empresas de la región Córdoba, indican que si bien hace años que el concepto de trazabilidad fue reconocido como importante por la Ingeniería de Software, existe gran cantidad de profesionales que no han tomado contacto con la temática, salvo aquellos que están saliendo actualmente de las Universidades o, habiendo egresado anteriormente, han efectuado estudios de posgrado. Esto permite sugerir que un mejoramiento en el tratamiento del tema dentro de las organizaciones se logrará naturalmente con el tiempo, pero puede reforzarse en el corto plazo con el apoyo de las empresas a sus empleados, para que efectúen estudios de posgrado (maestrías y doctorados).

La implantación de la trazabilidad de requerimientos es una decisión de la Dirección; para hacerla efectiva ésta debe asignar los recursos necesarios para la capacitación de su personal, para reforzar la I+D interna con el objeto de lograr la adaptación y/o el desarrollo de herramientas adecuadas y para determinar en cada proyecto un puesto específico de tiempo completo destinado a su diseño, establecimiento, control y gestión.

5. Agradecimientos.

Específicamente queremos agradecer a la gran cantidad de colegas, profesionales y estudiantes que de forma anónima y desinteresada contribuyeron con el trabajo, respondiendo las encuestas y participando de las numerosas entrevistas efectuadas.

También agradecemos a la Secretaría de Ciencia y Técnica de la Universidad Tecnológica Nacional, por el soporte financiero brindado al proyecto EIUTNCO 1214: *Aseguramiento de la Trazabilidad en Proyectos de Desarrollo de Sistemas y Software*.

6. Referencias

1. Equipo del Producto CMMI: CMM v1.1 (1987) y CMMI-DEV v1.3 (2010), CMU-SEI, USA.
2. Rilling, J., Charland, P., Witte, R.; Traceability in Software Engineering - Past, Present and Future, CASCON 2007 Workshop Report, IBM Technical Report: TR-74-211, Oct. 2007.
3. Blaauboer F., Sikkel K., Aydin M.; Deciding to Adopt Requirements Traceability in Practice, Libro: Advanced Information Systems Engineering, Serie: LNCS. Proceedings of 19th International Conference, CAiSE 2007, Trondheim, Norway. Pp 294-308. Springer-Verlag. Berlín, Alemania. (2007).
4. Kannenberg, A., Saiedian, H.: Why Software Requirements Traceability Remains a Challenge. CrossTalk: The Journal of Defense Software Engineering. July/August, 14-19. Utha, USA. (2009).
5. Giró, J., Vázquez, J., Meloni, B., Constable, L., Jornet, A.: Hacia una respuesta al interrogante de si será factible trazar un cierto proyecto de desarrollo de software. Informe Técnico 2012/01, Proyecto 1214, SCyT, FRC, UTN (2012).
6. Giró, J., Vázquez, J., Meloni, B., Constable, L., Jornet, A.: Modelos para anticipar la factibilidad de que un proyecto de desarrollo de software sea trazable. Workshop de Ingeniería de Software, CACIC 2011. Universidad Nacional de La Plata, 837-846 (2011).
7. Giró, J., Vázquez, J., Meloni, B., Constable, L., Jornet, A.: Uso del Análisis ROC para anticipar la conveniencia de trazar proyectos de software. Workshop de Ingeniería de Software, CACIC 2012. Universidad Nacional del Sur, Ciudad de Bahía Blanca (2012).
8. Giró, J., Vázquez, J., Meloni, B., Constable, L.; Evaluación de variantes en modelo destinado a anticipar la conveniencia de trazar proyectos de software, Workshop de Ingeniería de Software (WIS), CACIC 2013, CAECE, Ciudad de Mar del Plata, (2013).
9. Giró, J., Vázquez, J., Meloni, B., Constable, L.; Comparación de modelos destinados a anticipar la conveniencia de trazar proyectos de desarrollo de software, CoNaIISI 2013, Fac. Regional Córdoba UTN, Ciudad de Córdoba, (2013).

Modelo de referencia para trazar procesos Scrum: Moodle como un caso de estudio

Roberto Nazareno[1, 2], Silvio Gonnet[1, 3], Horacio Leone[1, 3]
1 INGAR (CONICET – UTN). Avellaneda 3657, Santa Fe, Argentina
2 Universidad Nacional de La Rioja, La Rioja, Argentina
3 Facultad Regional Santa Fe, Universidad Tecnológica Nacional, Santa Fe, Argentina
{rnazareno, sgonnet, hleone }@santafe-conicet.gov.ar

Abstract

La trazabilidad es considerada como una herramienta importante para construir sistemas de calidad, incluso utilizando procesos ágiles. Sin embargo, las prácticas de trazabilidad no pueden ser aplicadas en procesos ágiles de igual forma que en los procesos "pesados". Consecuentemente, es importante el desarrollo de modelos de referencia que consideren los procesos ágiles. En este trabajo se propone un modelo de referencia basado en las prácticas de Scrum. El principal objetivo es representar la trazabilidad entre los artefactos generados durante el proceso Scrum. En particular, se consideraron las siguientes preguntas de competencia: ¿cuál stakeholder es responsable de un artefacto?, si existiera un impedimento durante el desarrollo, ¿cuáles artefactos y eventos estarían involucrados?, si hay un cambio en un requerimiento, ¿cuáles artefactos, stakeholders, o eventos estarían involucrados? Las respuestas a estas preguntas facilitarán las tareas de trazabilidad que afectan al análisis de impacto de cambios, al acuerdo del producto, a la obediencia en el proceso, a la reproducibilidad en la línea base, y al aprendizaje organizacional.

1. Introducción

La trazabilidad es considerada en metodologías ágiles como un aspecto fundamental a estudiar para desarrollar sistemas de calidad [1], [2]. Trazabilidad es: (i) el grado en el cual se puede establecer una relación entre dos o más productos del proceso de desarrollo (traza), especialmente productos que posean una relación de predecesor-sucesor o superior-subordinado; o (ii) el grado en el cual se puede establecer la razón de la existencia de cada elemento en un proceso de desarrollo de software[3]. Los procesos ágiles ocurren en entornos donde no es frecuente encontrar un documento de especificación de requerimiento. Para capturar requerimientos muchas organizaciones utilizan Backlog Items, User Stories, Test, entre otras herramientas. Este escenario no permite aplicar técnicas clásicas de trazabilidad [4] come se venía aplicando en métodos pesados. En consecuencia, es fundamental desarrollar modelos de referencia que permitan definir las adquisiciones y representaciones de trazas bajo la perspectiva ágil.

Los métodos ágiles [5], [6] están centrados en el desarrollo, siendo su objetivo proveer una respuesta rápida a los cambios en los requerimientos, a las personas que componen los equipos y a los problemas que surgen durante el proceso de desarrollo. En la actualidad, una de las metodologías de desarrollo de software ágil más utilizada es Scrum [7][8].

Gotel y colab. definen un modelo genérico del proceso de trazabilidad [9]. Tal proceso está conformado por un conjunto de actividades claves como lo son: la planificación y gestión de la estrategia, la creación, el uso, y el mantenimiento de la trazabilidad. Por su parte, cada tarea está dividida en subactividades, por ejemplo la creación está dividida en Adquisición, Representación, Almacenamiento, y Validación de trazas. Este proceso sirve como guía para determinar si existen tareas de trazabilidad que sean compatibles con los métodos ágiles, más precisamente con Scrum. La trazabilidad puede ser usada en las metodologías ágiles para determinar si los requerimientos fueron cumplidos y además ayuda a evaluar el impacto de los cambios, entre otros beneficios[10]. Este trabajo propone un modelo de referencia basado en las prácticas de Scrum. Teniendo como objetivo principal representar trazas entre los artefactos generados durante el proceso Scrum. Esto nos permitirá responder: ¿Qué artefactos son origen y fuente de los enlaces de trazabilidad?, ¿qué información de trazabilidad es representada por esos artefactos?, ¿en qué eventos es adquirida o

modificada esta traza?, ¿cuáles stakeholders están involucrados en la creación, uso y mantenimiento de un artefacto? Las respuestas a estas preguntas facilitarán las tareas de trazabilidad que influyen directamente en el análisis del impacto de cambios, conformidad en el producto, obediencia del proceso, responsabilidad del proyecto, reproducibilidad de la línea base y aprendizaje organizacional [2].

Para refinar y validar nuestra propuesta, se emplea como caso de estudio el desarrollo de Moodle [11]. Moodle es una plataforma de aprendizaje la cual utiliza Scrum como proceso de desarrollo, para llevar a cabo este proyecto. Esta plataforma está diseñada con el objetivo de brindar un sistema integrado que permita crear ambientes de aprendizaje personalizados. Así mismo, se encuentra desarrollada y dirigido por un equipo perteneciente a la organización, por la comunidad internacional y además por una red de socios certificados.

Este trabajo se encuentra organizado de la siguiente manera: la siguiente sección reúne una revisión de trabajos relacionados. La sección 3 presenta el modelo de referencia para la trazabilidad en Scrum. Este modelo está estructurado en un conjunto de vistas dadas por los Artefactos, Eventos y Roles involucrados en el desarrollo de Scrum. En la sección 4 se detalla un caso de estudio basado en el proceso de desarrollo de Moodle. Por último, en la sección 5 son expuestos las conclusiones y trabajos futuros.

2. Trabajos Relacionados

Como punto de partida para la propuesta del modelo de trazabilidad se instancia el metamodelo de trazabilidad, ilustrado en la Figura 1 [4]. El metamodelo representa distintas dimensiones de la información de trazabilidad: ¿Qué información de trazas es representada?, ¿Dónde la información de trazabilidad es representada?, ¿Cuándo esta información fue capturada o modificada? ¿Cuáles Stakeholders están involucrados en la creación, uso o modificación de un objeto?. Estas dimensiones son representadas por tres entidades principales:

- Source: el cual hace referencia a aquellos artefactos donde la información de la trazabilidad es mantenida. Donde se enfatiza el rol de la gestión de documentos con los métodos y herramientas de trazabilidad.
- Stakeholder: son aquellos agentes involucrados en la gestión de la trazabilidad. Los roles de los stakeholders con respecto a la trazabilidad se vuelven más importantes a medida que el proyecto se extiende en el tiempo y se vuelve más complejo.
- Object: representan entradas y salidas del proceso de desarrollo de sistema y son creados por tareas organizacionales, como actividades de análisis o diseño.

Cada entidad y enlace en el metamodelo puede ser especializado e instanciado para crear modelos de trazabilidad específicos al proyecto u organización. Así mismo, nuestra propuesta instancia este metamodelo para representar la trazabilidad entre los artefactos generados, eventos, y roles involucrados en el proceso Scrum.

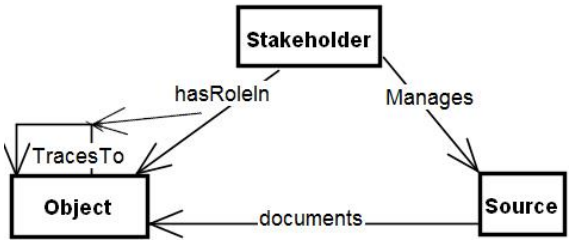


Figura 1 Metamodelo de trazabilidad [4]

Es bien conocido que los desarrolladores emplean una gran cantidad de esfuerzo en documentar y mantener la información de trazabilidad. Los estudios existentes realizados sobre trazabilidad de software tradicional, están enfocados en reducir costos asociados con este esfuerzo manual, desarrollando ayudas automáticas en el establecimiento y mantenimiento de enlaces de trazabilidad entre artefactos de software [12], [13]. La trazabilidad automática aplica técnicas de recuperación de información para generar enlaces candidatos. De esta forma se busca reducir considerablemente el efecto de los enfoques manuales, para construir y mantener una matriz de requerimientos de traza, y de igual forma proporcionar trazabilidad en los documentos legados [14], [15]. Estas propuestas también reflejan que implementando trazabilidad automática, el trabajo de los desarrolladores no debería verse incrementado de manera significativa. En este trabajo, se propone un enfoque diferente, se introduce un modelo inicial de trazabilidad de Scrum que integra los diferentes artefactos de Scrum (instanciados desde Object y Source, en la Figura 1) e identifica los eventos de Scrum como instancias de la relación TraceTo (Figura 1). A partir del modelo de Scrum propuesto podría implementarse distintas herramientas de recuperación, evaluación y validación de trazas.

Otra contribución relevante en el área de trazabilidad es la propuesta de Gotel y colab. [9], donde se presenta un modelo genérico del proceso de trazabilidad. El modelo describe un conjunto de actividades claves de la trazabilidad como lo son:

- Planificación y gestión de la estrategia: son aquellas decisiones realizadas con el fin de diseñar soluciones adecuadas de trazabilidad. Posteriormente, los recursos deben ser evaluados, para definir un modelo o solución de trazabilidad información que contemple un control y de herramientas necesarias. Continuando con la implementación del modelo, y finalmente con la evaluación de si el proceso de trazabilidad cumplió las necesidades inicialmente planteadas.
- Creación: es la actividad general de asociar dos artefactos (fuente y destino) a través de un enlace entre ellos con propósitos de trazado. Permitiendo su representación, almacenamiento y su posterior validación.
- Mantenimiento: son aquellas actividades relacionadas con la actualización de las trazas preexistentes como aquellos cambios realizados a los artefactos y la evolución de la trazabilidad.
- Uso: son aquellas actividades asociadas con la disposición de trazas para el uso y apoyo de actividades y tareas de verificación, validación, entre otras.

En la Figura 2 se muestra un diagrama de estado con el modelo del proceso de trazabilidad, en el cual, son resaltadas aquellas actividades de trazabilidad que serán tratadas en nuestro modelo de Scrum. El proceso puede comenzar de dos maneras, cuando una traza es prevista o cuando es requerida. Este proceso continúa con la creación de trazas (Creating), en este estado se representan las tareas de: adquisición (Acquiring Trace), representación (Representing Trace), el posterior almacenamiento de las trazas (Storing Trace) y la validación de las mismas (Validating Trace). Una vez creada se puede continuar con el mantenimiento (Maintaining) o con el uso de esas trazas (Using). Luego proseguir con la actividad de evaluación, por último, el proceso finaliza cuando una traza es retirada o cuando el proyecto es archivado. Sin duda, para que este modelo sea viable en un ambiente ágil, como lo es Scrum, algunas actividades deben ser automatizadas, como la recuperación y almacenamiento de las trazas. A fines prácticos, el mantenimiento y la planificación de forma manual o semiautomática no podrían ser tenidas en cuenta, debido a que el trabajo del equipo de desarrollo se vería incrementado. Este modelo hace explicitas ciertas contradicciones entre los principios ágiles [5] y la trazabilidad, denotando la necesidad de un modelo que permita recuperar la trazabilidad sin sobrecarga en el proceso de desarrollo.

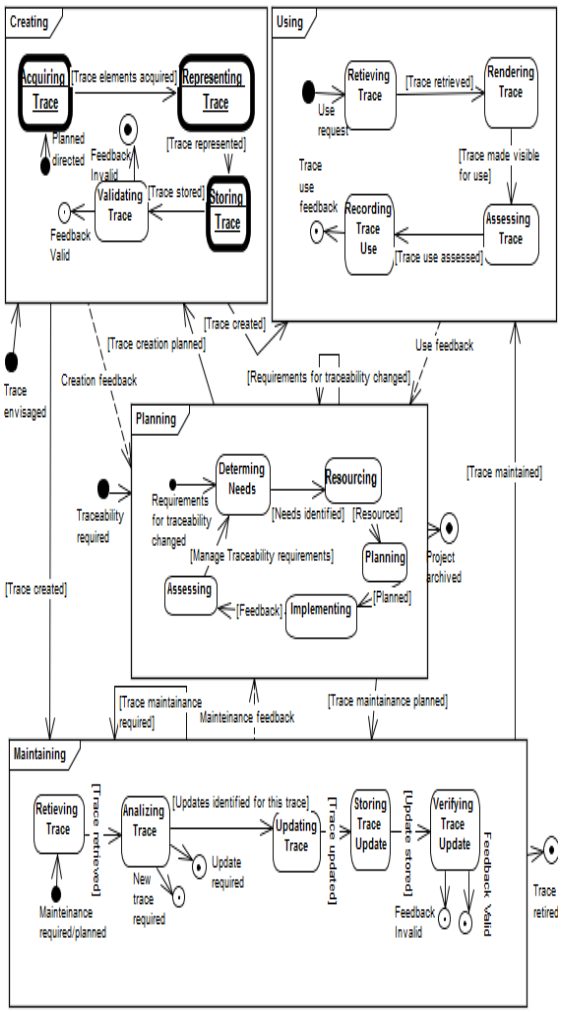


Figura 2 Modelo del proceso de trazabilidad[9]

Las necesidades en los proyectos ágiles varían de uno a otro y por lo tanto sus restricciones de trazabilidad difieren de forma significativa. Espinoza y Garbajosa [2], remarcan este hecho proponiendo un enfoque basado en modelos para la planificación de la trazabilidad en los proyectos ágiles. Este autor aboga por realizar trazabilidad ajustada a los objetivos, permitiendo al usuario establecer enlaces de trazabilidad. Los tipos básicos de trazabilidad son: trazabilidad hacia adelante y hacia atrás, también vertical y horizontal, y las combinaciones de ambos pares. En este trabajo se utilizaran algunas de estas combinaciones para poder responder a las preguntas de competencias planteadas en la sección 1.

3. Modelo de referencia de trazabilidad en Scrum

En esta sección, se propone un modelo de referencia de trazabilidad de Scrum. Este está centrado en los constructores esenciales de Scrum: roles, eventos, artefactos y las reglas que permiten asociar estos conceptos. Estos conceptos son instanciados desde los conceptos de Object, Source, y la relación TraceTo del metamodelo de trazabilidad (Figura 1) propuesto por Ramesh y Jarke [4].

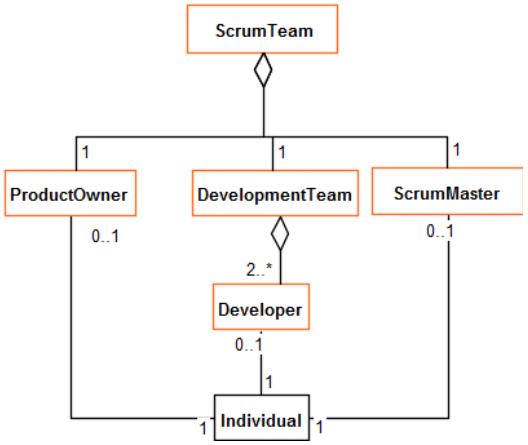


Figura 3 Roles de Scrum

Las reglas de Scrum asocian a los eventos, roles, y artefactos que gobiernan las relaciones entre ellos. Esto nos permite obtener vistas simplificadas del modelo, tales como: roles con artefactos, artefactos con eventos, y eventos con roles. Estos conceptos y sus relaciones nos permiten describir, de manera abstracta, las relaciones de trazabilidad implícitas en el proceso.

Los roles describen responsabilidades y niveles de autoridad de individuos, o grupos de individuos, que participan durante el proceso. En Scrum se define los roles: ScrumTeam, ProductOwner, ScrumMaster, y DevelopmentTeam (Figura 3). Para mantener la trazabilidad de un proceso ágil es preciso modelar los roles debido a que son estos los que tendrán la responsabilidad de llevar a cabo diferentes actividades durante la ejecución del proceso.

Los eventos (Event en Figura 4) son ocurrencias en el tiempo de un conjunto de reuniones o acciones, utilizados con el objetivo de sincronizar las diferentes etapas por las que atraviesa un proceso Scrum. Son importantes en el modelado debido a que es en ellos donde la trazabilidad tiene lugar en el espacio temporal. En la Figura 4 se representa la relación entre los eventos mediante la asociación predecessor – successor. Toda reunión ocurre en un determinado intervalo de tiempo, representado por TimeBox.

Los eventos son definidos como instancias de la relación TraceTo (Figura 1), ellos enlazan dos instancias del concepto Object (Figura 1), conformando la tripleta de elementos de traza, los artefactos Source y Target, y el enlace de traza.

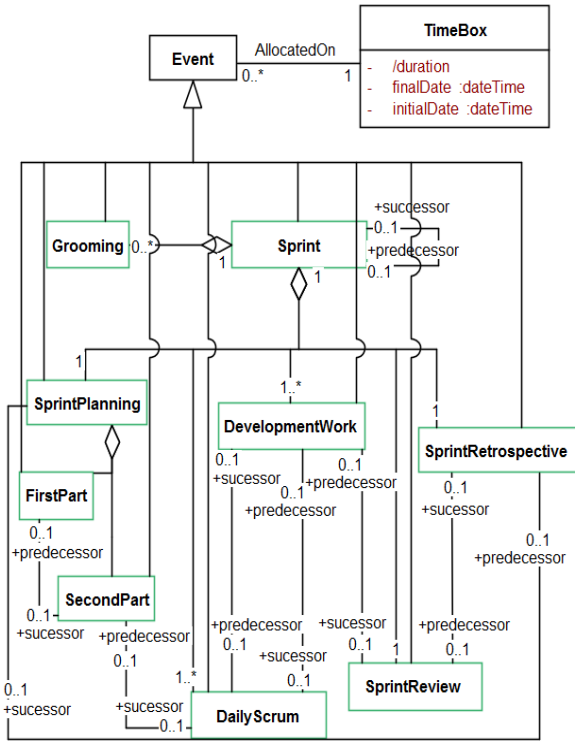


Figura 4 Eventos de Scrum

En este caso los eventos relacionan artefactos (Figura 5), además, los artefactos generados por un evento son enlazados con artefactos fuente (Figura 4) usados en el evento. Por definición un artefacto es la definición de un producto de trabajo[14]. Los artefactos pueden estar compuestos por otros artefactos y son productos de trabajo concretos consumidos, producidos o modificados por los distintos eventos del proceso Scrum. Así de esta manera, podemos referirnos a la actividad de creación de traza (Creating, Figura 2) para definir un artefacto fuente, un artefacto destino, y un enlace entre ellos. Por ejemplo, en la Figura 5 se define una tripleta de trazabilidad entre ProductBacklogItem (Source), ProductIncrement (Target), y Feedback (TraceTo). Por lo tanto, se podría almacenar, y/o recuperar, esta traza que representaría que requerimiento fue desarrollado en una versión del producto.

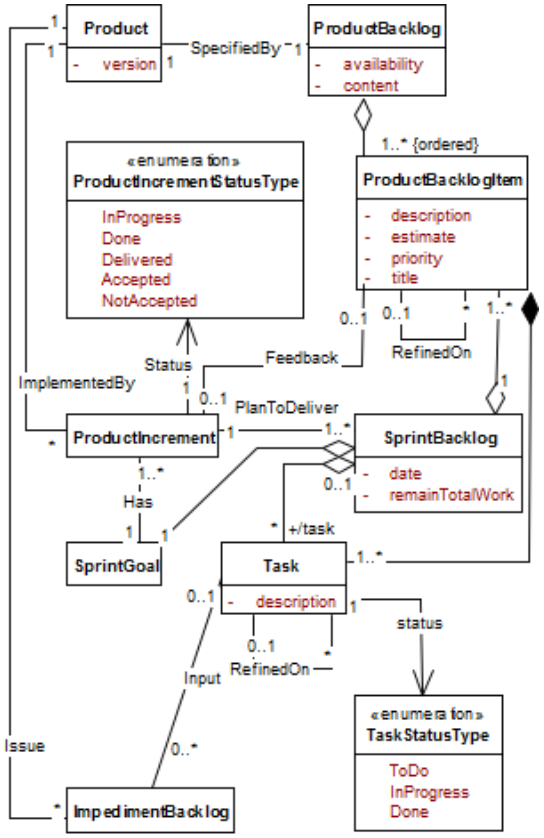


Figura 5 Artefactos de Scrum

3.1. Relaciones entre Roles, Eventos y Artefactos

Las relaciones entre los distintos constructores de Scrum (roles, artefactos y eventos) permiten a los Stakeholders (Figura 1) responder algunas preguntas de como un proceso Scrum fue llevado a cabo. Por ejemplo, ¿Cuál Stakeholder es responsable de un artefacto?

En Scrum, un rol es responsable de un artefacto; esto simplifica trazar a los distintos responsables de los artefactos. Sin embargo, otros roles pueden usar y actualizar, artefactos con el permiso adecuado. La Figura 6 explicita las relaciones existentes entre los roles y artefactos.

Otra pregunta que puede surgir es cuando se presenta un impedimento durante el desarrollo, en ese caso las trazas deben permitir responder ¿cuáles artefactos se verán involucrados en el impedimento?

Se debe considerar que los eventos son secuenciales, están asociados por una relación predecesor – sucessor (Figura 4). Esta característica indica que cuando un evento comienza otro termina. Un evento culmina con la entrega de un artefacto. Esta conclusión también representa un punto que puede y debe ser analizado y adaptado en el caso de ser necesario. Cada artefacto es una herramienta o un producto de trabajo intermedio que permite el desarrollo en un ambiente difícil. Este hecho está representado en la Figura 7 reificando la relación TraceTo mostrada en la Figura 1. Los conceptos de eventos, como Sprint, permiten trazar objetos de entrada, como ProductBacklogItem, en objetos de salidas, tales como ProductIncrement.

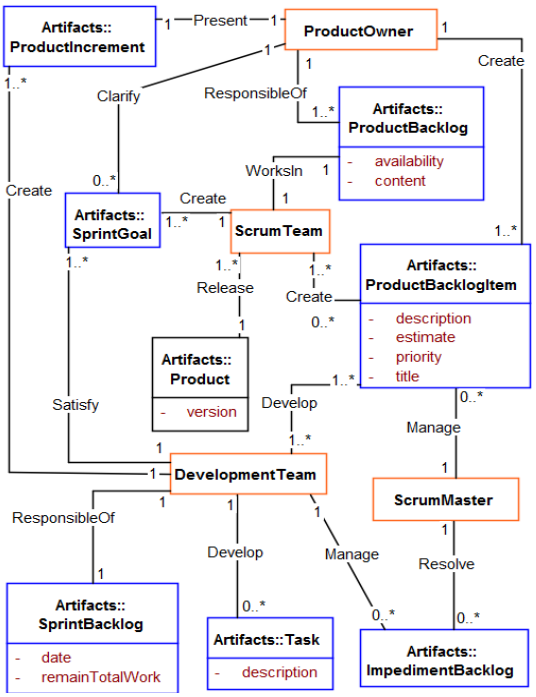


Figura 6 Roles y artefactos

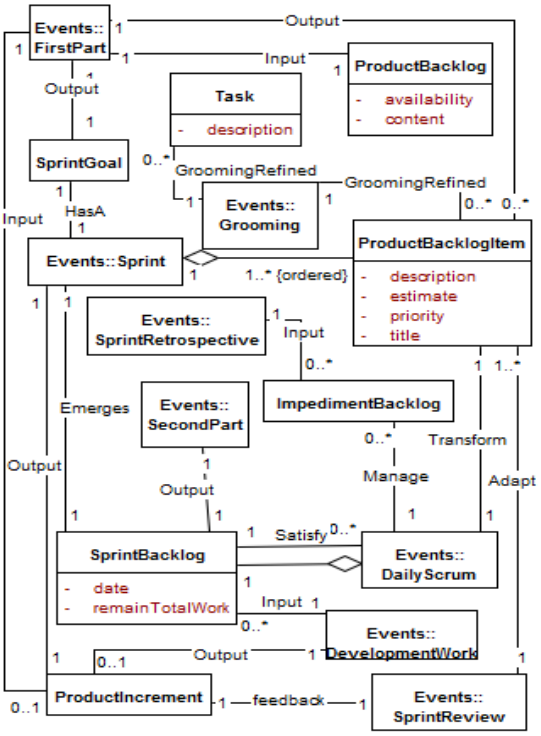


Figura 7 Eventos y artefactos

4. Moodle como un caso de estudio

Como caso de estudio se analizó al proceso de desarrollo de la plataforma de aprendizaje Moodle [11]. Moodle está diseñado con el objetivo de brindar un sistema integrado que permita crear ambientes de aprendizaje personalizados. Esta plataforma se encuentra desarrollada por un equipo perteneciente a la organización, por la comunidad internacional y además por una red de socios certificados. Está dirigido por colaboración abierta donde la comunidad sirve de apoyo al proyecto. Para llevar a cabo este proyecto se utiliza Scrum como proceso de desarrollo.

En Moodle los Stakeholders interpretan diferentes roles tales como: Moodle Developer, Component Lead, Integrator Team, Tester, Product Maintainer, Moodle Product Owner, Moodle Scrum Master, and Moodle Team (Figura 8). Estos roles especializan el concepto de roles de Scrum (Figura 3). A su vez se encontrarían involucrados en el proceso de trazabilidad (Figura 2) que abarca desde la determinación de necesidades, pasando por planificación e implementación, finalizando con la evaluación y retrospectiva del proceso. El MoodleTeam (un ScrumTeam, Figura 8) es el responsable de la creación y mantenimiento de los Issue, que una vez desarrollados serán parte de un ReleaseCandidate, pudiendo dar origen a nuevos Issues o Bugs. Esto refleja que prácticas de trazabilidad son llevadas a cabo de manera implícita y la recuperación, análisis, actualización y almacenamiento de estas trazas debe ser realizada de forma completamente automática. Esto se debe a que estas actividades generarían un trabajo adicional en los desarrolladores.

Los artefactos de Moodle creados durante su proceso de desarrollo son: Issues, List of Issue, Issues on Sprint, Release Candidate, Release, Roadmap, y Bug Issue (Figura 9). La información representada por estos artefactos muestra que distintos usos básicos de trazabilidad pueden implementarse en el modelo[9]. Un trazado hacia adelante o hacia atrás siguiendo, o no, un camino cronológico del proceso de Scrum (relaciones Predecessor-Successor Figura 4). Esto permitiría descubrir en que eventos fueron creadas, modificadas o eliminadas las trazas. Por ejemplo un requerimiento representado por un Issue (Figura 9), creado en el evento FirstPart, puede ser luego refinado en nuevos Issues en el evento de Grooming (Figura 7). Posteriormente puede ser pronosticado para ser desarrollado en un Sprint, finalmente implementado y aprobado en un ReleaseCandidate. Este modelo nos ayuda a encontrar las relaciones existentes entre Issues (Product Backlog Items), releases (Products versions), Releases Candidates, etc. Permitiéndonos especificar la granularidad de la información necesaria para las trazas. Esta información puede ser creada durante Issue Triage mientras tanto Issue y Task están siendo analizados.

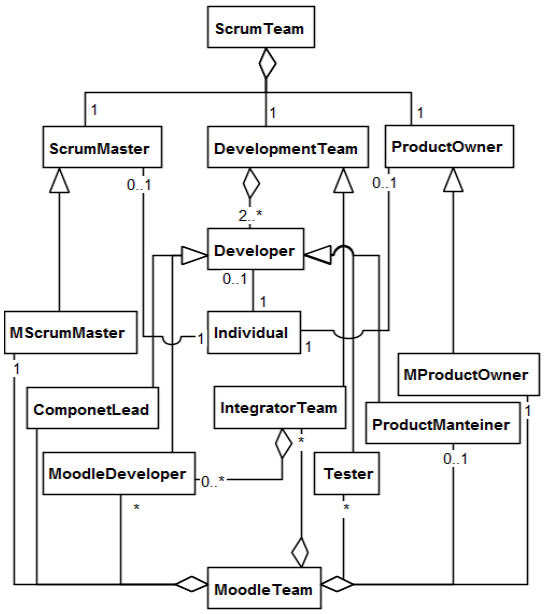


Figura 8 Roles de Moodle en Scrum

Como se mencionara anteriormente, la trazabilidad de artefactos puede ser representada por tripletas. Estas tripletas están constituidas por dos artefactos (artefacto origen y artefacto destino) y un enlace (TraceTo) que asocie estos elementos [9] (Figura 1). Esto determinará las trazas en este caso de estudio, un Issue puede interpretarse tanto como un artefacto destino, como origen de una traza. Por ejemplo: una traza podría ser un determinado Issue que fue planeado para ser desarrollado en un Sprint (IssueOnSprint). Luego otra traza podría ser un IssueOnSprint que es implementado en una versión de un Release. En un SprintReview pueden surgir nuevos requerimientos, dando lugar a nuevos Issues. Se podrían encadenar las trazas, permitiendo responder a una pregunta de competencia sobre qué requerimientos fueron desarrollados en un Release particular.

Las Figuras 10 y 11 fueron obtenidas desde The Moodle Tacker [16]. La Figura 10 “Event monitor: Create a subscription instance class”, es una instancia de Issue (Figura 9). La Figura 11 presenta otra instancia de Issue: “Create a Event Monitor tool”. Ambos Issues son Product Backlog Item debido a que Issue especializa a Product Backlog Item (Figura 9).

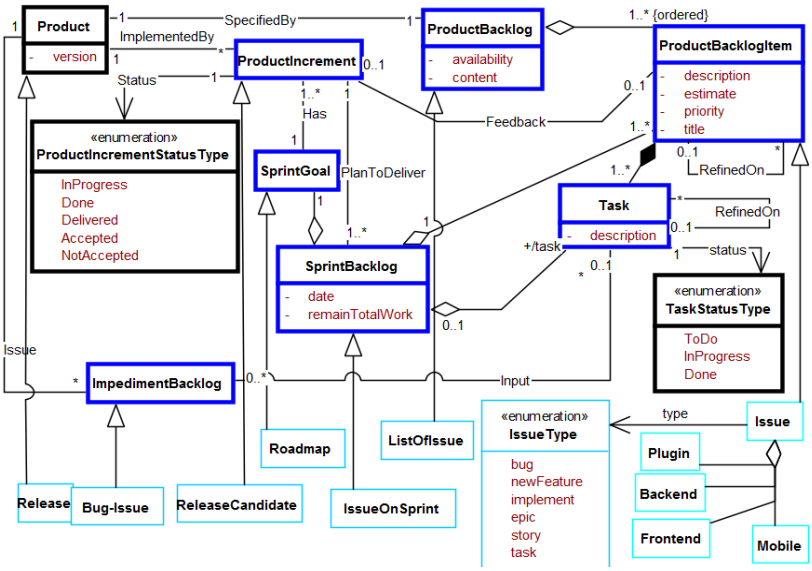


Figura 9 Artefactos de Moodle en Scrum

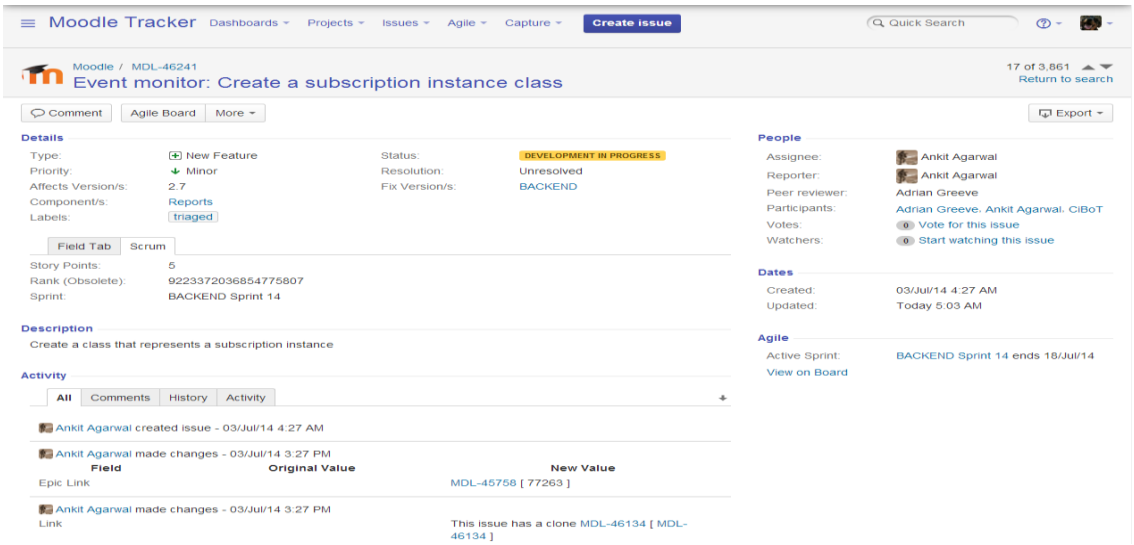


Figura10 Instancia de un Issue de Moodle

La información representada en la Figura 10 es:

- Descripción del Issue;
- Fecha de creación y actualización del Issue;
- Quien fue el Stakeholder creador del Issue;
- Qué MoodleDevelopers lo implementaron o modificaron al Issue;
- Sprint en el que fue creado o modificado;
- Release a la cual se afecta con la modificación del Issue.

La Figura 11 representa una relación entre dos o más Issues del Moodle Project (“Create a Event Monitor tool” y “Event monitor: Create a subscription instance class”). El Issue del tipo Epic (Figura11) está compuesto por otros Issue del tipo newFeature (Figura 10), del tipo task, entre otros. Para que “Create a Event Monitor tool” finalice todos los Issue que lo componen (“Event monitor: finalise the basic file and database structure”, “Event monitor: Create a subscription instance class”, “Event monitor: Create rule class”, etc) deben tener estatus Done. La Figura 11 también muestra los Issues que fueron desarrollados en el Sprint en curso, sus Status, y también que otros Issues se verían afectados por este. Para responder a las preguntas introducidas en la sección 1, es trascendental mencionar que cada Issue reúne un conjunto de requerimientos detallados en otro Issue (Product Backlog Item). La representación explicita de estos enlaces permite establecer qué requerimientos (Product Backlog Items) guiaron a los diferentes Release Candidates (Product Increments). Como se ha señalado en la sección 3, podemos instanciar este caso de estudio en las actividades de creación de trazabilidad (Figura 2).

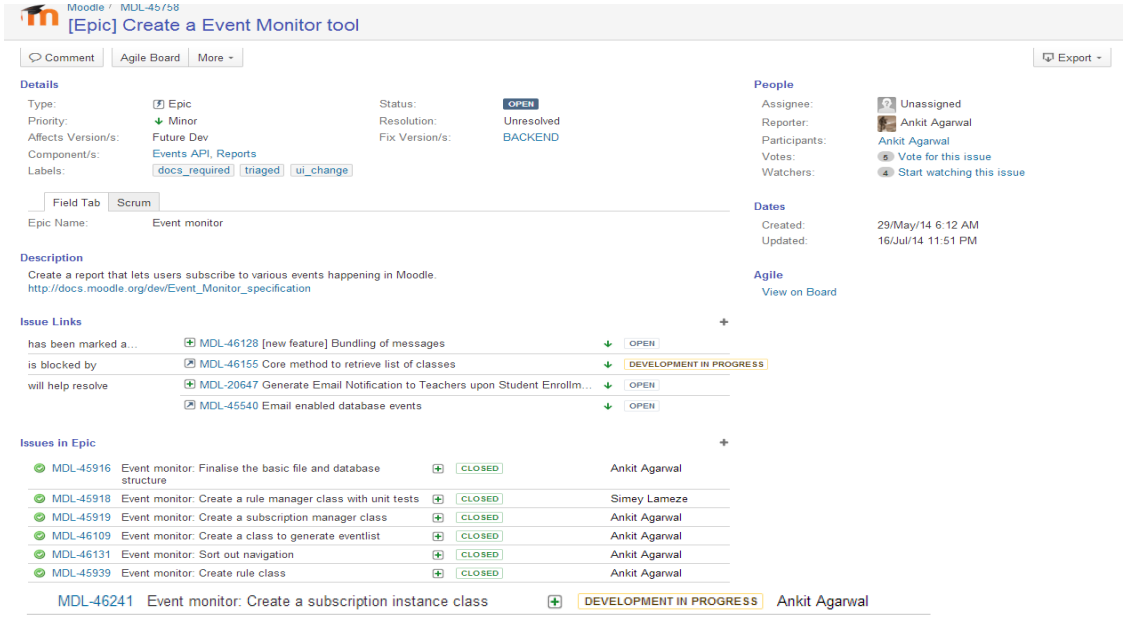


Figura 11 Instancia de un Issue de Moodle

Asimismo, se podría tomar como una tripleta potencial a: artefacto fuente: "Create a Event Monitor tool", artefacto destino: "Event monitor: Create a subscription instance class", enlace de trazabilidad: RefineOn. La representación de esta traza estaría dada por esta tripleta y su almacenamiento lógico como "Event monitor: Create a subscription instance class" es parte de "Create a Event Monitor tool", pudiendo aplicar una trazabilidad horizontal pudiendo ser usada de manera bidireccional (source-target, target-source). En el desarrollo de la plataforma, los enlaces de trazas son representados como enlaces de hipertexto (Figura 10 y 11).

Los enlaces son generados en forma automática por las herramientas de control de versiones y de seguimiento de cambios. Pero la recuperación y evaluación de estos es realizada de forma manual. Así se puede encontrar las trazas de todos los requerimientos que posean una relación explicita entre ellos. En Moodle Scrum esta búsqueda es realizada de forma semiautomática, pero si la trazabilidad es planificada de manera consciente, se podrían implementar herramientas que recuperen, representen y evalúen estos enlaces en forma automática.

Los stakeholders involucrados en el proceso de desarrollo de Moodle son los encargados de administrar los artefactos generados durante el proceso. Estos artefactos a su vez son posibles trazas y encuadran dentro de los artefactos tipo Object y relaciones TraceTo (Figura 1). Para brindar respuesta a las inquietudes realizadas en la sección 1, los modelos desarrollados reflejan que la creación de las trazas está a cargo del MoodleTeam. Los otros roles como el MoodleDeveloper (instancia de Scrum Developer en Figura 8) es el encargado de tomar un IssueOnSprint para luego desplegarlo en un ReleaseCandidate. Asimismo es el encargado del desarrollo de los Issues dentro del DevelopmentWork. El MProductOwner podría utilizar las trazas creadas para mejorar el Roadmap y la planificación del Sprint en curso. De esta manera, se podría maximizar la cantidad y calidad de desarrollo de los IssueOnSprint. El MScrumMaster utilizaría las trazas para corregir los Bug-Issue originados en alguno de los eventos del proceso, tales como MDailyScrum (instancia de DailyScrum), BugFixing (instancia de DevelopmentWork), IssueTriage (instancia de Grooming). El MoodleTeam es responsable de la planificación de la trazabilidad (Figura 2), esto no sería algo menor en el proyecto ni en el proceso. Porque es en este estado donde se definen las necesidades de las trazas, los recursos necesarios para llevarla a cabo, luego las trazas son implementadas (Creación, Uso y Mantenimiento) y finalmente se evalúa el proceso.

Los eventos de Moodle en donde los Issues son adquiridos son PlanningAndEstimation (especialización de SprintPlanning), IssueTriage (especialización de Grooming), PlanningDevelopment (especialización de DailyScrum), HQDailyScrum (especialización de DailyScrum), y ReleaseCandidateEvaluation (especialización de SprintReview). Las trazas luego son modificadas en IssueTriage, HQDailyScrum, BugFixing (especializaciones de

DevelopmentWork), QA y Testing (especializaciones de DevelopmentWork).

5. Conclusión

En este trabajo se presentó un modelo de trazabilidad de Scrum. Este es definido como una instancia del metamodelo de trazabilidad (Figura 1). Nuestra propuesta naturalmente integra la representación de artefactos del proceso Scrum (concepto Object) con los eventos (relación TraceTo) que ellos generaron, capturando también la participación de los actores (roles). El modelo fue extendido para representar un proceso de Scrum particular. Como primer resultado, el modelo propuesto muestra que algunas prácticas de Scrum pueden ser interpretadas como prácticas de trazabilidad. Sin embargo, es necesario un mecanismo formal que nos permita automatizar la recuperación de enlaces de trazabilidad en el desarrollo ágil. Nuestra propuesta es un primer paso en esta dirección, pero estas ideas primarias necesitan ser mejoradas y aún más trabajadas en el futuro.

Asimismo este modelo refleja una serie de actividades que podrían servir para establecer la trazabilidad de artefactos y hacerla usable. Como así también una descripción de los responsables de esas tareas y los recursos que serán necesarios para llevarlas a cabo. De igual forma que las entradas y salidas. Dentro del paradigma trazabilidad ágil las trazas que van desde el código o el diseño hasta los requerimientos, son realizadas mediante test unitarios y de aceptación. Estos test demuestran la trazabilidad de los requerimientos, debido a que se ejecuta código específico que implemente las necesidades. Existen herramientas simples como sistemas de control de versiones, wikis, herramientas de rastreo de cambios, entre otras que permiten el hacer uso de trazabilidad sin perder agilidad. El uso de estas herramientas no pretende sobreponerse a los principios ágiles porque buscan facilitar la tarea del desarrollo y de la satisfacción del cliente.

Se puede decir que en Scrum las trazas son creadas entre los diferentes artefactos pero esto no implica que exista trazabilidad, debido a que las otras partes del proceso de trazabilidad no son cumplidas (uso, planeación de la estrategia y mantenimiento). Además de la creación de los enlaces es importante definir la recuperación de los enlaces y la planificación de la trazabilidad para que su uso sea realmente útil, y no tan solo un sobre-trabajo. Si en Moodle se quisiera aplicar técnicas de trazabilidad, un aspecto fundamental a estudiar es la calidad de esos enlaces que se pretende trazar.

En proyectos como Moodle se vuelve difícil, sino imposible, el poder confiar en la memoria (individual o colectiva) para entender cuáles funciones específicas fueron implementadas en código, o recuperar partes del código que son sensibles a cambios debido a los intercambios interrelacionados. Para solucionar esta dificultad se puede realizar una combinación de diferentes técnicas como trazas básicas desde user stories a casos de pruebas y a código. Otra técnica es la trazabilidad justo a tiempo y la última podría ser user stories más refinadas a trazas de código con utilización de tags.

Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET, la Universidad Tecnológica Nacional, la Universidad Nacional de La Rioja y la Agencia Nacional de Promoción Científica y Tecnológica. Se agradece el apoyo brindado por estas instituciones.

Referencias

[1] L. Cao and B. Ramesh, “Agile Requirements Engineering Practices: An Empirical Study,” IEEE Softw., vol. 25, no. 1, pp. 60–67, Jan. 2008.

[2] A. Espinoza and J. Garbajosa, “A study to support agile methods more effectively through traceability,” Innov. Syst. Softw. Eng., vol. 7, no. 1, pp. 53–69, Mar. 2011.

[3] “IEEE Standard Glossary of Software Engineering Terminology,” IEEE Std 61012-1990, pp. 1–84, Dec. 1990.

[4] B. Ramesh and M. Jarke, “Toward reference models for requirements traceability,” IEEE Trans. Softw. Eng., vol. 27, no. 1, pp. 58–93, Jan. 2001.

[5] “Manifesto for Agile Software Development,” 2001.

[6] L. Williams, “What Agile Teams Think of Agile Principles,” Commun ACM, vol. 55, no. 4, pp. 71–76, Apr. 2012.

[7] K. Schwaber, “SCRUM Development Process,” in Business Object Design and Implementation, D. J. Sutherland, C. Casanave, J. Miller, D. P. Patel, and G. Hollowell, Eds. Springer London, 1997, pp. 117–134.

[8] J. Sutherland and K. Schwaber, “The Scrum Guide,” HttpswwwscrumorgScrum-Guide, 2013.

[9] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, and P. Mäder, “Traceability Fundamentals,” in Software and Systems Traceability, Springer London, 2012, pp. 3–22.

[10] P. Mäder and O. Gotel, "Ready-to-Use Traceability on Evolving Projects," in Software and Systems Traceability, J. Cleland-Huang, O. Gotel, and A. Zisman, Eds. Springer London, 2012, pp. 173–194.

[11] "The Moodle Project," 2014. [Online]. Available: www.moodle.org.

[12] P. Arkley, P. Mason, and S. Riddle, "Position paper: Enabling traceability," in Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering, Edinburgh, Scotland (September 2002), 2002, pp. 61–65.

[13] Y. Zhang, R. Witte, J. Rilling, and V. Haarslev, "Ontological approach for the semantic recovery of traceability links between software artefacts," IET Softw., vol. 2, no. 3, pp. 185–203, Jun. 2008.

[14] J. Cleland-Huang, R. Settimi, E. Romanova, B. Berenbach, and S. Clark, "Best Practices for Automated Traceability," Computer, vol. 40, no. 6, pp. 27–35, Jun. 2007.

[15] A. Mahmoud, "Toward an effective automated tracing process," in 2012 IEEE 20th International Conference on Program Comprehension (ICPC), 2012, pp. 269–272.

[16] "The Moodle Tracker," 2014. [Online]. Available: https://tracker.moodle.org/secure/Dashboard.jspa.

AId: Uma Ferramenta para Análise de Identificadores de Programas Java

Javier Azcurra
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
javierazcurra_m@yahoo.com.ar

Mario Berón
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
mberon@unsl.edu.ar

Germán Montejano
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
gmonte@unsl.edu.ar

Augusto Farnese
Departamento de Ciências da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
farnese@dcc.ufmg.br

Pedro Rangel Henriques
Departamento de Informática
Universidade do Minho
Braga, Portugal
pedrorangelhenriques@gmail.com

Maria J. Varanda Pereira
Departamento de Informática
Instituto Politécnico de Bragança
Bragança, Portugal
mjoao@ipb.pt

Resumo

As demandas atuais no desenvolvimento de software implicam uma evolução e manutenção constante do software com menor custo de tempo e recursos [15,16, 17, 18]. A Compreensão de Programas (CP), uma disciplina da Engenharia do Software, fornece os métodos, técnicas e estratégias para levar adiante esta tarefa. Em geral, as técnicas de compreensão fazem uso de duas classes muito importantes de informação: Estática e Dinâmica. Em ambas as classes, há um elemento que é informativo e sempre usado: os identificadores (Id). Estudos indicam que os Ids, mesmo quando abreviados ou compostos, encerram indícios das funcionalidades dos sistemas onde são usados [12, 7, 9, 8]. Por esta razão construir ferramentas de compreensão que automatizem o processo de extração e análise dos identificadores é uma contribuição muito importante para a CP.

Neste artigo apresenta-se a AId uma ferramenta que: i) automatiza a recuperação de identificadores encontrados em programas escritos em Java, e ii) aplica algoritmos de análise de identificares a fim de capturar o seu significado com vista a ajudar a compreender o programa.

1. Introdução

A Compreensão de Programas [20, 21, 31] é uma disciplina da Engenharia de Software cuja finalidade é facilitar o entendimento dos sistemas. Uma forma, se não a mais importante, de atingir este objetivo consiste em relacionar dois domínios muito importantes, como são: o Domínio do Problema e o Domino do Programa. Para poder fazer essa relação deve-se: i) construir uma representação do Domínio do Problema, ii) construir uma representação do Domínio do Programa, e iii) elaborar uma estratégia de vinculação de ambos domínios [19, 22, 23, 30].

Levando em consideração o Domínio do Problema, pode-se dizer que um possível caminho para elaborar uma representação consiste em: i) em extrair os conceitos usados no programa, e ii) estabelecer relações entre eles formando assim aquilo que se designa por mapa conceitual. Uma aproximação interessante para recuperar os conceitos consiste em capturar os identificadores do programa e realizar uma análise considerando o contexto onde aparecem. Os estudos realizados [1, 6, 7, 24] indicam que uma grande parte dos programas têm identificadores (Ids), os quais fornecem abundante informação. Geralmente, os Ids são compostos por mais de uma palavra em forma de abreviatura. Vários autores são de opinião [12, 7, 9, 8] que essas abreviaturas podem conter informação oculta que é própria do Domínio do Problema (conceitos). Uma forma possível de extrair essa informação dos identificadores consiste em expandir essas abreviaturas para as suas correspondentes palavras expressas em linguagem natural. Para atingir esta meta deve-se realizar as seguintes tarefas: i) extrair os identificadores do código, ii) aplicar técnicas de divisão onde o Id se decompõe nas distintas palavras abreviadas que o compõem, e iii) usar estratégias de expansão das abreviaturas para transformar as mesmas em palavras completas.

Normalmente, os nomes dos Ids são escolhidos mediante os critérios do programador [14, 7, 8]. Isto representa a principal dificuldade para as técnicas de expansão de abreviaturas de Ids (porque nem sempre os programadores usam as convenções adequadas e em muitos casos os nomes não refletem a semântica correta). Uma forma de enfrentar esta dificuldade consiste em usar as fontes de informação informal que se encontram disponíveis no código fonte. Por informação informal se entende aquela contida nos comentários dos módulos,

comentários das funções, literais, documentação do sistema e todos os demais recursos descritivos do programa que estejam escritos em linguagem natural. Sem duvida, os comentários têm como principal finalidade ajudar a compreender um segmento do código [13, 25]. Por esta razão, os comentários são uma fonte de informação natural para entender o significado dos Ids no código, como também para extrair conceitos do Domínio do Problema. Por outro lado para poder entender a semântica dos Ids, consideram-se também os literais (strings constantes). Eles representam um valor constante formado por sequencias de caracteres. Eles são geralmente utilizados nas instruções de escrita de mensagens para ficheiro ou écran ou para inicializar variáveis do tipo string. No caso destas fontes de informação informal serem escassas no código da aplicação, pode-se recorrer a alternativas externas como é o caso dos dicionários predefinidos de palavras em linguagem natural.

Na atualidade pode-se classificar as estratégias de análise de identificadores em dois grandes grupos. As que usam dicionários predefinidos e as que não. As primeiras são as mais arcaicas e as segundas as mais recentes. Infelizmente, nem todas as estratégias de análise de identificadores estão programadas em ferramentas automáticas.

O artigo está organizado como se descreve a seguir. Na seção 2, se explicam os conceitos subjacentes ao contexto da teoria de análise de identificadores. Na seção 3, se apresenta o AId uma ferramenta que reúne as características mencionadas na introdução deste artigo e que analisa detalhadamente identificadores e expande as abreviaturas. Na seção 4, se mostra o funcionamento do AId usando um caso de estudo. Finalmente, na seção 5 apresentam-se as conclusões e futuras extensões deste trabalho.

2. Análise de Identificadores

"Um identificador (Id) basicamente se define como uma sequência de letras, dígitos ou caracteres especiais de qualquer comprimento que serve para identificar as entidades do programa, isto é, para dar a cada uma um nome único que a individualize".

Cada linguagem tem as suas próprias regras que definem como podem ser construídos os nomes dos seus Ids. Por exemplo, em linguagens como C e Java não se pode declarar Ids que coincidam com as palavras reservadas ou que contenham alguns símbolos especiais ($,&,#) com exceção dos hifens ou underscore (-_).

No caso geral, um Id num programa está associado com um ou mais conceitos do programa.

Identificador ⇔ Conceito

Noutras palavras, um Id é um representante dum conceito que pertence ao Domínio do Problema [1, 2, 25]. Por exemplo, o Id *openWindow* está associado ao conceito "abrir uma janela". Daí a importância de analisar os Ids.

Para melhorar a CP se requere que os nomes dos Ids transmitam de maneira clara os conceitos que representam [1, 3, 4]. Contudo, na prática isto não é levado em conta. Durante o desenvolvimento dos sistemas, o critério seguido para atribuir nomes aos Ids está mais orientado para o estilo de escrita no formato do código e sua documentação do que no conceito que o Id representa. Por isso durante a etapa de manutenção de software o nome dos Ids pode não ser útil para compreender o sistema.

Antes de continuar descrevendo a importância dos nomes de identificadores serão classificadas as distintas formas em que se pode atribuir um nome a um Id. Estudos realizados com 100 programadores [4] sobre compreensão de Ids indicam que existem três tipos principais de nomes (tomando como exemplo o conceito *File System Input*): palavras completas (*fileSystemInput*), abreviaturas (*flSysIpt*), uma só letra (*f*). É importante ter em conta que também os nomes dos Ids podem estar compostos por mais de uma palavra como claramente mostra-se acima.

Os resultados do estudo mostram que as palavras completas são as melhor compreendidas, no entanto as estatísticas mostram em alguns casos que as abreviaturas, que se localizam em segundo lugar, não demostram uma diferença notória em relação às palavras completas [4].

Feild e o seu grupo de pesquisa fazem duas distinções conhecidas como *hardwords* e *softwords* [5, 6, 2]. As hardwords destacam a separação de cada palavra que compõe o identificador através duma marca específica; alguns exemplos são: *fileSystem*[1] ou *fileSYSTEM* (marca a separação com o uso duma maiúscula entre minúsculas) assim também como file-*system* ou file_system em que se utiliza um caráter especial, como o hífem ou o underscore. As *softwords* não tem nenhum tipo de separador ou marca que dê indícios das palavras que o compõem. Por exemplo: *textinput* ou *TEXTINPUT* estas palavras estão compostas por *text* e por *input*, mas a palavra composta não tem uma marca que destaque as partes. A nomenclatura das hardwords e softwords será utilizada no resto deste artigo.

Voltando a descrever a importância dos nomes de identificadores pode-se dizer que existem inúmeras convenções de nomes dos Ids, algumas delas são: no caso de Java, os nomes dos pacotes devem começar com minúscula (main.packed); o nome das classes deve ter uma maiúscula na primeira letra de cada palavra componente (MainClass). No caso de C#, as classes são nomeadas da mesma forma que em Java. Porém o nome

[1] Esta notação é conhecida no contexto da programação como camel-case.

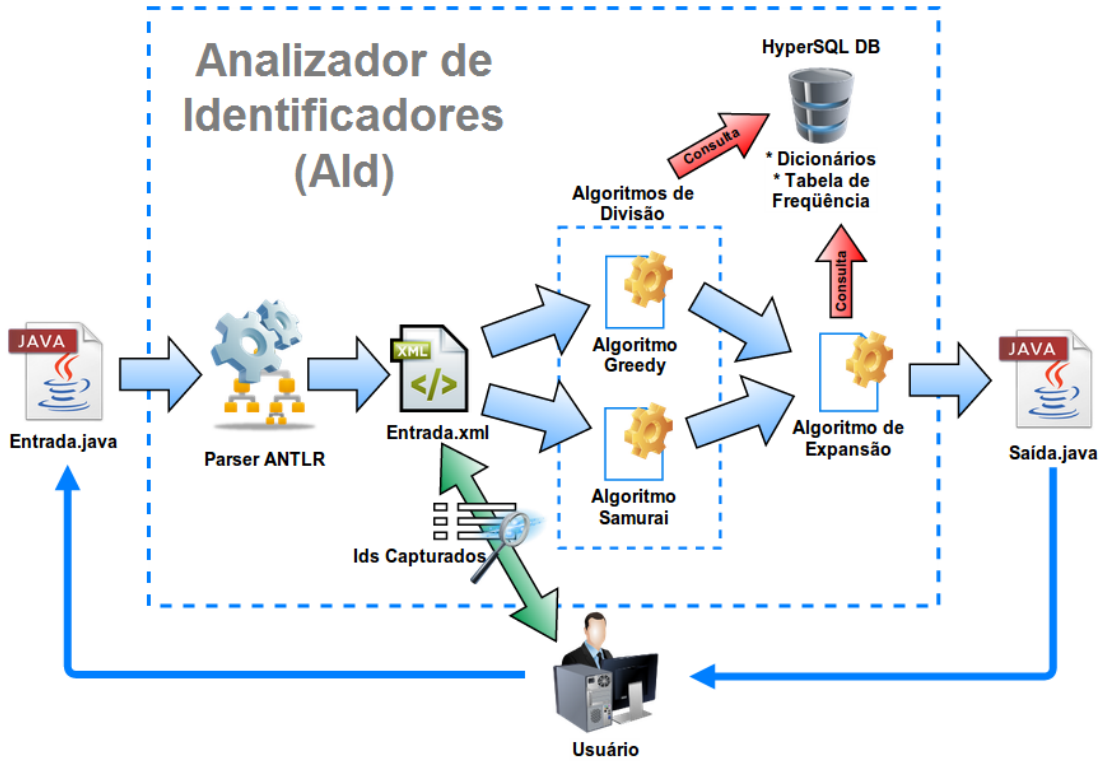


Figura 1 – Arquitetura do AId

dos pacotes em cada nível deve começar com maiúscula e o resto com minúscula (Main.Packed). Isto indica que o critério seguido se concentra mais em aspetos sintáticos do Id, ligados ao domínio da programação e não tanto nos aspetos semânticos ligados aos conceitos do domínio do problema. Uma evidência clara da importância do critério orientado para a semântica são as técnicas que se aplicam na proteção do código. Algumas delas se encarregam de substituir os nomes originais dos Ids por sequências de caracteres aleatórios e dessa forma reduzem a compreensão: técnicas de ofuscação de código.

Quando os programadores desenvolvem as suas aplicações não prestam importância à correta atribuição de nomes com semântica aos Ids. Existem três razões principais para isso: i) Os Ids são escolhidos pelos programadores sem ter em conta os conceitos associados; ii) Os desenvolvedores têm pouco conhecimento dos nomes usados em Ids localizados em outros setores do código fonte, e iii) Durante a evolução do sistema, os nomes dos Ids se mantêm e não se adaptam às novas funcionalidades (ou conceitos) a que possam vir a estar associados. Neste sentido, a má atribuição de nomes aos Ids se combate com as normas da chamada programação "menos egoísta". Esta consiste em fazer programas mais claros e compreensíveis para o futuro leitor que não esteja familiarizado. Para atingir este objetivo se devem respeitar duas regras de atribuição de nomes [1, 3]: i) Atribuição concisa: o nome do Id é conciso se a semântica do nome coincide exatamente com a semântica do conceito que o Id representa, e ii) Atribuição Consistente: cada Id deve ter associado um e só um único conceito.

O uso de *sinónimos* e *homónimos* prejudica a atribuição consistente de nomes aos Ids. Os homónimos são palavras que podem ter mais de um significado. Por este motivo, se o nome dum Id está associado a mais dum conceito, não estará claro que conceito representa.

Por outra parte, os sinónimos indicam que para um mesmo conceito podem ter associados diferentes nomes. Por exemplo, os nomes *accountBankNumber* e *accountBankNum* são sinónimos porque fazem referência ao mesmo conceito "número de conta bancária". Está demonstrado que a utilização de nomes consistentes, tal como foi mencionado atrás, é uma característica desejada. Isto é porque eles fazem que seja difícil identificar com clareza os conceitos no Dominio do Problema e, portanto os esforços para compreender o programa são maiores. Se os Ids tem nomes consistentes e concisos (identificando bem o conceito) os conceitos do Dominio do Problema podem ser descobertos mais facilmente [1, 3]. Intuitivamente, se necessita que os Ids representem bem um conceito, já que maior será o impacto que terá na interpretação do sistema [1, 3]. Contudo, durante as etapas de desenvolvimento e manutenção de software é difícil manter uma consistência global de nome nos Ids, sobretudo se o sistema for grande. Cada vez que um conceito se

modifica, o nome do Id associado deve mudar e adaptar-se à modificação.

Deissenboeck e o seu grupo de pesquisa em [1] propõem utilizar uma ferramenta que resolva os problemas da má atribuição de nomes descritos nos parágrafos precedentes. Dada a dificuldade de construir uma ferramenta totalmente automática para nomear corretamente os Ids, eles fizeram uma ferramenta semiautomática que necessita da intervenção do programador. Esta ferramenta, à medida que o sistema se vai desenvolvendo, constrói e mantém um dicionário de dados contendo informação sobre os Ids. No âmbito da Engenharia do Software o conceito de dicionários de dados é importante. Isto é, porque com os dicionários é possível descrever com clareza todos os termos utilizados nos grandes sistemas de software.

2.1. Tradução de Identificadores

As pessoas perante o código dos programas têm sérias dificuldades em perceber o propósito dos Ids, mas na verdade precisam de tempo a analisar o significado de sua presença. As estratégias automáticas dedicadas a facilitar esta análise são bem-vindas no contexto da CP.

A literatura dedicada a estratégias de análise de Ids indica que os Ids ocultam informação relevante do Domínio do Problema através de abreviaturas [7, 8]. Uma forma de exibir a informação oculta nos Ids é tentar converter estas abreviaturas em palavras completas em linguagem natural. Portanto, o foco da análise de Ids baseia-se na tradução das palavras abreviadas para palavras completas. O processo automático que se leva a cabo para realizar a tradução de Ids consta de dois passos [7]:

1. Divisão: separar o Id nas palavras que o compõem usando algum separador especial (Ejemplo: *flSys* ⇒ *fl-sys*).
2. Expansão: Expandir as abreviaturas que resultaram como produto do passo anterior.

3. AId: uma ferramenta para Análise de Identificadores

A figura 1 mostra a arquitetura do AId uma ferramenta que automatiza o processo de análise de identificadores. Esta ferramenta foi implementada usando a linguagem Java e o ambiente de desenvolvimento Netbeans. Na mesma figura pode-se constatar que AId permite realizar a análise em três etapas, a primeira consiste na extração de dados, a segunda tem por objetivo realizar a divisão dos Ids, e a terceira faz a expansão de Ids.

A fase de extração de dados (primeiro passo) recebe como entrada um ficheiro Java (*Entrada.java* na figura 1) que é processado por um analisador sintático (parser). Este parser foi programado utilizando a ferramenta ANTLR[2]. O analisador sintático extrai e armazena em estruturas internas informação estática sobre o código lido do ficheiro Java. Esta informação é composta por Ids como elemento principal, literais e comentários. Finalmente, toda esta informação recolhida se armazena num ficheiro XML (*Entrada.xml* na figura 1). Concluída a extração de informação se dá inicio à fase de divisão de Ids. Nessa etapa podem ser escolhidos dois algoritmos: *Greedy* e *Samurai*. Ambos os algoritmos recebem como entrada a informação da etapa anterior e dividem os Ids do código de entrada. As divisões são armazenadas em estruturas internas que serão consultadas na fase seguinte. Como estes algoritmos de divisão necessitam de dados externos, como dicionários (no caso de Greedy) ou tabelas de frequência de palavras (no caso de Samurai), estes dados se encontram armazenados na base de dados *HyperSQL DB* (parte superior da figura 1). A terceira e última fase implementa o algoritmo de expansão convencional que será explicado mais adiante. Este algoritmo toma os Ids divididos na etapa anterior e procede à sua expansão. Aqui também se necessita dos dicionários, por isso se fazem consultas à *HyperSQL DB*. Uma vez expandidos todos os Ids, estes são substituídos pelos Ids originais no ficheiro de entrada, gerando desta maneira o ficheiro de saída (*Saída.java* na figura 1).

3.1. Analisador Sintático

Foram pesquisadas ferramentas destinadas a construir analisadores léxicos e sintáticos. Neste sentido se deu preferencia àquelas ferramentas que se baseiam na teoria associada às gramáticas de atributos [28]. Da pesquisa previamente descrita, se determinou que a ferramenta ANTLR era a que melhor cumpria com o requisito antes mencionado. Esta ferramenta permite adicionar regras para o cálculo dos atributos, escritas em Java, à gramática independente de contexto inicial, neste caso também em linguagem Java. Estas regras de cálculo, ou ações semânticas, devem ser corretamente associadas à gramática para, por exemplo, implementar estruturas de dados e algoritmos que recolham os identificadores utilizados num programa [29]. Uma vez inseridas estas ações, a ferramenta se encarrega de ler a gramática e gerar o analisador adicionando as ações que foram programadas. Desta forma, se obtém um analisador sintático que não só examina o código Java para detectar erros ao nível sintático mas que também recolhe e armazena Ids. Além destas ações, se adicionaram outras ações semânticas que extraem comentários e strings literais. Estes elementos são necessários já que servem como entrada para os algoritmos de análise de Ids que serão explicados nas próximas secções.

[2] ANother Tool for Language Recognition - http://www.antlr.org

Algoritmo 1: Divisão Greedy
Entrada: idHarword *// identificador a dividir*
Saída: softwordDiv *// Id separado com espacios*
Variaveis Globais:
ispellList *// Palaras de ispell + Dicionário*
abrevList *// Abrevituras conhecidas*
stopList *// Palavras Excludentes*
1 softwordDiv← “”
2 softwordDiv← dividirCaracsEspDig(idHarword)
3 softwordDiv ← dividirCamelCase(softwordDiv)
4 ***Para cada substring s en softwordDiv fazer***
5 ***Se*** (s ∉ (stopList ∪ abrevList ∪ ispellList)) ***então***
6 sPrefixo ← procurarPrefixo(s,“”)
7 sSufixo ← procurarSufixo(s,“”)
// Se escolhe a divisão que fiz mais partições
8 s ← maxDivisão(sPrefixo,sSufixo)
9 ***return*** softwordDiv *// o Id dividido por espaços.*

Algoritmo 2: procurarPrefixo
Entrada: s // Abreviatura a dividir
Saída: abrevSeparada //Abrev dividido por espaços
// Punto de parada da recursión
1 ***Se*** (length(s) = 0) ***então***
2 ***return*** abrevSeparada
3 ***Se*** (s ∈ (stopList ∪ abrevList ∪ ispellList)) ***então***
4 ***return*** (s + ‘ ’+ procurarPrefixo(s,“”))
// Se extrae e se almacena o último caracter de s.
5 abrevSeparada ← s[length(s) - 1] + abrevSeparada
//Chamada recursiva sem o último caracter.
6 s ← s[0 , length(s) - 1]
7 ***return*** procurarPrefixo(s,abrevSeparada)

Algoritmo 3: procurarSufijo
Entrada: s // Abreviatura a dividir
Saída: abrevSeparada //Abrev dividido por espaços
// Punto de parada da recursión
1 ***Se*** (length(s) = 0) ***então***
2 ***return*** abrevSeparada
3 ***Se*** (s ∈ (stopList ∪ abrevList ∪ ispellList)) ***então***
4 ***return*** (procurarSufixo(s,“”) + ‘ ’+ s)
// Se extrai e se guarda o primer carater de s.
5 abrevSeparada ← abrevSeparada + s[0]
//Chamada recursiva sem o primeiro caracter.
6 s ← s[1 , length(s)]
7 ***return*** procurarSufixo(s,abrevSeparada)

3.2. Módulo de Divisão

O módulo de divisão implementa dois algoritmos, eles são: *Greedy* e o *Samurai*, os quais serão descritos a seguir.

3.2.1. Algoritmo Greedy

Como já foi várias vezes dito, o algoritmo Greedy, inicialmente proposto por o Lawrie, Feild, Binkley [10,5,6,7,8] utiliza três listas, a saber:

1. Palavras de dicionários: contêm palavras extraídas de dicionários públicos e do dicionário que utiliza o comando de Linux ispell.
2. *Abreviaturas conhecidas*: a lista se constrói com abreviaturas extraídas de distintos programas de autores especialistas. Incluem-se abreviaturas comuns (exemplo: *alt* → *altitude*) e abreviaturas de programação (exemplo: *txt* → *text*).
3. *Palavras a ignorar (stop words)*: possui palavras que são irrelevantes para realizar a divisão dos Ids. Inclui palavras chave da linguagem de programação (exemplo: *while*), Ids predefinidos (exemplo: *NULL*), nomes e funções de bibliotecas (exemplo: *strcpy*, *errono*), e todos os Ids que tenham um só caracter. Esta lista é muito grande.

O algoritmo Greedy utiliza as três listas mencionadas acima em forma de variável global. Isto ocorre porque as três listas são usadas por sub-rotinas.
O algoritmo procede da seguinte forma (ver algoritmo 1[3]), o Id que recebe como entrada se divide com espaços em branco, nas hardwords que o compõem (exemplo: *fileinput-txt* → *fileinput* y *txt* na linha 2 se é separada com caracteres especiais, ou, *fileinputTxt* → *fileinput txt* na linha 3) se é camelcase. Cada palavra resultante é pesquisada e caso esteja numa das três listas, é armazenada como uma só softword (exemplo: *txt* pertence à lista de abreviaturas conhecidas-linha 5). Se alguma palavra não está em nenhuma lista se considera como múltiplas softwords que necessitam subdividir-se (exemplo: *fileinput* → *file* e *input* - linha 5). Para subdividir estas palavras se procuram os prefixos e os sufixos mais longos possíveis dentro delas. Esta tarefa também se faz utilizando as três as listas antes mencionadas (linhas 6 e 7). Por um lado se procuram prefixos com um processo recursivo (ver algoritmo 2). Este processo começa analisando toda a palavra por completo. Vão sendo extraídos carateres do final até encontrar o prefixo mais longo que seja uma palavra válida, ou até que não haja mais caracteres (linhas 5 até 7 da função). Quando uma palavra se encontra numa lista (linha 3 da função) se coloca um separador (‘ ’). O resto que foi descartado se processa de novo através da função *procurarPrefixo* para procurar mais subdivisões (linha 4). De forma simétrica, outro processo recursivo se usa para tratar dos sufixos (algoritmo 3). Também extrai caracteres, mas neste caso desde a primeira posição até encontrar o sufixo mais longo presente numa lista ou até que não haja mais caracteres (linhas 5 até 7 da função).

[3] *softwordDiv* se modifica a traves de s na linha 8.

Algoritmo 4: Divisão Samurai
Entrada: token *// token a dividir*
Saída: tokenSep *// token dividido por espaços*
1 tokenSep ← divisãoHardWord (token)
2 ***return*** tokenSep

Algoritmo 5: divisãoHardWord
Entrada: token *// token a dividir*
Saída: tokenSep *// token dividido por espaços*
1 token← dividirCaracsEspDig(token)
2 token← dividirMinusSeguidoMayus(token)
3 tokenSep ← “”
4 ***Para cada substring s en token fazer***
5 ***Se*** (∃{i|esMayus(s[i]) ∧ esMinus(s[i + 1])}) ***então***
6 n ← length(s) – 1
// se determina o tipo de divisão
7 scoreCamel ← score(s[i,n])
8 scoreAlter ← score(s[i+1,n])
9 ***Se***(scoreCamel> $\sqrt{\text{scoreAlter}}$) ***então***
10 ***Se*** (i > 0) ***então***
11 s ← s[0,i − 1] + ‘ ’ + s [i,n]
12 ***em outro caso***
13 s← s [0,i] + ‘ ’ + s[i + 1,n]
14 tokenSep ← tokenSep + ‘ ’ +s
15 token ← tokenSep
16 tokenSep ← ‘ ’
17 ***Para cada substring s en softwordDiv fazer***
18 tokenSep← tokenSep + ‘ ’
+ divisãoSoftWord (s,score(s))
19 ***return*** tokenSep

Da mesma forma que a função de prefixos, quando encontra uma palavra, se insere um separador (‘ ’) e o resto se processa através da função *procurarSufixo* (linha 7). Quando ambos os processos terminarem, os resultados são retornados ao algoritmo principal. Mediante a função de comparação se escolhe o que tem maiores partições (linha 8 do algoritmo 1). Finalmente, o algoritmo Greedy retorna o Id destacando as palavras que o compõem usando como separador o carater espaço em branco (por exemplo *file input txt)* (linha 9 do algoritmo 1). A vantagem de fazer duas procuras (prefixo e sufixo) é aumentar as possibilidades de dividir o Id correctamente. Por exemplo, supondo que a palavra abreviada *fl* não se encontra em nenhum das três listas, mas sim as palavras *input* e *txt*. Dada esta situação, se o Id *flinputtxt* for processado por ambas as rotinas, o resultado de invocar *procurarPrefixo* é que não conseguirá dividir o Id. Isto sucede porque ao retirar caracteres do fim nunca se encontrará um prefixo conhecido. Não havendo divisão entre *fl* e *input* o resto da cadeia não se processará e tão pouco se dividirá em input e txt. Contudo, problema não surge quando se usa *procurarSufixo* porque ao retirar os caracteres do princípio da palavra *input txt* ficarão separados. Como o input é uma palavra conhecida se agregará um espaço em branco entre *fl* e *input*. Desta maneira o Id fica corretamente separado em três partes: *fl input txt*.

3.2.2. Algoritmo Samurai

Esta técnica, proposta por Eric Enslen, Emily Hill, Lori Pollock e VijayShanker [8], divide os Ids em sequências de palavras de forma semelhante ao algoritmo o Greedy mas a separação é mais efetiva. A estratégia utiliza informação presente no código para levar a cabo o objetivo. Isto permite que não seja necessário utilizar dicionários predefinidos. Além disso, as palavras que se obtiverem produto da divisão não estão limitadas pelo conteúdo destes dicionários. Desta forma, a técnica vai evoluindo com o tempo na medida em que novas tecnologias e as novas palavras se incorporem ao vocabulário dos programadores. O algoritmo escolhe a partição mais adequada nos Ids multi-palavra com base em uma função de pontuação (scoring). Esta função utiliza informação que se recolhe extraindo as frequências de aparição das palavras dentro do código fonte. Estas palavras podem estar contidas nos comentários, nas strings literais ou na documentação. Isto ocorre porque o objetivo de Samurai é mais amplo e não consiste somente em dividir Ids. Por esta razão, o nome do parâmetro de entrada do algoritmo é token e não simplesmente Id. O algoritmo primeiro se encarrega de extrair informação respeitante à frequência dos tokens no código fonte. Em seguida, constrói duas tabelas de frequência de tokens. Para a construção duma das tabelas primeiro executa o algoritmo que extrai do código fonte todos os tokens do tipo hardword. Estes tokens são agregados na tabela de frequências específicas. Uma entrada desta tabela corresponde à lista de tokens extraídos do programa em análise (cada token é único na tabela). A outra entrada corresponde ao número de ocorrência de cada token.

Por outro lado, existe a tabela de frequências globais. Esta tabela contém as mesmas duas colunas que a tabela anterior, tokens e suas frequências. A diferença principal é que a informação é recolhida de distintos programas de grande envergadura. Durante o processo de divisão do token, Samurai executa a função de scoring que se baseia na informação das tabelas mencionadas anteriormente. O algoritmo Samurai recebe como entrada o token a dividir e retorna como seu resultado o token separado com espaços. A execução se inicia invocando a rotina *divisãoHardWord* (ver algoritmo 4, linhas 1 e 2). Esta rotina se encarrega de dividir as hardwords (palavras que possuem underscore o são do tipo camel-case), e então cada uma das palavras obtidas são passadas à rotina *divisãoSoftWord* que será explicada mais adiante.

Algoritmo 6: divisãoSoftWord
Entrada: s *// softword string*
$score_{sd}$ *// pontuação de s sem dividir*
Saída: tokenSep *// token dividido por espaços*

```
1 tokenSep ← s, n ← length(s) - 1
2 i ← 0, maxScore ← − 1
3 enquanto (i < n) fazer
4   score_izq ← score(s[0,i])
5   score_der ← score(s[i+1,n])
6   preSuf ← esPrefijo(s[0,i]) ∨ esSufijo(s[i+1,n])
7   split_izq ← √score_izq > max(score(s),score_sd)
8   split_der ← √score_der > max(score(s),score_sd)
9   Se (!presuf ∧ split_izq ∧ split_der) então
10    Se ((split_izq + split_der) > maxScore) então
11      maxScore ← (split_izq + split_der)
12      tokenSep ← s[0,i] + ' ' + s [i+1,n]
13  em outro caso, Se(!presuf ∧ split_izq) então
14    temp ← divisãoSoftWord (s [i+1,n],score_sd)
15    Se (temp se dividido?) então
16      tokenSep ← s[0,i] + ' ' + temp
17  i ← i+1
18 return tokenSep
```

Na rotina *divisãoHardWord* (ver algoritmo 5) primeiro se executam duas funções (linhas 1 e 2). A primeira *dividirCaracteresEspeciasDigitos*, que substitui todos os possíveis caracteres especiais e números que o token possua por espaços em branco.

A segunda *dividirMinusSeguidoMayus*, da mesma forma que a anterior, agrega um espaço em branco entre dois caracteres que sejam uma minúscula seguido por uma maiúscula. Neste ponto só ficam tokens da forma softword ou que contenham uma maiúscula seguida de uma minúscula (Exemplos: *List*, *ASTVisitor*, *GPSstate*, *state*, *finalstate*, *MAX*). Os casos de softword que se obtiveram (*finalstate*, *MAX*) são passados diretamente à rotina *divisãoSoftWord*. As palavras obtidas que sigam o padrão maiúscula seguida de minúscula (*List*, *ASTVisitor*, *GPSstate*) vão ser submetidas a novo processo de divisão. Este caso é também do tipo camel-case onde a maiúscula indica começo da nova palavra. Contudo, o autor através de estudos que efetuou, encontrou variantes onde minúscula que surge a seguir a uma sequência de maiúsculas indica o fim da palavra (exemplo: SQLlist). O algoritmo decide entre ambas as variantes, calculando a pontuação (score) da parte direita de ambas as divisões (linhas 7 e 8). Aquela com pontuação mais alta entre as duas, será a escolhida (linha 9).

Para esclarecer melhor ambas as opções:

a) realizar a separação entre a mudança de maiúscula para minúscula (exemplo: *GPS state* - linha 11).
b) realizar a separação antes da primeira maiúscula segundo com a filosofia camel-case (exemplo: *GP Sstate* - linha 13)

Neste caso, a opção correta dever ser a a), já que state deve ter uma pontuação maior do que *Sstate*. No final as partes resultantes da divisão são enviadas a *divisãoSoftWord* (linha 18).

A rotina recursiva *divisãoSoftWord* (ver algoritmo 6) recebe como entrada um substring s que pode ter três tipos de variantes:

a) Todos os carateres em minúsculas.
b) Todos os carateres com maiúsculas.
c) O primeiro caráter com maiúscula seguido apenas por minúsculas (Visitor).

O outro parâmetro de entrada é a pontuação original $score_{sd}$ que corresponde a *s*. A rotina primeiro examina cada ponto possível de divisão em *s* dividindo em $score_{izq}$ $score_{der}$ respectivamente (linhas 4 e 5). A decisão de qual é a melhor divisão baseia-se em: a) substrings que não tenham prefixos ou sufixos conhecidos (linha 6), b) a pontuação da divisão escolhida sobressaia do resto das pontuações (linhas 7 até 9).

Para esclarecer este ponto, para cada partição (esquerda ou direita) obtida calcula-se o score (linhas 4 e 5). Depois este é comparado como a pontuação da palavra original ($score_{sd}$) e a pontuação da palavra atual (score(s)). No principio ambas são iguais, mas à medida que avança a

Algoritmo 7: expançãoBasica
Entrada: abrev *// abreviaturas ao expandir*
wordList *//Palavras extraídas do código*
phraseList *//Frases extraídas do código*
stopList *//Palavras Excludentes*
dic *//Dicionário em Inglês*
Saída: expansão *// abreviatura expandida*

```
1 Se (abrev ∈ stopList) então
2   return null
3 listaExpanção ← [ ]
  // Procuarar acrónimo
4 Para frase fra em phraseList fazer
5   Se (abrev é um acrónimo de fra) então
6     return fra
  // Procurar abreviatura comum
7 Para palavra w em wordList fazer
8   Se (abrev é uma abreviatura de w) então
9     return w
  // Procurar dicionário
10  listaCandidatos ←buscarDicionário(abrev,dic)
    listaExpanção.add(listaCandidatos)
11 unicaExp ← null
  //Se retorna o único resultado
12 Se (length(listaExpanção)=1) então
13   unicaExp ← listaExpanção[0]
14 return unicaExp
```

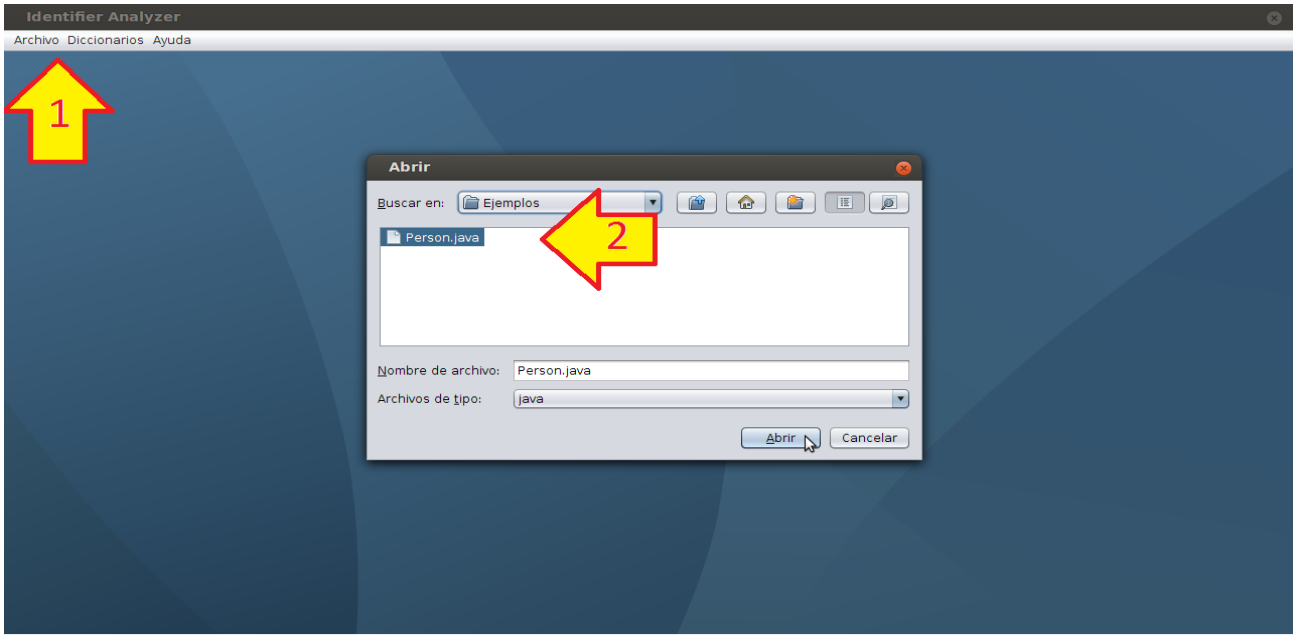


Figura 2 – Captura de uma écran do AId

recursão score(s) varia relativamente ao $score_{sd}$ (linhas 7 e 8). No caso de não ter prefixos e sufixos ordinários, se considera que a parte esquerda é um candidato. Por outro lado, a rotina é invocada recursivamente sobre a parte direita da cadeia para proceder a mais subdivisões (linha 14).

Se a parte direita finalmente se dividir, logo entra a parte esquerda e a direita também. Contudo, no caso que a parte direita não se dividir tão pouco deveria separar entre ambas as partes (o ***Se*** da linha 13 controla esta tarefa). As análises de dados [8] indicam que não se deve separar a atual posição, tendo em conta só a evidência na esquerda já que pode ser um erro; um caso de erro de divisão encontrado pelo próprio autor é string-ifed [8]. Este Id aparece muitas vezes no código fonte dos programas estudados pelo que o seu score é mais alto do que o resto. Por esta razão a raiz quadrada se utiliza no cálculo do resultado de score antes de comparar (linha 7 e 8), senão a divisão frequentemente seria errônea. Um exemplo é a palavra *performed*. Para que a técnica Samurai possa levar a cabo a tarefa de separação de instruções de controlo como é o caso do *se* (*if*) se necessita da função de scoring. Como foi explicado anteriormente esta função participa em duas decisões chave durante o processo de divisão: Na rotina *divisãoHardWord* para determinar se a divisão do Id é um caso camel-case ou não (linhas 7 e 8), na rotina *divisãoSoftWord* para pontuar as diferentes participações de substrings e escolher a melhor separação (linhas 4, 5, 7 e 8). Dada uma string *s*, a função score(s) indica i) a frequência de aparição de *s* no programa em análise e ii) a frequência num conjunto grande de programas predefinidos.

A fórmula é a seguinte:

$$Frec(s, p) + (globalFrec(s) / log10 (totalFrec(p)))$$

Onde *p* é o programa em estudo, *Frec(s,p)* é a frequência de ocorrência de *s* em *p*. A função *totalFrec(p)* é a frequência total de todas as strings no programa *p*. A função *globalFrec(s)* é a frequência de ocorrência de *s* num conjunto grande de programas tomados como amostra [8].

3.3. Módulo de Expansão

O algoritmo de expansão de abreviaturas, idealizado por Lawrie, Feild, Binkley (os mesmos autores da técnica de separação Greedy) [7], trabalha com quatro listas para realizar a sua tarefa:

i) Uma lista de palavras (em linguagem natural) que se extraem do código fonte de muitos programas,
ii) Uma lista de frases (em linguagem natural) presentes também no código fonte referido,
iii) Uma lista de palavras a ignorar (stop words), e
iv) Uma lista de palavras dum dicionário em inglês.

A lista de palavras se constrói utilizando uma ferramenta que extrai as palavras dos comentários que se encontram dentro e fora dos métodos, incluindo aqueles que são relativos à classe de estudo. A lista de frases se constrói utilizando uma ferramenta que as extrai [11], os principais recursos são os comentários e os Ids multipalavras. Neste ponto se constrói um acrônimo com

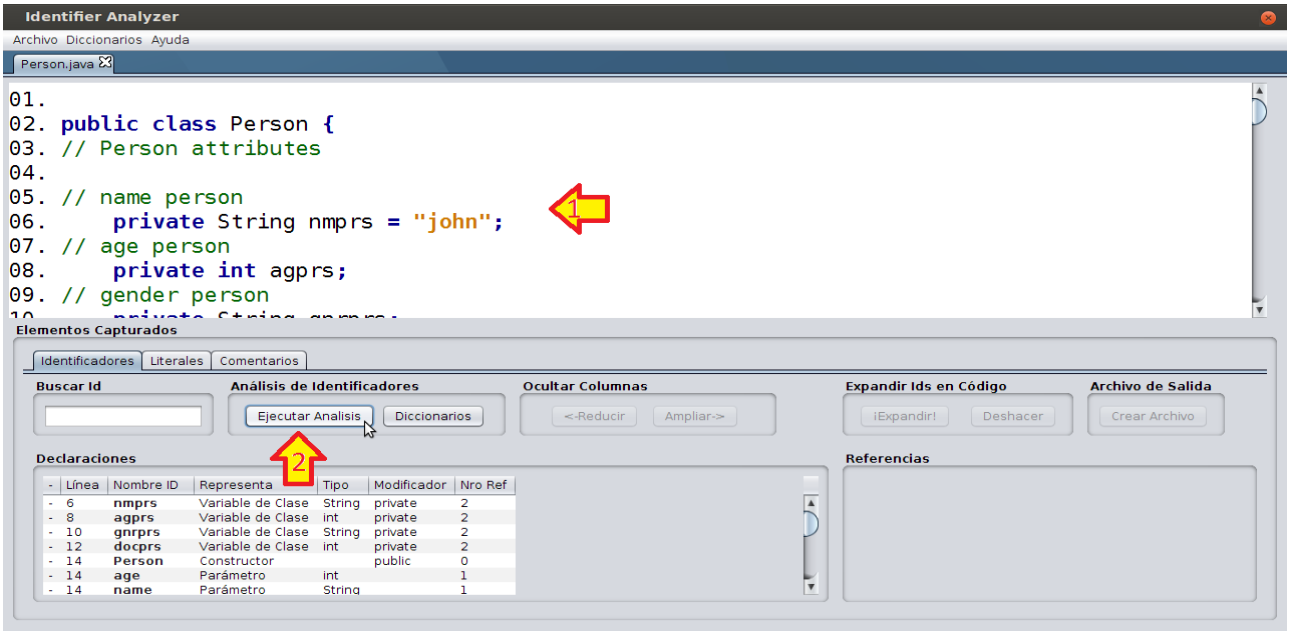


Figura 3 – Captura de um écran do AId

as palavras de alguma frase, se esse acrônimo coincide com algum dos Ids extraídos então essa frase se considera como potencial expansão [26, 27] (exemplo: a frase *file status* é uma expansão possível para o Id *fs_exists* → *file status exists*).

A lista de palavras irrelevantes e a lista de dicionários se constroem da mesma maneira que se faz no algoritmo Greedy (ver seção 3.2.1).

Quando as listas de palavras e frases potenciais estão criadas a execução do algoritmo se inicia. Este algoritmo (ver algoritmo 7) recebe como entrada a abreviatura a expandir e as quatro listas descritas previamente. O primeiro passo é ver se a abreviatura forma parte da lista de palavras irrelevantes (ou palavras a ignorar, linha 1). No caso que assim seja não se retornam resultados. A razão disto é porque estas palavras não aportam informação importante para a compreensão do código e são facilmente reconhecidas pelos engenheiros de software. Alguns casos são artigos/conectores (*the*, *an*, *or*) e palavras reservadas da linguagem de programação (*while*, *for*, *if*, etc.). Seguindo com a execução verifica-se se algumas das frases extraídas do código corresponde à abreviatura em forma de acrônimo (linha 5).

Em caso de não encontrar uma abreviatura se procura se as letras da abreviatura coincidem na mesma ordem com as letras duma palavra presente na lista de palavras recolhidas do código (linha 8). Exemplos: *horiz* → *horizontal*, *trgn* → *triangle*. No caso de não ter êxito, a procura continua, como último recurso, com o dicionário composto com palavras em inglês (linha 10). Esta técnica de expansão descrita só retorna uma única expansão potencial para uma abreviatura determinada; caso contrário não retorna coisa alguma (linhas 12 e 13). O motivo disto é porque não tem programado como decidir uma única opção entre várias alternativas de expansão. Esta funcionalidade é deixada, pelos autores, para trabalho futuro [7, 9].

4. Caso de Estudo

Nesta seção se descreve como funciona o AId aplicado a um caso de estudo, um programa Java de 600 linhas de código. Ao executar a ferramenta, o primeiro componente da interação é uma simples barra de menu localizada no topo da écran (ver figura 2). Ao fazer click no menu *Archivo* (ficheiro em português) da barra antes mencionada e logo em *Abrir Archivos(s) Java* (Abrir Ficheiro(s) Java) do menu suspenso mostra-se uma janela de seleção de ficheiros. Aqui o usuário pode escolher os ficheiros Java de entrada (na figura 2 o menu ficheiro está marcado com uma seta com um 1 e a janela de diálogo abrir com uma seta com um 2).

Uma vez escolhidos os ficheiros Java começa a execução do analisador sintático que extrai os Ids, comentários e literais. Ao finalizar a execução do analisador, aparecerá um painel que contem seções. Na parte de acima o código lido do ficheiro Java (ver seta com um 1 na figura 3). Na parte inferior, os elementos extraídos do código analisado.

Para proceder com a análise de Ids, se deve pulsar o botão *Ejecutar Análisis* (Executar Analise, ver seta com um 2 na figura 3) localizado no painel *Análisis de Identificadores* (Análise de Identificadores). Uma vez realizada essa tarefa se abre um novo painel; na parte inferior pode ver-se dos Ids recuperados (ver seta com um 1 na figura 4).

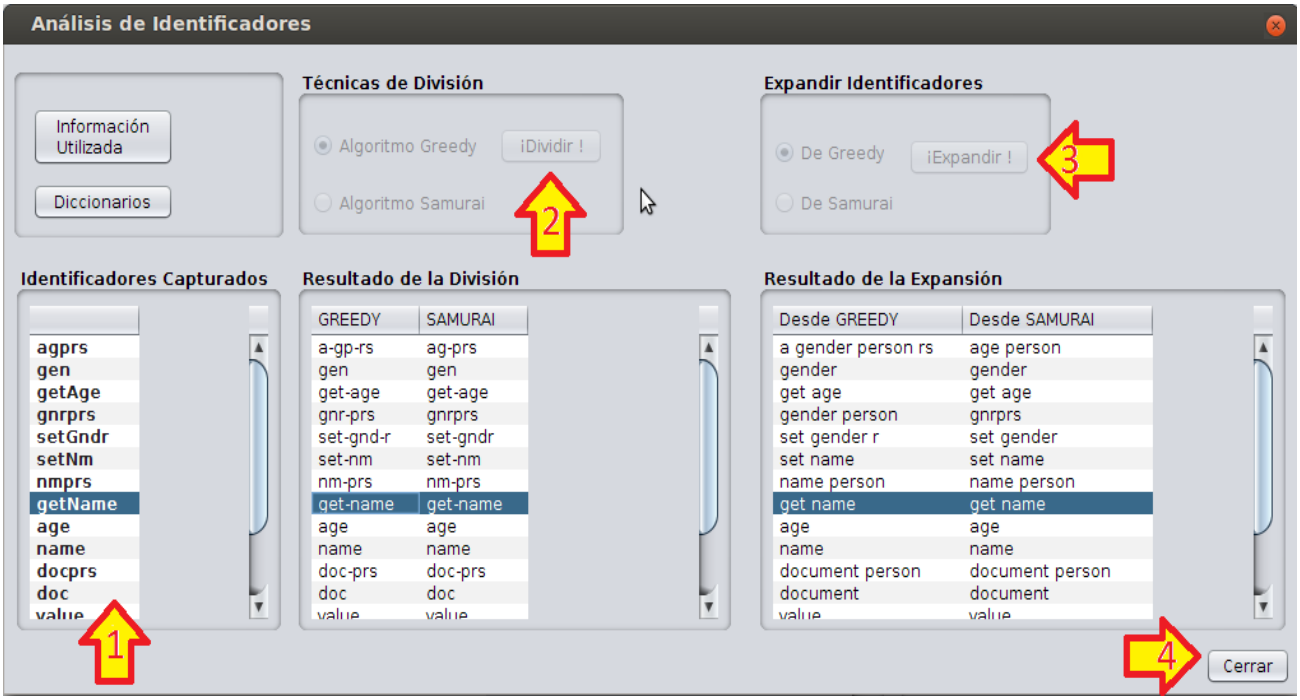


Figura 4 – Painel de Analise

No painel central superior se podem selecionar os algoritmos de divisão de Ids (Greedy e Samurai, ver seta com um 2 na figura 4), o botão *Dividir* do mesmo painel executa a técnica selecionada mostrando no painel inferior a tabela com os resultados da divisão (*Resultados de la División* na figura 4). Da mesma forma, os painéis subsequentes da direita permitem expandir as abreviaturas obtidas no painel de divisão (ver Figura 4). Pulsando o botão *Expandir* executa o algoritmo de expansão básico tomando como entrada os Ids divididos por Greedy ou Samurai, conforme o usuário decida (ver a seta com um 3 na figura 4). Os resultados se mostram no painel de abaixo. As três tabelas mostradas nos três painéis inferiores ajudam o usuário a comparar os resultados obtidos (ver figura 4).

Ao pulsar o botão *Cerrar* (Fechar em português e indicado pela seta com um 4 na figura 4), a janela do painel de análise se oculta e os resultados obtidos são carregados na tabela inferior do painel principal junto à informação que o analisador sintático havia recolhido (seta 1 na figura 5).

Na última coluna, se podem apreciar múltiplas caixas de seleção. Essas caixas permitem ao usuário escolher a expansão produzida pelo Greedy ou pelo Samurai (ver seta com um 1 na figura 5).

Uma vez concluída esta seleção, se procede a utilizar o painel *Expandir Id en Código* (seta 2 na figura 5). Este painel tem dois botões: *Expandir* e *Deshacer*. O primeiro leva a cabo a substituição dos Ids no código de acordo com o escolhido nas caixas de verificação mostrando o código resultante no painel de acima.

No caso de necessitar desfazer esta ação de substituição se pode pulsar o botão *Deshacer* (Desfazer em português) e restabelecer o código original. Para finalizar, ao pulsar o botão *Crear Archivo* (Criar Ficheiro em português indicado na seta 3 da figura 5) cria-se um novo ficheiro com o mesmo código porém com todos os Ids expandidos.

No exemplo mostrado anteriormente, o AId recebeu como entrada um programa Java de 600 linhas de código. Na tabela 1 se podem observar alguns identificadores analisados. Nas colunas *Algoritmo Greedy* e *Algoritmo Samurai* se podem apreciar os resultados de ter executado as técnicas de divisão. Logo nas próximas duas se mostra o resultado de haver aplicado o algoritmo de expansão básico a cada resultado dos algoritmos de divisão. Como se observa, a maioria dos casos de sucesso foram obtidos com o Samurai. Como observação importante se pode dizer que o Greedy consulta um grande volume de dados no dicionário predefinido e que

Identificador	Algoritmo Greedy	Algoritmo Samurai	Expansão do Greedy	Expansão do Samurai
nmprs	nm-prs	nm-prs	name person ✓	name person ✓
agprs	a-gp-rs	ag-prs	a gender person rs ✗	age person ✓
gnrprs	gnr-prs	gnrprs	gender person ✓	gnrprs ✗
docprs	doc-prs	doc-prs	document person ✓	document person ✓
age	age	age	age ✓	age ✓
setDocPrs	set-doc-prs	set-doc-prs	set document person ✓	set document person ✓
setGndr	set-gnd-r	set-gndr	set gender r ✗	set gender ✓

Tabela 1 – Resultado parcial do Analise

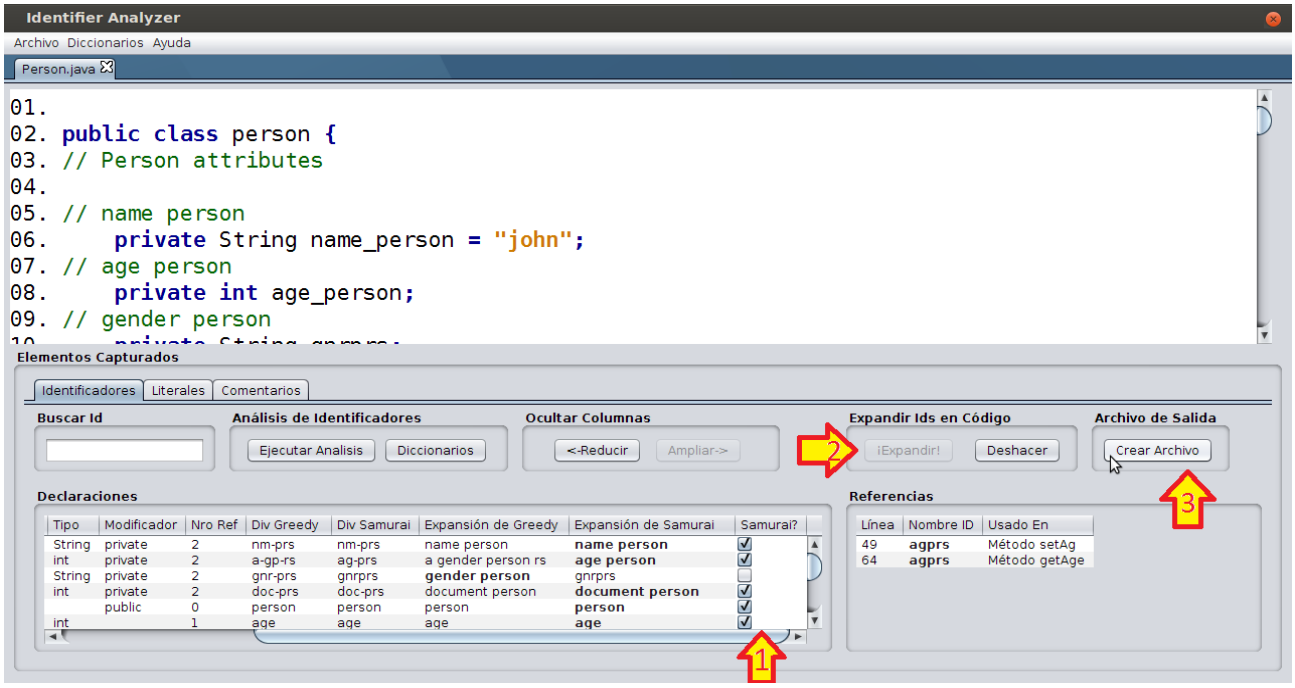


Figura 5 – Vista principal do Aid com foco na expansão de identificadores

o Samurai baseia-se nas tabelas de frequências com os dados próprios do código. O qual indica que este último deve ser mais preciso.

Seguindo com o caso de estudo, na parte superior da figura 6 pode-se observar uma parte do código em análise (dois métodos) com os identificadores não expandidos. Na parte inferior da figura 6, mostra-se o mesmo fragmento do programa depois de aplicar os algoritmos de expansão, ou seja, com os identificadores expandidos (ressaltados com cor amarelo). É evidente que a última versão dá mais informação semântica e, por conseguinte, torna mais fácil perceber o que faz essa parte do código.

5. Conclusão e Futuras Extensões

A motivação para desenvolver a ferramenta AId

```
public Person(int doc, String name, int age, String gen) {
    setDocPrs(doc);
    setNm(name);
    setAg(age);
    setGndr(gen);
}

public void setDocPrs(int value) {
    this.docPrs = value;
}
```

```
public Person(int document,String name,int age,String gender){
    set_document_person(document);
    set_name(name);
    set_age(age);
    set_gender(gender);
}

public void set_document_person(int value) {
    this.document_person = value;
}
```

Figura 6 – Comparação do Código em estudo, antes e depois da expansão de Identificadores

baseia-se na ausência de ferramentas com características similares. Não abundam ferramentas que possuam uma interface amigável com o usuário e executem técnicas de Análise de Identificadores (Ids) dum código recebido como entrada. Tão pouco foi possível encontrar muitas implementações que integrem extração, divisão, e expansão de abreviaturas de Ids. O uso do parser construído com ANTLR possibilitou extrair com facilidade os elementos estáticos presentes no código que são necessários para os algoritmos que analisam Ids. Estes elementos são os Ids como objetos principais, comentários, literais e documentação Java Doc. Cabe destacar que a ferramenta AId tem implementadas duas técnicas de divisão que são o Greedy e o Samurai. A primeira necessita consultar um dicionário de palavras em inglês e uma lista genérica de abreviaturas conhecidas para levar a cabo as suas tarefas. Ambas as estruturas ocupam muito espaço de armazenamento pelo que se utiliza uma base de dados para fazer as consultas mais eficientes [10, 5, 6, 7]. A segunda alternativa, o algoritmo Samurai [8], divide os Ids mediante a utilização de recursos específicos extraídos do código. Estes recursos são os comentários, os literais e documentação Java Doc. Com estes recursos, se constrói um vetor com as frequências de ocorrência das palavras, vetor esse que é depois usado na função de scoring (ver seção 3.2.2). Todavia é possível que estes recursos sejam escassos. Por esta razão os autores decidiram construir uma lista de palavras pertencente a um conjunto amplo de programas escritos em Java. Este vetor, não só ocupa menos espaço que os dicionários genéricos de Greedy como também é mais eficaz visto ser formado por palavras mais

adequadas no âmbito da engenharia do software. Isto implica que a divisão seja mais eficiente e, por conseguinte, que a expansão seja mais precisa. Por outro lado, o algoritmo de expansão básico usa os mesmos dicionários de palavras que utiliza o Greedy, mas com a diferença que consulta previamente a lista de frases capturadas do código, dando a preferencia a esta lista. É importante mencionar que AId incorpora na lista de frases não só os comentários como também as strings literais sendo esta uma caraterística não encontrada, no estado da arte, nas ferramentas atuais. Este algoritmo tem o problema que ante as múltiplas alternativas de expansão, não sabe escolher uma única opção. Como trabalho futuro se planeia implementar novas técnicas de análise de Ids. Uma delas é o AMAP (Automatically Mining Abbreviation Expansions in Programs) [9]. Esta técnica não necessita de dicionários com palavras em inglês, como é o caso do algoritmo básico de expansão; os seus proponentes observam gradualmente no código os comentários e literais presentes partindo desde o lugar do Id que se deseja expandir. Também resolve o problema que possui o algoritmo básico quando não conhece que opção escolher ante muitas opções de expansão. Para levar a cabo a tarefa atrás mencionada, o algoritmo prioriza a frequência de aparição das palavras. Isto se faz partindo do lugar onde se encontre o Id analisado. Também AMAP permite treinar com um conjunto de programas recebidos como entrada para recolher mais palavras e melhorar ainda mais a precisão da expansão. Outra melhora futura para a ferramenta AId é a possibilidade de construir um plugin para o NetBeans ou Eclipse. Isto permitirá que o usuário abra um projeto Java e imediatamente com Aid expanda os Ids para melhorar a compreensão. Também se pretende, nos próximos passos da investigação, estudar e propor novas técnicas de análise de identificadores e fazer um estudo profundo das suas vantagens, no que respeita à compreensão, para um conjunto significativo de sistemas.

6. Referências

[1] Florian DeiBenbock and Markus Pizka. Concise and Consistent Naming. In IWPC, page 97-106. IEEE Computer Society, 2005.

[2] Dave Binkley, Marcia Davis, Dawn Lawrie, Jonathan Maletic, Christopher Morrell, and Bonita Sharif. The impact of identifier style on effort and comprehension. Empirical Software Engineering, 18(2):219-276, 2013.

[3] Dawn Lawrie, Henry Feild, and David Binkley. Syntactic Identifier Conciseness and Consistency. In SCAM, page 139-148. IEEE Computer Society, 2006.

[4] D. Lawrie, C. Morrell, H. Feild, and D. Binkley. What's in a Name? A Study of Identifiers. In Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on, page 3-12, June 2006.

[5] Henry Feild, David Binkley, and Dawn Lawrie. An empirical comparison of techniques for extracting concept abbreviations from identifiers. In Proceedings of IASTED International Conference on Software Engineering and Applications, 2006.

[6] Binkley D. Feild, H. and D. Lawrie. Identifier splitting: A study of two techniques. In Proceedings of the Mid-Atlantic Student Workshop on Programming Languages and Systems, 2006.

[7] D. Lawrie, H. Feild, and D. Binkley. Extracting Meaning from Abbreviated Identifiers. In Source Code Analysis and Manipulation, 2007. SCAM 2007. Seventh IEEE International Working Conference on, page 213-222, Sept 2007.

[8] E. Enslen, E. Hill, L. Pollock, and K. Vijay-Shanker. Mining source code to automatically split identifiers for software analysis. In 6th IEEE International Working Conference on Mining Software Repositories, page 71-80. IEEE, may. 2009.

[9] Emily Hill, Zachary P. Fry, Haley Boyd, Giriprasad Sridhara, Yana Novikova, Lori Pollock, and K. Vijay-Shanker. AMAP: Automatically Mining Abbreviation Expansions in Programs to Enhance Software Maintenance Tools. In Proceedings of the 5th Int'l Working Conf. on Mining Software Repositories., page 79-88. ACM, 2008.

[10] Henry Field, Dawn Lawrie and David Binkley. Quantifying identifier quality: an analysis of trends. Kluwer Academic Publishers-Plenum Publishers. 2006.

[11] Fangfang Feng and W. Bruce Croft. Probabilistic techniques for phrase extraction. Inf. Process. Manage., 37(2):199-220, 2001.

[12] Bruno Caprile and Paolo Tonella. Nomen est omen: analyzing the language of function identifiers. In Proc. Sixth Working Conf. on Reverse Engineering, page 112-122. IEEE, October 1999.

[13] José Luís Freitas, Daniela da Cruz, and Pedro Rangel Henriques. The role of comments on program comprehension. In INForum.

[14] Bruno Caprile and Paolo Tonella. Restructuring program identifier names. In Proc. Int'l Conf. on Software Maintenance, page 97-107. IEEE, 2000.

[15] K. Bennett and V. Rajlich. Software maintenance and evolution: a roadmap. In ICSE - Future of SE Track, page 73-87, 2000.

[16] Roger S. Pressman. Software Engineering - A Practitioner's Approach. McGraw-Hill, 5 edition, 2001.

[17] P.F. Tiako. Maintenance in joint software development. In Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International, page 1077-1080, 2002.

[18] A. Von Mayrhauser and A. M. Vans. Program comprehension during software maintenance and evolution. Computer, 28(8):44-55, 1995.

[19] M. Beron, P. Henriques, and R. Uzal. Inspección de Programas para Interconectar las Vistas Comportamental y Operacional para la Comprensión de Programas. PhD thesis, Universidade do Minho, Braga. Portugal, 2010.

[20] M. Storey. Theories, methods and tools in program comprehension: past, present and future. In Program Comprehension, 2005. IWPC 2005. Proceedings. 13th International Workshop on, page 181-191, May 2005.

[21] Michael P. O'Brien. Software comprehension - a review and research direction. 2003.

[22] Mario Beron, Pedro R. Henriques, Maria J. Varanda, and Roberto Uzal. Program inspection to inter-connect the operational and behavioral views for program comprehension. 2007.

[23] David W Embley. Toward semantic understanding: an approach based on information extraction ontologies. In Proceedings of the 15th Australasian database conference-Volume 27, pages 3-12. Australian Computer Society, Inc., 2004.

[24] Dawn Lawrie, Henry Feild, and David Binkley. An empirical study of rules for well-formed identifiers. Journal of Software Maintenance and Evolution: Research and Practice, 19(4):205-229, 2007.

[25] G. Antoniol, G. Canfora, G. Casazza, A. D. Lucia, and E. Merlo. Recovering traceability links between code and documentation. IEEE Trans. Soft. Eng., 28(10):970–983, 2002.

[26] L. S. Larkey, P. Ogilvie, M. A. Price, and B. Tamilio. Acrophile: an automated acronym extractor and server. In Proc. Conf. Digital Libraries, 2000.

[27] S. Pakhomov. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In Proc. Association for Computational Linguistics, 2001.

[28] A. V. Aho, R. Sethi, and J. D. Ullman. Compilers principles, techniques and tools. 2006.

[29] Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft. Data structures and algorithms / Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. Addison-Wesley Reading, Mass, 1983.

[30] Thomas Eisenbarth, Rainer Koschke, and Daniel Simon. Aiding Program Comprehension by Static and Dynamic Feature Analysis. In ICSM, page 602-611, 2001.

[31] Albanes, J.L., Berón M., Henriques P. and Pereira MJ. In WICC 2011. Estrategias para relacionar el dominio del problema con el dominio del programa para la CP, 2011.

QuVi: Un enfoque estadístico para la obtención de sistemas Web de Calidad

Nicolás Tortosa, Noelia Pinto, Liliana Cuenca Pletsch, César J. Acuña, Bruno Demartino

Grupo de Investigación en Ingeniería y Calidad de Software (GICS), Universidad Tecnológica Nacional, Facultad Regional Resistencia.

Abstract

En los últimos años se ha incrementado notoriamente la necesidad de contar con Sistemas de Información basados en Web. Así, a medida que el número y la complejidad de las aplicaciones web crecen, es necesario mejorar sus procesos de obtención y analizar cómo mantener su calidad e integridad. En este sentido hemos presentado trabajos anteriores basados en WQF un framework para la Evaluación de Calidad de Aplicaciones Web, compuesto por un Modelo de Calidad (WQM) y un plugin web para realizar los procesos de evaluación de calidad (QUCO2). Como herramienta complementaria, en este artículo se presenta QuVi, una aplicación para la emisión de informes estadísticos en base a la información proporcionada por WQF, con el objetivo de permitir a los directivos de las organizaciones tomar decisiones en pos de la mejora de calidad en aplicaciones Web. Finalmente se incluyen los trabajos futuros que se afrontarán para afinar la herramienta presentada.

1. Introducción

La Ingeniería Web, como disciplina emergente, debe hacer frente a algunas dificultades respecto a los sistemas, actualmente, en desarrollo. Gran parte de estas dificultades surgen de la interacción entre las diversas áreas que forman la base de los Sistemas Web: sistemas de información, diseño gráfico, marketing, etc. Justamente, la Ingeniería Web consiste fundamentalmente en la integración eficaz de estos aspectos para lograr el desarrollo efectivo de aplicaciones de calidad [1]. Así, los desarrolladores tienen dos opciones: adoptar un enfoque ad hoc para el desarrollo con poco control sobre la calidad y un alto riesgo de lograr soluciones que no se ajusten a las necesidades reales, o adoptar un enfoque bien estructurado, basado en los principios de la Ingeniería Web y sus mejores prácticas, para la obtención de sistemas Web de calidad.

En los últimos años han surgido varias propuestas que presentan métodos, técnicas y herramientas para medir la calidad de las aplicaciones que se desarrollan. Por este motivo, la evaluación de calidad corresponde a un aspecto fundamental de la Ingeniería Web [2].

Sin embargo, más allá de la existencia de modelos de calidad y herramientas que permiten la evaluación de aspectos relacionados a la calidad, surge la necesidad de ofrecer a la Industria del Software, herramientas que faciliten el proceso de toma de decisiones en las empresas dedicadas al desarrollo de software Web que deseen producir aplicaciones de calidad.

En este sentido, se viene trabajando en el Grupo de Investigación en Ingeniería y Calidad de Software (GICS) de la Facultad Regional Resistencia de la Universidad Tecnológica Nacional (UTN-FRRe), en el desarrollo e implementación de un framework para la Evaluación de Calidad de Aplicaciones Web, WQF [3]; el cual actualmente se compone de un Modelo de Calidad (WQM) y de un plugin web para llevar a cabo los procesos de evaluación de calidad (QUCO2) por parte de usuarios finales.

Consecuentemente con esta línea de trabajo, en este artículo se presenta una aplicación Web, denominada QuVi, cuyo desarrollo, como componente del WQF, se encuentra en plena ejecución y que tiene como objetivo obtener informes estadísticos en base a la información obtenida y almacenada por QUCO2, que permitan tomar decisiones tendientes a mejorar la calidad de las soluciones Web desarrolladas.

El artículo se organiza en secciones. En la sección 2 se incluye una breve referencia al framework WQF, luego en la sección 3 se presenta la arquitectura propuesta para el software QuVi que extenderá las funcionalidades de WQF, y permitirá generar informes a partir de la información recolectada por otra de las herramientas de WQF. Finalmente en la sección 4 se presentan discusiones y líneas de trabajos futuros.

2. Framework para Evaluación de Calidad de Aplicaciones Web (WQF)

Con el objetivo de gestionar los elementos de calidad de una aplicación Web se desarrolló WQF (Web Quality

Framework), que incluye hasta el momento la definición de un modelo de calidad (WQM) y QUCO2, una herramienta que almacena información respecto a los resultados obtenidos en las evaluaciones.

2.1. WQM: Modelo de Calidad para Aplicaciones Web

Teniendo en cuenta trabajos anteriormente publicados [4][5], brevemente se presenta aquí el modelo de calidad desarrollado. El mismo, denominado WQM (Web Quality Model), incluye, hasta el momento, el estudio de 3 métricas, que a su vez engloban un conjunto de características que son objeto de evaluación:

a) Usabilidad: Permite medir la facilidad con la que las personas puede utilizar el sitio web para llegar a alcanzar sus propios objetivos.
b) Confiabilidad: A partir de esta métrica se evalúa el grado de seguridad del sitio web permitiendo siempre mantener la misma performance en el tiempo bajo las demandas por las cuales fue definido.
c) Funcionalidad: Esta métrica posibilita corroborar que el Sitio Web cumpla con las funciones que fueron definidas, evaluando la relación entre los resultados que se obtienen y los resultados esperados.

Los criterios de calidad que se evalúan por métrica, junto a las opciones de cada uno, se consignan en la Tabla 1.

Tabla 1. Métricas, Criterios y Opciones de evaluación.

Métrica	Criterios de calidad	Escala	Valor
Usabilidad	Facilidad de Aprendizaje	Muy complicado Complicado Fácil Muy fácil	-2 -1 1 2
	Retención en el tiempo		
	Consistencia	Consistencia en todo el sitio Consistencia en la mayor parte del sitio Inconsistencia en todo el sitio	3 2 1
	Recuperabilidad	Si No	1 -1
	Flexibilidad	El Sitio presenta diferentes opciones para una misma tarea El Sitio presenta una única alternativa para una tarea. Pero no presenta inconvenientes para llegar hasta la misma Es difícil intercambiar información dentro del Sitio No se puede intercambiar información en el Sitio	3 2 1 -1
Confiabilidad	Frecuencia y severidad de las fallas	Errores graves continuamente Errores leves continuamente Errores leves sólo en ciertas funciones No se observan errores	-2 -1 1 2
	Exactitud de las salidas Capacidad de Recuperación ante fallas Confiabilidad	Si No	1 -1
Funcionalidad	Adecuación Cumplimiento Confiabilidad	Si No	1 -1
	Seguridad	Completamente seguro Muy seguro Seguro Se observan zonas inseguras Completamente inseguro	3 2 1 -1 -2

2.2. QUCO2: Software para la evaluación de calidad de aplicaciones web

QUCO2, es el componente de WQF que permite gestionar los elementos incluidos en el modelo y almacenar información respecto a los resultados obtenidos luego de cada proceso de evaluación ejecutado. Esta herramienta se presenta como un plugin on-line, desarrollado con tecnología open-source, que admite la evaluación simultánea de diversas aplicaciones web por parte de un número ilimitado de usuarios finales, y ofrece al final del proceso un valor promedio de calidad asociado al sistema Web evaluado.

El software presenta interfaces diferentes, de acuerdo al usuario logueado. Hasta el momento se consideran 3 niveles: Evaluadores (cuya pantalla principal se muestra en la Figura 1), Desarrolladores y Administradores.

La información que se genera en cada proceso de evaluación es almacenada, y se encuentra disponible para su utilización desde otras fuentes de consulta, por ejemplo el software en desarrollo que aquí se presenta.

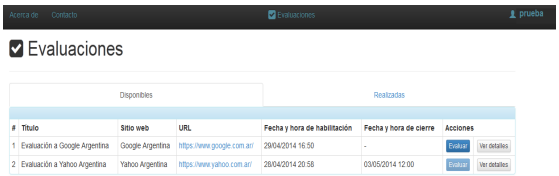


Figura 1. Menú Principal QUCO2 – Nivel Usuarios

3. QuVi: Software Web para la emisión de informes estadísticos

Como se explicó en las secciones anteriores, QUCO2 sólo almacena información respecto al valor final de calidad asociado a los procesos de evaluación llevados a cabo sobre diversos sistemas Web. Sin embargo, resulta necesario analizar los resultados de dichas evaluaciones llevadas a cabo por los usuarios finales para que, en base a ello, puedan tomarse decisiones orientadas a mejorar la calidad de los productos desarrollados en las empresas.

En este sentido, se ha diseñado una aplicación Web, denominada QuVi, que permita la emisión de informes estadísticos en base a la información recolectada por el uso de QUCO2. El desarrollo de esta herramienta se basa en la Estadística Descriptiva, disciplina que engloba métodos para la recolección, presentación y caracterización de un conjunto de datos con el fin de describir adecuadamente las características de los mismos [6].

Para mayor claridad, se describe aquí la arquitectura de la aplicación, el diseño de los reportes que incluirá y el significado de la información que ofrecerá para la toma de decisiones.

3.1. QuVi por dentro: Arquitectura y tecnología usada para su desarrollo

Para la realización de este proyecto se han utilizado diversas tecnologías. La principal es Python, pues el framework utilizado para realizar la aplicación, Django, está basado en este lenguaje.

La elección del framework de desarrollo Django surge debido a las siguientes características asociadas:

- Es un framework de desarrollo que promueve el desarrollo rápido de aplicaciones, se construyen aplicaciones en poco tiempo [7].
- Pone énfasis en el re-uso, la conectividad y extensibilidad de componentes y el principio “No lo repitas” (DRY, del inglés Don't Repeat Yourself) [8].
- Utiliza un Sistema de Plantillas, que permite separar diseño y programación. Es decir, es posible editar el HTML sin modificar el código Python. Así, Django denomina MPV (Modelo Plantilla Vista) a la arquitectura de desarrollo Web. El Modelo se encarga del manejo de los datos, la Plantilla sirve para la presentación de la información y la Vista se encarga del control y funcionalidad de la aplicación Web [9].

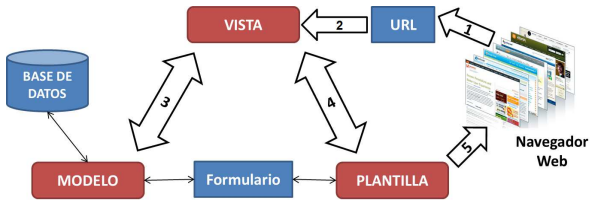


Figura 2. Arquitectura de Componentes de DJANGO

En la Figura 2 se presenta un diagrama de la interacción entre los diferentes componentes de Django para el funcionamiento de un sitio web [10]:

1. Un usuario envía una solicitud a través del navegador web.
2. El módulo *URL* interpreta la solicitud del usuario y la reenvía a la “Vista” correspondiente para que la procese.
3. La *Vista* interactúa con el *Modelo*, el cual hace directamente las consultas necesarias a la base de datos.
4. La *Vista* entrega la información procesada a la *Plantilla*, la cual generará el código HTML que presentará en el navegador web.
5. La *Plantilla* puede obtener datos directamente de los *Modelos* por medio de los *Formularios*, sin necesidad de la intervención de la *Vista*.

Para la obtención de gráficos estadísticos se utilizó la librería gráfica PyGal, escrita en Python y Javascript. La misma permite generar gráficos SVG estadísticos de forma dinámica pudiendo ser personalizados cada uno de ellos [11].

Además también se ha trabajado con la librería Request, con el objetivo de poder integrar los servicios web y consumir la información generada en QUCO2 [12].

En el caso de QuVi, los informes estadísticos se obtendrán a partir de los datos recolectados y almacenados por el uso de QUCO2.

El software solamente permite el acceso de usuarios habilitados, que serán aquéllos que analizarán la información brindada en los informes y tomarán decisiones en base a ello. Inicialmente se presentará un resumen de los sitios evaluados, y tal como se indica en la Figura 3, se ofrece al usuario la opción de acceder a los informes desde aquí. En esta pantalla inicial se incluye:

- Sitio Web
- URL
- Cantidad de Evaluadores (indica cuántos usuarios con perfil de Evaluador han utilizado QUCO2 para evaluar la aplicación web a estudiar).
- Período de Evaluación (especifica en qué fecha se inició y en qué fecha finalizó el proceso de evaluación).



Figura 3. Inicio de QuVi

3.2. Herramientas estadísticas y diseño de reportes

Los métodos estadísticos son herramientas eficaces para mejorar la producción y reducir defectos; ofrecen objetividad y precisión en las observaciones [13]. Entre

las herramientas para el control de calidad más estudiadas, se encuentra el DEE (Diseño de Experimentos Estadísticos, o Statistical Experimental Design) que se basa en el estudio de todas las variables que controlan las características de calidad en el producto, pero discriminando la contribución de cada una con posterioridad a la experimentación [14].

Así, en base a esta técnica y teniendo en cuenta los conceptos de la Estadística Descriptiva, se han diseñado diversos informes para QuVi, lo que permite a través del análisis de diversos casos de estudios y sus resultados, obtener conclusiones respecto a la calidad de los sistemas Web en estudio, y decidir acciones futuras.

3.2.1. Informe de Evaluación de Calidad

De acuerdo a la aplicación web que se seleccione para su estudio, este reporte informa el valor promedio de calidad asociado al nivel de calidad obtenido, según lo expresado en el modelo WQM.

El modelo indica que el valor de calidad máxima ocupa el rango entre 20 y 22, el valor de calidad media el rango entre 14 y 20, el valor de calidad regular entre 8 y 14 y el mínimo nivel con un valor por debajo de 7 [15]. Para una mejor lectura, no sólo se presenta el valor y el nivel, sino que gráficamente se asocia un color tal como se indica en la Tabla 2, lo que ofrece, al usuario final, mayor legibilidad al analizar este informe.

Tabla 2. Relación entre Nivel, Rango y Color.

NIVEL	RANGO	COLOR
Calidad Máxima	[20,22]	
Calidad Media	[14,20]	
Calidad Regular	[8,14]	
Calidad Baja	[-,8]	

En la aplicación QuVi, al ingresar al Informe de Evaluación de Calidad se muestra al usuario el detalle de las evaluaciones realizadas hasta el momento. En la Figura 4 se muestra un ejemplo de la información proporcionada al usuario que desee conocer datos de evaluación de un determinado sitio web.

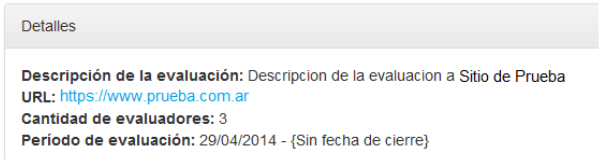


Figura 4. Detalle del Sitio Web evaluado

Luego se muestra el Nivel de calidad promedio, que incluye valor y color asociado tal como se indica en la Figura 5:

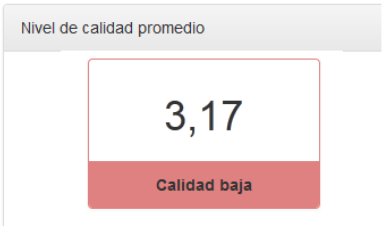


Figura 5. Ejemplo de Informe de Evaluación de Calidad

Para complementar a este informe, se ofrecen 2 (dos) gráficos estadísticos:

- *Diagrama de Frecuencias:* Representado por un gráfico de columnas en el cual las ordenadas representan la frecuencia con la cual determinada magnitud toma el valor o conjunto de valores que Figura en abscisas [16]. Por ejemplo en la Figura 6, se muestra el Diagrama de Frecuencias obtenido por QuVi luego del análisis de los resultados de evaluación. En este gráfico el eje "Y" incluye los valores en promedio obtenidos por cada métrica, Usabilidad y Confiabilidad, las cuales se representan en el eje "X". Esto permite identificar cómo influye cada métrica al Valor de Calidad, de manera que podrán tomarse decisiones tendientes a ajustar aquellas variables que afecten la calidad del producto evaluado.

En el caso representado en la Figura 6, se observa que para la aplicación evaluada la métrica de Usabilidad impacta en mayor medida que la métrica de Confiabilidad, por lo que ésta última deberá analizarse en términos de los criterios que la definen.

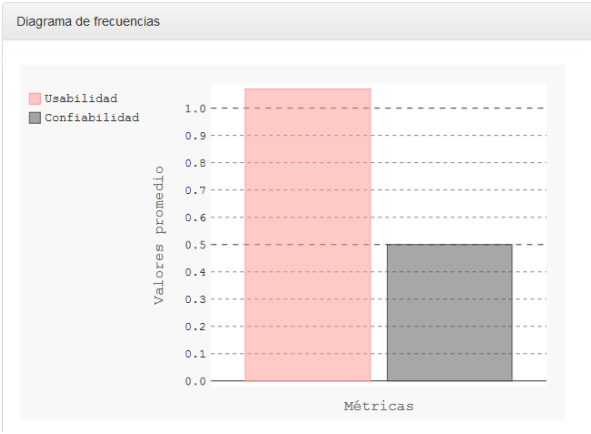


Figura 6. Ejemplo de Diagrama de Frecuencias

- *Gráfico de Caja y Bigotes*: Este diagrama permite analizar y resumir un conjunto de datos, facilitando el estudio de la simetría de datos y la detección de valores atípicos. En este gráfico se dividen los datos en cuatro áreas de igual frecuencia, una caja central

dividida en dos áreas por una línea vertical y otras dos áreas representadas por dos segmentos horizontales, denominados bigotes, que parten del centro de cada lado vertical de la caja [17]. Como se muestra en la Figura 7, ofrece una representación visual del Nivel de Calidad basada en el resumen de 5 valores: Valor Mínimo, Valor Máximo, Cuartil 1, Mediana y Cuartil 3.

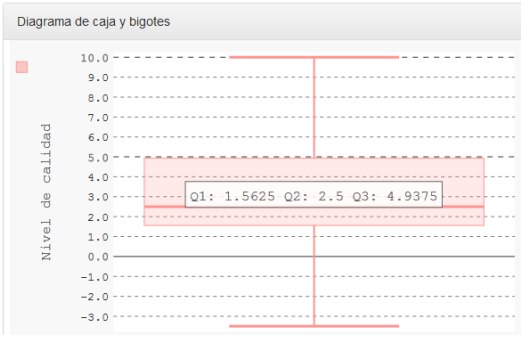


Figura 7. Ejemplo de Diagrama de Caja y Bigotes

Mientras que los bigotes marcan los valores mínimo y máximo, menor que -3 y 10 respectivamente, la caja concentra el 50% de los valores obtenidos. Así, para la mitad de la muestra se obtuvo un Nivel de Calidad entre 1.56 (Cuartil 1) y 4.94 (Cuartil 3). Por su parte la mediana, dividiendo la muestra a la mitad, toma un valor de 2.5.

Del gráfico ejemplo se observa que, si bien para algunos evaluadores se obtuvo un valor promedio cercano a una Calidad Regular, para más del 75% la calidad fue Baja. Se destaca además que la mediana, siendo menos afectada por los valores extremos que el promedio, resultó en un valor menor. Esto indica que más allá de haberse obtenido algunos resultados altos, el Nivel de Calidad se ve mejor representado por valores alrededor de 2.5, un nivel que difícilmente se acerca a la Calidad Regular.

De esta forma se puede apreciar que el diagrama de Cajas y Bigotes proporciona una visión general de la simetría de la distribución de los datos; si la mediana no está en el centro del rectángulo, la distribución no es simétrica. Es un informe que resulta muy útil para que los directivos analicen el impacto de valores atípicos y en qué aspectos se dan con mayor frecuencia.

3.2.2. Informe de Detalle por Métrica

En este informe se tiene en cuenta lo definido en el modelo WQM, así para cada métrica (Usabilidad, Confiablidad o Funcionalidad) QuVi obtiene también Diagramas de Frecuencias y Diagrama de Caja y Bigotes, basados en el valor promedio para la Métrica y una desagregación de la misma en Criterios.

Por ejemplo, la Figura 8 muestra un Diagrama de Frecuencias para la métrica Usabilidad, teniendo en cuenta la evaluación de un sitio web tomada como caso de estudio. Así se puede observar que, para esta evaluación, la Consistencia del sitio web analizado es la característica mejor ponderada por los evaluadores, por otro lado, la Retención en el tiempo presenta un valor promedio negativo. Utilizando esta información estadística, los Gerentes podrán solicitar mejoras respecto a los criterios que han obtenido menor valor promedio y que pueden afectar en gran medida a la Usabilidad, en este caso, del software evaluado. Asimismo, se podrán tener en cuenta las cuestiones relacionadas a la Consistencia y replicarse en el resto de las aplicaciones similares, pues evidentemente es lo que mejor se consideró en la evaluación.

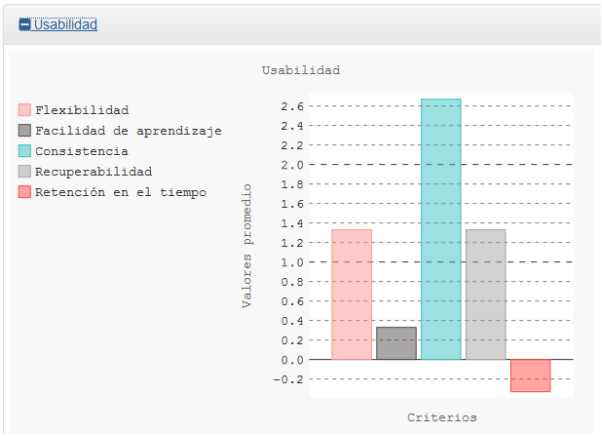


Figura 8. Ejemplo de Diagrama de Caja y Bigotes para el Informe de Detalle por Métrica: Usabilidad

3.2.3. Informe de Detalle por Criterio

Este informe será útil para los casos en que se necesitara mayor nivel de detalle al presentar la información, pues incluye el valor promedio del Criterio, junto con un diagrama de frecuencias en el que se grafica cuántas veces se ha seleccionado cada opción.

El caso de este informe, QuVi primero presenta, tal como se muestra en la Figura 9, los criterios por métrica y da la opción de elegir cuál se quiere analizar.



Figura 9. Lista de criterios para la métrica Usabilidad

Luego de seleccionar el criterio, la aplicación muestra un Diagrama de Frecuencias que grafica el resultado del proceso de evaluación teniendo en cuenta cada una de las opciones. En el caso de la métrica Recuperabilidad, que ofrece dos opciones al evaluador (SI – NO), se muestra un ejemplo del gráfico en la Figura 10.

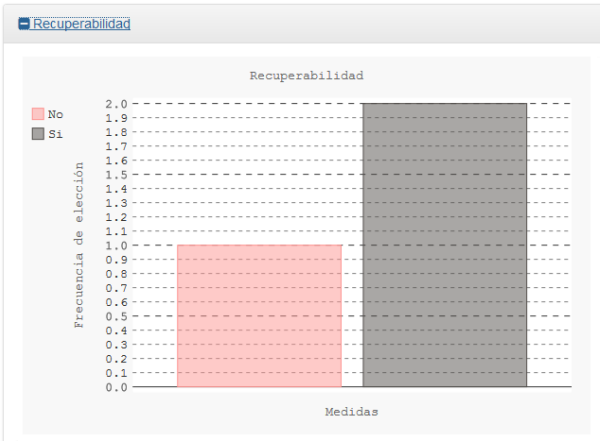


Figura 10. Ejemplo de Diagrama de Frecuencias por opción para el criterio Recuperabilidad

4. Conclusiones y Trabajos Futuros

La aplicación desarrollada permite obtener informes estadísticos en base a la información obtenida y almacenada por la herramienta de evaluación de calidad de aplicaciones web (QUCO2), aportando a la toma de decisiones vinculadas con la mejora de la calidad de las soluciones Web desarrolladas.

Se considera una herramienta de gran utilidad para las empresas dedicadas al desarrollo de SW y para las instituciones que cuentan con aplicaciones WEB, tanto para la gestión interna como para la interacción con los clientes.

Se destaca asimismo que al ser una aplicación open-source facilita su implementación y uso en cualquier ámbito.

Como trabajos futuros se propone continuar con el desarrollo del framework, en primer lugar incluyendo las métricas faltantes (Mantenibilidad, Seguridad, Disponibilidad y Escalabilidad) al modelo de calidad y a la herramienta de software a los efectos de obtener las estadísticas correspondientes.

También se prevé someter QuVi a la evaluación por parte de las empresas de Software asociadas a los Polos IT Chaco y Corrientes, a efectos de recabar sus aportes para mejorar e incrementar su utilidad.

5. Referencias

[1] Diaz, P., Montero, S., Aedo, I., "Ingeniería de la Web y Patrones de Diseño". Ed. Pearson Educación. ISBN 84-205-4609-7. Madrid, España. 2003.

[2] González, M., Abrahão, S., Fons, J., Pastor, O., "Evaluando la Calidad de Métodos para el Diseño de Aplicaciones Web". I Simpósio Brasileiro de Qualidade de Software. 2002. En: http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2002/011.pdf

[3] Pinto, N., Tortosa, N., Cuenca Pletsch, L., Acuña, C., Greiner, C., Estayno, M., "Aproximación a la Evaluación de la Calidad de Aplicaciones Web". Revista de Ciencia y Tecnología. Año XIII. N° 13. Universidad de Palermo.

[4] Tortosa, N., Pinto, N., Acuña, C., Cuenca Pletsch, L., Estayno, M.: "QUCO2: Desarrollo de una herramienta para medir la calidad de aplicaciones Web". 1° Congreso Nacional de Ingeniería en Sistemas de Información (CONAIISI 2013).

[5] Tortosa, N., Pinto, N., Acuña, C., Cuenca Pletsch, L., Estayno, M.: " Development of a tool for measuring the quality of Web applications ". X Workshop de Ingeniería de Software. CACIC 2013.

[6] Berenson, M., Levine, D.: "Estadística básica en administración: conceptos y aplicaciones". 6ta. Edición. Pearson Educación. ISBN: 968-880-784-2. México. 1996.

[7] Condori Ayala, J.: "Phython - DjangoFramework de desarrollo web para perfeccionistasBasado en el Modelo MTV". Revista de Información, Tecnología y Sociedad. ISSN 1997-4044. La Paz. 2012.

[8] Cumba Armijos, P., Barreno Pilco, B.: "Análisis de Python con Django frente a Ruby on Rails para Desarrollo Ágil de Aplicaciones Web. Caso Práctico: DECH". Tesis de Grado. Riobamba, Ecuador. 2012.

[9] Núñez Espinoza, D.: "Implementación de un prototipo de Software como Servicio (SAAS) para Pequeñas y Medianas Empresas". Quito, Ecuador. 2013.

[10] Información disponible en http://www.djangoproject.com

[11] Información disponible en http://www.python.org

[12] Información disponible en http://www. docs.python-requests.org

[13] Kume, H., Vasco, E.: "Herramientas estadísticas básicas para el mejoramiento de la calidad". Editorial Norma S.A. ISBN: 958-04-6719-6. Bogotá, Colombia. 1992.

[14] Sadarreta Jorba, L.: "El secreto de la calidad japonesa: el diseño de experimentos clásico, Taguchi y Shainin". Marcombo S.A. ISBN: 84-267-0913-3. Barcelona, España. 1993.

[15] Pinto, N., Tortosa, N., Cuenca Pletsch, L., Acuña, C., Estayno, M.: “QUCO2: Una herramienta para medir la calidad de aplicaciones Web”. WICC 2013.

[16] Hoffman, A.: "Los gráficos en la gestión". Editores Técnicos Asociados S.A. Barcelona, España. 1974.

[17] Pérez López, C., Santín González, D.: "Minería de datos: técnicas y herramientas". Editoria Paraninfo. ISBN: 978-84-9732-492-2. Madrid, España. 2008.

Ingeniería Inversa en el Contexto de ADM

Claudia Pereira[1], Liliana Martinez[1], Liliana Favre[1,2]
[1]*Universidad Nacional del Centro de la Provincia de Buenos Aires*
Paraje Arroyo Seco, B7000, Tandil, Argentina
[2]*Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, La Plata, Argentina*
{cpereira, lmartine, lfavre}@exa.unicen.edu.ar

Resumen

El proceso de analizar los artefactos de sistemas de software existentes para extraer información y proveer vistas de alto nivel del sistema (ingeniería inversa) es una etapa crucial de la modernización de software. La iniciativa Architecture–Driven Modernization (ADM) fue propuesta por OMG (Object Management Group) con el objetivo de definir estándares para soportar el proceso de modernización en el contexto de desarrollos dirigidos por modelos. Dentro de los estándares se definieron metamodelos relevantes para la ingeniería inversa tales como Knowledge Discovery Metamodel (KDM) y Abstract Syntax Tree Metamodel (ASTM). Se propone en este trabajo aplicar los principios de ADM para proveer vistas de alto nivel de sistemas heredados. Se describe un framework para la ingeniería inversa de código orientado a objetos a fin de extraer modelos UML. Se ejemplifica la propuesta con un caso de estudio que muestra cómo recuperar diagramas de casos de uso y de interacción a partir de código Java. La propuesta fue validada utilizando el framework de modelado Eclipse y la plataforma MoDisco que puede considerarse como la implementación oficial de los estándares ADM.

1. Introducción

Gran parte de los sistemas de información vitales en organizaciones de nuestro medio fueron implementados hace varios años con tecnologías que hoy pueden considerarse obsoletas, y no alineadas con los actuales objetivos estratégicos de las organizaciones. Estos sistemas, conocidos como heredados, involucran software, hardware, procesos de negocio y estrategias organizacionales. En general no están documentados o, si lo están, sus especificaciones no reflejan los cambios de requerimientos que se dieron a través de los años, englobando sólo el código la historia de su evolución. La modernización de software se refiere a la transformación de un sistema de software existente en uno nuevo que satisfaga nuevas demandas. Para tratar este problema, existen distintas soluciones, típicamente nuevos desarrollos, *wrapping* o migración, cada una de las cuales impactan en mayor o menor medida sobre el sistema y como consecuencia sobre la organización. Una buena solución debería permitir recuperar el valor del sistema existente extrayendo la inversión y el conocimiento para transformarlo en uno nuevo que incorpore los nuevos requerimientos y/o tecnologías.

El éxito de la modernización dependerá del grado de automatización de las herramientas CASE que soportan el proceso y de la estandarización de los artefactos heredados. Nuevos enfoques de desarrollo de software dirigidos por modelos enmarcados en lo que se referencia como MDD (Model-Driven Development) podrían dar respuesta a esta demanda. Las técnicas de metamodelado y transformación de modelos en las que se basa MDD pueden ser usadas en este proceso ya que permiten la automatización de muchas actividades básicas incluyendo la manipulación de código [1]. En este contexto, una nueva propuesta conocida como Architecture-Driven Modernization (ADM) ha surgido como una evolución de Model-Driven Architecture (MDA) y sus estándares, con el objetivo de establecer soluciones para la modernización de los sistemas de información [2] [3]. OMG ADM Task Force (ADMTF) está desarrollando un conjunto de estándares para facilitar la interoperabilidad entre las herramientas de modernización.

ADM lleva a cabo procesos de modernización considerando los principios esenciales de MDA: la representación de artefactos mediante modelos en diferentes niveles de abstracción y las transformaciones entre modelos. MDA distingue al menos los siguientes tipos de modelos: CIM (Computation Independet Model), PIM (Platform Independent Model) y PSM (Platform Specific Model). Una de las características esenciales de MDA es que todos los artefactos involucrados en un proceso de desarrollo son representados a partir del lenguaje de metamodelado MOF (Meta Object Facility) [4], un meta-metamodelo que define una forma común de capturar todas las construcciones de los estándares de modelado e intercambio que son usadas en MDA.

El proceso de modernización incluye tres etapas: ingeniería inversa, reestructuración e ingeniería *forward*. La ingeniería inversa, el proceso de analizar los artefactos de software existentes para extraer información y proveer vistas de alto nivel del sistema, es una etapa crucial dentro de la modernización del sistema. En el contexto de ADM, la meta de la ingeniería inversa es descubrir el conocimiento del sistema existente y producir modelos en diferentes niveles de abstracción. Estos modelos serán el punto de partida para el proceso de reestructuración y la posterior generación del nuevo sistema.

En este artículo nos enfocamos en la primera de las etapas, describimos un *framework* para la ingeniería inversa en el contexto de ADM y mostramos cómo recuperar modelos que representan una visión abstracta del sistema a partir del código, en particular se analiza la extracción de diagramas de casos de uso y de interacción a partir de código Java. Esta propuesta se validó utilizando el *framework* de modelado Eclipse (Eclipse Modeling Framework - EMF), tecnología central dentro de Eclipse para el desarrollo dirigido por modelos [5]. Además, se utilizó el lenguaje ATL (Atlas Transformation Language) [6] como lenguaje de transformación de modelos y MoDisco para descubrir modelos intermedios [7]. Cabe destacar, que hasta el momento MoDisco sólo provee soporte para recuperar diagramas de clases a partir de código Java, por lo tanto, nuestro trabajo puede considerarse una contribución a la comunidad MoDisco.

Este artículo está organizado como sigue. La sección 2 describe los estándares ADM utilizados en nuestra propuesta y el proyecto MoDisco. La sección 3 presenta trabajos relacionados. La sección 4 describe un *framework* para la ingeniería inversa en el contexto de ADM. La sección 5 presenta un caso de estudio que muestra cómo recuperar diagramas de casos de uso y de secuencia a partir de código Java. La sección 6 discute las conclusiones y trabajos futuros.

2. Marco Teórico

Existe un creciente interés tanto en la migración de software como en la modernización de sistemas heredados con el objetivo de extender su vida útil. El éxito de la modernización de sistemas depende de la existencia de *frameworks* técnicos que permitan la integración de información y la interoperabilidad entre diferentes herramientas.

ADMTF ha establecido un conjunto de estándares (metamodelos) para la modernización con el objetivo de alcanzar interfaces y formatos bien definidos para el intercambio de información sobre modelos de software facilitando la interoperabilidad entre las herramientas y los servicios de modernización a los adherentes de los estándares. Esto da lugar a una nueva generación de soluciones que beneficiará a la industria del software fomentando la colaboración entre distintos proveedores. Entre los estándares desarrollados, cabe mencionar a los metamodelos KDM y ASTM [8] [9]. KDM es la base para la modernización de software y representa no sólo al código sino al sistema completo. ASTM es la especificación de los elementos de modelado para expresar árboles de sintaxis abstracta (Abstract Syntax Tree - AST). KDM y ASTM son dos especificaciones de modelado complementarias. KDM establece una especificación que permite representar información semántica sobre el sistema de software, mientras que ASTM establece una especificación para representar la sintaxis del código fuente por medio de AST. ASTM actúa como la base de más bajo nivel para el modelado de software en el ecosistema de estándares de OMG, mientras que KDM puede considerarse como una puerta de entrada a los modelos OMG de más alto nivel.

El surgimiento de ADM trae aparejada la necesidad de desarrollar nuevas herramientas que suporten modelado, interoperabilidad y estandarización, transformación automatizada tanto para ingeniería forward como para inversa, acceso a la definición de estas transformaciones y soporte para trazabilidad. Actualmente, la tecnología más completa que soporta ADM es MoDisco, un componente Eclipse para la ingeniería inversa dirigida por modelos. Hay distintas formas de extraer modelos a partir de los sistemas heredados debido a su heterogeneidad. Por esta razón, MoDisco ofrece un *framework* extensible y genérico para facilitar el desarrollo de herramientas que permitan extraer modelos a partir de código o sistemas heredados. El proyecto MoDisco está trabajando en estrecha vinculación con ADMTF. MoDisco está organizado en tres niveles para facilitar la reutilización de componentes entre distintas soluciones de modernización. El nivel de Infraestructura contiene componentes genéricos independientes de cualquier tecnología heredada tales como las implementaciones de los metamodelos ASTM y KDM. El nivel de Tecnología contiene componentes dedicados a una tecnología específica tal como el metamodelo del lenguaje Java. El nivel de Casos de Uso contiene componentes que proveen soluciones para casos de modernización específicos.

MoDisco se ha utilizado en aplicaciones reales, en [10] se describen trabajos de investigación e industrialización realizados utilizando MoDisco en los últimos años ilustrando su uso concreto con escenarios de ingeniería inversa industriales diferentes. Sin embargo, los autores sostienen que se debería mejorar el desempeño de la herramienta cuando se trabaja con sistemas grandes.

3. Trabajos Relacionados

Numerosos trabajos han contribuido a la ingeniería inversa de código orientado a objetos. En [11], los autores presentan una visión general de las técnicas que han sido

recientemente investigadas y aplicadas en el campo de la ingeniería inversa de código orientado a objetos. Los autores describen los algoritmos utilizados para la recuperación de diagramas UML a partir de código, tales como diagramas de clases, de estado y de secuencia. Nuestra propuesta puede ser considerada como una formalización de los procesos de ingeniería inversa descriptos por Tonella y Potrich en [11] en términos de los estándares involucrados en ADM. En [12], se comparan trabajos existentes en el área de ingeniería inversa, discuten casos exitosos e importantes logros, y se provee una guía para posibles desarrollos futuros en el área.

Muchos trabajos están vinculados a la ingeniería inversa basada en MDD. En [13] se describe el uso de MDE como una propuesta eficiente, flexible y confiable para el proceso de migración de software. Este proceso, desarrollado en Sodifrance, incluye análisis automático de código existente, recuperación de modelos abstractos de alto nivel, transformación de modelo-a-modelo y generación de código. Sodifrance ha desarrollado un conjunto de herramientas para la manipulación de modelos llamada Model-In-Action (MIA), utilizada como la base para la automatización de la migración. En [14], los autores analizan las dificultades encontradas al usar soluciones existentes para la extracción de modelos en el contexto de la modernización de software y proponen un lenguaje llamado Gra2MoL (Grammar To Model Transformation Language). Gra2MoL ha sido usado para extraer modelos a partir de código Java y PL/SQL. En [15] se describe una propuesta de ingeniería inversa basada en MDA que integra técnicas basadas en teoría de compiladores, metamodelado y especificaciones formales. Los autores describen un proceso que combina análisis estático y dinámico para la generación de modelos MDA, muestran cómo los metamodelos MOF pueden ser usados para dirigir el proceso de recuperación de modelos y detallan cómo los metamodelos y transformaciones pueden integrarse con especificaciones formales de una manera interoperable.

Con el surgimiento de ADM se han desarrollado nuevas propuestas y herramientas. En [16], los autores presentan un proceso para extraer modelos que conforman al metamodelo KDM. Primero, extraen modelos que conforman al ASTM utilizando Gra2MoL. Luego, estos modelos son transformados en modelos KDM utilizando el lenguaje de transformación de modelo-a-modelo RubyTL, el proceso se ejemplifica con una migración de Struts a JSF. En esta propuesta no se recuperan modelos UML. En [17] se describe un método de ingeniería inversa dirigida por modelos basado en metamodelos y transformación de modelos y se ilustra la propuesta recuperando modelos UML a partir de sistemas COBOL. El método fue implementado en el módulo BLU AGE® Reverse. Los autores definen un lenguaje textual específico al dominio para transformar el código a modelos KDM que luego son transformados a modelos PIM expresados en UML. En [18], los autores proveen una visión general de ADM junto con un estudio de distintos casos sobre modernización que cubren una variedad de sectores industriales.

4. Un Framework para ADM

En [19], los autores distinguen tres etapas principales en el proceso de modernización: el descubrimiento del modelo, el entendimiento del modelo y la regeneración del código. La etapa del descubrimiento del modelo "descubre" un conjunto de modelos iniciales que representan al sistema heredado en el mismo nivel de abstracción. La etapa del entendimiento del modelo transforma los modelos iniciales en modelos a un nivel de abstracción más alto, los cuales son el punto de partida de la etapa de regeneración del código.

En este trabajo nos centramos en las dos primeras etapas, describimos cómo recuperar modelos que representan una vista abstracta del sistema existente a partir del código en el contexto de ADM. Describimos un *framework* para ingeniería inversa dirigida por la arquitectura con el objetivo de recuperar modelos a partir del código orientado a objetos (Figura 1). En el contexto de MDD, el proceso de ingeniería inversa extrae elementos a partir del sistema existente y los representa como modelos específicos a la plataforma (PSM) a partir de los cuales se extraen modelos independientes de la plataforma (PIM). En el contexto de ADM, KDM es utilizado como soporte para representar los PSM, utilizando árboles de sintaxis abstracta como una representación intermedia del sistema de software.

El nivel de modelos incluye modelos de código (Implementation Specific Model - ISM), modelos KDM (PSM) y modelos UML (PIM). Estos últimos proveen una representación uniforme del sistema en el contexto de ADM e incluyen diagramas de clases, de casos de uso, de actividad, de interacción y de estado.

El nivel de metamodelos incluye metamodelos definidos a través de MOF, siendo éste la base para la definición de las transformaciones a nivel de modelos. El nivel de metamodelos incluye a los siguientes estándares: ASTM que describe los modelos AST, KDM que describe familias de modelos PSM y UML que describe familias de PIM. En este nivel, cada transformación se especifica en términos de los metamodelos origen y destino. De esta manera, los modelos a nivel PIM se recuperan por la aplicación sucesiva de transformaciones de modelos a partir del código fuente.

El proceso de ingeniería inversa propuesto a nivel de metamodelo consiste de dos pasos fundamentales:

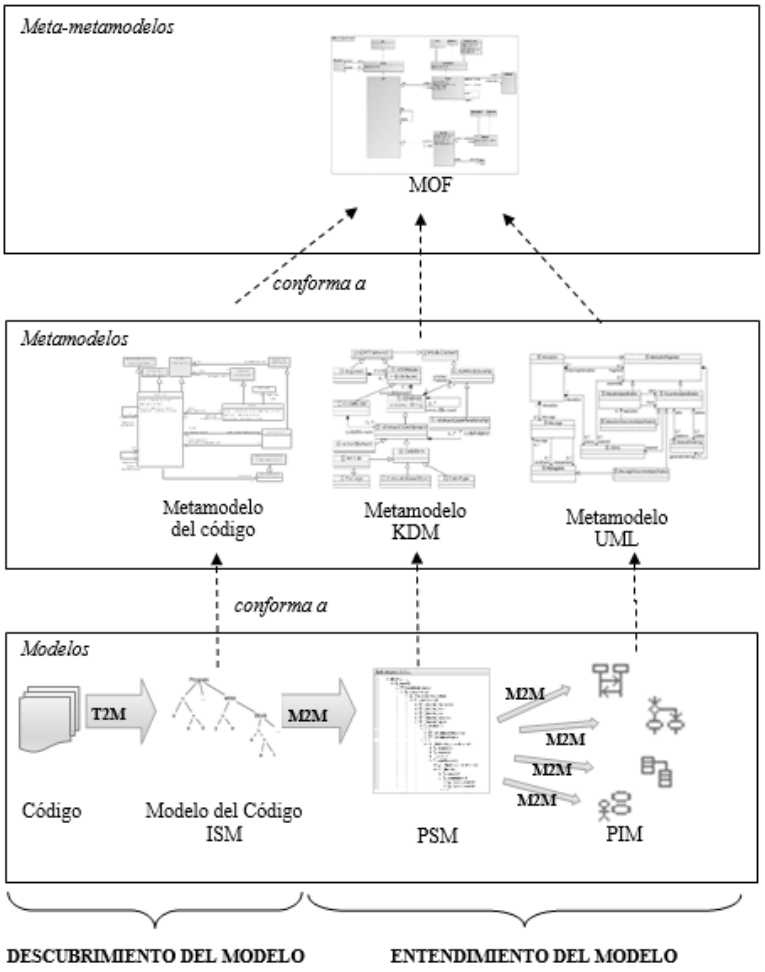


Figura 1. Framework para ingeniería inversa

1. Descubrimiento del modelo: el modelo del código se obtiene aplicando transformaciones texto-a-modelo (T2M) a partir del código fuente; el modelo obtenido, de bajo nivel de abstracción, es un árbol de sintaxis abstracta que conforma a ASTM.
2. Entendimiento del modelo: el objetivo es elevar el nivel de abstracción para obtener vistas de alto nivel de abstracción del sistema heredado que faciliten su análisis, entendimiento y futura regeneración. En esta etapa se obtienen modelos UML a través de transformaciones modelo-a-modelo (M2M) a partir de los modelos generados en la primera etapa. Este paso involucra dos transformaciones sucesivas:
   2.1. Transformación M2M para descubrir modelos KDM a partir del modelo del código.
   2.2. Transformación M2M para descubrir modelos UML a partir de modelos KDM.

Los modelos obtenidos en esta cadena de transformaciones son almacenados en el formato de intercambio XMI, el cual combina XML, MOF y UML permitiendo integrar herramientas, repositorios y aplicaciones en ambientes heterogéneos distribuidos [20].

En este trabajo se muestra cómo recuperar modelos PIM a partir de código orientado a objetos a nivel de transformaciones de metamodelos. Se analiza un proceso para recuperar diagramas de casos de uso y de interacción a partir de código Java y se lo especifica a través de transformaciones basadas en metamodelos. Para la implementación de las transformaciones modelo-a-modelo se optó por utilizar ATL debido a su grado de madurez. ATL es un lenguaje de transformación de modelos híbrido (permite escribir transformaciones declarativas y operacionales) y un conjunto de herramientas (de código abierto) creado por ATLAS Group que forma parte del proyecto GMT de Eclipse.

5. Caso de Estudio: Especificando Ingeniería Inversa de Código a Diagramas UML

Los diagramas UML no sólo son importantes para modelar el sistema durante la ingeniería *forward* sino también para entender el sistema, tanto su estructura como su conducta durante la etapa de ingeniería inversa. En esta sección, ejemplificamos el proceso de ingeniería inversa a nivel de metamodelos para recuperar diagramas de secuencia y diagramas de casos de uso a partir del código, utilizando el mismo caso de estudio empleado por Tonella y Potrich en [11], el programa Java *eLib* que soporta las principales funciones de una Biblioteca. El código Java (Figura 2) supone una colección de documentos de diferentes categorías (libros, revistas y reportes técnicos). Cada documento tiene un identificador. Los usuarios de la biblioteca pueden solicitar libros a préstamo. Mientras los libros pueden ser solicitados a préstamo por cualquier usuario, las revistas sólo pueden ser prestadas a usuarios internos y los reportes pueden ser consultados pero no prestados.

```
class Library {
  Map documents... Map users...Collection loans...
  public boolean addUser(User user) {...}
  private void addLoan(Loan loan) {...}
  public boolean borrowDocument(
                User user,Document doc)
   {if (user.numberOfLoans()<MAX_NUMBER_OF_LOANS&&
       doc.isAvailable()&&doc.authorizedLoan(user)){
          Loan loan = new Loan(user, doc);
          addLoan(loan); return true;}
    return false;
}
...}
class Document {
  int documentCode; Loan loan = null;...
  public boolean isAvailable(){return loan==null;}
 public boolean authorizedLoan(User u){
    return true;
}
...}
class Book extends Document {...}
class Journal extends Document {      ...
  public boolean authorizedLoan(User user) {
    return user.authorizedUser();
}
...}
class TechnicalReport extends Document{...
  public boolean authorizedLoan(User user){...}     ...}
class User {  int userCode; Collection loans ...
  public boolean authorizedUser(){return false;}
  public void addLoan(Loan loan){loans.add(loan);} ...}
class InternalUser extends User {     ...
   public boolean authorizedUser() {return true;}  ...}
class Loan {  User user; Document document;
  public Loan(User usr, Document doc) {...}
...}
```

Figura 2. Programa *eLib*

5.1. Descubriendo el Modelo

El primer paso del proceso de ingeniería inversa consiste en descubrir modelos a partir del código fuente del sistema existente. El código del programa *eLib* está escrito en el lenguaje Java, por lo tanto se utilizó el JavaAST Discoverer provisto por MoDisco para obtener el modelo AST correspondiente. Este Discoverer crea modelos Java a partir del código fuente contenido en el proyecto Java. Esta transformación se utilizó tal como la provee MoDisco sin ninguna modificación.

5.2. Entendiendo el Modelo

El segundo paso del proceso de ingeniería inversa consiste en obtener modelos de alto nivel de abstracción. Para lograr este objetivo, el modelo del código obtenido en la etapa previa es transformado en modelos PIM expresados en UML mediante la aplicación de transformaciones sucesivas que se describen en las siguientes subsecciones.

5.2.1. Recuperando el modelo KDM. La transformación aplicada en esta etapa, toma como entrada el modelo del código y devuelve un modelo instancia del metamodelo KDM. En la Figura 3 se muestra parcialmente este metamodelo.

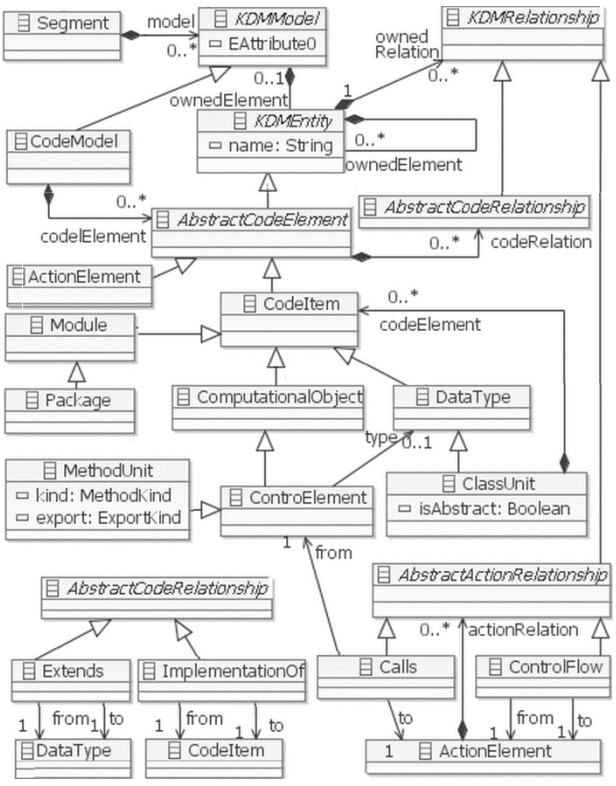


Figura 3. Metamodelo KDM

Las metaclases principales son *Segment*, *KDMModel*, *KDMEntity* y *KDMRelationship*. Un *Segment* contiene toda la información significativa sobre el sistema de software existente. Un segmento puede incluir instancias de *KDMModel* que representan vistas arquitecturales del sistema. Una instancia *KDMModel* posee entidades (instancias de *KDMEntity*) que representan artefactos del sistema de software tales como paquetes, clases y métodos. Una *KDMEntity* posee entidades y relaciones propias. Una relación, instancia de *KDMRelationship*, es una abstracción que especifica relaciones entre entidades. Cada instancia de *KDMRelationship*, tales como *Calls*, *Extends* e *ImplementationOf*, tiene exactamente un origen y un destino.

Para obtener el modelo KDM a partir del modelo del código generado en la primera etapa, se utilizó el Discoverer KDM provisto por MoDisco, pero con modificaciones. Este Discoverer está implementado como una transformación modelo-a-modelo en ATL, crea modelos que conforman al metamodelo KDM a partir del modelo del código Java. Esta transformación, tal cual la provee MoDisco, no está completamente especificada, existen elementos en el modelo del código Java que no son completamente transformados a sus respectivos elementos en el modelo KDM, lo que resulta en una pérdida de información, por ejemplo:

- En una expresión de invocación a método (*Calls*), los argumentos del método que son variables simples, tales como variables locales o parámetros, no están presentes en el modelo KDM.
- En una expresión de invocación a método, si el objeto sobre el cual se invoca el método es una variable simple, ésta no está presente en el correspondiente modelo KDM y por lo tanto falta la relación entre esta variable y la invocación al método.
- En expresiones infijas, los operandos que son variables simples no están presentes en el modelo KDM.

Para obtener toda la información necesaria que permitiera recuperar los diagramas UML (casos de uso e interacciones) a partir del modelo KDM obtenido en esta etapa, fue necesario enriquecer el Discoverer KDM provisto por MoDisco. Se anexaron nuevos *helpers* y se modificaron algunas de sus reglas. La nueva transformación se aplicó al modelo del código del programa *eLib* y se obtuvo como resultado el modelo KDM que se muestra parcialmente en la Figura 4.

El modelo KDM obtenido consiste de un *Segment* que posee tres modelos cada uno representando una vista arquitectural del sistema. El modelo *eLibrary* posee una instancia de *Package* llamada *LibraryPackage*. Éste contiene ocho instancias de *UnitClass* que representan a cada una de las clases definidas por el usuario en el programa *eLib* tales como *Library* y *Book*. La *ClassUnit Library* posee instancias de *StorableUnits* que representan a las variables miembros de la clase (*documents* y *users*) y *MethodUnits* que representan a los métodos miembros (por ejemplo, *borrowDocument*, *addUser* y *addLoan*). El método *borrowDocument* posee una *Signature* que representa la signatura del método y un *BlockUnit* que representa bloques de *ActionElements* (unidades básicas de comportamiento) relacionados tanto lógica como físicamente, por ejemplo, bloques de sentencia. El *BlockUnit* contiene *ActionElements* tales como sentencias *if*, invocaciones a métodos (*calls*) y sentencias *return*.

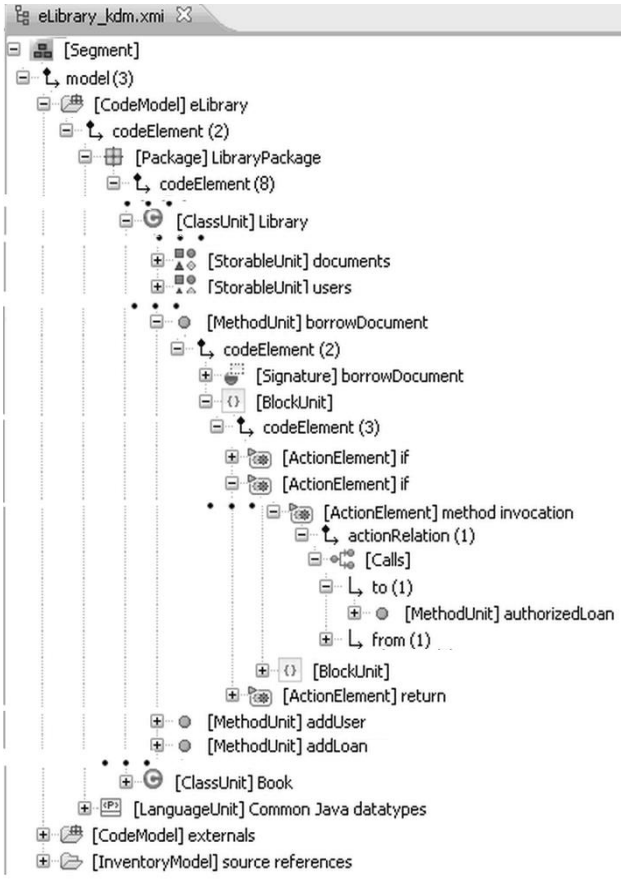


Figura 4. Modelo KDM del programa eLib

5.2.2. Recuperando modelos UML. El modelo KDM del programa *eLib* obtenido en la etapa previa es el punto de partida para recuperar los modelos PIM. Para esto, implementamos transformaciones modelo-a-modelo en ATL que generan modelos UML a partir de un modelo KDM. El metamodelo origen de cada transformación corresponde a KDM (Figura 3) y el metamodelo destino corresponde al metamodelo UML **[**21**]**. Este último se muestra parcialmente en la Figura 5, en particular, la parte del metamodelo UML que especifica los diagramas de interacción y los diagramas de casos de uso.

La Figura 5.a muestra el diagrama de interacciones: una *Interaction* posee líneas de vida (*lifeline*), mensajes y fragmentos. Una *lifeline* representa un participante individual en la interacción. Un mensaje define la comunicación entre participantes. Un fragmento puede ser una *Interaction*, una *ExecutionSpecification* u *OccurrenceSpecification*. Una *ExecutionSpecification* especifica la ejecución de una unidad de comportamiento o acción en la línea de vida, la duración de una *ExecutionSpecification* está representada por dos *ExecutionOccurrenceSpecifications*. Las *Occurrence-Specifications*, ordenadas a lo largo de una línea de vida, son unidades semánticas básicas de las interacciones. Las secuencias de ocurrencias especificadas por ellas dan el significado de las interacciones. Un *GeneralOrdering* representa una relación binaria entre dos especificaciones de ocurrencias la cual describe que una especificación de ocurrencia debe ocurrir antes que la otra en una secuencia válida.

La Figura 5.b muestra parcialmente el diagrama de casos de uso. Un *UseCase* es un clasificador con comportamiento que especifica algún comportamiento que un *subject* puede realizar en colaboración con uno o más actores. Un caso de uso puede estar relacionado con otros a través de relaciones de generalización, extensión o inclusión.

La Figura 6 muestra parcialmente las transformaciones *KDM2interaction* y *KDM2UseCases* que especifican cómo obtener a partir del modelo KDM diagramas de interacción y diagramas de casos de uso. A continuación se describen las transformaciones a través de sus reglas más importantes.

La transformación *KDM2interaction* (Figura 6.a) especifica la forma de producir diagramas de interacción a partir de modelos KDM:

- La regla *method2Interaction* transforma cada método público relevante (método complejo con un número significativo de llamadas a métodos) en una interacción. Su nombre se forma con el nombre del método precedido por el nombre de la clase donde el método fue declarado para facilitar su identificación. La primera *lifeline* de la interacción representa al objeto que es instancia de la clase donde el método fue declarado. El resto de las *lifelines* se obtienen de las variables locales y de los parámetros del método sobre los cuales un método es invocado. Los mensajes de la interacción se obtienen de las llamadas a métodos. El orden parcial entre los mensajes se establece a través de *generalOrderings* creados a partir de reglas *lazy*.
- La regla *methodInvocation2Message* transforma cada llamada a método, instancia de *Calls*, en un mensaje. Si el tipo del método invocado es 'constructor', el nombre del mensaje estará formado por el nombre del método precedido por la palabra 'create'. En el resto de los casos, el nombre del mensaje se forma con el nombre del método invocado precedido por el nombre de la clase donde el método fue declarado. Esta regla establece tanto el emisor como el receptor del mensaje (*sendEvent* y *receiveEvent*).
- La regla *CreateGeneralOrdering* crea una instancia de *GeneralOrdering* a partir de una llamada a un método, instancia de *Calls*. Su nombre se forma por el nombre de los mensajes vinculados separados por '->' indicando el orden entre los mismos. Esta regla establece instancias de *MessageOccurrence-Specification*, *before* y *after*, que permiten especificar el orden entre los mensajes.
- La regla *createLifeline* crea una *lifeline* a partir de la clase donde se declara el método que dio lugar a la interacción.

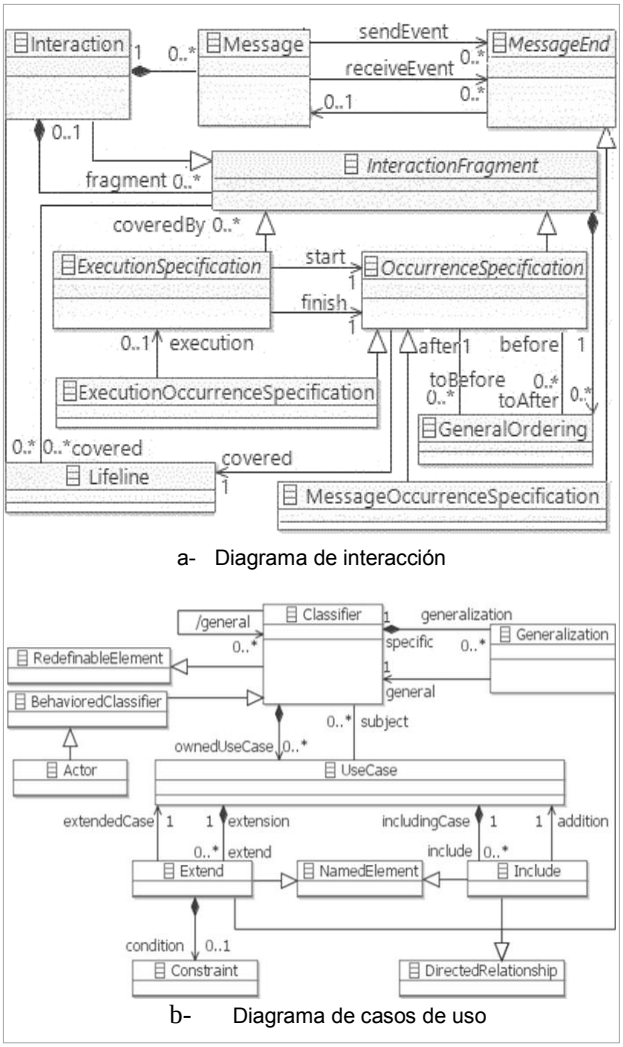


Figura 5. Metamodelo UML

```
module KDM2interaction;
create OUT : MM1 from IN : MM;
rule method2Interaction {
  from m:MM!MethodUnit (m.is_relevantPublicMethod())
  to interact:MM1!Interaction (
     name     <- m.get_ContainerClass().name + '::' + m.name
    ,lifeline <- thisModule.createLifeline(m)
    ,lifeline <- m.get_Variables()->collect(v|thisModule.object2Lifeline(v))
    ,message  <- m.get_calls()
    ,generalOrdering <- m.get_pairsOfCalls()->collect ( p | thisModule.CreateGeneralOrdering(p->at(1)))
    ...            )
}
rule methodInvocation2message {
  from call:MM!Calls (call.is_relevantCall())
  to msj:MM1!Message (
    name <- if (call.to.kind= #constructor)then call.get_ContainerClass().name +'::'+'create_' + call.to.name
                                          else call.get_ContainerClass().name + '::' + call.to.name endif
    ,argument <-call.get_Arguments()->collect(name|thisModule.createArgument(name))
    ,sendEvent<- send
    ,receiveEvent <- receive
    )
    ,send: MM1!MessageOccurrenceSpecification (...)
    ,receive: MM1!MessageOccurrenceSpecification(...)
}
unique lazy rule createLifeline {
  from m:MM!MethodUnit
  to obj:MM1!Lifeline (
    name <- m.get_ContainerClass().name.toLower()+': ' +m.get_ContainerClass().name
    ,coveredBy <- thisModule.createActionExecutionSpecification (m,m.get_ContainerClass().name.toLower()))
}
unique lazy rule object2Lifeline {
  from d:MM!DataElement to obj:MM1!Lifeline (     name <- d.name  + ': ' + d.type.name  )
}
unique lazy rule CreateGeneralOrdering {
  from call:M1!Calls   to    genOrdering: MM1!GeneralOrdering(
    name <- call.to.name + '->' + call.get_ContainerMethod().nextTo(call).to.name
    ,before <- ...         , after<- ...
}
unique lazy rule createActionExecutionSpecification {
  from m:MM!MethodUnit ,name:MM!StringType  to exeSpec:MM1!ActionExecutionSpecification (
    name <-   'controlFocus_of_'  + name
    ,start <- ...         , finish <- ...
    …)
}
```

Figura 6.a. Transformación *KDM2interaction*

```
module KDM2useCases;
create OUT : MM1 from IN : MM;
rule KDMPackage2UMLPackage{
from p:MM!Package
to UMLpackage:MM1!Package (
  name <-  p.name ,
  packagedElement <- p.get_MethodUnit()  )
}
rule MethodUnit2UseCase1{
  from method:MM!MethodUnit
     (method.kind = #method and method.export = #public and method.codeRelation.oclIsUndefined() )
  to useCase:MM1!UseCase (
    name <-  method.refImmediateComposite().name+ '_' + method.name       )
}
rule MethodUnit2UseCase2{
  from method:MM!MethodUnit
    (method.kind = #method and method.export = #public and  not method.codeRelation.oclIsUndefined())
  to useCase:MM1!UseCase (
    name <-  method.refImmediateComposite().name + '_' + method.name,
    general <- method.codeRelation ->collect(relation | relation.to )     )
}
rule Calls2Dependency{
  from call:MM!Calls ( call.to.is_publicMethod() and call.in_publicMethod() )
  to use_case_dependency :MM1!Dependency (
    name <- 'from_' + call.get_container().name + '_to_' + call.to.name,
    supplier <- call.to  ,
    client <- call.get_container()       )
}
```

Figura 6.b. Transformación *KDM2useCases*

Figura 6 – Transformaciones KDM a UML

- La regla *object2Lifeline* crea una *lifeline* a partir de una variable.
- La regla *createActionExecutionSpecification* crea un foco de control, instancia de *ActionExecution-Specification*, a partir de un método y un nombre de *lifeline*. El comienzo y la finalización (*start* y *finish*) del foco de control, instancias de *Execution-OccurrenceSpecification*, se crean a partir de reglas *lazy*.

La Figura 6.b muestra parcialmente la transformación *KDM2UseCases* que especifica la forma de producir diagramas de casos de uso a partir del modelo KDM:

- La regla *KDMPackage2UMLPackage* transforma cada paquete KDM en un paquete UML, cuyo nombre es igual al nombre del paquete KDM y que contendrá los casos de uso (elementos empaquetados) que son creados a partir de métodos en las reglas correspondientes.
- La regla *MethodUnit2UseCase1* transforma cada método público que no redefine métodos heredados en un caso de uso básico que sólo puede tener relaciones de dependencia. Para facilitar su identificación, el nombre del caso de uso se forma con el nombre del método precedido del nombre de la clase que lo contiene.
- La regla *MethodUnit2UseCase2* transforma cada método público que redefine métodos heredados en un caso de uso básico que tiene relaciones de generalización y puede tener relaciones de dependencias. El nombre del caso de uso se forma con el nombre del método precedido del nombre de la clase que lo contiene.
- La regla *Calls2Dependency* transforma cada relación *Calls* que vincula dos métodos públicos en una dependencia entre los casos de uso creados a partir de esos métodos.

Las Figuras 7 y 8 muestran parcialmente los modelos resultantes de las transformaciones, cuando son aplicadas al modelo KDM correspondiente al programa *eLib*.

La Figura 7 muestra en particular el modelo de interacción que representa el intercambio de mensajes entre los objetos desencadenado por la ejecución del método *borrowDocument* de la clase *Library*. En la figura se observa que la interacción está compuesta de *lifelines*, fragmentos, mensajes y el orden entre los mismos. La Figura 7.a muestra el modelo de interacción en formato XMI, mientras que la Figura 7.b muestra el diagrama de secuencia correspondiente.

La Figura 8.a muestra el modelo de casos de uso generado a partir del programa *eLib* en formato XMI, mientras que la Figura 8.b muestra el diagrama UML correspondiente. Se visualiza en particular el caso de uso *Library_borrowDocument* junto con los casos de uso que éste incluye.

6. Conclusiones y Futuros Trabajos

La ingeniería inversa es una etapa crucial dentro del proceso de modernización. Su rol es analizar el sistema de software existente para entender y extraer modelos de alto nivel de abstracción. En este artículo presentamos un *framework* para ingeniería inversa en el contexto de ADM que distingue niveles de abstracción vinculados a modelos y metamodelos centrándonos en este último nivel.

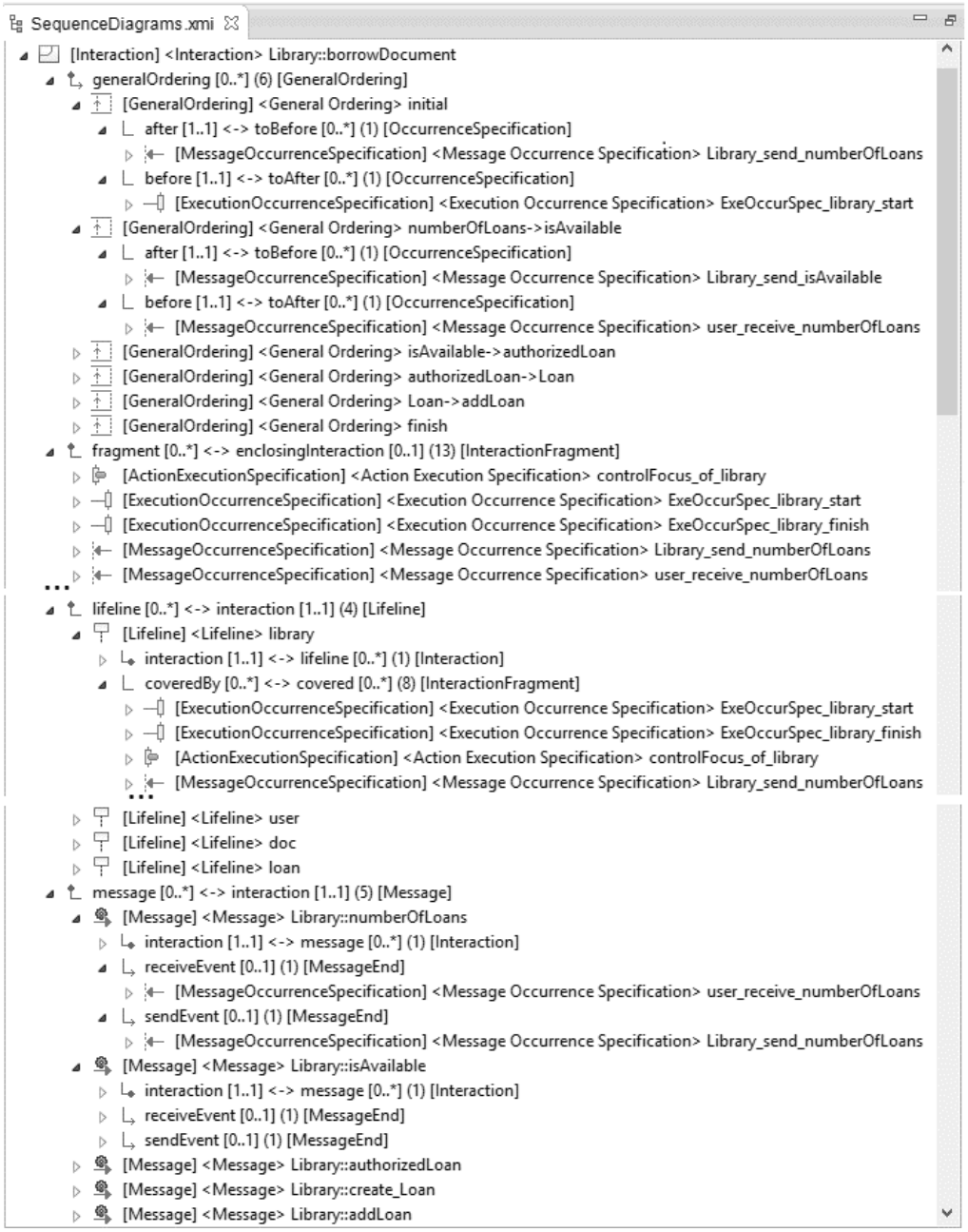
Proponemos una técnica de metamodelado basada en metamodelos MOF y transformaciones. Para conformar a ADM utilizamos estándares tales como ASTM y KDM ambos definidos vía MOF. El lenguaje seleccionado para implementar las transformaciones es ATL debido a su grado de madurez. La propuesta se validó usando Eclipse Modeling Framework y la plataforma MoDisco. Si bien en este trabajo ejemplificamos nuestra propuesta a través de un caso de estudio de ingeniería inversa a partir de código Java, cabe mencionar que se puede aplicar a cualquier lenguaje orientado a objetos.

Consideramos que nuestra propuesta provee beneficios con respecto a los procesos basados solamente en técnicas de ingeniería inversa tradicionales. La ingeniería inversa dirigida por modelos permite obtener modelos que representan el sistema heredado, donde cada modelo se centra en un aspecto diferente del sistema en distintos niveles de abstracción bajo un *framework* común y bajo los mismos estándares. Esto permite un mejor entendimiento del sistema y facilita la interoperabilidad entre las herramientas y servicios de los usuarios de los estándares, alcanzando interfaces y formatos bien definidos para el intercambio de información sobre modelos de software usados por las herramientas de modernización de software.

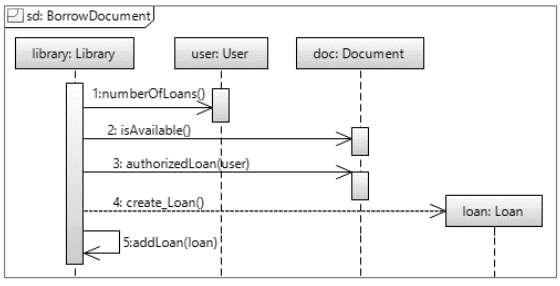
Los resultados obtenidos pueden ser considerados como una contribución a la comunidad MoDisco. Por un lado, proveemos soporte para recuperar diagramas de casos de uso y de interacción complementando el soporte para recuperar diagramas de clases provisto por MoDisco. Por otra parte, extendimos el Discoverer KDM de MoDisco para obtener la información necesaria para recuperar los diagramas de interacción.

Actualmente estamos trabajando, dentro del framework propuesto, en la recuperación de otros diagramas UML a partir de código Java y su integración con el EMF.

Se prevé especificar el metamodelo para el lenguaje de programación C++ y definir *discoverers* para recuperar modelos del código C++ (instancias de ASTM) y modelos KDM, que nos permitan recuperar modelos UML a partir del sistemas heredados escritos en C++.



a. SequenceDiagrams.xmi

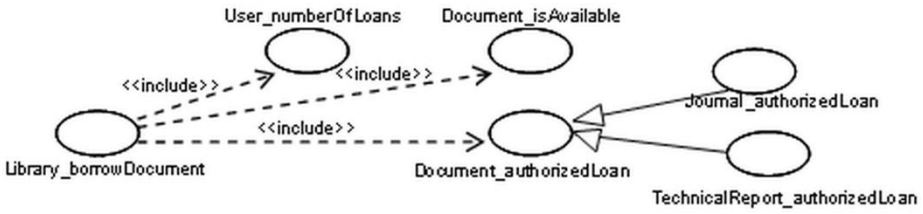


b. diagrama UML

Figura 7. Modelo de Interacción



a. UseCases.xmi



b. Diagrama UML

Figura 8. Modelo de casos de uso

7. Referencias

[1] L. Favre, *Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution*, IGI Global, 2010.

[2] "Architecture-Driven Modernization (ADM) Task Force," 2014. [Online]. Disponible en: http://www.omgwiki.org/admtf/doku.php.

[3] "MDA - The Model-Driven Architecture," 2014. [Online]. Disponible en: http://www.omg.org/mda/.

[4] "Meta Object Facility (MOF) Core Specification," OMG Document Number: formal/2013-02-01, 2013. [Online]. Disponible en: http://www.omg.org/spec/MOF/2.4.1.

[5] "Eclipse Modeling Framework," 2014. [Online]. Disponible en: http://www.eclipse.org/modeling/emf/.

[6] "Atlas Transformation Language (ATL) Documentation," 2014. [Online]. Disponible en: http://www.eclipse.org/atl/documentation/.

[7] "MoDisco - Model Discovery," 2014. [Online]. Disponible en: http://www.eclipse.org/MoDisco.

[8] "KDM - Knowledge Discovery Meta-Model," OMG Document Number: formal/2011-08-04, 2014. [Online]. Disponible en: http://www.omg.org/spec/KDM/1.3.

[9] "ASTM - Abstract Syntax Tree Metamodel," OMG Document Number: formal/2011-01-05, 2011. [Online]. Disponible en: http://www.omg.org/spec/ASTM.

[10] H. Bruneliere, J. Cabot, G. Dupé and F. Madiot, "MoDisco: a Model Driven Reverse Engineering Framework," *Information and Software Technology*, vol. 56, no. 8, pág. 1012-1032, 2014.

[11] P. Tonella and A. Potrich, *Reverse Engineering of Object-Oriented Code* (Monographs in Computer Science), Springer-Verlag, 2005.

[12] G. Canfora and M. Di Penta, "New Frontiers of reverse Engineering," en *Future of Software Engineering (FOSE '07)*, 2007.

[13] F. Fleurey, B. E., B. Baudry, A. Nicolas and J. Jézéquel, "Model-Driven Engineering for Software Migration in a Large Industrial Context," *MoDELS 2007- Lecture Notes in Computer Science*, vol. 4735, pág. 482-497, 2007.

[14] J. Cánovas Izquierdo and J. García Molina, "A domain specific language for extracting models in software modernization," *Model Driven Architecture - Foundations and Applications. Lecture Notes in Computer Science*, vol. 5562, pág. 82-97, 2009.

[15] L. Favre, C. Pereira and L. Martinez, "Foundations for MDA CASE Tools," en *Encyclopedia of Information Science and Technology, Second Edition*, M. Khosrow-Pour, Ed., Hershey, PA, IGI Global, 2009, pág. 159-166.

[16] J. Cánovas Izquierdo and J. García Molina, "Extracción de modelos en una modernización basada en ADM," en *Actas de los Talleres de las Jornadas de Ingeniería de Software y BBDD*, 2009.

[17] F. Barbier, G. Deltombe, O. Parisy and K. Youbi, "Model Driven Engineering: Increasing Legacy Technology Independence," en *Second India Workshop on Reverse Engineering (IWRE, 2011) en The 4th India Software Engineering Conference*, Thiruvanantpuram, India, 2011.

[18] W. Ulrich y P. Newcomb, *Information System Transformation. Architecture Driven Modernization Case Studies*, MK/OMG Press, Elsevier, 2010.

[19] M. Brambilla, J. Cabot y M. Wimmer, *Model-Driven Software Engineering in Practice*, Morgan & Claypool, 2012.

[20] "XML Metadata Interchange (XMI) Specification. OMG Document Number: formal/2014-04-04," 2014. [Online]. Disponible en: http://www.omg.org/spec/XMI/2.4.2/PDF.

[21] "Unified Modeling Language (UML) Superstructure," OMG Specification: formal/2011-08-06, 2011. [Online]. Disponible en: http://www.omg.org/spec/UML/2.4.1/.

Una Ontología para la Interoperabilidad de Modelos de Simulación

Mónica del Carmen Gil
Facultad de Ciencias Físico–Matemáticas y Naturales - Universidad Nacional de San Luis
Ejército de los Andes 950 - San Luis
Argentina
sistemas97@hotmail.com

Germán A. Montejano
Facultad de Ciencias Físico–Matemáticas y Naturales - Universidad Nacional de San Luis
Ejército de los Andes 950 - San Luis
Argentina
gmonte@unsl.edu.ar

Mario Berón
Facultad de Ciencias Físico–Matemáticas y Naturales - Universidad Nacional de San Luis
Ejército de los Andes 950 - San Luis
Argentina
mberon@unsl.edu.ar

Resumen

A pesar de que los modelos de simulación han sido usados con éxito, para entender diversos procesos, como así también para asistir la formulación y evaluación de diferentes sistemas, existen pocos esfuerzos para formalizar, sistematizar y comunicar métodos para su descripción en la etapa conceptual. La mayoría de los modelos no están bien especificados y documentados desde un nivel inicial, lo que conduce a problemas de interoperabilidad. Esto lleva a que se deban replicar tareas o bien no se pueda aprovechar toda la potencialidad que brindaría el uso del modelo de simulación más apropiado para cierto tipo de realidad con datos propios del modelo usado más aquellos que han sido utilizados/generados por otros modelos.
Con el propósito de resolver el problema descripto en el párrafo precedente, en este artículo se presenta una aproximación basada en el estándar LCIM ("Levels of Conceptual Interoperability Model") [1,2,3,4] *y ontologías para facilitar la interoperabilidad de Modelos de Simulación. Teniendo en cuenta lo antes mencionado, se toma como foco de estudio el Dominio Hidrológico y se define una Ontología que proporciona un lenguaje común para compartir y recuperar datos a partir de la descripción de los conceptos y términos que intervienen en los procesos de gestión hidrológica. La ontología definida es validada con un set de encuestas, las cuales se distribuyen entre los expertos en el dominio que se propone estudiar.*

1. Introducción

La simulación de sistemas, y en específico la simulación basada en modelos computacionales, supone el estudio de conceptos sobre las ciencias sociales, matemáticas y de la computación, y busca la creación de modelos simplificados de la realidad estudiada.

La simulación de sistemas implica la construcción de modelos. El objetivo es averiguar qué pasaría en el sistema si acontecieran determinadas hipótesis, para ello se utilizan los modelos de simulación que son un importante medio para aumentar la eficiencia de la investigación. Esto se debe a que éstos auxilian a los investigadores en la asimilación del conocimiento adquirido mediante la experimentación.

Un modelo es un subconjunto de la realidad; cuando se construye el modelo parte del mundo real y sus relaciones quedan excluidas.

A pesar de que los modelos de simulación han sido usados con éxito, para entender diversos procesos sociales, como así para asistir la formulación y evaluación de políticas en diferentes sistemas, existen pocos esfuerzos para formalizar, sistematizar y comunicar métodos para su descripción en la etapa conceptual.

Las investigaciones realizadas hasta el momento [5,6,7,8], permiten comprobar que no existen metodologías ni conceptualizaciones para unificar criterios de especificaciones en las primeras etapas de diseño de cualquier sistema, lo que conduce a problemas de interoperabilidad al no modelar las relaciones que son necesarias para asegurarla. La mayoría de los modelos no están bien especificados y documentados desde un nivel inicial.

No existe un método que permita integrar y compartir los datos entre distintos modelos de simulación. Esto trae como consecuencia el desarrollo de soluciones ad-hoc, lo que ocasiona gastos y esfuerzos humanos innecesarios.

Es posible resolver este problema a través de un vocabulario común para todos los modelos de simulación. Esto permite que las tareas realizadas en un modelo puedan ser interpretadas por otro modelo diferente.

Durante el Panel de discusión SCS-SISO (*Simulation Interoperability Standards Organization*) [9] / **SCS** (*Society for Modeling and Simulation International*) sobre las prioridades para estándares de modelos y

simulación, Seigler explicitó en su presentación que para asegurar la interoperabilidad entre sistemas la estandarización debe ser alcanzada en el nivel de modelado, por ej: el nivel de estandarización debe ser más alto que los estándares del nivel de programación aplicados actualmente. Para una "significativa interoperabilidad", compartir los datos estandarizados vía protocolo estandarizado, tales como el protocolo DIS *(Distributed Interactive Simulation)* o el HLA *(High-Level Architecture)*, son necesarios pero no suficientes [10].

La coordinación de los modelos conceptuales subyacentes, la armonización de las ideas de operación simuladas es el *quid de la cuestión* para crear soluciones interoperables. En lugar de estandarizar solamente los requerimientos de intercambio de información también se debe coordinar la cadena causa efecto en el modelo subyacente.

La única instancia del proceso de desarrollo del modelo y simulación que lleva a asegurar esto es el modelo conceptual. El modelo conceptual describe qué parte del mundo real es modelada y bajo qué restricciones, y algunas veces cuál es más importante y qué parte no será modelada.

Los sistemas de información hidrológica requieren interoperabilidad (capacidad para compartir datos y procesos) porque contienen grandes cantidades de información que se debe complementar para realizar procesos de análisis, predicción y estudios socioeconómicos, entre otros. Pese a existir estándares para el desarrollo de Sistemas de Agua, la interoperabilidad entre sistemas ya desarrollados es un problema, ya que la estructura de datos y procesos es propia de cada sistema y la gran cantidad de datos dificulta su migración a las estructuras estándares. Por ello, en este trabajo de investigación científica y tecnológica se elabora un caso de estudio que permita comprender los elementos subyacentes a la interoperabilidad entre Sistemas. Se desarrolla un vocabulario común para todos los modelos de simulación del dominio del caso de estudio. Esto permite que las tareas realizadas en un modelo puedan ser interpretadas por otro modelo diferente.

Para lograr una interoperabilidad significativa entre los sistemas de simulación se aplicará métodos de ingeniería que contribuyan a las soluciones independientes y combinables. En otras palabras, estas soluciones se pueden usar como entrada de otro modelo y satisfacer necesidades específicas de cada usuario. Para llevar a cabo esta tarea se basa en el estándar LCIM *("Levels of Conceptual Interoperability Model")*[11], hasta poder cumplir principalmente con lo requerido en el Nivel Conceptual para aplicaciones de modelos y simulación. De una forma más sintética se puede decir que el fin es *"establecer una Ontología para definir datos y facilitar a que los modelos sean interoperables".*

2. Modelos y Simulación

La simulación es la construcción de modelos informáticos que describen la parte esencial del comportamiento de un sistema de interés, así como diseñar y realizar experimentos con el modelo y extraer conclusiones de sus resultados para apoyar la toma de decisiones [12,13].

La simulación se usa como un paradigma, como un marco de trabajo válido, para analizar sistemas complejos [14,15]. La idea es obtener una representación simplificada de algún aspecto de interés de la realidad e imita la operación del sistema actual sobre el tiempo. La simulación permite experimentar con sistemas, reales o propuestos, en casos en los que de otra manera esto sería imposible o impráctico.

2.2. Modelado de un Sistema

En algún momento de la vida de la mayoría de los sistemas, hay una necesidad de estudiar su comportamiento para entender las relaciones entre varios componentes, o para predecir el funcionamiento bajo la consideración de nuevas condiciones.

El uso de modelos, a veces llamado "modelación", es un instrumento muy común en el estudio de sistemas de toda índole. El empleo de modelos facilita el estudio de los sistemas, aun cuando éstos puedan contener muchos componentes y mostrar numerosas interacciones como puede ocurrir si se trata de conjuntos bastante complejos y de gran tamaño. El trabajo de modelación constituye una actividad técnica como cualquiera otra, y dicha labor puede ser sencilla o compleja según el tipo de problema específico que deba analizarse.

Un sistema puede ser estudiado directamente mediante un experimento o con un modelo que lo representa [16,17]. Cuando la primera alternativa no es conveniente, por razones de costo o incluso porque el sistema a estudiar podría aún no existir, es usualmente necesario construir un modelo del sistema. El modelo es una "imitación" del sistema original. Como para poder imitar algo o a alguien es necesario conocerlo bien, será necesario reunir la información precisa respecto del sistema original. En el modelo participan las variables y sus relaciones.

Un modelo es una abstracción de la realidad que captura lo esencial para investigar y experimentar en lugar de hacerlo con el sistema real, con menor riesgo, tiempo y costo.[18]

3. Interoperabilidad

La globalización ofrece la posibilidad de sumar esfuerzos para la integración de los procesos y datos desarrollados y obtenidos aisladamente. El problema principal para intentar esta integración es el problema de compatibilidad en la estructura de los datos, las entradas y salidas de los programas [19].

Para enfrentar este problema es necesario avanzar en el tema de la interoperabilidad, que se define como la capacidad que exhiben los sistemas para compartir datos y procesos [20, 21,22].

Se entiende por interoperabilidad a la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos [23,24,25]. Así, la interoperabilidad debe asegurar que los datos se introducen correctamente en una única aplicación, y automáticamente éstos se propagan al resto de las aplicaciones del sistema. Un sistema interoperable debe tener reglas definidas, de tal manera, que controle las interacciones del usuario e intercambie datos entre aplicaciones de forma efectiva.

Es la capacidad de transferir y utilizar información de una manera uniforme y eficiente a través de múltiples organizaciones y sistemas. Hay varios niveles de interoperabilidad entre los sistemas totalmente interoperables y los no interoperables.

Una de las definiciones más citadas[1], y que define la interoperabilidad de la información a un alto nivel conceptual, es la que propuso la IEEE[2] en 1990: *"la habilidad de dos o más sistemas, redes de comunicación, aplicaciones o componentes para intercambiar información entre ellos y para usar la información que ha sido intercambiada*".

La interoperabilidad de sistemas no es una función cortada por el mismo molde. Hay varios niveles de interoperabilidad entre los sistemas totalmente interoperables y los no interoperables.

En el dominio técnico varios modelos de interoperabilidad ya existen y son usados satisfactoriamente para determinar el grado de interoperabilidad entre sistemas de tecnología de la información. Sin embargo, algunos modelos no han sido todavía establecidos en el dominio del modelado conceptual [26]. Por lo que se pretende construir un puente entre el diseño conceptual y el diseño técnico para la implementación e integración de sistemas.

Este diseño es además un marco de trabajo, un framework [27], para determinar en una etapa temprana del desarrollo de procesos, si es posible, la interoperabilidad entre sistemas.

Para este fin, el objetivo del modelo, va más allá del nivel de implementación de los estándares actuales, habla de definir bien los datos para que a partir de estos se puedan desarrollar nuevos sistemas de otros usuarios.

4. Ontologías

La creciente automatización de procesos, en general, ha provocado una proliferación de sistemas de información con semánticas que se solapan [28]. Por otro lado, existe la necesidad de reutilización de conocimiento, para hacer más eficiente esta automatización. Los desarrollos de sistemas basados en conocimiento y de software en general, se hacen en diferentes contextos, puntos de vista y suposiciones acerca de su materia de estudio. Cada uno usa su propio vocabulario, pueden tener distintos términos para designar un mismo concepto, y diferentes métodos y estructuras. Esto ha provocado problemas de:

- Comunicación por falta de entendimiento compartido y el mismo limitado.
- Interoperabilidad para reutilizar y compartir información.
- Confiabilidad.
- Especificación.

Una ontología especifica una conceptualización, una forma de ver al mundo [29], por lo que cada ontología incorpora un punto de vista. Una ontología contiene definiciones que proveen del vocabulario necesario para referirse a un dominio. Las definiciones dependen del lenguaje que se usa para describirlas [30,31].

Las ontologías se usan para favorecer la comunicación entre personas, organizaciones y aplicaciones, lograr la interoperabilidad entre sistemas informáticos, razonar automáticamente como así también para la ingeniería de software. El objetivo es describir un dominio de conocimiento general, que sirva de base para modelar un caso de estudio, especialmente a un nivel conceptual el modelo es dependiente del contenido.

Las ontologías son definiciones que especifican un vocabulario relativo a un cierto dominio. Este vocabulario define entidades, clases, propiedades, predicados, funciones y, las relaciones entre estos componentes. Las ontologías toman un papel clave en la

[1] EICTA. EICTA White Paper on Interoperability and Standardisation. Bruselas, 2006. 26 p. Disponible en: http://www.eicta.org/fileadmin/user_upload/document/document1166544474.pdf. [Consulta: 29/02/2008].

[2] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York: IEEE, 1990. [Consulta: 29/02/2008].

resolución de interoperabilidad semántica entre sistemas de información y su uso dentro del contexto [32,33].

En términos prácticos, el desarrollo de una ontología incluye:

- Definición de conceptos o clases que son las ideas a formalizar.
- Relaciones que representan las interacciones entre las clases.
- Instancias que son los objetos de una clase.

5. Definición de la Ontología en el Dominio Hidrogeológico con Methontology

Los sistemas de información geográfica (SIG) se están convirtiendo en importantes sistemas de gestión de información geo-referenciada en un amplio dominio de aplicación, especialmente en las ciencias de la tierra, la atmósfera y el océano. Esta gestión comprende procesos de análisis, predicción, estudios socioeconómicos y ecológicos, entre otros, para lo que resulta importante la posibilidad de inter-operar entre ellos [34].

Hoy en día, el empleo de modelos hidrológicos es indispensable, ya que pueden ser utilizados para analizar diferentes aspectos del agua como por ejemplo: los escurrimientos, las inundaciones, la operación de embalses, así como los sistemas de abastecimiento, los usos del agua, su aprovechamiento y protección del agua subterránea [35].

Con la llegada de los Sistemas de Información Geográfica y su aplicación en la modelación hidrológica, el hidrólogo encontró una forma de automatizar y hacer más eficiente los análisis que llevaba a cabo. Para ello surgieron distintos simuladores de flujos de agua como ser: el Hydrus (simula el flujo unidimensional, transporte de un sólo soluto y movimiento de calor, en un medio variablemente saturado), Feflow (modelación avanzada tridimensional de elemento finito para flujo de aguas subterráneas, calor y transporte de contaminantes)[36], Visual Help (estándar internacional para la modelación de hidrología de rellenos sanitarios y para la estimación de las tasas de recarga de aguas subterráneas)[37], Visual Modflow (estándar de la industria en modelación tridimensional de flujo de aguas subterráneas, transporte de contaminantes y de calor)[38] pero ninguno de estos simuladores pueden integrarse entre sí por no existir un Modelo Conceptual Compartido en cada uno de esos programas, es decir los resultados que brinda un simulador no pueden ser utilizados por otro [39].

Teniendo en cuenta lo importante que es el uso de estos simuladores para el dominio hidrológico se definió una ontología para la interoperabilidad de modelos y simulación con los conceptos del dominio hidrológico.
Los pasos para llevar a cabo la aplicación mencionada en el párrafo precedente son: i) estudiar el dominio, ii) definir el modelo conceptual y iii) especificar su ontología.

En las siguientes secciones del presente trabajo se describe como se define la ontología en el dominio hidrológico. Para ello se explican las tareas que se usaron en la aplicación de la metodología Methontology. Asimismo, se muestra como se confeccionó el set de encuestas, es decir, como está formado y como se validaron las mismas distribuyéndolas a los expertos del dominio. Se muestran los resultados obtenidos usando tablas y gráficos. Y por último se proporcionan algunas conclusiones al trabajo presentado.

5.1. Aplicación de Methontology para la Definición de la Ontología

Methontology propone un ciclo de vida basado en la evolución de prototipos que permite añadir, cambiar y eliminar términos en cada nueva versión de la ontología.

La actividad de especificación permite determinar por qué se construye la ontología, cuál será su uso, y quiénes serán sus usuarios finales. Las tareas que se llevan a cabo son el Análisis de los dominios y la Captura de requerimientos - reuniones con expertos.

En el caso de este trabajo se define como dominio la hidrología, para evaluar el balance hídrico abstrayendo los conceptos de evaporación y evapotranspiración. Particularmente, a modo de ejemplo, se enfoca al análisis de las relaciones entre una Cuenca Hidrográfica y las Curvas de Nivel.

La actividad de conceptualización se encarga de organizar y convertir una percepción informal del dominio en una especificación semi-formal, para lo cual utiliza un conjunto de representaciones intermedias, basadas en notaciones tabulares y gráficas, que pueden ser fácilmente comprendidas por los expertos de dominio y los desarrolladores de ontologías. Se define el Modelo Conceptual representado con un Diagrama de Clases. Siendo la actividad de formalización la que se encarga de la transformación de dicho modelo conceptual en un modelo formal o semi-formal computable.Una vez que se define la ontología se confecciona un set de encuestas al que se le realizará chequeos de consistencia sometiéndolo a la validación de su contenido por expertos en el dominio.

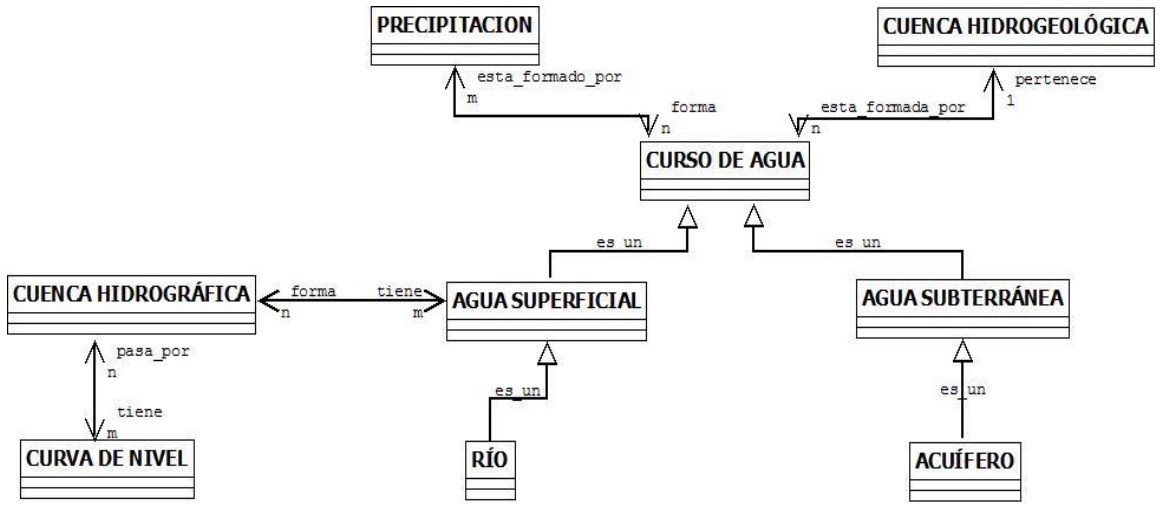


Figura 1 - Diagrama de Conceptos y Relaciones

El objetivo de esta encuesta es verificar si los conceptos, atributos y relaciones definidos para el dominio hidrológico son acordados por los expertos en el dominio. Como así también evaluar si éstos son considerados válidos y suficientes como datos comunes y más usados para ser utilizados en cualquier aplicación relacionada al Balance Hídrico abstrayéndose la evapotranspiración y la evaporación.

El set de encuestas está integrado por un Diagrama de Conceptos y Relaciones y por un conjunto de Tablas de Conceptos y Atributos.

En la Figura 1 se muestra el Diagrama de Conceptos y Relaciones que forman parte de la ontología. Los conceptos fueron identificados como los más representativos del dominio y utilizados por aplicaciones hidrológicas. En el diagrama de la Figura 1 los conceptos pasan a constituir las Clases del Diagrama formando así el Diagrama de Clases. Más adelante a cada clase se la describe con sus atributos, unidades de medida y valores que pueden tomar.

A continuación, existe un conjunto de tablas donde se encuentra una definición asociada a cada uno de los conceptos considerados para la ontología mostrada en el Diagrama de la Figura 1. El experto deberá validar cada concepto como verdadero, en caso que la definición sea correcta y la más usada o falso en caso contrario como se muestra en un ejemplo de la Tabla 1.

Asimismo, por cada concepto se encuentran definidos

Tabla 1 – Definición del Concepto

CONCEPTO	VALIDACION
1) Agua Superficial Es aquella que circula sobre la superficie del suelo. Se produce por la escorrentía generada a partir de las precipitaciones o por el afloramiento de aguas subterráneas. Se presentan en forma correntosa, como ser corrientes, ríos y arroyos; o quietas como ser lagos, reservorios, embalses, lagunas, humedales, estuarios, océanos y mares.	V ☐ F ☐

Tabla 2 – Atributos del Concepto y sus definiciones

VALIDACION	ATRIBUTOS DE AGUA SUPERFICIAL	UNIDAD DE MEDIDA	VALOR
■	**Precipitación:** es la fuente de agua. Cuantifica el drenaje que infiltra, el escurrimiento y la evapotranspiración.	mm^3	-
■	**Tipo:** se refiere al comportamiento que tiene el agua en su cauce o reservorio. Se clasifican en:	-	-
	Correntosa: corriente natural de agua que fluye con continuidad. Posee un caudal determinado, rara vez es constante a lo largo del año y en otros casos el caudal es escaso. Ej: ríos y arroyos.	-	*Sí/No*
	Quieta: acumulación de agua separada del mar producida por una obstrucción en el lecho de un río. Tranquilo, sosegado. Ej: lagos, reservorios, embalses, lagunas, humedales, estuarios, océanos y mares.	-	*Sí/No*
■	**Calidad Química:** se refiere a la cantidad de los distintos parámetros que debe tener el agua para garantizar la preservación de la fauna y flora, dependiendo de su uso. Es decir si se trata de agua para consumo humano (agua potable), para uso agrícola o industrial, para recreación, para mantener la calidad ambiental, etc.	-	-
■	**Caudal:** equivale a la cantidad de agua por unidad de tiempo.	$\frac{m^3}{s}$ *Dónde* m^3 *es volumen* s *es tiempo*	-
■	**Influjo de agua sub:** es el agua superficial que infiltra para recargar el agua subterránea.	mm^3	-

Observaciones…….....

.....…….....

cada uno de los atributos asociados al mismo. El experto deberá validar con un tilde si están correctos y son suficientes para desarrollar aplicaciones en el dominio hidrológico referidas al Balance Hídrico. En caso de diferir en alguno de ellos se le solicita completar el apartado de observaciones que se encuentra al pie de cada tabla. Para ejemplificar en este artículo se colocan algunos atributos del concepto "Agua Superficial" como se puede observar en la Tabla 2.

En la Tabla 3 se encuentran las relaciones definidas entre los conceptos identificados donde el experto coloca si existe o no dicha relación.

Se puede observar que existen cuatro relaciones llamadas "es_un", las mismas representan herencia. Por ejemplo: la relación Rio es_un Agua Superficial, indica que la clase <Río> tiene sus propios atributos y todos los atributos que hereda de la clase <Agua Superficial>. Lo mismo sucede con la clase <Acuífero> con respecto a la clase <Agua Subterránea>, <Agua Superficial> con

Tabla 3 – Relaciones entre Conceptos

RELACION	SI	NO	OBSERVACIONES
¿El **Agua Superficial** es_un **Curso de Agua**?			
¿El **Agua Subterránea** es_un **Curso de Agua**?			
¿El **Curso de Agua** está formado por **Precipitación**?			
¿La **Precipitación** forma **Curso de Agua?**			
¿El **Curso de Agua** pertenece a **Cuenca Hidrogeológica**?			
¿La **Cuenca Hidrogeológica** está formada por **Curso de Agua**?			
¿La **Cuenca Hidrográfica** tiene **Curva de Nivel**?			
¿La **Curva de Nivel** pertenece a **Cuenca Hidrográfica**?			
¿La **Cuenca Hidrográfica** tiene **Agua Superficial**?			

respecto a <Curso de Agua> y <Agua Subterránea> con respecto a <Curso de Agua>.

Por otro lado un <Curso de Agua> está formado por <Precipitación> y su inversa es <Precipitación> forma un <Curso de Agua>. Asimismo la <Cuenca Hidrogeológica> está formada por <Curso de Agua> y su inversa es <Curso de Agua> pertenece a una <Cuenca Hidrogeológica>.

A partir de las observaciones que hagan los expertos se determinará la cardinalidad de cada una de las relaciones.

6. Resultados - Validación de la Ontología

Como se ha venido describiendo en los puntos anteriores, para definir una Ontología se necesita de la participación comprometida de expertos y de opiniones consensuadas respecto a los conceptos. En consecuencia, para cumplir con dicha tarea, el set de encuestas, que define la ontología para este trabajo, fue distribuido a expertos del dominio hidrogeológico, entre los cuales se encuentran:

Licenciados en Geología

- Jefe Área Hidrología – Programa Infraestructura Hídrica (Gobierno de la Prov. De San Luis)
- Profesional Área Hidrología – Programa Infraestrcutura Hídrica (Gobierno de la Prov. De San Luis)
- Prof. Adjunto cátedra de Hidrogeología de la UNSL
- Jefe Trabajo Práctico de la Catedra de Geología de la UNSL
- Investigador de la UNSJ – Dr. En Geología. Esp. Glaciares.

Ingenieros Agrónomos

- Director EEA INTA
- Gerente Técnico de San Luis Agua
- Investigador INTA Laboratorio de Recursos Naturales, Medio Ambiente y Geomántica.
- Docente de Hidrogeografía en Profesorado de Geografía Instituto Superior Nuestra Señora del Carmen
- Investigador del CONICET
- Profesional Investigador en INTA EEA San Luis

Licenciada en Bioquímica – Química – Profesora de enseñanza media y superior en Química

- Jefe de área Dpto. Técnico Programa Infraestructura Hídrica- Ministerio de Hacienda

Tabla 4 – Atributos observados

Conceptos	Atributos	Observados
CURSO DE AGUA	4	
PRECIPITACION	10	2
CUENCA HIDROGEOLOGICA	5	
AGUA SUPERFICIAL	5	1
AGUA SUBTERRANEA	3	
CUENCA HIDROGRAFICA	18	
CURVA DE NIVEL	2	
RIO	15	1
ACUIFERO	12	2
TOTAL ATRIBUTOS	**74**	**6**

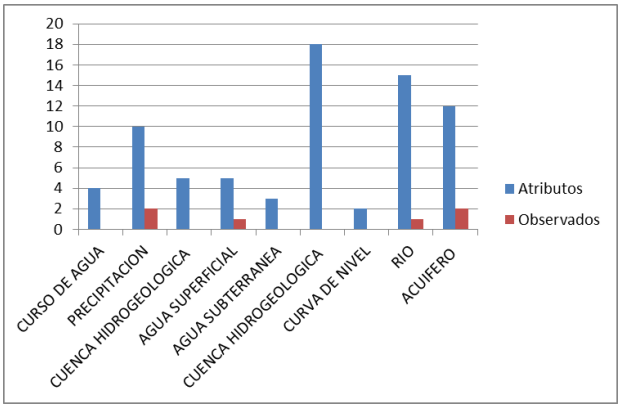


Figura 2 – Atributos Observados

y Obras Públicas- Gobierno de la Provincia de San Luis.

Licenciada en Geografía

- Profesora de Geografía en la Carrera de Turismo de la UCC.

El set de encuesta se entregó a 30 expertos del dominio hidrogeológico, hasta el momento fue validado por 13 expertos. A continuación se muestran los resultados.

Teniendo en cuenta que la población de estudio está formada hasta el momento por 13 expertos. El 100% está de acuerdo con el diagrama de clases definido y la definición de cada uno de los conceptos considerados para este trabajo.

La ontología está formada por 9 conceptos y 74 atributos en total.

La Tabla 4 y el gráfico de la Figura 2 representan que de 13 expertos encuestados hasta el momento, el 92% estuvo de acuerdo en los atributos considerados con sus definiciones correspondientes y solo el 8% acotó observaciones a la definición de algún atributo como se muestra en la Tabla 4.

Como se puede observar la Tabla 5- Discrepancias en definiciones de algunos atributos y la Tabla 6 - Relaciones observadas por algunos expertos, presentan ciertas discrepancias en algunas definiciones de relaciones y atributos debido a la formación de los expertos encuestados. Esto se ve principalmente entre los profesionales de agronomía y geología.

Asimismo, de 13 expertos del dominio hidrológico encuestados; el 85% estuvo de acuerdo con las relaciones definidas y el 15% observó 2 de las 10 relaciones establecidas.

Las discrepancias resultarán útiles para definir la cardinalidad de las relaciones cuando en trabajos futuros se formalice, transformando el modelo conceptual en un modelo formal, y así completar las etapas de la metodología Methontology. Por ejemplo la relación - Agua Superficial **es_un** Curso de Agua - tendrá una cardinalidad [1..n].

7. Conclusiones y Trabajos Futuros

El uso de ontologías se ha incrementado en varias áreas de la Informática y, hay también en la actualidad, un interés creciente sobre su uso en el diseño y desarrollo de los Sistemas de Información.

En este artículo se mostraron algunos de los usos actuales de las ontologías en los sistemas de información, como soporte para el análisis conceptual de distintos dominios que permite, entre otros, formular o reformular conceptos y sus relaciones para la modelación.

La ontología desarrollada aporta a la comunidad interesada, para su utilización y expansión, conocimiento bien estructurado, estandarizado y formalizado, adquirido de un grupo de expertos de distintas organizaciones avocadas a la temática hidrológica.

La ontología creada permite compartir información específica del área y hacer este conocimiento reutilizable.

El método METHONTOLOGY resulto de gran utilidad para la construcción de la ontología en el dominio hidrológico, ya que permitió estructurar el conocimiento adquirido mediante diferentes técnicas, a través de un conjunto de tablas y diagramas que pueden ser entendidos por los expertos del dominio.

Tabla 5 – Discrepancias en definiciones de algunos atributos.

CONCEPTO	ATRIBUTO	OBSERVACIONES	Definición original
PRECIPITACIONES	ISOHIETA	Son las líneas que unen puntos de igual cantidad de **precipitaciones**	…..igual cantidad de lluvia
	FRECUENCIA	Lo que se definió como frecuencia se denomina recurrencia (o período de retorno como se denominó al final), es decir cada cuanto ocurre un evento. La frecuencia indica el número de eventos en un período de tiempo dado.	es el tiempo en años en que una lluvia de cierta intensidad y duración se repite con las mismas características.
AGUA SUPERFICIAL	CALIDAD QUIMICA	En rigor deberíamos hablar de propiedades físico-químicas del agua y de ahí podríamos definir calidad en función al uso consuntivo y no consultivo.	….se refiere a la cantidad de los distintos parámetros que debe tener el agua para garantizar la preservación de la fauna y flora, dependiendo de su uso.
ACUIFERO	CALIDAD DE AGUA	Es el conjunto de propiedades físico-químicas del agua.	...se refiere a su composición química.
	PERMEABILIDAD	La permeabilidad es la capacidad del sustrato de conducir agua y se mide unidades de lámina (altura de agua) por unidad de tiempo (mm/h, cm/h, cm/d, mm/s, etc)	...Es el flujo de agua que atraviesa una sección unitaria de acuífero. Depende de la porosidad y el grado de intercomunicación entre los granos.
RIO	PENDIENTE	la pendiente tiene dos componentes rumbo e inclinación. La definición dada solo abarca rumbo	...es la dirección en la que se va a dirigir el río desde las mayores alturas hacia las menores.
	PENDIENTE	La pendiente es la inclinación del terreno, (por ejemplo mts de desnivel (vertical) en una distancia en metros dada (horizontal)	

Además, permite la actualización de términos a medida que se avanza en la actividad de conceptualización, lo que evidencia su flexibilidad.

El modelo conceptual, para el caso de estudio presentado en este artículo, fue sometido al análisis por parte de expertos en el dominio hidrológico. Esto posibilitó obtener conceptos consensuados por todos, lo que trae aparejado la posibilidad de diseñar y desarrollar modelos interoperables en el dominio hidrológico, ya que todos sus posibles usuarios estarán hablando en el mismo idioma.

Asimismo, un trabajo a futuro puede consistir en extender la aplicación de la metodología usada en esta tesis para el desarrollo de ontologías en distintos dominios y así lograr la interoperabilidad en los modelos de simulación.

Por último, hay que añadir que toda la experiencia ganada en la resolución de este problema, podrá ser extendida al análisis de las distintas aplicaciones de los simuladores mencionados en este trabajo, entiéndase Modflow, Hydrus, etc, para diseñar un esquema conceptual que los integre y poder así definir una ontología que facilite la interoperabilidad entre los mismos, lo que implicaría una nueva implementación de todos ellos.

Tabla 6 – Relaciones observadas por algunos expertos

RELACION	OBSERVACIONES
¿El **Agua Superficial** es_un **Curso de Agua**?	
	El concepto de curso de agua tiene una concepción más topográfica que hidrológica en ese sentido se interpreta como el cauce por donde escurre agua de manera permanente o temporal, en hidrología se habla de escurrimientos.
	No solamente, también hay lagunas que no corren
¿El **Agua Subterránea** es_un **Curso de Agua**?	
	Considerando el concepto de curso de agua como cauce lo incluimos dentro del agua superficial
	No solamente, hay agua subterránea que no se mueve. Y mucha se mueve tan lentamente que puede ser considerada estática para muchos problemas

8. Referencias

[1] Andreas Tolk, Ph.D., James A. Muguira, *"The Levels of Conceptual Interoperability Model"*, Virginia Modeling Analysis and Simulation Center Old Dominion University, Tolk/Muguira 2003.

[2] Tolk, A. and Muguira, J.A. (2003). The Levels of Conceptual Interoperability Model (LCIM). Proceedings IEEE Fall Simulation Interoperability Workshop, IEEE CS Press.

[3] Turnitsa, C.D. (2005). Extending the Levels of Conceptual Interoperability Model. Proceedings IEEE Summer Computer Simulation Conference, IEEE CS Press.

[4] Lisa L. Brownsword, David J. Carney, David Fisher, Grace Lewis, Craig Meyers, Edwin J. Morris, Patrick R. H. Place, James Smith, Lutz Wrage; *"Current Perspectives on Interoperability",* TECHNICAL REPORT, CMU/SEI-2004-TR-009, ESC-TR-2004-009, March 2004.

[5] F. Javier Sanchez San Román, *"Introducción a Visual Modflow"*, Dpto. Geología Universidad de Salamanca(España).

[6] Enric Peig Olivé; *"Interoperabilidad de Metadatos en Sistemas Distribuidos"*; Universitat Pompeu Fabra, Octubre 2003.

[7]Mark Kasunic, William Anderson; "*Measuring Systems Interoperability: Challenges and Opportunities"* Software Engineering Measurement and Analysis Initiative, CMU/SEI-2004-TN-003, Abril 2004.

[8] José Aurelio Martínez García; *"Simulación de link 16 sobre HLA"*; Universidad Europea de Madrid, Escuela superior politécnica -Área de telecomunicación, Curso 2006-2007

[9] SISO/SCS Panel Discussion: *"Priorities for M&S Standards,"* Spring Simulation Interoperability Workshop, Orlando, Florida, March 2003.

[10]IEEE Standard Group 1516: *"High Level Architecture"* (Revision 2000), IEEE CS Press.

[11] M. D. Petty and E. W. Weisel , Proceedings of the Spring 2003 Simulation Interoperability Workshop, "A Composability Lexicon", Orlando FL, March 30-April 4 2003, 03S-SIW-023.

[12] Ríos Insua, D., Ríos Insua, S., Martín, J. , Jiménez, A. (2008) Simulación: Métodos y Aplicaciones, RA-MA, Segunda Edición.

[13] Law, A. y Kelton, W. (1991) Simulation Modeling and Analysis, Academic Press.

[14] Antonio Caselles Moncho, La simulación se usa como un paradigma, Universitat de Valencia, 2008.

[15] Inmaculada Garcia, Ramón Mollá Vayá, Computer Graphics Section, Department of Computation and Computer Systems, Technical University of Valencia, Valencia E-46022, Spain.

[16] "Simulation Modeling and Analysis", Averil M. Law y W. David Kelton, Ed. Mc. Graw-Hill, (1991).

[17] "Computer Simulation and Modelling", Francis Neelankaveel, John Wiley & Sons, (1986).

[18] Deudes Meza; Oscar Vivas; Ronny Boneth, Modelos Concretos y Modelos Abstractos, 2013.

[19] Miller, Paul. "Interoperability: What is it and Why should I want it?" Ariadne, v. 24, Junio 2000. Disponible en:http://www.ariadne.ac.uk/issue24/interoperability/[Consulta: 29/02/2008].

[20] M. D. Petty and E. W. Weisel , Proceedings of the Spring 2003 Simulation Interoperability Workshop, "A Composability Lexicon", Orlando FL, March 30-April 4 2003, 03S-SIW-023.

[21] Mark Kasunic William Anderson, Software Engineering Measurement and Analysis Initiative, "Measuring Systems Interoperability: Challenges and Opportunities", April 2004, Technical Note CMU/SEI-2004-TN-003.

[22] Robert Richards, Ph.D., Chairman, Richard H. Stottler, Brian Spaulding, Final Report for the SISO Study Group, "Intelligent Tutoring System (ITS) Interoperability", April 2005.

[23] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.

[24] Bases para una estrategia iberoamericana de interoperabilidad. Documento marco iberoamericano de interoperabilidad ratificado en 2010 por la XX Cumbre Iberoamericana de Jefes de Estado y de Gobierno.

[25]Gascó, Mila; Jiménez, Carlos E. (2011). *Interoperability in the justice field: variables that affect implementation*. 11th European Conference on eGovernment. Slovenia.

[26] Antonio Jiménez Muñoz, "Interoperabilidad entre Plataformas", Proyecto SUMA elearning multimodal y adaptativo para la empresa, 2007.

[27] Moreno, H., S. Silveira-Netto y H. Sin (2007), *"Conceptualización de arquitectura de gobierno electrónico y plataforma de interoperabilidad para América Latina y el Caribe"*, serie Sociedad de la información (LC/W.140), Comisión Económica para América Latina y el Caribe (CEPAL), Santiago de Chile, julio.

[28] Monserrat Robles Viejo, Carlos Angulo Fernandez (2010), "Búsqueda Semántica en Repositorios de Conceptos Biomédicos Estandarizados", Universidad Politécnica de Valencia.

[29] Silvia Esther Sánchez López, "Modelo de indexación de formas en sistemas VIR basado en ontologías", Universidad de las Américas Puebla, México 2007. http://catarina.udlap.mx/u_dl_a/tales/documentos/mcc/sanchez_l_se/capitulo4.pdf.

[30] Gordijn, J. (2004). "e-Business Model Ontologies. Book chapter contribution to 'e-Business Modelling Using the e3value Ontology'". Wendy Curry (ed.), pp. 98-128. Elsevier Butterworth-Heinemann.

[31] Gruber, T.R. (1993). "A Translation Approach to Portable Ontologies". Knowledge Acquisition, 5(2):199-220.

[32] Guarino, N. (1998). "Formal Ontologies and Information Systems". En N. Guarino (ed.): 3-15.

[33] Mahesh, K. (1996). "Ontology Development for Machine Translation: Ideology and Methodology". Technical Report MCCS-96-292. NMSU. Computing Research Laboratory. New Mexico.

[34] Carlos Mario Zapata Jaramillo, Guillermo González Calderón, María Isabel Marín Morales, *"Un caso de estudio en interoperabilidad entre sistemas de información geográfica"*, Universidad de Medellín, 2009.

[35] M. en C. Víctor Hugo Guerra Cobián, *"Análisis del efecto de discretización espacial en el modelado de cuencas hidrológicas utilizando el modelo distribuido CEQUEAU-ONU"*, Toluca México, Septiembre de 2007.

[36]http://www.feflow.com/uploads/media/users_manual 62.pd.

[37]http://www.aquagri.com/pdf/Visual_Help_demo.pdf

[38]http://hidrologia.usal.es/Complementos/manual_mod flow.pdf

[39] F. Javier Sanchez San Román, *"Introducción a Visual Modflow"*, Dpto. Geología Universidad de Salamanca (España).

Detección de Conceptos en Programas Escritos con Lenguajes Multiparadigma

José L. Albanés
Departamento de Informática
Universidad Nacional de San Luis
San Luis - Argentina
Email: jlalbanes@gmail.com

Mario M. Berón
Departamento de Informática
Universidad Nacional de San Luis
San Luis - Argentina
Email: mberon@unsl.edu.ar

Resumen

La comprensión de programas, es una disciplina de la ingeniería de software cuyo, propósito es facilitar al programador la tarea de entender programas. El objetivo principal de esta disciplina, es elaborar métodos, técnicas, herramientas que faciliten la recuperación y la consecuente visualización de la relación (denominada DP-DP) existente entre los dominios del problema (la salida del sistema) y programa (las componentes de software usadas para producir la salida), facilitando de esta manera la comprensión.

Este trabajo de investigación tiene como objetivo la definición, y aplicación de la estrategia de Análisis Comportamental (AC) a sistemas de software; dicha estrategia es útil para vincular los dominios mencionados en el párrafo precedente. Para lograr su propósito, AC requiere de información dinámica y estática. La primera se obtiene mediante técnicas tales como la aplicación de un esquema de instrumentación de código, debugging, profiling, etc. Para recuperar el segundo tipo de información (estática) se aplican, técnicas de compilación tradicionales.

Basándose en los tipos de información antes mencionados, la experiencia del ingeniero de software y en un proceso iterativo, AC recupera la relación DP-DP, simplificando de esta forma el proceso de comprensión.

1. Introducción

La etapa de mantenimiento o evolución de un producto de software dentro del ciclo de vida de sistemas, presenta complejidades de diferentes índoles, abarcando posibles problemas de diseño, según los cambios en los requerimientos del usuario, análisis de la usabilidad del software y/o adición de nuevas funcionalidades por parte de los usuarios, corrección de errores de programación, etc.

Estas necesidades de mantenimiento, provocan un incremento significativo en horas de trabajo para el arquitecto o grupo de arquitectos de software, que desarrollan el proyecto; conllevando esto a elevar los costos del mismo.

Por lo mencionado en los párrafos anteriores, muchos son los esfuerzos de la ingeniería de software, para lograr encontrar soluciones a los problemas que se presentan en el desarrollo de software. Así es como han surgido diferentes paradigmas de programación, para resolver la lógica de negocio de un sistema; aportando cada uno de estos sus ventajas y/o mejoras para intentar resolver de la mejor manera los problemas de implementación, y hacer del desarrollo y mantenimiento de software una tarea más simple.

Debido a que cada paradigma de programación no logra resolver todos los problemas de la manera más eficiente; se hace cada vez más común encontrar sistemas desarrollados con implementaciones de múltiples paradigmas o multiparadigma [1], lo que complica aún más la tarea de mantenimiento y del proceso cognitivo que esto implica para el desarrollador que las lleva a cabo.

Por lo expresado en el párrafo anterior es que surge la necesidad de elaborar estrategias aplicadas al análisis de este tipo de sistemas, que permitan facilitar las tareas de comprensión, inspección y visualización de los artefactos de software que intervienen en la resolución de la lógica de negocio que aplica el sistema bajo análisis.

Este trabajo de investigación tiene por objetivo definir y aplicar una estrategia de extracción de información llamada análisis comportamental a sistemas multiparadigmas, de forma tal de mostrar la relación existente entre el dominio del problema y el dominio del programa.

Según la disciplina de comprensión de programas, lograr la relación DP-DP [2], resulta ser el mejor camino para obtener conocimiento sobre la salida del sistema y los componentes de software utilizados para lograr dicha salida; contribuyendo esto, a mejorar los aspectos cognitivos al programador que lleva a cabo tareas de mantenimiento, evolución o migración del sistema bajo estudio.

Este artículo está organizado como se indica a continuación. En la sección 2 se explica la disciplina de comprensión de programas (CP), ésta es un área fundamental para la aplicación de la etapa de mantenimiento del software. También se define una de las posibles estrategias para vincular el dominio del problema con el dominio del programa del sistema analizado, llamada estrategia de análisis comportamental (AC). Luego en la sección 3 se muestra la aplicación de la estrategia de AC a software desarrollado con múltiples paradigmas. El caso de estudio tomado como ejemplo, es un software editor de textos llamado EdiPy el cual está desarrollado con el lenguaje de programación Python, que permite utilizar funcionalidades de múltiples paradigmas en la programación de sistemas. Finalmente, en la sección 4 se detallan las conclusiones del trabajo expuesto e investigaciones futuras.

2. Comprensión de Programas

La necesidad de agilizar la tarea de mantenimiento de software y mejorar la comprensión del código del sistema bajo estudio; ha llevado al desarrollo de disciplinas para tal fin, la comprensión de programas es una de esas disciplinas [2,3,4,5,6].

La comprensión de programas se traduce en la habilidad de entender una pieza de código escrito en un lenguaje de alto nivel. Un programa no es más que una secuencia de instrucciones que serán ejecutadas de forma tal que se garantiza una determinada funcionalidad. El lector de un programa consigue extraer el significado del mismo cuando comprende de qué forma el código cumple con la tarea para la cual fue creado [7].

El área de comprensión de programas es una de las más importantes de la Ingeniería del software, porque es necesaria para tareas de reutilización, inspección, manutención, migración y extensión de sistemas de software [8,9,10]. Puede también ser utilizada en áreas como ingeniería reversa o en la enseñanza de lenguajes de programación. La tarea de comprensión de programas puede tener diferentes significados y puede ser vista desde diferentes perspectivas.

El usuario puede estar interesado en como la computadora ejecuta las instrucciones con el objetivo de comprender el flujo de control y de datos, o puede querer verificar los efectos que la ejecución tiene sobre el objeto que está siendo controlado por el programa. Considerando estos niveles de abstracción, el desarrollo de herramientas versátiles de inspección visual de código es crucial en la tarea de comprensión de programas.

La Comprensión de Programas es una disciplina que tiene como objetivo, desarrollar Modelos, Métodos, Técnicas y Herramientas basados en un proceso cognitivo y de ingeniería con el propósito de facilitar el entendimiento del software a los desarrolladores o arquitectos de software. Existen muchos productos destinados a facilitar la comprensión de software [11]. Sin embargo, en muchas situaciones, no es claro como las teorías cognitivas, estrategias de visualización y extracción de la información se instancian en dichos productos [9, 12, 13].

Actualmente se han realizado muchos trabajos que intentan facilitar la Comprensión de Programas [5]. Esos trabajos presentan interesantes estrategias que exploran y visualizan el Domino del Programa dejando al programador la tarea de interpretar lo que se lleva a cabo en el Dominio del Problema. También se pudo comprobar que son escasas las estrategias que intentan vincular (aunque sea débilmente) ambos dominios [14, 15].

Por lo expuesto en el párrafo anterior se puede afirmar que el principal desafío en esta área consiste en Relacionar correctamente el Dominio del Problema con el Dominio del Programa. En otras palabras lo que se pretende relacionar es la salida del sistema (Dominio del Problema) con las componentes de software utilizadas para producir dicha salida (Dominio del Programa).

Una forma de concretar la relación mencionada previamente, consiste en reconstruir la relación real entre el Dominio del Problema y el Dominio del Programa, por medio de una relación virtual. Para alcanzar este objetivo se debe: i) Construir una representación del Dominio del Problema; ii) Construir una representación del Dominio del Programa y iii) Definir un Procedimiento de Vinculación. La aproximación descripta anteriormente se puede conceptualizar como un meta modelo de comprensión el cual puede ser instanciado con modelos particulares de comprensión, los cuales a su vez pueden ser utilizados como base para la creación de estrategias de comprensión. Esta conceptualización es la utilizada para el estudio y creación de estrategias de interconexión de dominios.

2.1. Trabajos Relacionados. Estrategias Para Relacionar el Dominio de Problema y Programa.

A través del estudio del estado del arte se ha podido detectar la existencia de algunas estrategias que permiten relacionar el Dominio del Problema con el Dominio del Programa [16]. Entre esas estrategias se destacan:

- SVS (Simultaneous Visualization Strategy): Permite que las componentes de software usadas en tiempo de ejecución sean visualizadas mientras el sistema se ejecuta. Una debilidad de esta técnica es la imposibilidad de obtener explicaciones del comportamiento del sistema, debido a la forma en la que se representa la información dinámica.

- BORS (Behavioral-Operational Relation Strategy): Requiere que el sistema sea ejecutado y que cierta clase de información dinámica, como por ejemplo las funciones usadas durante la ejecución del sistema, sea recuperada. Después de eso dicha información es procesada por algoritmos que construyen explicaciones relacionadas con la ejecución del sistema. En esta técnica existe una fase de inspección manual que recupera nombres de funciones relacionadas con un determinado concepto del dominio del problema lo que introduce complejidades en el proceso de comprensión. Es importante notar que dicho proceso de inspección se realiza utilizando la técnica del grep. Esta técnica consiste en buscar palabras claves, por concordancia sintáctica, en el código fuente del programa, utilizando el comando grep provisto por los sistemas operativos Linux, Unix, etc.

- SVSi (Simultaneous Visualization Strategy Improved): Esta estrategia es una combinación de SVS y BORS. A través de SVS se puede observar la salida del sistema y las funciones del programa usadas para producir dicha salida. Dichas funciones, son almacenadas en un archivo y luego interpretadas por los algoritmos utilizados por BORS para construir explicaciones. Es importante notar que SVSi elimina la necesidad de inspeccionar manualmente el Dominio del Problema, ya que permite visualizar la salida del sistema de estudio sincronizada con la lista de funciones llamadas.

A pesar de los resultados obtenidos, en cuanto a la interconexión de los Dominios del Problema con el Dominio del Programa se refiere, se piensa que es posible encontrar relaciones más fuertes que las actuales. Con esto se hace referencia a construir un modelo del Dominio del Problema y decorar ese modelo con las componentes del programa que implementan cada una de sus partes. La no trivialidad de este proceso llevó al grupo de investigación a profundizar los estudios relacionados con la elaboración de estrategias de interconexión del Dominio del Problema con el Dominio del Programa con las características antes mencionadas. En el marco de esta investigación surgió el Análisis Comportamental, una estrategia que se basa en la recuperación de los conceptos utilizados en el programa y las relaciones existentes entre ellos para recuperar la relación DP-DP, lo cual se ha comprobado que facilita la comprensión [2].

2.2. Estrategia de Análisis Comportamental

Otra estrategia de interconexión de dominios posible; es el Análisis Comportamental (AC) [17]. AC provee una solución al problema de vincular un modelo del Dominio del Problema con las componentes del programa que implementan cada una de sus partes. Esta estrategia logra la relación antes mencionada, mediante el uso de información estática y dinámica del producto de software estudiado. La primera hace referencia a: tipos, variables, funciones, etc.; definidas en el código fuente del programa. La segunda se centra en la recuperación de las componentes de software usadas en tiempo de ejecución.

Para la extracción de información estática se utilizan técnicas de compilación tradicionales [18]. Es decir, se construye un analizador sintáctico con las acciones semánticas apropiadas para extraer la información deseada.

Para la extracción de la información dinámica, una de las posibilidades consiste en definir un esquema de instrumentación de código [19]. Esta técnica consiste en la inserción de sentencias dentro del código fuente del sistema bajo estudio, con la finalidad de recuperar las componentes del programa que se utilizaron para producir la salida.

El esquema de instrumentación de código mencionado en el párrafo anterior, inserta funciones de inspección que muestran los nombres de las funciones ejecutadas por el sistema y otra información que el programador considere importante.

Otra técnica para la extracción de información puede ser la técnica de Profiling [20, 21, 22], que mide el desempeño de un programa; usando la información del mismo cuando se ejecuta; esto permite medir los tiempos y cargas para cada unidad o componente del sistema.

La salida del profiler, puede ser una cadena de registro de eventos (trace) o un resumen estadístico de los eventos observados (profile).

Con base en lo anterior el desarrollador puede implementar cambios que mejoren el desempeño del sistema analizado.

El Debugging [23, 24, 25] es otra técnica para la extracción de información dinámica, que permite al programador encontrar errores en el código, esta técnica se basa en seguir paso a paso la ejecución del programa, informando al programador el estado del resto del sistema en cada momento. En otras palabras el debugging permite recuperar distintos niveles de información del sistema analizado, dependiendo esto del modo de uso.

Se puede resumir que para llevar a cabo el Análisis comportamental se debe: 1) Reproducir cada comportamiento del sistema; 2) Registrar las operaciones realizadas y 3) Procesar la información con el objetivo de recuperar conocimiento.

Por una parte, se utiliza alguna técnica de extracción de información dinámica destinada para tal fin, en este trabajo de investigación, se hace uso de una herramienta llamada Trace [26], perteneciente al módulo de debugging y profiling de la API de Python, para la extracción de dicha información. Por la otra, el programador debe ejecutar el sistema de manera sistemática; donde para cada funcionalidad, se registra una traza de ejecución y se asigna a cada una de ellas un nombre semántico.

Cuando se realizan las tareas explicadas previamente, se aplican técnicas de análisis del flujo de ejecución de las funciones y métodos, que posibiliten la extracción de la información más relevante.

Cuando el programador aplica los procedimientos descriptos para hacer uso de la estrategia de AC, a un concepto (entiéndase por concepto a una funcionalidad del sistema bajo análisis), obtiene programáticamente, solamente las funcionalidades del sistema aisladas y las componentes del programa que las llevan a cabo.

En otras palabras, lo que obtiene el programador es el conocimiento de lo que hace cada concepto del sistema analizado y las funciones y métodos implicados para llevarlos a cabo; esto permite al programador poder vincular dichas funciones y métodos con cada concepto analizado.

Según lo explicado en el párrafo anterior, es posible mejorar la relación entre los dominios mencionados, a través de la introducción de una representación visual del Dominio del Problema, como por ejemplo los Mapas Conceptuales [27], Modelos de Dominio o Casos de Uso de UML, los cuales contienen información sobre las funciones y métodos del sistema que implementan cada concepto y la interacción entre estos.

A modo de ejemplo suponga el análisis de un software editor de textos. Luego de la aplicación de los pasos de la estrategia de AC, la información obtenida se representa en un mapa conceptual como el de la figura 1 que incluye información sobre las funciones y métodos que intervienen en la resolución de la lógica de negocio de dicho editor de textos.

El mapa conceptual de la figura 1 representa mediante círculos los conceptos analizados según criterio del programador y utilizados para la resolución de la lógica de negocio del sistema y mediante flechas las invocaciones de funciones y métodos existentes entre estos.

El lector puede percibir que la relación comportamental-operacional se puede apreciar mejor que la que se obtiene a nivel programático, ya que permite visualizar cómo interactúan los componentes de software entre sí para lograr la funcionalidad del sistema.

Con respecto al procedimiento del programador para obtener la vista gráfica, de la relación comportamental-operacional mostrada en la figura 1. Es deseable que tal

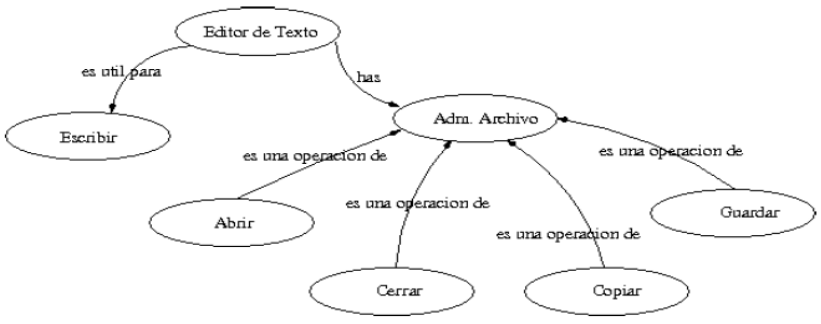


Figura 1. Mapa conceptual de un editor de textos.

procedimiento sea automático y no de manera manual, sobre todo en sistemas grandes, donde la cantidad de funcionalidades y elementos de software utilizados pueden ser significativos. Y además que el tipo de vista obtenido, pueda interactuar o ser utilizado como dato de entrada en algún otro sistema para su análisis, como por ejemplo, lograr una vista en el lenguaje unificado de modelado o UML [28, 29, 30], podría ser un ejemplo de esto.

2.3. Estrategia para Relacionar Conceptos

Para poder llevar a cabo las relaciones existentes entre los conceptos de un sistema, es necesario utilizar alguna estrategia que permita agilizar el trabajo y realizarlo de alguna manera estándar para todos los conceptos que el programador desee analizar.

A continuación se describe una posible estrategia para poder relacionar los conceptos identificados por el programador en el sistema bajo estudio.

La estrategia requiere que se identifiquen en primera instancia los conceptos del sistema a los que se desea aplicar el análisis.

Para poder identificar los conceptos, el programador debe considerar cada opción de menú del sistema como un concepto a ser analizado. Este punto es muy importante porque permite en cierta medida acotar el análisis a ciertos conceptos que elija el programador y por lo tanto se reduce considerablemente el volumen de información que se obtendrá y por consiguiente el tamaño del posible diagrama de la vista grafica que se obtendrá.

A modo de ejemplo suponga la identificación de conceptos para un editor de textos, éste en su menú puede tener las opciones ***de Archivo, Edición, Herramientas, Opciones, y Ayuda***. Por lo tanto el programador puede escoger como conceptos posibles de análisis, las opciones de este menú; por ejemplo el concepto ***Archivo***; el cual contendrá ciertas funcionalidades como ***Abrir un archivo*** o ***Guardar,*** etc.

Luego de la identificación de los conceptos del sistema, se aplica la estrategia de AC, descripta en esta sección a cada uno de ellos para recuperar información dinámica, correspondiente a funciones y métodos invocados en cada concepto.

Para poder representar las relaciones entre los conceptos analizados; la estrategia para relacionar conceptos busca identificar el origen de cada función y método invocado, para ser más específico, lo que busca la estrategia en cuestión, es lograr definir la relación entre el llamador y el llamado, al identificar a que concepto corresponde cada función y método invocado.

Con la información obtenida sobre la correspondencia entre llamador y llamado se puede representar la relación entre conceptos con una flecha entre estos, donde la orientación de la flecha indica quién es el concepto llamador de las funciones y métodos invocados (origen de la flecha) y quien es el receptor llamado (punta de la flecha).

La descripción del párrafo anterior se muestra ejemplificada en el mapa conceptual mostrado para el caso de un editor de textos en la figura 1.

A modo de ejemplo analice los conceptos ***Adm Archivo*** y ***Guardar***, representados por sus respectivos círculos en el mapa conceptual de la figura 1. Puede observarse que la flecha comienza en el círculo correspondiente al concepto ***Guardar*** y termina en el círculo del ***concepto Adm Archivo***. Aquí se puede ver la representación de la relación entre los conceptos citados, donde dicha relación está dada por funciones y métodos que programáticamente son invocados en el concepto ***Guardar*** y son implementados en el concepto ***Adm Archivos***.

2.4. Aplicación de la Estrategia de Análisis Comportamental a un sistema multiparadigma

A continuación se describe la aplicación de la estrategia de análisis comportamental descripta en la sección 2.2.

a) Reproducir sistemáticamente cada comportamiento del sistema.

La reproducción sistemática de cada comportamiento del sistema, consiste en ejecutar el mismo, y tomar una muestra de las funciones y métodos utilizados en cada objetivo; entiéndase por objetivo a la ejecución de una opción del sistema analizado; a este procedimiento se lo conoce también como recuperación de la información dinámica. Para obtener esta información, en este trabajo de investigación, si utilizó la herramienta Trace; que se menciona en la sección 2.2.

Para asegurar la obtención satisfactoria de las trazas de ejecución de los conceptos a ser analizados; la herramienta Trace requiere que se especifique mediante argumentos el tipo de información que debe capturar de la ejecución de cada concepto del sistema bajo estudio, mediante el argumento ***–listfuncs; y*** el nombre del archivo que pone en ejecución la aplicación; además

puede redireccionarse la salida estándar (pantalla) de las trazas de ejecución a un archivo de texto; para esto; es necesario indicar a Trace el signo >>, seguido del nombre del archivo de texto donde se almacenarán las trazas de ejecución obtenidas. En la figura 3 puede observarse el uso de Trace para obtener las trazas de ejecución de un editor de textos tomado como caso de estudio.

b) Registrar las operaciones realizadas.

Trace recolecta todas las llamadas a funciones y métodos utilizados, y almacena esta información en un archivo de texto, el cual, podría también representarse por ejemplo en algún dialecto xml para uso en futuros análisis, como dato de ingreso de información a otro sistema.

El resultado obtenido al reproducir el comportamiento del sistema, incluye información sobre funciones, métodos, y módulos o librerías importadas; lo cual puede llevar a que el volumen de información obtenida sea extremadamente grande.

La herramienta Trace genera un archivo de texto de salida para guardar la información, según el orden en que se ejecutan las llamadas a funciones y métodos. Si el lector observa la figura 4 en la sección 3.3, se puede ver que desde las primeras líneas del archivo, hasta la línea número 28 se registran dichas llamadas a funciones y métodos, que resuelven detalles propios de la implementación y que no hacen a la resolución de la lógica de negocio del sistema analizado; luego desde la línea número 29 hasta el final del archivo, queda el registro de todas las funciones y métodos, invocados que pertenecen a un módulo llamado “Editor”, que es en el cual se implementa la lógica del sistema, y por lo tanto es el que contiene información relevante para el análisis de la comprensión del programa.

El contenido del archivo generado por la herramienta Trace, contiene información adicional a la necesaria para mejorar el entendimiento de la lógica al programador; por lo tanto, es necesario aplicar un filtro a dicha información para conservar solamente aquella que esté referida a resolver la lógica de negocio del sistema bajo análisis.

c) Procesar la información con el objetivo de recuperar conocimiento.

A los archivos generados en el proceso descripto en el párrafo anterior, se les aplica un algoritmo para llevar acabo la tarea de filtrar información irrelevante a la lógica de negocio tal como llamadas a módulos o librerías externas; cargadas antes del análisis de cualquier objetivo y llamadas a funciones o métodos que no son parte de la lógica de negocio, sino que resuelven otros aspectos de la implementación.

En la figura 2, se muestra el código que describe el algoritmo implementado en un programa escrito en Python llamado depurafuncs.py.

El programa depurafuncs.py consta de la siguiente información:

- Entrada de datos: el nombre del archivo de texto generado por la herramienta trace, pasado como primer parámetro.

- Variable ***argv[]:*** almacena el nombre del archivo de entrada de datos. En la línea número 32 de la figura 2 se puede apreciar la llamada a la función ***leer_archivo()*** que hace uso de esta variable.

- Variable ***mod_buscado***: esta variable contiene el nombre del módulo que se buscará en el archivo pasado como parámetro, la variable esta inicializada en el programa, y no es pasado el valor por parámetro, aunque podría serlo.

- Variable ***cantidad***: esta variable se usa como contador de la cantidad de funciones y métodos encontrados, que coinciden con el valor de la variable ***mod_buscado***.

- Función ***leer_archivo()***: esta función es la encargada de abrir el archivo indicado en la variable ***argv[],*** abrir el archivo de salida de datos filtrados y hacer la búsqueda de información según el nombre del módulo que se especifique en la variable ***mod_buscado.*** Desde las líneas de código 15 a 26 de la figura 2, se puede ver en detalle la implementación de la función, donde se aplica un bucle cuyo objetivo es recorrer el archivo que se pasó como parámetro, en busca de líneas que contengan el nombre del módulo especificado en la variable ***mod_buscado.*** Esto es posible porque la herramienta trace registra todas las llamadas de funciones y módulos agregando la información sobre el nombre del módulo al que pertenecen.

- Salida: el programa depurafuncs.py, la función ***leer_archivo()*** es la encargada de generar un archivo de salida cuyo nombre está formado por la palabra “Solo_func_” concatenado con el valor de la variable ***argv[].*** Este archivo de salida, contiene la información filtrada que consta de las funciones y métodos que implementan la lógica de negocio del sistema analizado.

d) Identificación de las relaciones entre conceptos.

```python
import string
import sys
mod_buscado = "Editor"

def leer_archivo(arch):
    cantidad = 0
    archivo = open(arch,'r')
    linea = archivo.readline()
    info = open("Solo_func_" + arch,'a')

    print "Modulo: " + mod_buscado + '\n'
    print "Funciones llamadas: \n"
    info.write("Modulo: " + mod_buscado +'\n')
    info.write("Funciones llamadas: \n\n")
    while linea <> "":
        linea = archivo.readline()
        posicion_func = string.find(linea, mod_buscado)
        #Ubicar la posicion en la cadena, si la encuentra
        if posicion_func != -1 :
            funciones = linea [posicion_func: ]
            print funciones
            cantidad += 1
            info.write(funciones + '\n')
    print "Cantidad de Funciones: " + str(cantidad)

    info.write("Cantidad de Funciones: " + str(cantidad))
    #cierra el archivo
    archivo.close()
    info.close()

if len(sys.argv) == 2:
    leer_archivo(sys.argv[1])
else:
    print "Archivo No especificado. Necesita indicar el nombre de archivo a analizar.'
```

Figura 2. Algoritmo del programa escrito en python depurafuncs.py.

Una vez recuperados los conceptos con las funciones y métodos asociados se lleva a cabo la aplicación de la estrategia para recuperar las relaciones entre los conceptos analizados, que fue definida en la sección 2.3.

Al identificar las relaciones entre los conceptos analizados se puede construir un mapa conceptual que muestre dichas relaciones generando esto una mejora del proceso cognitivo para el programador que lleva a cabo el análisis del sistema bajo estudio.

3. Caso de Estudio – Editor de textos EdiPy– Python

EdiPy es un software desarrollado con el lenguaje multiparadigma Python, y consiste en un editor de textos; el cual dispone de las funcionalidades básicas tales como Abrir archivo, Guardar, Guardar como. Un menú para edición con funciones de copiar texto, pegar y cortar; permite personalizar la interfaz de usuario, cambiando colores del área de trabajo y un menú de herramientas, las cuales son externas al editor de textos, tales como abrir una terminal Linux, abrir un navegador para iniciar la búsqueda en google o Wikipedia, control de hardware tal como apagar pantalla.

A continuación se describe el proceso de aplicación de la estrategia de análisis comportamental al editor de textos, mostrando los resultados obtenidos.

3.1. Reproducir Cada Comportamiento del Sistema

Para llevar a cabo la tarea de reproducción de cada objetivo para analizar el comportamiento del sistema, se utilizó la herramienta Trace mencionada en la sección 2.2.

La aplicación de esta herramienta requiere ejecutar el comando a modo de consola o también conocido como interfaz de comandos del sistema operativo; indicando como parámetros el tipo de análisis que llevara a cabo, el cual puede ser:

- Hacer que se registren las funciones y métodos ejecutados solamente.

- Conteo de la cantidad de veces que se ejecuta cada una de estas funciones y métodos.

- Generar de un reporte por defecto, perteneciente a la herramienta, que muestra información más detallada tal como seguimiento de módulos a los que pertenecen las funciones y métodos ejecutados.

- Indicar el nombre de la aplicación sobre la que se aplicará dicho análisis, en este caso el editor de textos cuyo nombre de archivo es EdyPy.py.

La herramienta Trace, al estar incluida en el módulo de debugging y profiling de la API de Python, funciona con aplicaciones escritas en el mencionado lenguaje de programación únicamente. Esto significa que no puede especificarse otro programa para que Trace haga el análisis; tal como un programa compilado en el lenguaje de programación C por ejemplo.

En la figura 3 se muestra el comando ejecutado para el análisis de un objetivo; concretamente la opción ***Abrir*** perteneciente al menú ***Archivo***. Nótese la incorporación del parámetro ***– listfuncs***, este parámetro está indicando a Trace que registre las llamadas a funciones y métodos invocados durante el análisis dinámico del sistema.

El procedimiento descripto en el párrafo precedente; es el mismo que se aplica sistemáticamente para todas las opciones del menú correspondiente al editor de textos, para poder reproducir cada comportamiento del mismo.

3.2. Registrar las Operaciones Realizadas

La herramienta Trace muestra por defecto en pantalla el resultado del análisis llevado a cabo, pero es posible especificar la generación de un reporte con un formato propio de la herramienta, o como se muestra en la figura 3, donde se redirecciona a otro archivo el resultado de las funciones y métodos que fueron invocados durante la reproducción del objetivo. Este archivo de texto debe ser especificado como parámetro a la herramienta Trace. En la figura 3 se muestra a modo de ejemplo como se redirecciona la salida de pantalla a un archivo de texto cuyo nombre es “Archivo_abrir.txt”.

En la figura 4 se puede ver el contenido del archivo generado con la herramienta Trace. El archivo mostrado en la figura de referencia, se encuentra recortado por cuestiones de tamaño.

El lector puede observar que la información obtenida desde la línea 2 hasta la 28 del archivo de salida de la figura 4, hacen referencia a todas las llamadas a funciones y métodos utilizados por el sistema, es decir, carece de un filtro que permita ver solamente las funciones y métodos que están involucrados en la resolución de la lógica de negocio.

Estas funciones y métodos que hacen a la resolución de la lógica de negocio del sistema, son las registradas cuando se ejecuta una funcionalidad del mismo y se ubican a partir de la línea 29 del archivo de salida mostrado en la figura 4, justamente es en esta línea donde se muestra el comienzo de llamadas a funciones y métodos para el modulo llamado “Editor”; que es donde se implementa la resolución de la lógica de negocio del sistema analizado.

3.3. Procesar la Información con el Objetivo de Recuperar Conocimiento

En esta etapa de la aplicación de la estrategia de AC, se separa la información relevante a la lógica de negocio del sistema de aquella que no lo es; y se almacena en un nuevo archivo de texto. Para llevar a cabo esta tarea, se hace uso del programa depurafuc.py, descripto el inciso c en la sección 2.4. En la figura 5 se muestra un ejemplo de aplicación del programa depurafunc.py a un archivo de salida de la herramienta Trace; concretamente el archivo llamado ***Archivo_abrir.txt***, correspondiente al análisis de funcionalidad del sistema al ejecutar la opción archivo – abrir.

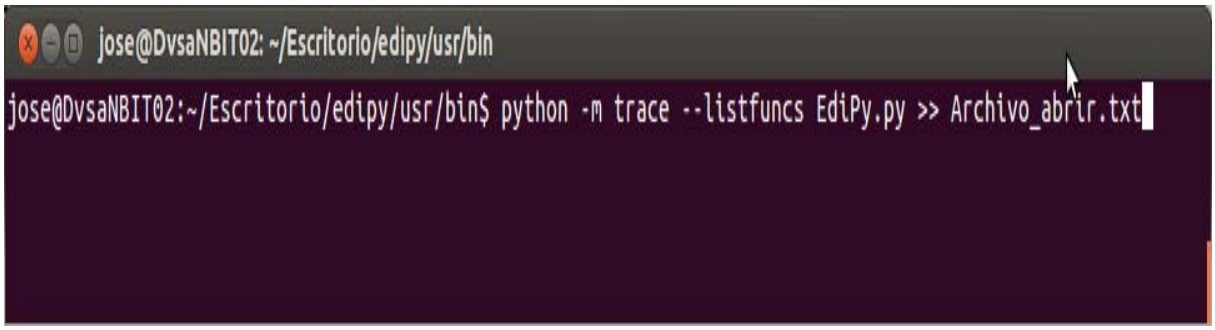


Figura 3. Ejecución de Trace y archivo de salida de la herramienta.

```
functions called:
filename: /usr/lib/python2.7/UserDict.py, modulename: UserDict, funcname: UserDict.__contains__
filename: /usr/lib/python2.7/UserDict.py, modulename: UserDict, funcname: UserDict.__getitem__
filename: /usr/lib/python2.7/genericpath.py, modulename: genericpath, funcname: _splitext
filename: /usr/lib/python2.7/genericpath.py, modulename: genericpath, funcname: isfile
filename: /usr/lib/python2.7/lib-tk/Tkconstants.py, modulename: Tkconstants, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/Tkinter.py, modulename: Tkinter, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/Tkinter.py, modulename: Tkinter, funcname: BaseWidget
filename: /usr/lib/python2.7/lib-tk/tkColorChooser.py, modulename: tkColorChooser, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/tkColorChooser.py, modulename: tkColorChooser, funcname: Chooser
filename: /usr/lib/python2.7/lib-tk/tkCommonDialog.py, modulename: tkCommonDialog, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/tkCommonDialog.py, modulename: tkCommonDialog, funcname: Dialog
filename: /usr/lib/python2.7/lib-tk/tkCommonDialog.py, modulename: tkCommonDialog, funcname: Dialog.__init__
filename: /usr/lib/python2.7/lib-tk/tkCommonDialog.py, modulename: tkCommonDialog, funcname: Dialog.show
filename: /usr/lib/python2.7/lib-tk/tkFileDialog.py, modulename: tkFileDialog, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/tkFileDialog.py, modulename: tkFileDialog, funcname: Directory
filename: /usr/lib/python2.7/lib-tk/tkFileDialog.py, modulename: tkFileDialog, funcname: Open
filename: /usr/lib/python2.7/lib-tk/tkFileDialog.py, modulename: tkFileDialog, funcname: Open._fixresult
filename: /usr/lib/python2.7/lib-tk/tkFileDialog.py, modulename: tkFileDialog, funcname: SaveAs
filename: /usr/lib/python2.7/lib-tk/tkMessageBox.py, modulename: tkMessageBox, funcname: <module>
filename: /usr/lib/python2.7/lib-tk/tkMessageBox.py, modulename: tkMessageBox, funcname: Message
filename: /usr/lib/python2.7/posixpath.py, modulename: posixpath, funcname: basename
filename: /usr/lib/python2.7/posixpath.py, modulename: posixpath, funcname: join
filename: /usr/lib/python2.7/posixpath.py, modulename: posixpath, funcname: split
filename: /usr/lib/python2.7/posixpath.py, modulename: posixpath, funcname: splitext
filename: /usr/lib/python2.7/trace.py, modulename: trace, funcname: _unsettrace
filename: <string>, modulename: <string>, funcname: <module>
filename: EdiPy.py, modulename: EdiPy, funcname: <module>
filename: EdiPy.py, modulename: EdiPy, funcname: Editor
filename: EdiPy.py, modulename: EdiPy, funcname: Editor.__init__
filename: EdiPy.py, modulename: EdiPy, funcname: Editor.controlClear
filename: EdiPy.py, modulename: EdiPy, funcname: Editor.controlQuit
filename: EdiPy.py, modulename: EdiPy, funcname: Editor.controlSetTitle
filename: EdiPy.py, modulename: EdiPy, funcname: Editor.fileOpen
```

Figura 4. Contenido del archivo de salida de Trace. Archivo_abrir.txt

Si se compara el volumen de información del archivo generado por la herramienta Trace con el volumen de información resultante de la aplicación del programa depurafuc.py a dicho archivo, lo que está representado en la figura 4 y figura 5 respectivamente; se puede apreciar una gran diferencia de reducción de volumen en este último; esto es gracias a la aplicación de la etapa de filtrado de información que permite mediante la aplicación del programa depurafunc.py, dejar solo las funciones y métodos relevantes a la lógica de negocio que resuelve este sistema.

Gracias a este filtrado de información, el programador acota la información a ser analizada a las funciones y métodos mencionados en el párrafo anterior, permitiendo mejorar el entendimiento del sistema y agilizar los tiempos en que se lleva a cabo esta tarea.

3.4. Identificación de las Relaciones Entre Conceptos

Con los archivos ya filtrados obtenidos del punto 3.3 se procede a aplicar la técnica de relación entre conceptos descripta en la sección 2.3.

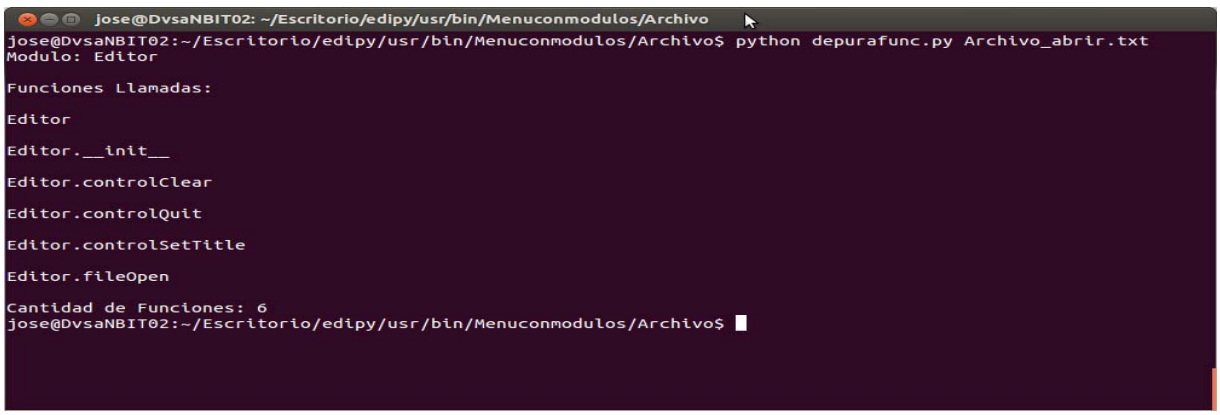


Figura 5. Salida del programa depurafucs.py

Se expone a continuación un ejemplo de relaciones entre conceptos solo para una opción de menú del editor de textos EdiPy, en este caso, el menú ***Archivo*** del citado editor.

En la figura 6 se puede ver las funciones y métodos filtrados y que intervienen en la lógica del concepto ***Archivo,*** por cuestiones de espacio no se muestran todos los archivos con sus respectivas funciones y métodos, sino que se hizo un extracto solo de funciones y métodos que fueron invocados al ejecutar el sistema y seleccionar cada una de las opciones del menú ***Archivo.***

Con la identificación de conceptos y relaciones obtenidas entre estos, se puede crear una vista grafica en un mapa conceptual, que permita mejorar el entendimiento al programador sobre el funcionamiento del software EdiPy. En la figura 7 se presenta el mapa conceptual resultante del análisis realizado, que muestra los conceptos (círculos) y relaciones (fechas) analizadas en esta sección para el menú ***Archivo*** del software EdiPy.

El lector puede apreciar con facilidad el propósito que cumple el menú archivo del software EdiPy; ya que está claramente representada la relación entre los conceptos e indicado con un nombre representativo tanto para de cada concepto como para las funciones o métodos invocados.

4. Conclusiones y Trabajo de Investigación Futuro

La comprensión de programas es un área muy importante dentro de la ingeniería de software, debido a que esta permite agilizar el entendimiento del software, en la etapa de mantenimiento del mismo.

Para lograr su propósito la comprensión de programas trabaja en el desarrollo de técnicas, métodos, estrategias y herramientas que ayuden a mejorar el proceso cognitivo del programador y así este pueda lograr un entendimiento global del sistema analizado. El éxito de esto radica en lograr reconstruir la relación existente entre el Dominio del Problema y el Dominio del Programa.

En este trabajo de investigación, se aplican técnicas de comprensión de programas a sistemas multiparadigma; con el objetivo de mejorar la comprensión de este tipo de sistemas.

Una de las técnicas desarrolladas para lograr la relación entre los dominios del problema y programa, es la técnica de Análisis comportamental, la cual hace uso de información estática y dinámica del sistema bajo estudio, para lograr su cometido.

Para la aplicación de la estrategia de análisis comportamental (AC), es necesario obtener información dinámica del sistema analizado, esto hace referencia a información que se obtiene mientras el sistema analizado es ejecutado.

Existen diversas técnicas para obtener información dinámica de un sistema, algunas de estas son el profiling; el cual mide el desempeño de un programa mientras es ejecutado; el debugging, utilizado para encontrar fallos en el funcionamiento del código fuente escrito. Otra técnica para extraer información dinámica utilizada en la estrategia de AC, es la instrumentación de código, la cual consiste en insertar sentencias en determinadas partes del código, para poder recuperar las componentes del mismo, ejecutadas para obtener la salida del sistema.

En este trabajo de investigación para lograr la extracción de información dinámica del sistema analizado se utilizó una herramienta llamada Trace [26], perteneciente al módulo de debugging y profiling de la API de Python; esta herramienta permite obtener información mientras el sistema es ejecutado, pudiendo almacenar la misma en archivos de texto, para su posterior análisis.

Para mejorar el entendimiento de la lógica del sistema al programador, es necesario aplicar técnicas de filtrado a la información obtenida mediante la estrategia de análisis comportamental; para centrar el estudio solo en aquellas funciones y métodos que estén altamente relacionados con la lógica del sistema o domino del problema. Por lo

```
funcname: Editor
funcname: Editor.__init__        ----->  Constructor del sistema inicial.
funcname: Editor.fileNew         ----->  Invocacion a funcion para generar un nuevo archivo.
funcname: Editor.fileOpen        ----->  Invocacion a funcion para abrir un archivo.
funcname: Editor.fileSave        ----->  Invocacion a funcion para guardar archivo.
funcname: Editor.fileSaveAs      ----->  Invocacion a funcion para gurdar con nombre un archivo
funcname: newWin                 ----->  Invocacion a funcion para abrir una ventadna nueva del editor
```

Figura 6. Extracto de relaciones para el concepto Menu Archivo. Software EdiPy

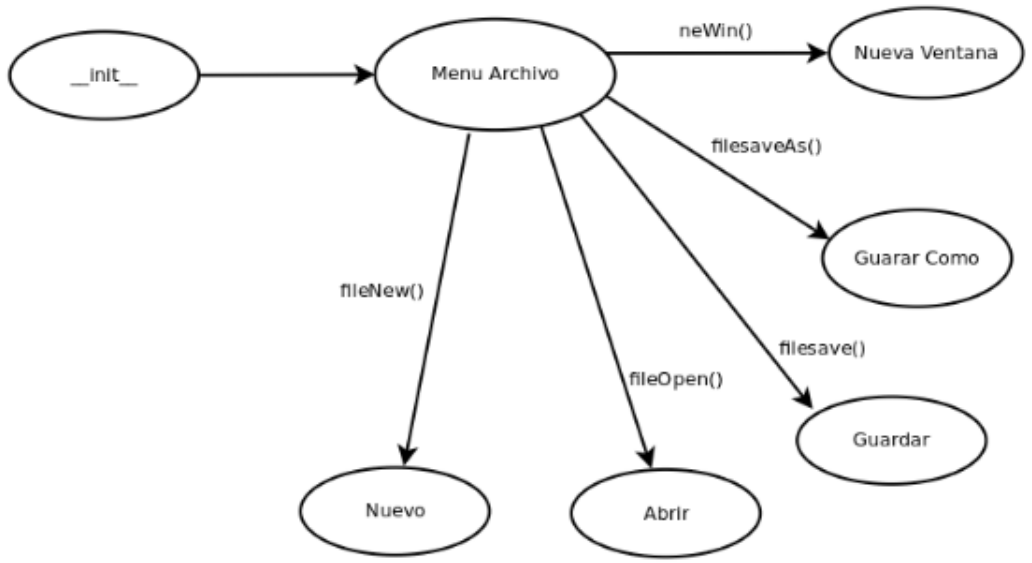


Figura 7. Mapa conceptual Menu Archivo. Software EdiPy

tanto se plantea un programa llamado depurafunc.py desarrollado en Python que permite llevar a cabo dicho filtro de información.

La aplicación de la estrategia de análisis comportamental, como herramienta para la comprensión de programas, permite aplicar pasos ordenados para lograr el objetivo de mejorar el proceso cognitivo al programador; que lleva a cabo tareas de mantenimiento de software.

Lograr correctamente el objetivo, de la implementación de la estrategia de análisis comportamental, depende de factores tales como el conocimiento técnico del programador sobre el lenguaje de programación utilizado en el desarrollo del sistema bajo análisis, las particularidades del sistema analizado, la capacidad cognitiva del programador para identificar conceptos a ser analizados y poder plasmar en alguna vista grafica las relaciones entre dichos conceptos; lo que permite hacer más fuerte la visualización de tales relaciones y mejorar el entendimiento de las funcionalidades de cada concepto.

La representación mediante vistas graficas de la información obtenida, es fundamental, para lograr hacer más fuerte la percepción para el programador de la relación existente entre el dominio del problema y el dominio del programa. Es por este motivo que se busca mejorar las vistas existentes o desarrollar nuevas vistas para ser aplicadas al análisis de sistemas multiparadigma.

Los trabajos de investigación a futuro, están dirigidos a:

- Mejorar las vistas existentes y adaptarlas a sistemas multiparadigma o desarrollar nuevas vistas para tales sistemas.

- Adaptar la técnicas de filtrado de información a sistemas multiparadigma o desarrollar nuevas; ya que es una herramienta fundamental, para controlar el tamaño de información obtenida del sistema bajo estudio y mejorar el entendimiento de la lógica al programador.

- Desarrollar e implementar la idea de estrategia de relaciones entre conceptos presentada en este trabajo de investigación.

- Extender el alcance de la aplicación de la estrategia de Análisis Comportamental y la estrategia para relacionar conceptos a sistemas que por sus características no disponen de un conjunto de opciones bien estructuradas bajo un menú de opciones.

5. Referencias

[1] C. Exton. Constructivism and program comprehension strategies. Program Comprehension, 2002. Proceedings. 10th International Workshop on, pages 281-284, 2002.

[2] M. Beron, P. Henriques, and R. Uzal. Program Inspection to interconnect Behavioral and Operational Views for Program Comprehension. Ph.D Thesis Dissertation at University of Minho. Braga. Portugal, 2010.

[3] Soh, Z., "Context and Vision: Studying Two Factors Impacting Program Comprehension," Program Comprehension (ICPC), 2011 IEEE 19th International Conference on , vol., no., pp.258,261, 22-24 June 2011

[4] Thomas D. Latoza, David Garlan, James D. Herbsleb, and Brad A. Myers. Program comprehension as fact finding. In In Proceedings of Foundations of Software Engineering, pages 361–370, 2007.

[5] M. A. Storey. Theories, methods and tools in program comprehension: past, present and future. Proceedings of the 13th International Workshop on Program Comprehension, pages 181–191, 2005.

[6] RamosCarvalho, N., "An ontology toolkit for problem domain concept location in program comprehension," Software Engineering (ICSE), 2013 35th International Conference on , vol., no., pp.1415,1418, 18-26 May 2013

[7] A. Walenstein. Theory-based analysis of cognitive support in software comprehension tools. In IWPC ’02: Proceedings of the 10th International Workshop on Program Comprehension, page 75, Washington, DC, USA, 2002. IEEE Computer Society.

[8] A. Von Mayrhauser and A. M. Vans. Program comprehension during software maintenance and evolution. Computer, 28(8):44-55, 1995.

[9] Cognitive design elements to support the construction of a mentalmodel during software visualization. Storey, M.A.D. and Fracchia, FD and Muller, HA. Program Comprehension, 1997. IWPC'97. Proceedings., Fifth Iternational Workshop on pages 17-28.

[10] T. Eisenbarth, R. Koschke, and D. Simon. Aiding program comprehension by static and dynamic feature analysis. In ICSM'01: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01), page 602, Washington, DC, USA, 2001.

[11] M. Beron, P. Henriques, and R. Uzal. Program Inspection to interconnect Behavioral and perational Views for Program Comprehension. Technical Report, 2007.

[12] M. A. Storey, F. D. Fracchia, and Muller. Cognitive design elements to support the construction of a mental model during software exploration.

[13] J. D. Novak and D. B. Gowin. Learning How to Learn. Cambridge University Press, 1984.

[14] H. Lieberman and C. Fry. Bridging the gulf between code and behavior in programming. In ACM conference on Computers and Human Interface, Denver, Colorado, April 1994.

[15] M. A. Storey, K. Wong, and M uller. How do program understanding tools affect how programmers understand programs?. SCI. Comput. Program Year 2000.

[16] Mario M. Ber_on, Maria Jo~ao V. Pereira, Nuno Oliveira, and Daniela da Cruz. Svs, bors, svsi: Three strategies to relate problem and program domains. In Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension, ICPC '10, pages 60-61, Washington, DC, USA, 2010. IEEE Computer Society.

[17] M. Berón, P. Henriques, M. Varanda, and R. Uzal. “Inspección de Programas para Interconectar las Vistas Comportamental y Operacional para la Comprensión de Programas”. Ph.D Thesis. Universidad Nacional de San Luis, San Luis, Argentina, 2009. Pag 143 – 146.

[18] A. V. Aho, R. Sethi, and J. D. Ullman. Compilers Principles, Techniques and Tools. 1986.

[19] Y. Wang, L. Qingshan, C. Ping, and Ren C. Dynamic Fan-in and Fan-out Metrics for Program Comprehension. Program Comprehension through Dynamic Analysis, 1:38-42,2005.

[20] E. Coppa, C. Demetrescu, and I. Finocchi, Input-Sensitive Profiling, Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2012), ACM SIGPLAN Notices, Vol. 47, No. 6, pp. 89-98, 2012.

[21] D. Zaparanuks and M. Hauswirth, Algorithmic Profiling, Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2012), ACM SIGPLAN Notices, Vol. 47, No. 6, pp. 67-76, 2012.

[22] Masri, W.; Daou, J.; Assi, R.A, "State Profiling of Internal Variables," Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on , vol., no., pp.332,335, March 31 2014-April 4 2014.

[23] Bill Blunden: Software Exorcism: A Handbook for Debugging and Optimizing Legacy Code, APress, 2003. ISBN 1-59059-234-4.

[24] Andreas Zeller: Why Programs Fail: A Guide to Systematic Debugging, Morgan Kaufmann, 2005. ISBN 1-55860-866-4.

[25] Layman, L.; Diep, M.; Nagappan, M.; Singer, J.; Deline, R.; Venolia, G., "Debugging Revisited: Toward Understanding the Debugging Needs of Contemporary Software Developers," Empirical Software Engineering and Measurement, 2013 ACM / IEEE International Symposium on , vol., no., pp.383,392, 10-11 Oct. 2013.

[26] Documentación Estándar Python. Módulo Debugging and Profiling. https://docs.python.org/2/library/trace.html. Último acceso Agosto 2014.

[27] D. P. Ausubel. Psicología Educativa: Un Punto de Vista Cognoscitivo. México, DF: TRILLAS, 1991.

[28] Especificación de UML. http://www.uml.org/ último acceso Julio de 2014.

[29] G. Booch, J. Rumbaugh, and I. Jacobson. Unied Modeling Language User Guide. Addison-Wesley Object Technology Series, 2005.

[30] G. Booch, Rumbaugh J., and I. Jacobson. The Unied Software Development Process. Addison-Wesley Object Technology Series, 2004.

Una Taxonomía de Atributos de Calidad para la Evaluación de Arquitecturas de Software por medio de Simulación

María Julia Blas, Silvio Gonnet, Horacio Leone
INGAR, Universidad Tecnológica Nacional, CONICET
Avellaneda 3657, 3000 Santa Fe, Argentina
{mariajuliablas, sgonnet, hleone}@santafe-conicet.gov.ar

Abstract

En este trabajo se presenta una taxonomía de atributos de calidad basada en la necesidad de evaluar arquitecturas de software utilizando técnicas de simulación. El modelo de calidad tomado como punto de partida en esta propuesta corresponde al descripto en el estándar ISO/IEC 25010 (2011), quedando conformado por un conjunto de características y subcaracterísticas aplicables a la mayoría de los productos de software existentes en la actualidad. Los criterios empleados en la elaboración de la clasificación resultante forman parte de las bases de este trabajo, brindando un esquema aplicable a otros contextos en los cuales sea necesario determinar la habilidad de una característica para ser simulada. La taxonomía final muestra que es posible estimar el grado de adecuación de un sistema a determinados atributos de calidad en una instancia previa a la implementación. De esta manera se realiza un primer aporte a las etapas iniciales del desarrollo de software, las cuales son claves para la calidad del producto final. Específicamente se brinda un contexto en el cual es posible obtener información fiable de la calidad del producto (para la toma de decisiones y la prevención de defectos) a partir de la simulación del diseño arquitectónico.

1. Introducción

La calidad a nivel de software es subjetiva [1]. La falta de consenso sobre su significado y la ausencia de un formalismo específico para su medición, han dado lugar a numerosos esfuerzos por lograr unanimidad sobre su ámbito y alcance. En este sentido, diversos autores y organismos han formulado definiciones que ayudan a comprender el concepto. La IEEE define calidad de software como "el grado en el cual un software posee una combinación de atributos deseados" [2]. Pressman establece que la calidad de un sistema de software es "la concordancia del sistema con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente" [3]. Una definición similar se presenta en [4], donde se especifica que la calidad de un sistema de software es "una característica directamente relacionada con la habilidad del sistema para satisfacer sus requerimientos funcionales y no funcionales, tanto implícitos como explícitos".

Aunque existen ciertas diferencias entre las definiciones precedentes, en todas se resalta la importancia de evaluar la calidad del software en función del estudio de los atributos asociados a los requerimientos del sistema. Este hecho guarda relación con el concepto de calidad tradicional que se define como "el grado en el cual el conjunto de características inherentes del producto en cuestión cumple con los requerimientos" [5]. Sin embargo, en el contexto de desarrollo de software la realización del análisis de calidad debe contemplar dos niveles de abstracción: calidad de diseño y calidad de concordancia [3]. La primera, refiere al conjunto de características específicas de un único elemento; mientras que la segunda trata el grado de cumplimiento de las especificaciones de diseño durante su construcción. De esta manera, la calidad de diseño comprende los requisitos, las especificaciones y el diseño del sistema, mientras que la calidad de concordancia apunta a la implementación.

En consecuencia, es evidente que la calidad moldea todos los aspectos del proceso de desarrollo de un software [3]. Desde la elicitación de requerimientos hasta la ejecución del sistema, es crítico que el equipo de trabajo comprenda la importancia de generar software de alta calidad. Para esto, es necesario traducir los requerimientos identificados en las primeras fases de la tarea de elicitación a un conjunto de atributos de calidad coherente y completo. Un atributo de calidad (AC) es una caracterización o propiedad específica de un sistema de

software que puede tomar un valor cuantitativo o cualitativo, el cual es medible u observable [4]. Cuando se especifica una valoración para un AC, se obtiene un requerimiento de calidad (RC). Debido a que entre los distintos AC existen relaciones e interacciones, no es posible que un sistema complejo alcance un único AC de forma aislada. Una taxonomía de AC se denomina modelo de calidad (MC) [6]. Un MC sirve como marco de trabajo para la especificación, evaluación y testeo de la calidad de un sistema. Un modelo que contemple atributos relacionados a artefactos intermedios del proceso de desarrollo, será un MC a nivel de diseño; mientras que un modelo que trabaje sobre la calidad de la implementación desarrollada en relación a los RC será un MC a nivel de concordancia.

En la actualidad, muchos autores han centrado su atención en el estudio y desarrollo de MC a fin de estimular su utilización en diferentes contextos. En [7] los autores presentan un análisis de los MC clásicos junto con una comparativa de los mismos en relación a su aplicabilidad en software comercial, mientras que en [8]–[10] se exponen MC desarrollados para determinar la calidad en ambientes específicos. De forma paralela, se han realizado numerosos esfuerzos por predecir la adecuación del sistema final a requerimientos de calidad individuales. Estas predicciones se basan en la utilización de técnicas computacionales sobre artefactos intermedios (obtenidos del proceso de desarrollo), como la especificación de requerimientos o la arquitectura de software [11], [12]. Toda técnica que permita valorar el ajuste de una arquitectura candidata con el sistema en cuestión, es conocida como técnica de evaluación de arquitecturas de software (EAS) [1], [13]. Su utilidad es clara: permite validar la conformidad del diseño arquitectónico desarrollado con los factores de interés asociados a los atributos de calidad requeridos. En este sentido, la EAS ha motivado a diferentes autores en sus trabajos [14]–[16]. Algunos de ellos, han utilizado técnicas de simulación para tales fines. Según Shannon [17], simular es "el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias para el funcionamiento del mismo". En el contexto de EAS, el modelo lo constituye el diseño arquitectónico y la finalidad es evaluar el comportamiento del sistema conforme un conjunto de AC que se derivan de un MC previamente seleccionado.

En este trabajo se presenta un análisis del MC de producto propuesto en la norma ISO/IEC 25010 junto con una clasificación de los atributos identificados según su posibilidad de derivarse por medio de técnicas de simulación. Además se analiza la factibilidad de utilizar estos atributos en el contexto del diseño arquitectónico, debido a que la información necesaria para su derivación no siempre se encuentra disponible. El objetivo principal es encontrar aquellos AC que puedan analizarse a partir de la arquitectura, pero que además puedan evaluarse por medio del uso de técnicas de simulación. De esta manera, se busca derivar un MC aplicable a nivel arquitectónico que facilite la determinación de las características de calidad del sistema futuro. Se ha elegido trabajar sobre el estándar ISO/IEC 25010 debido a que es una de las normas más estables de la disciplina. El modelo que propone el estándar es una evolución del esquema especificado en ISO/IEC 9126, el cual ha sido ampliamente difundido y utilizado desde sus orígenes hasta su reemplazo por este nuevo modelo [18]. Los aportes de esta nueva esquematización deben ser tenidos en cuenta al momento de evaluar la calidad esperada en un producto de software.

El resto del trabajo se encuentra estructurado de la siguiente manera. La sección 2 describe la familia de normas ISO/IEC 25000 en términos de sus divisiones, finalizando con un breve análisis de las relaciones existentes entre cada una de ellas y la importancia de su MC, el cual se presenta en la sección 3. La sección 4 enuncia los criterios aplicados para la clasificación de los AC según la posibilidad de evaluarlos por medio de simulación, mientras que la sección 5 presenta la taxonomía elaborada. Finalmente la sección 6 sintetiza las conclusiones y trabajos futuros asociados al resultado obtenido, haciendo énfasis en la aplicabilidad de la taxonomía creada como base para la evaluación de arquitecturas de software.

2. Serie ISO/IEC 25000: SQuaRE (Software Product Quality Requirements and Evaluation)

El objetivo de la serie ISO/IEC 25000 es la creación de un marco de trabajo común que facilite la evaluación de la calidad del producto software. ISO/IEC 25000 organiza, amplia y unifica el conjunto de estándares que cubren los procesos de calidad del software asociados a la especificación de requisitos y a la evaluación de la misma. Para tales fines, reemplaza a la primera generación de estándares de calidad del software compuesta por las normas ISO/IEC 9126 (Software Product Quality) e ISO/IEC 14598 (Software Product Evaluation). Mientras que ISO/IEC 9126 describía las características de un modelo de calidad del producto software, la norma ISO/IEC 14598 fijaba el proceso de evaluación de estos productos. En este sentido, la nueva serie forma una segunda generación de estándares de calidad del software.

2.1. Divisiones

Existen seis divisiones en la cuales se organizan los estándares de esta serie (Figura 1). Cada una de ellas

incluye un conjunto de normas de ámbito específico. Dado que esta familia de estándares es relativamente nueva, algunas de las normas especificadas aún no han sido publicadas.

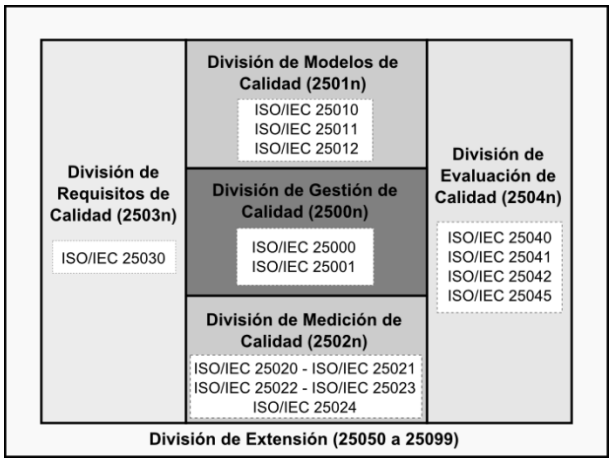


Figura 1. Divisiones ISO/IEC 25000 (Adaptado de [18]).

2.1.1. ISO/IEC 2500n: División de Gestión de la Calidad. Los estándares incluidos en esta división definen los modelos, términos y definiciones comunes a la serie.

La norma ISO/IEC 25000 se denomina "Guía para SQuaRE". Contiene el modelo de la arquitectura propuesta por la serie, la terminología, un resumen de las partes de cada uno de los estándares componentes, los usuarios previstos, un detalle de las secciones de interés para cada uno de ellos y los modelos de referencia. Por otro lado, ISO/IEC 25001 ha sido designado como el estándar de "Planeamiento y gestión", estableciendo los requisitos y las orientaciones que deben utilizarse para gestionar la evaluación y especificación de los requisitos del software.

En ambos casos la última versión es del año 2014.

2.1.2. ISO/IEC 2501n: División de Modelos de Calidad. El objetivo de los estándares incluidos en esta división es definir MC de utilidad para ámbitos específicos. Incluye tres normas: ISO/IEC 25010, ISO/IEC 25011 y ISO/IEC 25012.

La norma ISO/IEC 25010 denominada "Modelos de calidad de sistemas y de software", describe un MC aplicable al software junto con un modelo para la calidad de uso. En la sección 3 se presenta un análisis detallado del contenido de su última versión (2011). El estándar ISO/IEC 25012, "Modelo de calidad de datos", detalla un modelo general para la calidad de los datos, siendo aplicable sólo a aquellos datos que se encuentren almacenados de forma estructurada dentro de un sistema de información. La última versión publicada de este estándar es del año 2008.

En cuanto al ISO/IEC 25011 "Modelo de calidad para servicios de TI", aún no se ha presentado la primera versión estable. Se espera que esta norma defina un MC aplicable al diseño, transición, entrega y mejora de los servicios de TI, proveyendo además una guía que establezca como utilizar el modelo de calidad de uso expuesto en ISO/IEC 25010 para la descripción de la calidad de uso en este tipo de servicios.

2.1.3. ISO/IEC 2502n: División de Medición de Calidad. Esta división contiene un conjunto de normas que presentan modelos de referencia para la medición de la calidad del producto de software, definiciones de medidas de calidad y guías prácticas para su aplicación.

La ISO/IEC 25020 se denomina "Guía y modelo de referencia para la medición" y define un modelo de referencia común a los elementos de medición de la calidad. Proporciona además una guía para que los usuarios seleccionen, desarrollen y/o apliquen las medidas propuestas por las normas ISO. Por otro lado, la ISO/IEC 25021, "Elementos de medición de la calidad", especifica un conjunto recomendado de métricas que pueden utilizarse a lo largo de todo el ciclo de vida del desarrollo de un software. Las últimas versiones publicadas de estas normas corresponden a los años 2007 y 2012 respectivamente.

El resto de los estándares incluidos en esta división aún no han sido publicados. Tanto el ISO/IEC 25022, "Medición de la calidad de uso", como el ISO/IEC 25023, "Medición de la calidad del sistema/software", y ISO/IEC 25024, "Medición de la calidad de datos", se encuentran en desarrollo. La intención es que cada una de estas normas es definir las métricas asociadas a la medición de la calidad en relación a diferentes aspectos del software.

2.1.4. ISO/IEC 2503n: División de Requisitos de Calidad. Esta división tiene por objetivo especificar requisitos de calidad que puedan ser utilizados durante la elicitación de requerimientos o como entrada del proceso de evaluación. Para esto, incluye un único estándar que provee un conjunto de recomendaciones para la especificación de los requisitos de calidad del software. Esta norma es la ISO/IEC 25030 denominada "Requerimientos de calidad", cuya última versión publicada es del año 2007.

2.1.5. ISO/IEC 2504n: División de Evaluación de Calidad. Los estándares incluidos en esta división proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software.

La norma ISO/IEC 25040 propone un modelo general para la evaluación del software en el cual considera las entradas, las restricciones y los recursos requeridos para

generar las salidas deseadas. Por este motivo, se denomina "Guía y modelo de referencia para la evaluación". Su última versión publicada es del año 2011. Como complemento, la norma ISO/IEC 25041 "Guía de evaluación para desarrolladores, adquisidores y evaluadores independientes" del año 2012, describe los requisitos y recomendaciones para la implementación práctica de la evaluación del software desde el punto de vista de diferentes actores. Por otra parte, el estándar ISO/IEC 25042, "Módulos de Evaluación", especifica el contenido de un módulo de evaluación junto con la documentación, estructura y contenido que debe utilizarse al momento de definir este tipo de módulos. Tomando como base esta definición, la norma ISO/IEC 25045 define un módulo para la evaluación del atributo recuperabilidad.

2.1.6. ISO/IEC 25050 a 25099: División de Extensión. Esta división se reserva para normas o informes técnicos que traten dominios de aplicación específicos o que puedan ser utilizados para complementar otras normas de la serie SQuaRE. En la actualidad, incluye las normas ISO/IEC 25051, 25062, 25063, 25064 y ISO/NP 25065 junto con el reporte técnico ISO/IEC TR 25060.

2.2. Relación entre Normas e Importancia del Modelo de Calidad

La serie SQuaRE define un marco teórico que contextualiza múltiples aspectos de la calidad del software. Debido a que estos aspectos no son independientes, existen ciertas relaciones entre ellos que permiten vincular los distintos estándares.

De esta manera, mientras que las normas de la división de Gestión de la Calidad definen un marco de soporte para las divisiones restantes, las divisiones de Modelos, Medición, Requisitos y Evaluación de Calidad se relacionan por medio de distintas tareas. Ya sea para sistemas de software o para datos, se define un MC junto con una serie de métricas aplicables para su valoración. Además, se especifican recomendaciones a tener en cuenta al momento de definir los requerimientos de calidad en base a los propuestos por el modelo en cuestión. Esto se debe a que tales requerimientos pueden ser categorizados por medio del uso de la taxonomía propuesta en este modelo, lo que permite que el MC guíe la especificación. Complementariamente, una buena definición de requerimientos será fundamental para el proceso de evaluación del producto de software.

Como puede observarse, la base de estas relaciones está dada por el MC. Por este motivo, es importante analizar su estructura, a fin de conocer cuales son los atributos de calidad que el conjunto de estándares considera fundamentales para cada ámbito en particular. A los fines prácticos del presente trabajo, sólo se realiza una descripción detallada del MC propuesto para sistemas de software.

3. Modelo de Calidad de Producto del Estándar ISO/IEC 25010

Un MC es un conjunto de elementos que proporciona una base para la especificación de requerimientos y para la evaluación de la calidad de un producto de software. En términos globales queda constituido por un conjunto de factores de calidad y sus relaciones, siendo generalmente estructurado como una jerarquía de factores genéricos que se descomponen en factores específicos a distintos niveles de trabajo.

Como se mencionó con anterioridad, la norma ISO/IEC 25010 modifica el MC de producto presentado en el estándar ISO 9126-1. La actualización que describe la serie ISO/IEC 25000 incorpora dos nuevas características y redefine el conjunto de subcaracterísticas asociadas a cada una de las cualidades externas.

En términos específicos, el MC clasifica la calidad del producto de software en base a un conjunto estructurado de características. Cada característica representa una cualidad externa (es decir, una propiedad que puede ser experimentada por el usuario), la cual se consigue realizando un balance de las subcaracterísticas que la componen. Una subcaracterística puede medirse de forma externa o interna, siendo manifestada cuando el software es utilizado como parte de un sistema. Siguiendo esta jerarquía, cada subcaracterística se divide en atributos. Un atributo es una entidad que puede ser verificada o medida sobre el producto de software. La Figura 2 muestra esta jerarquía. Como puede observarse, una característica puede tener una o más subcaracterísticas y cada subcaracterística puede quedar conformada por uno o más atributos.

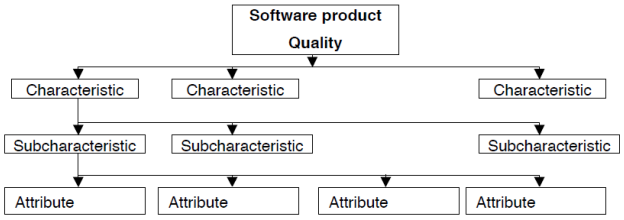


Figura 2. Jerarquía del modelo de calidad [18].

La Tabla 1 esquematiza el MC propuesto en la norma. Como puede observarse, consiste de ocho características que cubren las diferentes propiedades del software (tanto estáticas como dinámicas). Los atributos no se encuentran definidos en el estándar debido a que su aplicabilidad varía según el tipo de producto de software analizado. A continuación se presenta en detalle las

características y subcaracterísticas incluidas en el estándar junto con una referencia a los atributos asociados a cada una de estas últimas.

Tabla 1. Modelo de calidad del estándar ISO/IEC 25010.

	Característica	*Subcaracterística*
CALIDAD DEL SOFTWARE	Funcionalidad	Completitud
		Correctitud
		Adecuación
	Rendimiento	Comportamiento temporal
		Utilización de recursos
	Compatibilidad	Coexistencia
		Interoperabilidad
	Usabilidad	Adecuación
		Aprendizaje
		Operabilidad
		Protección contra errores
		Estética de interfaz de usuario
		Accesibilidad
	Confiabilidad	Madurez
		Disponibilidad
		Tolerancia a fallos
		Capacidad de recuperación
	Seguridad	Confidencialidad
		Integridad
		No repudio
		Autenticidad
		Responsabilidad
	Mantenibilidad	Modularidad
		Reusabilidad
		Analizabilidad
		Modificabilidad
		Capacidad de testeo
	Portabilidad	Adaptabilidad
		Facilidad de instalación
		Capacidad de reemplazo

3.1. Características y Subcaracterísticas

3.1.1. Funcionalidad. Es la capacidad del producto de software para proporcionar el conjunto de funciones que satisfacen las necesidades (tanto declaradas como implícitas) de los usuarios.

Se divide en tres subcaracterísticas: completitud, correctitud y adecuación. La completitud referencia el grado en el cual el conjunto de funcionalidades cubre el conjunto de tareas y objetivos especificados por el usuario. La correctitud trata la capacidad del producto o sistema para proveer resultados correctos con un nivel de precisión requerido. La adecuación refiere a la habilidad del producto de software para proporcionar un conjunto apropiado de funciones para tareas y objetivos específicos.

3.1.2. Rendimiento. Representa el desempeño del producto en lo referente a la cantidad de recursos utilizados bajo determinadas condiciones.

Se divide en dos subcaracterísticas: comportamiento temporal y utilización de recursos. El comportamiento temporal hace referencia a la evaluación de tiempos de respuesta, procesamiento y tasas de "throughput" en condiciones de ejecución particulares; mientras que la utilización de recursos apunta a la cantidad y tipos de recursos utilizados cuando el software es ejecutado en contextos específicos.

3.1.3. Compatibilidad. Es la capacidad de dos o más sistemas o componentes de intercambiar información y/o llevar a cabo sus funciones cuando comparten el mismo entorno de hardware o software. Esta característica se incorporó al modelo como parte del estándar en estudio. El MC presentado en ISO 9126-1 no contemplaba como característica la habilidad de los sistemas para intercambiar información.

Se divide en dos subcaracterísticas: coexistencia e interoperabilidad. La coexistencia refiere a la capacidad del producto para residir en un entorno común con otro software de forma tal que la compartición de recursos no le genere inconvenientes. La interoperabilidad apunta a la habilidad de dos o más sistemas (o componentes) de intercambiar información y, posteriormente, utilizar la información intercambiada para cumplir con sus funciones.

3.1.4. Usabilidad. Es la capacidad del producto de software para ser entendido, aprendido, usado y atractivo para con el usuario final.

Se divide en seis subcaracterísticas: adecuación, aprendizaje, operabilidad, protección contra errores, estética de interfaz de usuario y accesibilidad. La adecuación refiere a la habilidad del producto para dar a conocer sus capacidades y funciones, de forma tal que el usuario pueda entender si el software se ajusta o no a sus necesidades. El aprendizaje apunta a la capacidad que tiene el producto de facilitar su entendimiento en términos de enseñanza. La operabilidad trabaja sobre el grado en el cual el software permite a los usuarios operarlo y controlarlo con facilidad, mientras que la protección contra errores refleja la capacidad del sistema para proteger a los usuarios de cometer errores. La estética de la interfaz de usuario apunta a la habilidad de

la misma para agradar y satisfacer la interacción con los usuarios. Finalmente, la accesibilidad trata la capacidad del producto de ser utilizado por usuarios con determinadas discapacidades.

3.1.5. Confiabilidad. Es la capacidad de un sistema o componente de software de llevar a cabo sus funciones cuando se lo utiliza bajo condiciones y periodos de tiempo determinados.

Se divide en cuatro subcaracterísticas: madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación. La madurez refiere a la capacidad del sistema para satisfacer las necesidades de confiabilidad en condiciones normales de ejecución. La disponibilidad trata la capacidad del sistema o componente de estar operativo y accesible para su uso en el instante en que se lo requiere. La tolerancia a fallos apunta a la habilidad del software de operar según lo previsto en presencia de fallos (tanto de hardware como de software), mientras que la capacidad de recuperación trabaja sobre la destreza del componente para recuperar datos directamente afectados por un fallo o interrupción y restablecer el estado deseado del sistema.

3.1.6. Seguridad. Representa la capacidad del producto de mantener protegida su información y datos de forma tal que tanto personas como sistemas no autorizados no puedan leerlos y/o modificarlos. Junto con compatibilidad es una de las características incorporadas al MC a partir del surgimiento del estándar ISO/IEC 25000.

Se divide en cinco subcaracterísticas: confidencialidad, integridad, no repudio, autenticidad y responsabilidad. La confidencialidad refiere a la capacidad del software para protegerse de accesos no autorizados a datos e información (ya sean accidentales o deliberados). La integridad trata la habilidad del sistema para prevenir accesos y/o modificaciones no autorizados a datos o programas del equipo sobre el cual se está ejecutando. El no repudio representa la capacidad de demostrar las acciones o eventos que han tenido lugar, de forma tal que estas acciones o eventos no puedan ser repudiados posteriormente. La autenticidad apunta a la habilidad de demostrar la identidad de un sujeto o un recurso, mientras que la responsabilidad lo hace a la capacidad de rastrear de forma inequívoca las acciones de una entidad (ya sea una persona u otro sistema de software).

3.1.7. Mantenibilidad. Representa la capacidad del producto para ser modificado de forma efectiva y eficiente debido a necesidades evolutivas, correctivas o perfectivas.

Se divide en cinco subcaracterísticas: modularidad, reusabilidad, analizabilidad, modificabilidad y capacidad de testeo. La modularidad trata la capacidad de un sistema o software (formado por componentes discretos) de generar un mínimo impacto sobre los componentes cuando se realiza una modificación sobre uno de ellos. La reusabilidad refiere a la habilidad de un componente para ser utilizado en más de un sistema de software o en la construcción de otros componentes. La analizabilidad apunta a la facilidad con la cual se puede evaluar el impacto de un cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos y/o identificar las partes a modificar ante una eventual reformulación de los componentes. La modificabilidad referencia la habilidad del producto para ser modificado de forma efectiva y eficiente sin introducir defectos o degradar su desempeño actual. Finalmente, la capacidad de testeo representa la facilidad con la cual es posible establecer criterios de prueba (tanto para un sistema como para un componente) junto con la simplicidad que existe para llevar a cabo estas pruebas a fin de determinar si se cumplen o no los criterios establecidos con anterioridad.

3.1.8. Portabilidad. Es la capacidad del producto o componente de software de ser transferido, de forma efectiva y eficiente, de un entorno de ejecución a otro entorno diferente (ya sea en relación al hardware y/o al software).

Se divide en tres subcaracterísticas: adaptabilidad, facilidad de instalación y capacidad de reemplazo. La adaptabilidad refiere a la capacidad del producto para acomodarse de forma efectiva y eficiente a entornos de hardware y software previamente definidos. La facilidad de instalación representa la sencillez con la cual es posible instalar/desinstalar el producto exitosamente en un entorno determinado. La capacidad de reemplazo trata la habilidad del producto para ser utilizado en lugar de otro software con el mismo propósito en el mismo entorno.

3.2. Atributos

De acuerdo con la descripción precedente, el modelo de calidad del estándar ISO/IEC 25010 define un conjunto de características junto con sus subcaracterísticas asociadas. Los atributos (correspondientes al nivel inferior de la jerarquía) no forman parte de este esquema. Esto se debe a que los mismos varían en función del producto de software sobre el cual se esté analizando la calidad.

De esta manera, el estándar provee un marco de trabajo para que las organizaciones definan el modelo de calidad aplicable a un producto de software particular. La especificación precisa de este modelo (junto con los atributos de interés aplicables al mismo) queda a cargo de esta entidad.

4. Criterios de Simulación

Simular es el proceso de diseñar un modelo de un sistema real y realizar experimentos sobre el mismo a fin de entender su comportamiento y/o evaluar estrategias que mejoren su operación [19]. Es una técnica ampliamente utilizada en ramas como la química, física, y eléctrica con el objetivo de estudiar el comportamiento de componentes ante determinadas situaciones y/o de evaluar la capacidad de respuesta de un sistema en un momento dado.

En la actualidad, el ámbito de la ingeniería de software está buscando una forma de aplicar este tipo de técnicas sobre los productos intermedios del proceso de desarrollo a fin de predecir la calidad del sistema resultante. En estos casos, el costo de realizar una corrida de simulación es siempre menor a los costos asociados a la obtención de un producto defectuoso. Por este motivo, aunque el producto intermedio no refleje la totalidad del sistema, el uso de técnicas de simulación brinda una buena aproximación a la calidad en la que resultaría el producto final. En este sentido, es importante determinar cuáles aspectos asociados a la calidad del software son susceptibles de ser evaluados por medio de simulación. La Tabla 2 resume un conjunto de criterios establecidos a fin de clasificar tales características.

Tabla 2. Resumen de los criterios utilizados.

LINEAMIENTO	
N°	*Enunciado*
1	Sólo es posible simular aspectos observables en tiempo de ejecución.
2	Al momento de realizar la simulación toda la información asociada a la derivación del comportamiento evaluado debe estar disponible.
3	La evaluación del atributo debe ser cuantitativa.

Un primer criterio para facilitar la clasificación implica que "sólo es posible simular aspectos observables en tiempo de ejecución". En este sentido, tales propiedades establecen una relación entre el estado interno del sistema y la respuesta resultante. Si esta respuesta condiciona la medición del atributo, se lo considera dinámico. Un segundo criterio aplicable sobre el precedente establece que "al momento de realizar la simulación toda la información asociada a la derivación del comportamiento evaluado debe estar disponible". Esto implica que, de no contarse con todos los datos necesarios para realizar la medición del atributo, no será posible obtener un resultado fiable del proceso de simulación. Es importante destacar que no es necesario que toda la información se encuentre explícitamente declarada, sino que la misma puede obtenerse de resultados de simulaciones precedentes. Otro criterio válido implica que "la evaluación del atributo debe ser cuantitativa". Si la medición es subjetiva a la apreciación de los stakeholders, no será posible predecir el atributo por medio de simulación.

5. Taxonomía

Tomando como punto de partida el modelo presentado en el estándar ISO/IEC 25010, se propone a continuación una calificación de sus características que respeta los criterios detallados en la sección precedente a fin de evaluar el diseño arquitectónico. Las subcaracterísticas fueron tenidas en cuenta al momento de realizar la clasificación con el objetivo de determinar la pertenencia de cada una de las características a la taxonomía propuesta.

Debido a que sólo es posible simular aspectos observables en tiempo de ejecución (criterio 1), en primera instancia se realizó una división de las características según su correspondencia a propiedades estáticas o dinámicas. La Tabla 3 muestra la clasificación resultante.

Tabla 3. Evolución temporal de las características del estándar ISO/IEC 25010.

EVOLUCIÓN TEMPORAL	
Dinámico	*Estático*
Funcionalidad	Mantenibilidad
Rendimiento	Portabilidad
Compatibilidad	
Usabilidad	
Confiabilidad	
Seguridad	

Como puede observarse, la mantenibilidad y la portabilidad son las únicas características cuya evolución temporal corresponde a un esquema estático. Esto se debe a que podrán evidenciarse sin la necesidad de ejecutar el sistema. La medición de este tipo de propiedades no guarda relación con estados internos que puedan alterar los resultados obtenidos. En contraposición, resulta evidente que la funcionalidad de un sistema debe ser evaluada en tiempo de ejecución. Lo mismo ocurre con la confiabilidad, compatibilidad, seguridad, usabilidad y rendimiento. Esto se debe a que la evaluación de estos atributos requiere de una valoración del comportamiento del sistema ante estímulos específicos. Al encontrarse una dependencia entre el atributo y la respuesta del sistema, las características se convierten en dinámicas.

Una aplicación de los criterios 2 y 3 sobre los aspectos dinámicos identificados con anterioridad arrojó como resultado la clasificación presentada en la Tabla 4.

En relación a la arquitectura de software la funcionalidad, compatibilidad y usabilidad no son características predecibles por medio de simulación. La información necesaria para evaluar la funcionalidad conforme a las subcaracterísticas expuestas en 3.1.1 no se encuentra disponible a este nivel de diseño (viola el criterio 2), requiriendo en algunos casos una evaluación cualitativa (infringiendo el criterio 3). La medición de la compatibilidad por medio de simulación presenta el mismo problema (violación del segundo criterio). Esto se debe a que requiere simular el comportamiento de los sistemas vinculados, lo cual constituye información desconocida en la mayoría de los casos. Aun cuando se tuviese la certeza de que el comportamiento de los otros sistemas está reflejado en un modelo de simulación apropiado, la fiabilidad de los resultados obtenidos podría estar en duda. Finalmente, la usabilidad viola el tercer criterio ya que requiere de una comprobación cualitativa respecto al usuario final.

Tabla 4. Taxonomía de características según posibilidad de predecirse por simulación de arquitectura.

ARQUITECTURA DE SOFTWARE	
Predecible por Simulación	*No predecible por Simulación*
Rendimiento	Funcionalidad
Confiabilidad	Compatibilidad
Seguridad	Usabilidad

Por otro lado, se observa en la Tabla 4 que tanto el rendimiento como la confiabilidad y la seguridad son características simulables. Las tres propiedades cumplen tanto con el criterio 2 como con el 3, debido a que la información necesaria para su derivación existe en el momento en el cual se desarrolla la arquitectura y sus evaluaciones son cuantitativas. Puntualmente, en el caso del rendimiento el uso de una técnica de simulación requerirá de la especificación de condiciones de ejecución concretas. La información asociada a estas condiciones podría plantear inconvenientes en aquellos casos en los cuales refiera a parámetros específicos asociados a los equipos de hardware (sobre los cuales correrá la arquitectura) y/o a la conectividad de la infraestructura.

5.1. Características Complementarias

El MC del estándar ISO/IEC 25010 se corresponde con un modelo mixto. Esto implica que consta de un conjunto de factores de calidad abstractos que son reutilizables en múltiples proyectos, pudiendo ser refinados para un proyecto particular. De esta manera es posible incorporar características y subcaracterísticas complementarias (de interés para el producto de software bajo evaluación) sobre la clasificación presentada en la Tabla 1.

Teniendo en cuenta que la taxonomía presentada en esta sección toma como base dicho modelo, es posible afirmar que sobre la misma podrán realizarse modificaciones siempre que se respeten los lineamientos presentados con anterioridad. Al mismo tiempo, la clasificación podría crecer de forma tal que incorporen nuevos conceptos relevantes.

6. Conclusiones y Trabajos Futuros

En etapas de desarrollo tempranas, alcanzar un nivel de calidad apropiado implica lograr un diseño arquitectónico adecuado a la naturaleza de los atributos de calidad relevantes. Para esto, es necesario seleccionar los principios de trabajo adecuados buscando un balance entre aquellas propiedades que compiten entre sí. De esta manera, las arquitecturas de software sirven como vehículo para el estudio de los atributos de calidad asociados al sistema final.

Se ha presentado una breve descripción de la serie ISO/IEC 25000, poniendo énfasis en el modelo de calidad de producto definido en una de sus normas. Este modelo de referencia representa la calidad esperada de los productos de software, planteando un conjunto de necesidades y expectativas que ayudan a una definición clara y precisa de las características a evaluar. Por estos motivos, ha sido tomado como base en la elaboración de la taxonomía propuesta.

La simulación de los atributos identificados constituye el objetivo final del trabajo en proceso. Como siguiente etapa del trabajo se espera incorporar atributos específicos al nivel inferior de la jerarquía de las subcaracterísticas identificadas. Además, se incorporaran prácticas ingenieriles que ayuden a su perfeccionamiento junto con métricas apropiadas para su medición. De esta manera, se podrá comenzar a formular un modelo de simulación aplicable a contextos específicos (como por ejemplo aplicaciones web o cloud).

Desde un punto de vista práctico, los arquitectos de software pueden utilizar los modelos de calidad existentes como plantillas para derivar modelos de calidad de sistemas específicos. Muchas veces una aplicación particular no requiere la totalidad de características incluidas en estos esquemas, por lo que el arquitecto debería preparar (como parte de la especificación de requerimientos) un modelo específico que identifique las características, subcaracterísticas, atributos y métricas aplicables. Si se tomara como referencia el modelo del estándar ISO/IEC 25000, sería

posible construir una herramienta que brinde soporte a esta tarea. En este sentido, el diseño y desarrollo de tal herramienta constituye un posible trabajo futuro. La utilización de un metamodelo de calidad como base de este trabajo sería fundamental, dando lugar a la inclusión de los criterios de simulación como mecanismo para la derivación de modelos asociados a simulaciones de productos intermedios. Una primera aproximación a este tipo de herramienta (sobre un estándar precedente, el cual actualmente no se encuentra vigente) se realizó en [20].

En conclusión, el trabajo expuesto sirve como punto de partida para la utilización de estrategias de simulación sobre productos intermedios del proceso de desarrollo a fin de evaluar la calidad del sistema resultante.

7. Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET (PIP 112-20110100906) y la Universidad Tecnológica Nacional (PID 25/O144). Se agradece el apoyo brindado por estas instituciones.

8. Referencias

[1] M. Barbacci, M. Klein, T. Longstaff, and C. Weinstock, "Quality attributes. Technical Report CMU/SEI-95-TR-021 ESC-TR-95-021, SEI." Carnegie Mellon University, Diciembre-1995.

[2] "IEEE Std. 1061-1998 IEEE Standard for a Software Quality Metrics Methodology – Description." Institute of Electrical Electronic Engineering, 1998.

[3] R. Pressman, *Software Engineering: A Practitioner´s Approach*, 7th ed. McGraw-Hill, 2010.

[4] S. Albin, *The art of software architecture: design methods and techniques.* John Wiley & Sons, 2003.

[5] ISO, "Quality management. 1º Edición." ISO, 2005.

[6] S. Wagner, *Software Product Quality Control*. Springer, 2013.

[7] K. Musa and J. Alkhateeb, "Quality Model Based on Cots Quality Attributes," presented at the International Journal of Software Engineering & Applications, 2013, vol. 4.

[8] J. Lee, "A quality model for evaluating software-as-a-service in cloud computing," presented at the 7º International Conference on Software Engineering Research, Management and Applications, 2009, pp. 261–266.

[9] R. G. Dromey, "A Model for Software Product Quality," *IEEE Trans Softw Eng*, vol. 21, no. 2, pp. 146–162, Feb. 1995.

[10] M. Kläs, C. Lampasona, and J. Münch, "Adapting Software Quality Models: Practical Challenges, Approach, and First Empirical Results," Dec. 2013.

[11] J. L. Dargan, E. Campos-Nanez, P. Fomin, and J. Wasek, "Predicting Systems Performance through Requirements Quality Attributes Model," *Procedia Comput. Sci.*, vol. 28, p. 347, Feb. 2014.

[12] R. Roshandel, N. Medvidovic, and L. Golubchik, "A Bayesian Model for Predicting Reliability of Software Systems at the Architectural Level," presented at the Lecture notes in computer science, 2007, pp. 108–126.

[13] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Addison-Wesley Professional, 2003.

[14] A. Meiappane, B. Chithra, and P. Venkataesan, "Evaluation of Software Architecture Quality Attribute for an Internet Banking System," *Int. J. Comput. Appl.*, vol. 62, p. 21, Jan. 2013.

[15] V. Bogado, S. Gonnet, and H. Leone, "Modeling and simulation of software architecture in discrete event system specification for quality evaluation," *Simulation*, vol. 90, no. 3, p. 290, Mar. 2014.

[16] J. Rech and C. Bunse, Eds., *Evaluating Performance of Software Architecture Models with the Palladio Component Model*. IGI Global, 2008.

[17] R. E. Shannon, *Systems Simulation: The Art and Science*. Englewood Cliffs, N.J: Prentice Hall, 1975.

[18] "Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. 1º edición." ISO/IEC, 2005.

[19] A. M. Law, *Simulation Modeling and Analysis*. McGraw-Hill, 1991.

[20] J. Carvallo, X. Franch, G. Grau, and C. Quer, "QM: A Tool for Building Software Quality Models," presented at the RE -International Conference-, 2004, vol. 12, pp. 358–359.

Un Modelo para Soporte de la Investigación Asistido por Técnicas de Visión Artificial

Jorge Alejandro Kamlofsky, María Lorena Bergamini
CAETI - Universidad Abierta Interamericana
Av. Montes de Oca 725 – Buenos Aires – Argentina.
{Jorge.Kamlofsky, Maria.Bergamini}@UAI.edu.ar

Resumen

Muchos trabajos de investigación de diversas disciplinas basan sus conclusiones en el resultado de aplicar algoritmos y/o técnicas de minería de datos sobre grandes volúmenes de datos. Sin embargo, el proceso de obtención de las bases de datos de la propia investigación no es una tarea trivial. En este trabajo se presenta un modelo asistido por técnicas de visión artificial que permite analizar encuestas digitales y en papel, normaliza la carga de los resultados a la base de datos, posibilita su guarda en un único almacén, permitiendo y facilitando el posterior análisis integral y extracción de conocimiento subyacente mediante la implementación de algoritmos o técnicas de minería de datos.

1. Introducción

1.1. Trabajos relacionados

El corazón de gran parte de los trabajos de investigación consiste en el análisis de un gran caudal de datos obtenidos. Para ello, se suelen aplicar algoritmos o técnicas de minería de datos, analizar propiedades estadísticas o indicadores de diversos parámetros que forman parte de un conjunto de datos. Muchos bancos de datos son públicos y accesibles gratuitamente, mientras que otros son pagos. En general, presentan gran variedad, cantidad y calidad de datos. Por ejemplo, muchas investigaciones biomédicas, parten de búsquedas en el banco de datos médicos MedlinePlus[1] producido por la biblioteca Nacional de Medicina de EEUU [1]. Sin embargo, cuando el investigador decide encarar un nuevo proyecto, basado en ideas innovadoras, los bancos públicos de datos muchas veces no coinciden con las necesidades del investigador. En ese momento, el investigador decide recurrir a la generación de datos a los fines de satisfacer las necesidades de información del trabajo de investigación. Y se encuentra con que esto no es un trabajo trivial. Es clave el diseño de la investigación y el diseño del modelo de datos, a fines de poder ser usado y analizado mediante las técnicas disponibles.

Durante el diseño de la investigación, una tarea requerida es la definición del universo de estudio o población. A partir de ello se puede definir el tamaño de la muestra del experimento. En ese momento, surge el interrogante acerca de cómo obtener datos no sesgados, con el tamaño de la muestra requerido. Puede hacerse uso de herramientas de marketing, útiles a muchas disciplinas. Se han usado herramientas de marketing, por ejemplo, para mejorar la eficiencia de la salud pública, presentadas en [2]. Una de ellas es el marketing directo. El marketing directo es un sistema interactivo de marketing que utiliza uno o más medios publicitarios para conseguir una respuesta medible [3]. De este modo, es posible la obtención de datos relevantes, a escala necesaria.

Realizar encuestas web enviadas por correo electrónico a bases de datos de población segmentada útil a los fines del investigador, es una opción muy atractiva ya que soluciona gran parte de los inconvenientes mencionados previamente: Población sin sesgo, posibilidad de envío de la encuesta a una gran cantidad de destinatarios, y carga efectiva de las respuestas. Sin embargo, enviar gran cantidad de correos electrónicos no asegura obtener gran cantidad de respuestas. Mientras más segmentadas las bases, menor la cantidad efectiva de correos electrónicos disponibles. Pero si el tema de la encuesta está dentro del interés del segmento de la muestra, la proporción de respuestas suele ser mayor.

La eficacia del uso del correo electrónico en campañas de marketing fue ampliamente tratado en algunos trabajos [4]. La correcta segmentación de la lista de

[1] http://www.nlm.nih.gov/medlineplus/

correos electrónicos de los destinatarios, así como la pertinencia del mensaje aumenta la eficacia de respuesta de las campañas [5]. También se ha constatado que los consumidores suelen aceptar los mensajes que tienen contenidos relevantes para ellos [6]. La calidad técnica y la vistosidad gráfica del mensaje, dependen también de la cantidad y calidad de los elementos gráficos del arte final. Según el estudio realizado por Chittenden [5] el número de imágenes mejora las tasas de respuesta de las campañas de correo electrónico. El remitente y el asunto es lo primero que se ve en la bandeja de entrada. De ello depende que el mensaje se abra o no. En otros [7] se realizan recomendaciones acerca del correcto diseño y formato de e-mails que optimizan su aceptación. Los costos, características, ventajas y desventajas de los principales servicios más conocidos de envío masivo de correos electrónicos se presentan en un trabajo de Manchón y Fernandez [8].

Y luego surge un nuevo interrogante: ¿cómo cargar todo ese volumen de datos en una base de datos?

El uso de encuestas o formularios para la recolección de datos vía web ofrece una serie de ventajas por sobre las técnicas tradicionales (formularios de papel): ofrece menor dependencia en personal y equipamiento de ingreso de datos. Otra ventaja es la universalidad tecnológica: no requiere ninguna aplicación extra más allá del navegador de Internet, presentes en todos los dispositivos electrónicos de conexión a la web (PCs, tablets, celulares, etc.), con cualquier sistema operativo. También tiene bajo costo de implementación y permite la carga directa de las respuestas en una base de datos, listas para su posterior análisis por medio del uso de programas estadísticos. El formato web de las encuestas, además, incrementa la confiabilidad de los datos ingresados, disminuyendo la pérdida de datos por encuestas incompletas o datos erróneos, ya que permite el desarrollo de herramientas de control de campos de carga obligatorios y validación del tipo de dato ingresado, evitando errores de tipeo [9].

Experiencias más recientes muestran que se ha aumentado la tasa de respuestas de las encuestas web siendo ésta similar a la tasa de respuesta de las encuestas en papel. En parte, esto puede deberse a que el uso de Internet se ha hecho parte de la vida cotidiana permitiendo que esta no tan nueva tecnología se pueda utilizar para la recolección de datos vía web [9].

Los métodos tradicionales de recolección de datos basados exclusivamente en el papel tienen limitaciones inherentes y riesgos como la pérdida de datos entre su recolección y carga. Cualquier recolección de datos se vuelca luego a un formato de datos electrónicos: planilla de cálculo o base de datos, para su posterior utilización y análisis. Gracias a que el error humano de carga es inevitable, se necesita realizar una doble carga para la validación de los datos, lo que aumenta los costos de un proyecto de investigación [10]. La recolección de datos históricamente ha sido realizada por formularios pre-impresos, entrevistas telefónicas o personales. Esta metodología ha demostrado ser costosa y consume mucho tiempo, pudiendo condicionar la realización de la investigación. Por otro lado el tiempo entre la distribución del cuestionario y el análisis estadístico puede ser enlentecido por la necesidad de carga de datos posteriores [11].

A pesar de lo anterior, ésta suele ser la mejor (y muchas veces la única) opción en situaciones donde hay gran cantidad de potenciales encuestados (gracias a lo cual hay grandes posibilidades de obtener buena cantidad de datos), pero a su vez, no es posible presentar un formulario electrónico en un kiosco o terminal que contenga la encuesta. Podemos mencionar: salas de espera de centros médicos u oficinas públicas, estaciones de transporte público, bancos y entidades financieras, ingreso a centros educativos, etc.

Cargar automáticamente los datos provenientes de estos "formularios analógicos" soluciona los problemas relativos al tiempo, eficacia y costo de la carga de las respuestas de los encuestados al sistema. Usando técnicas de procesamiento de imágenes y visión artificial descriptas en [12], [13] y [14] entre otras, se pueden digitalizar las respuestas que los encuestados transcriben en los formularios de papel.

En este trabajo se presenta un modelo que permite obtener un conjunto de datos normalizados y con el formato adecuado para su análisis partiendo de datos provenientes de distintos orígenes: formularios electrónicos, y formularios en papel. Los distintos tipos de formulario se cargan a una única base de datos de transacciones (o sistemas OLTP: procesamiento de transacciones en línea) siguiendo el modelo relacional hoy vigente presentado en [15]. A los fines de optimizar su funcionamiento, la base se construye siguiendo las tres formas normales [16] lo que permite acceso a datos en forma rápida y estructurada. Los sistemas OLTP son sistemas de origen para los almacenes de datos o data warehouse (OLAP). Un data warehouse es un repositorio de datos históricos, integrados, especialmente organizados y referidos a un tema. Usar esta tecnología permite organizar los datos facilitando que los mismos se transformen en información. Es la única forma de evitar ser rico en datos y pobre en información [17]. Sobre el almacén de datos pueden implementarse algoritmos de minería de datos [18] y así obtenerse conocimiento nuevo. En [17] se presenta un modelo de evolución del soporte de decisión basado en cuatro categorías:

- Las consultas estándar: Es el método de análisis más difundido.
- El análisis multidimensional: Ofrece diferentes perspectivas de los datos mediante el uso de

dimensiones o puntos de vista como ser: tiempo, ubicación y producto.
- Segmentación: Permite dividir un conjunto de datos en segmentos. Luego, cada segmento puede analizárselo como dimensión.
- Descubrimiento del conocimiento: Usando poderosas técnicas y algoritmos se pueden encontrar patrones que permitan obtener conocimiento nuevo, oculto dentro de los datos.

1.2. Motivación y alcance

Este trabajo surgió como respuesta a diversos requerimientos, principalmente provenientes de organizaciones o instituciones relacionadas con la investigación médica entre las que se pueden mencionar: el sector de "Políticas de Salud" de la Sociedad Argentina de Cardiología y el CAESIS (Centro de Altos Estudios en Ciencias de la Salud) dependiente de la Universidad Abierta Interamericana.

Mientras que por un lado se busca la informatización de encuestas usando la tecnología de Internet para la recolección de datos, por otro lado, frecuentemente se presentan situaciones donde hay gran cantidad de posibles encuestados y no se dispone de infraestructura tecnológica acorde. En esos casos, resulta ideal disponer de formularios impresos en papel.

La integración eficaz de encuestas en papel y de origen electrónico dentro de un mismo modelo de datos permite disponer de mayor cantidad de datos, útiles no sólo para investigaciones médicas sino para investigaciones de otras disciplinas.

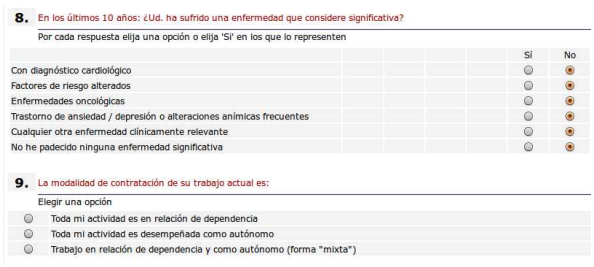
1.3. Esquema del trabajo

En la sección 2 presenta los detalles técnicos del sistema de software y del modelo de datos. La sección 3 muestra la implementación de las técnicas de procesamiento de imágenes y visión artificial para la carga automática de datos provenientes del papel al modelo de datos. La sección 4 presenta la implementación de un caso de estudio y sus resultados. Finalmente, en la sección 5 se presentan las conclusiones.

2. Descripción del Sistema

2.1. Presentación general del sistema

Consiste en un sistema de encuestas que permite la recolección de datos a partir de formularios analógicos (en papel) y digitales (en formularios web), concentra la carga en una base de datos única que permite el análisis posterior, unificado. Cada formulario consta de un conjunto de preguntas y opciones de respuesta. La figura 1 muestra un tramo de una encuesta en formato digital y en papel.



(a)

8 ¿Sufrió alguna enfermedad significativa en los últimos 10 años?		
☐ Con diagnóstico cardiológico		☐ Factores de riesgo alterados
☐ Enfermedades oncológicas		☐ Otra enfermedad relevante
☐ Trastorno de ansiedad /depresión o alteraciones anímicas frecuentes		☐ Ninguna enfermedad relevante
9 Su modalidad de contratación:		
☐ Rel. de dependencia	☐ Autónomo	☐ Mixto: rel. dependencia y autónomo

(b)

Figura 1: (a) Una porción de una encuesta en formato web. (b) La misma porción de la encuesta en formato papel.

En la figura 1 puede observarse que las mismas opciones presentes en el formulario electrónico lo están también en el formulario en papel.

Con los formularios web, la carga al sistema se produce en el momento de presionar el botón "Submit" del formulario. Con los formularios en papel, se escanea la totalidad de los formularios, generándose un archivo de imagen por encuestado-respuesta. La carga al sistema OLTP se produce luego del análisis de cada una de las imágenes con herramientas de procesamiento de imágenes.

Con origen en el sistema OLTP, el proceso de ETL (Extract, Transform, Load) permite que los datos se guarden adecuadamente en el almacén de datos des-normalizado: el data-mart, pequeño datawarehouse o cubo OLAP. Sobre estos almacenes o estructuras de almacenamiento de datos se puede realizar cómodamente el análisis de datos usando una amplia diversidad de las técnicas de diversa complejidad: desde simples consultas, hasta complejas técnicas de minería de datos.

2.2. El sistema OLTP: el modelo transaccional de datos

La figura 2 muestra el diseño lógico mediante un diagrama *patas de gallo* del modelo transaccional de los datos u OLTP. En el mismo, se observan las tablas de “preguntas”, de “respuestas” y del “encuestado”.

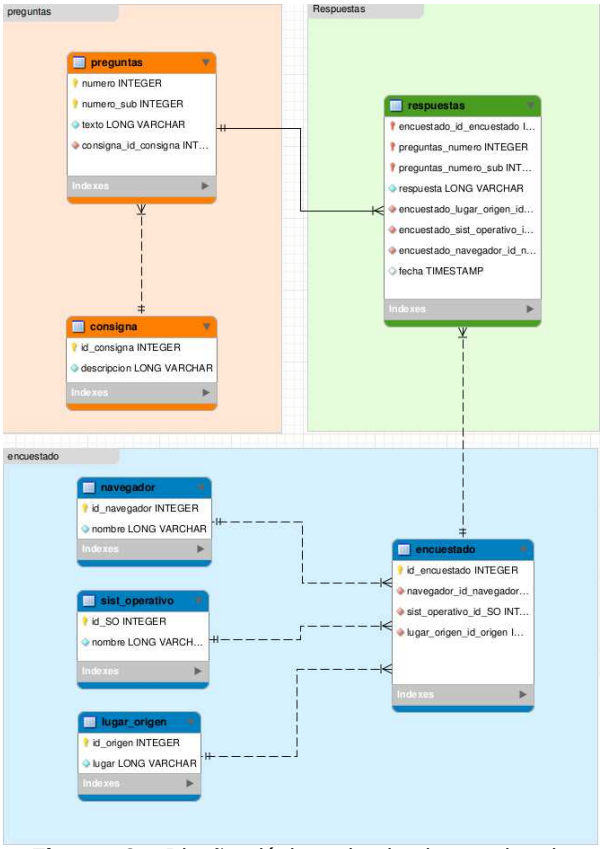


Figura 2: Diseño lógico de la base de datos transaccional.

En las respuestas obtenidas por web, el navegador y el sistema operativo para la tabla “encuestado” se obtienen con $_SERVER['HTTP_USER_AGENT'] de php. El lugar de origen se obtiene con $_SERVER['REMOTE_ADDR']. En caso de respuestas en papel, el lugar de origen se hace referencia al lugar donde se realizó la encuesta. La tabla “respuestas” incluye fecha y hora realizada la respuesta.

2.3. El sistema OLAP: El data-mart

La tecnología de data warehousing facilita el análisis muti-dimensional. Permite obtener en línea respuestas a consultas complejas. Estos sistemas son llamados OLAP (del inglés: procesamiento analítico en línea).

Un data-mart es un data-warehouse más pequeño, orientado a un tema. Son también estructuras multi-dimensionales. Una estructura multi-dimensional consiste en una gran tabla con columnas, donde cada columna se corresponde con una dimensión. Se obtienen a partir de los sistemas OLTP luego de realizarse operaciones ETL.

La figura 3 muestra un ejemplo una estructura de datos multidimensional típica en data-marts llamada "estrella".

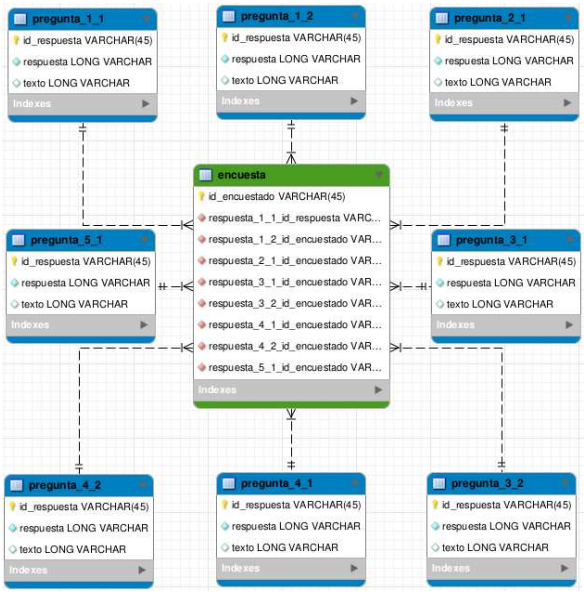


Figura 3: Una configuración de datos "estrella".

Este es un ejemplo de un data-mart de un sistema de encuestas. El núcleo o centro de la estrella tiene las respuestas de cada encuestado. Cada "punta de la estrella" representa una dimensión. En el sistema de encuestas, cada dimensión se corresponde con cada una de las preguntas.

También pueden usarse estructuras llamadas “copo de nieve” [17], en las que cada dimensión vuelve a ramificarse. En el caso de las encuestas, una dimensión corresponde a un tema dentro del cual se agrupan las preguntas.

3. La carga automática de datos provenientes de formularios en papel

3.1. Presentación del problema y esbozo de su solución.

El procedimiento de adquisición de datos a partir de encuestas contestadas en papel, consiste en la lectura del estado de casilleros o sectores específicos ubicados en posiciones predefinidas dentro de la imagen.

Se presenta el inconveniente que los dispositivos de adquisición automática de imágenes (escáneres) rara vez capturan a las imágenes exactamente en la misma posición y orientación, algo también observado en la manipulación imperfecta de hojas por parte de los operadores en escáneres manuales. Entonces, obtener acierto entre un casillero ubicado en un lugar predeterminado con su respectivo sector dentro de la imagen a partir de esto, sería pura coincidencia.

Para poder tener una referencia precisa de la orientación o inclinación del papel, se incluye en los formularios dos marcas en posiciones específicas. En particular, en este trabajo se ubicaron en el borde superior a una distancia fija de los bordes del papel.

Dichas marcas tienen la función de calibrar la ubicación de cada imagen escaneada dentro del marco de cada página. La calibración se realiza luego de la detección de las marcas.

La detección de las marcas se logra tras la comparación entre los patrones de forma de los distintos objetos obtenidos con el patrón de las marcas. Tras la detección, se obtiene la ubicación de las marcas dentro de la imagen adquirida, y con ello, se calcula el desplazamiento y cambio de orientación.

3.2. La detección de las marcas: uso de técnicas de visión artificial

La visión artificial comprende técnicas de procesamiento de imágenes para identificar un objeto dentro de una imagen digital a partir de patrones característicos. Existen distintas técnicas que permiten separar a los objetos del fondo de una imagen [12]. En este trabajo se utiliza el método de segmentación por umbralizado que asigna un valor booleano a los píxeles de acuerdo a si su valor de gris supera o no a un número definido como umbral.

Luego de separados los objetos del fondo, se pretende hallar patrones característicos de los objetos. Se destacan dos estrategias para el análisis de patrones: el análisis de sus bordes o de toda la región que comprende al objeto. El borde de una figura es de gran interés en el análisis de objetos dentro de imágenes, ya que muchas características geométricas (convexidad, tamaño, agujeros, etc.) pueden estudiarse analizando su borde. Además, el borde conforma la frontera entre el objeto digital y el fondo de la imagen. Un algoritmo simple para la obtención del borde de un objeto digital fue presentado por A. Rozenfeld en [19].

El proceso de adquisición y digitalización de una imagen introduce ruido que tiene evidente influencia en los bordes dificultando su estudio. La poligonalización de bordes reduce sus efectos y permite disminuir notoriamente la cantidad de puntos sin perder aspectos relevantes de la forma [20].

En [13] se presentó un método de representación de formas a fin de ser aplicado para el reconocimiento confiable de objetos, en forma eficiente: *el patrón de giro*. Se basa en aproximar inicialmente el borde de un objeto por un polígono y obtener luego una representación de la evolución de curvatura y concavidad del borde rectificado a lo largo de la curva.

Se considera una curva poligonal de n vértices ordenados $v_1, v_2, \ldots, v_n$. Sea $\kappa(i)$ el giro del polígono en el vértice v_i: el ángulo formado por los segmentos orientados $v_{i-1}v_i$ y v_iv_{i+1}, se define $\kappa_{acum}(i)$ al ángulo de giro acumulado desde el primer vértice hasta el vértice i.

Claramente, $\kappa_{acum}(n) = -2\pi$ si se recorre el polígono en sentido horario. Sea $l(i)$ la suma de las longitudes de los segmentos del polígono desde v_1v_2 hasta v_iv_{i+1} y L la longitud total del polígono. Entonces, $\lambda(i) = l(i)/L$ es la longitud normalizada. El patrón de giro de un polígono es la curva lineal por tramos en el plano longitud de arco-curvatura l-κ, cuyos vértices son $\beta(i) = (\lambda(i), \kappa_{acum}(i))$, siendo El punto inicial de esta curva se toma con $\lambda(0) = 0$, $\kappa_{acum}(0) = 0$. También puede pensarse esta curva como la gráfica lineal por tramos de la función $\kappa_{acum}(\lambda)$ con $\lambda \in [0,1]$, donde ocurre que $\kappa_{acum}(0) = 0$ y $\kappa_{acum}(1) = -2\pi$.

El análisis de la orientación de las formas, permite identificar puntos de inicio adecuados. Así, la representación constituye un patrón que describe la forma, independientemente de la posición, rotación y escala.

Basándose en esta representación normalizada se define una distancia entre formas. Llamamos distancia entre dos patrones κ_1 y κ_2 al valor:

$$D(\kappa_1, \kappa_2) = \int_0^1 (\kappa_1(\lambda) - \kappa_2(\lambda))^2 \, d\lambda$$

Encontrar un objeto (llamado "modelo") dentro de una imagen, entonces, consiste en comparar sus patrones de curvatura y medir la distancia entre ellos y el patrón modelo. La localización es positiva si la distancia obtenida no supera un umbral predefinido.

La figura 4 muestra un ejemplo del patrón de giro aplicado a la identificación de dos marcas dentro de una imagen digital. El patrón modelo de las marcas se encuentra almacenado en la base de datos del sistema. Para identificar a un objeto de la figura 4 (a) como uno de los elementos almacenados, se mide uno a uno la distancia entre el patrón modelo y el patrón de cada objeto de la imagen. Su resultado se lo compara con un umbral predefinido. Y se habrá reconocido al objeto, si la distancia medida está por debajo del umbral. En la figura 4 (b), se observan los patrones de giro de todos los objetos. En particular, los patrones en trazo negro continuo indican que se ha identificado al objeto modelo entre los distintos objetos.

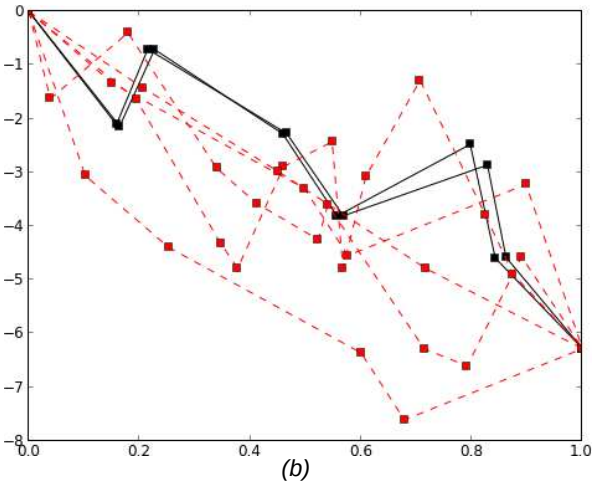
3.3. La calibración de la imagen: re-ubicación de los píxeles de la imagen

Mediante una aplicación de reconocimiento de formas se detecta la existencia de las formas de las marcas dentro de la imagen. Se devuelve también, la ubicación de dichas marcas. Luego, mediante comparación con la ubicación de cada marca en la imagen modelo, puede

obtenerse el vector de traslación, el factor de escala y el ángulo de rotación con los que el escáner transformó la imagen.



(a)



(b)

Figura 4: (a) Una imagen conteniendo dos marcas: flechas hacia arriba, y un texto. (b) Los patrones de todos los objetos. Entre ellos, en negro, los patrones de las marcas.

Sean las marcas A y B. Sean $A_0 = (x_{A0}, y_{A0})$ y $B_0 = (x_{B0}, y_{B0})$ las coordenadas del punto extremo superior de cada marca en la imagen modelo y $A_1 = (x_{A1}, y_{A1})$ y $B_1 = (x_{B1}, y_{B1})$ sus correspondientes ubicaciones en la imagen escaneada. Sean t el vector de traslación, e el factor de escala y α el ángulo de rotación. t se obtiene haciendo: $t = (x_{A1} - x_{A0}, y_{A1} - y_{A0})$. e se calcula mediante el cociente de las distancias euclídeas entre las marcas: $e = dist(A_1, B_1) / dist(A_0, B_0)$. α es el ángulo de inclinación de la recta que une los puntos extremos de las marcas en la imagen escaneada, pues las marcas en la imagen modelo se encuentran alineadas a 0º. Se puede entonces calcular $\alpha = \text{arctg}((y_{B1} - y_{A1})/(x_{B1} - x_{A1}))$.

El vector de traslación t, el factor de escala e y el ángulo de rotación α obtenidos caracterizan la transformación rígida hecha durante el escaneo de la imagen. La calibración de la imagen consistirá entonces en revertir estas transformaciones no deseadas realizando las transformaciones inversas a las realizadas durante el proceso de escaneo de la hoja de respuestas.

En el texto de Lengyel [21], se presentan varias herramientas matemáticas útiles para realizar esta tarea. Entre ellas, se eligió el uso de coordenadas homogéneas, que permiten realizar las operaciones de escalado, rotación y traslación simultáneamente luego de calcular el producto matricial entre la matriz X' de todos los puntos (expresados como vectores fila extendidos hasta la coordenada w) y la matriz de la transformación inversa F. Siendo $F = \{((1/e)\cos\alpha, -(1/e)\text{sen}\alpha, 0, 0); ((1/e)\text{sen}\alpha, (1/e)\cos\alpha, 0, 0); (0, 0, 1, 0); (x_{A0} - x_{A1}, y_{A0} - y_{A1}, 0, 1)\}$, la matriz X de los puntos de la imagen calibrados puede obtenerse calculando $X = X'.F$

Hay casos donde la calibración no puede realizarse: cuando la transformación realizada en el proceso de escaneo es tal que alguna respuesta queda fuera del área de adquisición. Esto se puede calcular conociendo el ángulo de rotación y el vector de traslación, y es dependiente del diseño de la encuesta.

3.4. Lectura de resultados: transferencia de las respuestas desde el papel a la base de datos

Responder a una pregunta de las encuestas en papel consiste en marcar con tinta, o grafito espacios destinados a tal fin. En cada encuesta, se leen estos espacios y se determina si están marcados o no.

Formalmente, sean $R_i = (x_i, y_i)$ las coordenadas del centro de la casilla de la respuesta i en la imagen modelo. Se asocia a esa casilla una ventana digital dada por $D_i = [x_i-\gamma, x_i+\gamma\ ]x[\ y_i-\delta, y_i+\delta]$, donde 2γ es el ancho y 2δ es el alto de la ventana, elegidos de acuerdo al diseño gráfico de las encuestas.

Para cada respuesta, se define el nivel de la ventana asociada: $N_{Di} = \sum_{j=1}^{4.\gamma.\delta}(v_j)$, donde v_j es el valor del nivel de gris leído por el escáner en el pixel p_i.

Para leer las respuestas, se leen los niveles de las ventanas de todas las respuestas, sobre la imagen ya normalizada (transformada según se explicó en la sección anterior). Si el nivel de la ventana supera un umbral predefinido, la respuesta no fue seleccionada por el encuestado. Y viceversa: un nivel por debajo del umbral indica que la respuesta fue seleccionada (en imágenes RGB Monocromáticas: Negro=0 y Blanco=1). Luego, se incorpora dicha respuesta al sistema transaccional.

4. Caso de Estudio.

4.1. Presentación

En Octubre de 2013, el sector "Políticas de Salud" de la Sociedad Argentina de Cardiología lanzó una encuesta a sus socios y clientes (mayormente cardiólogos), con la

finalidad de conocer la situación de los cardiólogos en Argentina [22].

La encuesta se realizó en formato web y se enviaron invitaciones a socios y a quienes sin ser socios, hubieran realizado capacitaciones en la sociedad. Las invitaciones se incluyeron en el newsletter semanal en cuatro oportunidades desde su lanzamiento hasta Marzo de 2014; enviándose aproximadamente 6500 correos electrónicos a su base de clientes (socios y no socios) en cada oportunidad.

Durante los días 16 y 17 de Mayo de 2014 se realizó en la ciudad de Junín el I Congreso Multidisciplinario de Cardiología [23]. En dicho evento se reunió a aproximadamente 700 personas. Para ello (sin conocer a-priori la cantidad de asistentes) se imprimieron 400 ejemplares de la encuesta en papel.

4.2. Tecnología utilizada

La aplicación web de la encuesta y las herramientas de ETL se desarrollaron en PHP[2]. El motor de la base de datos usada es Mysql[3]. Para el modelado de la base de datos se usó Mysql wordbench[4].

La encuesta en papel se transcribió en un procesador de textos con la consigna que la totalidad de la encuesta debe ocupar una hoja. Se imprimió en doble faz: en el frente, el título de la encuesta y el objeto de la misma. A continuación, las instrucciones. En la página siguiente, se colocó la totalidad de las preguntas y sus opciones. Para escanear los documentos en papel se usó un escáner Epson[5] con alimentador automático de documentos (ADF en inglés). La aplicación para tratamiento de los archivos de imagen se desarrolló en Python[6].

4.3. Resultados

4.3.1. Acerca del origen de las respuestas. Al día 30 de Mayo de 2014 se cuentan 236 respuestas por web y 28 respuestas en papel.

Las 236 respuestas hechas en la web fueron resultado de aproximadamente 26000 correos electrónicos no nominados enviados desde Octubre de 2013. Esto da un ratio de 0,9% respuestas por persona-envío.

Las 28 respuestas en papel se obtuvieron en un único evento donde participaron aproximadamente 700 personas. Lo cual arroja un ratio de 4% de respuestas por persona-asistente.

2 http://php.net/
3 http://www.mysql.com/
4 http://www.mysql.com/products/workbench/
5 http://www.epson.com/
6 https://www.python.org/

4.3.2. Las respuestas y el modelo de evolución. La implementación del modelo detallado en este trabajo aplicado al caso de estudio, ha permitido extraer información relevante a partir de los datos. Aquí se presentan algunas respuestas obtenidas siguiendo el modelo de evolución del soporte de decisión de cuatro categorías presentado en [17].

Consultas estándar:

Al 90% de los cardiólogos les preocupan los posibles juicios por mala praxis. La percepción de los encuestados acerca de su situación actual: Muy mala 15%, Mala 32%, Regular 31%, Buena 17%, Muy buena 4%.

Análisis multidimensional:

Desagregación de la sub-dimensión "Cardiólogos" (93% del total) de la dimensión "Especialidad médica" según Subespecialidad.

Unidad Coronaria: 31%, Ecodoppler Cardíaco - Vascular: 24%, Ninguna: 20%, Cardiología del Deporte: 7%, Electrofisiología: 6%, Hemodinamia: 5%, Investigación: 4%, Medicina Nuclear: 2%, Electrocardiografía: 1%, Otra: 0%.

Segmentación:

Segmentación de la dimensión "Género": Masculino 75%, Femenino 25%.

Segmentación de la dimensión "Lugar donde desempeña sus actividades": Ciudad de Buenos Aires y Conurbano 58%, resto de provincia de Buenos Aires 15%, interior de Argentina 26%, fuera de Argentina 1%.

Minería de datos:

Modelo de clasificación mediante árboles de decisión "ADTrees":

```
=== Run information ===
Scheme:weka.classifiers.trees.ADTree -B 10 -E -3
 [list of attributes omitted]
=== Classifier model (full training set) ===
Alternating decision tree:
: -0.9
|  (1)perc_su_situac_retiro_mala = 0: 0.219
|  |  (2)perc_su_situac_retiro_regular = 0: 0.467
|  |  |  (3)perc_su_situac_retiro_buena = 0: 0.839
|  |  |  |  (4)perc_su_situac_retiro_muy_buena = 0: 3.572
|  |  |  |  (4)perc_su_situac_retiro_muy_buena = 1: -1.914
|  |  |  (3)perc_su_situac_retiro_buena = 1: -1.545
|  |  (2)perc_su_situac_retiro_regular = 1: -1.635
|  (1)perc_su_situac_retiro_mala = 1: -1.54
|  (5)subesp_Electrofisiologia = 0: -0.827
|  (5)subesp_Electrofisiologia = 1: 0.111
|  (6)act.prof_H-privado_>60hs = 0: -0.507
|  |  (7)perc_su_situac_10anos_mala = 0: -0.438
|  |  (7)perc_su_situac_10anos_mala = 1: 0.024
|  (6)act.prof_H-privado_>60hs = 1: 0.126
Legend: -ve = 0, +ve = 1
Tree size (total number of nodes): 22
Leaves (number of predictor nodes): 15
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances         100       %
Incorrectly Classified Instances         0       %
=== Confusion Matrix ===
   a   b   <-- classified as
 132   0 |   a = 0
   0  21 |   b = 1
```

Figura 5: Implementación del árbol ADTree sobre un atributo de la encuesta.

En la figura 5 se muestra una impresión de pantalla de los resultados arrojados por el algoritmo ADTree

implementado en Weka[7] sobre el nodo "Percepción de su situación el momento de su retiro: Muy Mala".

5. Conclusiones y Trabajos Futuros

La automatización de la carga de encuestas en papel usando reconocimiento automático de imágenes es una respuesta rápida y económica frente a situaciones en las que no se dispone de infraestructura para su implementación electrónica.

La carga de las respuestas en un modelo de datos coherente con un sistema de encuestas electrónico facilita su integración y su posterior análisis.

Los acelerados avances en el campo de visión artificial y reconocimiento de imágenes permiten disponer de más y mejores prestaciones que pueden solucionar una mayor diversidad de requerimientos. Y aquí se presenta un nuevo caso. Los investigadores pueden tener opciones rápidas y económicas para obtener datos frente a distintas situaciones, dándoles flexibilidad en el momento de diseño de su investigación.

Futuros trabajos podrán mostrar la interpretación automática de textos manuscritos que podrán incluirse en cajas de texto y permitir la obtención de más información de los encuestados.

Agradecimientos

Se agradece a la Sociedad Argentina de Cardiología por permitir hacer uso de una de sus encuestas la cual fue parcialmente tomada como caso de estudio de este trabajo. El agradecimiento se extiende especialmente a los doctores: Adriana Salazar, Marcelo Boscaro, Carlos Boissonnet y María Inés Sosa Liprandi, del área de "Políticas de Salud" de dicha sociedad quienes trabajaron activamente en varias etapas de la presentación del caso de estudio aquí presentado.

Referencias

[1] Ospina, E; Reveiz Herault, L y Cardona, A. "Uso de bases de datos bibliográficas por investigadores biomédicos latinoamericanos hispanoparlantes: estudio transversal". *Rev Panam Salud Pública*, 2005, vol. 17, no 4, pp. 230-236.

[2] Beerli-Palacio, A; Martín-Santana, J, y Porta, M. "El marketing como herramienta para incrementar la eficacia de los planes de salud pública". *Informe SESPAS 2008.* Gaceta Sanitaria 22, 2008, pp. 27-36.

[3] Alet, J. "*Marketing directo e integrado"*. Barcelona: Gestión 2000, 1994.

[4] Serrano Ramos, M y Muñiz Velázquez, J. "La eficacia de la publicidad directa e interactiva a través del correo electrónico". *Congreso AE-IC*, Malaga, 2010.

[5] Chittenden, L and Rettie, R. "An evaluation of email marketing and factors affecting response". *Journal of Targeting, Measurement and Analysis for Marketing*, 11 (3), 2003, pp. 203-217.

[6] Carroll, A et al. "Consumer perceptions and attitudes toward SMS advertising: recent evidence from New Zealand." *International Journal of Advertising* 26.1, 2006, pp 79-98.

[7] Asociación de Industriales del Calzado de Elche. "Estrategia de Difusión y Marketing On-Line para el Sector Calzado de la comunidad de Valencia", *PCEV*, 2011.

[8] Manchón Pedroza, P y Benítez Fernández, Y. "MAGIC SMTP". *Tesis de grado. Escuela de Ingeniería, Universidad de Barcelona*, 2013.

[9] Otero, P. "¿Es una metodología válida la recolección de datos vía Web?". *Archivos argentinos de pediatría* 106.5, 2008, pp. 390-391.

[10] Weber B, Yarandi H, Rowe M and Weber J. "A comparison study: paper-based versus web-based data collection and management*". Appl Nurs Res*, 2005;18(3):182-5

[11] Bälter, O, and Bälter, K. "Demands on web survey tools for epidemiological research". *European journal of epidemiology* 20.2, 2005, pp. 137-139.

[12] Gonzalez, R; and Woods, R. "*Digital Image Processing*". Ed: Addison-Wesley publishing company, 1993.

[13] Kamlofsky, J y Bergamini, M. "Patrón de Evolución Discreta de Curvatura y Concavidad para Reconocimiento de Formas". *CONAIISI*, 2013.

[14] Rosenfeld, A, and Kak, A. *"Digital picture processing"*. Vol. 1. Elsevier, 2014.

[15] Codd, E. "*Relational completeness of data base sublanguages*". IBM Corporation, 1972.

[16] Morteo, F; Bocalandro, N; Cascon, H; Descalzo, C; De Rosa, K y Krauthamer, D. "*Fundamentos de diseño y modelado de datos*". Buenos Aires: Ediciones Cooperativas, ISBN: 978-987-1246-51-9, 2007.

[17] Dyche, J. "*E-Data: Turning data into information with data warehousing*". Addison-Wesley Professional, 2000.

[18] Hernández Orallo, J, Ramírez Quintana, M, y Ferri Ramirez, C. "*Introducción a la Minería de Datos*". Editorial Pearson Educación SA, Madrid, 2004.

[19] Rosenfeld, A. "Digital topology". *American Mathematical Monthly,* 1979, pp 621-630.

[20] Kamlofsky, J, y Bergamini, M. "Aproximación de formas de objetos digitales por polígonos." *VII Jornadas Argentinas de Robótica.* 2012.

[21] Lengyel, E. "*Matemáticas para Videojuegos en 3D*". Segunda Edición, Cengage Learning, 2011.

[22] Sociedad Argentina de Cardiología. "*Encuesta SAC 201303: Area Políticas de Salud*", 2013, http://www.encuestas.sac.org.ar/201303/ [Consulta: 20 de Julio de 2014].

7 http://www.cs.waikato.ac.nz/ml/weka/

Priorizar Requisitos: un Estudio sobre sus Propósitos

Graciela D.S. Hadad
Universidad Nacional del Oeste
DIIT, Universidad Nacional de La Matanza
ghadad@ing.unlam.edu.ar

Gerardo A. Riera
DIIT, Universidad Nacional de La Matanza
griera@ing.unlam.edu.ar

Jorge H. Doorn
Universidad Nacional del Oeste
DIIT, Universidad Nacional de La Matanza
jdoorn@exa.unicen.edu.ar

Abstract

La literatura propone diversidad de técnicas de asignación de prioridades a requisitos, las que a veces se aplican sobre otros elementos intermedios generados en el proceso de ingeniería de requisitos, tales como casos de uso, historias de usuario, funciones o intereses estratégicos. Así como existe una variedad de algoritmos, escalas de valoraciones cuantitativas y cualitativas, puntos de vista y criterios a utilizar, también se pueden observar distintos propósitos al asignar prioridades. Esto último es un tema poco tratado cuando se realizan estudios comparativos de las mismas. Asimismo, el uso de estas técnicas debiera ser coherente con su propósito, sin embargo, se ha observado que esto no siempre ocurre. El objetivo principal del presente artículo es justamente poner en evidencia los propósitos explícitos y subyacentes de las técnicas de priorización y la coherencia o su falta en oportunidad de su uso, lo que no es una cuestión menor porque gran parte de las actividades del proceso de desarrollo están influenciadas por estas prioridades.

1. Introducción

Un proceso de ingeniería de requisitos involucra actividades centrales de elicitación, modelado, análisis y gestión [1], mientras que otras actividades contribuyen al desenvolvimiento apropiado de éstas, tales como la negociación por conflictos y preferencias entre los involucrados [2, 3], la resolución de inconsistencias entre requisitos [4], el planeamiento de servicios (y de los requisitos involucrados) a implementar en cada entrega de producto [5, 6], y la toma de decisiones sobre qué servicios (y requisitos asociados) postergar para próximos proyectos o cancelar [7], entre otras. Muchas de estas actividades utilizan técnicas de asignación de prioridades como herramienta de soporte. Es importante observar que frecuentemente la literatura menciona erróneamente a estas actividades como métodos de asignación de prioridades, cuando en realidad la priorización no es el núcleo de dichas actividades.

Existen efectivamente múltiples técnicas para priorizar requisitos, en las cuales se establecen uno o varios criterios, que permiten asignar valores a los requisitos en base a esos criterios o agruparlos en función de ellos. En general, estas técnicas tienen como resultado final una lista de requisitos donde cada uno tiene asociado un valor numérico, o una lista de grupos de requisitos donde cada grupo es calificado por un valor nominal o numérico [5]. Aunque, en general la literatura menciona asociar prioridades a requisitos, en realidad, es frecuente que estos criterios se apliquen sobre otros elementos modelados y no necesariamente sobre los propios requisitos individuales. En otras palabras, la priorización puede recaer, por ejemplo, sobre características generales del producto, funcionalidades, aspectos no funcionales de alto nivel, casos de uso, historias de usuario, objetivos del sistema o requisitos dominantes [8, 9]. El hecho que estas técnicas se apliquen con distintos niveles de abstracción, sobre distintos productos y que se obtengan diferentes tipos de resultados, hace evidente la existencia de diferentes propósitos en las mismas. Por ejemplo, Greer [10] detalla una técnica de priorización para encarar desarrollos iterativos e incrementales, mientras que el estándar IEEE 29148:2011 para la especificación de requisitos [11] aconseja priorizar requisitos para establecer cuales de ellos pueden ser negociados por las partes interesadas frente a posibles alternativas. Otros enfoques atienden solo la priorización de requisitos no funcionales para establecer apropiadamente la arquitectura del software [12].

El presente artículo tiene por objetivo poner en un primer plano, a partir del estudio de una variedad de técnicas de priorización con sus características distintivas, los propósitos con que se realizan estas priorizaciones y los posibles usos que se hacen de los

resultados. Estrictamente hablando, el objetivo del artículo responde a una finalidad superior como es la de brindar herramientas que permitan analizar la coherencia entre la forma de definir las prioridades y el uso posterior de las mismas.

En la siguiente sección se introducen características generales que hacen a la priorización; la sección 3 resume algunas técnicas de asignación de prioridades; en la sección 4 se estudian los propósitos y usos posibles de las prioridades obtenidas; y finalmente, se exponen conclusiones y futuros pasos para mejorar la coherencia indicada más arriba.

2. Aspectos Relacionados con la Prioridad

Existen distintos criterios para calificar o ponderar los requisitos por prioridad. Algunas técnicas utilizan solo un atributo [9, 13, 14], llamado directamente prioridad, mientras que otras combinan varios atributos para establecerla [8, 10, 15]. Por lo tanto, una de las tareas iniciales en la priorización es definir los criterios y los atributos necesarios con sus posibles valores o rangos de valores.

Cabe enfatizar que existe una cuestión relevante en relación con la terminología utilizada en algunas de las contribuciones relacionadas con la priorización de requisitos. Obviamente, estos problemas lingüísticos son la parte visible de deficiencias conceptuales subyacentes. Es así que pueden observarse dos usos abusivos de la terminología:

i) El primero de ellos está relacionado con el uso de los términos característica, propiedad, factor, criterio y atributo en forma indiscriminada y, a veces, con indicación explícita de su uso como sinónimos. Esto no es apropiado, ya que una parte de la notoria ambigüedad que existe en el área de conocimiento relacionada con la priorización de requisitos proviene de ese uso descuidado de la terminología. Los autores de este artículo desean establecer claramente su opinión que *criterio* no es sinónimo de los restantes términos. Una propiedad, como por ejemplo: los riesgos relacionados con el rechazo de los usuarios a una determinada funcionalidad del sistema de software, no es un criterio; el criterio sería evitar incorporar estas funcionalidades o tomar el riesgo de hacerlo (ver otros ejemplos de criterios en [16: p.74 y p.80] y en [10: p.102]). Cabe mencionar que Greer [10] denomina *restricción* a los criterios que aplica sobre atributos, y a estos últimos los denomina *factores*. En lo que sigue, se ha restringido el uso de la palabra *criterio* para referirse a aceptar, preferir, rechazar, evitar, maximizar o minimizar ciertos valores de ciertos *atributos*. También, se observa en la literatura que se ha utilizado abundantemente el término *atributo*, descuidando o casi ignorando la aplicación de algún *criterio* sobre él. Esto indudablemente dificulta la puesta en práctica de la técnica, como así también poder establecer la coherencia en su uso.

ii) El segundo de ellos está relacionado con el uso de los términos *prioridad* e *importancia*. En primer lugar, cabe preguntarse qué representa la prioridad. En la propia literatura se ofrecen muchas interpretaciones, para algunos autores prioridad es sinónimo de *importancia*, para algunos otros tiene otra denotación, que la asocian, con *urgencia* o con *necesidad*. El término *importancia*, por sí mismo, conlleva ambigüedad cuando no se explicita importancia para qué, importancia en qué contexto, etc. A su vez, la importancia como sinónimo de prioridad es interpretada de muy variadas formas [5, 17], tales como: necesidad para el cliente o usuario, beneficio estratégico para el negocio, urgencia para implementar, relevancia para determinar su implementación, postergación o cancelación, oportunidad en el mercado, relevancia para la arquitectura del producto software, entre otros. Es decir, existe una sobrecarga apreciable sobre estos dos términos. Entonces, la semántica del atributo prioridad debe definirse con precisión cuando se pone en práctica una actividad de priorización, y esta semántica debería estar directamente ligada con el propósito de priorizar.

Como ya se ha mencionado, la prioridad con frecuencia toma un valor que suele determinarse a través de una combinación de atributos más detallados, dado lo cual su significado puede quedar asociado a la ponderación de los mismos cuando no hay criterios establecidos. Estos atributos detallados pueden ser, por ejemplo, la volatilidad de los requisitos, el beneficio financiero que proveerán, la penalidad (financiera, de prestigio, de pérdida de calidad u otra) por no disponer de los servicios asociados al requisito, el riesgo para el negocio (por ausencia o por presencia de los servicios asociados al requisito), el costo de implementación, el riesgo técnico de implementación, el tiempo de puesta en servicio, los recursos necesarios y la factibilidad de implementación en el negocio, entre otros. Varias técnicas permiten incluso dar pesos relativos a estos atributos [8, 10, 15].

Dependiendo de las técnicas utilizadas, se manejan distintos tipos de escalas de valores [18] asociados a los atributos involucrados: escalas cualitativas, cuantitativas ordinales, cuantitativas proporcionales o cuantitativas absolutas. Algunas técnicas aplican varias escalas, ya sea que combinen o no atributos. Por ejemplo, se usa primero la importancia agrupando requisitos mediante una escala cualitativa, y luego por grupos se los sub-califica ordinalmente, tal es el caso de una variante de la técnica Asignación Numérica propuesta en [19] y de la técnica Juego de Planeamiento [20]. En general, escalas cualitativas dan menos información que las cuantitativas y, dentro de estas, las ordinales dan menos información que las proporcionales, ya que las primeras solo indican

una posición mientras que las segundas además indican una relación cuantificable entre los requisitos [17]. Escalas absolutas son las más precisas aunque también las más costosas de establecer, siendo difíciles de aplicar cuando se combinan varios atributos con distintas unidades de medida (por ejemplo: tiempo, dinero, horas hombre).

Dado que quienes asignan valores a estos atributos son personas involucradas en el proyecto de software, es inevitable tener que considerar aspectos socio-culturales al priorizar. Como menciona Wiegers [8], los ingenieros de software no siempre saben qué requisitos son los más importantes para los clientes y los clientes suelen no poder juzgar el costo y la dificultad técnica asociada a cada requisito. Por lo tanto, según los atributos incluidos en la priorización, se necesitará la participación de distintas personas con distintos roles.

Adicionalmente, debe tenerse en cuenta que los clientes y usuarios tienen sus propios intereses y preferencias, entrando frecuentemente en conflicto entre ellos [21]. Por lo tanto, es fundamental determinar quién de ellos tendrá la decisión de asignar valores a los atributos establecidos o a la prioridad misma. Muchas veces son varios involucrados los que tomarán en conjunto la decisión del valor a asignar a un atributo. Esto lo podrán hacer, por ejemplo, a través de reuniones o cuestionarios, lo que requerirá inexorablemente algún tipo de negociación. Existen algunas técnicas de priorización que tienen incorporada esta concepción de varios participantes, permitiendo entonces que cada uno de ellos asigne un valor distinto a un dado atributo, es decir, permiten reflejar múltiples puntos de vista de los involucrados [10, 22-24], ya sea, considerando distintos roles de clientes y usuarios o distintas áreas representativas de la organización o distintos segmentos del mercado. Es claro que considerar los aspectos sociales que involucran la diversidad de intereses, preferencias y conflictos, dificulta en gran medida la aplicación de la técnica de priorización, cualesquiera sea ésta.

Algunas técnicas de priorización tienen en cuenta las interdependencias de requisitos [25], ya sean positivas o negativas. Las interacciones entre requisitos pueden representar conflictos (interacciones negativas) o pueden reforzarse unidireccional o bidireccionalmente (interacciones positivas), donde por ejemplo el cumplimiento de un requisito depende del cumplimiento de otro requisito, o la implementación de un par de requisitos tiene aspectos en común que pueden incrementar o reducir el costo de implementación de ambos [26]. Algunos de los atributos mencionados resultan completamente invariantes respecto de las interacciones, mientras que otros son fuertemente dependientes de estas. El acoplamiento entre los requisitos puede producir a su vez un acoplamiento entre la definición de las prioridades y su uso posterior, pudiendo requerir considerar este hecho en la toma de decisiones basadas en prioridades (ver sección 4). Continuando con el ejemplo del atributo costo de implementación, si ocurriera que dos requisitos están relacionados de manera tal que la implementación de cualquiera de ellos reduce el costo de implementación del otro y, además, se hubiera decidido implementar uno de ellos pero no el otro por razones de costo y utilidad para el negocio, esta decisión es potencialmente errónea ya que el costo real del requisito descartado no es el utilizado en la toma de decisiones. Las tres formas básicas de atender este problema son: i) Definir prioridades en forma paramétrica, siendo los parámetros de este enfoque las interacciones entre los requisitos, por ejemplo, en la técnica EVOLVE [10: p.103-104]; ii) Agrupar los requisitos con interacciones y tomar las decisiones que correspondan en forma escalonada, por ejemplo, en las técnicas Requirements Prioritization Framework [27] y WinWin Cuantitativo [15]; y iii) Reciclar desde la toma de decisiones hacia las definiciones de prioridades cuando corresponda.

El nivel de abstracción de las unidades a priorizar debiera depender de la granularidad con que se consideren los atributos que se toman en cuenta. Es poco factible utilizar un valor de costo de implementación cuando la unidad a priorizar son funcionalidades o características generales del sistema u objetivos, mientras que este mismo atributo es perfectamente aplicable a requisitos, casos de uso o historias de usuario. En realidad, el proceso de ingeniería de requisitos que se lleve a cabo puede influenciar sobre la elección de qué elementos priorizar, en virtud de los productos resultantes de dicho proceso. Es también frecuente priorizar requisitos haciendo uso de los productos intermedios obtenidos, ya sea priorizando inicialmente, por ejemplo, objetivos y/o escenarios [14, 22, 28].

Un aspecto que influye inevitablemente sobre la técnica de priorización a elegir es la cantidad de elementos a ser priorizados. Existen técnicas que son inviables en grandes proyectos, debido a la cantidad de comparaciones, ponderaciones o valoraciones que requieren sus algoritmos, tal es el caso de la técnica Analytic Hierarchy Process [13]. Una forma posible de alivianar estos inconvenientes es aplicando la técnica de priorización sobre conjuntos de elementos agrupados o utilizando unidades a priorizar con mayor nivel de abstracción, como escenarios o requisitos de alto nivel (funcionalidades o aspectos no funcionales). Es frecuente el uso de técnicas de priorización que solo encaran cierto tipo de requisitos, como requisitos no funcionales debido a la relevancia que los mismos tienen en productos software específicos [12, 29].

En resumen, como se muestra en la Figura 1, son varios los aspectos que pueden ser tenidos en cuenta a la

hora de asignar prioridades, aunque, como se expondrá en la siguiente sección, no todas las técnicas contemplan todos estos aspectos, en la gran mayoría solo algunos de ellos son considerados.



Figura 1. Aspectos que pueden incidir en la priorización.

Debe tenerse en cuenta que la prioridad asignada a un requisito puede cambiar a lo largo del tiempo, ya sea por cambios en el requisito mismo o por cambios en las necesidades del negocio. Por lo cual, es frecuente que la actividad de priorización se realice en forma iterativa a lo largo del desarrollo de software [10, 30], dependiendo básicamente del tamaño del proyecto, del proceso de desarrollo aplicado, de la volatilidad de los requisitos y de los cambios en el contexto de aplicación.

3. Técnicas de Priorización

Se han definido muchas estrategias de asignación de prioridades a requisitos, desde muy simples hasta otras bastante complejas. A veces son clasificadas por el tipo de escala de valoración utilizada [17]. Berander y Andrews [5] las clasifican en dos grandes categorías: enfoques negociadores [15, 31, 32] y enfoques cuantitativos, mientras que Azzolini [33] incorpora una tercera categoría: enfoques cognitivos, que tienen en cuenta aspectos socio-cognitivos de los participantes [28] [34] y, además, sub-clasifica a las técnicas cuantitativas por el tipo de escala de valoración utilizada, casi similar a la presentada en [17]. A continuación se presenta un breve resumen de cada técnica de priorización que será analizada en la siguiente sección respecto a sus propósitos.

Ranking (presentada en [5]) asigna un número único ordinal a cada requisito, ya sea en forma ad-hoc o aplicando algoritmos de ordenamiento, como burbujeo o árbol binario de búsqueda, y utilizando un solo atributo (habitualmente *importancia* con su ya señalada ambigüedad).

Asignación Numérica [35] (denominada también *Grouping*) agrupa cualitativamente a los requisitos en obligatorios, deseables y no esenciales, usando el atributo importancia como sinónimo de relevancia para el cliente.

MoSCoW [36] agrupa los requisitos cualitativamente en cuatro grupos de importancia, que se negocian entre los involucrados, teniendo igual prioridad todos los requisitos pertenecientes a un grupo.

En Requisitos Top-Ten [37], cada involucrado elige los diez requisitos que considera de mayor prioridad y se arma una lista con los requisitos más importantes (sin ordenar), balanceando en ella la cantidad de requisitos elegidos por cada involucrado. Se asume que todos los involucrados tienen el mismo peso.

En AHP (Analytic Hierarchy Process) [13, 38], un involucrado compara de a pares requisitos, asignando un valor relativo de importancia entre ellos y se genere una lista de requisitos con un valor proporcional de prioridad. Posteriormente, Saaty [39] incluyó el uso de varios atributos a ser ponderados, lo que requiere generar una matriz de comparación por cada atributo elegido. Dada la inviabilidad de su uso por la cantidad de comparaciones cuando se tienen muchos requisitos, se han propuesto algunas variantes para reducir el número de comparaciones [38], por ejemplo, haciendo generalizaciones de requisitos (técnica Hierarhcy AHP) o eliminando la redundancia en las comparaciones (técnica Minimal Spanning Tree).

En el Método de 100 Puntos [9], cada participante tiene 100 puntos para distribuir a favor de los requisitos más importantes para él y se arma una lista de requisitos ordenada por el total de puntos asignados por todos los participantes.

El enfoque Costo-Valor [40] utiliza AHP para comparar pares de requisitos evaluando separadamente el valor para el cliente y el costo de implementación, seleccionándose los requisitos que tienen un menor índice de costo y un alto índice de valor.

En el Juego de Planeamiento [20] dentro de la Programación Extrema, el cliente organiza historias de usuario en tres grupos según el valor que aportan al negocio y les da un orden dentro de cada grupo; es decir, combina Asignación Numérica con Ranking.

En VOP (Value-Oriented Prioritization) [41, 42] se establecen los valores centrales del negocio y los riesgos del negocio; los clientes le dan un peso a cada uno de ellos; luego califican en consenso la importancia de cada requisito respecto a cada valor central y a cada riesgo; y como resultado se obtiene una lista ordenada de requisitos en función de la ponderación del requisito para cada valor central y para cada riesgo.

En Requirements Triage [30], los desarrolladores establecen las interdependencias de requisitos y estiman el esfuerzo de implementación de cada requisito y de conjuntos dependientes de ellos; los clientes establecen la importancia relativa de cada requisito usando algún mecanismo de votación; y en base a estas estimaciones,

los clientes, desarrolladores y representantes financieros negocian qué requisitos implementar considerando los que den más probabilidad de éxito al producto software. Se considera que esto se repetirá para cada nuevo release.

Easy WinWin [43, 23] involucra elicitar, elaborar, priorizar y negociar requisitos, donde la priorización se realiza sobre los intereses de los involucrados (denominados *win conditions*) calificándolos por importancia para el negocio y dificultad de implementación, y sobre esa base se acuerdan los requisitos.

En SERUM [44] se realizan estimaciones de costo, beneficio, riesgo de desarrollo y riesgo operacional para cada requisito; se establecen criterios a aplicar sobre estos atributos según las características del proyecto; y se asignan prioridades a los requisitos en base a los criterios aplicados a las estimaciones.

El Método de Wiegers [8] consiste en: determinar que todos los requisitos (o casos de uso, características del sistema u otros elementos) tengan el mismo nivel de abstracción; analizar los requisitos vinculados para incluir sólo el requisito dominante; calcular una prioridad para cada requisito en base a la ponderación de cuatro atributos: beneficio para el cliente, penalidad por no incluir el requisito, costo de implementación y riesgo técnico, cada uno con un peso; y obtener una lista de requisitos ordenada por prioridad.

WinWin Cuantitativo [15] establece inicialmente un conjunto de requisitos agrupados en clases por necesidad de recursos, interdependencia y propósito; a cada requisito se le asigna un valor de importancia relativa dentro de la clase; a cada clase de requisitos se le estima su importancia relativa usando AHP; los involucrados son agrupados en clases, a las que se les asigna un valor de peso desde la perspectiva del negocio; cada clase de involucrados califica por importancia relativa a cada clase de requisitos, usando AHP; se establece un ranking de importancia de cada clase de requisitos; y se estima el esfuerzo de implementación de un conjunto inicial de requisitos; si este no es aceptable entonces se confecciona un nuevo conjunto de requisitos, agregando y eliminando requisitos, y se iteran los pasos anteriores hasta alcanzar un conjunto de requisitos aceptable en términos de esfuerzo e importancia (valor para el negocio).

En SQFD (Software Quality Function Deployment) [24], se identifican tipos de clientes, se les asigna un peso de importancia para la toma de decisiones; se organizan las necesidades del negocio considerando sus interdependencias; se priorizan las necesidades del negocio por tipo de cliente en función de cuanto satisfacen sus intereses y se pondera con el peso del tipo de cliente; se alinean las necesidades con los requisitos, asignando a cada requisito un valor numérico que representa el grado de contribución a la necesidad; y posteriormente se calcula la prioridad del requisito en función del peso de la necesidad del negocio asociada y su grado de contribución.

En Requirements Prioritization Framework [22, 27] se crean perfiles de los involucrados, calificándolos por la importancia de su opinión; cada involucrado califica la importancia de los requisitos, de los objetivos del negocio y de la influencia entre los objetivos y los requisitos, utilizando una escala continua e indicando un nivel de tolerancia para cada valor asignado; se calculan prioridades por grupos de requisitos dependientes, ponderando las calificaciones dadas individualmente y aplicando técnicas de análisis de riesgo para considerar valoraciones subjetivas o corporativas dadas por los involucrados según sus perfiles.

En Pirogov [16], los requisitos se ubican mediante algoritmos automáticos iterativos en clusters por varias características, donde cada requisito queda ubicado en un único cluster, aunque el mismo podría naturalmente pertenecer a más de uno; se calcula el grado de proximidad de cada requisito respecto a cada cluster; se priorizan los clusters negociando entre los involucrados, en base a múltiples criterios previamente definidos y se analizan las dependencias entre clusters; se prioriza por requisito, teniendo en cuenta a qué cluster pertenece, su grado de proximidad con otros y la prioridad de los clusters, armando finalmente un ranking de requisitos.

EVOLVE [10] consiste en estimar para cada requisito el riesgo de implementación, el uso de recursos y el esfuerzo de implementación, con un rango de tolerancia; se establece un peso de importancia para cada involucrado usando AHP; cada involucrado asigna un beneficio a cada requisito; y se utiliza un algoritmo automático que maximiza el beneficio ponderado, teniendo en cuenta las estimaciones de riesgo, de esfuerzo y de recursos, las dependencias entre requisitos y las restricciones asociadas a riesgo, esfuerzo, recursos y dependencias. El resultado de esta función son subconjuntos ordenados de requisitos para cada release, donde todo el conjunto tiene un dado nivel de confianza en cuanto a la estimación del esfuerzo realizado. Es decir, se maneja la incertidumbre en el esfuerzo estimado. Luego, la técnica permite, para distintos niveles de confianza en el esfuerzo estimado, obtener distintos subconjuntos ordenados de requisitos. Con lo cual quienes toman decisiones pueden elegir el resultado que tenga mayor probabilidad de no excederse en el límite de la capacidad de esfuerzo disponible.

4. Usos y Propósitos de la Priorización

A través de la revisión de la literatura, se puede observar que las técnicas de priorización son utilizadas frecuentemente como herramientas de otras actividades en la ingeniería de requisitos, tal es el caso de la

negociación entre involucrados, la resolución de conflictos o la toma de decisiones, e incluso en la selección de los involucrados como fuentes de información [45]. Sin embargo, algunas de estas técnicas de priorización tienen propósitos propios, donde la negociación, la resolución de conflictos o la toma de decisiones son parte de la priorización misma.

Básicamente, se pueden identificar tres propósitos básicos de las técnicas de priorización, lo que indica hay tres interpretaciones posibles de qué es priorizar:

- Seleccionar qué requisitos candidatos serán incluidos o no en el producto software. Esto puede involucrar desestimar algunos requisitos o elegir entre alternativos. En general, para cumplir este propósito se suelen considerar varios puntos de vista y más de un criterio.
- Establecer la *preferencia* o *necesidad* relativa entre los requisitos. Esto implica determinar cuáles son "*más preferidos*" y cuáles "*menos preferidos*" en función de algún criterio o combinación de estos, y eventualmente se puede establecer un orden de preferencia. Se usa la palabra *preferencia* para evitar el uso ambiguo de importancia, de manera tal que el significado de preferencia quedará definido por el atributo adoptado o combinación de ellos.
- Definir un orden secuencial en los requisitos. Usualmente este orden se refiere al orden de implementación en el producto software, lo que puede implicar un desarrollo de software por fases con productos parciales entregables. En la práctica profesional, este es el uso más frecuente de las técnicas de priorización. En general, para cumplir apropiadamente con este propósito se debería tener la visión de los clientes y de los desarrolladores, y contemplar la interdependencia de requisitos.

En este punto emerge nuevamente la cuestión de la terminología y las deficiencias conceptuales asociadas. Se suelen confundir algunas tareas involucradas en la priorización con propósitos de la técnica. Este es el caso de la resolución de conflictos entre requisitos o entre involucrados, la negociación de requisitos o la toma de decisiones. Debe quedar bien claro que las tareas anteriores no son propósitos de la priorización.

Respecto al uso de una técnica de priorización para establecer un orden de implementación, debe destacarse que no basta que un requisito sea de la mayor relevancia para que el mismo sea implementado en primera instancia. Esto dependerá, entre otros factores, de la necesidad técnica de disponer previamente o en simultáneo de otro/s requisito/s para su desarrollo. Es decir, frecuentemente el diseño e implementación de un requisito depende de su interacción con otros, y esto puede influenciar en qué fase debe ser desarrollado [11, 26]. Cabe destacar que aún cuando el desarrollo no sea con entregas parciales de software, suele priorizarse para establecer un orden factible de implementación de requisitos dependiente de las condiciones del proyecto, definiendo diversos criterios, tales como la volatilidad de los requisitos, sus interdependencias y restricciones de recursos, entre otros.

Asimismo, es de notar que los propósitos mencionados podrían ser combinados en pares o ternas, subordinando unos a otros. A continuación indicaremos qué técnicas de priorización estudiadas cubren qué propósitos, indicando por separado aquellas que presentan más de un propósito, donde en general tienen un propósito final pero cumplen complementariamente con otro propósito.

Las técnicas cuyo propósito, en un nivel alto de abstracción, es seleccionar requisitos de un conjunto son: Asignación Numérica [35], MoSCoW [36], Requisitos Top-Ten [37], Easy WinWin [43] y Costo-Valor [40]. En la primera técnica, la priorización establece qué requisitos serán incluidos en el software, cuáles se incluirán de ser posible y cuáles se postergarán o cancelarán. En MoSCoW, los requisitos se agrupan por consenso en una forma bastante similar, aunque con el fin de seleccionar qué requisitos deben implementarse en primera instancia. Tanto en Top-Ten Requisitos como en Easy WinWin y en Costo-Valor se seleccionan los requisitos que se incluirán en el software, teniendo en cuenta los puntos de vista de los involucrados. En la técnica de Costo-Valor, para seleccionar requisitos se establece un orden relativo, el que puede utilizarse para la implementación de los requisitos, es decir, esta técnica contempla optativamente el tercer propósito mencionado.

Las técnicas que tienen por fin establecer las preferencias entre requisitos son: AHP [13], 100 Puntos [9] y SQDF [24]. Las tres técnicas permiten establecer un orden relativo de preferencia entre los requisitos, siendo que en las dos últimas la preferencia es establecida entre todos los involucrados.

Las técnicas que establecen un orden secuencial de los requisitos son: Ranking [5], VOP [41, 42], SERUM [44], Método de Wiegers [8], Requirements Prioritization Framework [22, 27] y Pirogov [16]. El uso práctico de Ranking depende exclusivamente del atributo a utilizar, dada su simpleza de aplicación. En las restantes técnicas, su uso es establecer el orden de implementación de los requisitos, siendo algunas más elaboradas que otras en cuanto a la variedad de aspectos considerados (ver sección 2).

En cuanto a las técnicas que involucran más de un propósito están: Juego de Planeamiento [20], WinWin Cuantitativo [15], Requirements Triage [30] y EVOLVE [10]. Estas técnicas presentan dos combinaciones distintas de propósitos. El Juego de Planeamiento selecciona los requisitos realmente necesarios para el negocio y luego establece el orden en que se implementarán, mientras que EVOLVE, inversamente en

cuanto a los usos, establece el orden de implementación de los requisitos considerando varios conjuntos alternativos de requisitos y se selecciona el conjunto óptimo según criterios predefinidos. WinWin Cuantitativo permite seleccionar por consenso el mejor conjunto de requisitos a incluir en el software, estableciendo preferencias entre clases de requisitos como un propósito subordinado al anterior. En forma similar, Requirements Triage establece preferencias relativas de requisitos para luego seleccionar los requisitos a implementar en el próximo release.

4.1. Coherencia entre las Técnicas y sus Usos

Se evidencia por lo presentado en la literatura que, a veces, no existe una coherencia entre la técnica de priorización y el uso descripto. No toda técnica es compatible con todo uso. Por ejemplo, si una técnica usa los atributos costo de desarrollo y riesgo de implementación para decidir el orden de implementación de los requisitos, entonces no obtendrá un resultado acorde a lo esperado. Es decir, existen atributos usados para priorizar que no son acordes con todos los usos previsibles de la misma.

Para seleccionar requisitos de un conjunto candidato sería apropiado tener la visión de más de un cliente, lo que no está considerado en Costo-Valor ni en Asignación Numérica, agravado en esta última por utilizar solo un atributo para tal selección.

La misma consideración sobre tener más de un punto de vista o una negociación entre involucrados sería lo más apropiado para establecer preferencias entre requisitos, aspecto que no está mencionado en AHP.

Tanto VOP como SERUM no son técnicas adecuadas en la práctica para establecer el orden de implementación, dado que no tienen en cuenta la inter-dependencia de requisitos, pudiendo entonces no proporcionar la mejor solución al planear entregas parciales del producto software. En el caso de VOP, además no considera el punto de vista del desarrollador, con lo cual el orden establecido puede estar muy distorsionado respecto a la realidad del proyecto y sus recursos. Esto mismo ocurre con Requirements Prioritization Framework que tiene el mismo propósito.

En el caso de Juego de Planeamiento, la priorización está basada solo en el valor para el negocio dado por el cliente y aún cuando el desarrollador estima el esfuerzo para cada historia de usuario, esto no es considerado como parte de la priorización, con lo cual el orden de implementación no está afectado por dicha estimación. Respecto al otro propósito de esta técnica que es la selección de requisitos, esta recae exclusivamente sobre el cliente responsable de las historias de usuario sin considerar los puntos de vista de otros involucrados.

En la Tabla 1, se enumeran las técnicas por su propósito y se indica la coherencia de uso de las mismas respecto a los aspectos que ellas contemplan. En el caso de las técnicas que combinan propósitos, estas aparecen en más de una columna, donde la coherencia está indicada para cada propósito particular.

Tabla 1. Propósitos y Coherencia de Uso

Seleccionar Requisitos		Establecer Preferencias		Definir Orden Secuencial	
Técnica	**Coherencia**	**Técnica**	**Coherencia**	**Técnica**	**Coherencia**
Asignación Numérica	NO	AHP	NO	Ranking	NO
MoSCoW	SÍ	100 Puntos	SÍ	VOP	NO
Requisitos Top-Ten	SÍ	SQDF	SÍ	SERUM	NO
Costo-Valor	NO	WinWin Cuantitativo	SÍ	Método de Wiegers	SÍ
Easy WinWin	SÍ	Requirements Triage	SÍ	Req.Prioritization Framework	NO
Juego de Planeamiento	NO			Pirogov	SÍ
EVOLVE	SÍ			Juego de Planeamiento	NO
WinWin Cuantitativo	SÍ			EVOLVE	SÍ
Requirements Triage	SÍ				

Analizando la Tabla 1, es viable considerar que para cumplir con el propósito de seleccionar requisitos se debieran establecer previamente preferencias entre los mismos, al igual que para definir un orden de implementación es esperable haber seleccionado antes qué requisitos implementar. De la Tabla 1, se desprende que esto no suele cumplirse, lo cual indica que probablemente las tareas necesarias para llevar a cabo

esos propósitos subordinados no son considerados parte de la propia técnica de priorización.

El primer y tercer propósito, enunciados al inicio de esta sección, pueden ser útiles por sí mismos, mientras que el segundo referido a establecer preferencias entre requisitos es de alguna manera incompleto, por lo que quedaría subordinado a alguno de los otros dos propósitos (tal es el caso de las técnicas AHP y 100 Puntos), o sería parte de un proceso que no intenta priorizar requisitos (tal es el caso SQFD).

Es importante tener a la vista que las actividades de priorización y de uso de las prioridades están relativamente desacopladas en el tiempo, por lo que no es trivial elegir la técnica de priorización apropiada. Es decir, se debe prestar mucha atención en el uso de la técnica para elegirla apropiadamente. La concepción del trabajo debe ir del uso requerido hacia la técnica a aplicar y, no al revés, teniendo además especial cuidado frente a la posibilidad de la existencia de usos subordinados entre sí.

Dado entonces las numerosas situaciones de incoherencia entre las tareas que se realizan y los propósitos que se persiguen al priorizar, es necesario disponer de una mínima planificación para esta actividad, la que debe contener al menos los siguientes pasos: i) Seleccionar el propósito de la priorización, ii) Garantizar el uso no ambiguo de la terminología, de manera tal que esta sea comprendida por todos los involucrados, y iii) Definir los atributos, criterios y otros aspectos necesarios (ver sección 2) acordes con el propósito seleccionado. Es decir, cuando se va a priorizar, se debe saber para qué se hace (identificar el propósito) y cómo hacerlo (seleccionar la técnica apropiada).

Tal como fueron mencionados, los pasos anteriores pueden ser vistos como elementales y de realización muy obvia. Sin embargo, esto no es así, ya que el propósito final de priorizar es dependiente del proceso de requisitos que se lleva a cabo y del contexto en el que el mismo se desarrolla. Por ejemplo, si el proceso de requisitos define un conjunto de requisitos acordado, verificado y validado, el propósito no será seleccionar los requisitos, sino que posiblemente esté relacionado con el establecimiento de preferencias entre ellos. A su vez, en un contexto de gran volatilidad, es probable que sea necesario hacer una selección de requisitos en cada iteración. Esto significa que los tres pasos mencionados más arriba distan mucho de ser una heurística efectiva para llevar a cabo una priorización de los requisitos. En realidad estos pasos son meramente una enumeración que de ser respetada garantiza la coherencia de toda la actividad. Pese a que este ordenamiento es extremadamente básico, se ha comprobado que no es atendido en muchas de las técnicas observadas. La materialización de heurísticas efectivamente utilizables en la práctica requerirá considerar el proceso de requisitos específico y su contexto [46].

5. Conclusiones

La literatura ofrece una gran gama de técnicas de priorización, que encaran actividades propias de la Ingeniería de Software, incursionando en otras disciplinas como la Sociología y la Economía. No es siempre clara la distinción si realmente una técnica dada es de priorización por sí misma o si es parte de otras actividades que la usan a modo de selección u ordenamiento preferencial. Cuando se describen actividades de soporte a la decisión, de resolución de conflictos, de negociación o de identificación de aspectos cognitivos o sociales de los involucrados, es frecuente el uso de técnicas que asignan valores a factores o atributos que ayudan en dichas actividades. También en el ámbito de planeamiento de proyectos, se identifican los productos software entregables en plazos establecidos, debiéndose identificar qué entregar en qué momento según los recursos disponibles y las necesidades del cliente, para lo cual es inevitable establecer un orden de entregas. Todas estas técnicas suelen ser denominadas técnicas de asignación de prioridades.

Esta diversidad de usos y de técnicas presentadas en la literatura merece, al menos, un esclarecimiento sobre sus propósitos de manera de ayudar a quien tenga que aplicarlas para que pueda seleccionar aquellas que le resulten más útiles a sus fines.

El abordaje expuesto en este artículo solo intenta mostrar la diversidad de propósitos que se enmarcan en la asignación de prioridades e identificar los usos más adecuados para las técnicas de priorización descriptas, debiéndose tener presente además que hay un notorio desacople entre los usos de muchas de las técnicas de priorización y sus propósitos.

No solo hay numerosas técnicas sino varios usos potenciales por lo que no es evidente que un solo mecanismo de priorización sea suficiente. A veces, habría que aplicar dos o más técnicas de priorización para desarrollar un proyecto de software con éxito. A partir de este estudio preliminar, se evidencia la necesidad de establecer una coherencia entre el propósito de la técnica y los mecanismos que ella aplica, recomendando adaptaciones a las mismas para cumplir apropiadamente con sus propósitos.

Se han propuesto unos pasos muy elementales para garantizar la mencionada coherencia entre el uso de la técnica y su propósito. En tal sentido, se propone como trabajo futuro crear una heurística parametrizada para los distintos propósitos de la priorización, teniendo en cuenta diversos contextos de aplicación.

6. Agradecimientos

Este trabajo ha sido parcialmente financiado por la Universidad Nacional del Oeste, la Universidad Nacional de La Matanza dentro del programa PROINCE y la beca CIN otorgada al segundo autor dentro del Programa de Becas de Estímulo a las Vocaciones Científicas,.

7. Referencias

[1] Leite, J.C.S.P., "Understanding the Word 'Analysis' in the Context of Requirements Engineering", *Journal of Computer Science & Technology*, 5, 2, 2005, p.107.

[2] Grünbacher, P., y Seyff, N., "Requirements Negotiation", en Aurum, A., y Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, Springer-Verlag, ch.7, 2005, pp.143-162.

[3] In, H., Olson, D. y Rodgers, T., "A Requirements Negotiation Model Based on Multi-Criteria Analysis", en *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, IEEE Computer Society, USA, 2001. pp. 312-313.

[4] Yen, J., y Tiao, W., "A Systematic Tradeoff Analysis for Conflicting Imprecise Requirements", en *Proceedings 3rd IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, 1997, pp.87-97.

[5] Berander, P., y Andrews, A., "Requirements Prioritization", en Aurum, A., y Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, Springer, ch.4, 2005, pp.69-94.

[6] Regnell, B., y Brinkkemper, S., "Market-Driven Requirements Engineering for Software Products", en Aurum, A., y Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, Springer-Verlag, ch.13, 2005, pp.287-308.

[7] Ngo-The, A., y Ruhe, G., "Decision Support in Requirements Engineering", en Aurum, A., y Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, Springer-Verlag, ch.12, 2005, pp.267-286.

[8] Wiegers, K.E., "First Thing First: Prioritizing Requirements", *Software Development*, 7, 9, 1999, pp.24-30.

[9] Leffingwell, D., y Widrig, D., *Managing Software Requirements - A unified approach*, Addison-Wesley Object Technology Series, 2º ed, 2003.

[10] Greer, D., "Requirements Prioritisation for Incremental and Iterative Development", en Maté, J.L., y Silva, A. (eds.) *Requirements Engineering for Sociotechnical Systems*, Information Science Publishing, ch.VII, 2005, pp.100-118.

[11] IEEE 29148-2011: *IEEE Systems and software engineering - Life cycle processes - Requirements engineering*, IEEE, Nueva York, 2011.

[12] Daneva, M., Kassab, M., Ponisio, M.L., Wiering, R.J., y Ormandjieva, O., "Exploiting a Goal-Decomposition Technique to Prioritize Non-functional Requirements", en *Proceedings 10th Workshop on Requirements Engineering*, Toronto, 2007, pp.190-196.

[13] Saaty, T.L., *The Analytic Hierarchy Process: Planning, Priority Setting. Resource Allocation*, McGraw-Hill, EEUU, 1980.

[14] Hadad G., Doorn J., Ridao M., y Kaplan G., "Facilitando la asignación de Prioridades a los Requisitos", en *Proceedings 12th Workshop on Requirements Engineering*, Valparaíso, 2009, pp.75-84.

[15] Ruhe, G., Eberlein, A., y Pfahl, D., "Quantitative WinWin: a new method for decision support in requirements negotiation", en *Proceedings 14th International Conference on Software Engineering and Knowledge Engineering*, ACM, Nueva York, 2002, pp.159-166.

[16] Duan, C., Laurent, P., Cleland-Huang, J., y Kwiatkowski, C., "Towards automated requirements prioritization and triage", *Requirements Engineering Journal*, 14, 2, 2009, pp.73-89.

[17] Ma, Q., *The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review*, Master Thesis, Auckland University of Technology, 2009.

[18] Stevens, S.S., *On the Theory of Scales of Measurement*, Science, New Series, 103, 2684, 1946, pp.677-680.

[19] Karlsson, J., *Towards a Strategy for Software Requirements Selection*, Licentiate Thesis 513, Linköping University, 1995.

[20] Beck, K., y Andres, C., *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2º ed., Boston, 2004.

[21] Martínez Carod, N., "Ponderación de Requisitos de Software usando Técnicas Cognitivas y Orientación por Objetivos", en *Proceedings XVI Congreso Argentino de Ciencias de la Computación*, Buenos Aires, 2010, pp.1072-1077.

[22] Moisiadis, F., "Prioritizing Scenario Evolution", en *Proceedings 4th International Conference on Requirements Engineering*, IEEE Computer Society, 2000, pp.85-94.

[23] Boehm, B., Grunbacher, P., y Briggs, R.O., "EasyWinWin: A Groupware-Supported Methodology for Requirement Negotiation", en *Proceedings 8th European Software Engineering Conference*, 2001, pp.320-321.

[24] Zultner, R., "Quality Function Deployment (QFD) for Software: Structured Requirements Exploration", en Schulmeyer, McManus (eds) *Total Quality Management for Software*, NY: Van Nostrand Reinhold, 1992, pp.297-317.

[25] Giesen, J., y Völker, A., "Requirements Interdependencies and Stakeholders Preferences", en *IEEE Joint Intl Conference on Requirements Engineering*, 2002, pp.206-212.

[26] Dahlstedt A.G., y Persson A., "Requirements Interdependencies: State of the Art and Future Challenges", en Aurum, A., y Wohlin, C. (eds.) *Engineering and Managing Software Requirements*, Springer-Verlag, ch.5, 2005, pp.95-116.

[27] Moisiadis, F., "A Requirements Prioritisation Tool", en *Proceedings 6th Australian Workshop on Requirements Engineering*, Sydney, 2001.

[28] Kaiya, H., Shinbara, D., Kawano, J., y Saeki, M., "Improving the Detection of Requirements Discordances Among Stakeholders", *Requirements Engineering Journal*, 10, 4, 2005, pp.289-303.

[29] Ryan, A., "An Approach to Quantitative Non-Functional Requirements in Software Development", en *Proceedings 34th Annual Government Electronics and Information Association Conference*, 2000.

[30] Davis, A., "The Art of Requirements Triage", *IEEE Computer*, 36, 3, 2003, pp.42-49.

[31] Boehm, B.W., y Ross, R., "Theory-W software project management: principles and examples", *IEEE Transactions on Software Engineering*, 15, 7, 1989, pp.902-916.

[32] Fricker, S., y Grünbacher, P., "Negotiation Constellations – Method Selection Framework for Requirements Negotiation", en *Proceedings 14th Requirements Engineering: Foundation for Software Quality*, Springer Berlin Heidelberg, 2008, pp.37-51.

[33] Azzolini, C.M., *Un Enfoque de Priorización de Requerimientos, a partir de la Segmentación de las Preferencias de los Stakeholders*, Tesis de Maestría, UNLP, 2012.

[34] Martínez Carod, N., y Cechich, A., "Cognitive Influences in Prioritizing Software Requirements", en *Proceedings 5th Intl Conference on Software and Data Technologies*, Atenas, vol.2, 2010, pp.214-219.

[35] Brackett, J.W., "Software Requirements", en *SEI-CM-19-1.2*, ADA235642. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1990.

[36] Clegg, D., y Barker, R., *Case Method Fast-Track, A RAD Approach*, Addison-Wesley, Boston, 1994.

[37] Lausen, S., *Software requirements – styles and techniques*, Pearson Education, Essex, 2002.

[38] Karlsson, J., Wohlin, C., y Regnell, B., "An evaluation of methods for prioritizing software requirements", *Information and Software Technology*, 39, 14-15, 1998, pp.939-947.

[39] Saaty, T.L., "Decision making with the analytic hierarchy proces", *International Journal of Services Sciences*, 1, 1, 2008, pp.83-98.

[40] Karlsson, J., y Ryan,K., "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software*, 14, 5, 1997, pp.67-74.

[41] Azar, J., Smith, R.K., y Cordes, D., *Value-Oriented Prioritization: A Framework for Providing Visibility and Decision Support in the Requirements Engineering Process*, Technical Report, University Of Alabama, 2006.

[42] Azar, J., Smith, R.K., y Cordes, D., "Value-Oriented Requirements Prioritization in a Small Development Organization", *IEEE Software*, 2007, pp.32-37.

[43] Grünbacher, P., "Collaborative Requirements Negotiation with Easy WinWin", en *Proceedings 2nd International Workshop on the Requirements Engineering Process*, IEEE Computer Society, Londres, 2000, pp.954-960.

[44] Greer, D., Bustard, D., y Sunazuka, T., "Prioritisation of system changes using cost-benefit and risk assessments", en *Proceedings 4th IEEE Intl Symposium on Requirements Engineering*, 1999, pp.180-187.

[45] Leite, J.C.S.P., Moraes, E.A., y Castro, C.E.P.S., "A Strategy for Information Sources Identification", *X Workshop on Requirements Engineering (WER'07)*, Toronto, Canadá, Mayo 2007, pp.25-34.

[46] Welke, R.J., y Kumar, K., "Method Engineering: a proposal for situation-specific methodology construction", en Cotterman y Senn (eds.), *Systems Analysis and Design: A Research Agenda*, Wiley, Chichester, 1992, pp.257-268.

Extracción de información y cálculo de métricas en WSDL 1.1 y 2.0

Hernán Bernardis[(1)], Mario Marcelo Berón[(1)], Edgardo Bernardis[(1)], Daniel Edgardo Riesco[(1)]
Pedro Rangel Henriques[(2)]
[(1)]Departamento de Informática-Universidad Nacional de San Luis
Ejército de los Andes 950 – San Luis - Argentina
{hbernardis, mberon, ebernardis, driesco}@unsl.edu.ar
[(2)]Departamento de Informática-Universidade do Minho
Braga – Portugal
pedrorangelhenriques@gmail.com

Resumen

En la actualidad, existe un incremento importante en el desarrollo de sistemas basados en una arquitectura orientada a servicios. Dichos sistemas aprovechan la gran oferta de Web Services existentes en la red para implementar funcionalidades mediante el uso de estos servicios externos. Esto provoca una descentralización del sistema al no contener en su propio código la implementación de todas sus funcionalidades. El entendimiento de este tipo de sistemas se dificulta aún más debido a que se debe comprender cada Web Service usado para poder realizar un mantenimiento correcto del sistema. Comprender un Web Service no es una tarea simple debido a que demanda la interpretación de una funcionalidad de la cual no se posee el código fuente sino sólo una especificación en WSDL del mismo. En este artículo se describe una metodología que facilita la comprensión de los Web Services mediante la extracción de información y el cálculo de métricas de los mismos. También se presenta SIPi, una herramienta cuyo principal propósito es facilitar la inspección y comprensión de descripciones WSDL en sus dos versiones más utilizadas como lo son WSDL 1.1 y WSLD 2.0.

1. Introducción

Muchas investigaciones demuestran que, del proceso normal de desarrollo de software, la etapa de mantenimiento es aquella que mayores costos en tiempos y recursos genera [3, 13, 1, 4, 5, 23]. A su vez, la mayoría de esos recursos y tiempos son utilizados en tareas de comprensión del sistema al que se le desea realizar el mantenimiento [8, 6, 15, 22]. Esto se fundamenta en un razonamiento básico: para poder realizar un cambio sobre un sistema, es necesario primero poseer un conocimiento profundo del mismo. De lo contrario, con la aplicación de los cambios se pueden llegar a introducir nuevos errores que incluso pueden ser más costosos que los que originaron la modificación. En Ingeniería de Software, la actividad de lograr obtener un conocimiento profundo de los sistemas es conocida como Comprensión de Programas (CP). La CP abarca todas aquellas técnicas, estrategias, procesos y herramientas utilizadas para facilitar el entendimiento de los sistemas de software [6, 20, 25, 24].

Actualmente, con el auge de Internet, están surgiendo una gran cantidad de sistemas basados en Web Services (WS). Dichos sistemas utilizan las funcionalidades provistas por los distintos WS existentes en la red, realizando invocaciones a los mismos. Esto incrementa su complejidad para comprenderlos debido a que:

1. Las componentes del sistema asociadas a los WS no son conocidas en su funcionamiento interno sino solamente en su interacción. Una de las principales características de los WS radica en el hecho de ocultar su implementación y documentar solamente la forma en que se interactúa con ellos. Inicialmente puede parecer que esto facilita el mantenimiento debido a que ciertas funcionalidades del sistema no son implementadas sino solo "importadas" desde los WS. Sin embargo, si se desea modificar un sistema realizando cambios en aquellas partes que involucran WS, es necesario poder comprender dichos WS y la interacción de los mismos con el sistema.

2. El uso de los WS se hace posible debido a su fuerte estandarización. Comprender un sistema con WS implica poseer un conocimiento bastante amplio de esta estandarización, además del conocimiento propio del sistema en sí. Si un ingeniero de software se decide a modificar un sistema que utilice WS, no solo deberá entender el funcionamiento del sistema sino también todos aquellos aspectos relacionados con la definición y especificación de los WS.

3. Comprender un sistema centralizado ya supone una tarea sumamente compleja. Si a esto se le agrega el hecho de que los sistemas pueden estar descentralizados, utilizando WS, la complejidad del análisis se incrementa aún más.

4. El uso de los WS es una práctica bastante nueva y que cada vez posee una mayor aceptación y uso dentro de las empresas y los sistemas de gobierno. Sin embargo, este crecimiento nuevo hace que las técnicas de comprensión de los mismos sean aun más difíciles de encontrar. Entonces, con un auge cada vez mayor en su uso y con técnicas de comprensión casi inexistentes, se hace aún más difícil la tarea de comprensión de programas.

5. La estandarización de los WS permiten automatizar su construcción y la construcción de sistemas que los utilicen. Sin embargo, estos procesos automatizados suelen carecer de una documentación clara y completa.

6. Si bien los WS suelen documentación que los describe, el simple hecho de que esa documentación probablemente no haya sido escrita por las mismas personas que desarrollaron el sistema que se desea mantener, plantea ciertas diferencias. La forma de documentar el sistema es distinta con la forma de documentar el WS (terminología, idioma, nivel de abstracción, nivel de detalle, semántica asociada, etc.). Además, si bien los WS tienden a poseer una documentación actualizada (vital para poder vender el WS) esto no siempre se cumple. Esto se vuelve peor si el sistema utiliza múltiples WS, los cuales incluso pueden no haber sido realizados por las mismas personas.

7. El hecho de no poseer el código fuente de los WS puede restringir el nivel de entendimiento del sistema. Incluso puede restringir el modelo cognitivo a usar a la hora de comprender el sistema ya que se debe quedar solo con el funcionamiento a alto nivel del WS (Top Down).

Todas las razones antes mencionadas hacen que el foco de este artículo se centre en la comprensión de los sistemas basados en WS. En particular, en definir, calcular y obtener métricas desde los WS mediante el análisis directo de su descripción.

Este artículo se encuentra organizado de la siguiente manera: en la Sección 2 se describe la metodología definida en este trabajo. En la sección 3 se presenta y describe SIPi, una herramienta cuyo principal propósito es facilitar la inspección y comprensión de descripciones WSDL en sus dos versiones más utilizadas como lo son WSDL 1.1 y WSLD 2.0. En la sección 4 se muestra un caso de estudio utilizando SIPi. Finalmente, en la sección 5 se exponen las conclusiones y los trabajos futuros.

2. Comprensión de Web Services

La W3C define a un Web Service como un sistema de software diseñado para permitir la interoperabilidad máquina a máquina en una red. Estos poseen una interfaz descripta en un formato procesable por una máquina, denominado Web Service Description Language (WSDL) [27]. Un WSDL es un lenguaje basado en el formato XML que se utiliza tanto para describir las partes asociadas a un WS como la forma de comunicarse con el mismo. Cualquier interesado en utilizar un WS, debe buscar su especificación WSDL para conocer la forma correcta mediante la cual utilizarlo.

Suena lógico entonces centrarse en el análisis de los WS a través de sus WSDL con el objetivo de extraer y analizar información de los mismos que ayuden a comprenderlos. En las secciones a continuación se detalla las ideas principales relacionadas a esta investigación junto con la metodología utilizada.

2.1. Extracción de Información de los WSDL

La interpretación de los WSDL mediante su lectura directa no es una tarea sencilla. Detectar sus componentes, anidamientos, analizar los comentarios, los espacios de nombres, etc. a través de su lectura directa resulta una tarea casi imposible. Toda esta información es vital para crear un modelo cognitivo del WSDL que permita comprenderlo, motivo por el cual se hace necesario extraerla, aislarla y visualizarla para facilitar y agilizar la interpretación [10, 14, 9, 17].

Debido a la estandarización que poseen los WSDL en su creación, se puede automatizar el proceso de extracción de información. Recorriendo el WSDL en busca de aquellas palabras definidas como claves, se extrae información del mismo y se la preserva en alguna estructura de datos adecuada. Luego, a partir de la información extraída se pueden realizar distintos tipos de análisis que faciliten la interpretación de los WS. Por ejemplo, se pueden extraer los espacios de nombres del WSDL para: i) determinar si los identificadores que se encuentran en una descripción son correctamente utilizados, ii) acceder rápidamente a los identificadores de un espacio de nombres particular, iii) navegar una descripción basándose en los espacios de nombres y los identificadores definidos, iv) etc. Además de las palabras claves que poseen los WSDL para estructurar su contenido, también se puede extraer otra información vital para el entendimiento de los mismos y que nada tiene que ver con su manera de definirlos: los comentarios. Múltiples investigaciones hablan de la

importancia del análisis de los comentarios a la hora de comprender un programa [18, 21, 28]. Si a esto se le suma el hecho de que los WSDL en general están fuertemente comentados para facilitar su uso, se deduce fácilmente la utilidad de extraerlos.

2.2. Análisis y uso de la información extraída

La extracción de la información de los WSDL es solo el primer paso dentro del proceso de comprensión de los WS. De poco sirve poseer esta información si no se la aprovecha de la manera adecuada. Toda información extraída debe ser sometida a un análisis más profundo que permita obtener ideas más generales, conclusiones y un mayor conocimiento del WS que se intenta comprender. Todo esto con el objetivo de crear un modelo cognitivo mental del WS que le facilite, al ingeniero de software, realizar las tareas de mantenimiento sobre el sistema [11, 7, 29, 16].

En este trabajo particular se realizan múltiples tareas, tendientes a facilitar la comprensión, sobre la información extraída. Las mismas se detallan en las subsecciones siguientes.

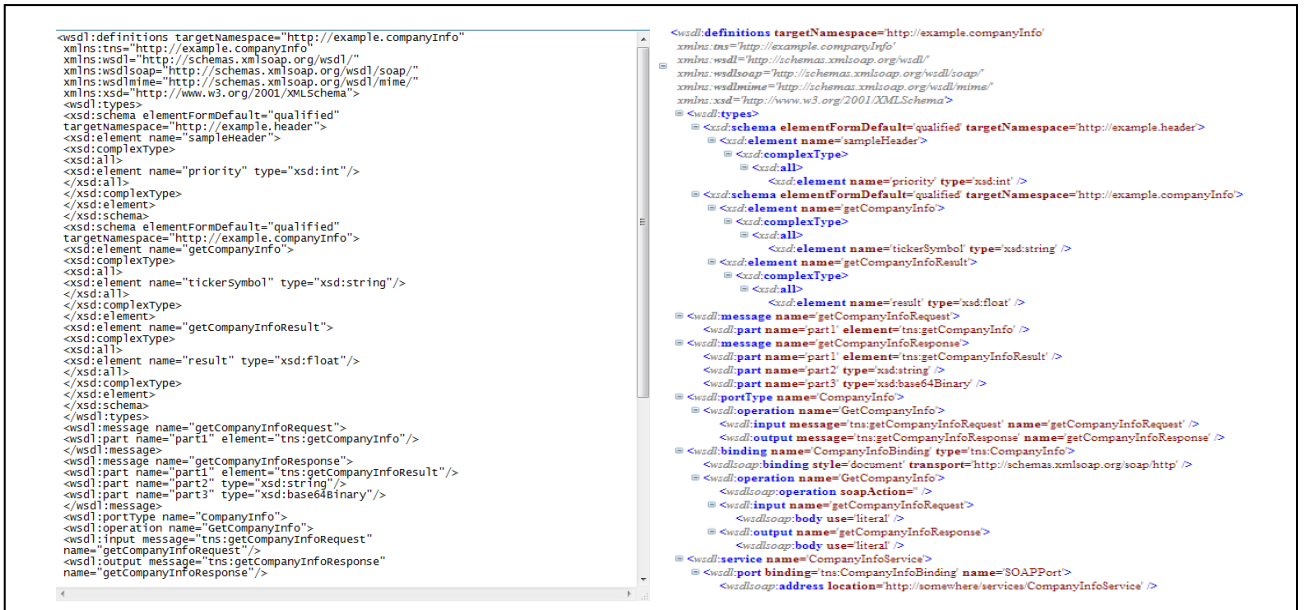
2.2.1. Cálculo de métricas de los WSDL. El cálculo de métricas de un WSDL permite conocer características importantes de los mismos. Las cantidades de tipos, mensajes, puertos, ligaduras y servicios, espacios de nombres, etc. que contiene un WSDL sirven para armar un modelo cognitivo inicial de su estructura y funcionamiento. Otras métricas más avanzadas pueden determinar la complejidad que posee el WSDL, como por ejemplo, la cantidad de tipos de datos utilizados en cada mensaje. Entre las distintas métricas que se calculan en esta investigación, se encuentran:

- **Líneas de código:** es la cantidad de líneas de código físicas que posee el WSDL, sin contemplar los imports de otros WSDL que pueda realizar. Esta métrica aporta una idea inicial e intuitiva de la complejidad del WS.
- **Cantidad de Messages:** es la cantidad de mensajes que posee el WSDL. Mientras mayor sea la cantidad de mensajes que utiliza el WSDL, mayor complejidad tendrá y, posiblemente, utilizará protocolos de comunicación más robustos.
- **Cantidad de Port Types:** esta métrica cuenta la cantidad de port types que posee el WSDL. Para el caso de WSDL 2.0, esta métrica se reemplaza por cantidad de Interfaces. Esto puede ser utilizado para medir la complejidad de un servicio en términos de la cantidad de operaciones y mensajes que este disponibiliza. Mientras más operaciones tenga un servicio y mientras más mensajes usen las operaciones más difícil será comprender el servicio. Además una descripción que provea diferentes servicios y los mismos no sean triviales implicará que la descripción del WS también será difícil de percibir.
- **Cantidad de Bindings:** el formato de los mensajes y los protocolos utilizados por un WS pueden, también, ser un parámetro para medir su complejidad y criticidad. Si un WS utiliza varios protocolos puede suceder que sea una componente central en varios sistemas heterogéneos y por lo tanto un error, o mal funcionamiento del mismo puede causar que dichos sistemas colapsen.
- **Cantidad de Operations:** Una descripción WSDL con muchas operaciones puede: i) representar un punto de sobrecarga y por lo tanto puede ser sometido a tareas de mantenimiento perfectivo, ii) contener muchos servicios (simples o complejos) lo cual dificulta la comprensión, iii) indicar un servicio crítico difícil de corregir, iv) etc.
- **Cantidad de Servicios:** Mientras más servicios contenga una descripción WSDL más compleja será su comprensión.

2.2.2. Representaciones de los WSDL. Construir representaciones gráficas de los WSDL puede acelerar el proceso de entendimiento. Por ejemplo, es más fácil interpretar la estructura de un WSDL mediante su representación con un árbol que con la lectura directa sobre su XML, tal como lo muestra la Figura 1. En este trabajo también se estudian las posibles representaciones y técnicas de visualización con el objetivo de determinar cuáles son las más adecuadas a la hora de comprender un WSDL.

2.2.3. Navegación interactiva de WSDL. No solo las representaciones de WSDL facilitan su comprensión. Poder navegar entre el código del WSDL y sus representaciones es de gran ayuda para el análisis minucioso y específico de sus elementos. Por ejemplo, acceder a un nodo particular del árbol que representa al WSDL y obtener automáticamente el trozo de código asociado del WSDL, permite rápidamente intercambiar entre una vista general del WSDL a una más específica. En el orden inverso, situarse en una línea particular del WSDL y automáticamente obtener donde se encuentra en la estructura general representada por el árbol, permite pasar de una vista específica a una más general de una forma sencilla, práctica y rápida.

2.2.4. Comparación de versiones de un mismo WS. Muchas veces los cambios en los sistemas basados en WS no se deben a cambios funcionales propios del sistema sino a cambios en los WSs que utiliza. Los WS también están sujetos a tareas de mantenimiento que implican cambios en los mismos por partes de sus desarrolladores. Es importante poder conocer



a) Representación en texto de un WSDL.

b) Representación en árbol de un WSDL.

Figura 1. Representaciones mediante texto y un árbol de un WSDL.

rápidamente cuales son los cambios que ha sufrido un WS de manera de poder calcular su incidencia sobre el sistema que los utiliza. Esto se logra mediante el análisis de las distintas versiones de los WSDL asociados a cada cambio realizado sobre el WS. La comparación entre la información extraída de cada WSDL puede facilitar la detección de aquellas partes que han sufrido cambios y, de estas, las que pueden afectar a los sistemas que los utilicen (no todos los cambios de un WSDL afectan a los sistemas que utilizan el WS asociado).

2.2.5. Comparación entre distintos WS. Todos los análisis mencionados en las secciones anteriores permiten conocer información sobre un WS en particular. Sin embargo, los sistemas basados en WS tienden a usar más de uno en su implementación. Es importante realizar un análisis en conjunto de todos los WSDL utilizados por un sistema de modo de conocer ciertas características importantes a la hora de comprenderlos. Conocer qué WS es más complejo de utilizar, cuál es más utilizado, cuál tiene más funcionalidades, más tipos de datos complejos, más comentarios, más tipos complejos, más espacios de nombres, etc. puede facilitar la comprensión de la magnitud de los cambios a realizar. Si el mantenimiento afecta a aquellos WS de mayor complejidad, los costos en tiempos y recursos serán mayores a si se realizan sobre los más simples. La misma lógica se aplica para el caso en donde se consideran cantidades de WS afectados.

El uso de gráficos, tablas, diagramas, etc. es vital para representar comparaciones entre WSDL y sus características.

2.3. Versiones de WSDL

Si bien existen múltiples versiones de WSDL, conceptualmente las mayores diferencias se encuentran entre las versiones 1.1 y 2.0. Ambas se encuentran vigentes y son utilizadas en la actualidad. Las diferencias que poseen entre ellas hacen que la extracción de información de los WS se deba realizar teniendo en cuenta las características propias de cada una.

2.3.1. WSDL 1.1. Esta primera versión del lenguaje está compuesta por los siguientes elementos principales:

- **Definitions:** es el elemento raíz que encapsula todo el WSDL y se utiliza para definir el nombre del servicio y declarar todos los espacios de nombres que este utiliza.
- **Documentation:** se utiliza para describir y/o documentar el WSDL o alguna de sus partes, con el objetivo de facilitar su entendimiento y uso por parte de los clientes del mismo. Generalmente es

escrito lenguaje natural por los desarrolladores del WS.

- **Types:** es un contenedor para las definiciones de los tipos de datos utilizados en el WS usando algún tipo de sistema. Generalmente, XSD es utilizado para definirlos.
- **Message:** es una definición abstracta de los datos a transmitir. Está compuesto de partes que están asociadas con una definición dentro de algún tipo de sistema.
- **Operation:** una descripción abstracta de una acción soportada por un servicio.
- **Port Type:** es un conjunto abstracto de operaciones en donde se especifica, para cada una de ellas, la descripción de su entrada y su salida. WSDL considera cuatro tipos primitivos de operaciones: i) One-Way, ii) Request-Response, iii)Solicit Response y iv) Notification. Además, cada Port Type posee un nombre que es único entre todos los Port Types del WSDL.
- **Binding:** define el protocolo concreto y la especificación formal de los datos para un Port Type particular.
- **Port:** describe un único endpoint definiéndolo como una combinación de binding y direcciones de red.
- **Service:** una colección de Ports relacionados.

2.3.2. WSDL 2.0. Esta última versión del lenguaje ha sufrido ciertos cambios en su especificación. La principal ventaja que se nota al compararla con su antecesora es el incremento en legibilidad y fácil comprensión de esta última. Entre las principales novedades que posee el WSDL 2.0 se encuentran:

- **Description:** reemplaza al elemento definitions de WSDL 1.1 cumpliendo su misma función.
- **Types:** este elemento se mantiene sin cambios respecto de la versión 1.1.
- **Message:** no existen en esta nueva versión.
- **Interface:** se utiliza para especificar las operaciones disponibles dentro del WS. Cada uno de estos componentes (pueden existir más de uno) define un conjunto de operaciones que se realizan dentro del WS. Similar a los Port Types de la versión 1.1, solo que en lugar de referenciar a messages, referencia a los tipos definidos en Types.
- **Operation:** igual que la versión 1.1
- **Binding:** similar a la versión 1.1, solo que se aplica para cada interface y no para cada PortType.
- **Service:** similar a la versión 1.1, pero aplicado sobre interfaces.

Todas las diferencias entre los elementos más generales de las versiones 1.1 y 2.0 de WSDL se ilustran en la Figura 2 [30].

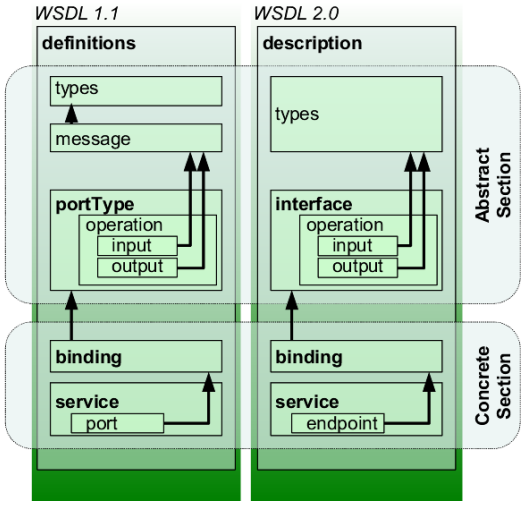


Figura 2. Diferencias entre los elementos principales de WSDL 1.1 y WSDL 2.0.

3. SIPi.

Todos los conceptos teóricos definidos en las secciones anteriores han servido como puntapié inicial para la definición y construcción de una herramienta que permita implementarlos. Dicha herramienta ha sido denominada SIPi: Simple Inspection Program improved. SIPi puede resumirse en base a las tareas principales que realiza:

1- Leer el código WSDL de los WS, tanto para la versión 1.1 como para la 2.0
2- Construir una representación intermedia del WSDL
3- Aplicar distintos comandos que permiten extraer información y calcular métricas del WS a partir de la representación intermedia construida.
4- Mostrar la información extraída y las métricas calculadas en su interfaz, valiéndose en muchos casos del uso de distintas técnicas de visualización.

Para comprender un poco más como logra realizar estas tareas, es necesario analizar su arquitectura interna.

3.1. Arquitectura de SIPi

SIPi se puede dividir en tres capas lógicas teniendo en cuenta su funcionamiento: la Capa de Construcción, la Capa de Extracción y Análisis y la Capa de Visualización. En la Figura 3 se pueden observar cada

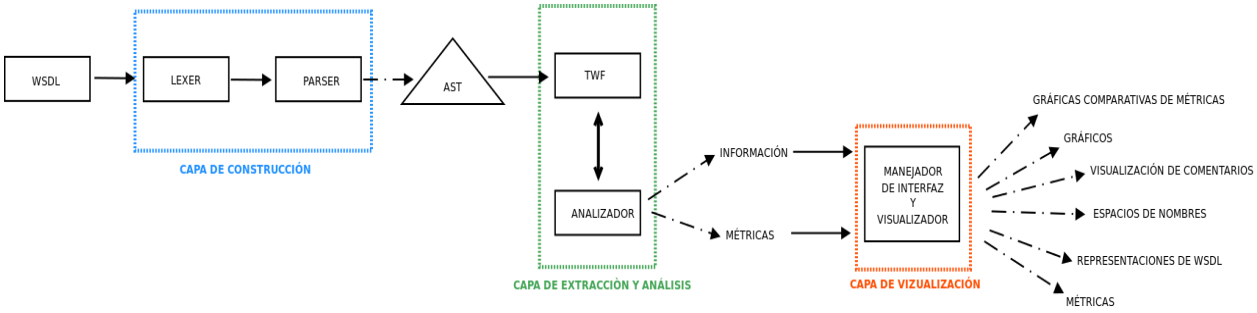


Figura 3. Arquitectura de SIPi.

una de las capas, su relación entre sí y los componentes que las conforman.

3.1.1. Capa de Construcción (CC). Esta capa agrupa todas aquellas componentes encargadas de crear la representación intermedia del código fuente del WSDL. Tal como lo muestra la Figura 3, un lexer y un parser son las componentes de CC. El lexer se encarga de leer el código fuente, recorrerlo y mapear cada elemento encontrado en un token que lo representa. El parser por su lado, recibe los tokens que el lexer le envía y realiza dos grandes tareas a partir de ellos: i) chequea que el WSDL no contenga errores sintácticos y ii) construye una representación del WSDL basada en un Árbol de Sintaxis Abstracta (AST) [2].

3.1.2. Capa de Extracción y Análisis (CEA). Esta capa contiene los componentes encargados de generar la información que será utilizada para comprender el WSDL. Está compuesta por dos módulos: el TWF y el Analizador. El primero es quien recorre el AST y extrae la información del mismo. Para lograr esto, utiliza diferentes funciones de Recorrido de Árboles (de ahí su nombre TWF, por sus iniciales en inglés Tree Walking Functions) con las cuales recorre los nodos del AST recuperando datos de los mismos que almacena temporalmente. El segundo, toma los datos extraídos y almacenados por el TWF, los analiza y a partir de estos calcula métricas o deduce información del WSDL.

3.1.3. Capa de Visualización (CV). Tal como su nombre lo indica, esta capa es la encargada de manejar la interfaz de la herramienta y de mostrar los resultados obtenidos del análisis realizado. Se compone de un único módulo que toma las métricas y la información generada por CAE, le aplica distintas técnicas de visualización y visualiza los resultados en su interfaz.

3.1.4. Independencia entre capas. Si bien todas estas capas trabajan en conjunto para lograr el objetivo de facilitar la comprensión del WSDL, se puede notar que entre ellas solo existe una relación a nivel de estructuras de datos. Esto produce una cierta independencia a nivel de funcionalidad que facilita la escalabilidad de SIPi. Se pueden introducir cambios o modificaciones en una capa de manera aislada a las restantes siempre que se mantenga la estructura de datos que la misma genera. Entonces, la independencia de funcionalidad entre capas se logra al generar una dependencia a solo nivel de estructura de datos entre las mismas. Por ejemplo, la capa de construcción y la capa de extracción y análisis se relacionan por medio del AST que genera la primera. Se podría entonces modificar o incluso reemplazar el método usado para generar el AST siempre y cuando, el tipo de estructura generada continúe siendo un AST. De la misma forma, se pueden agregar en la CAE nuevos recorridos al AST que permitan calcular nuevas métricas o extraer otra información sin necesidad de que la CC se vea afectada.

3.2. Construcción de SIPi

SIPi ha sido desarrollada en lenguaje JAVA con el objetivo de facilitar su portabilidad y aprovechar la gran cantidad de librerías existentes. En la actualidad, SIPi implementa la mayoría de las métricas descriptas en este trabajo y extrae información tanto de los comentarios como de los espacios de nombres de los WSDL. En la Figura 4 se puede observar la interfaz de SIPi.

3.2.1. Construcción de la representación intermedia. Los árboles son las estructuras más comúnmente utilizadas para construir representaciones de códigos fuentes. Para generarlos, es necesario un lexer y un parser que recorran dicho código fuente y lo construyan. Actualmente, existen múltiples herramientas que dada la gramática del código fuente, permiten generar de manera automática el lexer y parser asociado. Afortunadamente, el código fuente utilizado por los WSDL no es más que subconjunto bien definido de XML, lenguaje para el cual existen una gran cantidad de parsers (con sus lexers

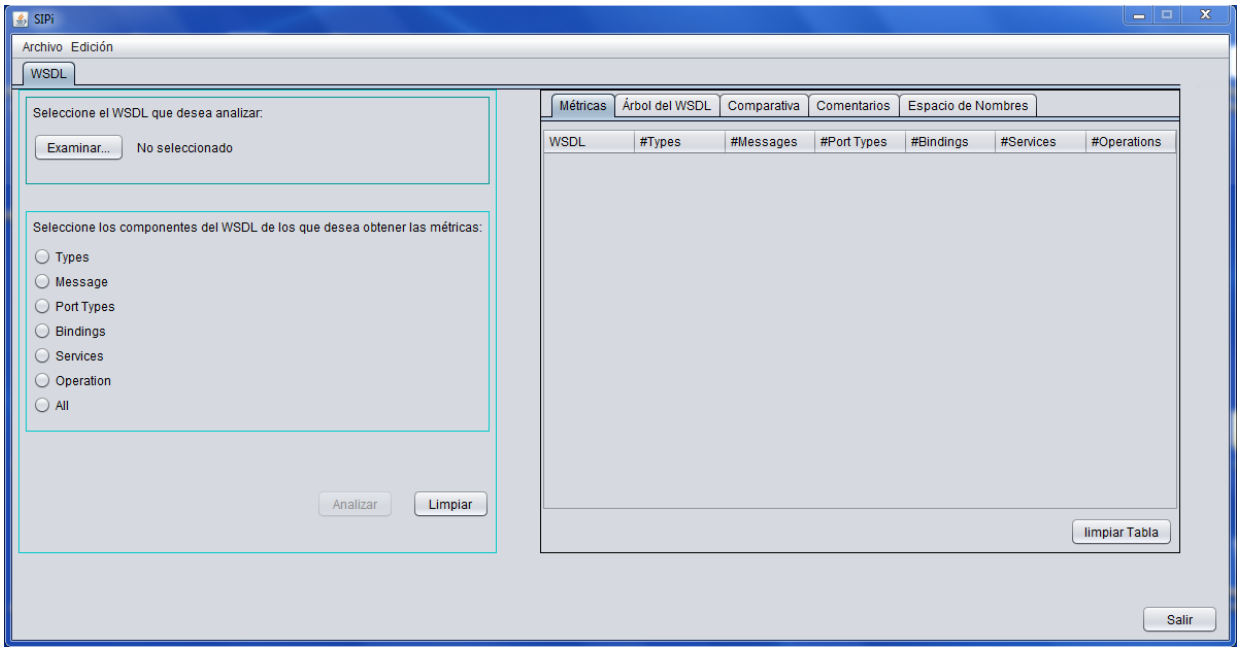


Figura 4. Interfaz de SIPi.

asociados) ya definidos. En este contexto, se pueden distinguir dos tipos principales de parsers: SAX y DOM [19, 26,]. Los parsers SAX (**S**imple **A**PI for **X**ML) se destacan por no construir de manera explícita el AST sino que leen el código fuente a medida que lo necesitan. Este tipo de estrategia se conoce como *on-fly* y tiene la ventaja de consumir menos memoria al no construir y mantener el AST. Sin embargo, esto también produce una pérdida de potencia en los análisis que puede realizar. Como resultado, se obtiene una API basada en eventos. El parser envía una consulta mediante un evento al event-handler y este se encarga de analizar el código y responderle con el resultado. Por otro lado, los parsers DOM (**D**ocument **O**bject **M**odel) recorren todo el código fuente de manera inicial, construyen el AST por extensión y lo almacenan en memoria. Esto posee la gran ventaja de que el AST queda disponible para su análisis posterior mediante distintos recorridos.

En el caso particular de este trabajo, se eligió utilizar un parser del tipo DOM debido a que: i) facilita la implantación de técnicas de extracción de información, ii) permite obtener información que requiere un análisis más complejo, iii) facilita la independencia funcional entre módulos (Ver Sección 3.1.4). Tanto SAX como DOM ya se encuentran implementados en Java mediante la librería JWSDL [12].

3.2.2. Cálculo de métricas y extracción de información. Una vez que se posee en memoria al AST generado por el parser DOM, se pueden aplicar recorridos al mismo dependiendo de la métrica a calcular o la información que se desea obtener de cada nodo. A continuación se explica la manera mediante la cual se calcula la métrica de número de operaciones y de qué manera se extraen los comentarios y los espacios de nombres del WSDL (versión 1.1).

- **Métrica de número de operaciones:** Cada puerto tiene un conjunto de operaciones que indica generalmente la tarea realizada por WS a través de ella. Si se desea conocer el número total de operaciones dentro de un WSDL, se deben analizar todos los elementos Ports del mismo, contando en cada uno de ellos sus operaciones. El Algoritmo 1 muestra cómo esta métrica puede ser calculada. En la línea 2 se extraen todos los servicios. Desde la línea 3 a la línea 9, se cuenta la cantidad de operaciones por puerto y se la suma al total de operaciones en el WSDL, almacenado en la variable *numOp*. Finalmente, en la línea 10 el algoritmo retorna la cantidad total de operaciones presentes en el WSDL.

Algoritmo 1: Calcular métrica de número de operaciones

Entrada: *wsdlRaiz* el nodo raíz del AST asociado a un WSDL.
Salida: *numOp* un entero con la cantidad de operaciones presentes en el WSDL.

```
1.numOp = 0;
2.servs = extraerServicios(wsdlRaiz)
3. for all srv in servs do
4.  puertos = extraerPorts(srv)
5.  for all prt in puertos do
6.   op= obtenerOperaciones(prt)
7.   numOp = numOp + op.sizeOf()
8.  end for
9.end for
10.return numOp
```

- **Extracción de comentarios:** los comentarios pueden encontrase en cualquier parte del código del WSDL, motivo por el cual se debe realizar un completo del árbol. Dicho recorrido debe ser de tipo Depth-First en Pre-Order de modo de obtener los comentarios en el orden original en el cual aparecen en el WSDL. A continuación se muestra, en pseudocódigo, la idea del algoritmo utilizado:

Algoritmo 2: Obtener comentarios

Entrada: *nodo* un nodo del AST asociado a un WSDL
cl lista donde se almacenan los comentarios
Salida: cl

```
1.comentario=obtenerComentario(nodo)
2.cl.append(comentario)
3.hijos = obtenerHijosEnOrden(nodo)
4.for all hijo in hijos do
5. Algoritmo 2(hijo, cl)
6.end for
```

En la línea 1 se obtienen los comentarios del nodo visitado y, en la línea siguiente, se inserta el mismo al final del parámetro *cl***.** En la línea 3 se obtienen todos los hijos del nodo actual ordenados de izquierda a derecha y, a partir de la línea 4 hasta la línea 6, se invoca recursivamente al método para continuar con la extracción de los comentarios. Una vez finalizado el recorrido recursivo del árbol, se poseen todos los comentarios ordenados en *cl*.

- **Espacios de nombres (EdN):** los espacios de nombres se encuentran al inicio de los WSDL, en el elemento *definitions* (o *descriptions* para la versión WSDL 2.0). Por ser el de mayor jerarquía en el WSDL, este elemento siempre será la raíz de su AST asociado. Por lo tanto, sólo se debe acceder al nodo raíz para obtener los EdN. En el caso del AST generado por el parser DOM, cada espacio de nombre representa un atributo del nodo, se hace necesario entonces, analizar todos los atributos del nodo raíz para obtener todos los EdN. La idea utilizada se muestra a continuación:

Algoritmo 3: Obtener espacios de nombres

Entrada: *wsdlRaiz* el nodo raíz del AST asociado a un WSDL.
Salida: EdN una lista que contiene todos los espacios de nombres de *wsdlRaiz*

```
1.EdN = {};
2.atribArray = extrAtribs(wsdlRaiz)
3.for all atr in atribArray do
4. atributo = getText(atr)
5. EdN.append(atributo)
6.end for
7.return EdN
```

En la línea 2 del Algoritmo 3 se extraen todos los atributos que posee en nodo wsdlRaiz (correspondiente al elemento definitions del WSDL). Luego, desde la línea 3 a la línea 6 se recorren cada uno de estos atributos, extrayendo su información y agregándola al final de la lista EdN. Finalmente, en la línea 7 se retorna EdN, la cual contiene todos los espacios de nombres del WSDL analizado.

3.2.3. Visualización de la información. La interfaz de SIPi provee distintas maneras de observar la información que extrae. Las métricas de cada WSDL analizado se insertan en una tabla de doble entrada. Cada columna de la tabla representa una métrica calculada y cada fila, un WSDL analizado. El uso de una tabla facilita el acceso rápido a todas las métricas para su análisis detallado.

En la Figura 5 se tiene una captura de dicha tabla luego de analizar cuatro WSDLs distintos. SIPi también provee una comparación de métricas mediante el uso de un gráfico. La principal ventaja del mismo consiste en que a simple viste se puede determinar que WSDL puede llegar a ser más complejo de analizar respecto de los demás (Ver Figura 6).

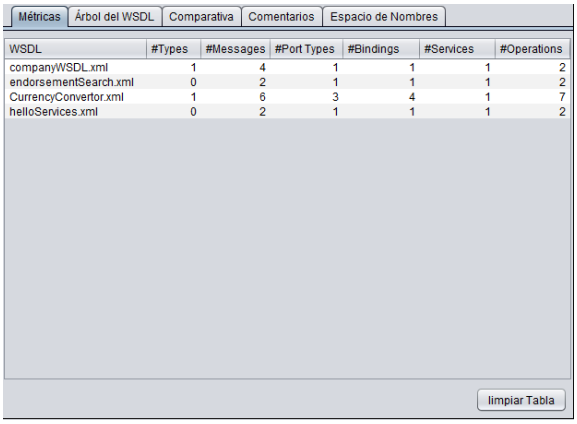


Figura 5. Tabla de métricas de SIPi para cuatro WSDLs distintos.

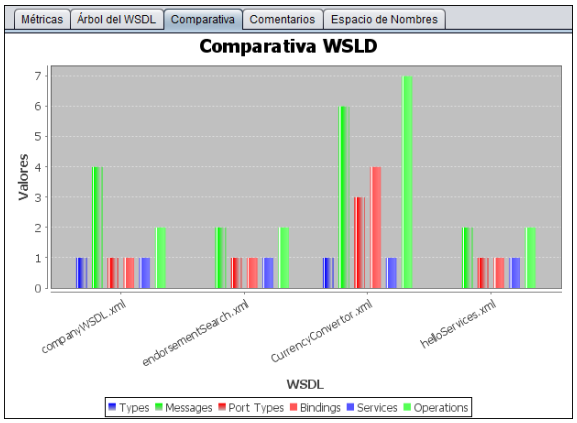


Figura 6. Gráfica comparativa de métricas de distintos WSDLs.

Además de la visualización de métricas, SIPi también utiliza su interfaz para visualizar otro tipo información extraída: los comentarios y los espacios de nombres. Ambos se visualizan mediante texto plano, separados por el nombre de cada WSDL. Esto permite el acceso directo a información que se debe mostrar tal cual figura en las especificaciones WSDL. De esta forma, se pueden leer todos los comentarios existentes en cada WSDL, tal como lo muestra la Figura 7, e intentar comprender la idea general del WS. La misma idea se aplica para los espacios de nombres (ver Figura 8).

3.2.4. Almacenamiento de información. Además del análisis en su propia interfaz, SIPi genera información persistente del AST, que permite continuar el análisis luego del final de la ejecución de SIPi. Dicha información actualmente consiste de un archivo de texto plano por cada WSDL analizado. Cada archivo contiene el nombre de cada nodo del AST junto con sus atributos y los valores que posee cada uno de ellos.

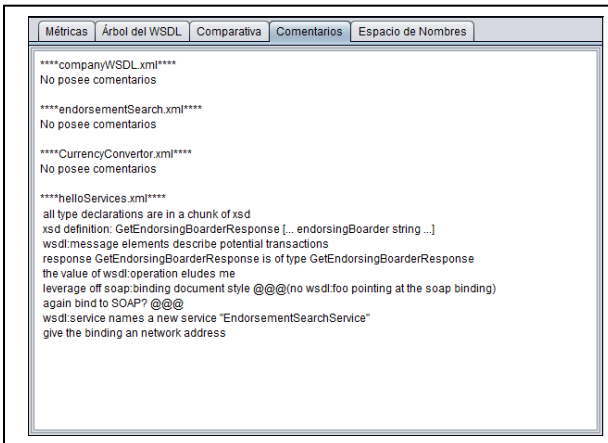


Figura 7. Visualización de los comentarios para cuatro WSDLs distintos.

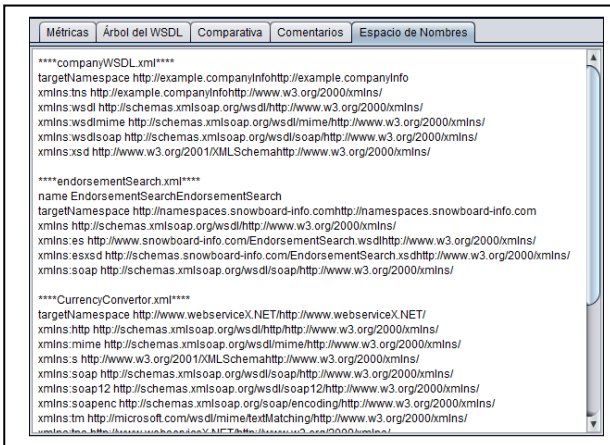


Figura 8. Ejemplo de un archivo de salida de SIPi.

4. Caso de estudio

Suponga que se tienen que desarrollar tareas de mantenimiento sobre un sistema de software que hace uso en sus funcionalidades de cuatro WSs distintos. Dichos WS, de nombres *companyWSDL*, *CurrencyConvertor*, *endorsementSearch* y *helloServices*, son los utilizados en la sección anterior para mostrar las capturas de SIPi. A modo de facilitar este análisis, se referencian dichas figuras. Mediante SIPi se puede comparar la complejidad de estos WSs en conjunto para detectar el nivel de complejidad que demandará su comprensión. Para esto, se ejecuta SIPi y mediante el botón *Examinar* se accede al seleccionador de archivos que permite buscar y seleccionar cada una de las

especificaciones WSDL de los WS y analizarlas. Una vez finalizada la búsqueda y selección, se puede proceder al análisis propiamente dicho. Por defecto, lo primero que se verá es la tabla comparativa de métricas de los WSDLs (ver Figura 5). Esto inserta de manera directa al ingeniero de software en el análisis de los números de cada WS, donde se puede tener una idea de cada métrica y evaluar las relaciones entre ellas. Por ejemplo, se puede observar que solo dos WSDL (*companyWSDL, CurrencyConvertor*) poseen tipos específicos mientras que los otros dos restantes no poseen. Además, todos los WS poseen un solo service, pero *CurrencyConvertor* posee mayor número de bindings y operaciones. Incluso, el número de operaciones de este último WSDL denota una sobrecarga y por lo tanto, podría ser sometido a tareas de mantenimiento perfectivo. De los análisis anteriores, se puede ir seleccionando a CurrencyConvertor como el WSDL de mayor complejidad de comprensión. Para confirmar o no está hipótesis, se puede ver la gráfica comparativa de métricas en donde se verá de manera más visual la diferencia. La misma se puede observar en la Figura 6. El resultado que refleja esta gráfica confirma la hipótesis antes mencionada, CurrencyConvertor es el WSDL con métricas más altas con lo cual se deduce que será el más costoso de comprender.

Para continuar el análisis, se dejan de lado un instante las métricas para analizar la información extraída por SIPi respecto de los comentarios y los espacios de nombres. Curiosamente, sólo un WSDL posee comentarios en su interior: *helloServices*. Esto significa que su comprensión posiblemente se facilitará al interpretar cada uno de esos comentarios. Además, si se vuelve a la tabla de métricas se observa que posee valores pequeños, por lo que no pareciera un gran desafío su comprensión. Finalmente, si se observa la información extraída respecto a los espacios de nombres (ver Figura 8), solo se obtiene una nueva prueba de la complejidad de *CurrencyConvertor*.

En base a lo analizado anteriormente usando SIPi, se puede llegar rápidamente a la conclusión de que *CurrencyConvertor y helloServices son los* WSs que mayor y menor dificultad de comprensión presentan en su especificación WSDL, respectivamente. Esto determina un rango de complejidad a la hora de realizar las tareas de mantenimiento sobre el sistema que los utiliza. Incluso permite determinar el costo de un cambio en base al WS que el mismo afecte.

5. Conclusiones y Trabajos Futuros

La Comprensión de Programas es una actividad central en el contexto de la Ingeniería de Software. Esto se debe a que la misma reduce en gran medida los costos y esfuerzo humano implicados en la tarea de Mantenimiento, Evolución y Migración de Software.

Actualmente, los servicios web han adquirido relevancia debido a su utilidad en la construcción de aplicaciones distribuidas, la cual es una tendencia que día a día se reafirma en el contexto del desarrollo de software.

Los servicios web son una clase particular de software y, como tal, también están sujetos a tareas de Mantenimiento, Migración y Evolución, y por ende también de comprensión. Una forma de abordar la comprensión de los WS consiste en analizar sus descripciones, las cuales están disponibles en diferentes repositorios con el propósito de que los usuarios de los WS las puedan acceder. Dichas descripciones están especificadas en un lenguaje basado en XML conocido con el nombre WSDL (Web Service Description Language). Desde el punto de vista del análisis de especificaciones WSDL, el hecho de que las descripciones de los WS estén realizadas en un lenguaje basado en XML es una ventaja. Esto se debe a que en la actualidad existen numerosas herramientas, cuyo funcionamiento es muy satisfactorio, que son necesarias para extraer información de las descripciones WSDL y elaborar estrategias de comprensión, tal es el caso de los analizadores sintácticos SAX y DOM. Dichos analizadores permiten automatizar una parte relevante del proceso de construcción de herramientas de comprensión como lo es: la construcción de una representación tanto a nivel conceptual (en el caso de SAX) como concretamente (en el caso de DOM) de las descripciones WSDL. Una vez llevada a cabo la tarea antes mencionada se debe pensar que información recuperar para elaborar estrategias de comprensión. Atento a este requerimiento, los autores de este trabajo recurrieron al cómputo de métricas debido a que las mismas proveen una forma, consensuada en la jerga de Ingeniería de Software, de poder vislumbrar la complejidad de una descripción WSDL y por lo tanto determinar la dificultad de comprender dichas descripciones. El cálculo de métricas es importante pero en igual grado también lo es la estrategia que se siga para mostrar la información proporcionada por dichas métricas. En este sentido, se apeló al estudio de técnicas de visualización de la información y se determinó que el uso de diagramas de barras es apropiado para visualizar el valor de métricas.

Teniendo en cuenta las observaciones mencionadas en los párrafos precedentes se construyó SIPi (Simple Inspection Program improved) una herramienta cuyo principal objetivo es posibilitar el análisis de especificaciones WSDL. SIPi calcula métricas orientadas a medir la complejidad de las descripciones WSDL y permite su visualización mediante diagramas de barra. Dichos diagramas tienen como finalidad facilitar el análisis por parte del ingeniero de software de la

complejidad de la descripción WSDL. Además de las tareas antes mencionadas, la herramienta: i) recupera los comentarios los cuales quedan disponibles para que sean procesados por otro tipo de herramientas como lo son aquellas que realizan Procesamiento de Lenguaje Natural y ii) provee funcionalidades de administración de proyectos las cuales posibilitan que un análisis de descripciones WSDL se puedan realizar en el tiempo que dispone el ingeniero.

Por otra parte, SIPi contempla el uso de diferentes versiones de especificaciones WSDL (actualmente reconoce las dos versiones más ampliamente usadas como lo son: WSDL 1.1 y WSDL 2.0) lo cual hace que sea una herramienta actualizada de amplio espectro. SIPi fue aplicado a un caso de estudio elaborado por los autores de este artículo y mostró ser fácil de manejar y efectivo en la muestra de resultados.

Entre los trabajos futuros a corto y largo plazo se pueden mencionar:

- Definir y calcular métricas más complejas que permitan inferir más información de los WSDL.
- Definir nuevas técnicas de visualización que faciliten la comprensión de la información extraída
- Detectar y analizar automáticamente relaciones entre WSDLs.
- Usar los resultados obtenidos con SIPi para iniciar el estudio de los WS implementados con BPEL (Business Process Execution Language). Esta tarea se llevará a cabo en el marco de una Tesis de Maestría y una investigación Postdoctoral. Ambos estudios son realizados en la Universidad Nacional de San Luis por docentes del área de Programación y Metodologías de Desarrollo de Software de la Universidad Nacional de San Luis.

6. Referencias

[1] A. Abran and H. Nguyenkim. "Analysis of maintenance work categories through measurement". In *Software Maintenance, 1991., Proceedings. Conference on*, pages 104–113, oct 1991.

[2] A. V. Aho, R. Sethi, and J. D. Ullman. "*Compilers Principles, Techniques and Tools*". Addison-Wesley, 1986.

[3] G. Alkhatib. "The maintenance problem of application software: an empirical analysis". *Journal of Software Maintenance*, 4(2):83–104, June 1992.

[4] L. J. Arthur. *"Software evolution: the software maintenance challenge"*. Wiley-Interscience, New York, NY, USA, 1988.

[5] B. W. Boehm. *"Software Engineering Economics"*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1981.

[6] T. A. Corbi. "Program understanding: challenge for the 1990's". IBM Syst. J., 28(2):294–306, June 1989.

[7] H. El Bouhissi, M. Malki, and D. Bouchiha. "A reverse engineering approach for the web service modeling ontology specifications". In *Sensor Technologies and Applications 2008. SENSORCOMM '08. Second International Conference on*, pages 819–823, Aug 2008.

[8] R. K. Fjeldstad and W. T. Hamlen. "Application program maintenance study: Report to our respondents". In *Proceedings GUIDE 48*, Apr. 1983.

[9] M. Fokaefs, R. Mikhaiel, N. Tsantalis, E. Stroulia, and A. Lau. "An empirical study on web service evolution". In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 49–56, July 2011.

[10] N. Gold and K. Bennett. "Program comprehension for web services". In *Program Comprehension, 2004. Proceedings. 12th IEEE International Workshop on*, pages 151–160, June 2004.

[11] H. Han and T. Tokuda. "A method for integration of web applications based on information extraction". In *Web Engineering, 2008. ICWE '08. Eighth International Conference on*, pages 189–195, July 2008.

[12] J.W. S. D. Language. http://sourceforge.net/projects/jwsdl/.

[13] B. P. Lientz and E. B. Swanson. *"Software Maintenance Management"*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1980.

[14] F. Liu, Y. Shi, J. Yu, T. Wang, and J. Wu. "Measuring similarity of web services based on wsdl". In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 155–162, July 2010.

[15] P. E. Livadas and D. T. Small. "Understanding code containing preprocessor constructs". In *IEEE Third Workshop on Program Comprehension*, pages 89–97, 1994.

[16] C. Mao. "Towards a data complexity metric set for web service composition". In *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*, pages 127–131, Aug 2011.

[17] L. O'Brien Lero and D. Smith. "Working session: program comprehension strategies for web service and service oriented architectures". In *Program Comprehension, 2004. Proceedings. 12th IEEE International Workshop on*, pages 260–261, June 2004.

[18] G. H. Okimoto. "The effectiveness of comments: A pilot study". IBM SDD 01.1347, July 1970.*Semantic Driven Program Analysis*, Kent State University, Kent, OH, USA, Doctoral Thesis, 2003.

[19] S. Project. http://www.saxproject.org.

[20] S. Rugaber. "Program comprehension", 1995.

[21] B. Shneiderman. "Measuring computer program quality and comprehension". *International Journal of Man-Machine Studies*, pages 465–478, 1977.

[22] T. A. Standish. "An essay on software reuse". *IEEE Trans. Software Eng.*, 10(5):494–497, 1984.

[23] E. B. Swanson and C. M. Beath. "Maintaining information systems in organizations". *John Wiley & Sons, Inc*., New York, NY, USA, 1989.

[24] A. von Mayrhauser and A. M. Vans. "Comprehension processes during large scale maintenance". In *Proceedings of the 16th international conference on Software engineering, ICSE* '94, pages 39–48, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[25] A. Von Mayrhauser and A. M. Vans. "Program comprehension during software maintenance and evolution." Computer, 28(8):44–55, 1995.

[26] World Wide Web Consortium (W3C). http://www.-w3.org/dom/faq.htmll.

[27] World Wide Web Consortium (W3C). http://www.-w3.org/tr/wsdl.

[28] L. Weissman. "Psychological complexity of computer programs: an experimental methodology." *SIGPLAN Not.*, 9(6):25–36, jun 1974.

[29] L. White, N. Wilde, T. Reichherzer, E. El-Sheikh, G. Goehring, A. Baskin, B. Hartmann, and M. Manea. "Understanding interoperable systems: Challenges for the maintenance of soa applications." In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 2199–2206, Jan 2012.

[30] Wikipedia, the free encyclopedia. http://en.wikipedia.org-/wiki/Web_Services_Description_Language

Un Sistema de Anotaciones para la Especificación de Componentes de una Línea de Productos de Software

Matias Pol'la, Maximiliano Arias, Agustina Buccella and Alejandra Cechich
GIISCO Research Group
Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue, Neuquén, Argentina
matias.polla, maximiliano.arias, agustina.buccella, alejandra.cechich@fi.uncoma.edu.ar
Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET

October 22, 2014

Abstract

Las Líneas de Productos de Software (LPS) han surgido como un paradigma para promover el reuso y mejorar los tiempos, costos y calidad de los productos obtenidos. A su vez, utilizar un enfoque basado en componentes para el diseño e implementación de los servicios requeridos por las mismas, proporciona un nivel aún mayor de reuso brindando mayor flexibilidad para la creación de nuevas o para la instanciación de las existentes. Sin embargo, tanto el diseño como la implementación de estos componentes posee una serie de desafíos que se deben considerar con especial interés para poder garantizar el reuso efectivo de los mismos. Así, en este trabajo, proponemos un sistema de anotaciones el cual se define mediante un modelo de metadatos que permite especificar la información necesaria sobre la semántica de los servicios implementados por cada componente, sus requerimientos técnicos, y sus variabilidades. Este sistema se aplica luego a componentes reales de una LPS previamente desarrollada para el dominio de ecología marina.

1 Introduction

El reuso de software es uno de los principales objetivos a alcanzar en todo desarrollo de software. No solo porque provee beneficios en cuanto a costos y tiempos, sino también porque facilita en gran medida el trabajo de etapas del desarrollo que consumen mucho trabajo y esfuerzo y son muchas veces relegadas en pos de otras que generan mayor visibilidad a los usuarios. El uso de artefactos, ya sean módulos, componentes, o subsistemas, previamente verificados y aplicados que sean luego fácilmente adaptados o integrados a nuevos sistemas en construcción, es uno de los temas de investigación que más auge ha tenido en los últimos años dentro de la ingeniería de software. Sin embargo, diseñar para el reuso es una tarea compleja ya que se deben considerar muchos aspectos en forma integral para garantizar justamente que los artefactos generados seas luego fácilmente reusados.

Dentro de los paradigmas más utilizados en la actualidad que apuntan al desarrollo de artefactos de software reusables son: el Desarrollo de Software Basado en Componentes [1] y las Líneas de Productos de Software (LPS) [2, 3, 4]. El primero, provee un conjunto de técnicas y metodologías para desarrollar nuevos sistemas a partir de la combinacin de componentes previamente creados y cuya funcionalidad ya fue comprobada. Busca producir software con el propósito de reusarlo en desarrollos futuros y en contextos distintos a aquellos para los que fue diseñado. El otro paradigma, LPS, se centra en dominios específicos con similares funcionalidades que serán luego adaptados a las necesidades específicas de los nuevos productos derivados a partir de ellas. El aspecto fundamental de toda LPS es llegar a la construcción de una arquitectura que permita incorporar componentes reusables, de modo que al desarrollar nuevos productos en lugar de hacerlo desde el principio, se instancien los artefactos necesarios. A su vez, a pesar de que las LPSs y el DSBC son dos paradigmas que han surgido con objetivos diferentes, pueden ser combinados para promover el reuso y facilitar el desarrollo de sistemas. Dicha combinación, aunque nos debería proveer de los beneficios de ambos en cuanto al reuso, tiempo, costo y calidad, nos heredarán inevitablemente sus complejidades. Por lo tanto, se debe poner especial énfasis en los aspectos más críticos de ambos paradigmas de manera de garantizar el éxito de un proceso de desarrollo de LPSs por medio de componentes. En particular, en este trabajo, hemos considerado dos de esos aspectos: la gestión de la variabilidad para las LPSs [5, 6, 7] y las especificaciones e información que pueda ser extraída desde los componentes [1, 8] para que puedan ser integrados y adaptados más fácilmente.

Con esto en mente, hemos creado un sistema de anotaciones aplicado a componentes JAVA que define un modelo de metadatos almacenando la información necesaria para representar variabilidades y especificaciones técnicas de los mismos. Dichas anotaciones contemplan además artefactos de diseño que han sido creados en trabajos previos [9], otorgándo a la LPS la trazabilidad necesaria para conocer la manera en que los requerimientos de los usuarios fueron diseñados e implementados. Tanto la definción del sistema de anotaciones, como su aplicación y uso son descriptos en este trabajo y aplicados a una LPS desarrollada en trabajos previos [9, 10, 11, 12]. Además, presentamos una herramienta prototipo que permite recuperar dichas anotaciones semánticamente para darles el uso necesario, tanto dentro del proceso de configuración y derivación de productos, como dentro del proceso de integración con otros componentes.

Este artículo está organizado de la siguiente manera. En la sección siguiente se presentan los trabajos relacionados y antecedentes. En la Sección 3 se detalla el sistema de anotaciones propuesto junto con su aplicación a componentes JAVA. Luego, en la Sección 4 se describe la aplicación de dicho sistema a componentes de la LPS previamente desarrollada y se describe la herramienta prototipo que hace uso de los mismos. Por último se enumeran las conclusiones obtenidas y los trabajos futuros.

2 Trabajos Relacionados y Antecedentes

En este trabajo debemos considerar dos líneas de investigación específicas: la gestión de la variabilidad y la integrabilidad de componentes. En cuanto a la primera, existen gran cantidad de trabajos y revisiones sistemáticas [6, 13] que proponen y analizan diferentes enfoques recientes. En particular, en [6] se realiza una revisión sistemática centrándose en la gestión de variabilidad utilizada en el contexto de la ingeniería de líneas de productos. En la misma se pueden observar diferentes enfoques que abordan la variabilidad en distintos momentos del ciclo de desarrollo de software como por ejemplo el modelado de la variabilidad, el soporte para la identificación de variabilidades y partes comunes de una LPS, el diseño de la arquitectura, la derivación de un producto, herramientas de soporte, entre otras. Dentro de los trabajos pioneros en variabilidad, podemos citar a FODA (Análisis de Dominio Orientado a Características) [14], el cual intenta capturar las características principales de un conjunto de sistemas similares pertenecientes a un dominio específico, diferenciando entre las características que son compartidas por todos los sistemas y las variabilidades. Otro trabajo, surgido a partir de FODA, es el Modelado de Características (FM-Feature Modeling) [15] que presenta un modelo de aplicación que permite definir tanto las variabilidades presentes como las partes comunes para soportar la derivación de los productos. En este sentido, FM presenta un modelo jerárquico, organizado en forma de árbol, que posibilita descomponer características de niveles superiores en características detalladas y específicas. Esta jerarquía impone ciertas restricciones de configuración al momento de llevar adelante la derivación de los productos, ya que la selección de una funcionalidad implica la selección del padre que la engloba. En [16] también se presenta un modelo de variabilidad jerárquica apoyado en la utilización del DSBC, debido a que este paradigma reduce la complejidad del diseño y reuso de software, distribuyendo el trabajo y la implementación del sistema en el desarrollo de diferentes componentes independientes.

Otro enfoques proponen la implementación de la variabilidad directamente sobre el código de un sistema, mediante la Programación Orientada Delta (POD) [17, 18]. En estos enfoques una LPS se representa por medio de un módulo núcleo y un conjunto de módulos delta. El primero define la plataforma común a todos los productos que pueden ser derivados de la línea, mientras que los "Deltas" especifican las posibles adaptaciones (variabilidades) para ajustarse a las necesarias particulares de cada producto. Cada delta genera cambios en el módulo principal (núcleo) a nivel de código tanto agregando, modificando o eliminando el mismo. Esta implementación para LPSs es más flexible y menos restrictiva que el modelado de características (FM) y otros enfoques modulares, ya que permite realizar modificaciones simples en el código. Además si se realiza una cuidadosa y detallada especificación y documentación de la línea, ésta puede brindar soporte para la trazabilidad de las variabilidades definidas, identificando qué delta implementa determinada funcionalidad.

En segundo lugar, debemos considerar aquellos trabajos relacionados con la especificación de componentes que ayuden a mejorar la integrabilidad de los mismos. En general existen una gran gama de trabajos en ésta línea, las cuales algunas de ellas se centran en las capacidades de composición de los componentes. El problema de composición dinámica ha sido abarcado desde distintas propuestas [19, 20, 21, 22] haciendo uso de archivos de configuración XML, programación orientada a aspectos, especificaciones FODA (Feature Oriented Domain Analysis) e incluso lenguajes formales como ADL. Por otro lado, el uso de modelos de metadatos se ha aplicado para asistir en un gran número de tareas relacionadas con el proceso de desarrollo [23, 24, 25, 26, 27] incluyendo testeo y validación, inclusión de descripciones en lenguaje natural para facilitar búsquedas en catálogos, configuración de herramientas, entre otras tareas.

En nuestro trabajo, considerando los trabajos menciona-

dos y de acuerdo a las particularidades de la LPS previamente desarrollada, se decidió realizar un sistema de anotaciones basado en un modelo de metadatos para gestionar la variabilidad y mejorar la especificación de los componentes. El sistema de anotaciones provee un enfoque mixto que intenta maximizar los beneficios tanto de los enfoques jerárquicos así como los que producen modificaciones directamente sobre el código. De esta manera el sistema de anotaciones permite gestionar la variabilidad a nivel de granularidad fina (a nivel de código), permitiendo una mayor flexibilidad y reduciendo el costo de derivación. A su vez, provee de anotaciones de metadatos que proveen una detallada especificación de los aspectos técnicos y semánticos de los componentes aportando al entendimiento y trazabilidad de la línea.

2.1 Antecedentes

En trabajos previos [9, 10, 11, 12] hemos desarrollado una LPS, en el dominio geográfico, que sirvió como una plataforma de servicios comunes, aplicados a todos los productos de la LPS, y servicios variables, aplicados solo a algunos productos. Dicha línea fue creada a partir de una metodología de desarrollo orientada a niveles de dominios [12, 28, 29] la cual combina ventajas de otras metodologías muy referenciadas tanto en el ámbito académico como en la industria [2, 3, 4]. Para la construcción de la línea se trabajó conjuntamente con dos organizaciones, el Instituto de Biología Marina y Pesquera "Almirante Storni"[1] y con el Centro Nacional Patagónico (CENPAT-CONICET)[2] las cuales brindaron el soporte necesario para la especificación y validación de las funcionalidades de los dominios analizados. A su vez, se han realizado esfuerzos para estandarizar los servicios requeridos dentro de los mismos de manera de mejorar la comunicación entre las partes y partir de diseños comunes y consensuados. Así, entre las partes interesadas (biólogos e informáticos) se definió una taxonomía de servicios y guías de uso de los mismos; ambas basadas en los estándares de servicios dentro del dominio geográfico. En particular se extendió la taxonomía definida en la ISO/DIS 19119[3] enfocándonos en los servicios del subdominio de ecología marina. Dichos servicios fueron utilizados para la migración del sistema a un enfoque de componentes de software [29] para permitir la implementación, utilización y ensamblaje de los mismos en forma independientemente. Estos trabajos nos permitieron desarrollar la línea de manera incremental agrupando las funcionalidades en componentes independientes de acuerdo a los servicios que implementan. La LPS cuenta con un arquitectura de tres capas, respetando los estandartes utilizados para los sistemas de información geográfica. En la Figura 1 se pueden observar dichas capas, las cuales involucran, la *Interfaz de Usuario, Procesamiento Geográfico* y *Modelo Geográfico*; y donde cada una de ellas esta conformada por un grupo de componentes que responden a diferentes servicios. De esta manera, cada uno de los componentes incluídos en la LPS se corresponde con un conjunto de servicios detallados en la taxonomía de servicios definida. Por ejemplo, en la figura podemos observar la funcionalidad "Visualizar estadísticas gegráficas" en donde vemos que el componente de *Estadísticas Geográficas*, pertenciente a la capa de *Procesamiento Geográfico*, implementa los servicios de *Determinar Características en diferentes zonas* (PS-S 2.1) y y *Determinar Características en diferentes estaciones* (PS-S 2.2); mientras que el componente *Estadísticas Gráficas*, pertenciente a la capa de *Interfaz de usuario* , implementa los servicios*Visualizar datos por Histogramas* y *Visualizar datos por tabla* (HI-GS1.1 y HI-GS1.1 respectivamente). Además, este componente presenta un *punto variante* que permite configurar el tipo de visualización deseada de los datos obtenidos.

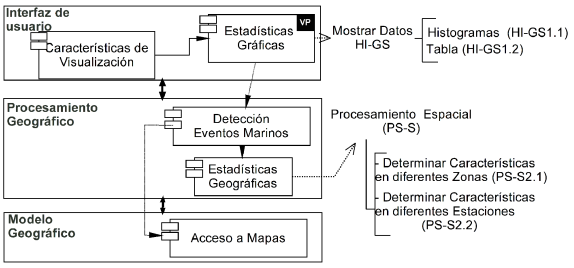


Figure 1: Arquitectura de la LPS junto diseño de componentes y servicios para la funcionalidad *visualizar estadísticas gegráficas*

El trabajo conjunto con el IBMPAS y el CENPAT-CONICET generó la derivación de dos productos de la LPS. Los servicios presentes en ambos productos generados son la visualización de las campañas, estaciones y zonas así como la visualización de diferentes estadísticas en diversos formatos. Además de contar con las funcionalidades comunes, el primer producto, implementado para el IBMPAS, cuenta con servicios de creación de histogramas para observar diferentes estadísticas de los fenómenos de estudio, junto con un modulo de detección de patrones de distribución de especies presentado en [11]. Un prototipo del primer producto puede visualizarse en `http://gissrv.fi.uncoma.edu.ar/SaoProjectUI/`. Por otro lado, el segundo producto posibilita la creación de diferentes mapas para visualizar dos o más zonas en paralelo. También presenta un componente de procesamiento encargado de analizar el avance (crecimiento y distribución) de determinadas especies a lo largo de un determinado periodo.

[1] http://www.ibmpas.org/
[2] http://www.cenpat.edu.ar/
[3] ISO/DIS 19119 : Geographic information Services, ISO/TS 2005

La implementación actual de la LPS fue realizada mediante una tecnología puramente basada en componentes utilizando Enterprise Java Beans [4] (EJB) junto con la herramienta Google Web ToolKit [5] (GWT) para ser utilizada bajo la Web. Esta tecnología e infraestructura desarrollada genera una estructura jerárquica que permite incrementar tanto el reuso efectivo de la plataforma, así como la modificabilidad de los componentes existentes y la creación e integración de los nuevos [9]. Por otro lado, la utilización de la herramienta GWT nos permite generar sistemas web a través de un proceso de compilación de un conjunto de componentes específicos implementados en Java, los que son traducidos a código JavaScript y HTML equivalente. De esta manera, se pueden distinguir dos tipos de componentes desarrollados. Por un lado, tenemos los componentes que conforman la capa de Interfaz de Usuario, implementados con GWT; y por otro lado los componentes EJB, implementados en lenguaje Java, que corresponden a la capa de Procesamiento Geográfico.

A pesar de que, todo el trabajo realizado ha generado buenos resultados a la hora de la derivación de los nuevos productos desde la LPS, todavía resta mucho trabajo para mejorar el reuso, trazabilidad e integrabilidad de los componentes diseñados e implementados. Además, se deben crear herramientas de soporte que, mediante la estandarización de los servicios y el diseño de los componentes, nos permitan asistir tanto al desarrollo de la plataforma de la LPS como a la derivación de nuevos productos.

3 Un Sistema de Anotaciones para Componentes de una LPS

El sistema de anotaciones es creado mediante la definición de un modelo de metadatos que nos permite representar la información estandarizada para los dominios involucrados incluyendo aspectos que ayuden a la gestión de la variabilidad e integrabilidad de los componentes implementados. Así, en esta sección describimos el proceso de definición de metadatos que hemos creado a nivel general y luego su aplicación a componentes Java dentro de las LPSs.

3.1 Proceso de definición del modelo de metadatos

En la Figura 2 podemos observar las tres etapas principales para crear los metadatos necesarios que conformaraán luego nuestro sistema de anotaciones dentro de los componentes. La figura muestra, a su vez, el detalle de las actividades de

[4]Oracle, http://www.oracle.com/technetwork/java/javaee/ejb/index.html
[5]http://code.google.com/intl/es-ES/webtoolkit/

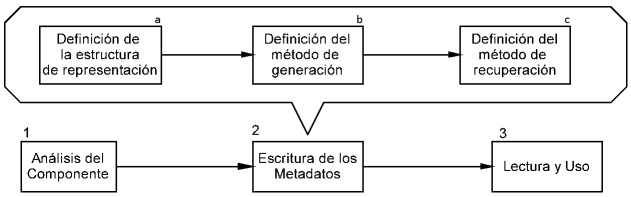


Figure 2: Proceso de definición del modelo de metadatos para el sistema de anotaciones de una LPS

la segunda etapa las cuales son necesarios para la especificación de otro proceso que permite definir, generar, y recuperar los metadatos definidos en los componentes.

Cada una de las etapas se describen brevemente a continuación:

1. *Análisis del componente*: Previo a definir los metadatos, los componentes deben ser analizados de acuerdo al conjunto de interfaces que le permiten describir sus funcionalidades y sus requerimientos de integración. De esta forma podemos identificar dos elementos: la interfaz funcional del componente [1] y los requerimientos (o interfaz) arquitecturales [30]. La primera describe el comportamiento del componente, incluyendo todos los atributos públicos, métodos, eventos emitidos y excepciones generadas. Por otro lado, los requerimientos arquitecturales definen las necesidades del componente para ser integrado al resto. Dentro de estos requerimientos podemos encontrar el lenguaje en el que se implementa el componente, puntos de entrada, dependencias externas (librerías, otros componentes, bases de datos, etc.), dependencias internas (en el caso que un componente esté compuesto por otros), etc. De esta manera el proceso de análisis debe identificar todos estos elementos para poder ser utilizados en el siguiente paso.

2. *Escritura de los metadatos*: Como el proceso de escritura de los metadatos requiere primeramente de una correcta definición del modelo a utilizar, hemos definido un proceso general para tal tarea. La Figura 2 muestra cómo se organiza dicho proceso, identificando tres actividades:

   (a) *Definición de la estructura de representación*: Durante esta actividad se realiza el análisis del problema que se desea resolver y a través de esto se obtiene el conjunto de metadatos que resulten pertinentes. Una vez definido dicho conjunto, se establece una estructura para representar los mismos. Existen distintos métodos de representación según las necesidades del problema que resolverán los metadatos. Por un lado, los

metadatos pueden almacenarse en archivos externos mediante el uso texto plano, archivos XML, entre otros. Por otro lado, puede suceder que se almacenen de manera interna al código haciendo uso de comentarios especialmente marcados, anotaciones, o alguna otra tecnología ofrecida por la plataforma sobre la que se desarrolla.

(b) *Definición del método de generación*: Una vez definida la estructura de representación se establece la forma en que los datos serán generados. Este proceso puede realizarse de manera manual, automática, o asistida. Esta tarea está muy relacionada a la plataforma sobre la cual se trabaja. Durante la misma se utiliza la estructura de representación definida en el paso anterior y se realiza un análisis tecnológico para seleccionar o construir el método que se usará para generar o escribir los metadatos.

(c) *Definición del método de recuperación*: Por último, de acuerdo a la estructura de representación y al método de generación establecido, se define un método de recuperación y uso de los metadatos generados. Este método se encuentra muy relacionado a los elementos anteriormente definidos y puede resultar necesaria la construcción de herramientas para tal fin. Al igual que en la actividad anterior, es necesario realizar un análisis tecnológico de la plataforma elegida y realizar la selección o creación del método de recuperación. Durante esta actividad puede surgir además la necesidad de desarrollar una herramienta capaz de leer y mostrar los metadatos escritos.

3. *Lectura y uso*: La etapa final del proceso es la lectura de los metadatos escritos y el uso de los mismos para el fin que hayan sido creados. De esta manera el proceso de lectura se adapta a lo establecido durante el proceso de definición del modelo y el uso estará definido por la tarea que se quiera cumplir. Esta tarea está íntimamente relacionada con el problema a resolver, como por ejemplo, la automatización o asistencia en la ejecución de un proceso, configuración de herramientas, trazabilidad, entre otras.

3.2 Implementación del sistema de anotaciones utilizando JAVA para componentes de una LPS

El proceso definido en la sección anterior (Figura 2) puede ser aplicado directamente a componentes JAVA considerando a su vez particularidades de las LPSs. Así, en cada una de las etapas se obtienen una serie de diferentes artefactos que ayudarán a la construcción y uso de los metadatos y definirán el sistema de anotaciones propuesto.

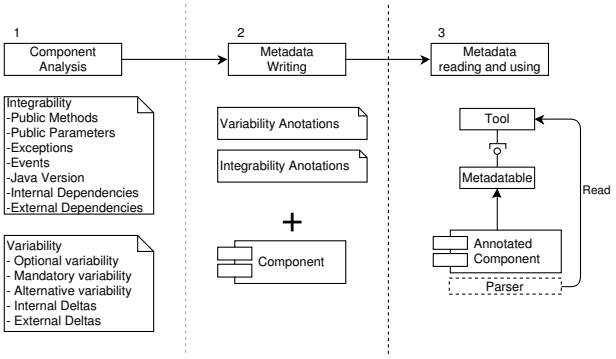


Figure 3: Aplicación de los metadatos para crear el sistema de anotaciones a componentes de una LPS

En la primera etapa, *análisis del componente*, debido a que JAVA no define por defecto una estructura para la creación de componentes, toda información de interfaz, excepciones y eventos generados está relacionada a los métodos públicos de las clases del componente. Si bien JAVA permite el uso del framework provisto por Enterprise Java Beans para desarrollo de componentes, la única diferencia con lo antes mencionado es que los métodos públicos se encuentran centralizados en un único archivo de interfaz. Las dependencias arquitecturales de un proyecto JAVA pueden encontrarse en sus archivos de configuración (properties,classpath, etc). Estos archivos suelen generarse automáticamente por los entornos de desarrollo utilizados como Eclipse[6] o NetBeans[7] y de ellos se puede conseguir una representación escrita de los elementos antes mencionados.

En la segunda etapa, *escritura de los metadatos*, se deben definir los metadatos necesarios para la escritura de los elementos identificados en la etapa anterior. Así, las tres actividades involucradas (Figura 2) se aplicaron al desarrollo de metadatos para componentes JAVA dentro de las LPSs y se detallan a continuación:

1. *Definir la estructura de representación de los metadatos*: Como el sistema de anotaciones persigue dos objetivos principales, lograr la representación y gestión de la variabilidad y promover la integrabilidad de los componentes, se definieron metadatos específicos para cada uno. Para el caso de la representación de la variabilidad hemos definido tres metadatos principales: *los deltas, los puntos variantes y los tipos de variabilidad requeridos* (Figura 4). Los primeros

[6] https://www.eclipse.org/
[7] https://netbeans.org/

son las modificaciones que sufre el código para posibilitar la instanciación correcta. Estos deben corresponderse con variantes externas, que son aquellas que son implementadas en componentes externos o subcomponentes, o con variantes internas asociadas a variabilidades implementadas en el mismo componente. Los segundos, representan el lugar donde debe instanciarse o seleccionarse una variante correspondiente a una determinada variabilidad, es decir, el punto configurable de acuerdo a las necesidades del usuario. Por último, los *tipos de variabilidad* son importantes ya que presentan diversas restricciones al momento de ser instanciados:

- ***Variabilidad Obligatoria:*** la instanciación de un determinado producto para el punto variante asociado no puede ser vacía.
- ***Variabilidad Opcional:*** la instanciación de las variabilidades definidas pueden estar o no presentes en el producto derivado. Este tipo de variabilidad, permite instanciar más de una de las opciones posibles.
- ***Variabilidad Alternativa:*** sólo es posible instanciar a lo sumo una de las alternativas presentadas.

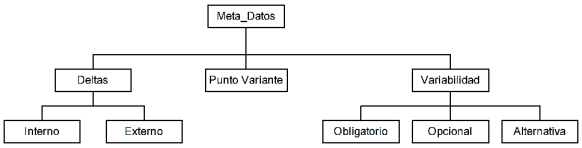


Figure 4: Metadatos para la representación de la variabilidad

Para el caso de los *Deltas*, tanto internos como externos, se almacena un *id* que lo identifica unívocamente, un *código* para invocar el método correspondiente, el conjunto de *librerías* utilizadas por el método junto con una *descripción* de la variante particular correspondiente al Delta. Además, para los *Deltas Externos* se debe registrar el/los *componente/s* donde se implementa la variante correspondiente. Luego, para cada uno de los metadatos de *Puntos Variantes* sólo es necesario indicar el *tipo de variabilidad*. Éste determina las restricciones al momento de realizar la configuración de cada punto variante. Para los metadatos de *Tipos de Variabilidad* (*Opcional, Alternativa Obligatoria*), se almacenan el *código de servicio* determinado por la taxonomía de servicios [9], junto con una *descripción* del mismo. Además se registra un *conjunto de pares ordenados* de las posibles variantes junto con los *Deltas* que implementa cada una de ellas. En el caso de que el tipo de variabilidad sea *Obligatoria*, se debe indicar el delta por defecto.

Por otro lado, los metadatos propuestos para la integrabilidad se asocian principalmente con los aspectos de interfaz funcional y arquitectural de los componentes. Con respecto a la interfaz funcional, es fundamental conocer la información pública del componente, es decir aquellos elementos que utiliza para comunicarse con otros componentes y para ser configurado. De esta manera, es necesario conocer los elementos que describen el comportamiento de un componente, incluyendo los métodos públicos, las variables de configuración o paramétros públicos, los eventos registrados y las excepciones que puede lanzar. La interfaz arquitectural se refiere a todos aquellos requisitos del componente necesarios para poder ser acoplado a otros. Esta categoría incluye la versión de JAVA mínima para una correcta ejecución, las dependencias internas (otros componentes de la línea con los que se relaciona) y las dependencias externas (librerías, componentes de terceros, bases de datos, etc.).

Para estructurar todos estos metadatos, de variabilidad e integrabilidad, se decidió utilizar un esquema de dos elementos: una etiqueta y los datos relacionados a la misma. A su vez, esta etiqueta está formada por dos componentes: una categoría y una subcategoría. La etiqueta permite identificar los metadatos y clasificarlos según el ámbito al que se refieren. La Tabla 1 muestra el modelo final de metadatos propuesto para aportar a los dos objetivos. Por ejemplo, podemos ver que el metadato *FuncionalInterface/Method* se encarga de mantener toda la información relacionada con la signatura de los métodos públicos definidos en un componente. Con respecto a la variabilidad, podemos observar el metadato variability/optional utilizado para demarcar un punto variante opcional, permitiendo definir los deltas y las instancias asociadas al mismo.

2. *Definir el método de generación*: Debido a que estamos considerando componentes en la plataforma JAVA, hemos elegido método de generación acorde a la misma, haciendo uso de las librerías que dan soporte a JAVA Annotations[8]. Éstas permiten insertar diferentes directivas a nivel de código y están estructuradas utilizando el carácter especial "@" junto con un nombre o etiqueta que las identifica. Las anotaciones cuentan con un conjunto de parámetros y presentan un objetivo y una política de retención. En nuestro caso utilizamos la política de retención "*source*" ya que sólo nos interesa mantener la anotación a nivel de código; y como objetivo solo utilizamos "*method*" ya que para nuestra estructura necesitamos anotar métodos. Por ejemplo, en el Código 1 se puede observar la implementación para la definición

[8] http://docs.oracle.com/javase/tutorial/java/annotations/

Etiqueta	Sub-Etiqueta	Datos Contenidos
Integrability	FunctionalIfc/Method	Métodos públicos del componente
Integrability	FunctionalIfc/PublicParameter	Parámetros públicos del componente
Integrability	FunctionalIfc/Events	Eventos generados por el componente
Integrability	FunctionalIfc/Exceptions	Excepciones que pueden generarse
Integrability	ArchitecturalIfc/JavaVersion	Versión de Java mínima necesaria
Integrability	ArchitecturalIfc/InteralDependencies	SubComponentes utilizados
Integrability	ArchitecturalIfc/ExternalDependencies	Librerías, componentes, bases de datos, etc.
Variability	Optional	Atributos para definir Variabilidad Opcional
Variability	Mandatory	Atributos para definir Variabilidad Obligatiria
Variability	Alternative	Atributos para definir Variabilidad Alternativa
Variability	Variant	Atributos para definir Variabilidad Variante
Variability	Delta_External	Parámetros, librerías y componentes externos
Variability	Delta_Internal	Definición de parámetros y librerías

Table 1: Modelo de metadatos defindo para componentes de una LPS

de la anotación *Meta_VariabiltyOptional* correspondiente al metadato *Variability/Optional* (Tabla 1). En la misma se distinguen la política de retención utilizada y el objetivo junto con los parámetros correspondientes. Este estilo de implementación se utilizó indistintamente, tanto para variabilidad como para integrabilidad. Cada una de las anotaciones definidas varia en el objetivo anotado, su nombre y sus paramétros. De esta manera, todas las anotaciones implementadas comparten la misma política de retención.

```
import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface Meta_VariabilityOptional
{
  public String serviceCode();
  public String serviceDescription();
  public String []instance();
  public String []deltaId();
}
```

Código 1: Implementación de una anotación java personalizada.

En la Figura 5 puede observarse el sistema de anotaciones definido de acuerdo al modelo de metadatos analizado previamente (Tabla 1). Esta estructura está, al igual que el modelo de metadatos, también dividida en dos grupos de anotaciones principales, aquellas relacionadas con la variabilidad y aquellas relacionadas con la integrabilidad. A su vez cada uno de ellos se encuentra subdivido en dos categorías. En primer lugar, las anotaciones relacionadas con la variabilidad se dividen en *Meta_VariantPoint* y los *Meta_Deltas* que se corresponden con los elementos *Punto Variante* y *Deltas* de la Figura 4. La primera categoría está formada por las anotaciones *@Meta_VariabilityOptional, @Meta_VariabilityMandatory y @Meta_VariabilityAlternativa*. Estas tres anotaciones son aplicadas para determinar los puntos variantes con su respectiva variabilidad. Cada una de ellas cuenta con los datos descriptos anteriormente. Cabe destacar que por limitaciones propias de la herramienta Java Annotations, el conjunto de pares (*Instancia, Delta*), el cual almacena las posibles variantes a través de la relación entre deltas e instancias, fue implementado por medio de dos arreglos simples de una dimensión, como puede observarse en la Figura 1. La segunda categoría,involucra a las anotaciones *@Meta_DeltaInternal y @Meta-_DeltaExternal*que son las encargadas de representar las diferentes modificaciones que sufre una LPS a nivel de código para cada variabilidad implementada. La distinción entre variante Interna y Externa nos permitirá generar una herramienta de derivación de productos flexible, que denote un rango de configuraciones lo suficientemente amplio para abarcar tanto los pequeños cambios en el código (granularidad fina) así como la utilización y composición de componentes reusables (granularidad gruesa). A su vez, estas anotaciónes también se definen a nivel de métodos y están representadas por un Id (DeltaId) que lo identifica unívocamente, por el código para invocar el método (CallerCode), las librerías(CallerImports) y componentes necesarios para su ejecución (Components), además de una descripción específica del Delta (InstanceDescription) que indica las particularidades del mismo.

El segundo grupo de anotaciones, que engloba todas aquellas relacionadas con la integrabilidad, se divide nuevamente en las dos categorías, la interfaz funcional y la interfaz arquitectural. La primera de las categorías incluye las anotaciones *@Meta_Method, @Meta_PublicParameter, @Meta_Events y @Meta_Exception*. Al igual que las anteriores cada una de ellas posee un conjunto de parámetros que las describe. *@Meta_Method* se escribe sobre todos los métodos públicos del componente y almacena su nombre, su lista de parámetros y su tipo de retorno. *@Meta_PublicParameter* anota aquellos parámetros públicos del componente que funcionan como variables de configuración, y posee el nombre y el tipo de los mismos. La anotacióon *@Meta_Events* almacena aquellos eventos que pueden

ser generados por el componente y almacena su nombre, sus parámetros y el tipo del manejador necesario. Por último, *@Meta_Exception* almacena información referida a todos los tipos de excepciónes que pueden dispararse durante la ejecución del componente. La categoría de interfaz arquitectural incluye las anotaciones *@Meta_JavaVersion, @Meta_ExternalDepencies y @Meta_InternalDependencies*. La primera de ellas, almacena la versión mínima de JAVA necesaria para el correcto funcionamiento del componente. La segunda almacena todas las dependencias externas del componente (librerías, bases de datos, componentes de terceros, etc) necesarias para que el componente pueda realizar sus tareas. Finalmente, *Meta_InternalDependencies* se encarga de almacenar toda la información relacionada con los componentes de la línea utilizados por el componente para llevar a cabo sus tareas.

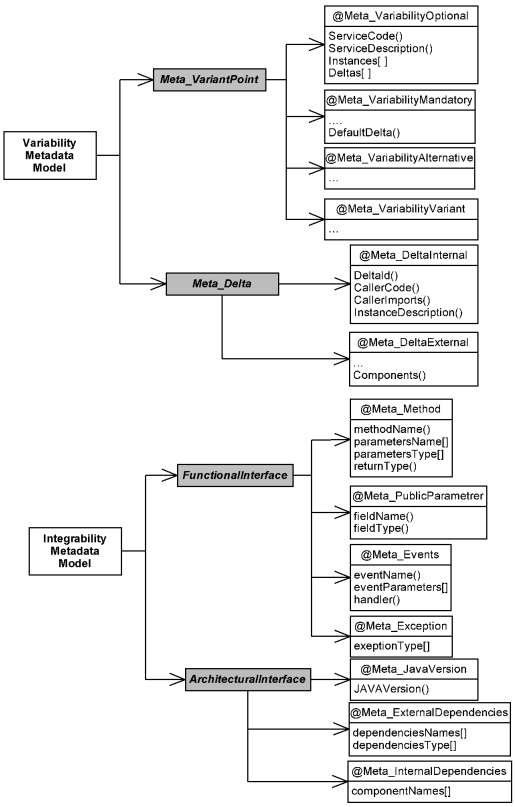


Figure 5: Sistema de anotaciones para los metadatos definidos en la Tabla 1

Como puede observarse los puntos variantes no son representados por medio de anotaciones ya que estos sólo indican el lugar puntual donde debe situarse la variabilidad. En la siguiente sección se explica como son representados estos puntos a nivel de código.

3. *Definir el método de recuperación*: Al tomar la decisión sobre la generación de los metadatos, se condiciona la definición del método de recuperación. Debido a que los metadatos se escriben de manera interna utilizando anotaciones y no se utilizan archivos externos para almacenarlos, se necesita diseñar un método de recuperación de los mismos. Para esto, y con el fin de facilitar la tarea de desarrollo de herramientas que puedan leer y usar estos metadatos se desarrolló un Parser de código JAVA capaz de extraerlos y volcarlos en objetos de un tipo predefinido "Metadato". El Parser definido sólo requiere que las clases que los utilizan implementen la interfaz "Metadatable". Se creó además una herramienta que respeta estos requerimientos y permite visualizar en forma de tabla los metadatos contenidos en un archivo JAVA.

Para la tercera etapa, *lectura y uso*, se desarrolló una herramienta de lectura en JAVA capaz de interpretar la interfaz "Metadatable". Su objetivo es realizar la lectura de los metadatos escritos en el componente haciendo uso del parser definido. Esta herramienta no realiza procesamiento alguno sobre los elementos leídos, definiendo una "plantilla" para el desarrollo de otros sistemas que sí deseen realizarlo. Todos los elementos descriptos anteriormente (anotaciones, interfaz y parser) se empaquetaron en una librería JAR, para ser importados por aquellos proyectos que quieran utilizarlos.

Estos dos últimos pasos, se describirán en detalle en la sección siguiente aplicados a componentes de la LPS previamente desarrollada.

4 Caso de Estudio

Una vez definido el sistema de anotaciones, se aplicó el mismo a la implementación de algunos servicios de la LPS desarrollada en trabajos previos. Para ello fue necesario modificar el conjunto de algunos de estos componentes que conforman la plataforma de la LPS, incorporando los metadatos de variabilidad e integrabilidad según corresponda. Luego se desarrolló una herramienta prototipo capaz de leer y mostrar los diferentes metadatos presentes en los componentes y asistir al proceso de derivación de nuevos productos. Esta herramienta facilita principalmente la ejecución de dos tareas. Por un lado permite utilizar los datos recolectados para mejorar la especificación de los componentes, sin necesidad de aplicar técnicas de ingeniería inversa; y por el otro lado, asiste al proceso de derivación.

En el ejemplo presentado en el Código 2 se muestra el código del componente *Estadísticas Gráficas* (Figura 1) que incluye el modelo de metadatos definido. En el mismo se pueden observar las anotaciones asociadas a la integrabilidad como *@Meta_PublicParameter* y *@Meta_Method*.

Además se puede ver el método dummy *variantPoint-MostrarDatos* que indica el Punto Variante para la variabilidad *"Mostrar Datos"*, junto con una de las posibles variantes (*"Visualización en forma de histogramas"*) representada por *Delta1*.

```
@Meta_JavaVersion("1.6")
@Meta_InternalDependencies(
componentNames = {"DeteccionDeEventosMarinos"})
@Meta_ExternalDependencies(
  names = {"Sencha","GWT-1.5"},
  types = {"Library","Framework"})
@Meta_Exception(
  exceptionType={"DataNotFoundException")
public class EstadisticasGraficas implements
    Metadatable{
  @Meta_PublicParameter
    (name="MAX_GRAPHS",type="int")
  public int MAX_GRAPHS;

  @Meta_Method(
    methodName="variantPointMostrarDatos",
    parameterNames={},
    parameterTypes={},
    returnType="void")
  @Meta_Variability_Optional (
    ServiceCode = "HI-GS",
    serviceDescription= "Visualizacin de datos
        estadsticos.",
    instance={"inst1"},
    deltaId={"Delta1","Delta2"})
  public void variantPointMostrarDatos()
    throws DataNotFoundException
  {...}

  @Meta_Delta_Internal(
    DeltaId = "Delta1'"
    CallerImports = {"DeteccionDeEventosMarinos"}
    CallerCode = {"Delta1(Parameters);"}
    InstanceDescription = "Visualizaci\'on en
        forma de histogramas")
  private void Delta1()
  {...}
  //More Code..
}
```

Código 2: Ejemplo código JAVA con anotaciones de integrabilidad y variabilidad aplicadas

En la Figura 6 podemos observar la visualización del resultado de la herramienta prototipo del sistema de anotaciones, aplicado al componente *Estadísticas Gráficas*. En dicha figura se pueden ver los datos de *InternalDependencies, Exception, PublicParameter, JavaVersion y ExternalDependecies* pertenecientes a los metadados de integrabilidad. Además podemos observar el *Delta_Internal* perteneciente a los metadatos de variabilidad, el cual esta asociado al "Delta1" correspondiente al servicio de *"Detección de Eventos Marinos"*.

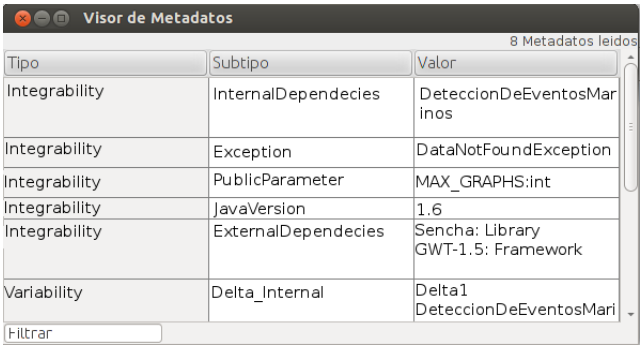


Figure 6: Herramienta de lectura del sistema de anotaciones

La herramienta desarrollada permite a su vez asistir al proceso de derivación de productos de una LPS. Esta herramienta recibe como entrada la información referente a la variabilidad recuperada de la lectura de los metadatos que, una vez obtenidos, ejecuta un proceso de reestructuración interno de los mismos para ser mostrados al usuario. En la Figura 7, puede observarse una interfaz de dicha asistencia, la cual presenta, a la izquierda, la taxonomía de servicios geográficos, y a la derecha los servicios pertenecientes a cada categoría indicando el tipo de variabilidad correspondiente. Por ejemplo, en este caso se observa para la categoría *InteracciónHumana/VisualizadorGeográfico/Manipulación de Mapas* la variabilidad *Opcional* definida para el servicio *AgregarMapa* junto con sus variantes (*Clonar Mapa* y *Manejo y almacenamiento de múltiples mapas*).

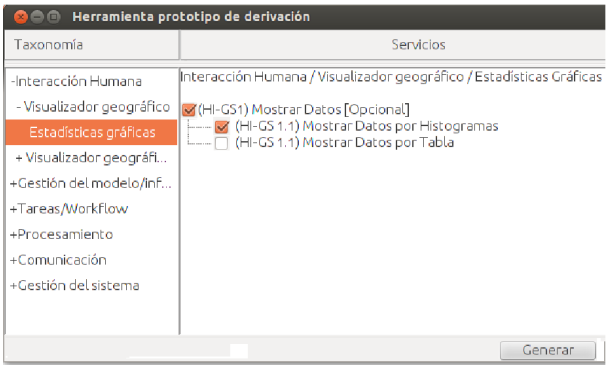


Figure 7: Intefaz para asistir al proceso de derivación de productos de la LPS en base al sistema de anotaciones implementado en los componentes

El resultado de este proceso de asistencia es el código instanciando las variantes seleccionas de todas las variabilidades presentes. El sistema de anotaciones definido está configurado para que luego de realizar la derivación de un determinado producto, sólo permanezcan en el código aquellos metadatos que asisten a la integrabilidad de componentes,

eliminando aquellos que definen la variabilidad. Esto implica que el producto derivado tendrá un código más legible y limpio, y a su vez contar con la información necesaria para permitirá añadir las funcionalidades particulares de cada producto. De esta forma, el modelo de metadatos de queda parcialmente oculto a los potenciales productos derivados.

5 Conclusiones y Trabajo Futuro

En este trabajo hemos definido un sistema de anotaciones sobre componentes JAVA que asisten a la gestión de la variabilidad y a la integrabilidad de los componentes. El principal aporte del sistema de anotaciones es un modelo de metadatos generado, el cual contempla la información más relevante a ser mantenida dentro de los componentes para maximizar su uso, tanto en la derivación de los productos como en la integración de unos con otros. Nuestros trabajos previos, en donde hemos generado varios artefactos para el desarrollo una LPS basada en componentes reusables, nos brindaron el soporte necesario para aplicar el sistema de anotaciones sobre experiencias reales que puedan medir el impacto de la aplicación del mismo y los beneficios obtenidos. Si bien sabemos que el uso de las anotaciones agrega una tarea más a los desarrolladores y es susceptible a errores cometidos al completar los metadatos, simplifica luego la tarea de un análisis automatizado mediante una herramienta que dependa de ellos. Para ilustrar esto, hemos desarrollado una herramienta prototipo que utiliza el sistema de anotaciones para soportar el proceso de derivación de nuevos productos en una LPS y permite recuperar sus requerimientos técnicos y semánticos.

Como trabajo futuro, se propone construir, por un lado, herramientas que asistan al proceso de anotación de los componentes para minimizar la posible tasa de error en el rellenado de los mismos. A su vez, es necesario conducir tareas de validación que permitan evaluar este sistema de anotaciones mediante dos parámetros principales: medir costo y esfuerzo de reimplementar el sistema de anotaciones sobre componentes previamente desarrollados, y medir los beneficios obtenidos a la hora de derivar nuevos productos y utilizar su especificación para fines como la trazabilidad o composición.

Acknowledgment

We would like to thank the two organizations that are collaborating in this proyect: IBMPAS and CENPAT-CONICET. This work is partially supported by the UNComa project “Reuso de Software orientado a Dominios” part of the program “Desarrollo de Software Basado en Reuso”.

References

[1] C. A. Szyperski, *Component software - beyond object-oriented programming*. Addison-Wesley-Longman, 1998.

[2] K. Pohl, G. Böckle, and F. J. v. d. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[3] P. C. Clements and L. Northrop, *Software Product Lines : Practices and Patterns*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[4] J. Bosch, *Design and use of software architectures: adopting and evolving a product-line approach*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000.

[5] M. A. Babar, L. Chen, and F. Shull, “Managing variability in software product lines,” *IEEE Software*, vol. 27, pp. 89–94, 2010.

[6] L. Chen and M. A. Babar, “A systematic review of evaluation of variability management approaches in software product lines,” *Information and Software Technology*, vol. 53, no. 4, pp. 344 – 362, 2011.

[7] M. Svahnberg, J. van Gurp, and J. Bosch, “A taxonomy of variability realization techniques: Research articles,” *Software-Practice & Experience*, vol. 35, no. 8, pp. 705–754, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1002/spe.v35:8

[8] G. T. Heineman and W. T. Councill, Eds., *Component-based software engineering: putting the pieces together*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[9] A. Buccella, A. Cechich, M. Arias, M. Pol’la, M. del Socorro Doldan, and E. Morsan, “Towards systematic software reuse of gis: Insights from a case study,” *Computers & Geosciences*, vol. 54, no. 0, pp. 9 – 20, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098300412003913

[10] M. Pol’la, A. Buccella, A. Cechich, and M. Arias, “Un modelo de metadatos para la gestió de la variabilidad en líneas de productos de software,” in *Proceedings of the ASSE’14: 15th Simposio Argentino de Ingeniería de Software*, Buenos Aires, Argentina, 2014.

[11] M. Pol’la, A. Buccella, A. Cechich, S. Doldan, E. Morsan, and M. Arias, “Detección de patrones de distribución en ecología marina: Un caso de estudio,”

in *Proceedings of the CONAIISI'13: 1st Congreso Nacional de Ingeniería Informática*, Córdoba, Argentina, 2013.

[12] P. Pernich, A. Buccella, A. Cechich, S. Doldan, E. Morsan, M. Arias, and M. Pol'la, "Product-line instantiation guided by subdomain characterization: A case study," *Journal of Computer Science and Technology, Special Issue 12(3)*, vol. 12, no. 3, pp. 116–122, 2012.

[13] M. Galster, D. Weyns, D. Tofan, B. Michalik, and P. Avgeriou, "Variability in software systems - a systematic literature review," *IEEE Transactions on Software Engineering*, vol. 40, no. 3, pp. 282–306, 2014.

[14] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA., Technical Report CMU/SEI-90-TR-21, 1990.

[15] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid, and A. Wäsowski, "Cool features and tough decisions: a comparison of variability modeling approaches," in *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*, ser. VaMoS '12. New York, NY, USA: ACM, 2012, pp. 173–182.

[16] A. Haber, H. Rendel, B. Rumpe, I. Schaefer, and F. van der Linden, "Hierarchical variability modeling for software architectures," in *Software Product Lines - 15th International Conference*, 2011, pp. 150–159.

[17] M. Helvensteijn, R. Muschevici, and P. Wong, "Delta modeling in practice: A fredhopper case study," in *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*, ser. VaMoS '12. New York, NY, USA: ACM, 2012, pp. 139–148.

[18] I. Schaefer, L. Bettini, F. Damiani, and N. Tanzarella, "Delta-oriented programming of software product lines," in *Proceedings of the 14th International Conference on Software Product Lines: Going Beyond*, ser. SPLC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 77–91. [Online]. Available: http://dl.acm.org/citation.cfm?id=1885639.1885647

[19] I. Kim, D. Bae, and J. Hong, "A component composition model providing dynamic, flexible, and hierarchical composition of components for supporting software evolution," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1797 – 1816, 2007.

[20] H. Mei, F. Chen, O. Wang, and Y. Feng, "Abc/adl: An adl supporting component composition," in *Proceedings of the 4th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering*, ser. ICFEM '02. London, UK, UK: Springer-Verlag, 2002, pp. 38–47. [Online]. Available: http://dl.acm.org/citation.cfm?id=646272.685804

[21] G. Succi, R. Wong, E. Liu, and M. Smith, "Supporting dynamic composition of components," in *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, 2000.

[22] S. Thaker, D. Batory, D. Kitchin, and W. Cook, "Safe composition of product lines," in *Proceedings of the 6th International Conference on Generative Programming and Component Engineering*, ser. GPCE '07. New York, NY, USA: ACM, 2007, pp. 95–104. [Online]. Available: http://doi.acm.org/10.1145/1289971.1289989

[23] M. Brohi and F. Jabeen, "A metadata-based framework for object-oriented component testing," *International Journal of Computer Applications*, vol. 41, no. 15, pp. 8–18, March 2012, published by Foundation of Computer Science, New York, USA.

[24] J. C. Grundy, "Aspect-oriented requirements engineering for component-based software systems," in *Proceedings of the 4th IEEE International Symposium on Requirements Engineering*, ser. RE '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 84–91. [Online]. Available: http://dl.acm.org/citation.cfm?id=647646.731259

[25] E. Martins, C. M. Toyota, and R. L. Yanagawa, "Constructing self-testable software components," in *Dependable Systems and Networks, 2001. DSN 2001. International Conference on*, July 2001, pp. 151–160.

[26] A. Orso, M. Harrold, and D. S. Rosenblum, "Component metadata for software engineering tasks," in *Revised Papers from the Second International Workshop on Engineering Distributed Objects*, ser. EDO '00. London, UK, UK: Springer-Verlag, 2001, pp. 129–144. [Online]. Available: http://dl.acm.org/citation.cfm?id=647869.737134

[27] J. A.. Stafford and A. l. Wolf, "Annotating components to support component-based static analyses of software systems," University of Colorado at Boulder, Tech. Rep., 1999, cU-CS 896-99.

[28] P. Pernich, A. Buccella, A. Cechich, S. Doldan, E. Morsan, M. Arias, and M. Pol'la, "Developing a

subdomain-oriented software product line," in *Proceedings of the CACIC'11: 17th Congreso Argentino en Ciencias de la Computación*, La Plata, Argentina, 2011.

[29] N. Huenchuman, A. Buccella, M. Pol'la, , A. Cechich, S. Doldan, E. Morsan, and M. Arias, "Reingeniería de una línea de productos de software: Un caso de estudio en el subdominio de ecología marina," in *Proceedings of the CACIC'13: XIX Congreso Argentino de Ciencias de la Computación*, Mar del Plata, Argentina, 2013.

[30] B. Y.. Alkazemi, "Towards a software product line framework to support cross-domain component composition," in *12th International Conference on Computational Science and Its Applications (ICCSA)*, 2012, pp. 130–133.

Detección de Agrupamientos en Glosarios del Universo de Discurso

Marcela Ridao
INTIA
Universidad Nacional del Centro
de la Provincia de Buenos Aires
mridao@exa.unicen.edu.ar

Jorge H. Doorn
Escuela de Informática
Universidad Nacional del Oeste
Departamento de Ingeniería
Universidad Nacional de La Matanza
jdoorn@exa.unicen.edu.ar

Abstract

Los modelos construidos a lo largo del proceso de desarrollo de software en general, y en las actividades de la Ingeniería de Requisitos en particular, son creados con propósitos y estructuras bien definidas. Estas estructuras han sido concebidas para maximizar la expresividad del modelo en relación con su propósito. Pese a esto y muy posiblemente debido a esto, puede ocurrir que en los mismos exista información no perceptible durante su uso rutinario. Desde un punto de vista epistemológico, esta es una situación muy similar a la que tiene lugar en la minería de datos. En el caso de los modelos de IR basados en lenguaje natural, se ha observado que una segunda lectura de algunos de ellos permite la adquisición de al menos parte de esa información oculta. En este artículo, se propone una estrategia para visualizar la existencia de agrupamientos de términos, en el Léxico Extendido del Lenguaje, los que efectivamente se corresponden con núcleos semánticos del Universo del Discurso[1] que está siendo estudiado. Esta estrategia se basa en la construcción automática de grafos utilizando los vínculos hipertextuales empotrados en el modelo.

1. Introducción

Últimamente han ido adquiriendo importancia las disciplinas dedicadas al estudio de fenómenos donde el aspecto dominante es la complejidad estructural y no la complejidad esencial de los elementos involucrados en la estructura [4] [9]. Existen numerosos ejemplos, en diversas disciplinas, donde la detección de agrupamientos representa una contribución significativa a la mejor comprensión del fenómeno que está siendo estudiado [29] [31] [32]. En particular, el método de priorización Pirogov, propuesto por Duan [10] analiza documentos del proceso de requisitos mediante la detección de agrupamientos. En dicho método, los requisitos se ubican en clusters mediante algoritmos automáticos iterativos, se priorizan los clusters negociando entre los involucrados, obteniendo así un ranking de requisitos de acuerdo al cluster al que pertenece cada uno.

En este tipo de problemas se destacan algunos tales como las redes organizacionales o redes sociales organizacionales, las redes de referencias bibliográficas o redes de grupos de interés entre muchas otras.

La representación visual clásica de estas redes se realiza mediante grafos. Pero ocurre que apenas superado un número moderado de nodos, estos grafos resultan inapropiados para distinguir los aspectos relevantes de la estructura.

Algunos de los modelos de la Ingeniería de Requisitos pueden ser estudiados desde el punto de vista estructural. En particular, uno de los más promisorios es el Léxico Extendido del Lenguaje (LEL) [19]. Este modelo registra el vocabulario del Universo del Discurso, mediante la descripción de los términos utilizados por el cliente-usuario, postergando la comprensión del problema. (Ver Sección 2).

Si se observa un LEL bajo la óptica estructural se puede construir un grafo donde los símbolos sean los nodos y las menciones a otros símbolos sean arcos dirigidos. Desde este punto de vista, el LEL puede visualizarse como una suerte de red lingüistica con una estructura claramente compleja. Es así que, además de la información explícita almacenada en cada nodo, existe una información implícita empotrada en la estructura de las relaciones entre los nodos. Epistemológicamente este enfoque es muy similar al utilizado en minería de datos [3], en el sentido que se hace visible información oculta

[1] UdeD: todo el contexto en el cual el software se desarrolla e incluye todas las fuentes de información y todas las personas, relacionadas con el software: usuarios, ingenieros de software, expertos del dominio, etc. Es la realidad acotada por el conjunto de objetivos establecidos por quienes demandan una solución de software [18].

mediante el uso de una técnica notoriamente diferente a la utilizada rutinariamente.

Construir y analizar el grafo de los símbolos del LEL se constituye así en una suerte de minería de información y de conocimiento, ya que se obtiene información sintáctica acerca de la estructura del LEL y al mismo tiempo se adquiere parte del conocimiento subyacente bajo esa estructura.

El conocimiento más elemental deducible de la estructura del LEL está relacionado con la existencia o no de agrupamientos de símbolos no siempre visibles en su navegación interactiva.

Estos agrupamientos o sub-agrupamientos, cuando existen, permiten la visualización de componentes de posibles taxonomías del proceso del negocio.

Intuitivamente, se puede suponer que si el Universo de Discurso contiene distintas áreas de interés, fragmentos de la organización o subprocesos diferenciados, entonces se debería esperar cierto grado de acoplamiento mayor entre términos que describen sujetos, objetos, verbos o estados de un fragmento determinado que entre los términos correspondientes a áreas diferentes.

La propuesta de este trabajo consiste en analizar el grafo construido a partir del LEL, con el fin de detectar los agrupamientos mencionados.

2. LEL: Léxico Extendido del Lenguaje

La construcción de un vocabulario que capture la jerga usada por los expertos del dominio ha sido propuesta por distintos autores [2] [19]. De hecho, varias experiencias han mostrado que un glosario del vocabulario de los clientes-usuarios es, en si mismo, una fuente de información para elicitar información del Universo de Discurso (UdeD) [5] [28] [22] [25] [24].

En este trabajo, se analizará un modelo de glosario en particular: el Léxico Extendido del Lenguaje (LEL).

El LEL es una representación de los símbolos del lenguaje del dominio del problema, que intenta capturar el vocabulario de una aplicación. Su objetivo principal es que el ingeniero de requisitos conozca el lenguaje que habla el usuario, entendiendo los términos que usa, sin preocuparse por entender el problema [20] [21].

El LEL involucra la denotación y la connotación de cada símbolo descubierto como una palabra o frase relevante al dominio de la aplicación. El propósito de la construcción del léxico no sólo es habilitar una buena comunicación y acuerdo entre los clientes/usuarios y el equipo de ingeniería sino también facilitar la construcción de escenarios y ayudar a su descripción, facilitando la validación.

Este léxico se construye utilizando lenguaje natural y está compuesto, en primera instancia, por símbolos que pueden ser objetos activos o Sujetos (realizan acciones), objetos pasivos u Objetos (las acciones se realizan sobre ellos), Verbos (acciones del sistema) y Estados significativos del sistema [18].

Cada símbolo en el léxico tiene uno o más nombres o frases que lo identifican y dos tipos de descripciones, la noción y el impacto. La noción describe la denotación de la palabra o frase. Indica quién, cuándo ocurre, qué procesos involucra, qué significado tiene el símbolo, etc. El impacto describe la connotación del símbolo, es decir, su repercusión en el sistema. Esta descripción puede contener acciones que se ejecutan sobre otros objetos o que se aplican al que se está describiendo, situaciones derivadas de la que se está definiendo, etc. Cada entrada puede contener una o más nociones y uno o más impactos.

En la descripción de los símbolos deben cumplirse simultáneamente dos reglas básicas [19]:

- ***Principio de circularidad***: en la descripción de la noción o impacto de los símbolos se debe maximizar el uso de otros símbolos del léxico. De esta manera, el conjunto de símbolos determina una red, que permite representar al LEL mediante un hipertexto que puede ser navegado para conocer todo el vocabulario del problema.
- ***Principio del vocabulario mínimo:*** se debe minimizar el uso de símbolos externos al lenguaje de la aplicación. De este modo, se acota el lenguaje al menor conjunto de símbolos posible. Si se utilizan símbolos externos, éstos deben pertenecer al vocabulario básico del leguaje natural que se está utilizando.

LEL: Representación de los símbolos en el lenguaje del dominio de la aplicación
Sintaxis: $\{Símbolo\}_1^N$

Símbolo: Entrada del léxico que tiene un significado especial en el dominio de la aplicación
Sintaxis: $\{Nombre\}_1^N$ + $\{Noción\}_1^N$ + $\{Impacto\}_1^N$

Nombre: Identificación del símbolo. Más de uno representa sinónimos
Sintaxis: Palabra | Frase

Noción: Denotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo
Sintaxis: Sentencia

Impacto: Connotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando un vocabulario mínimo
Sintaxis: Sentencia

Figura 1. Modelo del Léxico Extendido del Lenguaje.
Sentencia está compuesta por Símbolos y No-Símbolos pertenecientes al vocabulario mínimo, **+** significa composición, | significa or, y $\{x\}_M^N$ significa de M a N ocurrencias de x.

En la figura 1 se presenta el modelo utilizado para representar los símbolos:

3. Trazado de Grafos: Métodos dirigidos por fuerzas

La Teoría de Grafos tiene diversidad de aplicaciones. La representación mediante nodos y conexiones es usada para representar redes físicas como circuitos eléctricos, carreteras, moléculas orgánicas, etc. También se utilizan los grafos en la representación de interacciones menos tangibles como puede ocurrir en ecosistemas, relaciones sociológicas, bases de datos, o en el flujo de control de un programa computacional [14] [15] [6].

El Trazado de Grafos, como una rama de la Teoría de Grafos, aplica topología y geometría para derivar representaciones de grafos en dos dimensiones. El trazado de un grafo es básicamente una representación gráfica del grafo en el plano, usualmente destinada a una visualización conveniente de ciertas propiedades del grafo en cuestión o del o los objetos modelados por el grafo [8].

Un grafo G = (N,A) es un conjunto N de nodos y un conjunto A de arcos, en el cual un arco une un par de nodos. Normalmente, los grafos se dibujan con los nodos como puntos en un plano y sus arcos como líneas o segmentos curvos conectando esos puntos. Existen diferentes estilos de representación, adecuados a diferentes tipos de grafos o diferentes propósitos de representación.

En el trazado de grafos existen criterios estéticos, mayormente aceptados [12]:

- Distribuir los nodos uniformemente en el marco de trabajo.
- Minimizar los cruces de arcos.
- Hacer que las longitudes de los arcos sean uniformes.
- Reflejar simetría.
- Adecuarse al marco de trabajo.

La mayoría de estos criterios, sin embargo, atentan contra el objetivo de visualización de grafos que se desarrollará en esta propuesta, donde el énfasis se pone en la estructura de la red, y no en aspectos estéticos.

La generación automática del trazado de grafos tiene importantes aplicaciones en muchas áreas de las ciencias de la computación, tales como compiladores, bases de datos, ingeniería de software, VLSI, diseño de redes, interfaces gráficas, etc. La aplicación en otras áreas incluye análisis gráfico de datos (por ejemplo en todos los campos de la ingeniería, biología, o ciencias sociales) y la visualización de información en general (por ejemplo diagramas de flujo, mapas esquemáticos o toda clase de diagramas) [17].

La amplia variedad de familias de grafos ha hecho que los algoritmos de trazado desarrollados varíen según el tipo de grafos que permiten visualizar. Existen algoritmos específicos para dibujar árboles, para grafos dirigidos acíclicos, para grafos planares, etc. Entre esta gran variedad, existen algoritmos para trazado de grafos generales. Y entre ellos, se destaca una familia de métodos conocidos como “dirigidos por fuerzas”. Estos métodos son muy usados hoy en día para dibujar grafos, porque dan buenos resultados, son sencillos de implementar, y son muy flexibles, por lo que pueden ser fácilmente adaptados a aplicaciones concretas con requerimientos de visualización específicos [1] [30].

Los métodos de trazado de grafos dirigidos por fuerzas son una familia de algoritmos que usan analogías físicas para dibujar el grafo. Tienen como denominador común las siguientes características:

- Modelan al grafo como un sistema físico.
- El trazado del grafo es obtenido buscando el equilibrio del sistema físico.

Una tercera característica que se puede agregar, si bien no forma parte de la definición estándar, pero que se verifica en la mayoría de los casos, es que se aplican a grafos generales, es decir, los algoritmos no se basan en ninguna propiedad estructural del grafo (por ej., su planaridad), sino que se aplican a cualquier tipo de grafo [1].

En general, este tipo de algoritmos tiene dos componentes principales:

- Un modelo físico del grafo, que representa a los elementos del grafo (nodos y arcos) junto a los criterios estéticos que se desean obtener del dibujo.
- Un algoritmo para encontrar un equilibrio del sistema físico, que se corresponderá, en principio, con un trazado del grafo estéticamente agradable.

Los modelos físicos más comunes son los que consisten de un sistema de fuerzas (donde generalmente se definen fuerzas que actúan entre los vértices del grafo), en cuyo caso el objetivo del algoritmo es encontrar un equilibrio para este sistema de fuerzas, es decir, una posición para cada vértice, de manera que el total de la fuerza ejercida en cada vértice sea cero.

Entre los primeros autores aplicando analogías con sistemas físicos para el trazado de grafos, se destaca el “Spring Embedder” propuesto por Eades [11], que se basa en reemplazar los nodos por anillos de acero y cada arco con un resorte para formar un sistema físico, como se ve en la figura 2.

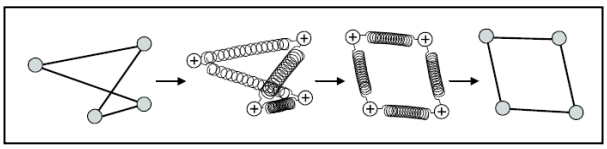


Figura 2. Spring Embedder [1]

Los nodos son ubicados en alguna disposición inicial, y se dejan actuar las fuerzas de los resortes hasta lograr un estado de energía mínima. La implementación de Eades, sin embargo, no siguió al pie de la letra la ley de Hooke, sino que incorporó al cálculo de las fuerzas resultantes, fuerzas repulsivas calculadas entre los nodos no conectados.

Otra forma de definir el modelo del grafo, en lugar de usando un sistema de fuerzas, es a través de un sistema de energía. En este caso, el propósito del algoritmo es encontrar una disposición para los vértices del grafo tal que la energía total del sistema sea mínima. Diferentes autores han optado por esta variante para el trazado de grafos, entre ellos Kamada y Kawai [16] y Davidson y Harel [7].

Fruchterman y Reingold [12] proponen un método derivado, principalmente, del Spring Embedder de Eades, basado en los siguiente principios:

- Los nodos conectados por un arco deberían ser dibujados cerca.
- Los nodos no deberían ser dibujados demasiado cerca uno de otro.

Cuán cerca se deberían ubicar los nodos, depende de cuántos haya y cuánto sea el espacio disponible.

Las guías para este algoritmo son obtenidas de la física de partículas:

> *A una distancia de cerca de 1 fm (femtómetro), la fuerza nuclear es atractiva y equivalente a cerca de 10 veces la fuerza eléctrica entre 2 protones. La fuerza decrece rápidamente a medida que crece la distancia, haciéndose completamente insignificante cuando se llega a cerca de 15 veces esta separación. Cuando dos núcleos están a una distancia de entre 0 y 4 fm uno de otro, las fuerzas nucleares se convierten en repulsivas.*

Si los nodos se comportan como partículas atómicas o cuerpos celestes, ejerciendo fuerzas atractivas y repulsivas sobre los demás, las fuerzas inducen movimiento. El algoritmo de Fruchterman y Reingold se basa en simulaciones moleculares o planetarias. Sin embargo, no se propone una simulación exactamente fiel a la realidad. Del mismo modo que en el algoritmo de Eades, sólo los nodos que son vecinos se atraen entre sí, mientras todos los vértices se repelen unos a otros. Esto es consistente con la asimetría propuesta por los dos principios antes enunciados.

4. Fuerzas dirigidas en la visualización del LEL

Con el fin de detectar agrupamientos de símbolos, se aplicó una modificación del algoritmo propuesto por Fruchterman y Reingold [12] en la visualización de los grafos correspondientes a los Léxicos de diferentes casos de estudio. Para ello, cada símbolo del LEL fue representado mediante un nodo, y las menciones a otros símbolos incluidas en su definición, se representaron como arcos dirigidos a los nodos respectivos.

En la figura 3, se presenta un pseudocódigo del algoritmo propuesto en [12].

```
area := W * L; { W and L are the width and
 length of the frame }
G := (V, E); { the vertices are assigned
 random initial positions }
k := √area/|V|;
function fa(z) := begin return z²/k end ;
function fr(z) := begin return k²/z end ;
for i := 1 to iterations do begin
   { calculate repulsive forces }
   for v in V do begin
      { each vertex has two vectors: .pos and
       .disp }
      v.disp := 0;
      for u in V d o
         if (u ≠ v) then begin
         { Δis short hand for the difference
          }
         { vector between the positions of
          the two vertices }
         Δ := v.pos - u.pos;
         v.disp := v.disp + (Δ/|Δ|) * fr(|Δ|)
      end
   end
   { calculate attractive forces }
   for e in E do begin
      { each edge is an ordered pair of
       vertices .v and .u }
      Δ := e.v.pos – e.u.pos
      e.v.disp := e.v.disp – (Δ/|Δ|) *
       fa(|Δ|);
      e.u.disp := e.u.disp + (Δ/|Δ|) *
       fa(|Δ|)
   end
   { prevent from being displaced outside
    frame }
   for v in V do begin
      v.pos := v.pos + ( v. disp/ |v.disp|) *
       v.disp;
      v.pos.x := min(W/2, max(-W/2, v.pos.x));
      v.pos.y := min(L/2, max(–L/2, v.pos.y))
   end
end
```

Figura 3. Algoritmo de visualización de grafos dirigido por fuerzas [12]

Para la aplicación del algoritmo, los nodos son ubicados al azar en el marco de trabajo, y posteriormente se va modificando su ubicación en forma iterativa. Cada iteración tiene tres pasos:

- Calcular el efecto de las fuerzas atractivas sobre cada nodo.

- Calcular el efecto de las fuerzas repulsivas.
- Limitar el desplazamiento total.

4.1. Las fuerzas

fa y fr son las fuerzas de atracción y de repulsión, respectivamente. Como puede observarse en la figura 3, las fuerzas propuestas en [12], son:

$$fa(d) = \frac{d^2}{k} \qquad k = C * \sqrt{\frac{Area}{NúmeroNodo \quad s}}$$

$$fr(d) = \frac{k^2}{d}$$

Donde d es la distancia entre los vértices y k el radio vacío alrededor de un nodo.

La aplicación de este sistema de fuerzas permitió detectar agrupamientos de símbolos para varios de los casos analizados. Sin embargo, se experimentó con fuerzas diferentes, con el fin de comprobar si se podía mejorar la visualización de los mismos.

En particular, se trabajó con el par de fuerzas propuesto por Eades [11]:

$$fa(d) = c1 * \log\left(\frac{d}{c2}\right) \qquad (1)$$

$$fr(d) = c3 * \sqrt{d} \qquad (2)$$

Las constantes c1, c2 y c3 se ajustaron experimentalmente, de acuerdo a sus efectos sobre la visualización de los agrupamientos.

Con este sistema de fuerzas, los agrupamientos que se habían detectado con las fuerzas originales se visualizaron de una forma mucho más clara. Por ello, el algoritmo fue modificado considerando (1) y (2) como fuerzas de atracción y repulsión respectivamente [26].

4.2. El marco de trabajo

El grafo debe ser confinado al marco especificado por el usuario. El algoritmo propuesto por [12] considera la ubicación de nodos ficticios en el perímetro del marco de trabajo, que ejercen fuerzas repulsivas sobre los nodos del grafo, pero ellos mismos permanecen fijos. De este modo, el marco se modela como cuatro paredes que contienen al grafo dentro de ellas.

Sin embargo, las experiencias realizadas permitieron determinar que no era necesaria la ubicación de nodos en las paredes del marco de trabajo, ya que, con la elección adecuada de las constantes que intervienen en el cálculo de las fuerzas, los nodos se mantienen dentro del marco definido.

5. Visualización de los Léxicos de diferentes casos de estudio

Con el fin de verificar si la estrategia propuesta permite detectar agrupamientos de símbolos, se la aplicó a diferentes casos de estudio cuyos Léxicos habían sido verificados y validados previamente.

5.1. Caso 1: Sin clusters conocidos a priori

Se aplicó la estrategia a un caso de estudio para el cual se prefirió no analizar en forma semántica la presencia o ausencia de agrupamientos de símbolos. Los casos analizados fueron:

- Sistema de Planes de Ahorro Previo para la Adquisición de Vehículos 0Km [27]

En la figura 4, se presentan los resultados obtenidos luego de aplicar la estrategia al grafo correspondiente al LEL de este caso de estudio. Cabe destacar que, tanto en ésta como en las siguientes figuras, no se muestran los arcos correspondientes a los vínculos entre los símbolos, para permitir una mejor visualización de la distribución de los nodos.

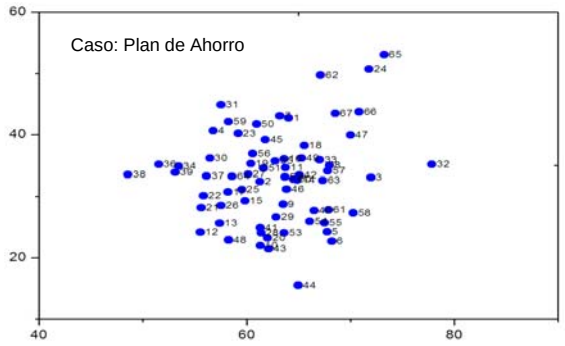


Figura 4. Distribución de nodos para el caso 1.

En el grafo presentado en la figura no se observan agrupamientos claramente diferenciados. El análisis semántico del Léxico correspondiente al caso de estudio, confirma la ausencia de agrupamientos destacados, validando el resultado arrojado por la estrategia.

Podría considerarse que los símbolos constituyen un solo cluster, del cual deberían descartarse aquellos que claramente se alejan del agrupamiento central. En este caso, los símbolos 24, 62, 65, 32 y 44 parecen mantener escasa relación con el resto.

5.2. Caso 2: con 2 clusters conocidos

A continuación, se aplicó el algoritmo a un caso de estudio para el cual se conocía de antemano la presencia de clusters.

El caso corresponde al LEL del proceso de Construcción de LEL y Escenarios [13]. Para este caso de estudio, existen al menos dos agrupamientos: uno constituido por los símbolos correspondientes a la

Construcción del LEL, y otro constituido por los símbolos correspondientes a la construcción de Escenarios.

En la figura 5 se presenta la distribución de los nodos obtenida con el algoritmo.

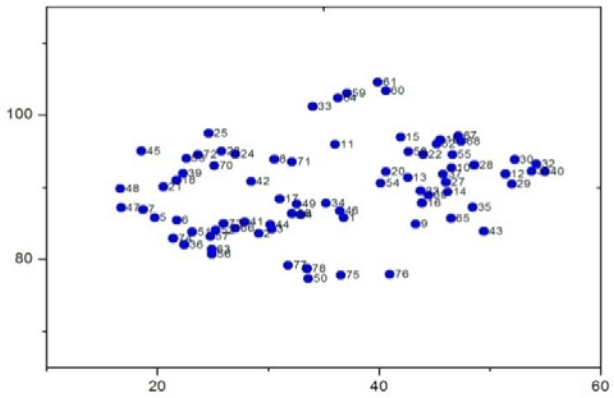


Figura 5. Distribución de nodos para el caso 2.

En la figura se observa que los símbolos correspondientes a la construcción del LEL tienden a agruparse hacia un lado, mientras los correspondientes a la construcción de escenarios, lo hacen al otro. En la zona central, se ubican aquellos símbolos relacionados tanto con el proceso de construcción del LEL como con el de Escenarios. Por ejemplo, el nodo 11 corresponde al símbolo Componente dudas, considerado en la construcción de ambos modelos. En la misma zona, se observan otros símbolos que están asociados en igual medida con el modelo del LEL y el modelo de los Escenarios, pero además se agrupan entre sí. Por ejemplo, los símbolos 50, 75, 76, 77 y 78 están relacionados con la Vista de Hipertexto en los modelos de LEL y Escenarios.

5.3. Caso 3: con 2 clusters conocidos

El último caso de estudio al que se aplicó el algoritmo describe la relación entre productores de papa y una empresa acopiadora. Un análisis semántico preliminar de este caso de estudio, determinó la existencia de dos agrupamientos de símbolos; un grupo correspondientes a Producción y Entrega de Papas y otro a Canje de Semillas.

Sin embargo, en la visualización del grafo luego de la aplicación de la estrategia, se observa claramente la presencia de tres agrupamientos, como puede verse en la figura 6.

Con el fin de comprobar los límites de los clusters visualizados, se aplicó el clasificador K-Means [23] a los datos obtenidos, indicando la presencia de 3 grupos. Los agrupamientos detectados por el clasificador coinciden con los que se pueden observar a simple vista. En la figura 6 se ha rodeado con una elipse cada uno de los clusters identificados, y en la tabla 1, se presentan algunos de los símbolos pertenecientes a cada uno.

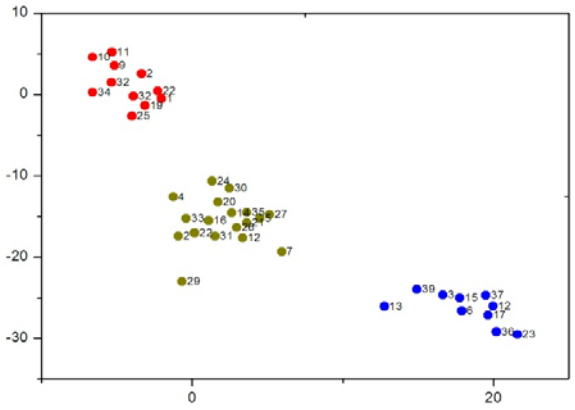


Figura 6. Distribución de nodos para el caso 3.

Tabla 1. Fragmento de la lista de símbolos correspondientes a cada uno de los clusters.

Producción y Entrega	Canje de Semilla	Calidad de la Papa
2 Bonificación	**3** Cancelar la operación de canje	**1** Agrietadura
4 Causa justificada	**6** Contrato de canje de semilla	**8** Corazón hueco
5 Contrato de adquisición y producción	**12** División de Semillas MC S.A.	**9** Defecto de calidad
14 El Productor	**13** El Comprador	**10** Defecto externo
16 Recibir papa	**15** El Vendedor	**11** Defecto interno
18 Establecer Programa de Entregas	**17** Entregar semillas	**19** Grado MC Consumo
20 Inspeccionar papa	**23** Orden de carga	**22** Malformación
21 MC S.A.	**26** Papa Grado MC Consumo	**25** Papa chica
24 Pagar por la papa	**37** Semilla de Papa	**38** Tolerancias establecidas
…	…	…

Revisando estos resultados se pudo confirmar la presencia de estos tres agrupamientos. En particular, el tercer grupo, correspondiente a Calidad de la Papa, no había sido detectado en la lectura rutinaria, lo que confirma la capacidad de revelar información oculta por parte del método. Los símbolos incluidos en el agrupamiento calidad de la papa, se habían considerado parte del cluster correspondiente a Producción y Entrega.

Como se indicó en la sección 4, la ubicación inicial de los nodos del grafo se hace al azar. Para todos los casos estudiados, se efectuaron múltiples pruebas, con diferentes configuraciones iniciales, produciéndose los mismos agrupamientos, o falta de ellos que se puede observar en las figuras 4, 5 y 6.

Se estudió también la posibilidad de ponderar los arcos entre símbolos según el número de referencias existentes

entre ellos, pero no se observaron efectos significativos en los agrupamientos detectados.

5.4. Determinación del número de clusters

Si se observa la forma geométrica de los grafos resultantes del análisis de los tres casos, se observan algunas características que pueden ser consideradas para determinar el número de clusters detectados.

Para los casos 1 y 2, se aprecia cierta simetría tanto en el eje X como en el eje Y, mientras que, para el caso 3, esta simetría no está presente.

Se calcularon los coeficientes de asimetría en X y en Y para cada caso, obteniendo los siguientes resultados:

Caso 1:

$AS_X = 0.02 \quad AS_Y = 0.14$

Sin embargo, si se descartan los nodos que se alejan del agrupamiento principal, lo que hace suponer que no guardan una estrecha relación con el resto, los valores calculados son:

$AS_X = -0.09 \quad AS_Y = 0.04$

Caso 2:

$AS_X = 0.05 \quad AS_Y = 0.14$

Caso 3:

$AS_X = 0.79 \quad AS_Y = -0.48$

Se puede inferir que, cuando los coeficientes de asimetría se alejan de 0, es posible que se esté en presencia de un LEL con un número de agrupamientos mayor que dos. No se han definido los límites para los cuales es razonable ignorar la asimetría o considerar que la misma indica la presencia de tres o más clusters.

Por otra parte, en el caso 2, donde las clases que se destacan son 2, se observa una forma elipsoidal en la distribución de los nodos, donde los símbolos correspondientes a cada cluster se agrupan alrededor de cada uno de los focos de la elipse. El cociente entre la variancia sobre el eje X y la variancia sobre el eje Y es:

$$CV = \frac{\sigma_X^2}{\sigma_Y^2} = 3{,}18 \quad (1)$$

Para los casos 1 y 3, en cambio, no es posible asociar las formas resultantes con una elipse, como en el caso de LEL y Escenarios.

Entonces, para aquellos casos donde se observa cierta simetría en los ejes X e Y (coeficientes de asimetría cercanos a 0), y un cociente de varianzas mayor que 1, se puede inferir la presencia de dos clusters principales. No se ha estudiado aún cuál es el límite a partir del cual, el valor de este cociente puede asegurar la presencia de dos agrupamientos.

Teniendo en cuenta los resultados analizados, se proponen, como un primer enfoque para determinar el número de clusters en un LEL, las siguientes reglas:

- Si $AS_X \cong 0$ y $AS_Y \cong 0$ sin considerar el valor de CV
  se puede inferir que todos los símbolos constituyen un único cluster.
- Si $AS_X \cong 0$ y $AS_Y \cong 0$ y $CV > 1$
  se puede inferir la presencia de 2 clusters.
- Si $AS_X \neq 0$ y $AS_Y \neq 0$, sin importar el valor de *CV*
  se puede inferir la presencia de tres o más clusters.

Como se dijo anteriormente, queda pendiente el análisis de más casos de estudio para definir los límites inferior y superior para CV y AS que permitan determinar más precisamente la presencia de un número determinado de clusters.

6. Conclusiones

Se ha propuesto una estrategia que permite detectar agrupamientos de símbolos en un modelo del proceso de Requisitos: el Léxico Extendido del Lenguaje.

El análisis de los resultados obtenidos, permite concluir que es posible efectuar una segunda lectura de algunos de los documentos de Ingeniería de Requisitos lo que brinda nueva información acerca del UdeD. Se comprobó que todos los clusters encontrados en los casos de estudio se corresponden con agrupamientos conceptuales observables en el mundo real.

El uso de grafos y el análisis de su estructura son herramientas apropiadas para el estudio de la información implícita empotrada en dichos documentos, en particular en el Léxico Extendido del Lenguaje (LEL).

Es posible determinar métricas de buena calidad que actúen como indicadores del número de clusters existentes.

Los resultados obtenidos se acoplan en forma muy natural con estrategias genéricas de clusterización como K-Means. El conocimiento previo del número de clusters es imprescindible para el uso de K-Means y en el caso de esta estrategia no es necesario.

En próximos trabajos se ampliará el número de dimensiones de la caja contenedora del grafo, permitiendo construir grafos en espacios de tres o más dimensiones. Se presume que esta forma de trabajar permitirá extraer aun más información del LEL y eventualmente poder encarar casos más complejos. Claramente esta ampliación de la cantidad de dimensiones obligará a utilizar técnicas de visualización más elaboradas.

Se ha considerado la posibilidad de ponderar los arcos entre símbolos según el número de referencias existentes entre ellos, pero las experiencias realizadas hasta el momento no arrojaron diferencias significativas en los agrupamientos detectados. Se proyecta estudiar diferentes

alternativas de ponderación, que pudieran incidir en los resultados.

Con el fin de comenzar a analizar las posibles aplicaciones de la información recuperada mediante el análisis estructural de los documentos en etapas posteriores de la Ingeniería de Requisitos, se planea comparar la técnica presentada en este artículo con otras técnicas de clusterización, como por ejemplo la propuesta por Duan [10].

7. Referencias

[1] Aiello, A. abd Silveira, A., *Trazado de grafos mediante métodos dirigidos por fuerzas: revisión del estado del arte*, Departamento de Computación. Facultad de Ciencias Exactas y Naturales. UBA, Tesis de Licenciatura en Ciencias de la Computación, 2004.

[2] Arango, G., Schafer, W. and Prieto, R., *Domain Analysis Methods – Software Reusability*, Ellis Horwood Ltd., 1993.

[3] Artz, J.M., "Data Driven vs. Metric Driven Warehouse Design" in: *Encyclopedia of Data Warehousing and Mining*, ISBN 1-59140-557-2, IGI Global, 2009, pp 382-387.

[4] Barabasi, A., *Linked, The New Science of Network*, Perseus publishing, 2002.

[5] Ben Achour, C., Rolland, C., Maiden, N. and Souveyet, C., "Guiding Use Case Authoring: Results of an Empirical Study", in: *Proceedings International Symposium on Requirements Engineering*, Limerick-Ireland, June 1999, pp. 36-43.

[6] Brandes, U., Kenis, P. and Wagner, D., "Communicating centrality in policy network drawing". *IEEE Transactions on visualization and computer graphics*, 9, 2, 2003, pp. 241-253.

[7] Davidson, R. and Harel, D., "Drawing graphs nicely using simulated annealing", *ACM Transactions on Graphics*, 15, 4, 1996, pp. 301-331.

[8] Di Battista, G. , Eades, P. , Tamassia, R. nad Tollis, I., *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.

[9] Dorogovtsev, S. and Mendes, J., *Evolution of networks: From biological nets to the Internet and WWW*, Oxford University Press, 2003.

[10] Duan, C., Laurent, P., Cleland-Huang, J., and Kwiatkowski, C., "Towards automated requirements prioritization and triage", *Requirements Engineering Journal*, 14, 2, 2009, pp.73-89.

[11] Eades, P., "A heuristic for graph drawing", in: *Proceedings Congressus Numerantium*, 42, 1984, pp. 149–160

[12] Fruchterman, T. and Reingold, E., "Graph Drawing by Force-directed Placement", *Software-Practice and Experience*, 21, 11, 1991, pp. 1129-1164.

[13] Garcia, O. and Gentile, C., *Diseño de una herramienta para contrucción de LEL y Escenarios*, Universidad Nacional del Centro de la Pcia. de Bs. As., Tesis de grado, 1999.

[14] Gross, J. and Yellen, J., Editors. *Handbook of Graph Theory*, CRC Press, 2003.

[15] Gross, J. and Yellen, J., Editors. *Graph Theory and Its Applications, Second Edition (Discrete Mathematics and Its Applications)*, Chapman & Hall/CRC, 2006.

[16] Kamada, T. and Kawai, S., "An algorithm for drawing general undirected graphs", *Information Processing Letters* 31, 1989, pp. 7-15.

[17] Kaufmann, M. and Wagner, D., (eds.) *Drawing graphs: methods and models, LNCS*, vol 2025, Springer-Verlag, 2001.

[18] Leite, J., Doorn, J., Kaplan, K., Hadad, G. and Ridao, M., "Defining System Context Using Scenarios", in: Leite, J., Doorn, J (eds.) *Perspectives on Software Requirements*, Kluwer Academic Press, 2004, pp. 169-199.

[19] Leite, J. and Franco, A.,"O Uso de Hipertexto na Elicitaçao de Linguagens da Aplicaçao", in: *Proceedings IV Simpósio Brasilero de Engenharia de Software, SBC*, Brazil, 1990, pp. 134-149.

[20] Leite, J. and Franco, A., "A Strategy for Conceptual Model Acquisition", in: *Proceedings IEEE International Symposium on RE*, 1993, pp. 243-246.

[21] Leite, J., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G. and Oliveros, A., "Enhancing a Requirements Baseline with Scenarios", *Requirements Engineeering Journal*, 2, 4, 1997, pp. 184-198.

[22] Oberg, R., Probasco, L. and Ericsson, M., *Applying Requirements Management with Use Cases*, Rational Software Corporation, 1998.

[23] Peña, J., Lozano, J. and Larrañaga P., "An empirical comparison of four initialization methods for the K-Means algorithm", *Pattern Recognition Letters*, 20, 1999, pp. 1027-1040.

[24] Prakash, S., Aurum, A. and Kox, K., "Requirements Engineering Practice in Pharmaceutical and Healthcare Manufacturing", in: *Proceedings 11th Asia-Pacific S.E. Conference*, 2004, pp. 402-409.

[25] Regnell, B., *Requirements Engineering with Use Cases – a Basis for Software Development*, Department of Communication Systems, Lund University, Doctoral Thesis, 1999.

[26] Ridao, M. and Doorn, J., "Semántica Oculta en Modelos de Requisitos", in: *XV Workshop de Investigadores en Ciencias de la Computación (WICC)*, Paraná, Argentina, Abril 2013, pp. 471-475.

[27] Rivero, L., Doorn, J., del Fresno, M., Mauco, V., Ridao, M. and Leonardi, C., "Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real", in: *Proceedings WER'98 - Workshop en Engenharia de Requisitos*, Maringá, Brasil, 1998, pp. 79-90.

[28] Rolland, C. and Ben Achour, C., "Guiding the construction of textual use case specifications", *Data & Knowledge Engineering*, 25, 1998, pp. 125-160.

[29] Sarmah, R., Kalita, J. and Bhattacharyya, D.K., "A pattern matching approach for clustering gene expression data", *Int. Journal Data Mining, Modelling and Management*, 3, 2, 2011, pp. 130-149.

[30] Walshaw, C., "A multilevel algorithm for force-directed graph-drawing", *Journal of Graph Algorithms and Applications*, 7, 3, 2003, pp. 253-285.

[31] Mo, Y., Cao, Z. and Wang, B., "Occurrence-Based Fingerprint Clustering for Fast Pattern-Matching Location Determination", *IEEE Communications Letters*, 16, 12, 2012, pp. 2012 – 2015.

[32] Zimmermann, M., Ntoutsi, I., Siddiqui, Z., Spiliopoulou, M. and Kriegel, H.P. "Discovering Global and Local Bursts in a Stream of News", in: *Proceedings 27th Annual ACM Symposium on Applied Computing. (SAC '12*), Italy, 2012, pp. 807-812.

TracEDaaS: Captura y Trazabilidad de Artefactos del Proceso de Diseño

Federico Hernandez, Luciana Roldán, Marcela Vegetti, Silvio Gonnet, Horacio Leone
INGAR(UTN-CONICET)
Facultad Regional Santa Fe, Universidad Tecnológica Nacional
Avellaneda 3657, 3000, Santa Fe, Argentina
{fhernandez, lroldan, mvegetti, sgonnet, hleone}@santafe-conicet.gov.ar

Abstract

Se presenta una herramienta, TracEDaaS, que permite la captura y la trazabilidad de la evolución de los productos del proceso de diseño de software, soportando el refinamiento y la elaboración iterativa del mismo. TracEDaaS ha sido implementada como un servicio en la nube bajo el modelo "Software as a Service" (SaaS) y permite representar junto a las versiones de los objetos del diseño, información que permite la trazabilidad de los artefactos del proceso de desarrollo, manteniendo la navegabilidad entre los distintos modelos y las diferentes versiones de los mismos. Esta información está estructurada en término de las operaciones aplicadas a los objetos de diseño. La herramienta es lo suficientemente flexible para soportar nuevos problemas de diseño en término de los conceptos particulares del dominio y las posibles operaciones que pueden ser aplicadas a las instancias de tales conceptos.

1. Introducción

Aportes recientes consideran trazabilidad como un método para seguir el proceso de desarrollo de software [14] [30]. En la última década se realizaron numerosos aportes con este objetivo, la mayoría de las cuales representan las relaciones de traza entre objetos del proceso de desarrollo [2][3][18][20][26], incluso el metamodelo de UML [21] permite representar la relación de trazabilidad como una dependencia entre elementos del modelo. Sin embargo, algunos trabajos abordan otros aspectos necesarios para poder representar y seguir la evolución de los productos del proceso de desarrollo. Alexander [1] al igual que Ballejos y Montagna [4] abordan la identificación de "stakeholders" en etapas tempranas del proceso de desarrollo. Otros autores se concentran en el modelado y captura del razonamiento [6][7][10][11][19][22]. Todos estos aspectos son relevantes para poder representar y reconstruir un proceso de desarrollo [23]. A pesar de su importancia, se han encontrado pocos trabajos que lo aborden de forma integrada, no permitiendo la representación del proceso de desarrollo junto al conocimiento empleado. Este problema ha sido reconocido por diversos autores en diferentes dominios de la ingeniería de software [15][16][17][23][27][30][8], quienes concuerdan en que tales decisiones constituyen conocimiento tácito, el cual a pesar de ser esencial para la solución alcanzada, no es explícitamente documentado, ni forma parte del artefacto resultante. Esto conduce a que en futuras revisiones o modificaciones del artefacto (sistema de información), es decir durante la evolución del mismo, sea difícil y costoso establecer correspondencias entre decisiones de diseño y las razones que las respaldaron.

Otra línea vinculada con la temática es la de administración de los productos de los procesos de diseño, entendiendo aquí como productos a los modelos, datos, diagramas, programas, etc. Existen desde hace tiempo, sistemas de administración de productos y de sus versiones [5], los cuales son ampliamente utilizados en la práctica. Esto no es sorprendente porque responden a la necesidad básica que los productos de un proceso de desarrollo deben ser almacenados y organizados [29]. Los sistemas de administración de productos, así como los sistemas de administración de configuraciones de software [28], están enfocados en productos, principalmente en código fuente y software ensamblado, y no contemplan el seguimiento del proceso de diseño [9]. El escenario descripto explica la dificultad que existe en la posterior recuperación de información sobre qué requerimientos fueron contemplados, qué restricciones fueron adoptadas/relajadas, qué decisiones de diseño fueron realizadas, qué alternativas fueron consideradas, y por qué esta fue la seleccionada, durante el proceso de

desarrollo. Una infraestructura que permita la captura y representación del seguimiento de artefactos del proceso de desarrollo es un elemento vital para brindar soporte a la persistencia de las dependencias entre los distintos artefactos y su posterior recuperación. En este sentido, Roldán y colab. [24][25] proponen un modelo conceptual para soportar los diferentes modelos que son generados cuando un producto ingenieril es diseñado. La propuesta soporta la representación explícita de los estados a través de los cuales el modelo del artefacto que está siendo diseñado evoluciona durante la actividad de diseño.

En virtud de lo antes expresado, el desafío actual consiste en la definición de una herramienta que facilite la administración de la evolución de los artefactos generados en un proceso de desarrollo de software. La herramienta propuesta deberá brindar soporte en la captura de la evolución de los distintos artefactos, considerando distintos niveles de abstracción, soportando el refinamiento y la elaboración iterativa del mismo. Es vital la captura y representación de la información que permita la trazabilidad de los artefactos del proceso de desarrollo, manteniendo la navegabilidad entre los distintos modelos y las diferentes versiones de los mismos. Esta representación debe considerar de manera integrada a la evolución de los productos, los actores que intervinieron, y el razonamiento de diseño asociado.

Por estos motivos, se trabajó en la propuesta de una herramienta, TracEDaaS, que permita la captura y la trazabilidad de la evolución de los productos del proceso de diseño. La herramienta se basa en un modelo genérico para la captura del diseño en término de las operaciones aplicadas a los objetos de diseño [12][13][24][25]. El objetivo principal de la herramienta es capturar las versiones del modelo desarrollado (estados del diseño) durante un proceso de diseño y cómo fueron obtenidos (razonamiento, decisiones, actores, etc.). Este ambiente fue diseñado con el requerimiento de brindar soporte a distintos dominios de diseño. Este enfoque permite su extensión a nuevos problemas de diseño en término de los conceptos particulares del dominio y las posibles operaciones que pueden ser aplicadas a las instancias de tales conceptos.

A continuación, en la Sección 2, se presenta el modelo conceptual de TracEDaaS. Luego, en la Sección 3, se define la arquitectura de TracEDaaS y su implementación en la nuble. Por último, en la Sección 4 se presentan las conclusiones del trabajo.

2. Modelo Conceptual de TracEDaaS

La herramienta propuesta, TracEDaaS, se sustenta en un esquema de administración de versiones que permite capturar y representar la evolución de los distintos productos del diseño en forma conjunta con las operaciones aplicadas a los mismos [24][25]. Para realizar tal captura, TracEDaaS considera el proceso de diseño como una secuencia de actividades que opera sobre los productos del proceso de diseño, denominados objetos de diseño. Los objetos de diseño representan modelos del artefacto que está siendo diseñado, requerimientos a cumplir, restricciones impuestas al modelo. Naturalmente, estos objetos evolucionan durante un proceso de diseño, dando lugar a múltiples versiones de los mismos. Estas versiones pueden ser consideradas como fotos de los objetos de diseño en un determinado instante, y un conjunto de estas versiones constituyen una versión del modelo. Una versión de modelo describe el estado del proceso de diseño en un determinado instante. En este esquema, cada versión de modelo es generada mediante la aplicación de una secuencia de operaciones en una versión de modelo predecesora. La secuencia de operaciones puede incluir la eliminación, creación, y modificación de versiones que forman la versión de modelo predecesora. En consecuencia, se definió en TracEDaaS un conjunto de operaciones básicas (agregar, eliminar, modificar) a través de las cuales se representan las operaciones más complejas (refinar, abstraer, aplicarMVC, etc.) que se ejecutan en las distintas etapas del proceso de desarrollo de los modelos. La definición de estas operaciones facilita el seguimiento de las operaciones realizadas en la generación de las distintas versiones, permitiendo construir el ciclo de vida de cada modelo en particular [12][13].

TracEDaaS puede ser adaptado a nuevos problemas de diseño. Esta adaptación se realiza en término de los conceptos del método de diseño y del lenguaje de modelado empleado. Además, es posible definir las operaciones que pueden ser aplicadas en un proceso de diseño.

En la Fig. 1 se presenta una vista global del ambiente TracEDaaS. Los dos componentes principales de TracEDaaS son el editor de dominios y el administrador de versiones. Mediante el empleo del Editor de Dominios, un experto del Dominio puede definir uno o más dominios de diseños. Esta definición del dominio se realiza en término de los conceptos o tipos de objetos de diseño y las operaciones que son aplicables a los mismos.

A partir de la definición de un dominio, empleando el Administrador de Versiones, los diseñadores pueden elegir un dominio existente y ejecutar/capturar las operaciones definidas junto a las versiones de los objetos de diseño generadas.

El esquema de representación de versiones divide la representación de los objetos de diseño en dos niveles: i) nivel repositorio (Repositorio en Artefactos de Fig. 1), donde se representa cada objeto de diseño por medio de un *objeto versionable*, y se mantienen las asociaciones con otros *objetos versionables* (que representan a otros objetos de diseño); ii) nivel versiones (Versiones en Artefactos de Fig. 1), donde se representan los distintos

estados alcanzados por cada objeto de diseño durante un proyecto. Los objetos que pertenecen a este nivel se denominan *versión de objeto*.

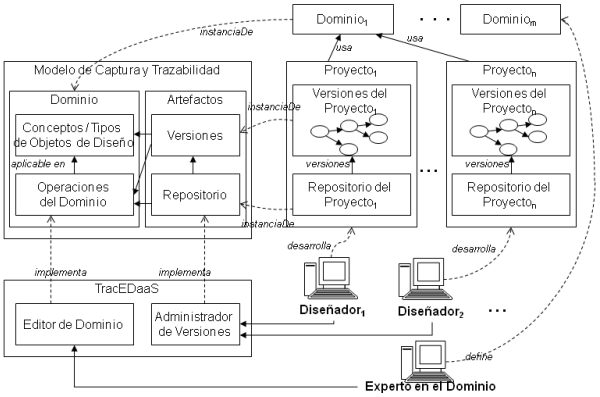


Figura 1. Vista Global de TracEDaaS.

El enfoque adoptado para la administración de las versiones de los productos del diseño considera que cada *versión de modelo* (m) surge a partir de una *versión de modelo predecesora* (m_p), sobre la cual se ha aplicado una secuencia de operaciones (ϕ) que, en general, implica la creación, eliminación, o modificación de algunos de los elementos del modelo. De esta manera, el esquema de representación de versiones sigue una estructura de árbol donde cada versión de modelo constituye un nodo, siendo la raíz la *versión de modelo inicial* (m_0). En la Fig. 2 se presenta tal estructura, donde a partir de la versión de modelo inicial (versión de modelo m_0 en Fig. 2), pueden obtenerse sucesivas versiones de la misma (versión de modelo m_i, y versión de modelo m_k). En la parte inferior de la Fig. 2 se ejemplifica la descripción visual de la arquitectura de software correspondiente a las versiones de modelo m_k y m_q. En este caso, se trata de dos versiones de modelo consecutivas tomadas de una porción del caso de estudio que se abordará posteriormente, en donde se obtiene la versión m_q por aplicación de la secuencia de operaciones ϕ_q sobre la versión de modelo m_k, en donde la secuencia de operaciones incluye cierta operación de refinamiento (*applyMVC*) que permite transformar al componente *WebApplication* en un conjunto de componentes (*Model*, *View* y *Controller*) y conectores con una configuración particular.

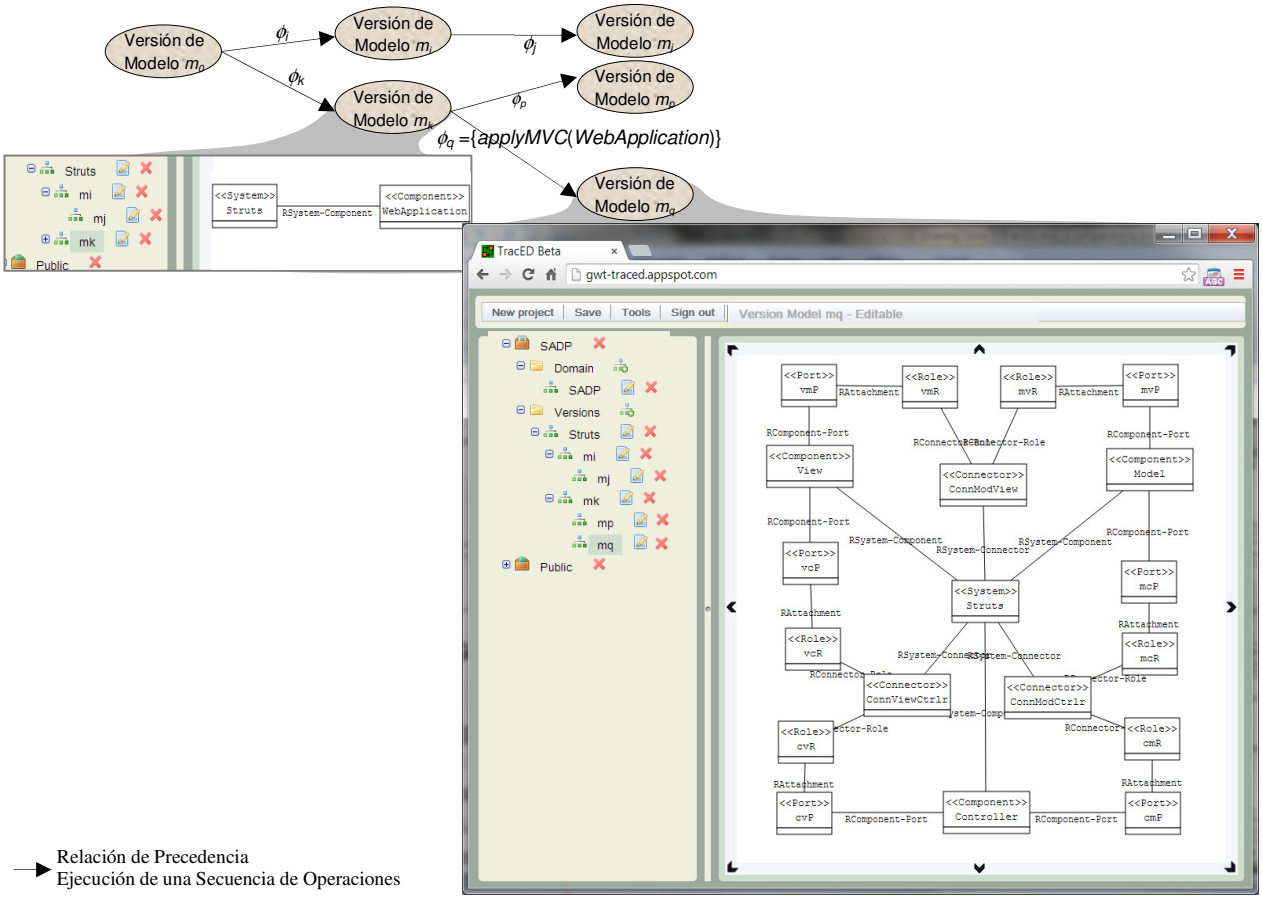


Figura 2. Ejemplo de una Evolución de Versiones de Modelo de una Arquitectura.

2.1. Editor de Dominio

El Editor de Dominio (Fig. 1) permite la definición de un dominio de diseño. El dominio se especifica en términos de los *conceptos* o *tipos de objetos de diseño* que se desean modelar y las *operaciones* que son aplicables a los mismos. Los conceptos están organizados en una estructura de árbol (Concept Tree en Fig. 3). Cada concepto puede tener cero o más descendientes pero un único padre. Cada concepto posee un conjunto de *propiedades* o atributos. Cada atributo es un par compuesto por un nombre y un tipo de dato primitivo. En la versión implementada se definieron los tipos de datos primitivos String, Integer, Boolean, y Float. En el ejemplo de la Fig. 3, el concepto *Component* posee dos atributos de tipo String, *description* y *type*. El primero describe al componente, y el segundo permite especificar si el componente es un módulo, un filtro, un almacenamiento de datos, etc.

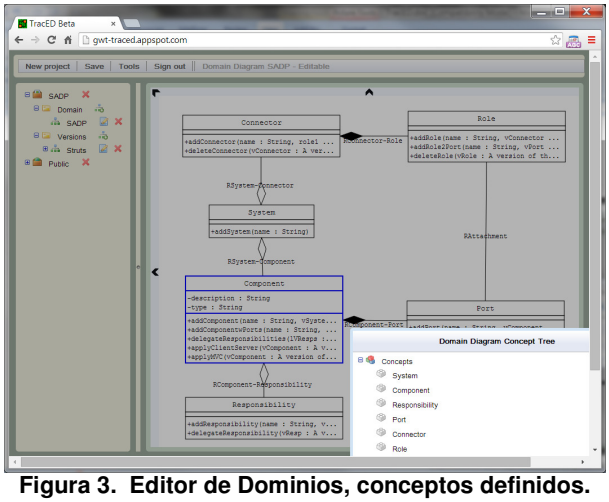
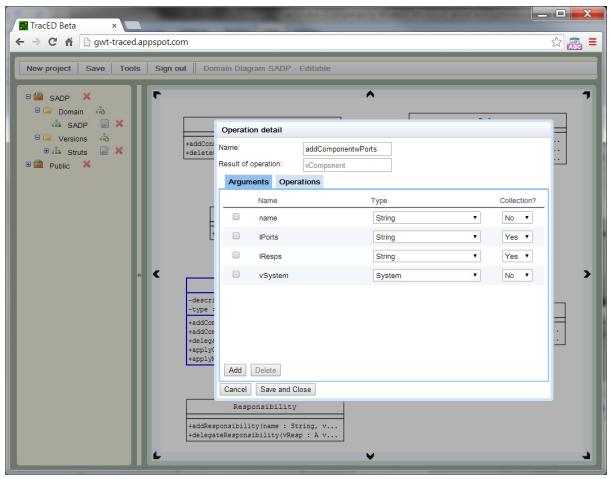


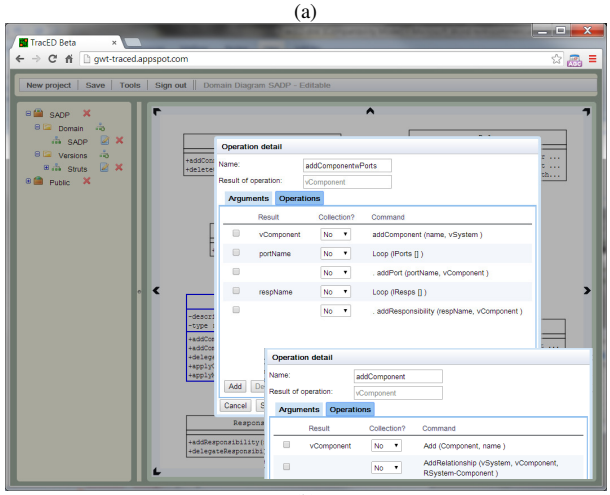
Figura 3. Editor de Dominios, conceptos definidos.

El Editor de Dominio permite la creación de relaciones binarias entre los conceptos y la especificación de operaciones, las cuales se ejecutarán luego desde el Administrador de Versiones. La Fig. 4 ilustra los argumentos y la definición de la operación *addComponentwPorts*. En este caso, los argumentos de *addComponentwPorts* (Fig. 4 (a)) son: un nombre de componente (*name*), una lista de nombres de puertos (*lPorts*), una lista de nombres de responsabilidades (*lResps*) y una versión de *System* (*vSystem*).

Como se ilustra en la Fig. 4 (b), la operación *addComponentwPorts* es definida en término de operaciones auxiliares (*AddRelationship*, *Loop*), y operaciones definidas en el dominio (*addComponent*, *addResponsibility*, *addPort*). En primer lugar se genera una instancia de *Component* (*vComponent*), cuyo nombre es *name*, mediante la operación *addComponent*. Luego, para cada nombre (*portName*) de la lista de puertos (*lPorts[]*) se instancia y agrega un puerto al componente recientemente creado (*addPort*(*portName*,*vComponent*)). En forma similar, se agregan las responsabilidades (*addResponsibility*(*respName*, *vComponent*)).



(a)



(b)

Figura 4. Editor de Dominios, edición de una operación. (a) argumentos de una operación. (b) especificación del cuerpo de una operación.

La versión de componente *vComponent* es agregada mediante la ejecución de *addComponent*. La ventana incluida en la parte inferior de la Fig. 4(b) muestra la especificación de *addComponent*, la cual incluye la operación primitiva *add* y la operación auxiliar *AddRelationship*. Esta última relaciona la instancia de componente generada por *add* (*vComponent*) con la instancia del sistema (*vSystem*).

A partir de simples operaciones, como las detalladas previamente, es posible definir operaciones de más alto nivel que representan decisiones del proceso de diseño. Por ejemplo, la operación *applyMVC* (Fig. 5 y Fig. 6)

especifica la aplicación del patrón de diseño Modelo-Vista-Controlador.

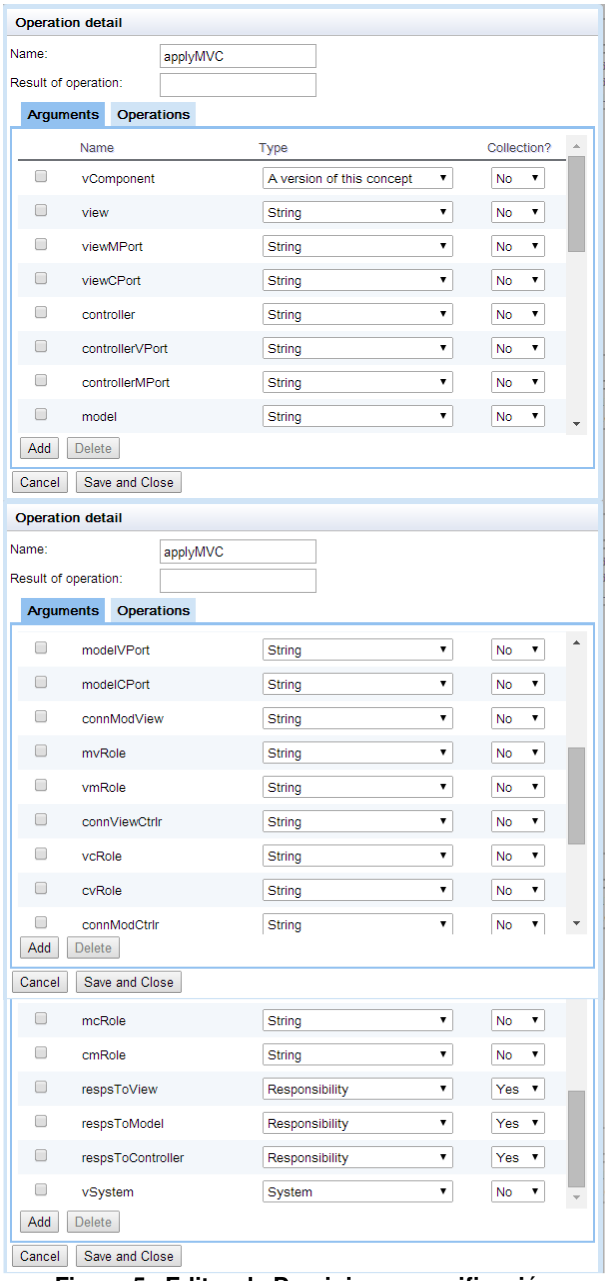


Figura 5. Editor de Dominios, especificación argumentos de *applyMVC*.

En la Fig. 5 se incluyen los argumentos de *applyMVC*. La operación es definida en el concepto *Component*, en consecuencia, el primer argumento, *vComponent*, es una versión de un *Component*, el cual será refinado por los elementos de MVC. Luego, los otros argumentos permiten especificar el nombre a los distintos elementos que serán creados al ejecutar *applyMVC*, como ser el nombre de la vista (*view*), el modelo (*model*) y el controlador (*controller*), o el conjunto de responsabilidades que serán delegadas a los distintos elementos (*respsToView*, *respsToModel*, *respsToController* en Fig. 5).

En la Fig. 6 se muestra el cuerpo de *applyMVC*. La operación es especificada en términos de operaciones previamente definidas en el dominio, tales como *addComponent*, *addPort*, *addConnector*, *delegateResponsibilities*, y de la operación primitiva *delete*, la cual elimina la versión del componente que es refinado (*vComponent*).

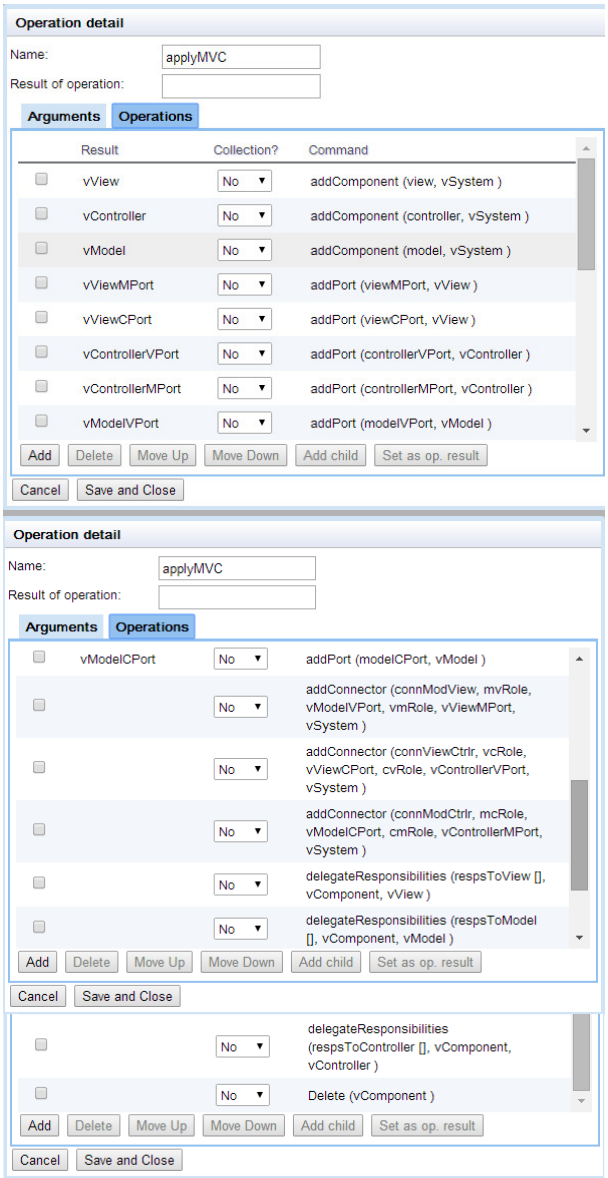


Figura 6. Editor de Dominios, especificación del cuerpo de de *applyMVC*.

2.2. Administrador de Versiones

El Administrador de Versiones permite la ejecución de un proyecto de diseño, capturando las operaciones realizadas y las versiones generadas.

Cuando se inicia un nuevo proyecto de diseño, se selecciona un dominio existente. De esta manera, la evolución de un proyecto se basa en la ejecución de las operaciones definidas en el dominio seleccionado y la instanciación de los conceptos de tal dominio. Además, TracEDaaS, mediante su Administrador de Versiones, permite mantener información sobre:

i) versiones de modelo predecesora y sucesora (si existe) de cada versión de modelo;

ii) relaciones de historia que trazan las secuencias de operaciones aplicadas y permiten generar una nueva versión de modelo;

iii) referencias al conjunto de versiones de objetos resultante de ejecutar las distintas operaciones.

La ejecución de una operación consiste en la ejecución de cada una de las operaciones que la especifican, además, se genera un objeto que mantiene información de la operación ejecutada, enlazando las versiones de objeto predecesoras (argumentos de la operación), con las versiones de objetos sucesoras (resultante de la operación). Esto permite reconstruir la historia de una versión de modelo a partir de la versión de modelo inicial y la aplicación de las operaciones llevadas a cabo.

A continuación, se muestra el funcionamiento del administrador de versiones empleando como ejemplo una porción del desarrollo del proyecto de diseño Struts. Una vez iniciado el proceso de diseño, éste se va desarrollando de la siguiente manera: cuando el diseñador desea crear una nueva versión de modelo, debe seleccionar en el árbol de versiones de modelo la versión con la que trabajará, es decir, aquella que será la versión de modelo sobre la que aplicará una secuencia de operaciones para generar una nueva versión de modelo. La versión de modelo seleccionada se denomina versión de modelo actual.

Realizando un click derecho sobre el diagrama, la herramienta muestra las operaciones disponibles para el dominio actual (Fig. 7), disponiéndolas en sub-menúes que clasifican a las operaciones según los distintos tipos de objeto de diseño del dominio. Por ejemplo, si se quiere ejecutar la operación para agregar un objeto de tipo *System*, se selecciona el ítem de menú "*addSystem*" que se ubica bajo el submenú *System*. En la Fig. 7 se observa la ejecución de la operación *addSystem*(*Struts*), la cual agrega la primera versión de objeto a la versión de modelo actual.

Luego, el actor (arquitecto) comienza a construir la estructura de la arquitectura de software. De esta manera, agrega el primer componente a través de la ejecución de una operación *addComponentwPorts*, la cual genera la versión de objeto de tipo *Component* denominada *WebApplication*. *WebApplication* es el componente de la arquitectura que tiene las siguientes responsabilidades: manejar las solicitudes provenientes de los navegadores o clientes (*RRequestHandling*), realizar validaciones (*RValidation*), delegar el control a los componentes específicos (*RControlDelegation*), interactuar con la base de datos (*RDBAccessing*), procesar las acciones solicitadas (*RBusinessLogic*), y actualizar el estado de la aplicación así como de las vistas del usuario (*RDataPreparation*, *RForwardResults*, *RGenerateResponse*). Los objetos responsabilidad (*Responsibility*) mencionados recientemente son incorporados a la versión de modelo *Struts Version 2* como consecuencia de la aplicación de la operación *addComponentwPorts* (Fig. 8).

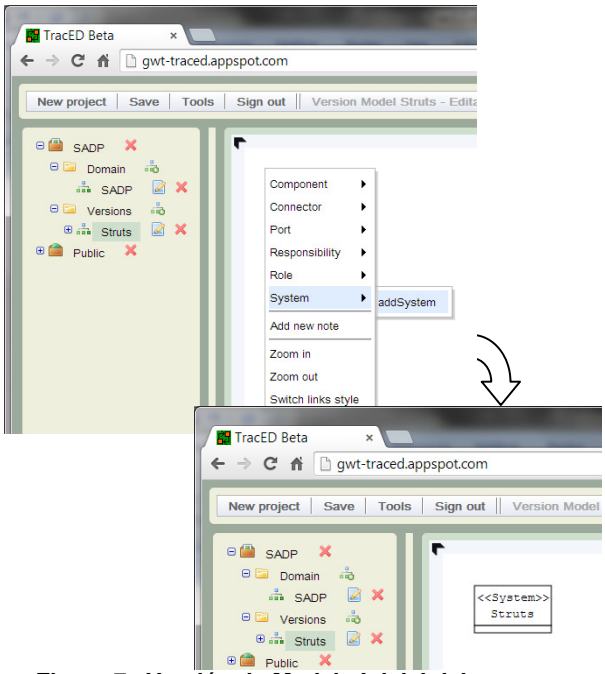


Figura 7. Versión de Modelo Inicial del proyecto Struts, ejecución de addSystem.

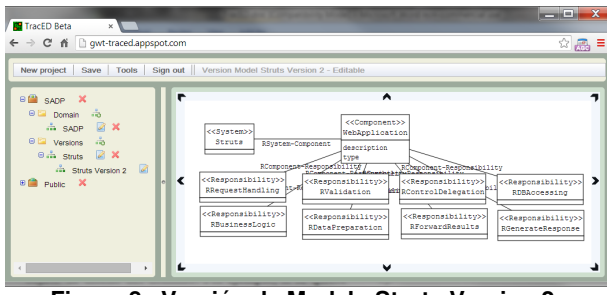


Figura 8. Versión de Modelo Struts Version 2.

Luego, el diseñador considera que es conveniente aplicar el patrón arquitectónico MVC, dado que posibilita la satisfacción de un requerimiento de modificabilidad que fuera inicialmente planteado. El arquitecto lleva a cabo esto mediante la ejecución de la operación *applyMVC* sobre la versión *WebApplication*, especificando los distintos argumentos para la operación (Fig. 9). Como resultado, se obtiene la versión de modelo *Struts Version 3* (Fig. 10), en la cual están presentes tres nuevas versiones de objeto de tipo *Component*: *Model*, *View* y *Controller*.

Entre los componentes *View* y *Controller*, existe un conector denominado *ConnViewCtrlr* (Fig. 10) que representa las interacciones entre ellos y son materializadas por las solicitudes que realiza la vista. Por otro lado, entre el componente *Model* y el componente *Controller*, existe un conector que se encarga de informar los cambios de estado que ocurren en el componente modelo, representado por la versión de objeto *ConnModCtrlr* de tipo conector. También puede observarse (Fig. 10) que la operación agrega otro conector (*ConnModView*) entre las versiones de objeto *Model* y *View* que representa las notificaciones enviadas por el modelo a la vista para que ésta se actualice. Finalmente, la ejecución de la operación *applyMVC* realiza la delegación de las responsabilidades del componente original (*WebApplication* en Fig. 8) en cada uno de los nuevos componentes. La versión de modelo resultante se ilustra en la Fig. 10. Cabe destacar, que la ejecución de la operación, además de modificar la versión de modelo actual, almacena la traza entre los argumentos (en este caso *WebApplication*) y los resultados de la operación. Esto permite navegar por las trazas desde las versiones de objeto origen a las versiones de objeto destino.

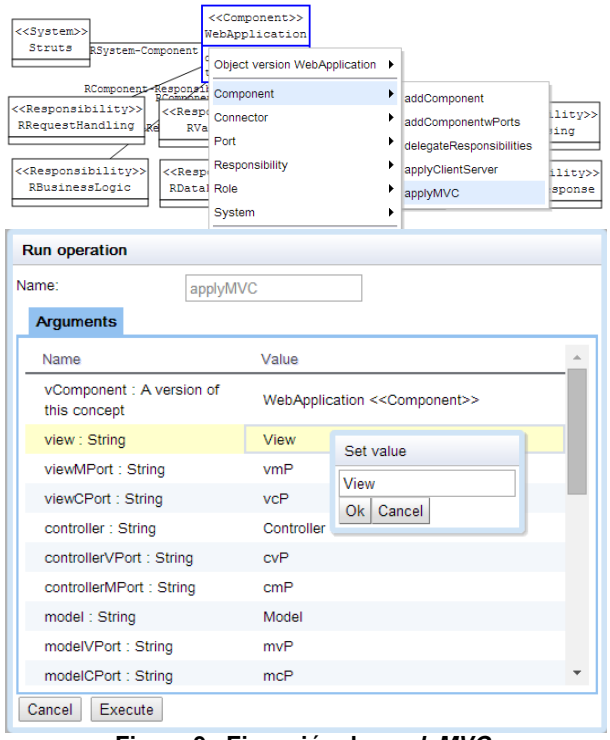


Figura 9. Ejecución de *applyMVC*.

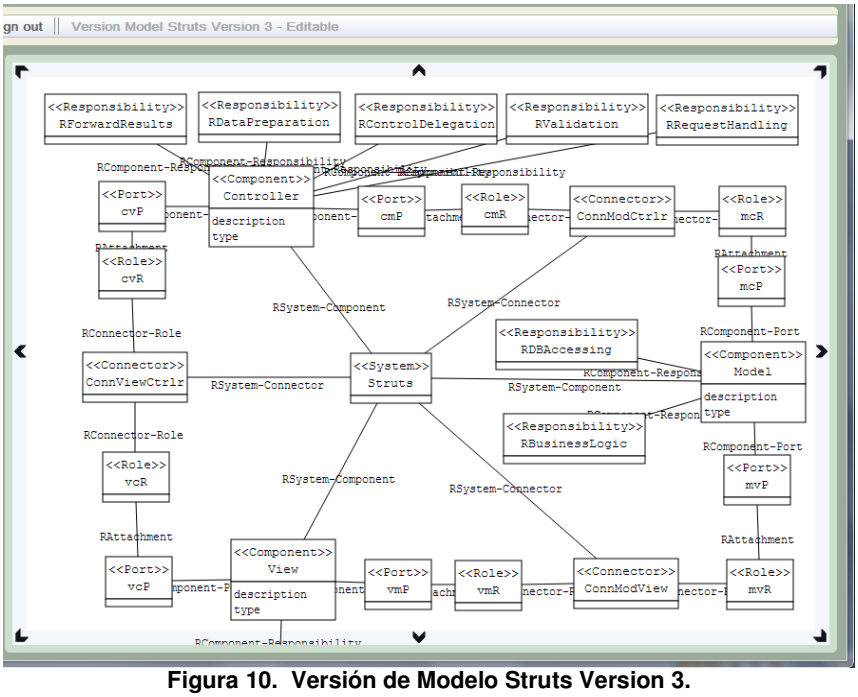


Figura 10. Versión de Modelo Struts Version 3.

Esta información puede luego ser recuperada mediante la consulta de la historia del proceso realizado. La Fig. 11 muestra la historia de cómo fue alcanzada la versión de modelo *Struts Version 3*. La figura presenta parcialmente tres porciones de la ventana de Historia, la cual informa todas las operaciones que han sido aplicadas para evolucionar desde la versión de modelo inicial a la versión de modelo seleccionada (*Struts Version 3*). Por cada operación ejecutada visualiza las operaciones que la componen y los productos generados.

3. TracEDaaS. Implementación en la nube

Para la implementación de TracEDaaS como un servicio en la web se eligió el lenguaje Java, y particularmente el framework GWT de Google. Se seleccionó este framework de desarrollo por las siguientes características que facilitaron la implementación y posterior empleo de TracEDaaS:

- Clientes ricos en el navegador: Las interfaces de usuario del editor de dominios y el administrador de versiones requieren de la edición de diagramas similares a diagramas UML. Entre los enfoques de generar una aplicación que haya que bajar e instalar en la PC y generar la aplicación directamente en un navegador, se eligió este último enfoque por los beneficios que esto presenta al usuario (compatibilidad multi navegador, interfaz optimizada, no se necesitan instalaciones, ni bibliotecas).
- Cloud runtime environment: La aplicación se puede correr en el AppEngine de Google, lo cual brinda un espacio para correr la aplicación en cualquier lugar sin restricciones. Además no requiere de la instalación de un servidor propio. Actualmente es posible acceder a la aplicación desde http://gwt-traced.appspot.com/
- Cloud datastore: La aplicación usa la tecnología de DataStore de Google, esto permite desacoplar la programación de la lógica de la aplicación con el almacenamiento.
- Soporte nativo HTML5: Permite realizar gráficos 2D y 3D en el browser de forma nativa, sin necesidad de instalar complementos adicionales.

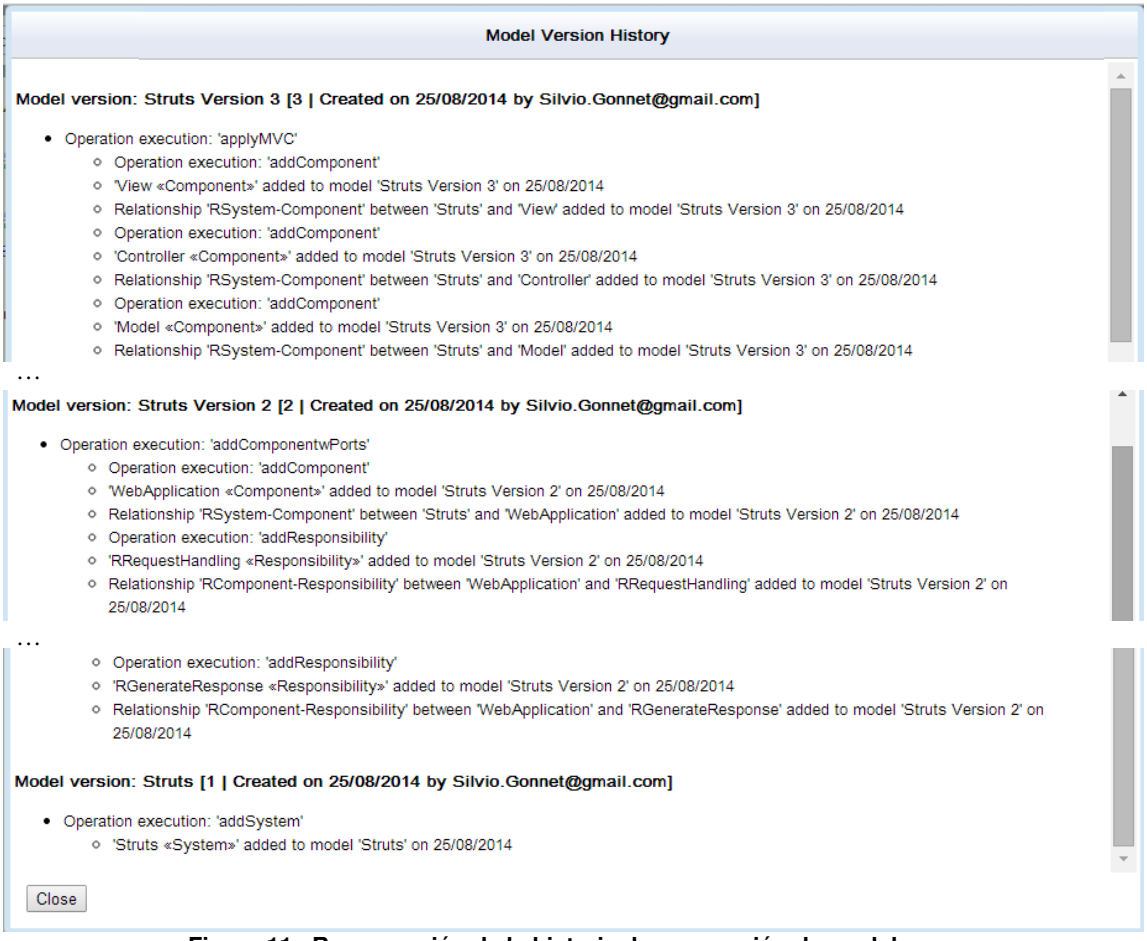


Figura 11. Recuperación de la historia de una versión de modelo.

• API de seguridad: La implementación de la aplicación en el AppEngine de Google permite utilizar como usuarios las cuentas de Google con relativa facilidad, lo cual ahorra el desarrollo de una capa de seguridad y gestión de usuarios.

La Fig. 12 ilustra la arquitectura de TracEDaaS implementada en la nube empleando el framework GWT de Google. Como se puede observar en la figura, la arquitectura de la aplicación está definida por las capas *UI*, *Services*, y *Data Access*. Estas tres capas se describen a continuación.

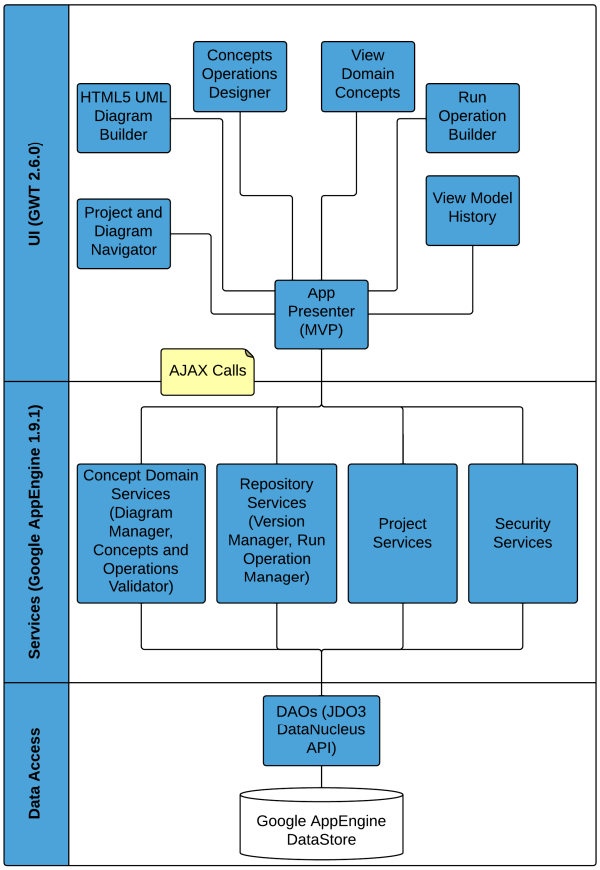


Figura 12. Arquitectura de TracED empleando el framework GWT de Google.

3.1. Capa de presentación o interfaz de usuario (UI)

Esta capa brinda soporte a la realización de los diagramas del editor de dominio y el administrador de versiones. Se diseñó la interfaz de la aplicación con el objetivo de maximizar el área de trabajo y minimizar el área de información y elementos auxiliares. Como resultado de esto, se dispone de 3 áreas bien diferenciadas (ver Fig. 2, versión de modelo m_q):

• Menú superior: Muestra un menú con las acciones:

◦ New Project: crear nuevo proyecto.
◦ Save: guarda versión de modelo actual.
◦ Tools -> Show domains concepts tree: muestra conceptos de un dominio en forma de árbol, como el ilustrado en la Fig. 3.
◦ Tools -> Validate domain operations: valida las operaciones de un dominio para establecer su consistencia.
◦ Tools -> Clone domain: clona un dominio.
◦ Tools -> New model version: crear una nueva versión de modelo a partir de la versión de modelo actual.
◦ Tools -> View model history: ver el historial de cómo se obtuvo una versión de modelo en particular, a partir de la aplicación de una secuencia de operaciones. En la Fig. 11 se ilustró la historia de la versión de modelo *Struts Version 3*.
◦ Tools -> Export to SVG: exportar a un formato imprimible.
◦ Sign out: Salir de la aplicación.
◦ Status bar: Barra de estado de la aplicación. En el editor de dominios visualiza el dominio que está siendo editado. Cuando el administrador de versiones está activo, visualiza la versión de modelo que está siendo editada.

• Barra de navegación: Muestra los proyectos, con su correspondientes Dominios y Versiones de modelos existentes. Los proyectos pueden ser públicos o privados (no compartidos).

• Área gráfica de trabajo: El área de trabajo principal, donde se editan los diagramas del editor de dominios y el administrador de versiones. Tiene un menú contextual dependiente del lugar donde esté posicionado.

Esta capa fue programada en Java, mediante el framework GWT 2.6.0. Se hace un uso intensivo de la biblioteca de widgets provista por el mismo. Asimismo, para la realización de gráficos, se ha utilizado como base el framework Tatami (https://code.google.com/p/tatami/), que es una implementación en GWT del framework DOJO (http://dojotoolkit.org/), que se utiliza para hacer gráficas nativas HTML5 en la web.

Esta capa está conformada por el componente principal *App Presenter* y seis componentes con responsabilidades específicas: *Project and Diagram Navigator*, *HTML 5 UML Diagram Builder*, *Concepts Operation Designer*, *View Domain Concepts Tree*, *Run Operation Builder*, y *View Model History*. Toda esta capa, se compila con el compilador GWT, que los transforma en código javascript optimizado para cada navegador y corre totalmente en la máquina del usuario.

3.1.1. App Presenter. Los componentes de la interfaz de usuario están programados mediante el patrón Model-View-Presenter, el cual permite distribuir el código entre la vista y la lógica, y ayuda a crear módulos bien

estructurados y testeables. De esta forma, la vista (View) puede ser diferente para cada tipo de dispositivo deseado, pero las llamadas a los servicios se hacen a través del Presenter con llamadas RPC.

Se hace uso de GWT EventBus para el manejo de eventos en el front-end.

Esta capa UI, se comunica con la capa de servicios a través de llamadas asincrónicas (*AJAX*), lo cual permite que la página no se tenga que recargar y genera una mejor experiencia de usuario. Por esta razón, se utilizó el patrón DTO (Data Transfer Objects), para minimizar el paso de objetos complejos entre el cliente y el server y simplificar el proceso de serialización de los mismos.

3.1.2. Project and Diagram Navigator. Este módulo permite organizar y gestionar los diagramas en proyectos, permitiendo crear, copiar, mover, borrar y hacerlos públicos y privados. Las versiones de modelos se presentan en una estructura de árbol respetando el enfoque conceptual introducido en la Fig. 2.

3.1.3. HTML5 UML Diagram Builder. Este componente se encarga de generar y manipular diagramas en forma ágil con tecnología Drag & Drop. Su uso difiere si se utiliza en el Editor de Dominios o en el Administrador de Versiones. En el Editor de Dominio permite definir un dominio, con sus conceptos y relaciones. Los conceptos están representados por un nombre, propiedades y operaciones. Para la definición de las propiedades y el nombre se utiliza un analizador léxico que permite identificar el nombre, el tipo y la presencia de estereotipos.

3.1.4. Concepts Operation Designer. Este módulo es un componente principal del Editor de Dominio, permite la especificación de operaciones. Para la definición del cuerpo de una operación se utiliza una combinación de comandos básicos que se pueden combinar, creando operaciones complejas. En la definición de las operaciones se establece el binding de los argumentos de las mismas, es decir, el elemento del cual va a tomar el valor al momento de correr la operación.

3.1.5. View Domain Concepts Tree. Componente del Editor de Dominio para ver el árbol jerárquico de los conceptos. Útil para ver todos los conceptos estructurados según las relaciones de generalización-especialización.

3.1.6. Run Operation Builder. Este módulo es un componente principal del Administrador de Versiones. Nos permite ir construyendo un modelo de versiones de un dominio en particular, ejecutando las operaciones asociadas al dominio. Se encarga de mostrar las operaciones disponibles y de devolver los resultados para su visualización gráfica en el diagrama.

3.1.7. View Model History. Componente del Administrador de Versiones para ver el historial de una versión en particular del modelo.

3.2. Capa de servicios

Esta capa corre en el server y contiene la mayoría de la lógica de la aplicación. La capa de servicios (*Services*, Fig. 12) está compuesta de 4 componentes: *Concept Domain Services*, *Repository Services*, *Project Services*, y *Securitiy Services*. Los dos primeros implementan la lógica del editor de dominios y el administrador de versiones respectivamente.

3.2.1. Concept Domain Services. Este componente posee dos servicios: *Diagram Manager* y *Concepts and Operations Validator*. El *Diagram Manager* se encarga de la gestión de los diferentes objetos de un dominio, identificando y validando *Conceptos* y *Relaciones*. El servicio *Concepts and Operations Validator* posee la responsabilidad de validar las operaciones definidas en los conceptos. La validación se realiza en término de los bindings establecidos y los argumentos requeridos por las operaciones.

3.2.2. Repository Services. Este componente está compuesto por los servicios *Version Manager* y *Run Operation Manager*. El servicio *Version Manager* se encarga de la gestión de versiones de un modelo: crear versión de modelo inicial, crear nuevas versiones de modelos a partir de versiones existentes, mantener el historial de operaciones ejecutadas y las diferentes versiones de modelo. El servicio *Run Operation Manager* es el encargado de ejecutar las operaciones de una versión del modelo en particular, generando nuevas versiones de objetos. Este servicio funciona como un intérprete de comandos, el cual interpreta la operación a ejecutar, que siempre es una sucesión de comandos que se ejecutan en forma secuencial. La ejecución de la operación modifica la versión del modelo y va guardando las trazas que permiten seguir la historia del diseño. Como resultado de la ejecución de una operación, devuelve un resultado formado por un log (historia) y un conjunto de componentes para ser ubicados correctamente en el diagrama.

3.2.3. Project Services. Se encarga de gestionar los proyectos en carpetas agrupadas por dominio y versiones de modelo.

3.2.4. Security Services. Se encarga de la autenticación de usuario, utilizando el API de seguridad de google,

permitiendo autenticar contra una credencial existente de Gmail.

3.3. Capa de persistencia

La capa de persistencia está programada siguiendo la especificación JDO (Java Data Object). El API utilizado para la implementación es DataNucleus JDO3.

En tiempo de ejecución, el API crea una PersistenceManagerFactory (PMF) que proporciona la conectividad con el DataStore. Como DataStore, se utiliza Google AppEngine DataStore, que es una base de datos cloud o en la nube, de tipo NoSQL, que provee acceso robusto y escalable, que permite almacenar los gráficos en forma serializada como BLOBs.

4. Conclusiones

En el trabajo se propone una herramienta en la nube que permite implementar un enfoque operacional para la captura y trazabilidad del proceso de diseño de software. La herramienta incluye un editor de dominio que permite la definición del lenguaje de modelado y las operaciones aplicables a los conceptos identificados. Además, TracEDaaS provee los mecanismos necesarios para llevar a cabo un proceso de diseño mediante la ejecución de una secuencia de operaciones definida en el dominio de trabajo. Esta secuencia de operaciones permite representar las decisiones de diseño realizadas en la evolución de una versión de modelo predecesora a una sucesora.

Cabe destacar que TracEDaaS fue desarrollada con el objetivo de verificar que es posible aplicar el modelo operacional propuesto para la captura y trazabilidad del proceso de diseño, y no intenta remplazar herramientas tradicionales de diseño. Al contrario, TracEDaaS debería ser integrado a herramientas existentes y funcionar en modo background. De esta manera, TracEDaaS capturaría las operaciones realizadas en ambientes de diseño tradicionales.

5. Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET, y la Universidad Tecnológica Nacional. Se agradece el apoyo brindado por estas instituciones.

6. Referencias

[1] Alexander, I., "A taxonomy of stakeholders, human roles in system development", *Issues and Trends in Technology and Human Interaction*, B.C. Stahl (ed.), pp. 25–71, IRM Press, 2006.

[2] Antoniol, G., Caprile, B., Potrich, A., Tonella, P., "Design-code traceability recovery: selecting the basic linkage properties", *Science of Computer Programming*, 40, pp. 213-234, 2001.

[3] Asuncion, H., Francois, F., Taylor, R., "An end-to-end industrial software traceability tool", *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering (ESEC-FSE'07)*, pp. 115-124, 2007.

[4] Ballejos, L., Montagna, JM., "Method for stakeholder identification in interorganizational environments", *Requirements Engineering*, 13, 4, pp. 281-297, 2008.

[5] Carnduff, T., Goonetillake, J., "Configuration management in evolutionary engineering design using versioning and integrity constraints", *Advances in Engineering Software*, 35, pp. 161-177, 2004.

[6] Carignano, M.C., Gonnet, S., Leone, H., "A model to represent architectural design rationale", *Proceedings of Joint Working IEEE/IFIP Conference on Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009*, pp. 301-304, 2009.

[7] Carignano, M.C., Gonnet, S., Leone, H., "Especificación del Concepto de Razonamiento del Estándar ISO/IEC 42010:2007", *Proceedings Conferencia Lationamericana de Informática (CLEI 2011)*, pp. 1124-1139, 2011.

[8] Carignano, M.C., Gonnet, S., Leone, H., "Reasoning and Reuse in Software Architecture Design: Practices in the Argentine Industry", *Electronic Journal of SADIO*, 12, 1, 2013.

[9] Cleland-Huang, J., Chang, C., Christensen, M., "Event-Based Traceability for Managing Evolutionary Change", *IEEE Transactions on Software Engineering*, 29, 9, pp. 796-810, 2003.

[10] Conklin, J., Begeman, M.L., "gIBIS: a hypertext tool for exploratory policy discussion", *ACM Transactions on Information Systems*, 6, 4, pp. 303-331, 1988.

[11] Dutoit, A., McCall, R., Mistrik, I., Paech, B. (eds.), *Rationale Management in Software Engineering*, Springer, 2006.

[12] Gonnet, S., Henning, G., Leone, H., "A Model for Capturing and Representing the Engineering Design Process", *Expert System with Application*, 33, 4, pp. 881-902, 2007.

[13] Gonnet, S., Roldán, M.L., Leone, H., Henning, G., "An Operational Perspective for Capturing the Engineering Design Process", *Modelling and Management of Engineering Processes*, P. Heisig, P.J. Clarkson, S. Vajna (eds.), pp. 103-114, 2010.

[14] Gotel, O., Cleland-Huang, J., Hayes, J., Zisman, A., Egyed, A., Grünbacher, P., Dekhtyar, A., Antoniol, G., Maletic, J., Mäder, P., "Traceability Fundamentals", *Software and Systems Traceability*, J. Cleland-Huang, O. Gotel, A. Zisman (Eds.), pp. 3-22, Springer, 2012.

[15] Jansen, A., Bosch, J., "Software Architecture as a Set of Architectural Design Decisions", *Proceedings of the 5th IEEE/IFIP Working Conference on Software Architecture (WICSA 2005)*, 2005.

[16] Kruchten, P., "An Ontology of Architectural Design Decisions", *Proceedings of 2nd Groningen Workshop on Software Variability Management*, 2004.

[17] Kruchten, P., Lago, P., van Vliet, H., Wolf, T., "Building up and Exploiting Architectural Knowledge", *Working Conference on Software Architecture (WICSA)*, 2005.

[18] Mäder, P., Gotel, O., "Towards automated traceability maintenance", *Journal of Systems and Software*, 85, 10, pp. 2205-2227, 2012.

[19] Moran, T., Carrol, J. (eds.), *Design Rationale: Concepts, Techniques, and Use*, Lawrence Erlbaum, 1995.

[20] Murta, L., van der Hoek, A., Werner, C., "ArchTrace: Policy-Based Support for Managing Evolving Architecture-to-Implementation Traceability Links", *21st IEEE/ACM International Conference on Automated Software Engineering (ASE '06)*, pp. 135-144, 2006.

[21] Object Management Group (OMG), *OMG Unified Modeling Language (OMG UML), Superstructure*, Version 2.4, 2011.

[22] Potts, C., Bruns, G., "Recording the reasons for design decisions", *Proceedings of the 10th International Conference on Software Engineering (ICSE'98)*, pp. 418-427, 1988.

[23] Ramesh, B., Jarke, M., "Toward reference models for requirements traceability", *IEEE Transactions on Software Engineering*, 27, 1, pp. 58-93, 2001.

[24] Roldán, M.L., Gonnet, S., Leone, H., "TracED: A Tool for Capturing and Tracing Engineering Design Processes", *Advances in Engineering Software*, 41, 9, pp. 1087-1109, 2010.

[25] Roldán, M.L., Gonnet, S., Leone, H., "Knowledge Representation of the Software Architecture Design Process based on Situation Calculus", *Expert Systems*, 30, 1, pp. 34-53, 2013.

[26] Schwarz, H., Ebert, J., Winter, A., "Graph-based traceability: a comprehensive approach", *Software System Model*, 9, pp. 473-492, 2010.

[27] Tang, A., Han, J., "Architecture Rationalization: a Methodology for Architecture Verifiability, Traceability and Completeness", *12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005)*, pp. 135-144, 2005.

[28] Völzer, H., MacDonald, A., Atchison, B., Hanlon, A., Lindsay, P., Strooper, P., "SubCM: A Tool for Improved Visibility of Software Change in an Industrial Setting", *IEEE Transactions on Software Engineering*, 30, 10, pp. 675-693, 2004.

[29] Westfechtel, B., "Models and tools for managing development process", *Lecture notes in computer science*, 1646, 1998.

[30] Winkler, S., von Pilgrim, J., "A survey of traceability in requirements engineering and model-driven development", *Software System Model*, 9, pp. 529-565, 2010.

Ontología para el Análisis Forense de Correo Electrónico

Beatriz P. de Gallo[a], Marcela Vegetti[b], Horacio Leone[b]
[a]*I.Es.I.Ing. /Facultad de Ingeniería, Universidad Católica de Salta*
Campo Castañares S/N, Salta, Argentina
[b]*INGAR/ Facultad Regional Santa Fe UTN*
Avellaneda 3657, Santa Fe, Argentina
[a]*bgallo@ucasal.net,* [b]*{mvegetti, hleone}@santafe-conicet.gov.ar*

Abstract

La gran cantidad de información técnica resultante del análisis forense de un correo electrónico debe insertarse en el conjunto de pruebas documentales de la causa judicial que lo aborda, colocándolo en un estadio de lectura que facilite la interpretación de esos datos técnicos por parte de los profesionales de la criminalística y el derecho. Resulta conveniente contar con un marco de referencia basado en la conceptualización formal del universo de discusión. Y en particular, las ontologías resultan una herramienta de uso pluridisciplinar para facilitar el análisis de la prueba documental, por parte de todo los actores (abogados, jueces, investigadores y peritos). Este trabajo tiene como objetivo sentar las bases para el desarrollo de una ontología que colabore en la interpretación de los datos producto del análisis forense de un correo electrónico.

1. Introducción

Cada día toma mayor importancia el uso del correo electrónico, en su carácter de ***registro formal de una comunicación entre dos partes***. En el contexto legal, durante los últimos años ha crecido exponencialmente la presentación de correos electrónicos como prueba documental en las causas judiciales[1].

En su mayoría, el análisis forense de correos electrónicos consiste en responder dos cuestiones sustanciales para el carácter de prueba documental, que este elemento tiene:

- la ***autenticación del correo***, es decir, la validación de que este documento digital presentado como prueba legal es verdadero y no ha sido alterado ni manipulado; y
- la ***identificación del autor/receptor*** del mismo.

Queda por resolver la discusión planteada respecto de si un análisis forense puede aseverar con plena certeza que el *contenido* del correo electrónico no ha sido adulterado y validarlo en su carácter de prueba legal[2].

Si bien la Forensia Digital avanzó en concordancia con la tecnología, es necesario aún trabajar un aspecto que no es propiamente del ámbito tecnológico y que genera un conjunto de interrogantes que impactan grandemente en los resultados que se obtienen, i.e., la ***interpretación de los resultados***.

El volumen de datos que se obtiene al realizar el análisis forense debe ser interpretado a la luz de la pesquisa. La enorme cantidad de información técnica resultante del análisis de un correo electrónico debe insertarse en el conjunto de pruebas documentales de la causa judicial, colocándolo en un estadio de lectura que facilite la interpretación de esos datos técnicos por parte de los profesionales de la criminalística y el derecho. Se requiere mucho más que la identificación de una dirección IP (Internet Protocol) o la trazabilidad del correo electrónico. Hoy en día se exige que estos datos se presenten ***sistemáticamente*** y ***semánticamente*** en el marco de la causa judicial, no como información técnica, sino como dato documental.

En el contexto de este requerimiento "no técnico", se encuentra la motivación de este trabajo. Resulta conveniente contar con un marco de referencia basado en la conceptualización formal del universo de discusión. Y en particular, las ontologías resultan una herramienta universal o pluridisciplinar para facilitar el análisis de la prueba documental, por parte de todo los actores (abogados, jueces, investigadores y peritos).

Existe abundante bibliografía a la que se puede recurrir para tomar los conceptos iniciales sobre ontologías. La definición de Gruber[1] parte del concepto de *conceptualización* como una abstracción o visión simplificada del universo que queremos representar con algún propósito. Las ontologías son una representación explícita o formal de esa conceptualización, recurriendo a la representación de los diferentes elementos que conforman el universo de discusión (objetos, conceptos y otras entidades), así como las relaciones que los vinculan.

1 De cada 10 causas judiciales en los que se solicitó la participación de un perito informático, 26 % tratan acerca de la autenticación y autoría de correos electrónicos. (Fuentes propias).

[2] En el ámbito del Derecho existen posturas contrapuestas acerca de si la verificación de autenticidad de un correo electrónico a partir del análisis forense, abarca (o nó) a su contenido, por ello, no se incluye esta cuestión en la problemática del presente trabajo.

Así, este trabajo tiene como objetivo sentar las bases para el desarrollo de una ontología que colabore en la interpretación de los datos producto del análisis forense de un correo electrónico.

La organización de este trabajo es la siguiente: la sección 2 describe el objeto de estudio (correo electrónico) señalando los aspectos de interés para el análisis forense y refiere brevemente el marco teórico de las ontologías. En la sección 3 se desarrollan las fases de especificación y conceptualización de una Ontología para el Análisis Forense de Correos Electrónicos. Finalmente, en la sección 4 se presentan las conclusiones y trabajos futuros.

2. Trabajos Relacionados

2.1. El Correo Electrónico

2.1.1. Definición y características propias

La literatura específica contiene diferentes definiciones del término correo electrónico. A los fines del presente trabajo, puede tomarse como válida la señalada por las Directrices de la Unión Europea 2002/58/CE [2] relativas a la protección de datos, en las que se definen el Correo Electrónico como "*Todo mensaje de texto, voz, sonido o imagen enviado a través de una red de comunicación pública que puede almacenarse en la red o en el equipo terminal del receptor hasta que éste acceda al mismo*".

El correo electrónico (o e-mail) se ha transformado en el medio de comunicación más utilizado en el tráfico de red facilitando grandemente la comunicación entre las personas. Además de acortar tiempos y distancia, permite el intercambio de múltiples tipos de datos (video, imagen, audio) y se encuentra accesible en prácticamente todos los medios de comunicación tecnológicos, habiendo avanzado rápidamente en la telefonía celular. Esta última característica de "portabilidad" abre instancias de comunicación que antes no estaban presentes, reforzando la inmediatez de la comunicación interpersonal, con el agregado de que ahora existe ***un registro de esta comunicación***. Si bien en el correo electrónico se adopta lenguaje coloquial, y es muy utilizado para la comunicación informal, es importante reconocer que es posible recuperar la conversación y utilizarla como prueba de que tal comunicación existió.

Desde el punto de vista legal, el correo electrónico tiene interés como documento probatorio en un juicio, por lo que resulta importante introducirlo con la fuerza y el rigor técnico suficiente para que actúe en el proceso judicial de igual manera que lo hiciera cualquier otra prueba material.

Es importante señalar que la característica de volatilidad de los datos digitales, impacta negativamente en el reconocimiento de un correo electrónico (o cualquier otro componente digital) como prueba documental.

Los profesionales del foro judicial quieren "ver" la prueba, darle forma, buscar el origen, su historia, imaginar todo lo que hay alrededor de este componente, cual si fuera un "arma homicida". Recién en esta instancia pueden reconocer la validez de la prueba digital, i.e., una vez que logran ver la "sustancia" y "consistencia" de elemento probatorio.

2.1.2. Naturaleza Jurídica del correo electrónico

Relacionado con el contexto jurídico, Castro Bonilla [3] menciona tres enfoques o miradas diferentes que se pueden considerar en un correo electrónico:

a) Como correspondencia o comunicación: con idéntica naturaleza que el correo postal tradicional, se encuentra protegido por las leyes que regulan la correspondencia epistolar. Tanto los datos recibidos cuanto los enviados desde la cuenta de correo, constituyen elementos protegidos bajo el principio de inviolabilidad de las comunicaciones. En nuestro país, la Ley 26.388 incluye el término "correo electrónico" en el tipo de violación de correspondencia privada establecida por los arts. 153 y 154 del Código Penal.
b) Como conjunto de datos: al ser datos personales, su manipulación se encuentra supeditada a las normas relativas a la protección de datos personales. A partir del análisis forense de un correo electrónico es posible identificar una serie de datos del individuo (receptor/emisor del correo) que vulnera el derecho a la autodeterminación informativa de una persona. En Argentina, la Ley 25.326 *"…tiene por objeto la protección integral de los datos personales asentados en archivos, registros, bancos de datos, u otros medios técnicos de tratamiento de datos, sean éstos públicos, o privados destinados a dar informes, para garantizar el derecho al honor y a la intimidad de las personas, así como también el acceso a la información que sobre las mismas se registre,…".*
c) Como transmisor de material protegido por derechos de autor: al ser un medio de transmisión de datos de gran diversidad de formatos, hace posible la difusión de material protegido por derechos de autor, con un impacto notorio en la vulnerabilidad de la propiedad intelectual del material contenido o adjunto en un correo electrónico.

2.1.3. Estructura y componentes

A los fines del presente trabajo, se abordará brevemente aquellos aspectos del correo electrónico que resulten de interés para un análisis forense.

Tomando como base la tipificación propuesta por Banday [4] para el análisis forense de un correo electrónico se puede identificar tres componentes principales: los actores participantes en la transmisión, la arquitectura lógica y la estructura interna de un correo electrónico.

En referencia a los ***actores participantes*** se puede decir que, si bien la comunicación de un correo electrónico requiere de un emisor y un receptor del mensaje, no son éstos los únicos partícipes de la transmisión. Existen procesos responsables de sostener el servicio –denominados *actores*- que actúan internamente durante la transmisión.

La ***arquitectura lógica*** de funcionamiento del envío/recepción de mail consta de varios componentes de hardware y software. Además de los dispositivos utilizados por el *Autor/Receptor*, sean éstos una computadora, un celular o cualquier otro equipo utilizado para el acceso al sistema de correo electrónico, se requiere de objetos transparentes al usuario principal que actúan en el proceso de creación, envío, transmisión, entrega y lectura del e-mail, como por ejemplo: componentes de la capa inferior de TCP/IP (routers, modem) y de la capa de aplicación de TCP/IP (nodos de e-mail) así como los diversos protocolos y paquetes de software utilizados (SMTP, HTTP, clientes de correo, etc.).

En cuanto a la ***estructura interna,*** un correo electrónico contiene un conjunto de *campos* asociados en *layers* que cumplen una función específica en la transmisión de datos. De ellos interesa identificar en particular: la cabecera del mensaje, el cuerpo o contenido, los archivos adjuntos asociados al mensaje, y las direcciones IP involucradas en la transmisión.

2.1.4. Análisis Forense de un Correo Electrónico

Una pericia [5] es un conjunto de operaciones técnicas científicas puestas en práctica para el esclarecimiento de un posible hecho ilícito y ordenadas por el Tribunal interviniente. En cuanto a los ***puntos de pericia***, su ofrecimiento permitirá al Juez determinar la procedencia de la prueba, es decir, la congruencia entre los aspectos a conocer y la necesidad de un técnico para que lo asesore. Los puntos periciales se proponen en un pliego que señala las cuestiones técnicas, de manera clara y precisa, siempre referidas al tema que se dilucida en la litis y que técnicamente puedan ser respondidas por el Perito[3].

Usualmente los puntos de pericia referidos a correos electrónicos abordan las siguientes cuestiones:

- Verificar la autenticidad, dirección, fecha y hora de emisión y recepción, autoría si fuera posible de los correos electrónicos cuyos impresos se adjuntan, así como cualquier otra información que estime relevante.
- Determinar la validez de origen de los e-mail enviados por A
- Expedirse acerca de la existencia de los envíos de los correos electrónicos emitidos entre A y B
- Informar si las cuentas de correo electrónico a@dominio.com y b@dominio.com están o estuvieron activas en los períodos consignados
- Informar si A tiene como correo electrónico la cuenta a@dominio.com
- Informar si en fecha dd/mm/aaaa desde la cuenta a@dominio.com se envió un correo a la cuenta b@dominio.com y en ese caso transcriba el texto del mensaje.
- Llevar a cabo la exploración y análisis de todos los soportes informáticos idóneos para el almacenamiento de información (notebooks, netbooks, pc's, discos rígidos, diskettes, cd's, dvd's, pen drives), instalados o que se encuentren en determinado lugar, ello con el objeto de determinar el soporte informático desde donde se hayan confeccionado documentos y/o remitido e-mails que dieran origen a las actuaciones.
- Realizar una correlación de eventos que relacione los datos de los correos electrónicos recibidos (dirección IP, emisor, destinatario/s, asunto, fecha, hora y observaciones varias) con los datos informados por la empresa X (nombre del usuario, titular del dominio, documento del titular, domicilio y localidad).
- Determinar si los correos que aparecen como enviados o recibidos son de las fechas que indica la parte en su demanda y si son coincidentes con los que mencionan como prueba la parte actora.
- Determinar si es posible que los mensajes enviados o recibidos, como los textos adjuntos que allí figuran, pueden haber sido alterados en sus fechas y horas de emisión o recepción
- Identificar cuáles fueron los equipos de origen y de destino del mensaje, si los servidores o proveedores de Internet – que operan a modo de enlace entre el remitente y el destinatario del e-mail- cuentan con información inherente a la fecha en que se produjo el envío o reenvío del e-mail, duración de conexión, archivos adjuntos, número de identificación de los equipos de origen y destino de las comunicaciones – datos de tráfico realizado.

En particular interesa definir el ***carácter probatorio*** de un correo electrónico, mediante las características de autenticidad y existencia.

[3] A los fines del presente trabajo, se tomará en consideración el contexto más usual de la Forensia Digital, es decir, el ámbito de la justicia, sin perjuicio de que lo dicho aquí se pueda aplicar en los restantes ámbitos en que actúa esta disciplina (investigaciones privadas, fraudes tecnológicos internos, etc.).

Los elementos que permiten verificar la ***autenticidad*** de un correo electrónico son los siguientes:

- la identificación de los datos del remitente (nombre de usuario, cuenta de correo y dirección IP),
- la trazabilidad del mismo (diferentes servicios o agentes que intervienen en la transmisión), y
- los datos del destinatario (nombre de usuario, cuenta de correo y dirección IP.

En cuanto a la ***existencia*** de un correo electrónico, ésta se puede probar fehacientemente cuando se comprueba la presencia del archivo digital del mismo tanto en el dispositivo emisor (o en el servidor del ISP[4] del emisor) como en el dispositivo receptor del correo (o en el servidor ISP del receptor); y ambos archivos digitales son idénticos.

Desde el punto de vista de la forensia digital, existen muchas técnicas y herramientas que ayudan al Perito Informático en el análisis de un correo electrónico.

En cuanto a las *técnicas*, la investigación forense de un correo electrónico se puede abordar desde varias ópticas: el análisis de los datos de cabecera, el análisis de los equipos emisores/receptores del correo, el análisis de los servidores ISP, el análisis de los metadatos ocultos en las aplicaciones utilizadas para la escritura del correo. Todas estas técnicas se utilizan habitualmente en conjunto para la confirmación redundante sobre la autenticidad del correo electrónico.

Respecto de las *herramientas* disponibles para analizar un correo electrónico, se puede citar como las más usuales las siguientes: eMailTrackerPro (analiza los encabezados de correo), Email Tracer (identifica el trazado o camino recorrido por un correo electrónico desde la emisión hasta la recepción), Adcomplain (informe los abusos de mailing), Aid4Mail Forense (utilizado para la migración y conversión de correos de/a diversos formatos), FTK (utilizada para el descifrado de claves), Encase (para la obtención de un archivo imagen del correo con el objeto de preservar la prueba original libre de manipulación), FINALeMAIL (para restaurar y recuperar mails borrados o perdidos).

Incluso existen frameworks integrados para el análisis forense de correos electrónicos, tales como el *Integrated E-mail Análisis Forense Framework (IEFAF)*, propuesto por Hadjidj et alt. [6], que consta de 5 módulos: Navegador Interbase de datos, Explorador de estadísticas, Explorador de minería de datos, submódulo Weka y Explorador de E-mail. Consta además de una interfase gráfica e intuitiva con 5 visores: Visor de Edición de Detalles del e-mail, Visor de mapas para la ubicación geográfica de las IP encontradas en el e-mail, Visor de Estadísticas acerca de la frecuencia de uso de palabras, eventos u objetos repetitivos; Visor de red social que muestra la vinculación entre todos los IP y nombres de clientes de mail; y Visor de minería de datos que opera con Weka.

Para el caso particular de los dispositivos móviles que se encuentran en el mercado, se pueden utilizar las siguientes herramientas de análisis forense: TULP2G, PARABEN y MOBILedit! FORENSIC[7]. Incluso existen herramientas diseñadas específicamente para el análisis forense de determinada tecnología celular, tal como el WOLF de Sixth Legion diseñada específicamente para el celular iPhone 3G [8].

Si bien las técnicas y herramientas mencionadas constituyen el marco formal y científico que califican la profesionalidad y rigor metodológico que se requiere en un análisis forense, los resultados que se obtienen no siempre cumplen su cometido: brindar información fundada sobre los puntos en litigio, o mejor dicho, responder los puntos de pericia de manera clara y contundente. El principal inconveniente radica en las dificultades que tienen los partícipes no informáticos de la causa (jueces, fiscales, abogados, investigadores forenses de otras disciplinas) para *interpretar los datos técnicos* a la luz de la causa judicial, y en el contexto del resto de las pruebas documentales presentes en el litigio.

De allí que se requiera de un sistema de representación que haga posible mostrar los datos en función del objetivo que se persigue (expresado en los puntos de pericia) y vinculados semánticamente en base a la relación que los mismos mantienen entre sí.

2.2. Las Ontologías

2.2.1. Definiciones básicas y características

Las ontologías proponen un marco de referencia basado en el conocimiento, mediante un vocabulario de representación que describe cada elemento según una definición declarativa y axiomas formales que acotan la interpretación y permiten una aplicación correcta de esos términos.

En el conjunto de sistemas de representación del conocimiento, las ontologías se definen según sus características distintivas:

- Permiten consensuar el significado de los elementos y relaciones de un universo de discusión, de manera que es posible desarrollar un software para modelar los procesos de toma de decisiones por parte de los gestores del conocimiento.
- Abordan siempre un dominio acotado del conocimiento. Si bien uno de los principales problemas al definir una ontología es identificar *donde está el límite* de lo que queremos representar, esa misma acotación sustenta y valida la representatividad de la ontología. Es decir, una vez definido el dominio, las reglas de representación de

[4] ISP = Internet Service Provider, proveedor del servicio de internet.

una ontología permiten modelar acabadamente ese ámbito restringido denominado *universo de discusión*.

- Se recurre a la lógica formal para representar los componentes mediante los conceptos tradicionales de *objetos, clases, instancias, restricciones y propiedades*. Al utilizar modelos formales para la representación, es posible la aplicación de lenguajes compatibles con entornos abiertos y comprensibles para una máquina, tales como OWL, RDF, XML.
- Se recurre a la *semántica* como hilo conductor para definir los componentes que se representarán, así como las relaciones de vinculación entre ellos, permitiendo expresar el dominio en base al significado que tienen sus componentes en el marco de referencia en el que actúan.

2.2.2. Metodologías y herramientas para la construcción de ontologías

En la Ontology Summit 2007[9] se discutió sobre la gran variedad de metodologías de diseño, algunas de las cuales enfatizan una que otra propiedad en la construcción de la ontología: la ingeniería de requerimientos o la evaluación y validación de la aplicación informática resultante. Incluso se proponen metodologías sin diseño como en el caso de las folsonomías que parten del comportamiento local de miles de individuos.

No debe perderse de vista el concepto en sí de una metodología: como herramienta solo es útil en la medida en que su uso acompaña el logro del objetivo propuesto. Y en el caso particular de las ontologías semánticas, en las que la definición del dominio es una parte sustancial, se debe orientar la construcción de la ontología según sea la característica distintiva del tema.

Así, las ontologías que tratan sobre vocabularios o taxonomías deben reforzar las instancias de significación de las palabras en el dominio que están abarcando; en otros casos, como en ontologías de integración de datos, es de interés profundizar la etapa de validación de los metamodelos de datos; o, en contextos específicos como la Forensia Digital, cobra vital importancia la validación de las instancias de captura de la prueba digital y su correspondiente "cadena de custodia".

En particular, interesa el trabajo de Corcho et al. [10] en el que presentan una adaptación al dominio legal español de una taxonomía de clases sobre entidades legales propuesta por Breuker[11], aplicando la metodología METHONTOLOGY y la herramienta WebODE.

Esta metodología ha sido desarrollada por el Grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid, permite construir ontologías en el nivel de conocimientos, y tiene sus raíces en las actividades identificadas por el proceso de desarrollo de software propuesto por la IEEE y en otras metodologías de ingeniería de conocimientos. ODE y WebODE se construyeron para dar soporte tecnológico a METHONTOLOGY.

Esta metodología propone guías de actividades para la especificación, conceptualización, formalización, implementación y mantenimiento de la ontología a construir, bajo un esquema de procesos iterativos que ayudan en el ajuste del modelo a construir.

A continuación se sintetizan estas fases:

- La actividad de ***especificación*** permite determinar por qué se construye la ontología, cuál será su uso, y quiénes serán sus usuarios finales.
- La actividad de ***conceptualización*** se encarga de organizar y convertir una percepción informal del dominio en una especificación semi-formal, para lo cual utiliza un conjunto de representaciones intermedias (RRII), basadas en notaciones tabulares y gráficas, que pueden ser fácilmente comprendidas por los expertos de dominio y los desarrolladores de ontologías. El resultado de esta actividad es el modelo conceptual de la ontología.
- La actividad de ***formalización*** se encarga de la transformación de dicho modelo conceptual en un modelo formal o semicomputable.
- La actividad de ***implementación*** construye modelos computables en un lenguaje de ontologías (Ontolingua, RDF Schema, OWL, etc.). La mayor parte de las herramientas de ontologías permiten llevar a cabo esta actividad de manera automática. Por ejemplo, WebODE puede importar y exportar ontologías desde y a los siguientes lenguajes: XML, RDF(S), OIL, DAML+OIL, OWL, CARIN, FLogic, Jess y Prolog.
- La actividad de ***mantenimiento*** se encarga de la actualización y/o corrección de la ontología, en caso necesario.

METHONTOLOGY también identifica actividades de gestión (planificación, control y aseguramiento de la calidad), y de soporte (adquisición de conocimientos, integración, evaluación, documentación y gestión de la configuración).

2.3. Trabajos de aplicación de ontologías en el análisis forense

Existen varios trabajos de investigación que relacionan ambos temas: las ontologías y el análisis forense.

Se recurre a las ontologías para representar la multiplicidad de dominios expertos que se requieren en el análisis forense (desde conocimiento de redes hasta conocimientos de sistemas contables) [12], o bien para el

diseño de un sistema inteligente en red aplicado a la forensia [13]. Para el caso particular de los correos electrónicos, es de interés el trabajo de Balakumar et al. [14] sobre la definición de una ontología para la clasificación y categorización de e-mail con el objetivo de conformar un filtro para la detección de spam mediante la conformación de una *whitelists* de remitentes conocidos, así como el trabajo de clasificación de e-mails propuesto por Taghva [15] en referencia a la exigencia legal de resguardar ciertos registros de datos residentes o anexados a correos electrónicos.

3. Ontología para el Análisis Forense de Correo Electrónico

En esta sección se presenta una ontología que permite definir la semántica de los conceptos relacionados con el análisis forense de un correo electrónico a fin de formalizar los mismos, así como sus relaciones y restricciones.

En este trabajo solo se abordarán las dos primeras fases propuestas por METHONTOLOGY, llegando a una primera conceptualización que servirá de insumo para el proceso iterativo que marca esta metodología.

3.1. Fase de Especificación

La primera actividad a desarrollar consiste en determinar el objetivo de la ontología, su funcionalidad y destinatarios finales.

El objetivo de una ontología para el análisis forense de correos electrónicos es el de construir un marco de referencia formal y científico, que sustente el análisis semántico de un correo electrónico en su carácter de prueba documental, basando esa interpretación en la relación que esos datos tienen en el contexto de la causa.

La función de esta herramienta, será la de servir de apoyo a los profesionales forenses no informáticos, facilitando la interpretación de los datos obtenidos con foco en la relación que los mismos tienen con el contexto de análisis y otras pruebas documentales, mediante la identificación de los conceptos, atributos, valores y relaciones que mantienen estos datos, más allá de las características técnico-informáticas que se obtienen como resultado del análisis forense de los correos electrónicos.

Los destinatarios finales son aquellos usuarios no informáticos, o con escasa experiencia con la tecnología (abogados, jueces, fiscales, otros investigadores forenses), que necesitan interpretar los resultados del análisis forense de un correo electrónico en el marco de la causa judicial, y a quienes –usualmente- los datos técnicos "crudos" que se obtienen como resultado de un análisis forense digital les resultan complejos o difíciles de comprender.

La definición del dominio parte de un conjunto de preguntas que guían la delimitación o acotación del universo de discusión, y que ayudan a decidir qué objetos son relevantes y cuales no son representativos para el análisis.

Estas preguntas tienen como cometido los siguientes:

- establecer las funcionalidades a las que debe responder la aplicación informática resultante de la ontología;
- delimitar el dominio tal como lo conciben los usuarios finales; y
- considerar los puntos de vista de los usuarios a la hora de modelizar los conceptos de la realidad.

Los interrogantes base de esta ontología se pueden buscar en los puntos de pericia que usualmente se proponen al solicitar un análisis forense de un correo electrónico y que se han enunciado en el apartado anterior, de allí se extractan las ***preguntas de competencia*** en lenguaje natural:

1. ¿Cuáles son las partes de un correo electrónico que resultan de interés para un análisis forense?
2. ¿Cuáles son los componentes informáticos a través de los cuales se escribe y se lee un correo electrónico?
3. ¿Cuáles son los datos o componentes que permiten validar la existencia de un correo electrónico? Esta pregunta puede descomponerse en las siguientes:
   3.1. ¿Cuál es la fecha, hora y dirección IP de emisión del correo electrónico?
   3.2. ¿Cuál es la fecha, hora y dirección IP de recepción del correo electrónico?
4. Dado un correo electrónico ¿Cuáles son los datos que permiten identificar la autoría y recepción del mismo? Esta pregunta puede descomponerse en las siguientes:
   4.1. ¿Cuál es el nombre de usuario y dirección de e-mail del Autor del mismo?
   4.2. ¿Cuál es el nombre de usuario y dirección de e-mail del Receptor del mismo?
   4.3. ¿Es posible establecer el seguimiento del mensaje desde que se envía hasta que se recibe?
   4.4. ¿Cuáles son los diferentes actores/servicios que participaron de la transmisión?

3.2. Fase de Conceptualización

En esta segunda fase, corresponde trabajar sobre la definición del dominio, a partir de la percepción informal inicial que se tiene del mismo, estructurándolo en un modelo conceptual que luego sea posible formalizarlo.

Se comienza por definir el vocabulario de términos que se utilizará como base para responder a las preguntas de competencia. La Tabla 1 muestra el ***Glosario de Términos*** que integra la ontología propuesta.

Tabla 1: Glosario de Términos

Nombre	Sinónimos	Acrónimos	Descripción	Tipo
Correo Electrónico	e-mail	-	Todo mensaje de texto, voz, sonido o imagen enviado a través de una red de comunicación pública que puede almacenarse en la red o en el equipo terminal del receptor hasta que éste acceda al mismo	Concepto
Usuario	-	-	Persona que envía/recibe el correo electrónico	Concepto
Hardware	-	-	Componente informático que interviene en el emisión/transmisión/recepción del correo electrónico	Concepto
Equipo emisor/receptor	-	-	Hardware utilizado por el usuario para la escritura/lectura del correo electrónico	Concepto
Dispositivo de Transmisión	-	-	Hardware que interviene en el proceso de transmisión del correo electrónico	Concepto
Servidor de correo	-	-	Servidor que almacena el correo electrónico enviado/recibido	Concepto
Gestor de servicio	Actor	-	Componente de hardware o software que participa en la emisión/transmisión/recepción del correo	Concepto
Software	-	-	Aplicación informática que interviene en la emisión/transmisión/recepción del correo electrónico	Concepto
Cliente de correo	-	-	Aplicación informática que interviene en la escritura-emisión y lectura-recepción del correo electrónico	Concepto
Emisor	Remitente	-	Usuario que envía un correo electrónico	Concepto
Receptor	Destinatario	-	Usuario que recibe un correo electrónico	Concepto
Cuenta de correo	Dirección de mail	-	cuenta electrónica del usuario emisor o receptor	Atributo de Instancia
Fecha y hora	Date	-	Fecha y hora del correo electrónico emitido/recibido	Atributo de instancia
Dirección IP	-	IP	Dirección IP del correo electrónico a la cual se envío o desde la cual se remito	Atributo de instancia
Asunto	Subject	-	Tema de referencia del contenido del correo electrónico	Atributo de Instancia
Cuerpo	Contenido	-	Cuerpo del mensaje del correo electrónico	Atributo de Instancia
Archivo adjunto	Adjunto	-	Archivos que se remiten conjuntamente con el mensaje	Atributo de Instancia
Rol del usuario	-	-	Identificación del usuario respecto de si actúa como emisor o receptor del correo electrónico	Atributo de Instancia
Alias	-	-	Sigla o apodo con que se identifica el usuario. Usualmente este nombre está asociado al correo electrónico. Ej: pepe<juan.perez@dominio.com>	Atributo de Instancia
Nombre del usuario	-	-	Apellido y nombre real del usuario	Atributo de Instancia
Firma	-	-	Datos de identificación del usuario (nombre completo, cargo, teléfono, domicilio, etc.). Opcionalmente figuran estos datos en el cuerpo del mensaje	Atributo de Instancia
Tipo de Hardware	-	-	Referencia del hardware en cuanto a si actúa como equipo emisor/recetpor del correo electrónico o si participa como dispositivo intermedio en la transmisión.	Atributo de Instancia
Identificación única del equipo	ID equipo	-	Número de identificación única del equipo emisor/receptor (Serial Number, MAC address, etc.)	Atributo de Instancia
Dirección IP del equipo	-	-	Dirección IP del equipo emisor/receptor del correo electrónico	Atributo de Instancia
Nombre del equipo	-	-	Nombre básico del equipo emisor/receptor del correo electrónico. (PC,celular, notebook, tablet, etc.)	Atributo de Instancia
Descripción del equipo	-	-	Datos de descripción del equipo emisor/receptor del correo electrónico (tipo, marca, características volumétricas, configuración física, software de base, etc)	Atributo de Instancia
Ubicación geográfica del equipo	-	-	Ubicación geográfica del equipo emisor/receptor del correo electrónico	Atributo de Instancia
Identificación única del dispositivo	ID dispositivo	-	Número de identificación única del dispositivo que interviene en la transmisión del correo electrónico (Serial Number, MAC address, etc.)	Atributo de Instancia
Dirección IP del dispositivo	-	-	Dirección IP del dispositivo que interviene en la transmisión del correo electrónico	Atributo de Instancia
Nombre del dispositivo	-	-	Nombre básico del dispositivo que interviene en la transmisión del correo electrónico. (router, etc.)	Atributo de Instancia
Descripción del dispositivo	-	-	Datos de descripción del dispositivo que interviene en la transmisión del correo electrónico (tipo, marca, características volumétricas, configuración física, software de base, etc)	Atributo de Instancia

Nombre	Sinónimos	Acrónimos	Descripción	Tipo
Ubicación geográfica del dispositivo	-	-	Ubicación geográfica del dispositivo que interviene en la transmisión del correo electrónico	Atributo de Instancia
Identificación única del servidor	-	-	Número de identificación única del servidor de correo (Serial Number, MAC address, etc.)	Atributo de Instancia
Dirección IP del servidor	-	-	Dirección IP del servidor de correo	Atributo de Instancia
Dominio del servidor	-	-	Nombre de dominio del servidor que almacena el correo electrónico emitido/recibido	Atributo de Instancia
Descripción del servidor	-	-	Datos de descripción del servidor de correo (tipo, marca, características volumétricas, configuración física, software de base, etc)	Atributo de Instancia
Ubicación geográfica del servidor	-	-	Ubicación geográfica del servidor de correo	Atributo de Instancia
Tipo de servicio	-	-	Referencia de la actividad que desarrollan los componentes gestores de servicio, tales como: gestión de direcciones, gestión de reenvíos, gestión de distribución, etc	Atributo de Instancia
Identificación única del software	-	-	Número de identificación única del software (Serial Number, versión, etc.)	Atributo de Instancia
Nombre del software	-	-	Nombre comercial del software	Atributo de Instancia
Descripción del software	-	-	Datos de descripción del software (tipo, autoría, configuración actual, funcionalidades más notorias, etc)	Atributo de Instancia
Identificación única del cliente de correo	-	-	Número de identificación única del software cliente de correo (Serial Number, versión, etc.)	Atributo de Instancia
Nombre del cliente de correo	-	-	Nombre comercial del software cliente de correo	Atributo de Instancia
Descripción del cliente de correo	-	-	Datos de descripción del software cliente de correo (tipo, autoría, configuración actual, funcionalidades más notorias, etc)	Atributo de Instancia
Emisión (usuario, equipo, cabecera)			Relación que muestra el envío de un correo electrónico	Relación
Recepción (usuario, equipo, cabecera)	-	-	Relación que muestra la recepción de un correo electrónico	Relación
Seguimiento (hardware, software, datos)	Received	-	Relación que muestra la información de seguimiento de un correo electrónico durante la transmisión	Relación
Lista de distribución (Cuenta de correo 1, Cuenta de correo 2,…, Cuenta de correo N)	List	-	Conjunto de correos electrónicos que participan en la recepción simultánea de un correo	Relación

Se recomienda buscar vocabularios controlados existentes, es decir, recurrir a la reutilización de ontologías, para refinar y extender recursos ya establecidos para el dominio en discusión. Esto es muy útil cuando se requiere la interacción con otras aplicaciones que ya cuentan con representaciones ontológicas[5]. Un ejemplo se puede encontrar en la *LKIF core legal ontology* [16], consta de 15 módulos, cada uno de los cuales describe un conjunto de conceptos estrechamente relacionados a dominios tanto legales como de sentido común.

Tomando los componentes descriptos en el apartado 2 de este trabajo, los términos de interés deben surgir de los tres conjuntos definidos: los actores participantes en la transmisión, la arquitectura lógica y la estructura interna de un correo electrónico.

Siempre teniendo presente que las preguntas de competencia apuntan al carácter probatorio del correo electrónico, en función de las características de autentificación y existencia citadas anteriormente, se definen los conceptos, atributos de instancia y relaciones más notables

Una vez identificados los términos, se seleccionan aquellos que actúan como *conceptos*, para armar la ***Taxonomía de Conceptos***, que se muestra en la Figura 1.

METHONTOLOGY propone cuatro relaciones taxonómicas: Subclase-de, Descomposición-Disjunta, Descomposición-Exhaustiva, y Partición. A los fines del presente trabajo se han identificado las relaciones de Subclase-de que se señalan en la Figura 1. La identificación de las restantes relaciones taxonómicas se realizará en las siguientes instancias de revisión del modelo dentro del proceso iterativo señalado por la metodología.

La taxonomía desarrollada señala la relación de subclase de entre los usuarios más destacados (el emisor y el receptor del correo), que se distinguen a la vez del resto de los gestores del servicio porque son aquellos sobre quienes finalmente impacta el carácter probatorio del correo electrónico.

[5] Existen bibliotecas de ontologías disponibles en la web, tales como DAML, Ontolingua, SUMO o DOLCE.

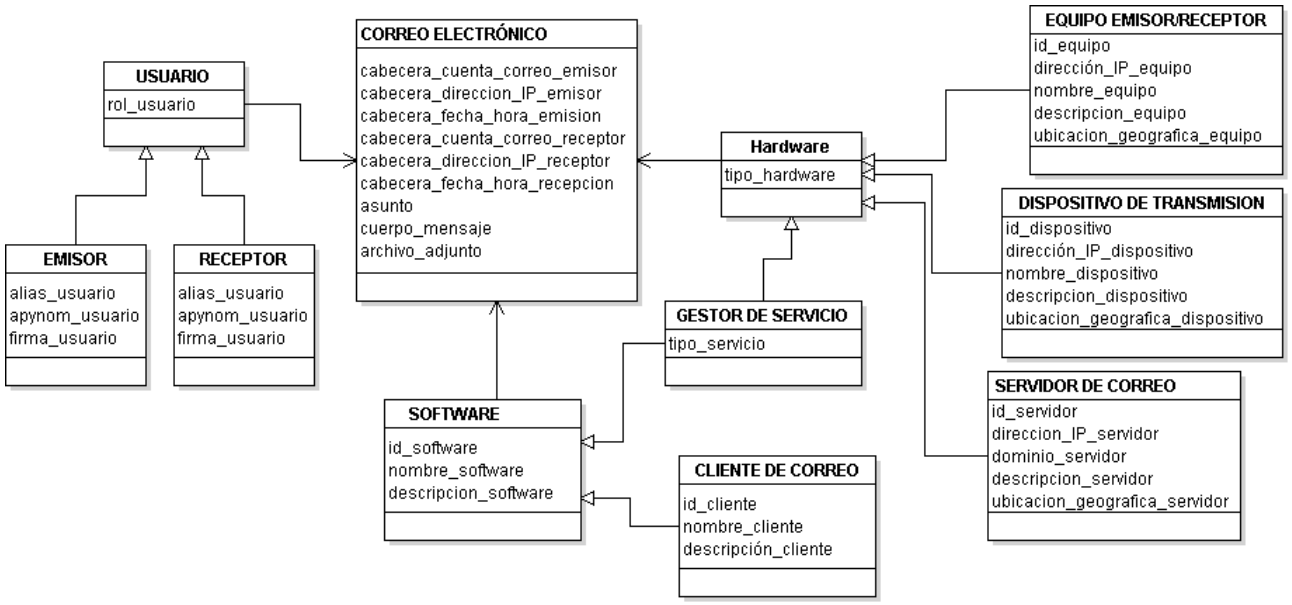


Figura 1: Taxonomía de Conceptos

La clase "software" representa los diversos programas que intervienen en la emisión / transmisión / recepción de un correo electrónico, de ellos resulta de interés la subclase "clientes de correo" ya que permiten *vincular* el correo electrónico con una persona (usuario emisor/receptor).

En cuanto a la especialización de la clase "hardware", en particular se identifica como sub-clase a los equipos utilizados para la emisión/recepción del correo, los servidores de correo y los dispositivos que participan de la transmisión del correo electrónico. Por su parte los equipos de emisión/recepción colaboran en la identificación de la vinculación usuario-correo electrónico, en los servidores de correo se encuentran almacenados los correos electrónicos, mientras que los dispositivos de transmisión muestran la *trazabilidad* del mensaje.

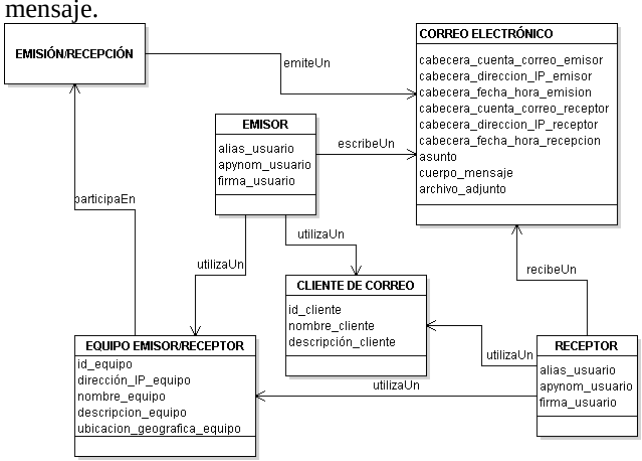


Figura 2: Relaciones ad-hoc para emisión/recepción del correo electrónico

Partiendo de la narrativa escrita en lenguaje natural – los puntos de pericia- se establecen las ***relaciones ad hoc*** existentes entre los conceptos definidos en la taxonomía. Las relaciones deben establecer con exactitud y precisión, indicando el origen y destino de cada una, evitando imprecisiones o sobreespecificación de esos puntos.

La Figura 2 muestra el proceso principal en el análisis forense, señalando los conceptos que participan y la vinculación que existe entre ellos.

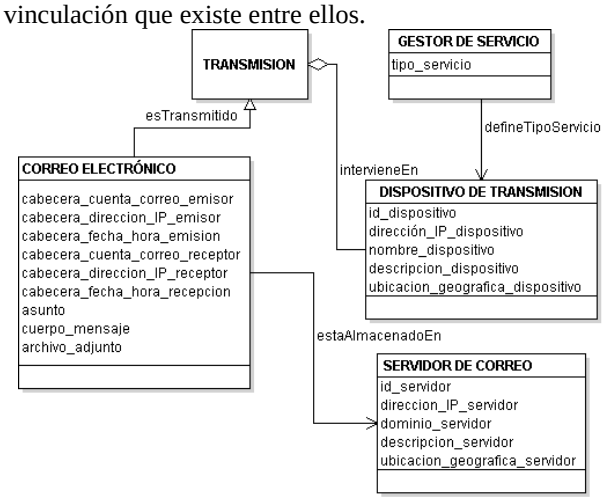


Figura 3: Relaciones ad-hoc para transmisión del correo electrónico

Una pregunta reiterada a los peritos es acerca de cómo es posible aseverar con plena certeza que el correo escrito por una persona es el que efectivamente recibió la otra (y viceversa). La relación ad-hoc que permite observar la vinculación de los componentes intermedios durante el proceso de transmisión se muestra en la Figura 3.

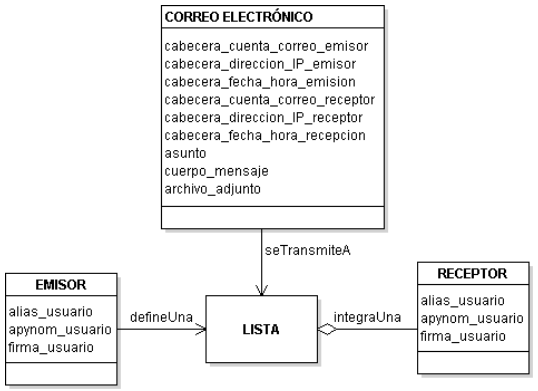


Figura 4: Relaciones ad-hoc para la Lista de Distribución del correo electrónico

Mientras que la Figura 4 muestra la vinculación entre los distintos partícipes de un correo electrónico, cuando se realiza un envío masivo, a través de una lista de distribución de correos.

Luego de identificadas las relaciones ad-hoc, METHONTOLOGY propone como siguiente paso la generación de una serie de tablas que definen el conjunto de metadatos de la ontología: el Diccionario de Conceptos, la Tabla que describe las Relaciones Binarias ad-hoc, los atributos de instancia y de clase, y las constantes.

Por razones de espacio, solo se incluye en este trabajo el Diccionario de Conceptos que se presenta en la Tabla siguiente:

Tabla 2: Diccionario de Conceptos

Concepto	Instancia	Atributos de Clase	Atributos de Instancia	Relaciones
Correo Electrónico	Prueba Digital Prueba documental	-	Cuenta de correo, Fecha y hora, Dirección IP, Asunto, Cuerpo, Archivo adjunto	emiteUn recibeUn esTransmitido seTransmteA
Usuario	-	Atributo de Clase	Rol del usuario	-
Hardware	-	Atributo de Clase	Tipo de Hardware	-
Equipo emisor/receptor	PC Celular Tablet	-	Identificación única del equipo, Dirección IP del equipo, Nombre del equipo, Descripción del equipo, Ubicación geográfica del equipo	participaEn utilizaUn
Dispositivo de Transmisión	Switch Router Servidor DNS	-	Identificación única del dispositivo, Dirección IP del dispositivo, Nombre del dispositivo, Descripción del dispositivo, Ubicación geográfica del dispositivo	intervienEn
Servidor de correo	Servidor del ISP	-	Identificación única del servidor, Dirección IP del servidor, Dominio del servidor, Descripción del servidor, Ubicación geográfica del servidor	estaAlmacenadoEn
Gestor de servicio	-	Atributo de Clase	Tipo de servicio	defineTipoActividad
Software	Protocolos de transmisión Firmware	Atributo de Clase	Identificación única del software, Nombre del software, Descripción del software	-
Cliente de correo	Outlook WebMail Mozilla Thunderbird	-	Identificación única del cliente de correo, Nombre del cliente de correo, Descripción del cliente de correo	utilizaUn
Emisor	Actor Demandado	-	Alias, Nombre del usuario, Firma	escribeUn defineUna
Receptor	Actor Demandado	-	Alias, Nombre del usuario, Firma	recibeUn integraUna

Si bien la metodología seguida propone especificar los ***axiomas formales*** con todo rigor (nombre, descripción en lenguaje natural, expresión lógica que define de manera formal el axioma usando lógica de primer orden, y los conceptos, atributos y relaciones ad hoc utilizadas en el axioma, así como las variables utilizadas), a los fines del presente trabajo solo se presentan dos de los axiomas destacados, y que en cierta forma, conforman la base del análisis forense de un correo electrónico. Éstos se presentan a continuación, y escritos en lenguaje natural:

- ***Axioma 1: sobre la autenticidad de un correo electrónico***

Un correo electrónico es auténtico cuando se identifican: los datos del remitente (nombre de usuario, cuenta de correo y dirección IP), la trazabilidad del mismo (diferentes servicios o agentes que intervienen en la transmisión) y los datos del destinatario (nombre de usuario, cuenta de correo y dirección IP).

- ***Axioma 2: sobre la existencia de un correo electrónico***

Un correo electrónico existe cuando se comprueba la presencia del archivo digital del mismo tanto en el

dispositivo emisor (ó en el servidor del ISP del emisor) como en el dispositivo receptor del correo (ó en el servidor ISP del receptor); y ambos archivos digitales son idénticos.

Hasta aquí se presentó la primera iteración de la fase de conceptualización de la ontología, resta realizar las iteraciones necesarias hasta ajustar el proceso y avanzar luego en las fases de formalización, implementación y mantenimiento propuestas por METHONTOLOGY.

4. Conclusiones

Si bien se avanzó en una primera formulación de la ontología, que seguramente será actualizada y mejor representada en cada proceso iterativo que conlleva la metodología elegida, lo escrito hasta aquí permite mostrar los beneficios de la propuesta. Esta primera formulación del modelo conceptual deberá ajustarse tanto en la profundidad del análisis de cada paso, como en la actualización de los conceptos, atributos, relaciones y axiomas, cuidando de agregar términos o descartar aquellos que fueran subsumidos en otros como resultado del trabajo iterativo.

Las ontologías generan un marco de trabajo adecuado y suficiente para el análisis e interpretación de datos en contextos de cierta complejidad, al proporcionar un sistema de representación basado en un vocabulario común y consensuado entre todos los partícipes del tema, facilitando el intercambio y enriqueciendo la base de conocimiento a partir de la interpretación que cada quién realiza sobre el mismo conjunto de datos.

Así, es posible establecer un contexto específico de comunicación y reutilizar el conocimiento existente haciendo que resulte más fácil incrementar ese conocimiento y difundirlo entre la comunidad interesada en la temática de estudio.

Estos beneficios –propios de cualquier tipo de ontología- se ven incrementados cuando facilita la comunicación y vinculación entre dos áreas muy separadas entre sí: por un lado el Derecho (que se apoya en la formalidad de la norma escrita) y por otro la Informática (que se apoya en el contexto virtual).

5. Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET, la UTN (PID 25-O156) y el Consejo de Investigaciones de la Universidad Católica de Salta. Se agradece el apoyo brindado por estas instituciones.

6. Referencias

[1] Gruber, Thomas R., "A Translation Approach to Portable Ontology Specifications", *Appeared in Knowledge Acquisition*, 5(2):199-220, 1993.

[2] DIRECTIVA 2002/58/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO SOBRE LA PRIVACIDAD Y LAS COMUNICACIONES ELECTRÓNICAS, 12 de julio de 2002, p.37

[3] Castro Bonilla, A."El uso legítimo del correo electrónico", *II Congreso Mundial de Derecho Informático*, 2002

[4] Banday, M. Tarik, "TECHNIQUES AND TOOLS FOR FORENSIC INVESTIGATION OF E-MAIL*", International Journal of Network Security & Its Applications (IJNSA),* Vol.3, No.6, November 2011.

[5] Fernández, Eduardo Enrique: "Aspectos legales del peritaje". *Revista INDICIOS*, Año 2. Vol. 2. La Rioja (Argentina) 2011. pp. 24-33.

[6] Hadjidj, R., Debbabi, M., Lounis, H., Iqbal, F., Szporer, A., & Benredjem, D. "Towards an integrated e-mail forensic analysis framework". *Digital Investigation*, *5*(3), 124-137.2009

[7] Agualimpia, C., & Hernández, R. Análisis forense en dispositivos móviles con Symbian OS. *Documento de maestría, Dept. Ingeniería electrónica, Pontifica Universidad Javeriana, http://www. criptored. upm. es/guiateoria/gt_m142e1. htm*.

[8] Ariza, A., Ruíz, J., & Cano, J. iPhone 3G: Un Nuevo Reto para la Informática Forense. *Universidad Pontifica Javeriana, Bogotá-Colombia*.

[9] Ontology Summit 2007 Communiqué, 2007, version 1.0.0 / 2007.04.24

[10] Corcho, Óscar y Fernández-López, M. y Gómez-Pérez, A. y López-Cima, A., "Construcción de ontologías legales con la metodología METHONTOLOGY y la herramienta WebODE", *Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications. Springer-Verlag*, pp. 142-157. ISBN 0302-9743, 2005

[11] Breuker, J., Elhag, A., Petkov, E., & Winkels, R." Ontologies for legal information serving and knowledge management", *Legal Knowledge and Information Systems, Jurix 2002: The Fifteenth Annual Conference* (pp. 1-10).

[12] Schatz, B., Mohay, G. M., & Clark, A. (2004). Generalising event forensics across multiple domains. *School of Computer Networks Information and Forensics Conference*, Edith Cowan University. .

[13] Saad, S., & Traore, I. (2010, August). Method ontology for intelligent network forensics analysis. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on* (pp. 7-14). IEEE.

[14] Balakumar, M., & Vaidehi, V. (2008, January). Ontology based classification and categorization of email. In *Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on* (pp. 199-202). IEEE.

[15] Taghva, K., Borsack, J., Coombs, J., Condit, A., Lumos, S., & Nartker, T. (2003, April). Ontology-based classification of email. In *Information Technology: Coding and Computing, International Conference on* (pp. 194-194). IEEE Computer Society.

[16] LKIF-Core Ontology: A core ontology of basic legal concepts. http://www.estrellaproject.org/lkif-core/ vigente al 2-ago-2014.

Representación de la evolución y refactoring de arquitecturas de software mediante la aplicación y captura de operaciones arquitectónicas

María Luciana Roldán, Silvio Gonnet, Horacio Leone
INGAR(UTN-CONICET)
Facultad Regional Santa Fe, Universidad Tecnológica Nacional
Avellaneda 3657, 3000, Santa Fe, Argentina
{lroldan, sgonnet, hleone}@santafe-conicet.gov.ar

Abstract

La evolución de arquitecturas de software ocurre por la aparición de cambios como la redefinición o modificación de requerimientos, y cambios en la infraestructura o la tecnología en la que se basa un sistema. La introducción de estos cambios de parte de los arquitectos de software debe en lo posible estar guiada por algún método y por la aplicación de patrones, para que sea realizada en forma sistematizada y evitar así la erosión en el diseño arquitectónico. Pero además, es de vital importancia que estos cambios queden documentados adecuadamente, a fin de que quienes realicen posteriores modificaciones o mantenimiento en el sistema, comprendan las soluciones aplicadas y cómo éstas impactaron en la arquitectura de software. En este trabajo se presenta un modelo para la captura y representación del conocimiento durante la evolución de arquitecturas de software, el cual se basa en la captura de las operaciones de evolución y refactoring aplicadas, junto con los nuevos elementos arquitectónicos generados, los objetivos perseguidos durante el proceso de evolución, y manteniendo trazabilidad a los elementos arquitectónicos afectados.

1. Introducción

La evolución de arquitecturas de software se produce como consecuencia de la aparición de los cambios, una constante en la ingeniería de software, y sobre los que los arquitectos y desarrolladores tienen poco o ningún control. Los cambios en las arquitecturas de software tienen un origen diverso, como la redefinición o modificación de requerimientos, el cambio en la infraestructura o la tecnología en la que se basa un sistema, o como consecuencia de errores o malas decisiones que deben ser corregidos, cambios en las estructuras organizaciones o procesos de negocios, o nuevas legislaciones. Para que la arquitectura de software no se torne de un tamaño inmanejable y se vuelva difícil de mantener, los arquitectos deberían evitar la introducción caótica de cambios y hacerlo de una manera sistematizada y ordenada. En este aspecto, el refactoring de arquitecturas de software propone mejorar su estructura sin que ello implique cambiar el comportamiento externo del sistema [9]. Para evitar el fenómeno que se conoce como "erosión arquitectónica", se han propuesto métodos y patrones de refactoring que brindan a los ingenieros de software soluciones probadas para afrontar necesidades de refactorización recurrentes [17, 16]. Un ejemplo de estos patrones de refactoring o reestructuración arquitectónica, es el que propone transformar una arquitectura monolítica en una arquitectura cliente-servidor, o en una arquitectura de tres capas que separa claramente la interfaz de usuario, de la lógica de negocio, y la capa de datos.

En este contexto, la evolución de una arquitectura de software se trata de la implementación sistemática de cambios arquitectónicos que se traducen en la adición, eliminación y modificación de los elementos que la conforman, a fin modificar el modelo arquitectónico. Estos cambios ciertamente tienen impactos en la arquitectura de software original, por lo que no deberían llevarse a cabo sin conservar ninguna información de las causas de los mismos, objetivos perseguidos, ni qué elementos se vieron afectados, o qué razones existieron que fundamentan las decisiones de los mismos. Sin embargo por diversos motivos, este "conocimiento" que explica mediante qué transformaciones se llevó a cabo el refactoring de una arquitectura de software no queda documentado o codificado, perdiéndose con el tiempo. Las causas más comunes de esta ausencia de documentación del proceso de evolución son las mismas que se aplican al diseño de la arquitectura de software [3, 8]; están relacionadas a las limitaciones que presentan las herramientas de soporte al proceso, negligencia de parte de los arquitectos o mantainers como consecuencia de falta de tiempo y sobrecarga de trabajo, o la percepción de que la documentación es un "overhead" del cual no obtienen beneficios, ya que ellos quienes producen ese conocimiento, pero muchas veces no son quienes van a emplearlo o "consumirlo". Se hace entonces necesario contar con herramientas que posibiliten la captura de ese conocimiento en forma natural, donde la carga extra en

trabajo de refactoring debido a su documentación sea mínimo. A fin de brindar una base para el desarrollo de tales herramientas, en este trabajo se propone un modelo para la captura y representación del conocimiento durante la evolución arquitecturas de software, el cual se basa en la captura de las operaciones de refactoring aplicadas, las cuales son conservadas junto con los nuevos elementos arquitectónicos generados, los objetivos perseguidos con la actividad de refactoring, y que además mantiene la trazabilidad a los elementos arquitectónicos afectados. Dicho modelo se basa en propuestas anteriores de los autores para la captura y representación de diseños ingenieriles [10, 16].

2. Trabajos relacionados

Se han reportado numerosos trabajos de investigación en relación a la evolución de arquitecturas de software [5]. Los principales temas dentro de esta línea se relacionan a las propuestas de técnicas que dan soporte al diseño de arquitecturas de software teniendo en cuenta atributos de calidad, la evaluación de la calidad de arquitecturas de software, valoraciones económicas, gestión del conocimiento arquitectónico, y técnicas de modelado. Prácticamente no existen herramientas dedicadas al soporte de la evolución y refactoring de arquitecturas de software, y mucho menos encaran la problemática de conservar y documentar la evolución de una arquitectura de software, es decir, la historia del proceso de refactorización.

Stal [17] ha contribuido en trasladar los conceptos de refactorización de código [9], como "code smells" (potenciales áreas de código que deberían ser mejoradas), a la evolución de arquitecturas de software. También ha catalogado numerosas soluciones recurrentes para el refactoring de arquitecturas de software [16] (patrones de refactoring), para lo cual ha propuesto una forma canónica de documentación de patrones.

Barnes y colaboradores [4] describen una estrategia teórica para planificar y razonar sobre evolución arquitectónica. Ésta provee elementos para modelar caminos de evolución potenciales y realizar análisis para seleccionar entre los caminos candidatos. El enfoque tiene similitudes con el propuesto en este trabajo en cuanto a que un camino de evolución planificado se modela formalmente como una secuencia de arquitecturas en la cual el primer elemento en el camino es la arquitectura del sistema actual, y el elemento final es la arquitectura deseada. Los enlaces entre los nodos sucesivos en un camino se describen en término de transformaciones arquitectónicas que son seleccionadas a partir de un conjunto predefinido de operadores de evolución. Para mantener registro de cómo evolucionan elementos individuales de la arquitectura, se proponen el uso de etiquetas, empleando una propiedad de identidad de evolución que relaciona los elementos idénticos que existen en los distintos modelos arquitectónicos. En cambio, nuestro enfoque, se basa en un modelo de versionamiento que representa a los elementos de la arquitectura en dos niveles: repositorio y versiones, manteniendo así solo el objeto genérico y todas las versiones que de él se tenga a lo largo del proceso de evolución.

Ivkovic y Kontogiannis [13] propusieron un framework para el refactoring de arquitecturas de software usando transformaciones de modelos y anotaciones semánticas relacionadas con mejoras de calidad. El contexto de refactoring es descripto utilizando perfiles UML. Además, emplean anotaciones semánticas que se enlazan a las transformaciones aplicadas con el fin de indicar sus efectos sobre objetivos y las métricas que son aplicables. Este framework está enfocado en la aplicación de las transformaciones orientadas al logro de determinados objetivos para mejorar la calidad de la arquitectura de software. Sin embargo, no provee ningún soporte para conservar la historia del proceso de refactorización de la arquitectura de software, no guarda registro de cómo ésta evoluciona, lo único que se conserva es el artefacto final alcanzado. De esta manera, se hace imposible la realización de un análisis posterior al diseño que permita descubrir de qué manera el diseño original cambió, y cuáles fueron los elementos arquitectónicos impactados.

Ahmad y colab. [1] han propuesto PatEvol, un framework conceptual para la adquisición y la aplicación de conocimiento relativo a la evolución de arquitecturas de software, con el objetivo de atacar sistemáticamente los cambios frecuentes en arquitecturas de software. Dicho framework aspira a unificar los conceptos de minería en repositorios de software (para la extracción de patrones) y evolución de software, a fin de posibilitar la adquisición y la aplicación de conocimiento de evolución arquitectónica. Propone realizar la adquisión de conocimiento a mediante el análisis post-mortem de historias de evolución registradas en *logs* de cambios. Los cambios conservados en el log son convertidos en grafos que capturan las operaciones de cambio sobre elementos arquitectónicos, y luego sobre éstos se aplican técnicas de minería para identificar patrones de evolución. Por otro lado, dicho framework pretende la aplicación de este conocimiento descubierto, haciendo posible el reuso del mismo durante la evolución de la arquitectura. Sin embargo, se advierte que los resultados reportados aún son preliminares, y se requiere mayor desarrollo de las propuestas. El enfoque de dicho trabajo tiene algunas similitudes con el que presentamos en este trabajo, pero difiere en que nuestra propuesta los cambios aplicados sobre un modelo arquitectónico se capturan directamente en un repositorio basado en tecnología orientada a objetos, lo que ahorra la tarea de trabajar

sobre logs de cambios que pueden tener otras representaciones que requieren procesamiento adicional. Además nuestra propuesta, se basa en un modelo para la captura y representación de procesos de diseño, que ha sido aplicado en otros dominios y tiene un desarrollo bastante avanzado, así como una herramienta que permite la validación de las ideas aquí presentadas.

3. Evolución de arquitecturas de software

Diversas son las causas de los cambios sobre las arquitecturas de software. A continuación enumeraremos algunas, con el objetivo de tener una visión amplia de los distintos dominios de aplicación de la propuesta, y por ende la variedad de operaciones de evolución y refactoring con que un arquitecto debería contar.

1) *Redefinición de requerimientos existentes o aparición de nuevos requerimientos*. Cualquier nuevo requerimiento implica una nueva funcionalidad o calidad que la arquitectura debe satisfacer. La redefinición de requerimientos puede significar la eliminación de componentes o responsabilidades de componentes que ya no se necesitan, y el agregado de otras nuevas.
2) *Necesidad de cambios en la infraestructura o tecnología*. Un ejemplo de este tipo de cambio se ha dado en los últimos años con la migración de sistemas tradicionales a un entorno cloud. Las organizaciones buscan minimizar la inversión en infraestructura propia y además adquirir nuevas soluciones informáticas que se adapten rápidamente a los cambios dinámicos del entorno. De esta manera, las soluciones contratadas van desde almacenamiento a servicios de computación, lo que implica un cambio significativo en la arquitectura de software existente.
3) *Erosión en el diseño.* Muchas veces los problemas de calidad en una arquitectura de software, son consecuencia de la "erosión arquitectónica" producida por los cambios que se han aplicado a la arquitectura en forma caótica o poco sistematizada, sin contar con la documentación adecuada. Este fenómeno hace necesario aplicar nuevos cambios para mejorar aspectos de calidad de la arquitectura de software sin afectar el comportamiento externo del sistema, lo cual se denomina *refactoring*. En este sentido, el cambio busca evolucionar la arquitectura de software, reorganizando sus elementos en nuevas capas o components lógicos, en vistas a incrementar la calidad de la arquitectura en su totalidad, y superar limitaciones arquitectónicas que atacan ciertos atributos de calidad como por ejemplo escalabilidad, extensibilidad, capacidad de evolución, capacidad de prueba, etc.
4) *Nueva iteración en el proceso de desarrollo del sistema*. Si bien el proceso de refactoring es aplicable a todo tipo de modelo del proceso de desarrollo, es particularmente útil en el contexto de desarrollo ágil. Por ejemplo, en un proceso Srum, una restructuración de la arquitectura puede ser integrada mediante la inclusión de actividades de refactoring en los sprints o iteraciones. Es recomendable no hacerlo con la misma frecuencia que el refactoring de código, ni tampoco realizarlo inmediatamente luego de un reléase [17].
5) *Aparición de errores,* debidos a malas decisiones de diseño arquitectónic*o.*

La documentación de los cambios que provocan la evolución de una arquitectura de software, así como de sus causas, contribuye a que en una nueva evolución, los diseñadores partan de la comprensión de las transformaciones previas que ya se han aplicado, disminuyendo también así la aparición de erosión arquitectónica. Para que esta documentación realmente se realice es imprescindible contar con herramientas que posibiliten la captura efectiva del proceso de evolución y refactoring de la arquitectura de software.

4. Enfoque empleado

El modelo propuesto adopta un enfoque operacional, en donde las decisiones de refactoring son materializadas como la ejecución de secuencias de operaciones arquitectónicas, las cuales se aplican comenzando por la versión de modelo de la arquitectura del sistema existente, a la cual llamamos *versión de modelo origen* (Fig. 1). Se parte de la premisa que de la arquitectura de software sobre la que tiene lugar la evolución, fue generada mediante el mismo enfoque [15], aunque no es imprescindible.

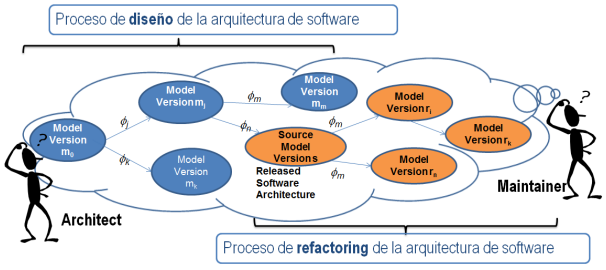


Fig. 1. Enfoque del modelo para representación del proceso de refactoring arquitectónico

Así, durante el proceso de evolución, un proyecto existente es "extendido" , lo que implica: a) reabrir un proyecto existente; b) indicar a partir de qué versión de modelo de arquitectura de software vamos a continuar el

proceso (la cual deberá ser un nodo hoja del árbol de versiones que se exploró); c) ampliar el dominio con el que se trabajó (si fuera necesario) con operaciones de evolución/refactoring apropiadas; d) avanzar en el proceso, aplicando operaciones de evolución/ refactoring, las cuales materializan las nuevas decisiones de diseño que toma el arquitecto responsable del mantenimiento ("*maintainer*"); e) realizar evaluación de la arquitectura (mediante herramientas y técnicas que se puedan aplicar); f) seleccionar la arquitectura final de la evolución.

El modelo subyacente posibilita la captura de las operaciones aplicadas junto a los resultados (productos del diseño), así como también indicar cuál es el objetivo perseguido con cada transformación aplicada sobre la arquitectura.

5. Modelo

A continuación se introduce el modelo propuesto, el cual ha sido también empleado en el proceso de diseño de arquitecturas de software [15]. Dicho modelo, considera a un proceso de diseño como una secuencia de actividades que operan sobre los productos del proceso, llamados *objetos de diseño* (*DesignObject* en Fig. 2). Los *objetos de diseño* son representaciones de los artefactos diseñados (por ejemplo, componentes y conectores) y las especificaciones a ser satisfechas (por ejemplo, requerimientos de calidad como disponibilidad o performance). Estos objetos evolucionan a medida que se introducen cambios en la arquitectura de software, dando lugar a las distintas versiones que deben ser mantenidas.

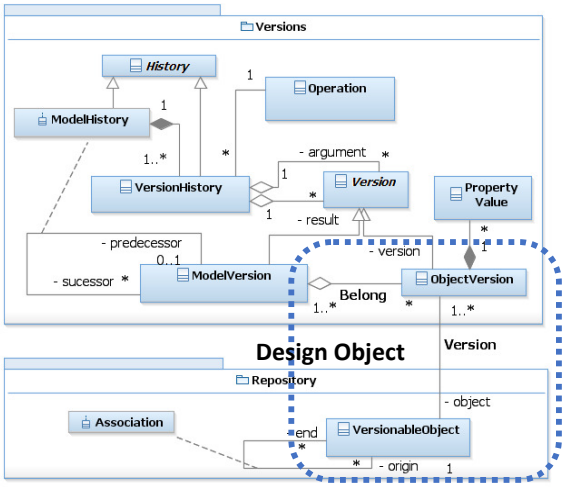


Fig. 2. Esquema de versionamiento para representar objetos de diseño y capturar su evolución

Los objetos de diseño son representados en dos niveles, el *nivel de repositorio* y el *nivel de versiones*. El *nivel de repositorio* mantiene una única entidad para cada *objeto de diseño* que ha sido creado y/o modificado durante un proyecto de diseño, o durante la evolución o refactoring de una arquitectura de software existente. Este objeto es llamado *objeto versionable* (*VersionableObject* en Fig. 2). Además, en el repositorio se representan las relaciones entre los distintos *objetos versionables* (*Association* en Fig. 2). Por otro lado, el *nivel de versiones* mantiene las diferentes versiones de cada *objeto de diseño*. Éstas son llamadas *versiones de objeto* (*ObjectVersion* en Fig. 2). La relación entre un *objeto versionable* y sus *versiones de objeto* es representada a través de la asociación *Version*. Por lo tanto, para un *objeto de diseño* dado se mantiene una única instancia en el *repositorio* y todas sus versiones se alojan en el *nivel de versiones*.

En un momento determinado durante la evolución de una arquitectura de software, los estados asumidos por el conjunto de *objetos de diseño* relevantes proveen una fotografía del estado del proceso, el cual se denomina *versión de modelo* (*ModelVersion* en Fig. 2). Una nueva versión de modelo es generada aplicando una *secuencia de operaciones* (ϕ) sobre *una versión de modelo predecesora*. Por consiguiente, el esquema de representación de versiones de modelo tiene una estructura de árbol, dónde cada *versión de modelo* es un nodo y la raíz es una *versión de modelo inicial*. Debido a que la evolución de modelo es planteada como una historia formada de situaciones discretas, en trabajos previos [14] se adoptó el cálculo de situaciones para modelar formalmente el proceso de generación de versiones. Allí se define el predicado *belong(v,m)* usando el formato de axiomas de estado sucesor, el cual permite conocer las versiones de objeto (v) que pertenecen a una *versión de modelo* (m). Esto permite reconstruir la versión de modelo m_{i+1} aplicando todas las secuencias de operaciones desde la versión de modelo inicial m_0. Las operaciones primitivas que han sido propuestas en tal esquema de versionado para representar las transformaciones de las *versiones de modelo* son *add, delete* y *modify*. La operación *add(v)* incorpora una *versión de objeto v* que no existía en una *versión de modelo* previa. En cambio, la operación *delete(v)* elimina, en la *versión de modelo* actual, una *versión de objeto v* que existía en la *versión de modelo* previa. Además, si un *objeto de diseño* tiene una versión v_p, la operación *modify*(v_p,v_s) crea una nueva versión, v_s.Cada operación aplicada a una *versión de modelo* es capturada a través de la clase *VersionHistory* (ver Fig. 2), la cual permite realizar el seguimiento de la evolución del modelo manteniendo referencias entre las *versiones de objeto* sobre las que se aplica una *operación* (*Operation* en Fig. 2) y aquellas que surgen como resultado de su ejecución. Además, todas aquellas instancias de *VersionHistory* que consisten en la captura de una operación individual son agregadas en una instancia de

ModelHistory para representar la *secuencia de operaciones* que causa la evolución.

En la Fig. 3 se ilustra con un ejemplo simple de qué manera el esquema de versionamiento posibilita la captura de la evolución de una arquitectura de dos capas a una arquitectura de tres capas. En ella se observa que se parte de la *versión de modelo i* que contiene a la arquitectura de software de partida de la evolución. En esta versión de modelo se cuenta con objetos que representan al sistema y los dos componentes que representan a las capas lógicas del mismo. A continuación se decide avanzar en el refactoring de esa arquitectura, para lo cual se ejecuta la secuencia de operaciones θ_i que da lugar a una nueva *versión de modelo i+1*, en la cual aparece una nueva versión de sistema, *Sistema-v2*. Ese objeto sistema mantiene las asociaciones con otros objetos (componentes) que este objeto ya poseía en la versión de modelo predecesora. Una nueva decisión del diseñador/maintainer de la arquitectura es adicionar una nueva capa en la arquitectura. Esa decisión se materializa en la aplicación de la secuencia de operaciones θ_{i+1}, la cual da lugar a la generación de la *versión de modelo i+2* que contiene al nuevo componente lógico que representa a la capa adicionada.

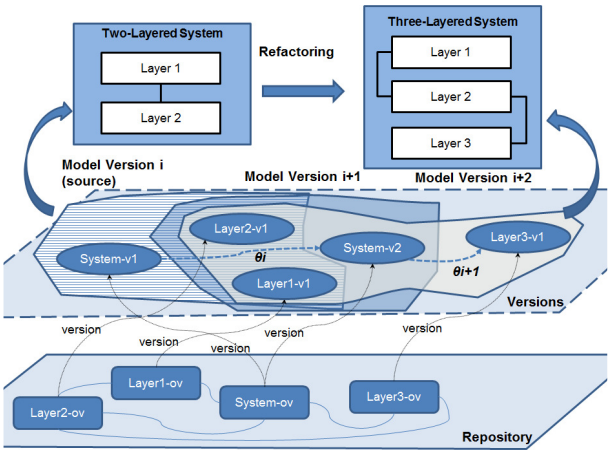


Fig. 3. Representación de objetos de diseño y captura de la evolución de una arquitectura de software

El modelo orientado a objetos propuesto provee el paquete *Domain* (Fig. 4, lado izquierdo), el cual permite la definición de diferentes dominios de arquitecturas de software. Un dominio abarca un conjunto de conceptos específicos al contexto del sistema existente que se desea evolucionar, los cuales constituyen los diferentes tipos de objetos de diseño de los cuales se desea conservar su evolución (*DesignObjectType*), y los posibles tipos de relaciones que pueden existir entre ellos (*DomainRelationship*). Los tipos de objetos de diseño pueden ser abstractos (*AbstractDesignObjectType*) o concretos (*ConcreteDesignObjectType*), con el objetivo de establecer relaciones de generalización/especialización entre conceptos del dominio. Además, para un tipo de objeto de diseño se definen las propiedades que lo caracterizan (*Property*) y las operaciones que son aplicables sobre objetos de tal tipo (*Operation*). Un dominio específico surge como instanciación de las clases mencionadas. En la Fig. 4 (derecha) se presenta un posible modelo de dominio de arquitecturas de software, el cual incluye conceptos que un arquitecto podría utilizar. Un primer conjunto de tipos de objetos de diseño que constituye el dominio propuesto en este trabajo surge del lenguaje de descripción de arquitecturas empleado, por ejemplo ACME o UML2. En este dominio particular se incluyen conceptos de dos vistas o *viewpoints* [12]: la vista de componentes y conectores (C&C), y la vista de despliegue. La vista C&C principalmente se enfoca en la división de responsabilidades y comportamiento de los componentes de la arquitectura, y en cómo estos se relacionan. Por ejemplo, un arquitecto experto puede definir en el dominio los siguientes tipos de objetos de diseño *Component*, *Connector*, *Port*, *Role* and *Attachment* (Fig. 4, derecha), los cuales son tipos de objetos de diseño derivados del lenguaje ACME [10]. Adicionalmente, *Characteristic* and *Responsibility* design object types se pueden incluir en este dominio para representar propiedades y comportamiento relevantes de los components y conectores. Por cuestiones prácticas en este dominio, se incluye el tipo de objeto de diseño *RHasResponsibility* que permite enlazar *Component* y *Responsibility* y representar el traspaso o delegación de responsabilidades de un componente a otro. *RHasResponsibility* tiene propiedades como *assignedBy* para representar quién fue el actor responsable de la acción.

Consideramos ahora tipos de objetos de diseño relativos a la vista de *Despliegue* (*Allocation*) [6], la cual es útil para describir cómo los elementos de una arquitectura de software interactúan con otros elementos que no son software, o cómo están lógicamente contenidos, en el ambiente en donde el sistema va a ejecutarse. En este caso se definen los tipos de objetos de diseño *ProcessingNode*, *Network, CommunicationChannel* and *NetworkDevice* (Fig. 4).
A esta vista también pertenecen otros tipos de objetos de diseño ya incluidos en este dominio, como *Component* y *Connector*. *RAllocated-to* permite expresar que de qué manera un elemento de software reside sobre otro elemento del entorno, el cual puede ser la unidad física sobre la que reside. Cabe aclarar, que *RAllocated-to* es una relación de dominio *reificada*, es decir es un tipo de relación de dominio que se ha transformado en un tipo de objeto de diseño a fin de aumentar su semántica. Un componente puede ser asignado a un mismo componente

físico o virtual durante todo su ciclo de vida (*RFixedAllocated-to*), o puede ir cambiando el componente sobre el que se aloja dinámicamente durante la vida del sistema. En este caso dependiendo de la naturaleza del tipo de alojamiento dinámico, se proponen en el dominio distintos tipos de objetos de diseño: *RMigrates-to*, *RCopy-Migrates-to*, and *RExecution-Migrates-to*. La semántica de estas relaciones reificadas es la brindada por Clements y colab. [6]. Otra relación reificada incluida en el dominio es *RChannelledBy*, que posibilita al diseñador dejar explícito en un diseño arquitectónico cuál es el conector físico a través del cual una comunicación lógica tiene lugar. Dependiendo de la necesidad de expresividad en el dominio, los tipos de objetos de diseño pueden ser especializados en elementos más concretos. Por ejemplo un *componente* puede ser especializado en *base de datos*, *componente de sincronización de datos*, *componente gateway*, *virtual* machine, etc. (no se presentan en la Fig. 4 por limitaciones de espacio).

Si bien no se detalla en este trabajo por cuestiones de espacio, en un dominio es posible incluir tipos de objetos de diseño que sean relativos a conocimiento de contexto y de razonamiento ("*rationale*") de la arquitectura, que es deseable documentar o capturar durante el proceso de evolución. Posibles tipos de objeto de diseño a incorporar son *FunctionalRequirement* y *QualityRequirement*, *QualityScenarios*, y *Constraints*. Modelos de dominio conceptuales que incluyen estos tipos de objetos de diseño han sido propuestos en trabajos anteriores [14, 15].

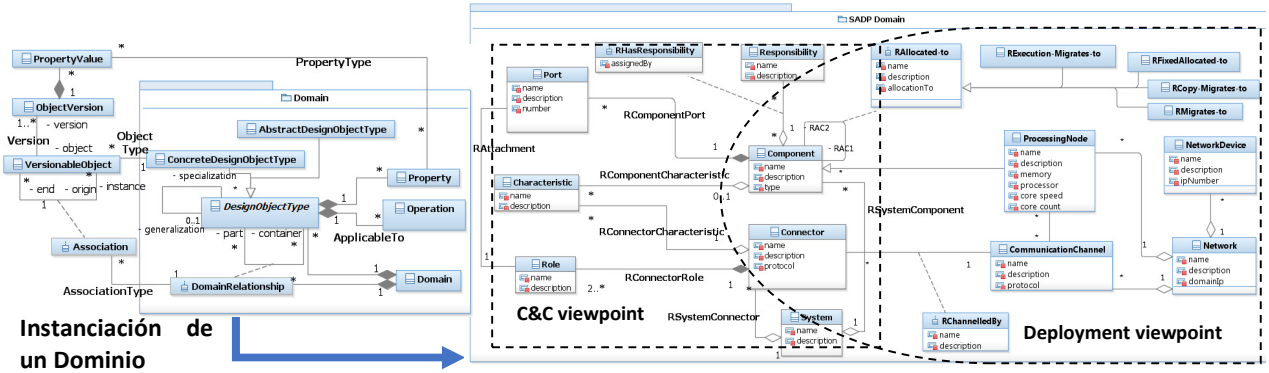


Fig. 4. Definición de un dominio con elementos de las vistas de C&C y Despliegue

5.1. El modelo de operaciones

Las operaciones primitivas *add*, *delete* y *modify* básicas para la transformación de versiones de modelo, no son suficientes para capturar la semántica de los patrones de evolución y refactoring, por lo que deben ser extendidas con operaciones de mayor complejidad, que permitan además manipular los tipos de objetos de diseño específicos del dominio. Se propone a continuación un modelo de operaciones que sienta la base conceptual para el desarrollo de herramientas computacionales flexibles que permitan especificar y ejecutar operaciones de evolución. En la Fig. 5 se observa que una operación (*Operation*) es definida como un macro comando (*MacroCommand*), subclase de *Command*, que simplemente ejecuta una secuencia de comandos. Para un macro comando debe definirse su cuerpo (expresado mediante la relación de composición entre *MacroCommand y Command*). Los comandos del cuerpo pueden ser primitivas (tales como *add*, *delete* o *modify*), o funciones auxiliares. Estas últimas son comandos predefinidos en el modelo para iteración, asignación de variables, asignan asociaciones entre objetos del repositorio, etc. Cada comando tiene uno o más argumentos tipados (*Argument* en Fig. 5), los cuales pueden ser considerados cómo variables. Una variable (*Variable* en Fig. 5) tiene un tipo y puede ser declarada y usada en el cuerpo de una operación. La clase *DataType* generaliza los tipos disponibles: *PrimitiveDataType*, *Collection*, *DesignObjectType* y *DomainRelationship*.

La definición de operaciones en un dominio se realiza como instanciación de las clases del modelo de operaciones. Para su especificación en forma textual, se ha definido una gramática libre de contexto que define cada parte de una operación: su nombre, argumentos y tipos, los comandos que conforman su cuerpo y los resultados.

6. Operaciones

Como se ha introducido previamente, cuando un *maintainer* va a iniciar un proceso de evolución arquitectónica, parte de una arquitectura inicial. Esta arquitectura ha sido obtenida mediante un proceso de diseño en el que el diseñador trabajó empleado los elementos definidos en un dominio dado. Para proceder con la evolución de la arquitectura, puede ser necesario extender ese dominio, con nuevos tipos de objetos de

diseño específicos y operaciones de evolución y refactoring que no habían sido consideradas previamente. A continuación se mencionarán y detallarán algunas operaciones que podrían incluirse en el dominio que se ha propuesto en la sección anterior.

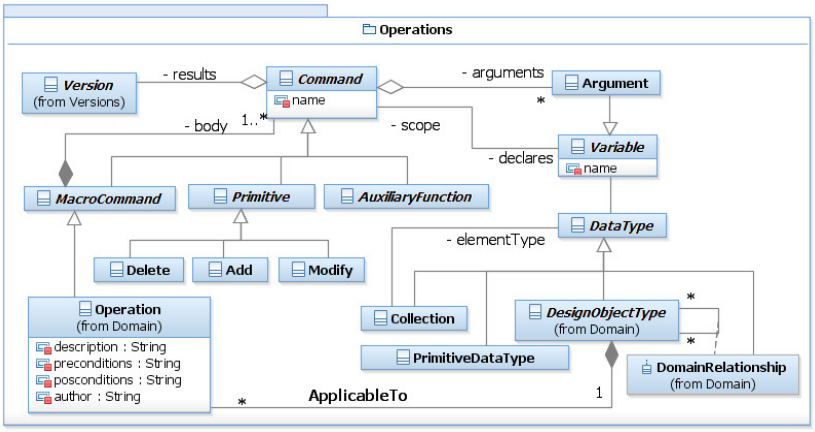


Fig. 5. Modelo para la definición de operaciones de evolución y refactoring

6.1. Operaciones de inicio del proceso de evolución

Las operaciones de inicialización se encargan de cuestiones relativas a la gestión del proceso de evolución arquitectónica que se va a llevar a cabo, como por ejemplo indicar cuál será la arquitectura de "partida", o fijar el contexto del proceso definiendo los requerimientos que se desea alcanzar en la próxima evolución arquitectónica. A continuación se detallas algunas operaciones, y se presenta en la Fig. 6 su correspondiente especificación funcional.

- *EvolveSourceSystem*. Se aplica sobre la versión de sistema que representa la arquitectura de software actual, a partir de la cual tendrá lugar la evolución (argumento *s*). Además, se debe proveer el argumento *evolutionDescription* que es la descripción de la causa de la evolución, el *nombre* para la nueva versión de sistema. Como resultado de esta operación se obtiene una nueva versión de modelo, en donde se tiene una nueva versión de *sistema* que mantiene todas las relaciones existentes con otros elementos de la arquitectura. Sobre esta versión de sistema además se fijan valores para propiedades *fecha* y *autor* en forma automática.
- *AddNewQualityRequirement*. Es similar a la operación *addQualityRequirement* [15]. Se lo propone como nueva operación, a fin de agregar la semántica de que se trata de un requerimiento que aparece durante la evolución de la arquitectura.
- *AddNewFunctionalRequirement*. Idem anterior para requerimientos funcionales.
- *DeleteOldQualityRequirement*. Elimina un requerimiento de calidad que ya no es necesario satisfacer.

```
evolveSourceSystem(s: System, newName: String,
                   evolutionDescription: String)
  modify(s, [newName, evolutionDescription]).
end
```

```
addNewQualityRequirement(s: System, rn: String,
                         [description: String, priority: Integer])
  addQualityRequirement(s, rn, [description, priority])
end
```

```
addQualityRequirement(s: System, rn: String,
                      [description: String, priority: Integer])
 r := add(rn, QualityRequirement,
          [description, priority, 'NoSatisfied'])
 rs := add(null, RReqSist, ['NoEvaluated'])
 addAssociation(rs, s, RRSR)
 addAssociation (rs, r, RRSEA)
end
```

```
deleteOldQualityRequirement (qr: [QualityRequirement])
  lsce := get(QualityScenario, qr)
  for each s en lsce
    deleteQualityScenario(s)
  end for
  delete(qr)
end
```

Fig. 6. Ejemplos de operaciones de inicio del proceso de evolución

6.2. Operaciones de evolución/refactoring

Contando con el modelo de operaciones, los arquitectos y diseñadores expertos, pueden codificar conocimiento acerca de patrones de refactoring, o cualquier operación de evolución arquitectónica. Este conocimiento puede provenir de sus propias experiencias o provenir de diversas fuentes, como literatura existente

sobre soluciones a problemas recurrentes [2, 4, 7, 13, 17], o bien de prácticas bien conocidas en la industria [11]. En los últimos años, los patrones de cambio arquitectónicos o de refactoring y los estilos de evolución han probado ser exitosos en promover reuso de expertise para atacar la problemática de la evolución de arquitecturas de software.

Presentamos a continuación algunos ejemplos de operaciones de evolución y refactoring arquitectónico. La cantidad y el tipo de operaciones refactoring dependerá del dominio definido, y por consiguiente de los tipos de objetos de diseño incluidos. Aquí se presentan algunos ejemplos representativos, introduciendo una breve explicación de la operación, y su especificación funcional. Dado que algunas operaciones son extraídas de catálogos de patrones existentes, mantendremos el nombre original de la operación en inglés [11, 13, 16]:

- *Rename Entities*. El objetivo es introducir nombres intuitivos para las entidades, de manera de que sea fácil entender el rol de cada entidad en la arquitectura, y ganar expresividad.
- *Add New Component/ Add New Connector*. El nombre es autoexplicativo. Se define una operación con este nombre especial para indicar que se trata de un componente agregado durante el proceso de evolución o refactoring.
- *Add Cloud Provider Network.* Agrega un subtipo de *Network*, que se representa el conjunto de servicios ya sea de plataforma, infraestructura o software contratado con un proveedor.
- *Add Virtual Machine*. Agrega un tipo de componente especial que representa a un servidor virtual.
- *Split Subsystems*. Como el nombre lo indica, se busca dividir un componente o subsistema en componentes separados, como por ejemplo una base de datos que se encuentra acoplada e internamente contenida por el componente padre. Es necesario aplicar esta operación de refactoring cuando las partes de un sistema tienen un bajo acoplamiento, cuando se necesita mejorar la modularización de un sistema, para comprenderlo mejor, o cuando un componente posee demasiadas responsabilidades. Patrones más específicos son *Client-Server* y *Three Layers.*
- *Consolidate Duplicated Components/Connectors.* Tiene como objetivo unificar varios componentes o conectores que estén duplicados en uno solo, con el objetivo de disminuir la complejidad y tamaño del sistema.
- *Break Dependency Cycles.* Un subsistema A podría depender tanto directa como indirectamente de un subistema B (es decir, A podría depender de un un subistema C, el cual a su vez dependa de B). Los ciclos de dependencias conducen a sistemas menos mantenibles, menos fáciles de cambiar, menos reusables, probables y fáciles de comprender.
- *Insert Transparency Layer / Add Wrapper.* La capa de trasparencia es una capa de indirección que puede estar integrada al subsistema, como una máquina virtual o ser una entidad independiente como un Adapter o Facade que reenvía las invocaciones al subsistema. En ambos casos los componentes que acceden no saben nada acerca de los componentes que brindan los servicios. Una forma de implementar la capa de transparencia es a través de wrappers que reenvían las invocaciones a los componentes ocultos del subsistema y realizan algunas acciones antes y después de las invocaciones.
- *Add New Responsibility.* Simplemente consiste en agregar nuevas responsabilidades sobre componentes ya existentes.
- *Delegate Responsibility.* Consiste en mover una responsabilidad que poseía un componente a otro componente.
- *Extract Responsibility / Extract Interface*. Similar a *Delegate Responsibility,* pero en lugar de moverse la responsabilidad, ésta se "exporta". Para ello se emplea una interfaz abstracta como un *bridge* o un *decorator* para separar interfaz de implementación.
- *Load balancing.* Incorpora un componente que tiene la responsabilidad de distribuir la carga de trabajo de otros componentes.
- *Remove Duplicates*. Esta operación apunta a mejorar un diseño en donde existen componentes con las mismas responsabilidades.
- *Move Entities.* Con esta operación se busca mover componentes de un subsistema a otro. Aplicable cuando un componente tiene alto acoplamiento con otro que no se encuentra contenido en el mismo subsistema o componente contenedor, es decir, se ajusta mejor semánticamente a otro subsistema. Implica eliminar conectores existentes, crear otros nuevos entre los puertos que estén involucrados en la nueva configuración.
- *ReallocateComponent.* Similar a la anterior en cuanto a la configuración de elementos arquitectónicos que puede obtenerse, pero se aplica específicamente para describir el re-despliegue de un componente lógico en un nuevo componente (que puede ser lógico o físico). Como argumento opcional se puede proveer una descripción de qué interfaces o responsabilidades es necesario adecuar. Por esa razón, se crea una nueva versión del componente que ha sido reubicado.
- *Map Port*, se emplea cuando por ejemplo se hace el despliegue de un componente lógico sobre uno físico, y hay que remapear los puertos.

- *Replicate Database*. Esta operación se aplica cuando es necesario replicar una base de datos, o una partición de ella, por cuestiones de rendimiento, seguridad o disponibilidad/redundancia. Los argumentos de esta operación son el componente base de datos a replicar, el nombre de la base de datos nueva, el componentes adonde se alojará la base de datos replicada y los scripts para generar la partición o esquema de base de datos a replicar.
- *Synchronize Database*. Esta operación se aplica cuando existen (particiones de) bases de datos replicadas que requieren mantenerse sincronizadas con la base de dato maestra. Consiste en la creación de un componente o servicio dedicado a esta tarea. Sus argumentos son el nombre del nuevo componente, sus responsabilidades, y las bases de datos que sincroniza. Puede ser necesario indicar un componente lógico gateway sobre el cual se canalizan las solicitudes de actualización, ya que las bases de datos posiblemente se encuentran desplegadas sobre redes diferentes.

En la Fig. 7 se presentan las especificaciones funcionales para algunas de las operaciones mencionadas. Además se incluyen operaciones arquitectónicas que ya han sido definidas previamente en el dominio, y que se usan en la definición de otras operaciones (*addComponent* y *addConnector*) [15].

Cabe aclarar que aquí presentamos un posible conjunto de operaciones de evolución y refactoring, pero en un dominio pueden definirse todas aquellas que sean necesarias.

Por otro lado, el modelo de operaciones brinda la posibilidad de definir *operaciones orientadas a objetivo (oo)*. Estas operaciones son similares a las presentadas, pero incorporan un argumento adicional, el cual puede ser el requerimiento de calidad que se espera satisfacer con la ejecución de la operación. Al ejecutarse la operación además de capturarse la operación junto con sus resultados, se generan asociaciones especiales entre los objetos agregados/modificados y el objetivo, de manera de representar la semántica de la "probable satisfacción del objetivo perseguido". Por ejemplo, si la operación *replicateDatabase* persigue el objetivo de *performance* se podría aplicar en su lugar una operación alternativa *replicateDatabase-oo(arg1, arg2, ..., goal)*, en donde el valor para el argumento *goal* sea el requerimiento de calidad *PerformanceV1* que se ha elicitado para la arquitectura nueva.

7. Escenario de aplicación

Para ilustrar cómo se emplea el modelo, consideraremos un escenario, que ilustra un proceso de migración de una arquitectura de software de un sistema tradicional a una cloud híbrida.

```
addNewComponent(s: System, cn: String, lResps: Collection[String], lPorts: Collection[String],
lprops: Collection[PrimitiveDataType])
   addComponent(s, cn, lResps, lPorts)
end
```

```
addNewConnector(s: System, nc: String, lRoles: Collection[String],
lPorts: Collection[Port], lprops: Collection[PrimitiveDataType])
  addConnector(s, cn, lRoles, lPorts, lprops)
end
```

```
addComponent(s: System, nc: String, lResps: Collection[String],
lPorts: Collection[String], lprops: Collection[PrimitiveDataType])
  c:= add(nc, Component, lprops)
  for each nr in lResps
    addResponsibility(c, nr, [ ])
  end for
  for each np in lPorts
    addPort(c, np, [ ])
end for
 addAssociation(s, c, RSystemComponent)
end
```

```
addConnector(s: System, nc: String, lRoles: Collection[String],
lPorts: Collection[Port], lprops: Collection[PrimitiveDataType])
  c :=add(nc, Connector, lprops)
  i := 0
  for each nr in lRoles
    r:= addRole(c, nr, [])
    addAssociation(lPorts(i), r, RComponentPort)
    i++
  end for
  addAssociation(s, c, RSystemConnector)
end
```

```
replicateDataBase(s: System, c:Database, rn:String, h: Component, schemaScripts: String)
  dbr := addDatabase(rn, schemaScripts)
  setFixedAllocation(dbr, h)
end
```

```
synchronizeDataBases(s: System, nsync: String, lresps: Collection[Responsibility] , lports: Collection[Ports], mdb: Database, rdb: Database, gatewayComponent: Component)
  sync := addNewComponent(s, nsync, lresps, lports)
  portsSync := get(Port, sync)
  portsMdb := get(Port, mdb)
  portsGwy := get(Port, gatewayComponent)
  np11 := select(portsSync)  // interacts with the user
  np21 := select(portsMdb)  // interacts with the user
  addNewConnector(s, 'ConnSyncMdb', [R1, R2], [np11, np21])
  np12 := select(portsSync)  // interacts with the user
  np31 := select(portsGwy)  // interacts with the user
  addNewConnector(s, 'ConnSyncGwy', [R3, R4], [np12, np31])
  np32 := select(portsGwy)
  np41 := select(get(Port, rdb))
  addNewConnector(s, 'ConnGwyRdb', [R5, R6], [np32, np41])
end
```

```
addWrapper(n: Network, wn: String, lreps: [Responsibility], lports: [Ports], c1: Component, c1: Component)
  w := addNewComponent(n, wn, lresps, lports, [])
  pw1 := select(get(Port, w))
  pc1 := select(get(Port, c1))
  addNewConnector(n, 'Conn1', ['Role11', 'Role12'], [pw1, pc1])
  pw2 := select(get(Port, w))
  pc2 := select(get(Port, c2))
  addNewConnector(n, 'Conn2', ['Role21', 'Role22'], [pw2, pc2])
end
```

Fig. 7. Ejemplos de operaciones de evolución/refactoring.

Se mostrará cómo, partiendo de la versión de la arquitectura de software actual y a través de la aplicación de operaciones de evolución, se puede ir avanzando en el proceso de evolución de la arquitectura, mientras se capturan las decisiones tomadas y los productos generados/modificados durante el mismo.

Sports es una empresa dedicada a la venta mayorista de ropa deportiva, en los últimos años abrió su propio web store para venta minorista. Este nuevo modelo de negocio ha sido exitoso y sus proyecciones de ventas online van en ascenso, por lo que han surgidos nuevos requerimientos de escalabilidad y disponibilidad. Por tal motivo y con el objetivo de abaratar costos de infraestructura han decidido migrar parte de su sistema de ventas minoristas a la cloud. Por otro lado, la empresa además posee un sistema legacy para la gestión de pedidos y ventas mayoristas. Actualmente este sistema es operado por los empleados administrativos de la empresa. Particularmente, los pedidos realizados por los preventistas son comunicados via fax o telefónica, y cargados al sistema por los operadores. Se desea liberar a los operadores de esta carga de trabajo, por lo que se ha pensado incorporar un nivel de acceso especial para preventistas en el sitio web, para que ellos carguen sus propios pedidos. Para que sea posible interactuar con el sistema de carga de pedidos, se contará con un componente *wrapper* en la nube, que brindará los servicios de acceso al componente de Pedidos Mayoristas. Además, por cuestiones de seguridad, por el momento las bases de datos se mantenendrán bajo control de la empresa, por lo que se conservará el servidor de datos como parte de la infraestructura. Sin embargo, para lograr un mayor rendimiento cuando los clientes minoristas realizan compras se pretende contar en la cloud con un esquema de bases de datos replicado, que abarque solamente el catálogo y stock actual de productos de venta minorista. De esta manera, los datos sobre descripción de producto y la cantidad de unidades disponibles podrán obtenerse sin tener que recuperar dicha información de la base de datos que se encuentra en la intranet de la empresa. Por ese motivo, se contratarán servicio de hosting de bases de datos, que se actualizará con cierta frecuencia para reflejar el catalogo y stock actual.

7.1. Sequencias de operaciones aplicadas

Se parte de una arquitectura de software inicial, dada por el sitio de e-commerce actual para ventas minoristas y los componentes legacy de la empresa para gestión de ventas mayorista (Fig. 8). Esta versión de modelo es la arquitectura de software "source" del proceso de evolución de la arquitectura. El maintainer que inicia el proceso ejecuta la siguiente secuencia de operaciones:

θ_1={*selectSourceSystem*(*SystemSport*$_{v1}$, '*SystemSportEv1*', '*Hibride cloud migration…*')}.

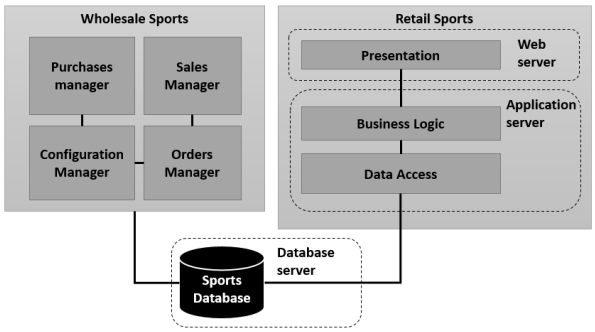


Fig. 8. Arquitectura de partida del proceso de evolución

A continuación se identifican los requerimientos de calidad y funcionales que se desean alcanzar en el proceso de evolución a llevar a cabo:

θ_2={*addNewQualityRequirement(SystemSportEv1, 'Scalability')*, *addNewQualityRequirement(SystemSportEv1, 'Performance')*, *addNewQualityRequirement(SystemSportEv1, 'Availability')*}) .

A partir de estos requerimientos se toman las primeras decisiones de evolución y refactoring de la arquitectura de software tendientes a alcanzar dichos requerimientos. Se comienza por decidir contratar servicios a un proveedor de servicios de cloud computing (*Cloud Services Provider*, *CSP*), en la modalidad IaaS, y de esta manera migrar los componentes que constituyen la aplicación del web store minorista. Para ello *CSP* provee una máquina virtual para servidor web sobre la cual se desplegará la capa de presentación del web store. Será necesario adecuar algunas interfaces a este nuevo entorno. Para la capa de lógica de negocios, y la capa de acceso a datos se procede de la misma manera, siendo reubicadas éstas en un componente que es una máquina virtual del servidor de aplicación. También se crea un servidor virtual para la base de datos que se encontrará en la cloud, para evitar la disminución en el rendimiento percibido por los clientes del webstore.

θ_3={*addCloudProviderNetwork(SystemSport*$_{v1}$*, CSPNetwork')*}

θ_4={*addVirtualMachine(CSPNetwork*$_{v1}$*, 'VM-WebServer'), addVirtualMachine(CSPNetwork*$_{v1}$*, 'VM-AppServer') , addVirtualMachine(CSPNetwork*$_{v1}$*, 'VM-DBServer')*}

θ_5={*realocateComponent(Presentation*$_{v1}$*, VM-WebServer*$_{v1}$*), realocateComponent(BusinessLogic*$_{v1}$*, VM-AppServer*$_{v1}$*), realocateComponent(DataAccess*$_{v1}$*, VM-AppServer*$_{v1}$*)*}

Para mantener conexión con la red privada virtual de la empresa con el *CSP*, se contrata un servicio VPN Gateway que permite la conexión segura a la infraestructura on-premises.

θ_6={*addNewComponent(CSPNetwork$_{v1}$, 'VirtualNetworkGateway', [Resp-ConnectVPN],[PortVPNG1, PortVNPG2, PortVPNG3, PortVPNG4, PortVNG5])* }

Se procede a documentar todas las nuevas responsabilidades arquitectónicas que se deben incorporar al componente de presentación del sistema web, ya que se incorpora el acceso para preventistas para la carga de pedidos mayoristas.

θ_7={*addNewResponsibility(Presentation$_{v2}$,), addNewResponsibility(Presentation$_{v2}$, 'Resp-Autentication'), 'Resp-OrderFormPresentation'), addNewResponsibility(Presentation$_{v2}$, 'Resp-OrderFormDefaultValues'), addNewResponsibility(Presentation$_{v2}$, 'Resp-OrderFormValidation'), addNewResponsibility(Presentation$_{v2}$, 'Resp-Submit')*}

Se requiere de un componente *wrapper* que estará a cargo de encapsular los pedidos y proveer el acceso al módulo de Pedidos del sistema legacy de la empresa. Tener en cuenta que los pedidos mayoristas se continúan cargando en la base de datos maestra de *Sport*. Las solicitudes enviadas por el wrapper se encaminarán a través del VPN gateway.

θ_8={*addWrapper(CSPNetwork$_{v1}$, 'Orders-Wrapper, [Resp-ReceiveRequest, Resp-ForwardRequest],[PortOW1, PortOW2], Presentation$_{v2}$, VPNGateway$_{v1}$)*}

Esta operación incluye el establecimiento de conectores entre los componentes entre los que se inserta el wrapper como intermediario. Por lo tanto, no son explícitamente operaciones ejecutadas por el *maintainer*, aunque es necesario solicitarle interactivamente algunos valores de argumentos a fin de mapear puertos y roles de conectores apropiadamente. Si no se contara con la potencia del modelo de operaciones para definir operaciones complejas, sería necesario ejecutar un mayor número de operaciones. Por ejemplo:

$\theta_{8\text{-}1}$={*addNewConnector(CSPNetwork$_{v1}$, 'OW-VPNGateway', ['Role11', 'Role12'],[PortOW1$_{v1}$, PortVPNG1$_{v1}$])* }
$\theta_{8\text{-}1}$={*addNewConnector(CSPNetwork$_{v1}$, 'OW-Presentation', ['Role21', 'Role22'],[PortOW2$_{v1}$, PortPresentation1$_{v1}$])* }

Continuando con el escenario, para mejorar el rendimiento del sistema es deseable que el componente de acceso a datos del webstore minorista tenga un acceso rápido (sin latencia) al catálogo de productos a la venta, junto con los precios y promociones activas. Para ello se requiere replicar el esquema de la base de datos que corresponde a dicha partición, el cual se despliega sobre el componente virtual *VM-DBServer* que ya ha sido creado. Además para mantener esta base de datos en la cloud actualizada, se crea un servicio de sincronización que funciona en un servidor de la empresa. Tal sincronización se realiza a través de la conexión de la VPN con el gateway.

θ_9={*replicateDataBase(DBSports$_{v1}$, 'DBSports-Catalogue', 'Products catalogue and marketing schemes', VM-DBServer$_{v1}$)*}
θ_{10}={*addNewConnector(CSPNetwork$_{v1}$, 'DataAccess-DBCatalogue', ['Role31', 'Role32'],[PortDA2$_{v2}$, PortDBC1$_{v1}$])* }
θ_{11}={*sincronizeDatabases(SystemSportsEv$_{v1}$, 'DBSynchronizer', [Resp-FindChanges, Resp-UpdateSchema],[PortSync1, PortSync2], DBSportsv1, DBSports-Cataloguev1, VPNGatewayv1)*}

La ejecución de esta última operación también es bastante compleja, como se mostró en la Fig. 7, ya que abarca la creación y el establecimiento de tres conectores entre la base de datos maestra y el sincronizador, entre el sincronizador y el gateway, y entre el gateway y la base de datos reducida en la cloud.

Es necesario incorporar además una conexión entre el componente de acceso a datos del webstore con el servidor de la base de datos maestra, a través de la cual se pueden incorporar las ventas minoristas que han sido concretadas.

θ_{12}={*addNewConnector(CSPNetwork$_{v1}$, 'DataAccess-VPNGateway', ['Role31', 'Role32'],[PortDA2$_{v1}$, PortVNPG3$_{v1}$])* }

La arquitectura de software resultante (Fig. 9) puede continuar evolucionando creándose nuevas instancias de servidores web y de aplicación según demanda, e incorporando componentes dedicados para el balanceo de carga.

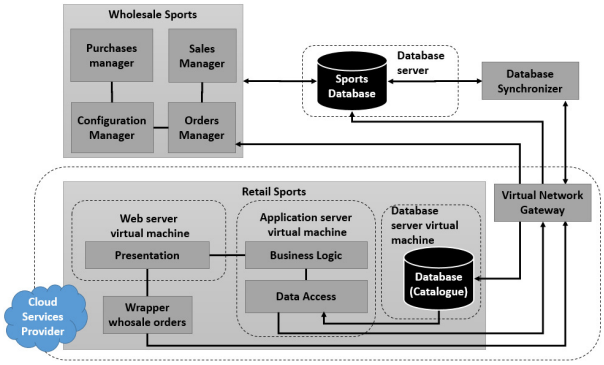


Fig. 9. Arquitectura final

7.2. Consultas / recuperación de información

A partir del proceso de evolución representado mediante la captura de las operaciones ejecutadas y los productos generados, es posible responder a diferentes preguntas acerca de cómo se llevó a cabo dicho proceso:

i) ¿Cuáles fueron los elementos arquitectónicos afectados por la evolución / el refactoring?
ii) ¿Qué operaciones de evolución o patrones de refactoring se ejecutaron?

iii) ¿Qué requerimientos de calidad se cambiaron, se incorporaron, o se intentaron mejorar?
iv) ¿Qué nuevos productos surgieron?
v) ¿Qué componentes se mantuvieron pero sufrieron modificaciones?
vi) ¿Qué cantidad de operaciones de refactoring fue necesario aplicar?
vii) ¿Qué operación (que materializa una decisión de diseño arquitectónico) dio lugar a la aparición de determinados productos (componentes, responsabilidades, conexiones, etc.)?

8. Conclusiones

El modelo propuesto posee una doble ventaja. Por un lado favorece y facilita la documentación del proceso de refactoring de arquitecturas de software, conservando la completa evolución a partir de una arquitectura inicial. El uso de esta documentación colabora en la compresión de los cambios que ya se han aplicado, de sus impactos en la arquitectura original. Por otro lado, permite sistematizar la aplicación de cambios en la arquitectura, ya que éstos se traducen en la aplicación de patrones de refactoring que han probado ser soluciones efectivas para problemas recurrentes.

La propuesta no está limitada a un lenguaje de descripción de arquitecturas particular, ni a un conjunto de vistas específicas, ni está atada a un método de diseño dado. El modelo de dominio es lo suficiente flexible para especificar todos los tipos de objetos de diseño y operaciones que sean necesarias para el dominio del sitema.

9. Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET, y la Universidad Tecnológica Nacional. Se agradece el apoyo brindado por estas instituciones.

10. Referencias

[1] Ahmad, A., Jamshidi, P., and Pahl, C., "A framework for acquisition and application of software architecture evolution knowledge", 14. *SIGSOFT Softw. Eng. Notes* 38, 5 (August 2013), pp. 1-7.

[2] Ahmad, A., Jamshidi, P., Pahl, C., and Khaliq, F. "PatEvol – A pattern language for evolution in component-based software architectures", in *Proceedings First Workshop on Patterns Promotion and Anti-patterns Prevention PPAP'2013*, Mar 2013, Genoa, Italy.

[3] Avgeriou, P., Kruchten, P., Lago, P., Grisham, P., and Perry, D., "Architectural knowledge and rationale: issues, trends, challenges", SIGSOFT Softw. Eng. Notes 32 (4), 2007, pp. 41–46.

[4] Barnes, J., Garlan, D., and Schmerl, B., "Evolution Styles: Foundations and Models for Software Architecture Evolution", *Journal of Software and Systems Modeling*, Vol. 13(2), May 2014, pp. 649-678.

[5] Breivold, H. P., Crnkovic, I., and Larsson, M., "A systematic review of software architecture evolution research", *Information and Software Technology*, 54, 1, January 2012, pp. 16-40.

[6] Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., and Stafford, J., *Documenting Software Architectures: Views and Beyond* (*2nd. ed.*), Addison-Wesley Professional, 2010.

[7] Côté, I., Heisel, M., and Wentzlaff, I., "Pattern-Based Evolution of Software Architectures", in *ECSA*, 2007.

[8] Falessi, D., Cantone, G.v and Kruchten, P., "Value-based design decision rationale documentation: principles and empirical feasibility study", in *Proc. Seventh Working IEEE/IFIP Conference on Software Architecture* (*WICSA 2008*), Vancouver, BC, Canada, 2008, pp. 189–198.

[9] Fowler, M., Beck, K., Brent, J., Opdyke, W., Roberts, D., *Refactoring: improving the design of existing code*, Addison-Wesley, 1993.

[10] Garlan, D., Monroe, R. T., and Wile, D., "Acme: Architectural Description of Component-Based Systems", in *Foundations of Component-Based Systems*, G. T. Leavens and Sitaraman (eds.), Cambridge University Press, 2000, pp. 47–68.

[11] Homer. A., Sharp, J., Brader, L., Narumoto, M., and Swanson, T., *Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications. Microsoft patterns & practices,* 2014.

[12] ISO/IEC/IEEE, "Systems and software engineering - Architecture description," ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), 2011.

[13] Ivkovic I. and Kontogiannis, K., "A Framework for Software Architecture Refactoring using Model Transformations and Semantic Annotations", in *Proceedings of the Conference on Software Maintenance and Reengineering* (CSMR '06). IEEE Computer Society, Washington, DC, USA, 2006, pp. 135-144.

[14] Roldán, M. L., Gonnet, S., and Leone, H., "Knowledge representation of the software architecture design process based on situation calculus", *Expert Systems,* **30**(1), 2013, pp. 34–53

[15] Roldán, M. L., Gonnet, S., and H. Leone, "TracED: a tool for capturing and tracing engineering design processes", *Advances in Engineering Software,* **41**(9), 2010, pp. 1087–1109.

[16] Stal M. Architecture refactoring foundation incl. a refactoring pattern catalog: https://dl.dropbox.com/u/2228034/ArchitectureRefactoringCatalog.pdf.

[17] Stal, M., "Chapter 3 - Refactoring Software Architectures", in Ali Babar, M. Brown, A. W., and Mistrik, I. (eds.) *Agile Software Architecture*, Morgan Kaufmann, 2014, pp. 63-82.

Modelado de Procesos en el Ministerio de Educación, Ciencia y Tecnología de Catamarca

Acosta Parra, Carlos Alberto[1]; Manfredi, Silvia Elisa[3]; Mazzurco, María de los Angeles[3]; Ovejero, Marcelo Alejandro[3]; Vilallonga, Gabriel Domingo[1 2]; Zurita Perez, Franco[3]

carlosacostap@tecno.unca.edu.ar; smanfredi@catamarca.edu.ar; angelesmazzurco@catamarca.edu.ar; marcelovejero@catamarca.edu.ar; gvilallo@unsl.edu.ar; francozurita@catamarca.edu.ar

[1]Universidad Nacional de Catamarca, Facultad de Tecnología y Ciencias Aplicadas

[2]Universidad Nacional de San Luis, Facultad de Ciencias Físico, Matemáticas y Naturales

[3]Gobierno de Catamarca, Ministerio de Educación, Ciencia y Tecnología

Abstract

El presente trabajo tiene como objetivo la aplicación de Business Process Management (BPM) en el ámbito de la Dirección Provincial de Recursos Humanos (DPRRHH) del Ministerio de Educación, Ciencia y Tecnología (MECyT) de la Provincia de Catamarca. Para ello, se plantearon las siguientes preguntas que sirvieron como eje a la presente investigación:

- *¿Qué procesos se ejecutan en la organización?*
- *¿En cuáles procesos debería enfocarse la organización?*

Durante la investigación, se emplearon herramientas y se llevaron a cabo un conjunto de actividades contempladas por la disciplina BPM. Más precisamente, y en base al ciclo de vida de BPM, se aplicaron sus dos primeras fases: la identificación de procesos y el descubrimiento de procesos. Como producto de la primera fase, la identificación de procesos, se obtuvo un entendimiento y una priorización de los procesos llevados a cabo en la organización. Como producto de la segunda fase, el descubrimiento de los procesos, y en base al producto de la primera fase, se obtuvo el modelo actual del proceso de mayor prioridad: la licencia de un docente y alta de un suplente.

Palabras Clave

Proceso de negocio – Modelado de procesos de negocio – Business Process Management – Factores críticos de éxito.

1. Introducción

Los procesos de negocio representan un activo fundamental de las organizaciones, ya que poseen un impacto directo en la percepción que el mercado posee sobre los productos y servicios brindados por éstas. Son los procesos los que determinan las tareas, trabajos y responsabilidades, y por ello, dan forma al trabajo de cada empleado. Además son los procesos los encargados de determinar el potencial de una organización para adaptarse a las nuevas circunstancias y para cumplir con un creciente número de requisitos legales.

Sin embargo, a diferencia de otros activos de la organización, la importancia de los procesos de negocio no fue apreciada durante un largo periodo [1].

Son las crecientes demandas de la globalización, la integración, la normalización, la innovación, la agilidad y la eficiencia operativa las que han logrado aumentar en las organizaciones la necesidad de mejorar sus procesos de negocio. En respuesta a esto, en las últimas dos décadas se ha desarrollado un conjunto de herramientas, técnicas, métodos y metodologías para prestar apoyo a todas las etapas del ciclo de vida de los procesos de negocio. Contribuciones relevantes han sido hechas por diversas disciplinas como la ingeniería industrial, la gestión de operaciones, la gestión de calidad, la gestión del capital humano, el gobierno electrónico, el modelado conceptual, la gestión de flujo de trabajo y la ingeniería de sistemas. Business Process Management (BPM) es la disciplina que consolida e integra la gran cantidad de estos enfoques.

Se destaca una segunda característica especial de BPM, la cual consiste en su activa práctica e investigación. Es un

campo donde existen prácticas tanto probadas y establecidas, así como nuevos desafíos abiertos. Organizaciones de todo el mundo están llevando a cabo iniciativas de BPM con el objetivo de superar a sus competidores o de satisfacer las necesidades de la demanda. Los académicos en campos como la informática, la gestión de la ciencia, la sociología y la ingeniería están trabajando en el desarrollo de métodos y técnicas para apoyar este tipo de iniciativas. Es apropiado ver a BPM como un campo de "teoría en la práctica". Por un lado, las exigencias prácticas inspiran el desarrollo de nuevos métodos y tecnologías. Por otro lado, la aplicación de estos métodos y tecnologías en la práctica alimenta de nuevo a los tableros de dibujo en las universidades y centros de investigación [1].

Es importante el estudio y la aplicación de BPM en contextos nuevos, como ser el caso de este trabajo, a través del cuál se buscó la aplicación de BPM en el ámbito de la Administración Pública de la Provincia de Catamarca.

2. Elementos de trabajo y metodología

2.1. Procesos de negocio

Un proceso de negocio abarca una serie de eventos y actividades. Los eventos se corresponden con las cosas que suceden de forma atómica, lo que significa que no tienen duración. Cuando una actividad es bastante simple y se puede ver como una sola unidad de trabajo, la denominamos como una tarea. Además de los eventos y actividades, un proceso típico involucra puntos de decisión, es decir, los puntos en el tiempo cuando se toma una decisión que afecta la forma en que se ejecuta el proceso. Esta decisión afecta lo que sucede más adelante en el proceso.

Un proceso también implica una serie de actores (actores humanos, organizaciones o sistemas de software que actúan en nombre de los actores humanos u organizaciones), los objetos físicos (equipos, materiales, productos, documentos de papel) y objetos inmateriales (documentos electrónicos y registros electrónicos).

La ejecución de un proceso conduce a uno o varios resultados. Idealmente, el resultado de un proceso debe ofrecer valor a los actores involucrados. La Figura 1 muestra todos los elementos que componen un proceso de negocio.

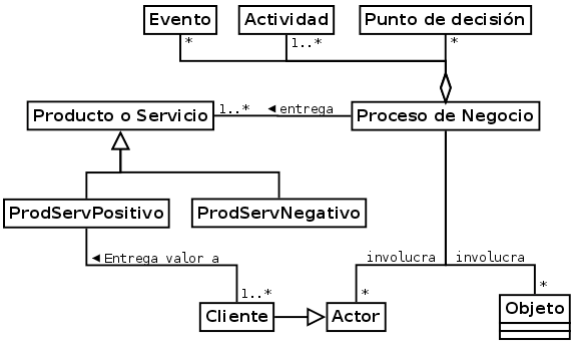


Figura 1. Definición de proceso de negocio

2.2. El ciclo de vida de BPM

El ciclo de vida de BPM consta de 6 fases [1], según se puede apreciar en la Figura 2. Este trabajo se enfocó en el uso de las primeras dos fases, las cuales serán descritas brevemente a continuación.

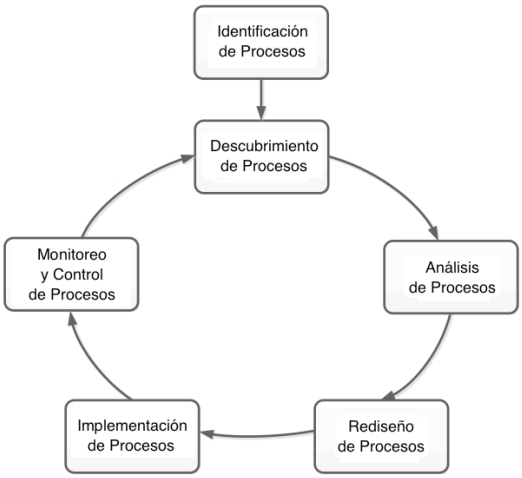


Figura 2. Ciclo de vida de BPM

2.2.1. La identificación de procesos

La identificación de procesos es un conjunto de actividades destinadas a definir de manera sistemática el conjunto de procesos de negocios de una organización, y a establecer claros criterios para priorizarlos. El producto de la fase de identificación de procesos es una arquitectura de procesos, que representa a

los procesos de negocio y sus interrelaciones. Una arquitectura de procesos sirve como marco para la definición de prioridades, el alcance de los proyectos de modelado y el rediseño de procesos.
La identificación de procesos está formada de dos fases sucesivas: la designación y la evaluación. El objetivo de la fase de designación es comprender los procesos de una organización, así como sus interrelaciones. La fase de evaluación propone desarrollar una priorización entre los procesos identificados.

2.2.2. El descubrimiento de procesos

El descubrimiento de procesos se define como el acto de reunir información acerca de un proceso existente y organizarla con el fin de obtener modelos actuales del proceso. El descubrimiento de procesos es una actividad mucho más amplia que el modelado de procesos. Claramente, el modelado es parte de esta actividad. El problema radica en que para realizar el modelado se necesita reunir la información necesaria. De hecho, la recopilación de información a menudo resulta ser engorrosa y lleva mucho tiempo llevarla a la práctica. Por lo tanto, tenemos que definir primero un entorno en el que la información puede ser obtenida de manera efectiva. Con el fin de abordar estas cuestiones, podemos describir las siguientes tres fases dentro del descubrimiento de procesos:

- Definición inicial: el objetivo de esta fase es reunir un equipo que se encargará de trabajar en el proceso.
- Recopilación de información: esta fase se encuentra relacionada con la construcción de una comprensión del proceso. Diferentes métodos de descubrimiento pueden utilizarse para adquirir información sobre un proceso.
- Modelado: esta fase se refiere a organizar la creación del modelo de proceso. El método de modelado brinda una orientación para representar el proceso de una manera sistemática.

2.2.2.1. La evaluación de procesos

La fase de evaluación se sustenta en la idea de que no todos los procesos son igualmente importantes, y por lo tanto no pueden recibir el mismo grado de atención. Existen varios criterios para realizar esta evaluación:

- Importancia: el objetivo es averiguar qué procesos tienen mayor impacto de acuerdo a los factores críticos de la organización.
- Disfunción: este criterio pretende evaluar la "salud" de cada proceso. La cuestión aquí es determinar qué procesos poseen problemas más profundos.
- Viabilidad: para cada proceso, se debe determinar su grado de susceptibilidad respecto a una iniciativa de gestión de procesos.

3. Resultados

3.1. Identificación de procesos

3.1.1. Designación de procesos

La primera actividad realizada fue una entrevista con la Directora Provincial de Recursos Humanos, con el fin de obtener información respecto de la organización. De esta entrevista se pudo obtener la siguiente información:

- Listado de procesos: el cual fue extraído del sistema de gestión informático de la DPRRHH. Vale aclarar que del listado completo de los tipos de trámites, la Directora decidió eliminar del listado entregado, aquellos trámites que ella consideraba irrelevantes para ser tenidos en cuenta.
- Manuales de procesos: confeccionados en el año 2004, pero nunca implementados. Los procesos actuales de la DPRRHH no se llevan a cabo teniendo en cuenta estos manuales.
- Organigrama del MECyT: graficado a través de un árbol, en el cual se puede apreciar que la DPRRHH depende de la Subsecretaría de

Coordinación Administrativa (SSCA), y que a su vez esta Subsecretaría depende del Ministro.
- Organigrama y manual de objetivos y funciones de la SSCA: este documento contiene a su vez el organigrama y manual de objetivos y funciones de la DPRRHH

A continuación de detallará la información más relevante obtenida a partir de cada uno de los ítems anteriormente mencionados.

3.1.1.1. Listado de procesos

1. Alta: alta de un agente en un cargo.
2. Baja: baja de un agente de un cargo.
3. Licencia: registro y/o autorización de licencia de un agente.
4. Licencia y alta de suplente: registro y/o autorización de licencia de un agente, y de acuerdo a la normativa vigente, ese cargo debe ser cubierto por un suplente, este mismo proceso contempla el alta de ese suplente.
5. Cambio de situación de revista: registro del cambio de situación de revista de agente en un cargo (titular, suplente, interino, contratado, planta permanente, etc.).
6. Reasignación de cargo: cambio en la relación de un agente de un cargo hacia otro cargo, debido por ejemplo a situaciones en la cuales la planta orgánica funcional de un organismo ha sido modificada.
7. Adscripción: es la autorización para que un agente preste servicios en otro organismo por un lapso de tiempo estipulado.
8. Reclamo salarial: todo tipo de reclamo asociado a errores en la liquidación de haberes.
9. Certificación de servicios: pedido de certificaciones de agentes relacionadas a todas las prestaciones de servicios realizadas o que realiza actualmente.
10. Reconocimiento de deuda: llevado a cabo debido a los retrasos incurridos en las altas, ya sea por la cobertura de un nuevo cargo, o por la cobertura de una suplencia.
11. Cambio de funciones: debido a alguna situación, un agente puede solicitar un cambio de funciones, el caso más común es el cambios de funciones a tareas pasivas por motivos de salud.
12. Traslado: es la transferencia de un agente de un organismo a otro.
13. Salario familiar: contempla los trámites iniciados por los agentes para cobrar salarios por poseer familiares a cargo.
14. Seguro de vida: todo agente posee un seguro de vida obligatorio, para lo cual debe iniciar un trámite para que este seguro entre en vigencia.
15. Control de asistencia: control mensual que se realiza para todos los agentes en todos los organismos dependientes del Ministerio.

3.1.1.2. Manuales de procesos

a) Novedades laborales: altas, bajas, inicios de licencia, fines de licencia, cambios de situación de revista, reubicación y otros movimientos de agentes en un puesto de trabajo.
b) Novedades salariales: toda novedad que implique un movimiento salarial de un agente docente y no docente, como ser: datos de salario familiar, antigüedad, afiliaciones, embargos judiciales, títulos obtenidos, reclamos de liquidación de haberes, u otra situación relacionada.
c) Reestructura de plazas: relacionado a las altas o bajas de puestos de trabajo.

3.1.1.3. Tipos de casos

Se pudo diferenciar a los procesos de acuerdo al tipo de cliente y al tipo de servicio.

3.1.1.3.1. Tipos de cliente

- Docente: presta funciones frente a alumnos.
- No docente: todo el personal que no es docente.

3.1.1.3.2. Tipo de servicio

- Novedad laboral: novedad que afecta la relación de un agente con un puesto de trabajo.
- Novedad salarial: novedad que afecta el salario de un agente.
- Certificación: entrega de certificación relacionada a la información que administra la DPRRHH.

3.1.1.4. Procesos asociados a cada tipo de cliente

Listado de trámites asociados a los docentes: alta, baja, licencia, licencia y alta de suplente, cambio de situación de revista, reasignación de cargo, reclamo salarial, certificación de servicios, reconocimiento de deuda, cambio de funciones, salario familiar, seguro de vida, y control de asistencia.

Listado de trámites asociados a los no docentes: alta, baja, licencia, cambio de situación de revista, adscripción, reclamo salarial, certificación de servicios, traslado, salario familiar, seguro de vida, y control de asistencia.

3.1.1.5. Procesos asociados a cada tipo de servicio

Procesos asociados a novedades laborales: alta, baja, licencia, licencia y alta de suplente, cambio de situación de revista, reasignación de cargo, adscripción, cambio de funciones, traslado, y control de asistencia.

Procesos asociados a novedades salariales: reclamo salarial, reconocimiento de deuda, salario familiar, y seguro de vida.

Procesos asociados a certificaciones: certificación de servicios.

3.1.1.6. Organigrama del Ministerio

Donde se puede visualizar que la DPRRHH depende de la SSCA, la cual depende del Ministro.

3.1.1.7. Organigrama de la Subsecretaría de Coordinación Administrativa con sus objetivos y funciones

Dentro de este documento se encuentra el organigrama de la DPRRHH con sus respectivos objetivos y funciones. De este documento la información más relevante extraída fue el listado de áreas que dependen de la Dirección Provincial de Recursos Humanos, el cual detallamos a continuación:

1. Control y carga de datos.
2. Seguros de vida.
3. Sistemas.
4. Asesoría legal.
5. Auditoría de plantas orgánicas funcionales y nominales.
6. Control del gasto.
7. Despacho.
8. Liquidación de haberes.
9. Licencias.
10. Mesa de entradas y salidas.
11. Movilidad docente.
12. Patrimonio.
13. Salario familiar.
14. Legajos.
15. Director.

3.1.1.8. Procesos asociados a cada área de la DPRRHH

La siguiente etapa realizada consistió en la realización de entrevistas al personal de cada una de las áreas anteriormente listadas. Para el diseño de las entrevistas de cada una de las áreas, se tuvo en cuenta la información del organigrama y manuales de objetivos y funciones de la DPRRHH. Además, se indagó en cada una de las entrevistas sobre la relación del área con cada uno de los procesos antes listados. Con lo cuál se pudo obtener como resultado un cruce de información entre los procesos y las áreas, lo cual nos permitió identificar a

través de que áreas fluyen cada uno de los procesos.

Alta (Alt): control del gasto, despacho, liquidación de haberes, mesa de entradas y salidas, movilidad docente, legajos, y director.

Baja (Baj): control y carga de datos, despacho, liquidación de haberes, mesa de entradas y salidas, movilidad docente, legajos, y director.

Licencia (Lic): control y carga de datos, despacho, licencias, mesa de entradas y salidas, legajos, y director.

Licencia y alta de suplente (LiS): control y carga de datos, control del gasto, despacho, liquidación de haberes, licencias, mesa de entradas y salidas, movilidad docente, legajos, director.

Cambio de situación de revista (CSR): despacho, mesa de entradas y salidas, movilidad docente, legajos, y director.

Reconocimiento de deuda (ReD): liquidación de haberes, y director.

Cambio de funciones (CaF): mesa de entradas y salidas, movilidad docente, legajos, y director.

Traslado (Tra): control y carga de datos, mesa de entradas y salidas, y legajos.

Salario familiar (SaF): mesa de entradas y salidas, salario familiar, y legajos.

Seguros de vida (SeV): seguros de vida, mesa de entradas y salidas, y legajos.

Control de asistencia (CoA): control y carga de datos.

3.1.2. Evaluación de procesos

Se inició la evaluación de procesos con una nueva entrevista a la Directora Provincial de Recursos Humanos, de la cual se decidió fijar nuestra atención en los procesos de novedades laborales asociados a los docentes. Esta decisión se basó en que la mayor cantidad de cargos que se administran son cargos docentes (alrededor del 93%), según se puede apreciar con exactitud en las Figuras 3 y 4.

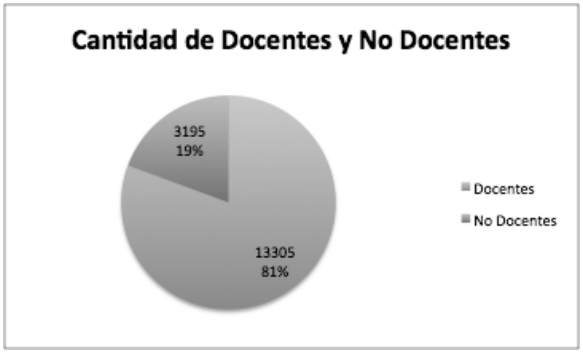


Figura 3

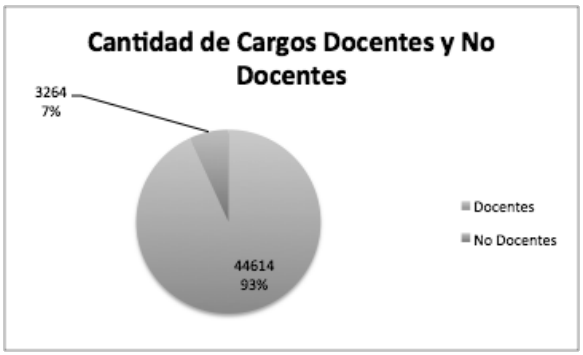


Figura 4

A continuación, en la Tabla 1 se puede apreciar el listado de los procesos de novedades laborales y su relación con los docentes y no docentes.

Procesos	Docente	No Docente
Alt	X	X
Baj	X	X
Lic	X	X
LiS	X	
CSR	X	X
ReC	X	
Ads		X
CaF	X	
Tra		X
CoA	X	X

Tabla 1. Procesos de novedades laborales

3.1.2.1. Búsqueda del proceso de mayor relevancia

3.1.2.1.1. Factores críticos de éxito (FCE)

Para la búsqueda del proceso que requiere atención prioritaria, se emplean los FCE, los cuales describen pautas que la organización debe llevar a cabo para alcanzar sus objetivos. Los FCE descriptos a continuación, fueron obtenidos a partir de

una entrevista con la Directora de la DPRRHH:

A. Incremento en la automatización de operaciones y validaciones.
B. Eliminación de tareas redundantes que no agregan valor al proceso.
C. Capacitación de todos los involucrados.
D. Mejoramiento de los canales de comunicación con todos los involucrados.
E. Mejoramiento de la interoperabilidad entre las distintas áreas involucradas.
F. Aumento de la motivación de los involucrados.
G. Normativa acorde a los cambios esperados.
H. Mejoramiento de la atención al público

3.1.2.1.2. Ponderación de la relación entre los procesos y los factores críticos de éxito (FCE)

En esta técnica el objetivo es ordenar los procesos involucrados según su prioridad, teniendo en cuenta la relación de cada Proceso con los FCEs. Para ello, se emplearán las Tablas 2 y 3, cuya estructura es la siguiente:

- Filas: todo proceso bajo consideración.
- Columnas: los FCEs considerados.
- Contenido de cada nodo: la evaluación de la relación entre un proceso y un FCE:
  - Cero: no existe relación.
  - Uno: apenas relacionado.
  - Dos: medianamente relacionado.
  - Tres: fuertemente relacionado.

	A	B	C	D
Alt	3	3	2	3
Baj	2	3	3	3
Lic	2	2	3	2
LiS	3	3	3	3
CSR	1	1	2	1
ReC	3	2	2	2
CaF	2	2	2	3
CoA	1	3	2	3

Tabla 2. FCE parte 1 de 2

	E	F	G	H	Tot
Alt	3	2	1	3	20
Baj	3	3	2	3	22
Lic	3	3	3	3	21
LiS	3	3	3	3	24
CSR	1	3	0	3	11
ReC	3	3	3	3	21
CaF	3	3	3	3	21
CoA	3	2	1	1	16

Tabla 3. FCE parte 2 de 2

3.1.2.1.3. Relación entre la ponderación de procesos y el nivel de calidad actual de los procesos

Esta técnica tiene como objetivo detectar los procesos en los cuales se debe comenzar a trabajar primero, de manera de obtener un fuerte efecto inicial. Los procesos con peor calidad de implementación actual y con una importante cantidad de FCE asociados deben ser encarados inicialmente.

La estructura de la matriz es la siguiente:

- Filas: todos los procesos que tienen la misma cantidad de FCE asociados.
- Columnas: Evaluación de la calidad de la actual implementación de los procesos.
  - "E": Pésima calidad de implementación.
  - "D": Mala.
  - "C": Apenas aceptable.
  - "B": Aceptable.
  - "A": Alta calidad de la implementación de los procesos.

- Contenido de cada nodo: la denominación de los procesos que tienen un determinado número de FCE asociados y una específica evaluación de calidad.

Los procesos que se encuentran en el extremo superior derecho son los que necesitan atención primaria, debido a que están íntimamente relacionados con los objetivos que se persiguen y tiene un grado muy bajo de calidad de implementación.

	A	B	C	D	E
24				LiS	
22			Baj		
21		Lic	CaF	ReC	
20			Alt		
16				CoA	
11		CSR			

Tabla 4. Priorización de procesos

Luego de la aplicación esta técnica, se concluyó que el proceso de licencia y alta de suplente debe ser el primer proceso sobre el cual debemos trabajar.

3.2. Descubrimiento del proceso de licencia y alta de suplente

3.2.1. Descubrimiento por medio de taller con los encargados de todas las áreas implicadas

De la información obtenida en la identificación del proceso, se pudo observar que el proceso de licencia y alta de suplente fluye a través de las siguientes áreas:

- Control y carga de datos.
- Control del gasto.
- Despacho.
- Liquidación de haberes.
- Licencias.
- Mesa de entradas y salidas.
- Movilidad docente.
- Legajos.
- Director.

Basados en esta información se organizó un taller con los encargados de cada una de las áreas involucradas. A lo largo del taller se llevaron a cabo varias iteraciones, las cuales pasaremos a detallar a continuación:

3.2.1.1. 1º iteración: identificación de los participantes en el ámbito de la DPRRHH

La primera actividad que se llevó a cabo tuvo como objeto obtener un diagrama que represente la secuencia de intervenciones de cada una de estas áreas a lo largo del proceso. El resultado de la secuencia es el siguiente:

1. Mesa de entradas y salidas.
2. Licencias.
3. Control y carga de datos.
4. Legajo.
5. Mesa de entradas y salidas.
6. Control del gasto.
7. Movilidad docente.
8. Director.
9. Despacho.
10. Mesa de entradas y salidas.
11. Liquidación de haberes.
12. Legajos.

3.2.1.2. 2º iteración: identificación de los participantes fuera de la DPRRHH

Luego, se indagó a los participantes sobre los organismos externos que participan de este proceso. Para ello, se confeccionó una nueva versión de la secuencia, pero esta vez incluyendo a los organismos externos en ella.

1. Docente.
2. Reconocimientos médicos.
3. Escuela.
4. Mesa de entradas y salidas.
5. Licencias.
6. Control y carga de datos.
7. Legajos.
8. Escuela.
9. Dirección de nivel.
10. Junta de clasificación de nivel.
11. Suplente.
12. Escuela.
13. Mesa de entradas y salidas.
14. Control del gasto.
15. Movilidad docente.
16. Director.
17. Despacho.
18. Mesa de entradas y salidas.

19. Liquidación de haberes.
20. Legajos.

3.2.1.3. 3º iteración: identificación de las actividades

En esta tercera iteración, se agregó a cada uno de los pasos de la secuencia una descripción de la tarea realizada:

1. Docente: solicita licencia.
2. Reconocimientos médicos: autoriza licencia.
3. Escuela: envía información sobre la novedad laboral generada por la licencia del docente.
4. Mesa de entradas y salidas: controla y recibe la novedad laboral de la licencia.
5. Licencias: controla y registra la licencia del docente.
6. Control y carga de datos: controla que la documentación de la licencia y los datos cargados coincidan y se encuentran en correcto estado.
7. Legajos: archiva la documentación en el legajo del agente.
8. Escuela: envía solicitud de cobertura por parte de un suplente.
9. Dirección de nivel: autoriza solicitud de cobertura por parte de un suplente.
10. Junta de clasificación de nivel: designa un suplente para cubrir el cargo.
11. Suplente: concurre a la escuela y realiza la toma de posesión del cargo.
12. Escuela: completa formularios y envía toda la documentación necesaria para iniciar el trámite de alta del suplente.
13. Mesa de entradas y salidas: recibe y controla que la documentación se encuentre en perfecto estado.
14. Control del gasto: controla que el suplente luego de tomar el cargo no quede en estado de incompatibilidad por acumulación de cargos de acuerdo a la normativa vigente.
15. Movilidad docente: controla información del cargo, registra el alta del suplente e imprime disposición.
16. Director: firma disposición.
17. Despacho: protocoliza la disposición.
18. Mesa de entradas y salidas: notifica a la Escuela.
19. Liquidación de haberes: activa la designación del suplente en el cargo, para que pueda ser liquidado.
20. Legajos: archiva toda la documentación.

3.2.1.4. 4º iteración: identificación de los flujos de control

Actividad 2: si el organismo rechaza el pedido de licencia al docente, el proceso finaliza.

Actividades 3 y 8: estas actividades pueden ser realizadas en paralelo.

Actividad 4: si la documentación posee algún tipo de error o faltante, se devuelve a la Escuela para que realice las correcciones pertinentes.

Actividad 5: si la licencia es incompatible, se devuelve la documentación a la mesa de entradas y salidas, y luego a la escuela para que realice las correcciones pertinentes o informe del rechazo de la licencia al docente.

Actividad 6: si se encuentre algún error en la carga de datos o documentación se devuelve la documentación al área de licencias.

Actividad 9: si no se autoriza la cobertura, se informa a la escuela del rechazo.

Actividad 13: si la documentación posee algún tipo de error o faltante, se devuelve a la escuela para que realice las correcciones pertinentes.

Actividad 14: si el suplente quedara incompatible por acumulación de cargos luego de tomar el nuevo cargo, se devuelve la notificación a la mesa de entradas y salidas para que informe sobre el rechazo a la escuela solicitante.

Actividad 15: Si existe algún tipo de error relacionado al cargo que intenta cubrir el suplente, se devuelve la documentación a la mesa de entradas y salidas para que

notifique a la escuela sobre las correcciones necesarias que debe realizar.

3.2.1.5. 5º iteración: desglose en subprocesos

Se procedió a desglosar al proceso anterior en cuatro grandes subprocesos (Figura 5):

- Solicitud de licencia: comprende las actividades 1 y 2 (Figura 6).
- Registro de licencia en la DPRRHH: está formado por las actividades 3, 4, 5, 6 y 7 (Figura 7).
- Cobertura de suplencia: comprende las actividades 8, 9, 10 y 11 (Figura 8).
- Alta del suplente: formado por las actividades 12, 13, 14, 15, 16, 17, 18, 19 y 20 (Figura 9).

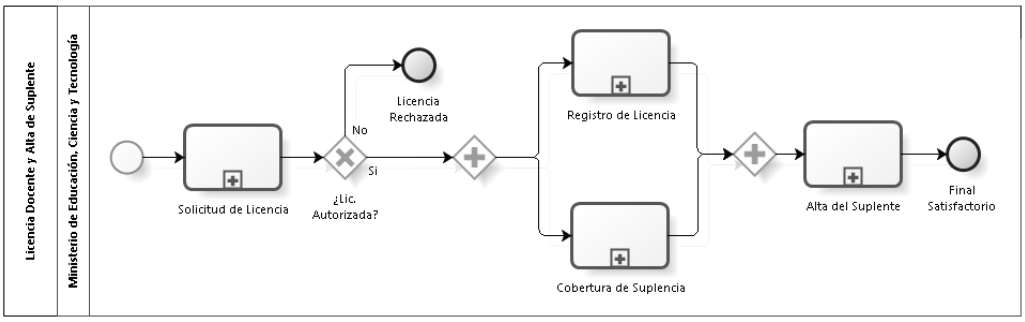


Figura 5. Digrama desglosado del proceso de licencia y alta de suplente

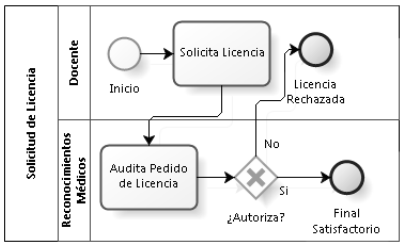


Figura 6. Subproceso solicitud de licencia

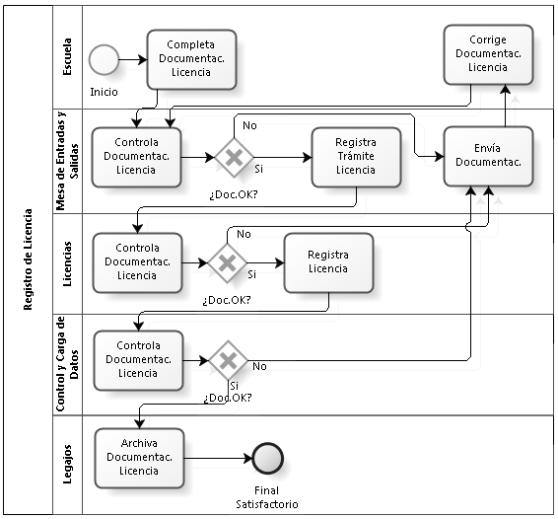


Figura 7. Subproceso registro de licencia

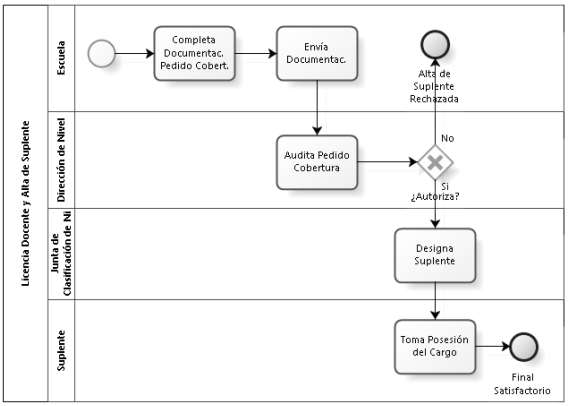


Figura 8. Subproceso cobertura de suplencia

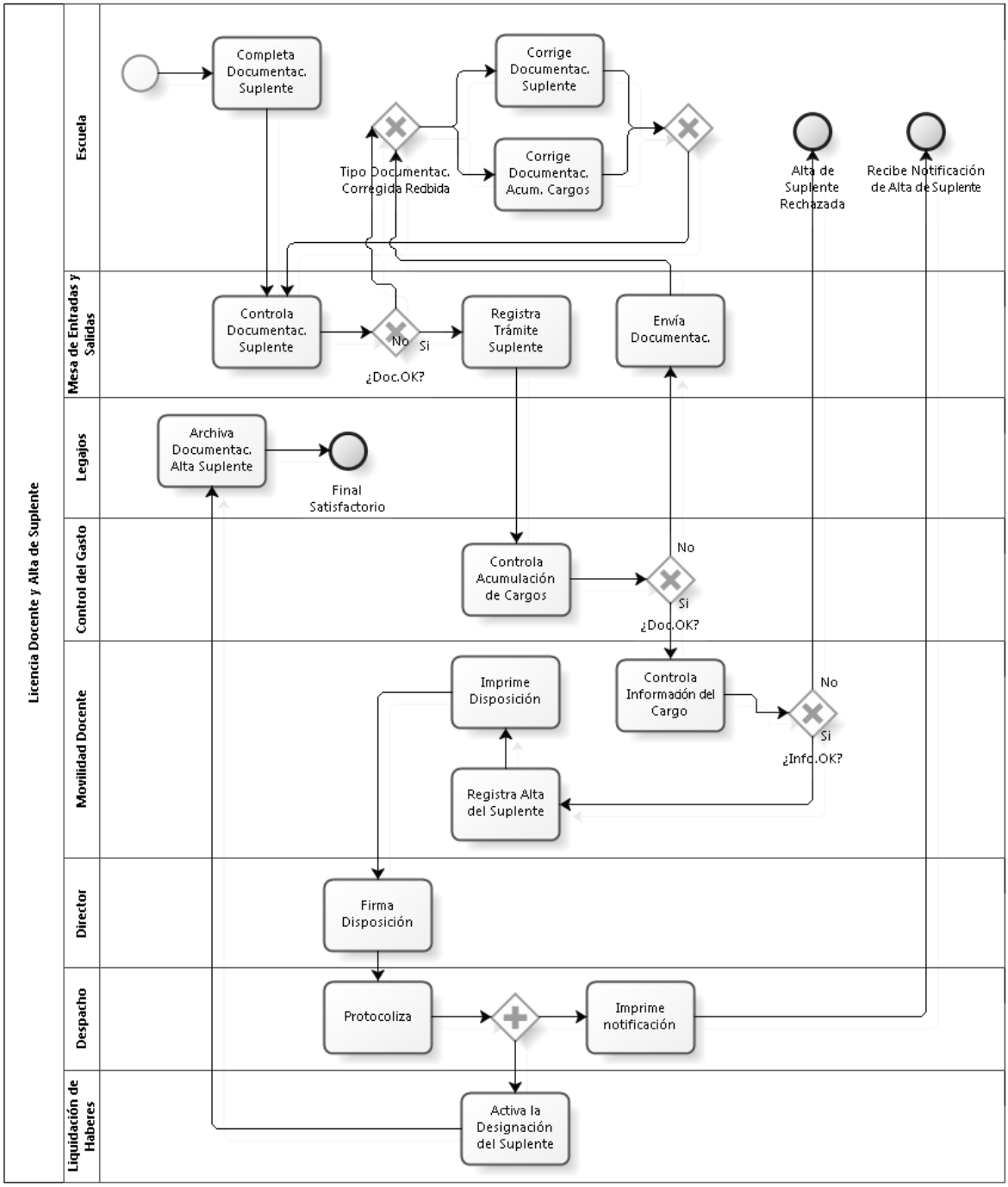


Figura 9. Subproceso alta del suplente

4. Discusión

Se pueden refinar aún más los modelos obtenidos, realizando iteraciones que se centren en los eventos intermedios, objetos de datos, creación de subprocesos, información sobre tareas (tareas automáticas, iterativas, etc.).

Este trabajo sienta las bases para el análisis y rediseño del proceso modelado. Se podrían modelar los procesos que siguen el orden de prioridad posterior al proceso modelado. A su vez, se podrían llevar a cabo similares trabajos en otros ámbitos de la Administración Pública.

5. Conclusión

Se ha demostrado que utilizando BPM, es posible determinar por medio de entrevistas, estudio de documentación y talleres cuales son los procesos que se ejecutan en la organización. Además, tomando como referencia los FCE, se ha

podido determinar cuál es el proceso en el cual la organización debe enfocarse. Los diagramas BPMN resultantes son una buena representación de la ejecución real del proceso.

6. Agradecimientos
Al Personal de la Dirección Provincial de Recursos Humanos del Ministerio de Educación, Ciencia y Tecnología de Catamarca.

7. Referencias

[1] Dumas, M., La Rosa, M., Mendling J. y Reijers H. (2013). Fundamentals of Business Process Management. Alemania: Springer.

[2] Perez Fernandez de Velasco, J. (2010). Gestión por procesos. España: Alfaomega.

[3] White S. y Miers, D. (2008). BPMN Modeling and Reference Guide. Estados Unidos: Future Strategies.

[4] Grover, V. y Kettinger, W. (1998). Business Process Change: Concepts, Methods Technologies. Estados Unidos: Idea Group Publishing.

[5] Chang, R. (1996). Mejora Continua de Procesos. Argentina: Ediciones Granica.

[6] Accenture. (2010). Aceleración para el desarrollo de sistemas y procesos. Argentina: Pearson Education.

[7] Ardoino, J. (2014). *El diseño de procesos en organizaciones efectivas. Metodología INTRéGRA*. Argentina: Edicon.

Modelo de calidad de productos de software

Paula M. Angeleri, Alejandro Oliveros, Amos Sorgen, Rolando Titiosky, Jaquelina Wuille Bille.

Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano

Av. Federico Lacroze 1947, C.P. 1426 C.A.B.A., Buenos Aires, Argentina

{paula.angeleri}@ub.edu.ar; {alejandro.oliveros, amos.sorgen, rolando.titiosky, Jaquelina.wuillebille}@comunidad.ub.edu.ar

Resumen

Los modelos de calidad de productos software no contemplan todos los contextos en los que un software es desarrollado, operado o mantenido a lo largo de su ciclo de vida, ni soportan todas las características de los sistemas actuales. En este artículo se presenta un modelo de calidad de productos de software denominado QSAT[1] cuyo propósito es ser más completo y actualizado que sus predecesores: en particular el modelo FURPS+, los factores de calidad de McCall, la norma ISO/IEC 9126-1 y su revisión bajo el número ISO/IEC 25010.

1. Introducción

La calidad de un producto software se evalúa utilizando un modelo de calidad de productos software. Los modelos de calidad deben acompañar la evolución del software, a fin de evaluar todas las capacidades de un software desarrollado en la actualidad y de contemplar todos los contextos en donde un software opera o se desarrolla. En respuesta a esta necesidad se desarrolló el modelo de calidad QSAT, que proporciona una base conceptual e instrumental para evaluar la calidad de un producto software. Este modelo integra el Framework de evaluación de software MyFEPS[2] [1]. Sus principales aportes son: proporciona algoritmos procedimentales para dar claridad a las métricas que se utilizan para medir la calidad de un producto, incrementa los contextos de la evaluación, aumenta los ítems de calidad que se pueden evaluar, define el grado de Rigurosidad (Rig) con el que hay que medir cada Atr, y proporciona mecanismos para ponderarlos en función de su importancia para la evaluación.

En respuesta a la necesidad de certificación de los productos software el Instituto Nacional de Normalización y Certificación de Argentina (IRAM) utilizaba el proceso de evaluación referenciado en la serie de normas ISO/IEC 14598 y el modelo de calidad de la serie ISO/IEC 9126. La pertinencia de esta utilización fue debatida en el Subcomité de Calidad en Tecnología de la Información de IRAM. Por ejemplo, los aspectos de seguridad de la información, incluidos como una sub-característica de la Funcionalidad del producto, eran tratados en forma incompleta, haciendo foco en la seguridad de acceso y en las trazas para auditorías, en desmerito de otros aspectos como ser la integridad de los datos, el no repudio, etc.

Por su parte los expertos que desarrollaron el proceso de certificación de productos software de IRAM[3] y de la Comisión de Calidad de la Cámara de Empresas de Software y Servicios Informáticos de Argentina (CESSI), analizaban la adecuación de las métricas para evaluar las sub-características de calidad y su interpretación. La norma utilizada no explicitaba los criterios de ponderación de las mediciones en función de la importancia relativa de una característica en evaluación o respecto de los objetivos de la evaluación.

La certificación de software, considerada en la Ley 25.922[4] Ley de Promoción de la Industria del Software, ofrecía la posibilidad de certificar el software bajo estas normas ISO/IEC [2][3] como una de las condiciones para ingresar al régimen de promoción (Anexo VI de la Resolución 61) [4]. De allí que en la Comisión de

[1] QSAT es una sigla proveniente de los apellidos de los creadores del modelo: Amos **S**orgen, Paula **A**ngeleri, y Rolando **T**itiosky la **Q** es por ***Q**uality* que significa Calidad en Inglés.

[2] MyFEPS es la sigla proveniente del nombre del proyecto de investigación que le dio origen: *"**M**etodologías **y F**ramework para la **E**valuación de **P**roductos **S**oftware"*, planificado en el año 2009 e iniciado en el año 2010, con el propósito de cubrir las falencias detectadas en modelos de calidad de productos de software.

3 Domingo F. Donadello, Jorge Ceballos, Paula M. Angeleri.

4 Al momento de redactar este artículo el período de vigencia de la Ley 25.922 había concluido, y se había aprobado la nueva Ley 26.692 con el objeto de prorrogar el Régimen de Promoción de la Industria del Software de Argentina [5].

Calidad de CESSI se buscaba asignar una importancia relativa adecuada a cada sub-característica de calidad, para encontrar la forma más apropiada de evaluar el grado de calidad de una característica.

Por ejemplo, en el caso que se quiera medir el grado de calidad de la característica Funcionalidad de un producto software: ¿qué sub-característica deberíamos considerar más relevante?, ¿su adecuación?, ¿su exactitud o precisión?, ¿su capacidad para inter operar o interoperabilidad?, ¿su seguridad de acceso?, ¿su cumplimiento legal? [6] ¿O se podría asignar igual peso de importancia a todas estas Sub-características? Este debate instaló la necesidad de ponderar la importancia relativa de cada ítem de calidad en el contexto de una evaluación.

Por lo expuesto, surge la necesidad de elaborar un nuevo modelo de calidad de productos software actualizado al estado actual de la tecnología, con definiciones precisas de procedimientos que indicaran cómo realizar una medición de uno o más atributos de calidad, sin ambigüedades. En definitiva se trata de lograr un modelo en el que dos evaluadores interpreten una métrica de la misma manera y que resulte de fácil comprensión la valoración de las características de calidad de un producto software.

El modelo QSAT surge en respuesta a la problemática que enfrentaba Argentina y el mundo respecto de la evaluación de software. Al momento de concebirse la idea del proyecto MyFEPS.

El resto de este artículo está estructurado en las siguientes secciones: en la sección 2 se presenta el estado de arte en modelos de calidad de software más difundidos, la sección 3 describe el modelo QSAT, la sección 4 presenta un caso de estudio que usa el modelo de calidad QSAT en la evaluación de un sistema informático de administración de historias clínicas, la sección 5 presenta las conclusiones, y finalmente en la sección 6 están las referencias.

2. Estado del arte

Los modelos de calidad analizados fueron los de mayor difusión en el ámbito de la Ingeniería de Software en general: FURPS+, el de Boehm y el de McCall, e ISO/IEC 9126-1. Estos modelos tenían al menos uno de los siguientes problemas:

- No diferenciar todos los entornos en los que los sistemas modernos se desenvuelven, como el de *desarrollo* o *en testing*, entre otros.
- No incorporar todos los ítems actualmente medibles, dejando afuera de la evaluación aspectos del sistema software.
- No explicar en detalle el proceso de evaluación y cuando lo hacen (como ISO/IEC), no incluyen una guía de bajo nivel. Esto hace que las Organizaciones dependan de las personas y sus interpretaciones de las normas o modelos.
- No especificar cómo se establece la importancia de un ítem de calidad respecto de otro y de los objetivos de la evaluación (ver ejemplo anterior de la característica funcionalidad).

En la Tabla 1 se resumen los modelos y normas analizadas, incluyendo la norma ISO/IEC 25010 [7] que reemplazó a la norma ISO/IEC 9126 (se ha reemplazado, a la fecha, la parte 1), faltan publicar sus métricas por lo que han quedado fuera de este análisis[8][9]. La tabla registra los contextos en los que el software opera o se desarrolla, la cantidad de Características (C) de Nivel 1, la cantidad de Subcaracterísticas (SC) de Nivel 2, si hay o no descripciones procedimentales de la forma de realizar las mediciones y por último un nivel de complejidad para las personas que intervengan en una evaluación. No se indicó en la tabla por redundante que ninguna incluye un procedimiento para obtener la importancia relativa de ítems de calidad (peso utilizado para ponderarlas) ni una definición formal completa del proceso de evaluación,

Tabla 1: Comparación de modelos

Modelo /Norma	Entornos	C	SC	Descripción del Proceso de evaluación	Com-plejidad
ISO/ IEC 9126	Interno Externo En Uso	10	27	No. Se complementa con ISO/IEC 14598	Media
ISO/ IEC 25010		13	41	Si	Media
FURPS	En Uso	5	23	No	Baja
Boehm		7	15		Media
McCall	Revisión Operación Transición	3	11		Alta

El modelo de McCall es demasiado complejo para ser bien comprendido y utilizado. Tiene una valoración indirecta: para obtener una valoración de calidad se usan coeficientes de correlación para derivar un Factor de Calidad de una característica particular. Al respecto de este tema, Pressman señala que "por desgracia, muchos de los parámetros definidos por McCall se pueden medir sólo subjetivamente" [10] Por lo tanto, es difícil utilizar este marco para establecer requisitos precisos y

específicos de calidad. Mc Call hace hincapié en la perspectiva de la calidad del producto en detrimento de las otras perspectivas, por ejemplo, la calidad de la Ingeniería de software. Si bien la ISO/IEC 25010 ha actualizado el modelo de calidad propuesto por la ISO/IEC 9126 [9], aún no puede utilizarse seriamente para evaluación por no estar publicadas la actualización de las normas ISO/IEC 9126-2, que establece las métricas externas, ISO/IEC 9126-3 que define las métricas internas, e ISO/IEC 9126-4, que define las métricas que recomienda el comité técnico conjunto de ISO y de IEC para medir la calidad en el uso.

En síntesis, los modelos FURPS+, Boehm, e ISO/IEC 9126 son limitados pues sus métricas algunas veces resultan difíciles de comprender. Las entradas indicadas para calcular las métricas o los atributos a medir no siempre son suficientes para poder realizar las mediciones con resultados precisos. Presentan limitaciones a la hora de considerar todos los aspectos de calidad que presenta el software. Por su parte,

El modelo QSAT incluye una auto comparación con los modelos ISO/IEC, siendo los modelos internacionales considerados en la certificación de software. QSAT fue desarrollado con la meta de subsanar varios de los problemas existentes.

3. El modelo de calidad de productos software QSAT

3.1. Estructura del modelo

El modelo de Calidad QSAT tiene el mismo tipo de estructura que la norma ISO/IEC 9126-1 [6]. Para cada Contexto se definen Características Básicas (CB) de calidad del producto software. Una CB tiene a su vez Sub-características (SubC) que la definen, de manera de que puede calcularse su grado de calidad a partir del grado de calidad de sus SubC, ponderadas por la importancia relativa (IR) que le han dado los stakeholders en el contexto de la evaluación. Las SubC a su vez pueden medirse mediante determinados Atributos (Atr), utilizando Métricas (Métr). Este modelo, a diferencia de ISO/IEC, incluye uno o más niveles intermedios correspondientes a Sub-sub-característica (SubSubC). Asimismo el nivel de abstracción de una CB está dado, conceptualmente, por el principio que establece que una CB no puede ser derivada de otra CB ni Sub-C. La estructura del modelo se representa en la Figura 1.

El Contexto representa el ámbito de aplicación de una Característica determinada. El modelo respeta la definición de CB de ISO 8402, donde una CB es “una característica de una entidad que le confiere aptitud para satisfacer las necesidades establecidas o implícitas”[11]. Un “Atributo” de una SubC es una propiedad que posee el producto de software, que es evaluada usando métricas bien definidas, entendiendo que la medición del Atributo es necesaria para calcular el Grado o nivel de calidad de una SubC o SubsubC.

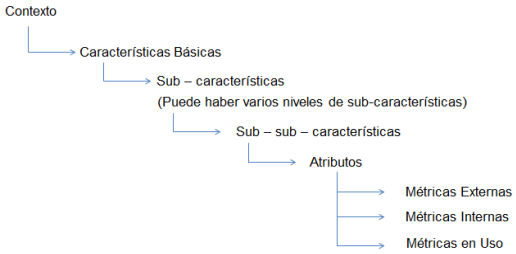


Figura 1

En resumen, el modelo presenta 6 contextos (*Ser usado, Ser probado, Ser instalado, En su interacción con el entorno, Ser Desarrollado, Ser adaptado*), 17 Características Básicas (*Adaptabilidad, Calidad de los artefactos, Facilidad de mantenimiento, Estandarización, Facilidad de Instalación, Capacidad para ser probado, Constancia, Correctitud, Cumplimiento Legal, Efectividad, Eficiencia, Manejo de fallas, Satisfacción de los stakeholders que no son usuarios, Satisfacción subjetiva de los usuarios, Usabilidad (objetiva), Seguridad de no causar daños, Seguridad Informática*), y 71 SubC.

3.2. Características Básicas consideradas

A continuación se definen las Características Básicas del modelo, en función de sus contextos (ámbitos de aplicación).

Contexto: Ser adaptado (1 CB)

Adaptabilidad: En qué medida el software se instala y funciona con el mismo o mayor grado de calidad que el pre-establecido, en diferentes ambientes y por diferentes actores, de modo que no sean necesarios recursos adicionales a los pre-establecidos para mantener su nivel de prestación.

Contexto: Ser Desarrollado (2 CB)

Calidad de los artefactos: En qué medida los artefactos poseen grados de calidad por sobre un nivel objetivo predeterminado.

Facilidad de Mantenimiento: En qué medida el software puede ser mantenido en su entorno, de modo que la implementación de cambios sólo requieran un esfuerzo menor a uno pre-determinado.

Contexto: En su interacción con el entorno (1 CB)

Estandarización: En qué medida el software tiene interfaces que se ajustan a estándares reconocidos, en el contexto de su uso por usuarios y otros actores, de modo que pueda ser usado en lugar de (o ser reemplazado por) otros sistemas que suministren funciones similares en el mismo dominio, sin que se necesite invertir recursos por encima de los pre-establecidos.

Contexto: Ser instalado (1CB)

Facilidad de instalación: En qué medida el software se instala y re-instala, en el contexto de la primera instalación y la instalación de nuevas revisiones y versiones, de modo que no son necesarios recursos adicionales a los pre-establecidos.

Contexto: Ser probado (1 CB)

Capacidad de ser probado: En qué medida el software puede ser probado, en el contexto de su uso, de modo que el grado de calidad del proceso de testing no esté por debajo de un grado determinado sin necesidad de recursos que excedan niveles determinados.

Contexto: Ser usado (11 CB)

Constancia: En qué medida el software no degrada la calidad de sus funciones con el aumento de usuarios u otros actores.

Correctitud: En qué medida el software produce datos y procesos válidos de modo que la utilización posterior de esos datos y de los resultados de los procesos sea correcta.

Cumplimiento legal: En qué medida el software cumple con todos los requisitos legales, en el contexto de exigencias legales vigentes aplicables al sistema.

Efectividad: En qué medida el software suministra todas (o el máximo posible) las funcionalidades, en el contexto de las necesidades de todos los stakeholders.

Eficiencia: En qué medida el software realiza sus funciones usando cantidades de recursos iguales o menores a los pre-establecidos.

Manejo de fallas: En qué medida el software mantiene un nivel de funcionamiento predeterminado de modo que el daño causado por una falla no supere un límite pre-determinado.

Satisfacción de los stakeholders que no son usuarios: En qué medida el software se desempeña, en el contexto del uso del Sistema, de modo que los stakeholders que no son usuarios tienen una subjetiva percepción de satisfacción del comportamiento del sistema.

Satisfacción subjetiva de los usuarios: En qué medida el software hace sentir confortable al usuario, en el contexto de una nueva versión, de modo que los usuarios tienen una subjetiva percepción de satisfacción por el uso del sistema.

Seguridad de no causar daños: En qué medida el software no causa daños indirectos, en el contexto del uso del Sistema, de modo que no cause cambios de estado en el entorno del sistema por los cuales una o más personas se vean negativamente afectadas.

Seguridad Informática: En qué medida el software no permite el acceso a los datos que manipula ni a las funciones que ejecuta, en el contexto de su uso diario y repetitivo, de modo que no se puedan acceder a Datos o Funciones No Autorizados.

Usabilidad (objetiva): En qué medida el software posibilita a los usuarios desempeñar su rol con una eficiencia mayor a una pre-establecida.

3.3. Sub-Características consideradas

Como ejemplo de las 71 SubC consideradas en el modelo se presentan las utilizadas en el Caso de Estudio que se informa más adelante. Las CB consideradas en el Caso de Estudio fueron *Facilidad de Instalación* y *Satisfacción de los Stakeholders que no son Usuarios*.

CB Facilidad de Instalación

SubC Primera instalación: En qué medida se instala el sistema en el contexto de la primera instalación de modo que no son necesarios recursos adicionales a los pre-establecidos.

SubC Upgrades: En qué medida se re-instala el Sistema en el contexto de la instalación de nuevas versiones de modo que no son necesarios recursos adicionales a los pre-establecidos.

CB Satisfacción de los stakeholders que no son usuarios

SubC Considerado útil: En qué medida los stakeholders que no son usuarios consideran útiles las funciones del Sistema en el contexto de uso del Sistema de modo que perciben un valor en su utilización.

SubC Considerado apropiado: En qué medida los stakeholders que no son usuarios consideran apropiadas las funciones del Sistema en el contexto de los problemas a los cuales esperan que dé solución.

SubC Considerado confiable: En qué medida los stakeholders que no son usuarios consideran que el sistema da información correcta, en el lenguaje adecuado, en el contexto de su utilización, de modo que la información esté en un tiempo menor o igual al pre-establecido.

3.4. Atributos y Métricas del Caso de estudio

Con el mismo criterio de selección seguido con las SubC a continuación se presentan los algunos atributos

(Atr) y Métricas (Metr) que se utilizarán en el caso de estudio.

CB Facilidad de instalación/Primera Instalación

Atributo: **Esfuerzo necesario para su primer instalación en el entorno de testing**

Métrica:

a. Establecer el Tiempo Óptimo para la primer instalación en entorno de testing: TO1
b. Medir el Tiempo Transcurrido en la primer instalación en entorno de testing: TT1
c. M = Abs[((TT1- TO1) / TO1) – 1]; si (M>=1) R=1 sino R=M;

CB Facilidad de Instalación/SubC Upgrades

Atributo: **Esfuerzo necesario para la instalación de sus versiones en el entorno de testing**

Métrica:

1. Establecer el Tiempo Óptimo para la instalación de versiones en entorno de testing: TOv
2. Medir el Tiempo Transcurrido durante la instalación de versiones en entorno de testing: NHv
3. M = Abs[((NHv - TOv) / TOv) **–** 1]; si (M>=1) R=1 sino R=M;

CB Satisfacción de los stakeholders que no son usuarios/SubC Considerado útil

Atributo: **Porcentaje de Cobertura de Funciones Útiles**

Métrica:

Con un CUESTIONARIO conciliar la siguiente información

1. Sea NS (número de stakeholders)
2. Para cada módulo X
   1.1. Para suma=0 e i=1 hasta NS, de 1 en 1, hacer
      1.1.1. Obtener CFRi (Cobertura de Funciones Útiles para el Modulo X, según la opinión del StakeHolders “i”. Se mide entre 0 y 1).
      1.1.2. suma+=CFRi
   Fin
3. Valoración del módulo X = Suma/NS.

CB Satisfacción de los stakeholders que no son usuarios/Sub C Considerado apropiado

Atributo: **Porcentaje de Funciones inútiles**

Métrica: Con un *cuestionario* conciliar la siguiente información

1. Sea NS (número de stakeholders)
2. Para cada módulo X
   1.1. Para suma=0 e i=1 hasta NS, de 1 en 1, hacer
      1.1.1. Obtener PFIi (Proporción de Funciones inútiles para el Modulo X, según la opinión del Stakeholders “i”. Se mide entre 0 y 1).
      1.1.2. suma+=PFIi
   Fin
3. Valoración del módulo X = Suma/NS.

CB Satisfacción de los stakeholders que no son usuarios/SubC Considerado confiable

Atributo: **Grado de Confiabilidad percibida en Uso**

Métrica: Con un CUESTIONARIO conciliar la siguiente información

1. Para cada módulo X
   1.1. NS (número de stakeholders)
   1.2. Para sumG=0 e i=1 hasta NS, de 1 en 1, hacer
      1.1.1. Obtener el GCIi(Grado de confiabilidad del módulo i medida entre 0 y 1)
      1.1.2. SumG+=GCIi
   Fin
2. Valoración del módulo X = SumG/NS.

3.5. Principios del modelo

El diseño del modelo QSAT siguió una serie de principios que glosamos a continuación.

Principio 1. Una Característica se considera CB si cumple dos condiciones:

a. No pude derivarse de otra CB o SubC
b. En un dado contexto, evaluar la medida de cumplimiento de una condición es evaluar la CB

Por ejemplo, el modelo define a la CB *Satisfacción de los stakeholders que no son usuarios*, como ´En qué medida se desempeña el Sistema, en el contexto de “*ser usado”*, de modo que los *stakeholders* que no son usuarios tienen una percepción de satisfacción del comportamiento del sistema. Esto es evaluable en términos de opiniones subjetivas de los *stakeholders* que no son usuarios´.

Principio 2. Toda Característica cumple que evaluarla requiere evaluar las de nivel inmediato inferior.

Por ejemplo, evaluar la CB *Satisfacción de los stakeholders que no son usuarios*, requiere evaluar, entre otras, la SubCB es *Considerado útil*.

Principio 3. La elección de un Atributo para una SubC se justifica si se puede evaluar la SubC evaluando el Atributo. Por ejemplo, dada la CB *Satisfacción de los stakeholders que no son usuarios*, y la SubC *Considerado útil*, un Atr es el *Porcentaje de Cobertura de Funciones Útiles*.

Principio 4. La elección de las Métricas para evaluar un Atributo se justifica si aquello que miden las métricas contribuye a evaluar al Atributo.

4. Caso de Estudio

A la fecha se han realizado tres proyectos para validar la utilidad del Framework MyFEPS y su modelo de calidad de producto *software* QSAT [12], con distintos niveles de complejidad, y con diferentes objetivos de evaluación, que dependen del contexto de la evaluación y de la necesidad de los *stakeholders*. A continuación se describe brevemente uno de ellos. La evaluación de un Sistema de Gestión de Historias Clínicas (SGHC). Se trata de un aplicativo Cliente/Servidor de Gestión de Historias Clínicas y Facturación, de gran importancia para la empresa, porque le permite gestionar servicios de sus clientes, principalmente ARTs, como ser: servicios de call center, alta de denuncias, seguimientos de los casos, traslados de pacientes, cirugías, etc.

El objetivo de este proyecto de evaluación definido abarca la evaluación de dos características básicas del modelo MyFEPS: *Facilidad de instalación* y *Satisfacción de los stakeholders que no son usuarios*. El objetivo es consistente con los problemas que enfrentaba el Dpto. de Sistemas, el contenedor del software había sido construido en una versión que ya la empresa proveedora no mantenía, lo que dificultaba la instalación y re instalación del software. Asimismo, dada la importancia de este nuevo software para la Empresa, se buscaba medir el nivel de satisfacción de accionistas y otros stakeholders que no son usuarios. Se definieron cinco stakeholders no usuarios, que cumplen cuatro roles: Socio de la Empresa, Director de la Clínica, Gerente de Sistemas y Médico. A su vez los recursos humanos que llevaron adelante el proyecto de evaluación tienen cuatro roles: Analista de QA, Ingeniero de Software, Project Manager y Tester. La duración del proyecto fue de 1 mes.

El proceso de evaluación que acompaña a este modelo fue tomado principalmente de la norma ISO/IEC 25040 [13], que ha reemplazado a la ISO/IEC 14598-1. Se han respectado las 5 actividades principales: 1 Establecer los requisitos de la evaluación, 2 Especificar la evaluación, 3. Diseñar la evaluación, 4. Realizar la Evaluación, y 5 Concluir la Evaluación, pero se han agregado tareas necesarias para registrar la Importancia Relativa (IR) que tienen los Ítems de Calidad para los stakeholders, y tareas para ponderar las Metr, Atr, SubSubC, SubC, y CB. En las dos primeras etapas se definió el objetivo de evaluación ya citado, el modelo de calidad a emplear (QSAT), y las CB, SubC, Atr a evaluar, y las Metr a medir. Se definió la IR de cada Ítem de Calidad, y el nivel de rigurosidad de las mediciones. Luego se definió el plan de evaluación cuantitativo, y se definieron la infraestructura de evaluación, y los casos de prueba. Se ejecutaron las pruebas y calcularon las métricas.

4.1 Ejemplo de Pruebas:

CB Instalabilidad

Atributo: **Esfuerzo en horas-hombre necesarios para su primer instalación en el entorno de testing**

El resultado de la prueba fue de **0,57.**

CB Facilidad de Instalación/SubC Upgrades

Atributo: **Esfuerzo en horas-hombre necesarios para la instalación de sus versiones en el entorno de testing**

1. Establecer el óptimo número de horas hombre para la instalación de versiones en entorno de testing: ON = 90s
2. Medir el óptimo número de horas hombre para la instalación de versiones en entorno de testing: Primero se llevó a cabo la medición de la métrica por cada caso de prueba y después se hizo el promedio para obtener un solo resultado M.

Caso 1: "Upgrades – Despliegue Ok."
M1 = 0,42
Caso 2: "Upgrades – Despliegue Ok. Error configuración BD."
M2 = 0,67
Caso 3: "Upgrades – Despliegue Ok. Error configuración base de usuarios."
M3 = 0,66

Mprom. = M que se usará para calcular G. Resultado promedio de todos los M obtenidos.
Mprom. = (M1 + M2 + M3) / 3
Mprom. = (0,42+0,67+0,66) / 3
Mprom. = **0,583**

4.2- Satisfacción de los stakeholders que no son usuarios

Por cada SubC se calcularon las Metr en base a los cuestionarios de Satisfacción que respondieron todos los stakeholders no usuarios, dando por resultado:

Sub-característica "Considerado Útil":
Sea NS (número de stakeholders) = 5
Para cada módulo: **Alta Historia Clínica**
Para suma=0 e i=1 hasta NS=5, de 1 en 1, hacer
CFRi1 – Stakeholder1: 0,82
CFRi1 – Stakeholder2: 0,8
CFRi1 – Stakeholder3: 0,7
CFRi1 – Stakeholder4: 0,86
CFRi1 – Stakeholder5: 0,8
suma=+=CFRi1
suma= 0,82 + 0,8 + 0,7 + 0,86 + 0,8 = 3,98

Sea NS (número de stakeholders) = 5
Para cada módulo: **Facturación**
Para suma=0 e i=1 hasta NS=5, de 1 en 1, hacer
CFRi2 – Stakeholder1: 1
CFRi2 – Stakeholder2: 0,92
CFRi2 – Stakeholder3: 0,87
CFRi2 – Stakeholder4: 0,83
CFRi2 – Stakeholder5: 0,92
suma= 1 + 0,92 + 0,87 + 0,83 + 0,92 = 4,54

CFRi = (CFRi1 + CFRi2) / 2 = (3,98 + 4,54) / 2 = 8,52 / 2 = 4,26
1.1. fin

Valoración del módulo X = 4,26 / 5 = **0,85**

Sub-característica "Considerado Apropiado":
Sea NS (número de stakeholders) = 5
Para cada módulo: **Alta Historia Clínica**
Para suma=0 e i=1 hasta NS=5, de 1 en 1, hacer
CFRi1– Stakeholder1: 0,9
CFRi1 – Stakeholder2: 0,8
CFRi1 – Stakeholder3: 0,9
CFRi1 – Stakeholder4: 1
CFRi1 – Stakeholder5: 0,8
suma= 0,9 + 0,8 + 0,9 + 1 + 0,8 = 4,40

Para cada módulo: **Facturación**
Para suma=0 e i=1 hasta NS=5, de 1 en 1, hacer
CFRi2 – Stakeholder1: 0,67
CFRi2 – Stakeholder2: 0,5
CFRi2 – Stakeholder3: 0,83
CFRi2 – Stakeholder4: 1
CFRi2 – Stakeholder5: 0,58
suma= 0,67 + 0,5 + 0,83 + 1 + 0,58 = 3,58

CFRi = (4,40 + 3,58) / 2 = 7,98 / 2 = 3,99
fin

Valoración del módulo X = 3,99 / 5 = **0,80**

Sub-característica "Considerado Confiable":
Sea NS (número de stakeholders) = 5
Para cada módulo **Alta Historia Clínica**
Para suma=0 e i=1 hasta NS=5, de 1 en 1, hacer
CFRi1– Stakeholder1: 1
CFRi1 – Stakeholder2: 1
CFRi1 – Stakeholder3: 1
CFRi1 – Stakeholder4: 1
CFRi1 – Stakeholder5: 1
suma=1 + 1 +1 + 1 + 1 = 5

CFRi2 – Stakeholder1: 1
CFRi2 – Stakeholder2: 1
CFRi2 – Stakeholder3: 0,88
CFRi2 – Stakeholder4: 1
CFRi2 – Stakeholder5: 1
suma= 1 + 1 + 0,88 + 1 + 1 = 4,88

CFRi = (CFRi1 + CFRi2) / 2 = (5 + 4,88) / 2 = 9,88 / 2 = 4,94
fin
Valoración del módulo X = 4,94 / 5 = **0,99**

Tabla 2: Resultados obtenidos para SubC de CB Satisfacción de stakeholders no usuarios

Resultados obtenidos de los Cuestionarios de Satisfacción	Stakeholders
12 Satisfacción de los stakeholders que no son usuarios	Media
12.1. Considerado Útil	0,85
12.2 Considerado Apropiado	0,80
12.3 Considerado Confiable	0,99

4.3- Obtener Grado de Calidad

Concluidas las mediciones se ponderan los resultados considerando los pesos de Ponderación de cada ÍC, relevados a partir de cuestionarios con preguntas cerradas, cuyas alternativas de respuesta son: IC considerado *Muy Importante, Importante, Medianamente Importante, Algo Importante y Totalmente sin importancia*, donde *Muy Importante* se evalúa con 1, *Totalmente sin importancia* se evalúa con 0, y las respuestas intermedias son de 0,75, 0,50 y 0,25 respectivamente, como lo muestra la Tabla 3.

Tabla 3: Cuestionario para establecer la IR de cada IC

Encuesta a Stakeholder1 para establecer la IR de los IC	Muy importante	Importante	Importa Mediana-mente	Algo importante	Totalmente sin importancia	Puntuacion
9 Instalabilidad		x				0,75
9.1 Primera Instalación		x				0,75
9.1.1.B Esfuerzo necesario para su primer instalación en el entorno de testing		x				0,75
9.2 Upgrades	x					1
9.2.1.B Esfuerzo necesario para la instalación de sus versiones en el entorno de testing	x					1
12 Satisfacción de los stakeholders que no son usuarios		x				0,75
12.1 Considerado Útil		x				0,75
12.1.1 Porcentaje de Cobertura de Funciones Útiles		x				0,75
12.2 Considerado Apropiado	x					1
12.2.1 Porcentaje de Cobertura de Funciones Inútiles	x					1
12.3 Considerado Confiable	x					1
12.3.1 Grado de Confiabilidad percibida en Uso	x					1

Una vez que todos los stakeholders responden estos cuestionarios, se calcula la IR como el promedio de las IR que han asignado los staholders (a no ser que en un proyecto se considere que algún stakeholder tiene mas Peso de importancia que el resto, a la hora de establacer las IR de los IC). En este proyecto se utilizó el promedio, como se ve en la Tabla4.

Tabla 4: Pesos obtenidos para Ponderar los Ítems de Calidad

Peso de Ponderación para cada IC	Puntuacion Total asignada por Stakeholders	Media de Stakeholders
9 Instalabilidad	2	0,4
9.1 Primera Instalación	2	0,4
9.1.1.B Esfuerzo en horas-hombre necesarios para su primer instalación en el entorno de testing	2	0,4
9.2 Upgrades	2,5	0,5
9.2.1.B Esfuerzo en horas-hombre necesarios para la instalación de sus versiones en el entorno de testing	2,5	0,5
12 Satisfacción de los stakeholders que no son usuarios	3	0,6
12.1 Considerado Útil	4	0,8
12.1.1 Porcentaje de Cobertura de Funciones Útiles	4	0,8
12.2 Considerado Apropiado	3,75	0,75
12.2.1 Porcentaje de Ausencia de Funciones Inútiles	4	0,8
12.3 Considerado Confiable	4,5	0,9
12.3.1 Grado de Confiabilidad percibida en Uso	4,5	0,9

En la Tabla4, la columna "Media de Stakeholders" es el promedio de la IR asignada a cada IC por todos los Stakeholders. Este valor se toma como "Peso" en Tabla5, que es donde se calcula el Grado de Calidad del Producto en Evaluación (G). Por ejemplo, *Instalabilidad* tiene una IR de 0,40 y *Satisfacción de los stakeholders que no son usuarios* tiene una IR de 0,60. Asimismo aparecen las IR a nivel de SubC y de Atr.

Tabla 5: Cálculo del Grado de Calidad del Producto

CB	Sub CB	Sub-sub CB	Descripcion Atributo	GCAT: Grado de Calidad del Atributo	Fidelidad de la Medición (Rigurosidad)	PAT: Peso Atributo	GCSC: Grado de Calidad de la SC	PSC: peso SC	GCCB: Grado de Calidad de la CB	PCB: peso CB	Grado de Calidad del Producto
											0,76
Instalabilidad									0,57	0,40	
	1. Primera Instalación						0,56	0,40			
			Esfuerzo en horas-hombre necesarios para su primer instalación en el entorno de testing	0,56	0,40	0,40	=REDONDEAR (SUMA(J8*G8)/ SUMA(J8:J8);2)		=SUMA(K7*J7+ K9*J9) / SUMA(K7:K10)		=REDONDEAR(SUMA (L6*M6+L11*M11) / SUMA(M6;M11); 2)
	2. Upgrades						0,58	0,50			
			Esfuerzo en horas-hombre necesarios para la instalación de sus versiones en el entorno de testing	0,58	0,40	0,50					
Satisfacción de los stakeholders que no son usuarios									0,89	0,60	
	1. Considerado Útil						0,85	0,80			
			Porcentaje de Cobertura de Funciones Útiles	0,85	0,60	0,80					
	2. Considerado Apropiado						0,80	0,75			
			Porcentaje de Funciones inútiles	0,80	0,60	0,80					
	3. Considerado Confiable						0,99	0,90			
			Grado de Confiabilidad percibida en Uso	0,99	0,60	0,90					

En Tabla 5, las Filas representan los Ítems de Calidad (CB, SubC y Atr) medidos. Las columnas son: Peso C (Peso ponderado de la CB), Peso SC (Peso ponderado de la SubC), Peso Atributo (Peso ponderado del Atr), Fidelidad de la medición Rig, GCAT (grado de calidad del Atr: el resultado obtenido de medir el Atr), Grado de calidad de la SubCB, Grado de calidad de la CB y por último, el Grado de calidad del Producto. En este caso. el resultado obtenido es de **0,76 /1**.

Observando la tabla, a simple vista se puede apreciar que los stakeholders no usuarios están satisfechos con el producto evaluado, pero que sin duda requiere una mejora referida a su propiedad para ser instalado, tanto la primera vez, como en cada upgrade.

5. Conclusión

Se ha presentado el Modelo de Calidad de Productos de Software QSAT con sus características y principios, se enunciaron los *contextos* considerados y se definieron las *características básicas.* A modo de ejemplo se analizaron algunas *sub-características*, *atributos* y *métricas*. Se presentaron los principales modelos de calidad del software (Tabla 1) considerando algunas variables. Esas variables para el modelo QSAT, toman los siguientes valores:

Entornos: desarrollado, probado, instalado, adaptado, usado, interacción con el medio ambiente

CB: 18

QSAT: 71

Descripción del Proceso de evaluación: si

Complejidad: media

Pero cabe agregar que ninguno de los casos considerados tiene una *Descripción procedimental de métricas* y tampoco establece la *Importancia relativa de ítems de calidad*, como es el caso de QSAT.

Se ha podido definir un modelo que resuelve restricciones y falencias de los modelos conocidos. La validación del modelo, esto es su aplicación al establecimiento de la calidad del producto software resulta especialmente arduo. Hasta la fecha se han desarrollado tres casos de estudio y en este artículo se informa del desarrollo exitoso de uno. El concepto del modelo y los entornos en consideración en el caso (con las correspondientes CB, SubC, atributos, métricas y procedimientos) han demostrado su aplicabilidad y utilidad en el mundo real.

Los trabajos futuros se orientan a continuar desarrollando casos de estudio y elaborar conclusiones generales del agregado de los casos

6. Referencias

[1] Sorgen, A., Angeleri, P., "El Modelo de Evaluación del Proyecto MyFEPS", *40JAIIO - ASSE 2011* - ISSN: 1850-2792, 2011, pp. 180-191.

[2] International Standardization Organization (ISO), http://www.iso.org, 27/04/2014.

[3] International Electro technical Commission (IEC), http://www.iec.ch, 27/04/2014.

[4] Ley 25.922 Ley de Promoción de la Industria del Software, http://www.infojus.gov.ar/legislacion/ley-nacional-25922-ley_promocion_industria_software.htm?1, 27/04/2014.

[5] Ley N° 26.692 Ley de Promoción de la Industria del Software, http://www.industria.gob.ar/lps/, 27/04/2014.

[6] IRAM-NM-ISO/IEC 9126-Tecnología de la información-Ingeniería de software - Calidad del producto Parte 1 - Modelo

de calidad (ISO/IEC 9126-1:2001, IDT) Information technology - Software engineering - Product quality Part 1 - Quality Model 1, IRAM, 2009.

[7] ISO/IEC 25010. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, ISO, 2011.

[8] ISO/IEC JTC1/SC7/WG6 Evaluation and metrics Working Group. Grupo de trabajo 6 del Subcomité 7 del Comité Técnico Conjunto ISO/ IEC JTC1 Information Technology es el que trabajó en la revisión de las normas de evaluación de software y sistemas, a cargo del estudio de la serie de normas ISO/IEC 25000,http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45086, 27/04/2014.

[9] ISO Standards catalogue, Standards and projects under the direct responsibility of ISO/IEC JTC1/SC7 Secretariat, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45086&development=on, 27/04/2014.

[10] Pressman, Roger S., "Ingeniería de software: un enfoque práctico", McGrawhill, 1998.

[11] ISO 8402, Quality management and quality assurance -- Vocabulary, *ISO*,1994.

[12] Rolando Titiosky, Paula M. Angeleri, Amos Sorgen, Jaquelina Wuille Bille: "Proyectos de Evaluación de Productos de Software con un nuevo Framework de Calidad", *XV Workshop de Investigadores en Ciencias de la Computación 2013*, ISBN: 9789872817961, http://sedici.unlp.edu.ar/bitstream/handle/10915/27225/Documento_completo.pdf?sequence=1

[13] Boehm, B.W., Brown, J.R., Lipow, M., Macleod, G.J., Merritt, M.J.; "Characteristics of Software Quality", *North-Holland*, 1978.

[14] Cavano, J.P., McCall, J.A., "A Framework for the Measurement of Software Quality", *Proc. of the ACM Software Quality Assurance Workshop*, pp. 133-139, Nov. 1978.

[15] ISO/IEC 9126-2 Information technology - Software engineering - Product quality - Part 2: External metrics, *ISO*, 2003.

[16] Marc-Alexis Côté, Witold Suryn, Elli Georgiadou, "Software Quality Model Requirements for Software Quality Engineering", http://profs.etsmtl.ca/wsuryn/research/SQE-Publ/ Quality%20model_requirements.%20SQM 2006.pdf.

[17] ISO/IEC 25040. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation Process, *ISO*, 2011

[18] ISO/IEC 14598-1 Information technology -- Software product evaluation -- Part 1: General overview, *ISO*, 1998.

[19] Ventura, Agustín, "Descriptivo del Proyecto Piloto Sistema de Gestión de Clínicas", *documento de trabajo, Descriptivo del Proyecto.PILOTO SGC.docx, Proyecto MyFEPS, Universidad de Belgrano*, 06/03/2013.

[20] Ventura, Agustín, "Proceso de evaluación del Sistema de Gestión de Clínicas", *documento de trabajo, E.2.Proceso de evaluación de productos de software.PILOTO SGC.docx, Proyecto MyFEPS, Universidad de Belgrano*, 07/07/2013.

[21] Ventura, Agustín, "Aplicación del Framework MyFEPS para Evaluar la Calidad de un Sistema de Gestión de Clínicas (SGC)", *tesina de Universidad de Belgrano*, 15/05/2014.

Una Estrategia Orientada a la Comprensión de Programas para Extracción y Análisis de Información Estática en Sistemas OO

Miranda Enrique A.,Berón Mario M., Riesco Daniel E.
Universidad Nacional de San Luis
Departamento de Informática
{eamiranda,mberon,driesco}@unsl.edu.ar

Resumen

Sin lugar a dudas, una de las tareas más complejas y que más tiempo consume en el ciclo de vida de una aplicación es la de Mantenimiento. En este entorno, las actividades que más tiempo consumen son aquellas que debe ejecutar el programador para lograr un completo entendimiento del sistema; a partir de esta necesidad, es que surge una disciplina de la Ingeniería de Software denominada Comprensión de Programas (CP). En este contexto, muchos autores afirman que un programador efectivamente comprende un programa cuando puede relacionar el Dominio del Problema con el Dominio del Programa. El primero hace referencia a la salida del sistema en estudio. El segundo se relaciona con los componentes del sistema utilizados para producir dicha salida. En este artículo se presenta SSPIA (Strategy for Static Program Information Analysis), una estrategia que asiste al programador durante el proceso de Comprensión de Programas de sistemas Orientados a Objetos (OO). Dicha estrategia realiza lo siguiente : i) extrae el Grafo Estático de Llamadas a Métodos (GELM), ii) aplica filtros que eliminan métodos, muy relacionados con detalles de implementación, del GELM y iii) posibilita el análisis del GELM reducido con el propósito de inferir funcionalidades. De esta manera la estrategia planteada pretende asistir al arduo proceso cognitivo que implica comprender un sistema.

1. Introducción

Sin lugar a dudas, una de las tareas más complejas y que más tiempo consume en el ciclo de vida de una aplicación es la de Mantenimiento y Evolución de Software (MES) [3, 24]. Dentro del entorno de MES, las tareas que más tiempo consumen son aquellas que debe ejecutar el programador para lograr un completo entendimiento del sistema [7, 26]. A partir de la necesidad de asistir al arduo proceso de comprensión mencionado anteriormente, es que surge una disciplina de la Ingeniería de Software denominada Comprensión de Programas (CP). La CP se presenta como un área de investigación útil e interesante para impulsar el trabajo de MES a través de técnicas y herramientas que ayuden al ingeniero de software en el análisis y la comprensión de sistemas. A través de extensos estudios en Modelos Cognitivos y Teorías del Aprendizaje[11, 20, 31], ciertos autores afirman que el principal desafío en esta área consiste en relacionar el Domino del Problema con el Dominio del Programa [4, 19, 26]. El primero hace referencia a la salida del sistema. El segundo a las componentes de software usadas para producir dicha salida [4, 13, 29]. La reconstrucción de este tipo de relación es muy compleja ya que implica (ver figura 1):

- Construir una o varias representaciones para el Dominio del Problema.

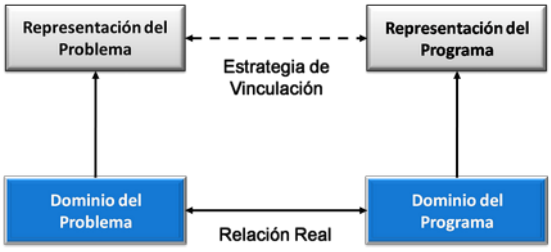


Figura 1. Modelo de Comprensión de Programas

- Construir una o varias representaciones del Dominio del Programa.
- Elaborar un procedimiento de vinculación.

Los tres pasos antes mencionados son de extrema importancia debido a que la integración de los mismos resulta en la elaboración de "verdaderas" estrategias de comprensión [4].

Para crear cada una de las representaciones mencionadas anteriormente, primero es necesario extraer información de ambos dominios. Por esta razón, se utilizan técnicas que permiten obtener distintos tipos de datos relacionados con determinados aspectos de la estructura y el comportamiento del programa. Dichas técnicas se conocen en el contexto de CP como Técnicas de Extracción de Información (TEI). Dentro de lo que a extracción de información del programa se refiere, existen muchas técnicas las cuales pueden ser clasificadas en base al tipo de información que extraen en: Estáticas y Dinámicas. Las TEI estática indagan en el código fuente sin ejecutar el sistema [14, 15, 25]. Las TEI dinámica están relacionadas con información de tiempo de ejecución [2, 12, 15]. Estas últimas acotan el espacio de búsqueda, ya que en cada ejecución no se ven implicadas todas las secciones del código. Es importante mencionar que ambos tipos de técnicas son muy útiles para la elaboración de estrategias de comprensión consistentes y eficaces [4].

En este trabajo se presenta SSPIA (*Strategy for Static Program Information Analysis*), una estrategia que propone asistir al programador durante el proceso de Comprensión de Programas mediante la extracción y posterior análisis de información estática, con el objetivo de detectar las principales funcionalidades de un sistema OO.

El artículo está organizado de la siguiente manera: en la sección 2 se introduce una de las estructuras más utilizadas en las TEI estática, en la cual se basa la estrategia propuesta, dicha estructura es el Grafo Estático de Llamada a Métodos. En la sección 3 se explican las tareas que se llevan a cabo en SSPIA para filtrar información poco relevante respecto de la lógica del sistema bajo estudio. En la sección 4 se muestra una aproximación para asistir al ingeniero de software a detectar funcionalidades de manera estática. En la sección 5 se presenta un caso de estudio de una herramienta para medición y evaluación denominada NESSy. Finalmente, en la sección 6 se exponen las conclusiones y los trabajos futuros.

2. Análisis Estático de Sistemas

Las TEI estáticas proveen información valiosa de la *estructura de los programas*[1], analizando artefactos de software relativos al código fuente del programa, sin llevar a cabo ejecuciones de los mismos. Por este motivo han sido utilizada con diferentes propósitos en distintas investigaciones [6, 17, 25].

Una clase de lenguajes muy utilizados son los que posibilitan el desarrollo OO (Orientado a Objetos). Su popularidad se debe a la facilidad con la que se puede interpretar su estructura, las características y/o restricciones que provee el encapsulamiento como también las ventajas que presentan en todas las etapas de desarrollo de software [8, 21].

Una de las herramientas más utilizadas en el contexto del análisis estático es el Grafo de Llamadas a Funciones (GLF) [27]. En el caso de los sistemas Orientados a Objetos, es el Grafo Estático de LLamada a Métodos (GELM) [22].

[1] Es decir, como están relacionadas los componentes de software que constituyen el sistema.

2.1. Grafo Estático de Llamada a Métodos

Teniendo en cuenta lo explicado en el apartado precedente, es posible hacer uso del GELM para inferir y analizar las distintas funcionalidades que presenta el sistema.

Definición 1: un Grafo Estático de Llamada a Método (GELM) es una tupla G =$< M, I >$ donde M es el conjunto de nodos que representan los métodos del sistema $M = M_1, M_2, M_3, ...$ e I es una relación binaria definida en MxM donde una tupla (M_i, M_j) representa una llamada estática[2] desde el método M_i al método M_j.

Con el propósito de ilustrar el grafo previamente definido, en la figura 2 se muestra un programa escrito en Java. En la figura 3 se muestra el Grafo Estático de Llamada a Métodos para dicho programa. Desde un punto de vista estático, es posible inferir qué métodos invocados *determinan* el comportamiento del método que los invoca; en otras palabras, los métodos mantienen fuerte relación entre el *llamador* y el *llamado*. Por otra parte, teniendo en cuenta todo lo expuesto en este apartado, es necesario resaltar un aspecto fundamental que concierne a esta temática: el volumen de información extraída o en este contexto, el tamaño del GELM.

3. Síntesis de Dependencias

Debido al tamaño que puede llegar a alcanzar el GELM, es necesario contar con métodos de reducción de información [5, 28]. Como es posible inferir, dichos métodos deben reducir considerablemente la pérdida de artefactos de software sensibles a la lógica del programa. Para esto, es necesario aplicar técnicas que permitan discernir entre artefactos que son relevantes respecto de la lógica subyacente del sistema y los que no lo son. En SSPIA las tareas de filtrado se dividen en dos fases, de acuerdo a la esencia de la información filtrada. En los siguientes apartados se describen cada una de estas.

[2]En esta investigación se considera una llamada estática cuando figura en el código fuente.

```
public class X {
        // declaracion de atributos de clase
        public void X(){
                  // inicializacion
                  }
        public void a() {
            b();
            }
        public void b() {
            d();
            c();        }
        public void c() {
            // Se deja vacio a proposito
            }
        public void d() {
            e();
            }
        public void e() {
            f();
            g();
            }
        public void f() {
            // Se deja vacio a proposito
            }
        public void g() {
            // Se deja vacio a proposito
            }
        public void h() {
            i();
            }
        public void i() {
            d();
            }
    }
public class Y {
        // declaracion de atributos de clase
        public void Y(){
             // inicializacion
        }
        public void j() {
             X variableX = new X();
             variableX.d();
        }
    }
```

Figura 2. Ejemplo de Programa Java.

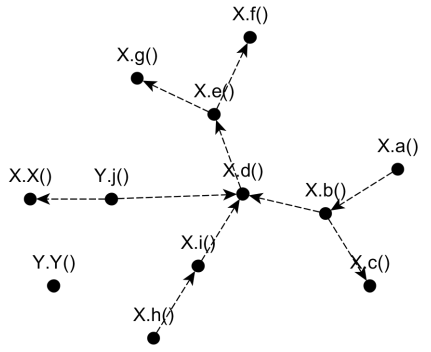


Figura 3. GELM para el código mostrado en la figura 2.

3.1. Primera Fase de Filtrado

Durante la primera fase, se filtran los métodos que por naturaleza son externos a la lógica subyacente del sistema. Teniendo en cuenta esto, los mismos se podrían remover sin afectar la comprensibilidad global del objeto de estudio, ya que son detalles de implementación de bajo nivel. A continuación se mencionan los elementos filtrados en esta instancia de la investigación:

Librerías nativas del lenguaje: por ejemplo, en el caso de Java: *java.lang*, *java.util*, *java.net*, entre otras tantas.

Librerías que el usuario defina como irrelevantes.

Métodos redundantes: teniendo en cuenta lenguajes OO, hasta el momento se filtran los siguientes métodos: a) constructores externos al código analizado; b) métodos destructores y c) métodos aislados de clases anónimos [9].

Métodos sobrecargados: se considera importante reducir los métodos sobrecargados que se llaman sucesivamente con distintos parámetros; por ejemplo, *getDatos(java.lang.String)* llama a *getDatos(java.util.Map)* que a su vez llama a *getDatos(java.lang.long)*. Notoriamente, este tipos de transitividades aisladas no aportan a la lógica del programa ya que, en el contexto del GELM, simplemente revelan detalles de implementación.

Esta fase requiere de muy poca intervención por parte del usuario, ya que la mayoría de los elementos que serán filtrados poseen altas probabilidades de ser externos al código o de no aportar información relevante respecto de la lógica del programa analizado.

3.2. Segunda Fase de Filtrado

Durante la segunda fase se llevan a cabo diferentes análisis con el objeto de filtrar métodos propios del sistema, pero que podrían ser ocultados sin afectar la comprensión de las representaciones; en este caso en particular, el GELM. Para detectar los elementos referenciados en el párrafo anterior se ha utilizado un enfoque similar al planteado por Hamou-Lhadj y Lethbridge [18] para sintetizar trazas de ejecución. En dicho trabajo, los autores proponen identificar elementos que no son importantes con respecto a la lógica del programa. A dichos elementos se los denomina *utilidades*. El razonamiento subyacente que aplican los autores para determinar las utilidades es el siguiente: *mientras más llamadas desde diferentes "lugares" posea un método, entonces es muy probable que haya sido diseñado con múltiples propósitos, por lo tanto, es muy probable que sea una utilidad.*

Un enfoque similar al expuesto en el párrafo precedente puede ser utilizado con el GELM, es decir, se eliminan los nodos que tienen altas probabilidades de ser utilidades. Para llevar a cabo este filtrado se tienen en cuenta un conjunto de métricas que, por medio de ciertos cómputos, permiten determinar el "grado de utilidad" para cada nodo (método). Hasta el momento se consideran las siguientes métricas: *fan-in*, *fan-out* e *intermediación* de cada nodo en el grafo. La primera indica la cantidad de métodos que invoca el método actual (grado de entrada); la segunda, indica la cantidad de métodos que son invocados por el método actual (grado de salida). La tercera es una medida de centralidad que cuantifica la frecuencia o el número de veces que un nodo actúa como un "puente" a lo largo del camino más corto entre otros dos nodos [10].

Las primeras dos métricas son utilizadas en la siguiente fórmula:

$$U(r) = \frac{FanIn(r)}{N} * \frac{Log(\frac{N}{\frac{FanOut(r)+1}{}})}{Log(N)}$$

donde *r* representa el método para el cual se realiza el cálculo y *N* es el número total de métodos en el grafo. Si $U(r) = 0$, entonces *r* no es una utilidad; si $U(r)$ tiende a 1 (de acuerdo al sistema analizado), muy posiblemente lo sea. Las operaciones realizadas en la fórmula que computa $U(r)$ son explicadas más detalladamente en [18].

Claramente, es necesario definir una cota que determine a partir de que valor obtenido, un método se considera como utilidad. Este valor lo debe

determinar el usuario y se debe establecer teniendo en cuenta distintos factores como el tamaño del proyecto, el tipo de aplicación, la proporción de información que se desea filtrar, entre otros. Sin embargo, es necesario brindar al usuario distintos mecanismos para la determinación de un valor aproximado. Una forma de aproximar un valor umbral es mediante el cálculo del promedio de los valores de utilidad de todos los métodos. De este modo, el valor estimado se calcula teniendo en cuenta los valores que se obtienen en base al sistema analizado y así el mismo está relacionado con el dominio atacado.

Por otra parte, la métrica *intermediación* es utilizada para filtrar posibles falsos positivos determinados mediante la fórmula $U(r)$; es decir, en el supuesto caso que un método se determine como utilidad, si su valor de intermediación (normalizado) supera un umbral determinado, entonces muy posiblemente dicho nodo no sea una utilidad. Para esta métrica, el usuario también debe especificar un valor como umbral para filtrar falsos positivos. De la misma forma que con la función de utilidad, es posible calcular el promedio de todos los valores de intermediación.

Finalmente se procede a eliminar los nodos con utilidades superiores al valor umbral, con el objeto de reducir la información que será representada. Este proceso de filtrado se realiza en cascada; es decir, si se elimina un nodo determinado, además se borran los arcos entrantes y salientes del mismo, como así también lo nodos descendientes.

3.3. Algoritmo de Filtrado

Con el objetivo de resumir los pasos que se llevan a cabo para filtrar información en SSPIA, en este apartado se muestra el algoritmo de filtrado descripto en las subsecciones precedentes. El mismo recibe como entrada un sistema OO, construye el GELM para el mismo, procesa esta información con el objetivo de filtrar detalles de implementación y finalmente retorna dicho grafo sintetizado. A continuación se muestran los pasos que lleva a cabo el algoritmo:
Paso 1: generar el GELM.
Paso 2: filtrar artefactos que por naturaleza son externos al sistema o detalles de implementación (primera fase de filtrado, apartado 3.1).
Paso 3: establecer los valores umbrales para a determinación de utilidades e intermediación.
Paso 4: realizar análisis *fan-in/fan-out* e *intermediación* (segunda fase de filtrado, sección 3.2).

Al finalizar, se visualiza el GELM resultante y la cantidad de nodos y arcos en el mismo. El usuario analiza los resultados y de acuerdo a su criterio, este puede ejecutar nuevamente el algoritmo especificando librerías externas a ser filtradas y/o valores umbrales diferentes, estimados de acuerdo al nivel de filtrado deseado.

Una vez finalizadas las tareas que permiten filtrar información irrelevante respecto de la lógica subyacente del programa, SSPIA provee al usuario de ciertas facilidades para detectar y explorar las funcionalidades del sistema bajo estudio. En la siguiente sección se describen dichas facilidades

4. Identificación de Funcionalidades

Todo los procesos explicados en secciones precedentes sirven como base para mejorar la comprensión de un sistema en particular. Con este objetivo, SSPIA provee un algoritmo que permite filtrar elementos considerados de poca relevancia con respecto a la lógica subyacente de dicho sistema. En este apartado se explica un conjunto de facilidades que proporciona dicha estrategia para asistir al usuario en el proceso de detección y análisis de funcionalidades.

Teniendo en cuenta que en el GELM han quedado sólo métodos sensibles a la lógica subyacente del sistema, una estrategia para aproximar estas funcionalidades es mediante el análisis de fan-in. Los nodos cuyo fan-in es igual a 0 podrían ser considerados como los puntos de entrada al sistema. Mediante esta aproximación, todos caminos que comiencen desde dicho nodo describen la funcionalidad que el método implementa. De esta manera, se pueden desagregar las funcionalidades que el sistema presenta, con el fin de analizar por separado el propósito de cada una. Tomando como ejemplo el

grafo mostrado en la figura 3, el resultado de aplicar este método equivale a los siguientes conjuntos de nodos y arcos:

$Func_{X.a()}$ = {(X.a(),X.b()); (X.b(),X.c()); (X.b(),X.d()); (X.d(),X.e()); (X.e(),X.f()); (X.e(),X.g()) }
$Func_{X.h()}$ = {(X.h(),X.i()); (X.i(),X.d()); (X.d(),X.e()); (X.e(),X.f()); (X.e(),X.g()) }
$Func_{Y.j()}$ = {(Y.j(),X.d()); (X.d(),X.e()); (X.e(),X.f()); (X.e(),X.g()); (Y.j(),X.X()) }
$Func_{Y.j()}$ = {}

Como se puede notar, las funcionalidades se ven plasmadas en cada conjunto $Func_{<n>}$, donde un arco desde el método $M.i()$ a $M.j()$ se representa con la tupla $(M.i(), M.j())$. En términos de operaciones de grafo, el conjunto $Func_{<n>}$ representa la cerradura transitiva del nodo *n*.

Sin embargo, cuando el sistema es grande, el GELM puede llegar a alcanzar un tamaño inmanejable, aún cuando se hayan filtrado elementos irrelevantes a la lógica del sistema; en estos casos, es necesario incorporar otro tipo de técnicas de visualización y procesamiento de información. Por este motivo, en SSPIA, se utilizan técnicas de agrupamientos (más conocidas como técnicas de *clustering*) para agrupar nodos que posean similaridades estructurales.

En pocas palabras, SSPIA incorpora las siguientes técnicas, algoritmos y facilidades para detectar funcionalidades:

- Implementa las etapas de filtrado definidas en la sección 3.
- Aplica un algoritmo de clustering que permite especificar el grado de agrupamiento, de forma interactiva. De esta manera se permite al usuario especificar la intensidad del agrupamiento de acuerdo a lo que éste requiera.
- Analiza cada agrupamiento individualmente, sin tener en cuenta todo contexto del GELM. Por cada agrupamiento se señalan los nodos con fan-in cero.

Por medio de las características destacadas previamente, es posible inferir las distintas funcionalidades que presenta el sistema bajo estudio. En la siguiente sección se presenta un caso de estudio para probar la utilidad del enfoque.

5. Caso de Estudio

En esta sección se presenta un caso de estudio para mostrar la aplicabilidad de la estrategia SSPIA (*Strategy for Static Program Information Analysis*) en el análisis estático de un sistema OO. En el mismo se analiza una herramienta de medición y evaluación denominada NESSy (New Evaluation Software System) [16]. Dicha herramienta está escrita en Java y utiliza la librería gráfica *Prefuse*[3].

Como se mencionó en la sección precedente, para poder llevar a cabo el análisis de este caso de estudio, se construyó un prototipo que implementa la estrategia planteada en este artículo. Éste recibe como entrada el GELM extraído del sistema bajo estudio y permite filtrar (sección 3) y asistir al usuario en el proceso de detección y análisis de funcionalidades del mismo (sección 4).

La construcción del GELM fue llevada a cabo utilizando *Dependency Finder*[4]. Se utilizó esta herramienta porque presenta un *toolkit* que posibilita la extracción y manipulación de todas las dependencias de un sistema escrito en Java; de la misma manera que posibilita exportar las dependencias en diferentes formatos (XML, GraphML, entre otros). Por otra parte, se ha utilizado Document Object Model (DOM) [23] para implementar los filtros explicados en los apartados 3.1 y 3.2.

Dependecy Finder extrae el GELM inicial de NESSy el cual contiene 2444 nodos y 17268 arcos. Luego de pasar por las dos fases de filtrado, el grafo contiene 730 nodos y 543 arcos; el mismo se puede examinar en la figura 4. Como es posible notar, la estrategia de filtrado redujo significativamente la cantidad de información (aproximadamente, 70 % respecto de nodos y 96 % respecto de arcos), prevaleciendo sólo información esencial respecto de la semántica del sistema. Sin embargo, se puede comprobar que el grafo sigue siendo difícil de manejar debido a su tamaño. Además, a simple vista se

[3] http://prefuse.org/
[4] http://depfind.sourceforge.net

observan diferentes componentes conexos, que no posee arcos entre sí.

Como se explicó en la sección precedente, una vez obtenido el GELM filtrado, se utiliza una técnica de clustering que permite agrupar nodos relacionados entre sí estructuralmente. El primer agrupamiento natural, a simple vista, es el de los componentes conexos.

En la figura 5 se exhibe una captura del prototipo en donde se ha realizado un acercamiento al GELM representado en la figura 4. Se puede observar un conjunto de nodos de un mismo color (pertenecen a un mismo agrupamiento "expandido") y otros de color negro (nodos con fan-in cero); además se visualizan nodos blancos con formas y tamaños distintos a los normales, los mismos indican agrupamientos "colapsados" de nodos. Los arcos grises representan llamadas entre nodos de distintos clusters y los arcos negros, llamadas entre nodos del mismo cluster (en la figura 6 se puede observar ambos tipos de arcos). En la figura 5 se muestra una captura de la interfaz del prototipo. En la misma se pueden distinguir los siguientes componentes:

- Un panel visualizador central, en donde se muestran los nodos y arcos del GELM.
- En el sector derecho, se encuentra un *slider* (o barra deslizante) que permite definir el nivel de agrupamiento deseado de acuerdo al algoritmo de clustering implementado. A medida que se aumenta el *slider*, la granularidad del agrupamiento tiende a ser más fina.
- En el sector inferior izquierdo, se puede distinguir un conjunto de botones que permiten: i) agrupar o desagrupar elementos (*Collapse Cluster* o *Expand Cluster* respectivamente) para mejorar la visualización de los clusters y ii) analizar un agrupamiento por separado (*Recall Clustering*).
- En el sector inferior derecho, se encuentra un área de texto en donde se puede visualizar los nombres de los métodos que representan los nodos con fan-in cero o en su defecto, el/los nodos raíz de cada cluster seleccionado.

A modo de ejemplo y con el propósito de que el artículo quede autocontenido, se ha optado por analizar un agrupamiento obtenido al observar todo el conjunto de agrupamientos iniciales (figura 5). En la figura 6 se exhibe una captura del prototipo que muestra el agrupamiento seleccionado. En el mismo se puede visualizar un cluster con fan-in cero (triángulo blanco, rótulo azul), es decir, un punto de entrada a la funcionalidad. Al seleccionar diho cluster, se puede observar en el área de texto el método con fan-in cero *...actionPerformed(java.awt.event.ActionEvent)*.
Este tipo de métodos, particularmente, suelen invocarse cuando se presiona un botón en una interfaz. Además se puede observar que invocan a un cluster cuyo nodo raíz representa un método con signatura *...botonEvaluacionGralActionPerformed (java.awt.event.ActionEvent)*. Esto reafirma que el agrupamiento de nodos analizado implementa una funcionalidad relacionada con *Evaluación General*. Al estudiar NESSy en funcionamiento, es posible distinguir dicha funcionalidad en la etapa de *Evaluación* [16]. Por medio de un botón rotulado *Evaluación General*, la herramienta realiza la evaluación final de los elementos analizados. Es posible destacar que dicha funcionalidad es una de las principales en la etapa de *Evaluación*. Éste último cluster (forma de pentágono) invoca a otro cuyo nodo raíz representa al método *...Evaluar_Sistema(Clases.Evaluacion.Sistema)*.
Dicho cluster se encuentra "expandido", a través del mismo se puede inferir, desde un enfoque estático, el conjunto de métodos que explican dicha funcionalidad.

De esta manera, el usuario puede interactuar con el prototipo de forma tal de obtener, de acuerdo a sus requerimientos, información necesaria para detectar funcionalidades en el sistema analizado.

Finalmente, es importante destacar que el análisis de la signatura de los métodos es muy importante, ya que brinda información semántica valiosa respecto de la funcionalidad del método.

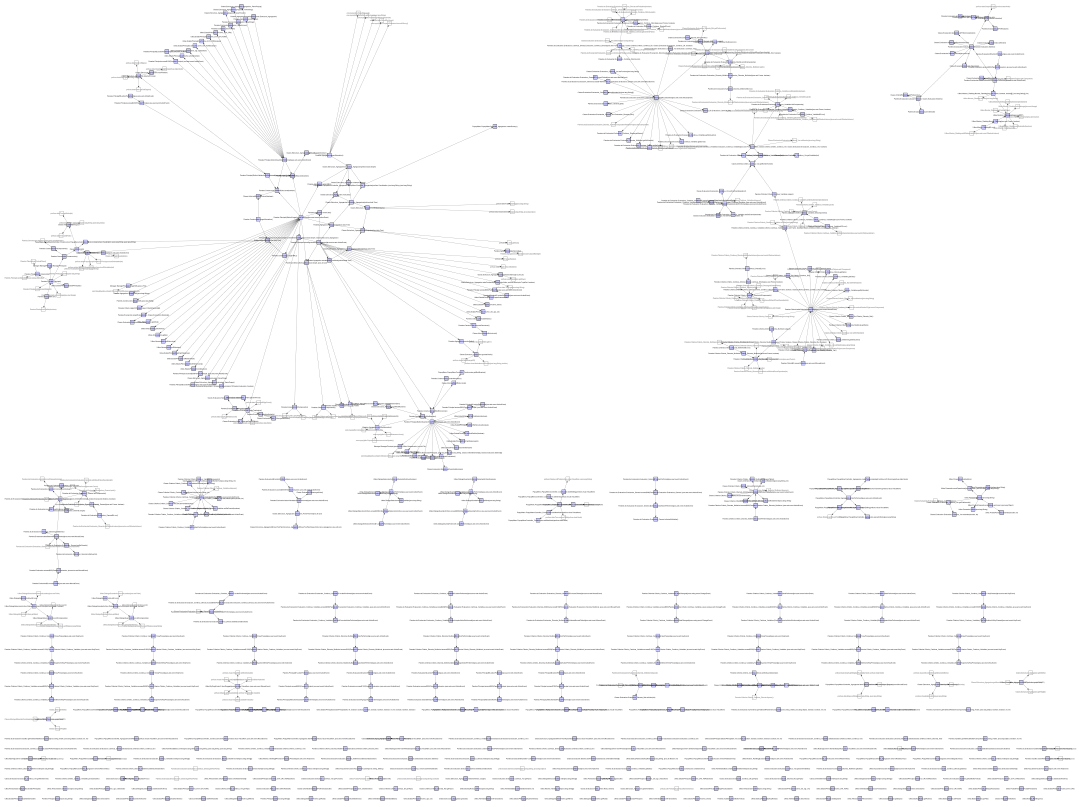


Figura 4. Grafo Estático de Llamada a Métodos correspondiente a la herramienta NESSy, luego de las fases de filtrado (sección 3). Contiene 730 nodos y 543 arcos. Grafo visualizado con la herramienta yEd.

6. Conclusiones y Trabajos Futuros

En este artículo se definió SSPIA (Strategy for Static Program Information Analysis), una estrategia que asiste a la Comprensión de Programas de sistemas OO. La misma propone extraer el GELM del sistema (o módulo) bajo estudio y aplicar filtros de información con el objetivo de disminuir el volumen del mismo. Para esto se utiliza un enfoque basado en la detección de métodos que no aportan información relevante respecto de la lógica del sistema analizado. Una vez filtrada la información, el GELM queda compuesto, en gran proporción, por métodos altamente relacionados con el Dominio del Problema del sistema. Finalmente, con el objetivo de inferir las funcionalidades del sistema, se analiza el GELM tomando como base los nodos cuyo fan-in sea cero. Además SSPIA incorpora las siguientes características para lograr inferir funcionalidades del sistema bajo estudio: i) incorpora técnicas de clustering para agrupar nodos que posean similaridades estructurales; ii) permite al usuario definir la granularidad del agrupamiento de nodos en el GELM; iii) posibilita el análisis de cada cluster individualmente, sin tener presente todo el grafo en la visualización.

Para probar la eficacia de la estrategia propuesta, la misma se aplicó a NESSy, una herramienta para medición y evaluación. El resultado de la aplicación de la estrategia fue muy satisfactorio debido

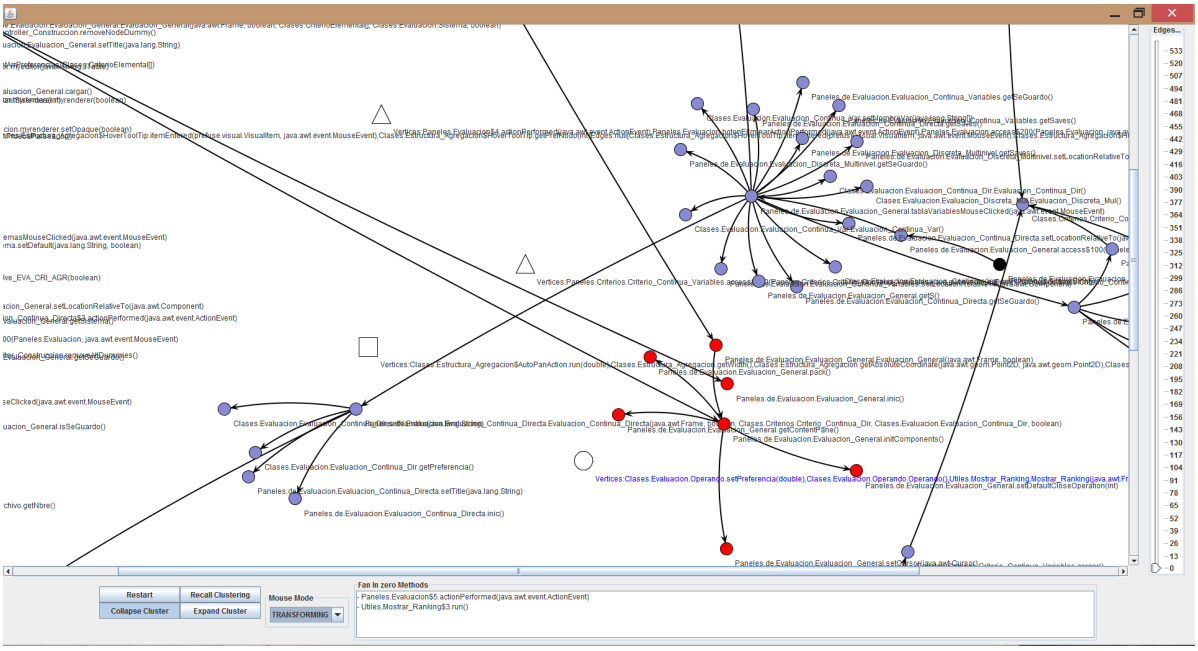


Figura 5. Captura del prototipo que implementa SSPIA, analizando la herramienta NESSy.

a que se pudo: i) lograr una significativa reducción del GELM (un porcentaje en reducción de nodos y arcos del 70 % y 96 % respectivamente), prevaleciendo sólo información esencial respecto de la semántica del sistema y ii) identificar funcionalidades que ofrece la herramienta bajo estudio, en este caso en particular la funcionalidad de *Evaluación General*.

El primer enfoque planteado en esta investigación es un buen punto de partida para elaborar una estrategia más robusta, eficiente y funcional para sistemas de software escritos en lenguajes OO. Claramente, para esto es necesario orientar la investigación en dirección a las siguientes temáticas:

- Enriquecer las dos fases de filtrado propuestas por medio de la detección de otros tipos de métodos que reflejen detalles de implementación, con el principal objetivo de identificar los Casos de Uso del sistema.
- Añadir a la estrategia la detección de polimorfismo como se propone en los trabajos [1, 30].
- Definir un método más sofisticado y preciso para el cálculo de los valores umbrales de *utilidad* e *intermediación* en la estrategia.
- Utilizar estrategias de generalización de conceptos como otro método de síntesis de información [28].
- Complementar la estrategia con extracción y análisis de información dinámica de los sistemas.
- Incorporar diferentes técnicas de clustering que mejoren en eficiencia y eficacia a la propuesta.
- Incorporar distintas técnicas de Visualización de Software para brindar diferentes perspectivas del sistema.

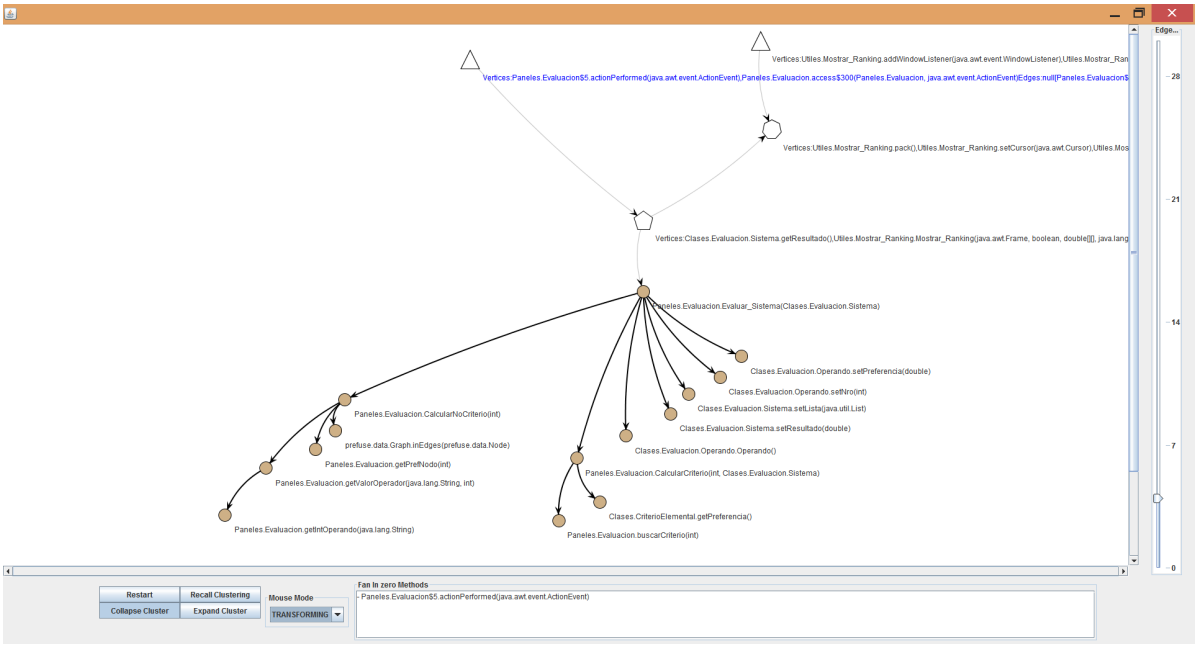


Figura 6. Captura del prototipo que implementa SSPIA, analizando la herramienta NESSy.

La incorporación de las estrategias mencionadas en los ítems precedentes asistirá significativamente al arduo proceso cognitivo que implica comprender un sistema, brindando solución a uno de los principales desafíos en Comprensión de Programas: Relacionar los Dominios del Problema y Programa.

Referencias

[1] D. F. Bacon and P. F. Sweeney. Fast Static Analysis of C++ Virtual Function Calls. *ACM Sigplan Notices*, 31(10):324–341, 1996.

[2] T. Ball. The Concept of Dynamic Analysis. In *Software Engineering - ESEC/FSE99*, pages 216–234. Springer, 1999.

[3] K. Bennett and V. Rajlich. Software Maintenance and Evolution: a Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 73–87, New York, NY, USA, 2000. ACM.

[4] M. Beron. Program Inspection to interconnect Behavioral and Operational Views for Program Comprehension. *Ph.D Thesis Dissertation at University of Minho. Braga. Portugal*, 2010.

[5] M. Berón, P. Henriques, M. J. Varanda Pereira, and R. Uzal. Static and Dynamic Strategies to Understand C Programs by Code Annotation. *European Joint Conference on Theory and Practice of Software - Satellite Event - Open Cert.*, 2007.

[6] D. Binkley. Source Code Analysis: a Road Map. In *2007 Future of Software Engineering*, pages 104–119. IEEE Computer Society, 2007.

[7] B. Boehm. Software Engineering Economics. *Software Engineering, IEEE Transactions on*, (1):4–21, 1984.

[8] G. Booch, R. James, and I. Jacobson. *The Unified Software Development Process*. Addison-Wesley Object Technology Series, 2004.

[9] G. Bracha, J. D. Darcy, and P. Von Der Ahe. Introspection Support for Local and Anonymous Classes, Feb. 23 2010. US Patent 7,669,184.

[10] U. Brandes. A Faster Algorithm for Betweenness Centrality*. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[11] R. Brooks. Using a Behavioral Theory of Program Comprehension in Software Engineering. *Proceedings of the 3rd International Conference on Software Engineering*, pages 196–201, 1978.

[12] B. Cornelissen, A. Zaidman, D. Holten, L. Moonen, A. van Deurse, and J. Wijk. Execution Trace Analysis through Massive Sequence and Circular Bundle Views. *Report TUD-SERG-2008-008*, 2008.

[13] D. Da Cruz, P. Henriques, and J. Pinto. Code Analysis: Past and Present. *Proceeding of the Third International Workshop on Foundations and Techniques for Open Source Software Certification*, 00:1–10, 2009.

[14] A. De Lucia. Program Slicing: Methods and Applications. In *Source Code Analysis and Manipulation, 2001. Proceedings. First IEEE International Workshop on*, pages 142–149. IEEE, 2001.

[15] T. Eisenbarth, R. Koschke, and D. Simon. Aiding Program Comprehension by Static and Dynamic Feature Analysis. In *Software Maintenance, 2001. Proceedings. IEEE International Conference on*, pages 602–611. IEEE, 2001.

[16] M. Enrique, B. Mario, M. Germán, V. P. M. J., and H. Pedro. NESSy: a New Evaluator for Software Development Tools. In J. P. Leal, R. Rocha, and A. Simões, editors, *2nd Symposium on Languages, Applications and Technologies*, volume 29 of *OpenAccess Series in Informatics (OASIcs)*, pages 21–37, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[17] R. Gupta, M. L. Soffa, and J. Howard. Hybrid Slicing: Integrating Dynamic Information with Static Analysis. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 6(4):370–397, 1997.

[18] A. Hamou-Lhadj and T. Lethbridge. Summarizing the Content of Large Traces to Facilitate the Understanding of the Behaviour of a Software System. In *Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on*, pages 181–190. IEEE, 2006.

[19] H. Lieberman and C. Fry. Bridging the Gulf Between Code and Behavior in Programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 480–486. ACM Press/Addison-Wesley Publishing Co., 1995.

[20] D. Littman, J. Pinto, S. Letovsky, and E. Soloway. Mental Models and Software Maintenance. *Journal of Systems and Software*, 7(4):341–355, 1987.

[21] J. Maletic, J. Leigh, A. Marcus, and G. Dunlap. Visualizing Object-Oriented Software in Virtual Reality. In *Proceedings of the 9th International Workshop on Program Comprehension (IWPC'01*, pages 26–35. Society Press, 2001.

[22] G. C. Murphy, D. Notkin, W. G. Griswold, and E. S. Lan. An Empirical Study of Static Call Graph Extractors. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7(2):158–191, 1998.

[23] G. Nicol, L. Wood, M. Champion, and S. Byrne. Document Object Model (DOM) Level 3 Core Specification. 2001.

[24] T. Pigoski. *Practical Software Maintenance: Best Practices for Managing Your Software Investment*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[25] A. Rohatgi, A. Hamou-Lhadj, and J. Rilling. An Approach for Mapping Features to Code Based on Static and Dynamic Analysis. In *Program Comprehension, 2008. ICPC 2008. The 16th IEEE International Conference on*, pages 236–241. IEEE, 2008.

[26] S. Rugaber. Program comprehension. *Encyclopedia of Computer Science and Technology*, 35(20):341–368, 1995.

[27] B. Ryder. Constructing the Call Graph of a Program. *Software Engineering, IEEE Transactions on*, (3):216–226, 1979.

[28] K. Spärck Jones. Automatic Summarising: the State of the Art. *Information Processing & Management*, 43(6):1449–1481, 2007.

[29] M.-A. Storey. Theories, Methods and Tools in Program Comprehension: Past, Present and Future. *Proceedings of the 13th International Workshop on Program Comprehension*, pages 181–191, 2005.

[30] V. Sundaresan, L. Hendren, C. Razafimahefa, R. Vallée-Rai, P. Lam, E. Gagnon, and C. Godin. Practical Virtual Method Call Resolution for Java. *ACM SIGPLAN Notices*, 35(10):264–280, 2000.

[31] A. von Mayrhauser and M. Vans. Comprehension Processes During Large Scale Maintenance. In *Proceedings of the 16th International Conference on Software Engineering*, pages 39–48. IEEE Computer Society Press, 1994.

Web Application Frameworks Evaluation

Ana Funes, Aristides Dasso
Software Engineering Group,
Departamento de Informática,
Universidad Nacional de San Luis,
Ejército de los Andes 950, 5700 San Luis, Argentina
e-mails: {afunes, arisdas}@unsl.edu.ar

Abstract

We present here a development of system evaluation models employing the LSP (Logic Score of Preference) method [14], [15]. For this particular instance we concern ourselves with developing models for the evaluation of Web Application Frameworks (WAF). The use of the LSP method allows expressing in the model characteristics that other additive evaluation techniques do not permit. Therefore we can have the possibility of constructing models that can be fit with greater precision to the potential user's needs, either for those that want to compare the advantages/disadvantages of different WAF, or for those who, building their own model can choose the tool best suited for his/her needs.

1. Introduction

In this work, we present a model for the evaluation and selection of Web Application Frameworks (WAF).

There are a great number of WAF; e.g. [2], [3], [4], [6], [20], [21], [22], among others. In some cases, more than a hundred can be counted (e.g. see [18] or [16]).

The model has been developed following the Logic Score of Preference (LSP) method [7] - [15] –a method that allows the creation of models for the evaluation, comparison and selection of complex systems. It prescribes a number of steps to be followed and it provides a number of operators of a Continuous Logic that combined in the right way permits the development of evaluation models according to the user's requirements.

Choosing a system or tool, when these have many different characteristics and features, it is never a simple task. Users are confronted with assessing their own requirements and matching them with the features of the different systems or tools to be able to make the right choice. This decision can have dire consequences since choosing wrong –even if the tool chosen is free— can mean delays in the development, a steep learning curve not according with the capabilities of the team, etc.

Both in the literature and in the web, there are several methods for evaluation of different complex systems or tools. These go from the simple additive methods to more elaborate models. These models can be adapted to evaluate WAF and WIDE (Web Integrated Development Environments), and some of these models are on occasion used to compare WAF and WIDE.

In [1] some criteria are given but there is no prescription of how to integrate the results of applying the data obtained in a model. A lot of interesting and important data can be found in [18] but here again there is no model to integrate the data, leaving a potential user to design the model. Others, such as [19] or [24] also proposed a set of requirements but fall short of producing a model.

In [5] a detailed analysis of costs requirements is done but the model is a simple additive one.

More interesting is the fuzzy system used in [23] "In order to enhance the evaluation result accuracy and evaluation process efficiency, this paper presents a Distributed Fuzzy Qualitative Evaluation System (DFQES) to be applied in complex distributed evaluation scenarios"

The rest of this work is organized as follows. Section 2 presents an overview of WAF and WIDE and a brief description of the LSP method is presented in Section 3, both necessary for the reading of this work. In Section 4, we present a possible model for evaluating WAF only, showing part of the identified requirement tree, some elementary criteria as well as part of the developed aggregation structure. Finally, in Section 5, we close the work with some conclusions and future work.

2. WAF and WIDE

Web Application Frameworks (WAF) are frameworks designed to support development of web sites and also other applications, services and resources

tailored for the web. This kind of frameworks generally provide most of the functionality needed in web applications such as user session administration, data persistence, and templates, etc., helping also with good coding techniques and development guidance when using programming languages.

WAF are supported for different platforms as well as programming languages (e.g. [2], [3], [4], [6], [17], [20], [21], and [22]).

On the other hand, Web Integrated Development Environments (WIDE) are more than an Integrated Development Environment (IDE); they are geared to help in the development of applications services and other resources for the web. The objective of a WIDE is to provide the environment to ease systems development oriented to the web.

Some WIDE support several platforms and programming languages, such as Koding that support Python, Java, Perl, Node.js, Ruby, C, C++, PHP and Go. Other WIDE support only one language, e.g. PythonAnywhere, a WIDE specialized for Python.

Not always the difference between WAF and WIDE are clear cut and is also possible that with time they will disappear completely. For the rest of this work, we shall concern with showing parts of a model as an example of how it is possible to construct a model to evaluate WAF.

3. The Logic Score of Preference method

The Logic Scoring of Preference (LSP) method [7] - [15] we have applied in the development of the model presented here is a method employed for the realization of complex criterion functions and their application in the evaluation, optimization, comparison and selection of general complex systems.

The LSP method can be used to evaluate complex systems and, since it is a general evaluation method, it can also be employed in particular in the evaluation processes involved in the choosing of a WAF.

Since this method is not a simple additive scoring method but allows the use of complex and/or decisions, is especially useful where these conditions apply.

As a starting point in LSP, it must be clearly determined what are the user requirements, the main attributes of the system and their value preferences. These attributes are called *performance variables*. Each one of these variables is mapped into an *elementary preference* by defining and applying the corresponding *elementary criteria*.

An elementary criterion is a function that transforms a real value, coming from a performance variable, into a value belonging to the [0,100] interval –i.e. the corresponding elementary preference. An elementary preference represents the degree of fulfilment of a requirement, where 0 means that the requirement has not been fulfilled at all and 100 that it has been completely satisfied. Therefore, to define an elementary criterion it is necessary to have some previous experience to determine what the acceptable range of values is for the corresponding performance variable.

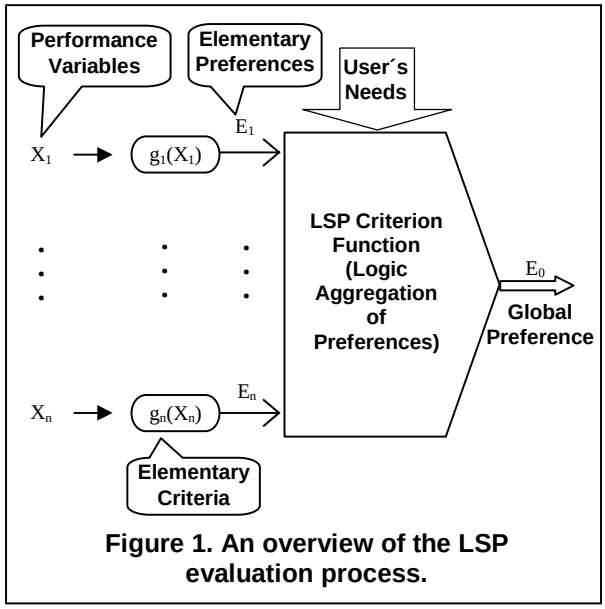


Figure 1. An overview of the LSP evaluation process.

The elementary preferences, obtained from the transformation of the performance variables via the corresponding elementary criteria, are used as input to the *LSP criterion function*.

The LSP criterion function or *Aggregation Structure* is a function that yields a single global indicator between 0 and 100 of the degree of fulfilment of the whole system requirements. It is built by aggregating the elementary preferences by means of a set of operators. To aggregate preferences means to replace a group of preferences (the input preferences) by a single preference (the output preference), which denotes the degree of satisfaction of the evaluator with respect to the whole group of input preferences. The output preferences must be aggregated again until a single global preference is obtained.

To calibrate the LSP criterion function it is necessary to take into account the needs of the end users. The process of calibration obviously represents the most complex phase in the whole evaluation. The global preference –obtained as output of the LSP function– is the result of the combination of the elementary preferences taking into account both the relative importance of each preference and the necessary logic relationship between them.

Once the calibration of the LSP criterion function has finished, the evaluation of each competitive system can start. It means that, for each competitive system, we must provide as input to the aggregation model the set of

values corresponding to the performance variables to obtain as output a global performance indicator for each system under evaluation.

An overview of the applied method is shown in Figure 1, where the performance variables X_1, …, X_n from the system requirement tree are transformed, by means of the elementary criteria g_1,…, g_n, into the elementary preferences E_1, …, E_n. These elementary preferences are provided as input to the aggregation structure or LSP criterion function, which returns a single global indicator E_0 representing the fulfilment degree of all the requirements of the system under evaluation.

4. A model for assessing WAF

We show, in this section, the use of the LSP method by constructing short examples of models that can help to evaluate WAF. The steps proposed by the LSP method to create a final evaluation model (or LSP criterion function) are applied in this section. This LSP criterion function delivers a quantitative indicator of the requirement satisfaction level (global preference) that a WAF must fulfil.

We also give an overview of the main activities needed to obtain the model as the method prescribes; i.e. development of the system requirement tree, definition of elementary criteria and aggregation of preferences. These are described in Sections 4.1, 4.2 and 4.3.

4.1. Requirement tree

A requirement tree should include the characteristics, properties, services, etc. that are useful to a particular user's needs.

The requirement tree can be as extensive as is deemed necessary. The requirement tree is a hierarchical structure having as many levels as the user needs. Each level would group requirements that the user considers they are related to each other. The leaves of the tree constitute the performance variables.

As was said above, next we present a short example to illustrate the method. The example used is taken from the tests proposed in Web Framework Benchmarks [18]. The fact that the requirements tree can be constructed with test descriptions will allow the definition of elementary criteria that will take as input the values obtained from the tests –as performance variables– and will transform them as elementary preferences to serve as inputs to the criterion function. In Table 1 the first level of the tree is shown, this has only six branches.

Each of the tests mentioned above has several conditions that can also be used for the requirement tree; however, for reason of space we are showing only part of them (Single query) in Table 2.

Table 1. Requirement tree (1st level)

1. JSON serialization
2. Single query
3. Multiple queries
4. Data Updates
5. Fortunes
6. Plain Text

Table 1 shows the top level of the requirement tree. The **first** requirement –JSON serialization– is a test that "exercises the framework fundamentals including keep-alive support, request routing, request header parsing, object instantiation, JSON serialization, response header generation, and request count throughput." [18].

Single query, that is the **second** requirement, "exercises the framework's object-relational mapper (ORM), random number generator, database driver, and database connection pool." [18].

The **third** requirement, Multiple queries, "is a variation of Test #2 (Single query) and also uses the World table. Multiple rows are fetched to more dramatically punish the database driver and connection pool. At the highest queries-per-request tested (20), this test demonstrates all frameworks' convergence toward zero requests-per-second as database activity increases." [18].

Updates, the **fourth** requirement in Table 1, "is a variation of Test #3 (Multiple queries) that exercises the ORM's persistence of objects and the database driver's performance at running `UPDATE` statements or similar. The spirit of this test is to exercise a variable number of read-then-write style database operations." [18].

The **fifth** requirement –Fortunes– "exercises the ORM, database connectivity, dynamic-size collections, sorting, server-side templates, XSS countermeasures, and character encoding." and the fourth –Plain Text– "is an exercise of the request-routing fundamentals only, designed to demonstrate the capacity of high-performance platforms in particular. The response payload is still small, meaning good performance is still necessary in order to saturate the gigabit Ethernet of the test environment." [18].

The **sixth** requirement –Plaintext– "is an exercise of the request-routing fundamentals only, designed to demonstrate the capacity of high-performance platforms in particular. The response payload is still small, meaning good performance is still necessary in order to saturate the gigabit Ethernet of the test environment." [18]

Table 2. Expansion of the item "2 Single query" in the Requirement tree shown in Table 1

2. Single query
2.1. Database-access responses per second for concurrent accesses
2.1.1. 8-32 concurrent accesses
2.1.1.1. (8)
2.1.1.2. (16)
2.1.1.3. (32)
2.1.2. 64 -256 concurrent accesses
2.1.2.1. (64)
2.1.2.2. (128)
2.1.2.3. (256)
2.2. Latency of database-access responses
2.2.1. Average
2.2.2. Max
2.2.3. Standard Deviation
2.2.4. Errors
2.3. Framework overhead of database-access responses

In [1], a set of thirteen criteria are given. These are more general and not so precise and based on actual tests as those shown in Table 1 and Table 2. However, we can construct a requirement tree with these more general criteria as shown in Table 3. So, if a user does not want to use requirements that can be too technical, she/he can still build a model that can serve her needs even if it is a simple one.

Table 3. A more general requirement tree

1. Business
1.1. Audience
1.2. Licensing
1.3. Other Biz
2. Technical
2.1. Core
2.2. Data
2.3. Testing
2.4. Documentation
2.5. Extension & Plugins
3. Operations
3.1. Servability
3.2. Security and Speed
4. Team
4.1. Longevity
4.2. Community
4.3. Learning Curve

4.2. Definition of elementary criteria

According to the LSP method after developing the requirement tree and the performance variables have been determined, the elementary criteria can be defined.

Each elementary criterion is a mapping from a performance variable value into an elementary preference value. Since the interpretation of an elementary preference is the degree of fulfilment of a given requirement, this is a real number belonging to the interval [0, 100]. Therefore, the value 0 corresponds to a situation where the performance variable does not satisfy the requirements and the value 100 to the case that the requirement is completely fulfilled, while the values between 0 and 100 denote partial satisfaction of requirements.

Since only the leaves of the requirement tree correspond to performance variables, the elementary criteria must be defined only for the requirement tree's leaves.

To illustrate the construction of elementary criteria we have taken requirements from Table 2, i.e. 2.2. "Latency of database-access responses":

2.2.1. "Average Latency".

Considering that for this item lower values are better, a possible elementary criterion is given by the function g_{AL}, where x represents the value of the Average Latency:

$$g_{AL}(x) = \begin{cases} 0 & x > 10 \\ 100 - (x * 10) & otherwise \end{cases}$$

Since x ranges from 1.4ms to more than a 100ms, as not to get a value of $g(x)$ too low when x takes values lower than 10, we multiply x by 10.

On the other hand, g_{AL} also eliminates (assign the value 0) those WAF that exceeds 10ms.

The same goes for item 2.2.3. "Standard Deviation". However, we do not multiply *Max* by 10 since the range of this item exceeds 100, so we have divided it by 10.

Item 2.2.4. "Errors", on the other hand, needs to be divided by 100, since it ranges from 0 to more than 10,000.

On the contrary, when considering sub items included in 2.1.1. "Database-access responses per second for concurrent accesses", higher values are better, so we define the elementary criteria by dividing them by 1000; for example, for eight concurrent access (item 2.1.1.1. (8)), the elementary criteria is given by the function g_{DBa8}, where x gives the number of seconds 8 concurrent accesses to a database take.

$$g(x) = x/1000$$

Considering that the data given in [18] for this item ranges from around 1,000 to more than 15,000 is that we decided dividing these values by 1,000 to be able to bring them around the interval [0, 100].

Those are some of the very many possible elementary criteria.

For a more complete understanding of the construction of elementary criteria we refer the reader to the explanation given in [7], [15].

In the next section, we show how all of these items play one to another when aggregating the preferences, using the corresponding aggregation functions as well as the adequate weights.

4.3. Aggregation of preferences

The process of aggregation of preferences can start when the requirement tree has been finished and the elementary criteria defined. This process uses the structure of the system requirement tree to build a new tree structure, i.e. the final aggregation structure or LSP criterion function.

These structure aggregates groups of related elementary preferences and generates, in this way, subsystem preferences. So, the elementary preferences –corresponding to the leaves of the requirement tree– are thus aggregated in new elementary preferences. This is a bottom up process that is repeated with the groups of preferences until a single global preference is obtained. This logic aggregation structure created by applying the described process must reflect the user requirements that, in this case, are the requirements of a WAF.

We can describe the process saying that to aggregate n elementary preferences $E_1,...,E_n$ in a single preference E, the resulting preference E –interpreted as the degree of satisfaction of the n requirements– is expressed by a function having the following properties:

1. The relative importance of each elementary preference E_i (i= 1...n) is expressed by a weight W_i,
2. $min(E_1,...,E_n) \le E \le max(E_1,..., E_n)$.

These functions are obtained from the instantiation of the weighted power means:

$$E(r) = (W_1 E_1^r + W_2 E_2^r + ... + W_n E_n^r)^{1/r}$$, where

$0 < W_i < 100$, $0 \le E_i \le 100$, $i = 1, ..., n$, $W_1 + ... + W_n = 1$, $-\infty \le r \le +\infty$

The choice of r determines the location of $E(r)$ between the minimum value $E_{min}=min(E_1,...,E_n)$ and the maximum value $E_{max}=max(E_1,...,E_n)$. For $r = -\infty$ the weighted power mean reduces to the pure conjunction (the minimum function) and for $r = +\infty$ to the pure disjunction (the maximum function), giving place to a Continuous Preference Logic (CPL). The range between pure conjunction and pure disjunction is usually covered by a sequence of equidistantly located CPL operators named: C, C++, C+, C+–, CA, C–+, C–, C– –, A, D– –, D–, D–+, DA, D+–, D+, D++, D. For a more detailed description of the technique for selection of r see [7] and [11].

The weights associated to each elementary preference are assigned by the user according to the importance that each elementary preference has in the model being constructed. The same goes when choosing the different CPL operators.

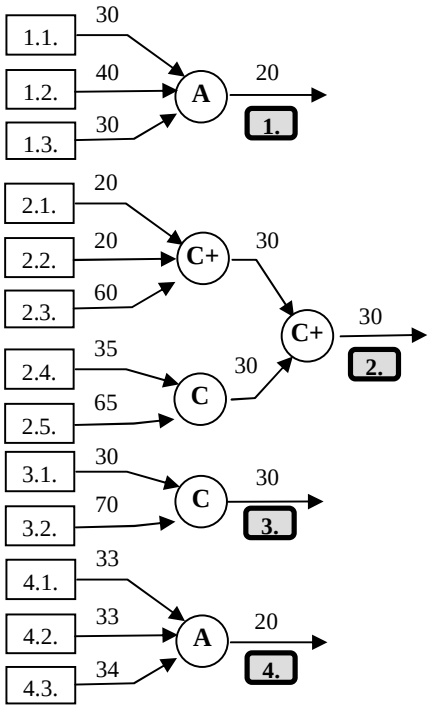


Figure 2. A possible aggregation structure for items in Table 3

To illustrate this stage of the LSP method, we show a possible aggregation structure in Figure 2, constructed from the requirement tree given in Table 3. In this case, items "2. Technical" and "3. Operations" have been considered more important and this is shown in Figure 2, not only by the use of different weights but by the application of more conjunctive CPL operators.

In the final aggregation structure, shown in Figure 3, where as well as we have used different weights to the items that are considered more important, we have also used more conjunctive operators, to make sure that the mandatory items "2. Technical" and "3. Operations" do not take very low values or zero; if they do then the whole structure will return zero. This concept is enacted by using a strong 'andness' function, namely C–; in that way, we make sure that the input items do not have low values, and if they do they will be lowered still by the CPL operator (C–) used, so the end result (E_0) will be

quite low also.

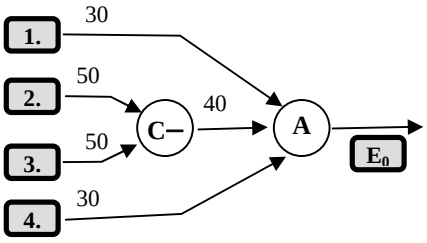


Figure 3. A possible final aggregation structure for items in Table 3 and Figure 2.

In Figure 4 we illustrate a possible aggregation structure for item "2. Single query" taken from Table 2. All the items are aggregated using the CPL operator 'A', which corresponds to the arithmetic mean. The relative importance assigned to each item is implemented using different weights.

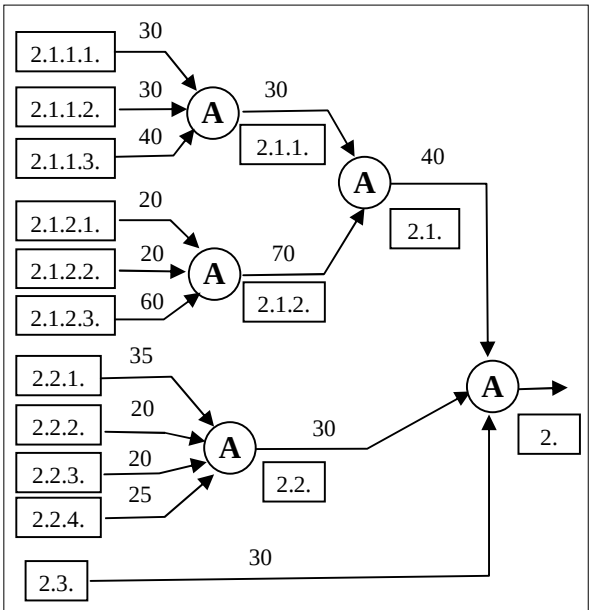


Figure 4. A possible aggregation structure for item "2.1. Single query" in Table 2

Figure 5 shows a different version of a structure for item "2. Single query" taken from Table 2, showing only the top three items and where more importance has been given to items "2.1. Database-access responses per second (concurrent)" and "2.2. Latency of database-access responses" over the other item "2.3. Framework overhead of database-access responses", using weights and a special aggregation structure. Bear in mind that in considering the values for these items greater is not better (see section 4.2. Definition of Elementary Criteria); but once we have the correct elementary criteria we can use weights to assign the desired importance to each item.

So, in Figure 5 we have used a special aggregation structure named "partial absorption" that allows aggregation of a group of preferences where some are mandatories, i.e. they must be 'present' with a value, while other are optionals, i.e. they could have a very low value. In this case, we considered, as was said above, items "2.1. Database-access responses per second (concurrent)" and "2.2. Latency of database-access responses" as mandatory and item "2.3. Framework overhead of database-access responses" as optional.

If we use a GPL function with a high degree of andness, like C, C+ or CA, to aggregate both items, we run the risk of getting a value of zero if the optional preference where zero, no matter what the values –of the mandatory preferences have. This would transform, in practice, the optional preference as mandatory; something that is not what is desired.

Therefore in Figure 5 two partial absorption aggregation structures are used, first one to aggregate items 2.2. and 2.3. so if item 2.3. would be zero it will not make the result of the aggregation of items 2.2. and 2.3. as would be in case a function with a more 'andness' –such as CA for example– where used. Then the output of this structure is aggregated, again using a 'partial absorption' structure, with item 2.2. For a more detailed explanation of the partial absorption structure see [13].

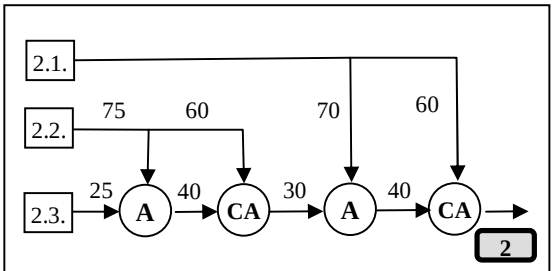


Figure 5. A possible aggregation structure for item "2. Single query", fromTable 2, using 'partial absorption'.

The whole aggregation structure for Table 1 is shown in Figure 6. There, we can see that items 2 and 3; 4 and 5 are considered more important. This has been implemented by using again two "partial absorption" structures, i.e. one aggregating items 2 and 3 first and then adding item 1 (this is at the top of the figure); and at the bottom of the figure items 4 and 5 are aggregated

together and then with another partial absorption item 6 is added.

Considering that we have presented two different requirement trees (see Table 1 and Table 3), there would be no problem at all to construct a requirement tree that join both of them and then the corresponding aggregation structure for this new tree.

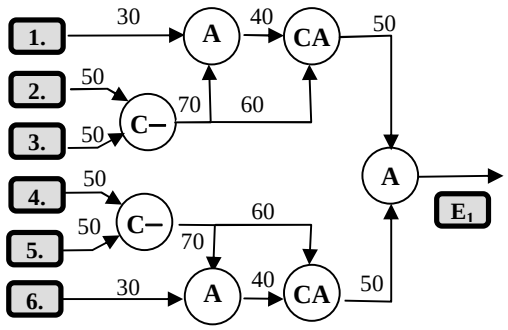


Figure 6. A final aggregation structure for items in Table 1.

So, for example, the whole of Table 1 could be now a subtree for item "2.3. Testing" in Table 3. From this new requirement tree, much more complete, a corresponding aggregation structure could be built. This shows that the LSP method is flexible enough to construct models with as much complexity as desired by the user.

5. Conclusions and Future Work

As it was said, there is a great number of WAF. Therefore, when the time comes for a company to choose a WAF the company is confronted with a serious problem, namely which is the best WAF that would satisfy the needs of the enterprise and what would be the costs, sometimes hidden, of choosing wrong. All this means that having a tool, a model, and data to be able to choose rightly is of the utmost importance.

In this work, we have presented an outline of a model that can help to choose the right WAF. The method used follows all the steps needed from requirements to the construction of a model based on a continuous logic that can be used for the evaluation and comparison of different WAF.

Although it has not been shown in the model proposed, cost –that is obviously an important side– can be incorporated into the model. However cost analysis is not a simple matter and needs to be carefully assessed to be able to add into the model. Items to be considered in this case are not only equipment's cost, but also implementation, learning curve, maintenance, adaptability, amortization, storage, transportation, etc.

That is a work that is being attacked and we hope in future to be able to present the results of this work.

Another line that we are currently exploring is modifying the model to adapt it to evaluate WIDE, since as we said above WAF and WIDE are not so dissimilar.

6. References

[1] 13 Criteria for Evaluating Web Frameworks. https://blog.safaribooksonline.com/2013/10/14/13-criteria-for-evaluating-web-frameworks/ (Retrieved 7/8/14)

[2] ASP.NET, http://www.asp.net/ (Retrieved 3/8/14)

[3] CakePHP http://cakephp.org/ (Retrieved 3/8/14)

[4] Catalyst http://www.catalystframework.org/ (Retrieved 3/8/14)

[5] Changpil Lee. "An Evaluation Model for Application Development Frameworks for Web Applications". The Ohio State University. 2012.

[6] ColdBox, http://www.coldbox.org/ (Retrieved 3/8/14)

[7] Dujmovic, J. J. and Bayucan, A., "Evaluation and Comparison of Windowed environments", Proceedings of the IASTED Interna Conference Software Engineering (SE'97), pp 102-105, 1997.

[8] Dujmovic, J. J. and Elnicki, R., "A DMS Cost/Benefit Decision Model: Mathematical Models for Data management System Evaluation, Comparison, and Selection", National Bureau of Standards, Washington, D.C., No. NBS-GCR-82-374, NTIS No. PB82-170150 (155 pages), 1982.

[9] Dujmovic, J. J., "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG96 Proceedings, vol. 1, pp.368-378, 1996.

[10] Dujmovic, J. J., "Preferential Neural Networks. Chapter 7 in Neural Networks – Concepts, Applications, and Implementations", Vol. II. Edited by P. Antognetti and V. Milutinovic. Prentice Hall Advanced Reference series, Prentice – Hall, pp 155-206, 1991.

[11] Dujmovic, J. J., "Quantitative Evaluation of Software", Proceedings of the IASTED International Conference on Software Engineering, edited by M.H. Hamza, pp. 3-7, IASTED/Acta Press, 1997.

[12] Dujmovic, J. J., "Characteristic forms of generalized conjunction/disjunction"; En Fuzzy Systems, 2008 (FUZZ-IEEE 2008). (IEEE World Congress on Computational Intelligence). 1-6 June 2008, pp. 1075 – 1080, ISSN: 1098-7584, E-ISBN: 978-1-4244-1819-0, Print ISBN: 978-1-4244-1818-3.

[13] Dujmović, J.J., Partial Absorption Function. Journal of the University of Belgrade, EE Dept., Series Mathematics and Physics, No. 659, 1979, pp. 156-163.

[14] Dujmovic, Jozo J., "Continuous Preference Logic for System Evaluation", IEEE Transactions on Fuzzy Systems, Vol. 15, Nº 6, December 2007.

[15] Dujmovic, Jozo, "Quantitative Evaluation of Software", Proceedings of the IASTED International Conference

on Software Engineering, edited by M.H. Hamza, pp. 3-7, IASTED/Acta Press, 1997.
[16] http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks
[17] JetBrains. http://www.jetbrains.com/webide/ (Retrieved 3/8/14)
[18] TechEmpower. Web Framework Benchmarks. http://www.techempower.com/benchmarks/#section=intro&hw=peak&test=json. (Retrieved 3/8/14)
[19] TreeFrog Framework http://www.treefrogframework.org/documents/comparison-of-performance (Retrieved 3/8/14)
[20] WebStorm. http://www.jetbrains.com/webstorm/ (Retrieved 3/8/14)
[21] WICKET, http://wicket.apache.org/ (Retrieved 3/8/14)
[22] Yesod, http://www.yesodweb.com/ (Retrieved 3/8/14)
[23] YUEN, K.K.F.; LAU, H.C.W. "A Distributed Fuzzy Qualitative Evaluation System". Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06), 2006.
[24] zeroturnaround; http://zeroturnaround.com/rebellabs/developer-productivity-report-2012-java-tools-tech-devs-and-data/ (Retrieved 3/8/14)

Análisis del comportamiento de los eventos de cambios a través de una ontología para gestionar versiones de una familia de producto

Sonzini M. Soledad[1,2], Vegetti Marcela[1], Leone Horacio[1]
[1]*INGAR, Instituto de Desarrollo y Diseño (CP3000)*
Avellaneda 3657, Santa Fe, Argentina
[2]*Universidad Nacional de La Rioja (CP5300)*
Luis M. de la Fuente S/N, La Rioja, Argentina
[1]*{ssonzini, mvegetti, hleone}@santafe-conicet.gob.ar*

Abstract

La información de una familia de producto puede ser modificada durante su ciclo de vida, por eventos de cambio originados por diversos motivos. Dada la naturaleza del cambio, la gestión de ciclo de vida de productos (PLM - Product Lifecycle Management) requiere de soluciones que permitan registrar los cambios ocurridos y gestionar las versiones de familia de producto, para mantener la integridad y la consistencia de los modelos de datos de productos. Teniendo en cuenta esta necesidad, el objetivo del trabajo es presentar un análisis sobre el impacto y el comportamiento de los eventos de cambio que pueden ocurrir en la información de producto, basado en una ontología de gestión de versiones. Esta ontología de versiones es utilizada en este artículo para representar los cambios y las versiones generadas por éstos en dos modelos diferentes de representación de productos: PRONTO (PRoduct ONTOlogy) y FM (Feature Model). En cada uno de estos casos, se identifica los elementos que pueden ser cambiados, provocando una nueva versión, y se analiza cómo los cambios pueden propagarse afectando otros elementos de cada modelo.

1. Introducción

La administración del ciclo de vida de producto (PLM) es una actividad eficiente para gestionar toda la información relacionada con un producto desde su inicio, desarrollo, madurez, hasta su final [1]. Es importante gestionar la información de los productos en todas las fases de su vida, debido a que un cambio en el mismo durante una determinada fase, podría propagarse y afectar la consistencia e integridad de la información en el resto de las fases. Los sistemas PLM requieren soluciones robustas que le permitan representar de forma consistente los modelos de datos de producto con el fin de compartir y operar con otras organizaciones, stakeholders, procesos y etapas en el ciclo de vida.

En muchos casos, es necesario que un producto cambie con el propósito de dar soporte a diferentes segmentos del mercado, adoptar nuevas tecnologías, mejorar una característica para competir en el mercado o modificar el diseño del producto. Esta capacidad de cambio de un producto recibe el nombre de variabilidad, definido como la habilidad de un producto para ser eficientemente extendido, cambiado, personalizado o configurado para su utilización en un dominio particular [2]. Pohl y colab. en [3] sostienen que es fundamental hacer una distinción entre variabilidad en el tiempo y variabilidad en el espacio. La primera de ellas es definida como "la existencia de diferentes versiones de un producto que es válido en diferentes tiempos", este tipo de variabilidad denota la evolución definiendo puntos de variación que ayudan a mantener el control del impacto de pequeños cambios. La variabilidad en el espacio es definida como "la existencia de un producto en diferentes formas en un mismo tiempo". Esta dimensión abarca de forma simultánea el uso de un producto variable en diferentes formas, es decir diferentes variantes de productos. Varias propuestas se han presentado en diferentes dominios para la representación de la variabilidad en el espacio. En el dominio de las industrias de manufactura muchas de estas propuestas recurren al concepto de familia de productos, entre ellas, PRONTO [4], una ontología de representación de información de producto para gestionar de forma eficiente la variabilidad. En tanto, en la industria del software es muy utilizado el modelo de características (FM- Feature Model) y el modelo de variabilidad ortogonal (OVM) [3], entre otras. Durante el desarrollo de este trabajo, se presta atención a la propuesta de PRONTO, en el dominio de la industria de manufactura, y FM en la industria del software.

Con el fin de modelar la variabilidad en el tiempo y registrar los eventos de cambio, este artículo describe un modelo de datos para la gestión de versiones y la representación del cambio ocurrido en la información de productos durante su ciclo de vida, a través conceptos generales que pueden ser especializados, independientemente del modelo de productos utilizado. El principal desarrollo de este trabajo se centra en el análisis del comportamiento y el impacto de los eventos de cambio. Para ello, una vez implementado el modelo mediante el uso de herramientas de la web semántica tales como el editor de onto-

logías Protégé v5.0, se desarrolla un conjunto de reglas formuladas en SWRL (Semantic Web Rule Language). Este conjunto de reglas, permite razonar nuevo conocimiento sobre el comportamiento de los individuos o instancias del modelo de gestión de versiones, mediante el uso de Pellet v2.2. Estas inferencias nos permitirán analizar los eventos de cambio y su propagación en diferentes niveles de representación de la información de una familia de producto. En [5] se describe SWRL con mayor detalle.

En base a esto, el trabajo se organiza de la siguiente forma: la sección 2 describe una ontología para la gestión de versiones. La sección 3 describe los conceptos básicos de la propagación de los cambios y su clasificación. Además, desarrolla un caso sencillo para ser implementado en PRONTO y FM. De este modo, se analiza la propagación de cambios en ambos enfoques. Finalmente, en la sección 4 se presentan las conclusiones y trabajos futuros.

2. Ontología para la gestión de versiones de Familias de Producto

Esta sección describe brevemente el modelo de datos propuesto en [6] que permite capturar y representar los cambios en la información de una familia de producto durante su ciclo de vida. El desarrollo de la ontología propuesta ha sido guiado por una metodología ad-hoc basada en principios ampliamente aceptados. La misma consta de las siguientes 4 etapas:

1. Especificación de requerimientos: mediante la definición de preguntas de competencias permite identificar el alcance y el propósito de la ontología,
2. Conceptualización del dominio: una visión informal del dominio es convertido en modelos semi formales utilizando UML ,
3. Implementación de la ontología mediante un lenguaje formal: permite codificar la ontología en un lenguaje formal,
4. Evaluación de la calidad de la ontología respecto a la etapa 1.

Estas etapas no son totalmente secuenciales, es común iterar sobre las mismas para refinar la ontología. En la Figura 1 se muestra los conceptos identificados durante la fase de conceptualización de la ontología de versiones propuesta en [6]. La entidad *ProductConcept*, que representa el concepto cuyas versiones se van a gestionar. Este concepto, deberá ser especializado dependiendo del modelo utilizado para representar la variabilidad en el espacio, esta especialización se explica en la sección 3.

La ocurrencia de al menos un cambio (*ChangeEvent)* genera una nueva versión (*Version*) de la entidad *Producto Concept*, afectando a un elemento de cambio (*ChangedEntity*). La entidad que es afectada por un determinado evento de cambio se representa por la asociación *AffectTo*. El concepto *ChangedEntity* permite representar los conceptos del modelo de variabilidad que son afectados por un evento de cambio.

Para capturar un conjunto de versiones asociadas a *ProductConcept*, se introduce la entidad *History*, la cual permitirá representar todas las versiones que surgen durante el ciclo de vida de una familia de productos. Cada *Version* tiene asociado un instante (*DateTime*) en que la versión comienza a ser válida (ver relación *validSince* en la Figura 1). Además, ese instante puede indicar, dependiendo del caso, que una versión anterior quede obsoleta y sea sustituida por una nueva versión, o que siga vigente y paralela a la última versión que se generó.

Cada versión se asocia con su versión predecesora por medio de la relación *previous,* excepto la versión inicial por no tener predecesora. La entidad *Specification* permite capturar la información acerca de la versión generada, tal como las causas del cambio ocurrido y un identificador de la persona responsable del registro de la versión. Cada versión tiene asociada a través de la relación *involves* a uno o más elementos afectados por un evento de cambio. Cada elemento susceptible de ser modificado, se representa por medio de la entidad *ChangedEntity*.

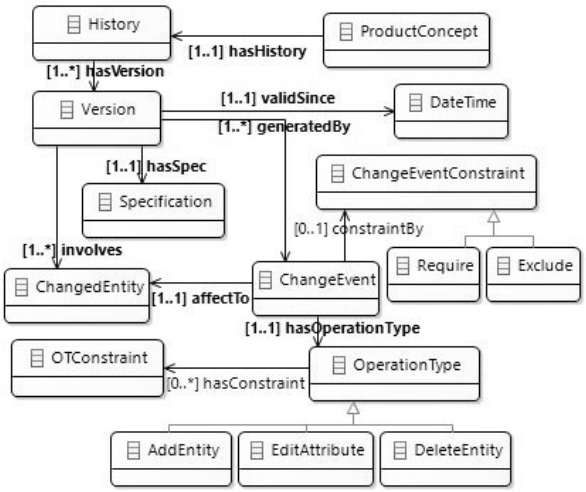


Figura 1. Modelo Conceptual para la gestión de versiones.

Un *ChangeEvent* se relaciona con un tipo de operación (*OperationType*) por medio de la relación *hasOperationType*. Cada evento de cambio (*ChangeEvent*) puede afectar a una entidad de cambio (*ChangedEntity)*, de tres formas diferentes: agregar una entidad (*AddEntity*), eliminar una entidad (*DeleteEntity*) o modificar atributos de una entidad (*EditAttribute*). Dependiendo del dominio de aplicación, no todos los elementos pueden ser afectados por todos los tipos de operaciones. Por este motivo, el modelo define la entidad *OTConstraint* para restringir que un tipo de operación *AddEntity*, *DeleteEntity* o *EditAttribute* pueda afectar o no a un *ChangedEntity* específico. Por el contrario, la propuesta define la entidad

ChangeEventConstraint para describir una relación de dependencia entre dos *ChangeEvent.* Un evento de cambio específico puede requerir (*Require*) o excluir (*Exclude*) un nuevo evento de cambio. Es decir, un evento de cambio causa otro evento de cambio. *ChangeEventConstraint* es muy diferente a la entidad *OTConstraint* ya que la primera es aplicada a la ocurrencia de un cambio y la segunda restringe la aplicación de un tipo de operación a una entidad específica. La definición de este conjunto de restricciones es necesaria para mantener la consistencia del modelo conceptual y asegurar la correcta interpretación del mismo.

El conjunto de entidades descriptas permite identificar los componentes afectados, cómo fueron modificados por un tipo de operación, la causa del evento de cambio y el instante en que una versión comienza a ser válida. Además, mediante el concepto de *History*, es posible capturar todas las versiones de una familia de productos durante su ciclo de vida.

La tercera etapa de la metodología, consistió en implementar formalmente en OWL (Ontology Web Language) [7], el modelo definido mediante la utilización de la herramienta Protégé 5.0. Inicialmente, se definió un espacio de nombre con el prefijo *asc*, para contener todos los identificadores únicos de los elementos de la ontología. Una vez definido el espacio de nombres, se identificaron todos los conceptos y se clasificaron para organizarlos en una estructura de jerarquía. Es decir, se crea una taxonomía donde la clase *OperationType* tiene como subclases a *AddEntity*, *DeleteEntity* y *EditAttribute*. Del mismo modo, la clase *ChangedEventConstraint* tiene las subclases *Require* y *Exclude.* Seguido de esta clasificación, se definieron los tipos de datos, las propiedades y los dominios y rangos de cada propiedad. La propiedad es la encargada de relacionar los individuos del dominio con los individuos del rango, formando una tripleta. Es decir, la propiedad *validSince* relaciona a un individuo de *Version* con un único individuo de *DateTime*, por ejemplo "*Version*_02 *validSince 22/08/2013*" indica que una versión es válida desde la fecha indicada. Luego se describieron las expresiones de cada clase especificando las clases equivalentes, las clases disjuntas, subclases y axiomas. Estas expresiones aseguran la consistencia del modelo conceptual, por ejemplo se definió una expresión de clase, como condición necesaria, para indicar que un "*ChangeEvent* debe estar asociado al menos con un *OperationType*" o que "una clase *ProductConcept* debe estar asociado a una clase *History*". Además, del conjunto de expresiones junto con las relaciones de herencia entre las clases, es posible inferir conocimiento que no está explícito en la taxonomía definida, por ejemplo inferir nuevas clases o clasificación de los individuos. La implementación OWL de estas expresiones, se puede ver con mayor detalle en [6].

Con el fin de definir de forma más precisa el comportamiento y la semántica de las relaciones, se utilizó el lenguaje SWRL. Este lenguaje extiende el conjunto de axiomas de OWL para incluir reglas condicionales que permitirán inferir nuevo conocimiento, asegurar un correcto uso e implementación de la información y definir formalmente el comportamiento de los recursos gestionados [5]. Una regla SWRL tiene dos partes, el antecedente y el consecuente. En este sentido, si todos los conceptos atómicos en el antecedente de una regla son verdaderos, entonces la consecuencia debe ser verdadera también. En la Tabla 1 se representa un conjunto de reglas limitado para inferir conocimiento sobre la ocurrencia de eventos de cambio, relaciones inversas y entidades de cambio que se ven afectadas, por ejemplo la regla 1 permite inferir que un elemento es incorporado a la información que representa la entidad *ProductConcept*. Además, se definió un conjunto de reglas para inferir las relaciones inversas. Un Ejemplo de esta definición, es la regla 2. Finalmente, la regla 3 permite inferir un elemento que fue modificado por la ocurrencia de un cambio.

Tabla 1. Conjunto Resumido de Reglas SWRL

N	Regla	Definición
1	ProductConcept(?f) ∧ History(?h) ∧ Version(?v) ∧ EventChange(?ch) ∧ GeneratedBy(?v,?ch) ∧ Add(?o) ∧ hasTypeOperation (?ch,?o) ∧ ChangedEntity (?e) → addEntity(?e,?f)	Esta regla infiere que una entidad de cambio *e* es agregada al ProductConcept *f* en una versión *v*.
2	ProductConcept(?*x*) ∧ History(?*y*) ∧ hasHistory(?*x*, ?*y*) → isHistoryOf(?*y*, ?*x*)	*x* es un ProductConcept asociado a history *y*. *y* es history de *x*.
3	EventChange(?*x*) ∧ EditAttribue(?*y*) ∧ hasOperation(?*x*,?*y*) ∧ ChangedEntity(?*t*,?z) → ChangedEntity(?z)	Un evento *x* que tiene un tipo de operación EditAttribute *y*. Afecta a un ChangedEntity z.

3. Análisis de la Propagación de Cambios en PRONTO y Feature Model

Esta sección se centra en describir la propagación de los cambios y una clasificación de los mismos, para analizar los eventos de cambios que ocurren en una ontología de productos (PRONTO) y en el modelo de características (FM). Estos enfoques representan la información de familias de productos, incluyendo las variantes de los mismos (variabilidad en el espacio). En el anexo del presente trabajo, se describe con mayor detalle la propuesta de PRONTO.

A continuación, se describe un sencillo ejemplo sobre un producto ficticio basado en la telefonía celular, con el objetivo de representarlo en los enfoques nombrados y, de esta forma, analizar los eventos de cambios y la pro-

pagación de los mismos. Para fabricar un teléfono celular, se requiere una configuración adecuada durante el proceso productivo, principalmente en la etapa de diseño, donde se establecerán las partes componentes del producto. El producto en cuestión, tiene una gran cantidad de componentes, pero para el análisis del presente trabajo, sólo se acota a 4 componentes: procesador, sistema operativo, cámara digital y flash. Por lo tanto, se considera la familia de productos denominada SE, compuesta de un procesador chipset Qualcomm msm 8227 CPU Snapdragon dual-core 1Ghz, cámara digital lateral 5Mp res 2592x1944, Flash LED con 0.5 d/s (disparos por segundo) y un sistema operativo SESO versión 2.3. Esta configuración es considerada como la versión inicial de la familia SE.

Para analizar el impacto y la propagación de cambios, se introduce un conjunto de requerimientos que surgen durante la etapa de diseño:

- Requerimiento 1: Incorporar una segunda cámara digital de 2Mp res 1733x1155 en el lado frontal del producto, previo a su inserción en el mercado.
- Requerimiento 2: Agregar una segunda unidad del componente flash para aumentar la fuente de luz y así mejorar la escena fotográfica.
- Requerimiento 3: Dado el lanzamiento de un nuevo procesador con mayor rendimiento, se remplaza el actual modelo (Qualcomm msm 8227) por Qualcomm msm 8230.
- Requerimiento 4: la versión actual del sistema operativo (*SESO2.3*) es una versión obsoleta, y se desea actualizar a la versión *SESO4.0*.

A partir de estos requerimientos, el ejemplo descripto es representado en el modelo de PRONTO y en el modelo de características, para su posterior análisis.

3.1. Propagación de los Cambios

Para comprender la propagación de cambios, se adoptó la definición de Griffin y colab. [8], quienes la definen como el proceso por el cual un cambio ocurrido en un sector o en un elemento particular, produce nuevos cambios en uno o más sectores o elementos. Un evento de cambio raramente ocurre de forma individual sin provocar nuevos cambios, dado la alta conectividad de los sistemas actuales.

En el ámbito de la industria, se requiere una gestión eficiente de los cambios con el fin de predecirlos y evitar que impacten de forma negativa en los costos, tiempos o recursos asignados durante el proceso de fabricación de un producto. Además, la creciente demanda de los clientes para obtener productos y servicios acorde a sus necesidades, implica cambios constantes en todas las etapas del ciclo de vida de un producto. Ecker y colab. en [9] diferencian dos tipos de cambios que se gestionan de una forma similar pero son originados de formas diferentes:

- Los cambios emergentes: aquellos que ocurren en respuesta a los problemas que surgen durante el proceso de diseño, tales como problemas en el diseño, en las pruebas, en el prototipo, en el proceso de fabricación y aquellos relacionados con la performance durante el uso del producto.
- Los cambios iniciales: surge en respuesta a los requerimientos del cliente para una nueva versión del producto al comienzo de proceso de diseño, tales como requerimientos de certificación, innovaciones en los materiales, problemas con diseños anteriores, entre otros. Sin embargo, también pueden surgir durante el transcurso del diseño debido a recientes innovaciones, nuevos requerimientos del cliente para agregar nuevos componentes luego de que el producto le fuese entregado.

Los cambios ocurren a lo largo de tres etapas del proceso de diseño. La primera de ellas, es la etapa donde se determina los requerimientos y se desarrolla un diseño conceptual, logrando un acuerdo con el cliente por medio de un contrato. La etapa siguiente es el periodo desde que se estableció el contrato hasta la entrega del producto ya fabricado y, la etapa final, es el periodo de uso o consumo del producto. Cada área o elemento susceptible de ser modificado por un evento de cambio puede responder de diferentes formas, ya sea transmitiéndolo a otro elemento o absorbiéndolo sin afectar a otros elementos o áreas. En [9] propone una clasificación basada en cuatro tipos de reacciones ante la ocurrencia de eventos de cambio y se resumen a continuación:

- *Constantes*: son aquellos elementos que no se ven afectados por un cambio, es decir no absorben ni transmiten a otros elementos, permaneciendo en su estado en todo momento.
- *Absorbedores*: son los elementos que absorben más cambio de los que pueden generar y transmitir a otros elementos.
- *Portadores*: son elementos que pueden absorber la misma cantidad de cambios que pueden transmitir a otros elementos.
- *Multiplicadores*: generan más cambios de los que pueden absorber, generando nuevos eventos de cambio.

En la sección 3.3, se aplicará al ejemplo ya descripto, esta clasificación con el fin de analizar el comportamiento de los eventos de cambio. Para evaluar la propagación de los cambios es necesario distinguir entre cambios *directos* y cambios *indirectos*. El primer caso, incluye a los elementos que son afectados porque otro elemento es afectado sin tener impacto en un tercer elemento. Los cambios indirectos ocurren cuando un tercer elemento es afectado indirectamente por un primer elemento. Esta clasificación permitirá predecir la cadena de cambios mediante dos actividades: la detección de las causas y los

efectos del evento de cambios ocurrido, permitiendo conocer el cambio que origina la cadena de propagación y el cambio directo o indirecto que es resultado del anterior.

3.2. Entidades de Cambio de PRONTO

La propuesta del modelo de PRONTO es analizada mediante el uso de la ontología de la sección 2, con el fin de representar la gestión de versiones en los tres niveles de la Jerarquía de Abstracción (AH) definida en la mencionada propuesta (Ver Anexo). Se especializa la entidad *ProductConcept* en *Family*, *VariantSet* y *Product*. De esta forma es posible capturar la historia, en términos de versiones, en los tres niveles propuestos por PRONTO. En cada uno de estos niveles, los cambios afectan a elementos distintos. Así, por ejemplo, en el nivel más alto las modificaciones están dadas en la estructura de los productos que forman la familia. En el nivel intermedio pueden ser cambiados la selección de componentes, así como las restricciones entre conjuntos de variantes, y en el nivel más bajo puede verse afectado la especificación de los productos concretos.

Para representar las diferentes entidades que pueden ser modificadas, así como el evento de cambio correspondiente, se especializa los conceptos *ChangedEntity* y *ChangeEvent* del modelo propuesto.

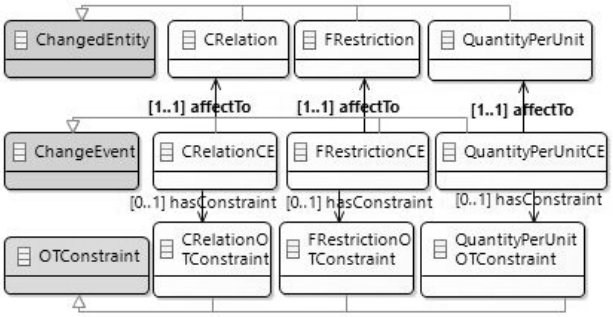


Figura 4. Entidades de cambio a nivel de Familia.

En la Figura 4 se muestran las entidades de cambio y los eventos de cambio definidos para el nivel de Familia de la AH. De esta forma, la entidad *ChangedEntity* se especializa en *FRestricion, CRelation/DRelation* y *QuantityPerUnit*/*ProductionFactor*. *FRestriction* representa el conjunto de restricciones a nivel de familia y *CRelation*/*DRelation* representan las relaciones de composición o de derivados entre componentes respectivamente. En la figura 4 únicamente se representa la entidad *CRelation* para una mejor legibilidad de la representación. La entidad *QuantityPerUnit* representa la cantidad de componentes que son necesarios para la manufactura de un producto. La entidad *ProductionFactor*, no se muestra en la Figura 4 para una mejor comprensión. Estas entidades son afectadas respectivamente por *FRestrictionCE, CRelationCE/ DRelationCE* y *QuantityPerUnitCE*, siendo éstas especializaciones de *ChangeEvent.* Cada evento de cambio tiene asociado un tipo de operación, que a su vez tiene asociado una restricción. Para el nivel de familia, se identificaron un conjunto de restricciones para indicar que sólo los tipos de operación *AddEntity* y *DeleteEntity* pueden afectar a una *FRestriction* (*FRestrictionOTConstraint*) o a una entidad *CRelation* (*CRelationOTConstraint*). Por el contrario, una entidad *QuantityPerUnit/ ProductionFactor* puede ser afectada únicamente por un tipo de operación *EditAttribute* (*QuantityPerUnitOTConstraint*).

En el nivel de conjunto de variantes (*VariantSet*) se identifica un conjunto de entidades de cambio y son especializaciones de *ChangedEntity* (Figura 5). Estas entidades se denominan como *VSRestriction* y *Change. VSRestriction* representa el conjunto de restricciones a nivel de conjunto de variantes y *Change* representa los cambios con respecto a la estructura de Familia asociada. *Change* está compuesta por un conjunto de cambios especializado en *FamilySpecification, FamilyRemoval* y *QuantityChange.*

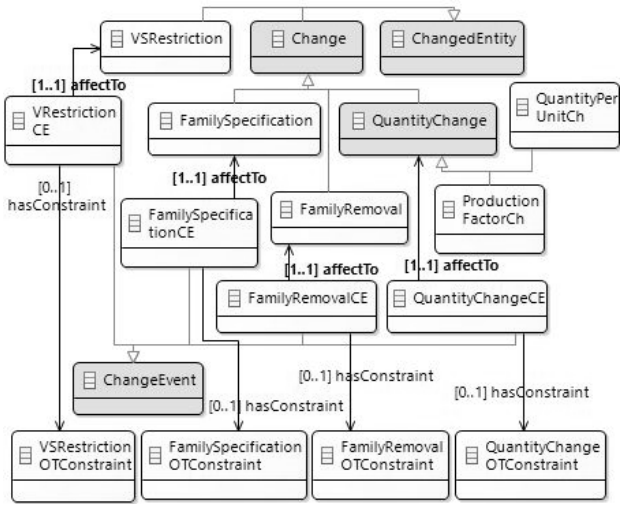


Figura 5. Entidades de cambio a nivel de C. de Variantes.

FamilySpecification permite representar el cambio en la selección de un componente dentro de varios alternativos, siempre y cuando exista un tipo de relación *AlternativeRel* a nivel de Familia. *FamilyRemoval* representa la eliminación de una relación *affectedRelation* de la estructura asociada. *QuantityChange* representa las modificaciones de los valores de los atributos *quantityPerUnit* o *productionFactor*, dependiendo si es una relación de composición o una relación de descomposición, respectivamente. En la Figura 5, se puede ver que cada una de

estas entidades de cambio a nivel de conjunto de variantes, son afectadas respectivamente por los eventos de cambios *VRestrictionCE, FamilySpecificationCE, FamilyRemovalCE y QuantityChangeCE*. Además, cada *EntityChanged* tiene asociado una restricción para controlar el tipo de operación que las afecta. Es decir, un atributo *quantityPerUnit* o *productionFactor* pueden ser editados por una instancia de *EditAttribute*. Por su parte, una entidad *VSRestriccion* puede ser agregada o eliminada, es decir pueden restringirse con los tipos de operación eliminar (*DeleteEntity*) y agregar (*AddEntity*).

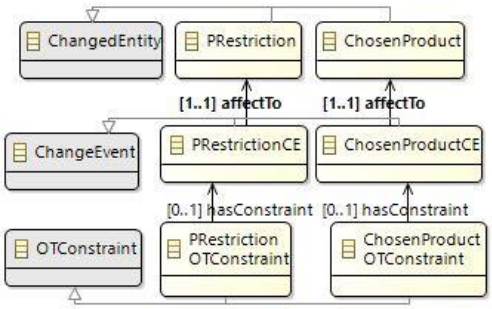


Figura 6. Entidades de Cambio a nivel de Producto.

En el nivel de *Producto* (ver Figura 6), las entidades de cambio identificadas son: *PRestriction* y la relación *chosenProduct,* que se reifica en una entidad, dado que es susceptible a los cambios en el modelo. *PRestriction* representa el conjunto de restricciones a nivel de producto y la entidad *chosenProduct* permite representar el vínculo de un producto con sus componentes. Ambas entidades pueden ser afectadas por los eventos de cambio *PRestrictionCE* y *ChosenProductCE*. Estas entidades de cambio están restringidas por las entidades *PRestrictionOTConstraint* y *ChosenProductOTConstraint* para indicar que los tipos de operación válidos son *AddEntity* y *DeleteEntity.*

3.3. Propagación de los Cambio en PRONTO

Para estudiar la propagación de los eventos de cambios, en la Figura 7 se representa el ejemplo descripto al comienzo de esta sección, mediante la instanciación de las entidades del modelo conceptual de PRONTO. La familia *SE* se representa mediante *Family SE* que posee una estructura de composición (*CStructure SE*). Las partes componentes se encuentran vinculadas por instancias de *CRelation.* Es decir, el componente *Processor* se vincula a la estructura mediante *CRelation1*, el componente *Camera* mediante *CRelation2*, el componente *CameraFlash* por medio de *CRelation3* y *SO* a través de *CRelation4.* Cada una de estas relaciones de composición se asocia a un *Value* por medio de *quantityPerUnit,* para indicar cuantas unidades se requieren de cierto componente en el proceso de producción. Para facilitar la comprensión de la Figura 7, únicamente se muestra el componente *Processor*, ya que las otras partes componentes se representan de forma similar. En el nivel de conjunto de variantes se identifica la variante *SEMH*, miembro de *Familia SE*, que adopta la estructura de composición de la familia *SE* a través de la relación *has. SEMH* se compone de otros conjuntos de variantes como: *Qualcomm msm 8227, Lateral Camera 5Mp, FlashLED* y *SESO.* Estos conjuntos de variantes son miembros respectivos de *Processor, Camera, CameraFlash* y *SO.* En la Figura se representa únicamente el conjunto de variante miembro de *Processor* (*Qualcomm msm 8227*).

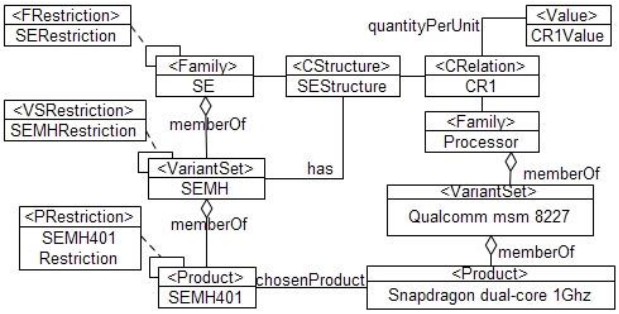


Figura 7. Representación en PRONTO del Producto *SE*.

A su vez, en el nivel de Producto, se identifica al producto concreto *SEMH401* miembro de *SEMH*, del cual infiere la estructura de composición. Ésta estructura posee los siguientes componentes: *snapdragon dual-core 1GHz, Camera lateral 5Mp res 2592x1944px, Flash Led 0.5d/s* y *SESO2.3*, y cada uno de los productos son miembros respectivamente de los conjuntos de variantes mencionados.

Un vez descripto el caso mediante PRONTO, analizamos los requerimientos 1 y 2 mencionados al inicio de la sección 3. Estos requerimientos se aplican a nivel de familia para analizar su propagación hacia los siguientes niveles (Conjunto de variantes y Producto). Además, es importante resaltar que los cambios siempre se aplican a la última versión válida para el análisis de este caso.

El primer evento de cambio se clasifica como un cambio inicial, según la clasificación en [9], dado que se desea dar respuesta al requerimiento 1 innovando el producto antes de ser liberado al mercado. Este cambio genera una nueva versión de la familia (*SEv2*) e implica agregar una nueva relación de composición (*CRelation*) a la estructura para vincular al nuevo componente (*FrontCamera*).

En la Figura 8 se representa tal situación mediante su instanciación en la ontología de gestión de versiones. De este modo, la familia *SE* es un individuo de la clase *ProductConcept* asociada a su historial *SEHistory* compuesta de dos versiones *V1* y *V2*. La nueva versión (*V2*) es gene-

rada por *CRelation5CE,* el cual es individuo *de Event-Change.* Este evento de cambio afecta mediante el tipo de operación *AddCRelation5* a la entidad de cambio *CRelation5*, indicando que se agrega una nueva relación de composición del componente *FrontCamera.* Además, *CRelation5CE* posee una restricción asociada (*inferVSStructre*) para indicar que una nueva versión de conjunto de variante debe inferir la estructura generada para incluir el nuevo componente.

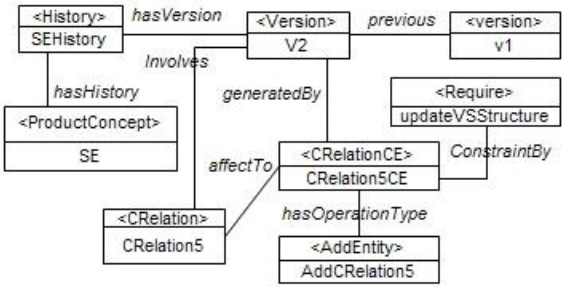


Figura 8. Representación en la ontología de versiones.

Dada la situación anterior, se genera una nueva versión *SEMHv2* para incluir el componente *FrontCamera 2Mp*, mediante la selección de la nueva estructura de la familia a la que pertenece. Es decir el cambio que tiene lugar a nivel de familia se propaga al nivel de conjunto de variantes. Sin embargo, la versión inicial del conjunto de variantes *SEMHv1* sigue siendo válido con la nueva versión de familia sin incluir el nuevo componente. El análisis de compatibilidad será explicado con más detalle en la sección 3.4.

La nueva versión del conjunto de variantes *SEMHv2*, es compatible con la versión inicial (*v1*) del producto *SEMH401*. Sin embargo, para incluir el componente *FrontCamera 2Mp 1733x1155px* a nivel de producto, es necesario inferir la nueva estructura donde la relación *chosenProduct* asocia el nuevo componente. Este evento genera una nueva versión de producto (*SEMH401v2).* La propagación del evento de cambio se clasifica como un evento de tipo *Portador*, dado que absorbe y transmite el mismo número de eventos de cambios. La cadena de propagación del evento de cambio analizado, tiene origen a nivel de familia e impacta de forma directa en el nivel de conjunto de variantes e indirecta en el nivel de producto.

Este mismo efecto de propagación, ocurre si el evento de cambio impacta en la entidad *QuantityPerUnit* en el nivel de familia. Para este caso vamos a considerar el requerimiento 3, cual surge en base a un problema detectado en la etapa de prueba, es decir un cambio emergente durante el transcurso del proceso de diseño.

El evento de cambio genera una nueva versión de familia *SEv3* e impacta en la entidad *QuantityPerUnit* editando el valor *Number* = 2. Este valor indica que se requieren dos unidades del componente *Flash* para la fabricación del producto. En la Figura 9, se representa en la ontología de versiones la propagación de este cambio al siguiente nivel, generando una nueva versión a nivel de conjunto de variantes (*SEMHv3),* donde el cambio tiene lugar en la entidad *QuantityPerUnitCh* (subclase de *QuantityChange).* Este evento de cambio no genera una nueva versión a nivel de producto, dado que el número de componentes requeridos se infieren desde la estructura de composición. Por lo tanto, para este caso la versión *SEMHv3* sigue siendo compatible con la versión *v2* del producto (SEMH401v2).

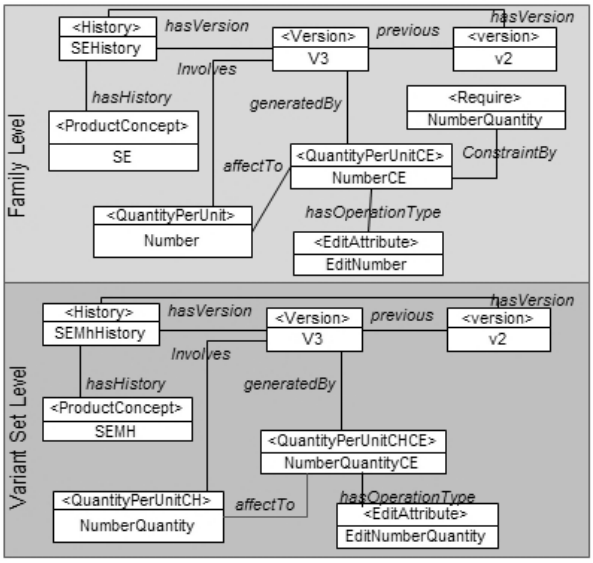


Figura 9. Representación de la propagación de cambios en la ontología de versiones.

La clasificación para este evento es de tipo *Portador* y afecta de forma directa a la entidad *QuantityPerUnit Ch*. Para esta situación se define la siguiente regla para indicar que al editar la entidad *QuantityPerUnit* a nivel de familia, genera un nuevo evento de cambio a nivel de conjunto de variantes afectando la entidad *QuantityPerUnitCH* (ver Figura 10).

QuantityPerUnit(?o) ∧ QuantityPerUnit (?oc) ∧ affectTo(?o, ?oc) ∧ editAttribute(?ot) ∧ hasOperationType(?oc,?ot) ∧ QuantityPerUnit OTConstraint(?vc) ∧ hasConstraint(?vc, ?ot) → QuantityPerUnitCh(?n) ∧ QuantityChangeCE (?m) ∧ affectTo(?m,?n)

Figura 10. Regla de propagación para el ejemplo analizado.

Para analizar la propagación de cambios con origen en el nivel de conjunto de variantes, se considera el requerimiento 3, donde el evento de cambio genera un

nuevo conjunto de variantes de la familia. En este trabajo se considera tal situación como una nueva versión denominándola *SEMHv4*. Se requieren dos eventos de cambio que afectan a la estructura de composición. El primero consiste en eliminar un componente *Qualcomm msm 8227* y el segundo, agregar el nuevo componente *Qualcomm msm 8230*. Para esta situación, el conjunto de variantes SEMHv4 ya no es compatible con la versión de producto *SEMH401v2* y el evento se propaga a nivel de producto generando una nueva versión del mismo (*SEMH401v3)*. En esta nueva versión, el cambio ocurrido afecta a la estructura de composición, donde la entidad *chosenProduct* debe remplazar el componente *Snapdragon dual-core 1GHz* por el nuevo componente *Adreno dual-core 1GHz*. Este evento se clasifica como *Portador*, dado que recibe y transmite la misma cantidad de eventos de cambios.

Para analizar el impacto a nivel de producto, se tiene en cuenta el requerimiento 4, el cual solicita la actualización del sistema operativo y se clasifica como un evento de cambio del tipo *inicial* dada a una reciente innovación. Este cambio genera una nueva versión del producto denominada *SEMH401v4*. En esta nueva versión se actualiza el sistema operativo *SESO2*.3 por *SESO4.0*. Para este caso, consideramos que el evento de cambio es compatible con los niveles superiores y es válida su aplicación, dado que el evento de cambio no afecta a los niveles superiores. Sin embargo, es necesario evaluar si la nueva versión del sistema operativo es compatible con las otras partes componentes. Este cambio no tiene impacto en otras entidades, por lo que no existe una propagación de eventos a nivel de producto y se clasifica como un cambio *absorbedor*.

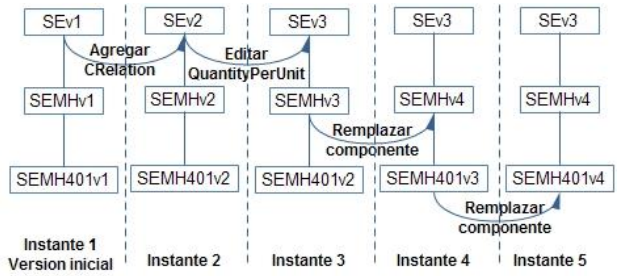


Figura 11. Propagación de Cambios en PRONTO.

En la Figura 11 se representa los 5 instantes donde se generaron las nuevas versiones en diferentes niveles de la AH de PRONTO. Estas versiones surgieron en base al conjunto de cambios descriptos anteriormente. Para tal situación, se observa que la propagación de cambios va desde arriba hacia abajo, es decir los eventos de cambios que afectan las entidades analizadas a nivel de familia impactan en las entidades de cambio a nivel de conjunto de variantes y a nivel de producto. Por el contrario, esto no ocurre desde el nivel de producto hacia el nivel de familia.

En la figura se representan los eventos de cambios principales que generan la nueva versión, y dependiendo de cada situación en particular, una versión podría tener uno o más eventos de cambios.

Además de las entidades de cambio analizadas, existen otras entidades tales como: *FRestriction*, *VSRestriction* y *PRestriction* que son del tipo *absorbedor*, dado que un cambio que impacta en cada una de ellas, no se transmite a otras entidades. Es decir, se puede incorporar o eliminar una restricción sin afectar otras entidades de cambio que pertenezcan al mismo nivel o diferente nivel. Estas restricciones se encargan de mantener la consistencia de la información de producto y, de esta forma, evitar que un diseñador cometa errores u omita la selección de algún componente durante el proceso de diseño del producto.

3.4. Compatibilidad de las versiones

Una vez identificado el cambio y realizado el análisis de su impacto y comportamiento, existe una fase muy importante, que es la evaluación de la nueva versión resultante. Esta fase permitirá evaluar, en todos los niveles de PRONTO, si la nueva versión creada es compatible con otras versiones o no.

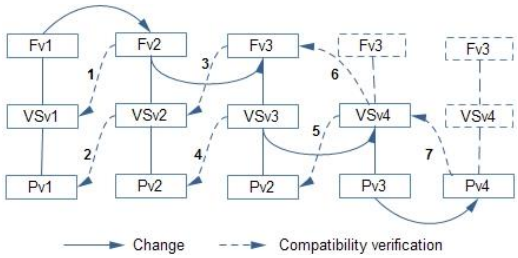


Figura 12. Compatibilidad de versiones.

Para analizar la compatibilidad de versiones que surgieron en el caso analizado, en la Figura 12 se representa con línea dirigida continua, los eventos de cambio, y con línea dirigida punteada los instantes donde se verifica la compatibilidad de versiones. Para esta última situación, los instantes 1, 2, 3 y 4 verifican la compatibilidad de versiones de forma descendente, dado que los eventos de cambios afectan a los niveles inferiores. El análisis de compatibilidad de versiones inicia desde el nivel de familia (*F*) y se verifica la compatibilidad con la versión anterior del nivel de conjunto de variantes *VS* (instantes 1 y 3). Lo mismo ocurre a nivel de conjunto de variantes con la versión anterior del nivel de productos *P* (instantes 2,4 y 5). Para estos 5 instantes, ninguna de las versiones previas es compatible con la versión analizada, dado que

los eventos de cambios introducen o modifican elementos de cambio que modifican la estructura a nivel de familia y conjunto de variantes, propagándose a los niveles inferiores, generando nuevas versiones de los mismos.

En los instantes 6 y 7 se verifica la compatibilidad de versiones de forma ascendente, donde uno o más cambios que dan origen a una nueva versión a nivel de conjunto de variantes o a nivel de productos, son compatibles con versiones anteriores de niveles superiores, dado que los cambios no los afectan. A partir del instante 3, la versión de la familia se mantiene para los instantes 4 y 5. Es decir, los eventos de cambios a nivel de conjunto de variantes y nivel de producto (instantes 4 y 5 respectivamente) no afectan a la versión en el nivel de familia. Del mismo modo ocurre en el instante 5, donde la nueva versión de producto no afecta a los niveles superiores. Es decir, una nueva versión a nivel de conjunto de variantes *VSv4* es compatible con la versión anterior de la familia *Fv3*, o una nueva versión de producto *Pv4* es compatible con la versión *VSv4* y *Fv3* (ver rectángulos con líneas de punto en la Figura 12), dado que adquieren o heredan la estructura de niveles superiores y de esa forma mantiene la consistencia en la representación de información de producto.

Para el caso analizado, la versión de familia *SEv3* es compatible con las versiones de los conjuntos de variantes *SEMHv3* y *SEMHv4*. Del mismo modo, la versión del conjunto de variantes *SEMHv4* es compatible con las versiones de producto *SEMH401v3* y *SEMH401v4*. Analizado lo anterior, es importante destacar que cada familia, conjunto de variantes y productos son instancias de *ProductConcept* y se asocian a un historial que contiene el conjunto de versiones. Por ello, es necesario un conjunto de reglas en SWRL para conectar los eventos de cambios que ocurren en un nivel y se propagan a otros niveles manteniendo la consistencia.

3.5. Entidades de Cambio de FM

En esta sección se representa la gestión de versiones del modelo se características o Feature Model por medio de la ontología de la sección 2. Este enfoque utiliza una estructura de árbol para representar la información de familia de producto (individuo de *ProductConcept)* y está compuesto de un conjunto de características, relaciones y restricciones. En las referencia [10] y [11] se detalla con mayor profundidad el modelo de características.

En la Figura 13 se identifican las entidades mencionadas por medio de la especialización del concepto *ChangedEntity* en: *Feature*, *Relationship* y *Restriction*. Estas tres entidades representan respectivamente a los conjuntos de características, de relaciones y de restricciones, dado que son conceptos susceptibles de ser modificados por un evento de cambio. Una relación puede ser de 5 tipos, especializándose en: *ConsistOf*, *Optional*, *Mandatory*, *Selection* y *Alternative*. Por su parte, una restricción puede ser de dos tipos: *Require* o *Exclude*. La primera indica que una característica requiere de otra característica, y la segunda indica que la selección de una característica excluye a otra.

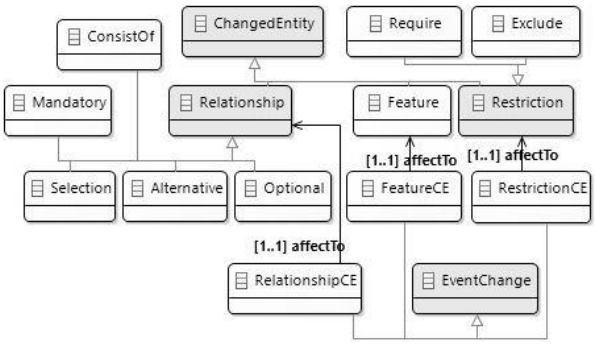


Figura 13. Entidades de Cambio de Feature Model.

Como ya se mencionó, el concepto de *ChangeEvent* permite representar los eventos de cambio que afectan a las entidades de cambio (*ChangedEntity*). Por lo tanto, *ChangeEvent* se especializa en las entidades *FeatureCE*, *RelationshipCE y RestrictionCE*, A su vez, *RelationshipCE*, se especializa en *ConsistOfCE*, *MandatoryCE*, *OptionalCE*, *AlternativeCE* y *SelectionCE*. Estas entidades no se muestran en la figura para mayor claridad. Finalmente, la entidad *RestrictionCE* se especializa en *RequireCE* y *ExcludeCE*.

3.6. Propagación de los Eventos de Cambio en FM

Con el fin de analizar la propagación de los cambios en FM, en la Figura 14a se representa un conjunto de características en una estructura de árbol, para describir el caso presentado en la sección 3.

La versión inicial se compone de una característica raíz para representar la familia *SE* y se compone de las características hijas *Processor*, *Camera*, *CameraFlash 0.5 d/s* y *SO*, siendo estas características, obligatorias para la configuración del producto.

La característica *Processor* consiste de la característica *Snapdragon dual-core 1Ghz*. La característica *Camera* consiste de *Lateral Camera 5Mp res 2592x1944* y la característica *SO* consiste de *SESO 2.3*. Para analizar la propagación de cambios en este enfoque, primero se considera el requerimiento 1, el cual genera una nueva versión de la familia (*SEv2*). El evento de cambio afecta a la relación *ConsistOf*, que vincula las características *Camera* y *Lateral Camera 5Mp res 2592x1944*, eliminándola y una relación del tipo *Mandatory*. Del mismo

modo este evento genera dos nuevos cambios: agregar la característica *FrontCamera 2Mp1733x1155* y una relación de obligatoriedad. Esta situación se representa en la ontología de versiones en la Figura 15, donde el árbol de características es un individuo de la clase *ProductConcept* y se lo denomina *FeatureTree.* Este individuo tiene un historial (*FeatureTreeHistory*) que se compone de 2 versiones (*v1* y *v2*). La nueva versión (*v2*) es generada por un evento de cambio (*ConsistOf1CE*) que afecta a la entidad *ConsistOf1*. Este evento de cambio posee una restricción que genera nuevos eventos de cambio para agregar dos relaciones de tipo obligatoria y una característica. Estos eventos de cambios se representan en la figura como: *Mandatory5CE*, *Mandatory6CE* y *FrontCameraCE,* los cuales afectan respectivamente a las entidades de cambio: *Mandatory5*, *Mandatory6* y *FrontCamera2Mp 1733x 1155.*

Los requerimientos 2, 3 y 4, no se representan en la ontología de gestión de versiones, por cuestiones de espacio. Sin embargo estos 3 requerimientos tienen una representación similar al del requerimiento 1.

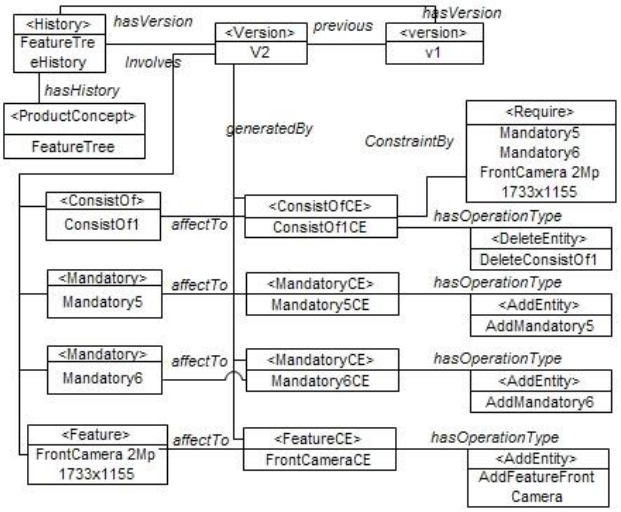


Figura 15. Propagación de Cambios en FM.

Este cambio se clasifica como *multiplicador*, dado que genera un número mayor de eventos de cambio que el número de eventos que absorbe (Ver Figura 14b). Con el fin de representar el cambio causado por el requerimiento 2, se introduce una restricción para controlar el número necesario de componentes flash. Para este caso, en la Figura 13c se representa tal evento con líneas punteadas y consiste en agregar la restricción *UnitRestriction*. Este evento de cambio genera la versión 3 de la familia de productos *SE* (*SEv3*) y se clasifica como *absorbedor.*

El requerimiento 3 introduce un cambio de tipo Portador e implica eliminar la característica *Qualcomm msm 8227* y agregar la característica *Qualcomm msm 8230*, generando la versión 4 de la familia *SE* (ver Figura 14d). Finalmente, el requerimiento 4 genera una nueva versión de familia *SEv5* (Figura 14e), donde el evento de cambio requiere eliminar la característica *SESO2.3* y agregar una nueva característica *SESO4.0.* Este requerimiento es de tipo *portador*.

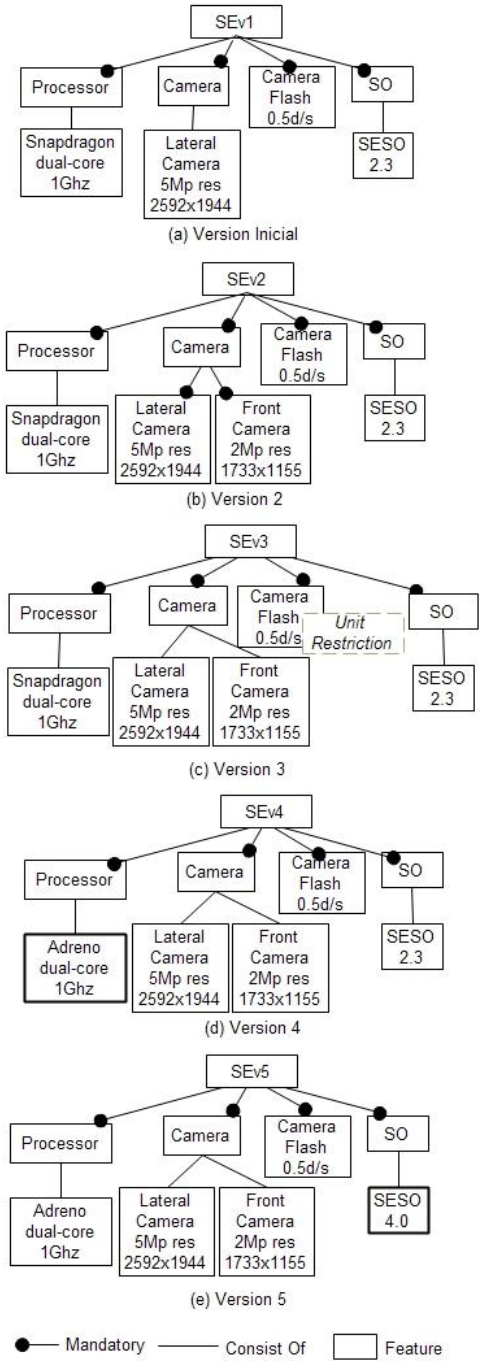


Figura 14. Propagación de Cambios en FM.

La propagación de los cambios puede controlarse mediante instancias de la entidad *EventChangeConstraint,*

ya sea para indicar que se requiere un nuevo evento de cambio o no. Es importante mencionar que la familia de producto en FM se representa íntegramente como instancia de *ProductConcept*, a diferencia de PRONTO que cada nivel de la AH son individuos diferentes de *Product Concept*.

4. Conclusiones

El desarrollo del trabajo permitió analizar a dos modelos de representación de familias de productos basando en una ontología de gestión de versiones. Ambos modelos poseen un conjunto de entidades de cambio que se encuentran enlazadas una a otras conduciendo a la propagación de los eventos de cambio. Estos eventos pueden comportarse e impactar de diversas formas sobre el conjunto de entidades, ya sea re transmitiendo el cambio, absorbiéndolo o multiplicándolo. Por esta razón, es importante estudiar el comportamiento de los cambios para gestionarlo de forma eficiente y mantener la consistencia de la representación de la información de productos en el tiempo y espacio.

Con respecto a la propagación de cambios y gestión de versiones analizadas en PRONTO y FM, se obtuvo como consecuencia que ambas implementaciones son muy diferentes entre sí. Para PRONTO, se crean versiones independientes de los niveles de la AH, y mediante un conjunto de reglas, es posible mantener la consistencia durante la propagación de los cambios a través de esos niveles. Sin embargo, durante la implementación del modelo de características, no es posible crear versiones independientes, dado que las versiones surgen por eventos de cambios que involucran a todos los niveles de un árbol de característica, siempre perteneciendo a una misma versión.

Como trabajo futuros, se desea definir un conjunto de reglas que permitan mantener la consistencia entre los diferentes niveles de PRONTO y la compatibilidad de versiones de los mismos. Además, se desea implementar la propuesta en un prototipo con el fin de validar los requerimientos de la gestión de versiones y de este modo completar la etapa 4 de la metodología adoptada. Finalmente, se implementará la propuesta en otros casos de estudio, con el fin de identificar comportamientos repetitivos de los eventos de cambios y de esta forma, definir patrones de comportamientos para una gestión eficiente de los cambios. Esta gestión deberá conducir a la predicción de los efectos de tales cambios y sus dependencias, con el fin de administrarlos mediante un proceso de gestión de cambios.

5. Agradecimientos

Este trabajo ha sido financiado en forma conjunta por CONICET, la UTN (PID 25-O156) y la Universidad Nacional de La Rioja. Se agradece el apoyo brindado por estas instituciones.

6. Referencias

[1] Matsokis A. Kiritsis D. An ontology-based approach for products lifecycle management. Computer in Industrial 61 787-797. (2010)

[2] Asikainen T., Mannisto T., Soininen T. Kumbang: A domain ontology for modeling varia-bility in software product families. Advances engineering informatics. (2006)

[3]. Pohl K., Bockle G., Van Der Linden F. Software Product Line Engineering. Foundations, principles, and Techniques. ISBN-10 3-540-24372-0 Springer Berlin Heidelberg New York. (2006)

[4]. Vegetti, M., Leone, H., Henning, G. PRONTO: An ontology for comprehensive and consistent representation of product information. Engineering Applications of Artifi-cial Intelligence 24 (8), pp. 1305-1327. (2011)

[5] Horrock I., Patel-Schneider P., Boley H., Tabet S., Grosof B., Dean M.. SWRL: a Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission. (2004) http://www.w3.org/Submission/SWRL/ (accedido 25/09/2014)

[6] Sonzini S.,Vegetti M., Leone H.Toward an Ontology for Product Version Management. International Journal of Product Lifecycle Management. En prensa (2014)

[7] Deborah L. McGuiness and Frank van Harmelen. OWL Web Ontology Languaje. Overview. W3C recomendation.(2004). http://www.w3.org/TR/owl-features/ (accedido 26/09/2014)

[8] Griffin M., Weck O., Bounova G. Change Propagation Analysis in Complex Technical System. Journal of Mechanical Design. Vol. 131/081001-1. (2009)

[9] Ecker C., Clarkson J.P., Zanker W. Change and Customization in complex engineering domains. Research in Engineering Design. Volume 15, Issue 1 , pp 1-21. (2004)

[10] Kang K.C. Lee K., Lee J.: Feature Oriented Product line Software Engineering: Principles and guidelines. Domain-Oriented Systems Development: Practices and Perspectives. Cap 2. ISBN 0203711874. (2003)

[11] Kang K., Lee J., Donohoe P.: Feature-Oriented Product Line Engineering. IEEE software. (2002)

7. Anexo: PRONTO

PRONTO permite la representación de datos de productos en diferentes niveles de abstracción y en diferentes dominios de la industria [4]. Seleccionamos esta ontología por su característica de representación de variantes en la información de producto de forma consistente.

PRONTO define una jerarquía estructural (SH- Structural Hierarchy) para representar de forma eficiente la información concerniente a los productos y componentes que participan en la manufactura de productos y una jerarquía de abstracción (AH- Abstraction Hierarchy) que permite la representación de información no estructural de productos en diferentes niveles de abstracción y la representación de procesos de agregación y desagregación de información entre estos niveles. (Ver Figura 16)

La AH consta de 3 niveles: nivel de Familia (Family), nivel de Conjunto de Variantes (VariantSet) y nivel de Producto (Product). Estos tres niveles se relacionan entre sí mediante una asociación "memberOf", indicando que las entidades comprendidas en un nivel inferior, son miembros de una instancia de abstracción de un nivel superior. La SH considera dos tipos de relaciones de estructuras, que se especializan como "componentOf", para aquellas estructuras que relacionan al producto con sus partes componentes, y "derivateOf", para aquellas estructuras que enlazan al producto con sus derivados constituyentes. Esto quiere decir que una entidad puede tener otras entidades como componentes o derivados siempre en el mismo nivel de abstracción, lo cual permite representar las diferentes BOM (Bill Of Materials) de productos que son manufacturados por medio del ensamblado de partes componentes o la desagregación de materias primas no atómicas.

Una familia representa a un conjunto de productos que son similares en cuanto a su información y su estructura. Una familia puede tener una estructura simple (*SFamily*), o compuesta (*CFamily*). La primera de ellas hace referencia a aquellas familias que no tienen estructura. Por el contrario, una familia compuesta tienen al menos una estructura (*Structure*) asociada. Las familias compuestas pueden asociarse a dos tipos de estructuras: estructuras de composición (*CStructure*), donde el producto es obtenido por el ensamblado de productos como partes componentes, y estructuras de descomposición (*DStructure*), donde es posible obtener varios productos derivados por medio del proceso de desensamblar un producto. Dado los tipos de estructura, para identificar las partes de una *CStructure* o *DStructure*, se identifican dos tipos de relaciones: *CRelation* para identificar los componentes y *DRelation* para identificar los derivados. Estas relaciones se asocian a un valor *Value* por medio de la relación *quantityPerUnit*, para *CRelation*, y *productionFactor*, para *DRelation*. Estas asociaciones representan respectivamente el número de partes componentes necesarias para la fabricación del producto y el número de productos intermedios que se obtienen de la descomposición. Cada relación puede clasificarse de 3 formas:

- Opcional: el componente o derivado deben estar o no presentes en una estructura.
- Obligatorio: el componente o derivado debe estar presente en la estructura.
- Alternativo: el componente o derivado debe ser seleccionado de un conjunto de componentes, pero sólo un miembro debe ser parte de una estructura.

Por su parte, el nivel de conjunto de variantes representa el número de variantes de un producto, que poseen a una misma familia, pero introducen pequeñas modificaciones en su estructura. Del mismo modo que el nivel de Familia, a nivel de conjunto de variantes existen estructuras simples (*SVariantSet*) y estructuras compuestas (*CVariantSet*). Esta última, se asocia a la estructura de la familia a la que pertenece como miembro, por medio de la relación *Has*.

Una vez identificada la estructura asociada y la familia a la que pertenece un conjunto de variantes, es posible definir un conjunto de cambios (*ChangeSet*) para representar las modificaciones que se aplican a la estructura asociada. Los diferentes cambios se especializan en *QuantityChange*, *FamilyRemoval* y *FamilySpecification. QuantityChange* se especializa en *quantityPerUnitCH* y *productionFactorCH* para representar modificaciones en los valores asociados a *quantityPerUnit* y *productionFactor* de a relación *affectedRelation. FamilyRemoval* representa la exclusión de la relación *affectedRelation* desde la estructura de la cual pertenece. Finalmente, *FamilySpecification* está relacionada a los cambios en la elección de una sola alternativa de la estructura asociada. Por su parte, el nivel de producto representa de forma concreta al producto y pertenece a un único conjunto de variantes. Del mismo modo que los niveles anteriores, Cada producto puede tener dos tipos de estructuras: simple (*SProduct*) o compuesta (*CProduct*). Para identificar las partes componentes o derivados de una estructura *SProduct* se define la relación *chosenProduct.*

Finalmente, PRONTO considera 3 tipos de restricciones (*FRestriction, VSRestriction, PRestriction*) según el nivel de la AH para restringir entidades que se encuentran en un mismo nivel.

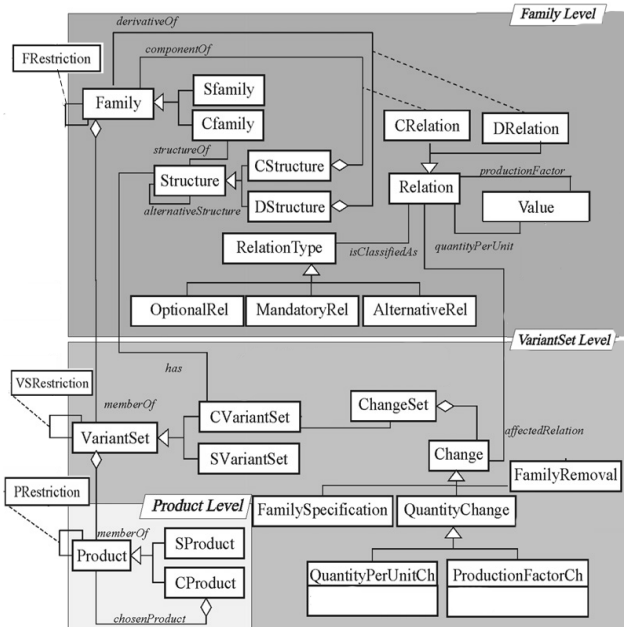


Figura 16. Modelo de PRONTO

Medición y Evaluación de Transformaciones a Nivel Metamodelo: un Enfoque Orientado al Mantenimiento

Abdelahad Corina N.,Miranda Enrique A.,Perez Norma B.
Universidad Nacional de San Luis
Departamento de Informática
{cabdelah,eamiranda,nbperez}@unsl.edu.ar

Resumen

Desde hace algunos años, la Ingeniería Dirigida por Modelos está en auge y junto con ella han surgido diversas propuestas de lenguajes de transformación de modelos. QVT Relation es un componente de QVT y permite formalizar transformaciones dirigidas por modelos. Teniendo en cuenta el aumento de la utilización de estos lenguajes, es importante comenzar a considerar aspectos relacionados al mantenimiento, re-utilización y comprensión de las transformaciones, ya que estas desempeñan un papel fundamental proporcionando un mecanismo para expresar el refinamiento de modelos. En este trabajo se propone una estrategia de medición y evaluación de transformaciones QVT a nivel metamodelo, la cual asistirá al ingeniero de software a llevar a cabo tareas de mantenimiento y evolución en las mismas.

1. Introducción

La Ingeniería Dirigida por Modelos (Model-Driven Engineering- MDE)[7, 24] ofrece un escenario ideal para potenciar el papel de la trazabilidad en el desarrollo de software. MDE propone un proceso de desarrollo de software en el cual la clave son los modelos y las transformaciones entre ellos. En este proceso, el software es desarrollado construyendo uno o más modelos, y transformando estos en otros modelos o transformarlos hasta llegar a código ejecutable. La transformación de modelos se refiere al proceso de transformación (relaciones y mapeo) de elementos de un modelo a elementos correspondientes de otro modelo. Los modelos son especificados a través de los metamodelos los cuales definen su sintaxis. Es decir, un metamodelo define el lenguaje con el cual se construyen modelos. Una transformación de modelos toma como entrada un modelo acorde a un determinado metamodelo y produce como salida otro modelo acorde también a un determinado metamodelo. Más precisamente, y de acuerdo con la definición de Kleppe et al [12]: *"Una transformación es la generación automática de un modelo destino a partir de un modelo origen de acuerdo a una definición de transformación"*. Teniendo en mente este objetivo varias organizaciones proponen lenguajes para expresar consultas y definir transformaciones entre modelos.

Debido al importante crecimiento en la utilización de los lenguajes de transformación, es imprescindible comenzar a contemplar tareas de mantenimiento y evolución en dichas transformaciones. En este contexto, las transformaciones poseen dificultades similares vinculadas a las tareas de mantenimiento y evolución en sistemas escritos en lenguajes de programación de propósitos generales como Java, C, C#, etc.

Por lo general, los desarrolladores de software tradicionales toman acciones correctivas como refactorizaciones [8] o revisiones de código para con-

servar el código en un estado mantenible. Estos indicadores aun no se hacen presentes en los lenguajes de transformación declarativos como QVT Relations (QVT-R). Sin embargo, algunas investigaciones iniciales se dirigen a las métricas para los lenguajes de programación funcional como Lisp o Haskell. Teniendo en cuenta que los lenguajes de transformación declarativos son parte de la familia del paradigma funcional, algunas métricas para este tipo de paradigma pueden servir como punto de partida para la definición de métricas para los lenguajes de transformación declarativos [11].

La definición de métricas puede ayudar a medir el mantenimiento, comprensión y reutilización de las transformaciones de modelos. Estas métricas definen como representar y obtener los valores de los atributos. A través de los indicadores se define como se interpretarán los valores de estos atributos y además especifica e implementa la evaluación de la métrica.

Teniendo en mente este objetivo, en este trabajo se definen criterios y métricas las cuales pueden ser utilizadas para medir y evaluar la comprensión, mantenimiento y el reuso de las transformaciones relacionales. Además se muestra que la dificultad de comprender, mantener y reutilizar aumenta a medida que las transformaciones crecen y las relaciones individuales se vuelven más complejas. Para esto, en base a nuestra experiencia con QVT-R, se analizaron las ideas de [11]. Además se muestra un caso de estudio al cual se le aplican las métricas especificadas en este trabajo.

El trabajo se estructura en secciones. La sección 2 describe los trabajos relacionados. La sección 3 introduce los lenguajes de transformación, en particular QVT. La contribución de este artículo se presenta en la sección 4. En la sección 5 se presenta un caso de estudio al cual se le aplican las métricas definidas en la sección 4. Finalmente, en la sección 6 se presentan las conclusiones.

2. Trabajos Relacionados

Son muchas las investigaciones, en el estado del arte, que se centran en medición de la calidad. En este contexto, se han definido nuevas métricas a través de técnicas y estrategias que permiten llevar adelante un proceso de medición más específico. La mayoría de los estudios que definen métricas cubren un amplio conjunto de paradigmas y lenguajes de programación como: lenguajes OO, procedurales, lógicos, funcionales, entre otros.

A continuación se describe, de manera sucinta, los trabajos más relevantes en este campo y que han motivado el desarrollo de este trabajo.

Luis Reynoso et al. [20] en su artículo persiguen dos objetivos principales: (i) Proponen un conjunto de métricas para medir las propiedades estructurales de las expresiones OCL (Object Constraint Language). (ii) Proponen una validación teórica de las métricas a fin de asegurar la correctitud de las mediciones empleadas.

Mohagheghi y Dehlen [15] presentan un proceso de 7 pasos sobre cómo definir un framework de calidad que se adapte a MDE, y que integra la ingeniería de calidad con evaluación de la calidad. A modo de ejemplo, aplican el framework sobre la calidad de transformación.

Amstel et al. [22] presentan las primeras investigaciones sobre calidad en transformaciones de modelos. Con el transcurso de los años, este grupo de investigación propuso diferentes enfoques centrados en transformación de modelos [21], utilizando lenguajes como ASF + SDF [23] y ATL [10].

Lucia Kapova et al. [11] proponen un conjunto inicial de indicadores de calidad que permiten evaluar las transformaciones escritas en QVT-R. Estos indicadores se aplicaron a diversas transformaciones con el objeto de mostrar cómo juzgar la mantenibilidad de la transformación. Un característica a destacar del enfoque propuesto es la extracción automática de métricas.

Los trabajos expuestos, en los párrafos anteriores, evalúan distintos aspectos de calidad definiendo un conjunto inicial de métricas y/o atributos. Sin embargo, estos carecen de una especificación detallada de la estrategia de medición y evaluación de atributos.

Por otro lado, la mayoría de los casos de estudios propuestos no han sido aplicados a transformaciones definidas en un entorno de desarrollo industrial. Además dichos casos de estudio se exhi-

ben subespecificados, aspecto que dificulta la interpretación y comprensión de la aplicabilidad de los enfoques propuestos.

En este trabajo, se propone la definición de un conjunto de métricas y atributos para la medición y evaluación de las transformaciones de modelos en lenguajes declarativos como QVT-R. Se empleo un framework basado en el trabajo de Olsina [16]. Para la evaluación experimental, a diferencia de otras investigaciones, se muestra un caso de estudio relevante (implementado por uno de los autores de este trabajo [1]) que fue diseñado y desarrollado para la industria.

3. Lenguajes de Transformación

Como se mencionó anteriormente, MDE impulsa la utilización de modelos como artefactos principales a ser construidos y mantenidos. Cualquier especificación puede ser expresada con modelos, en consecuencia un proceso de desarrollo de software se convierte en un proceso de refinamiento y transformación entre modelos de manera tal que el nivel de abstracción vaya decrementando hasta llegar al código para una plataforma específica. La definición de transformaciones de modelos requiere la aplicación de lenguajes específicos. Estos lenguajes deberían tener una base formal, y al menos un metamodelo que describa su sintaxis abstracta [9]. Un metamodelo describe el conjunto de modelos admisibles y define formalmente los elementos de un lenguaje de modelado junto con sus relaciones y restricciones, como se mencionó anteriormente.

Los lenguajes de transformación híbridos combinan distintas características que proveen los lenguajes declarativos y los imperativos. Los lenguajes declarativos por lo general se limitan a escenarios donde los metamodelos de entrada y de salida poseen una estructura similar y en consecuencia permiten realizar un mapeo. Además estos poseen la ventaja de ser razonados matemáticamente, lo que permite el uso de mecanismos matemáticos para optimizar el rendimiento de los programas. Si bien los lenguajes imperativos permiten manejar un mayor número de escenarios, poseen un nivel de abstracción menor. Esto deriva en un mayor esfuerzo por parte del usuario para resolver cuestiones como la trazabilidad. Los lenguajes híbridos combinan ambos enfoques para sumar sus ventajas a las transformaciones.

3.1. QVT

QVT es un lenguaje estándar para la transformación de modelos definido por la OMG[14]. Su especificación depende de los estándares OCL 2.0 y MOF 2.0. Con QVT es posible definir transformaciones genéricas entre metamodelos, de esta manera cualquier instancia de un metamodelo puede ser transformado en una instancia de otro metamodelo. La especificación QVT cuenta con una naturaleza híbrida, es decir declarativa e imperativa. QVT provee un lenguaje imperativo llamado Operational Mappings. Este lenguaje está especificado como estándar para proporcionar implementaciones imperativas. El lenguaje declarativo que provee QVT es QVT-R. Este brinda una especificación declarativa de las relaciones entre los modelos MOF y soporta reconocimiento de patrones. En este lenguaje es necesario declarar parámetros en una transformación para manejar los modelos involucrados en la misma. Estos parámetros están tipados sobre los metamodelos apropiados.

Una transformación en QVT-R especifica un grupo de relaciones que los elementos de los modelos involucrados deben satisfacer. Una transformación puede tener cualquier número de parámetros de entrada y salida llamados dominios. Los parámetros de entrada corresponden a los modelos a los cuales se les aplicará la transformación para generar los parámetros de salida. Para cada parámetro de salida, una nueva instancia de modelo será creada y/o modificada de acuerdo al metamodelo al que pertenece dicho modelo destino. Cada dominio identifica una correspondencia de un conjunto de elementos definidos llamados patrones. Una relación define reglas de transformación, y esa relación implica la existencia de clases para cada uno de sus dominios. Las relaciones pueden contener cláusulas *when* y *where*. Estas cláusulas pueden contener cualquier expresión OCL además de las expresiones de invocación de la relación. Un domi-

nio puede ser marcado como *checkonly* o *enforce*. Un patrón puede ser visto como un conjunto de variables y un conjunto de restricciones que los elementos de los modelos deben satisfacer. Un tipo de modelo está definido por un metamodelo y un conjunto opcional de expresiones de condición. El lector interesado puede consultar estos conceptos en la especificación QVT [19].

Es importante destacar que la introducción de estos conceptos (lenguajes de transformación y conceptos propios de QVT-R) son las bases, en que los autores de este trabajo se apoyaron, para definir métricas para transformaciones QVT-R.

4. Una Estrategia para Medir y Evaluar Transformaciones a Nivel Metamodelos

Como bien se induce en apartados previos, el proceso involucrado en medir los posibles cambios y mejoras en una transformación no ha sido presentado en ningún trabajo hasta el momento. Se considera que la definición de métricas y posterior evaluación permitirá mejorar el proceso de desarrollo mediante distintos tipos de tareas relacionadas al mantenimiento, reuso y comprensión.

Para especificar el conjunto de métricas se ha utilizado un enfoque similar al propuesto por Olsina et. al [3, 16]. En dichos trabajos, los autores proponen modelos y *frameworks* de calidad en conjunto con estrategias de evaluación que apuntan a un enfoque integrado para medir y evaluar diferentes áreas de las nuevas aplicaciones que se utilizan en la web.

En general, al momento de medir y evaluar cualquier tipo de artefacto (ya sea, un programa, una técnica, un proyecto, etc.) de forma sistemática, es necesario definir una estrategia que contemple un conjunto de principios, métodos, técnicas y herramientas; que permitan especificar, definir y recolectar métricas e indicadores y sus respectivos valores [16]. Además, con el objetivo de llevar a cabo el análisis y el proceso de toma de decisión, es necesario asegurar que las medidas (etapa de especificación de métricas, apartado 4.2) y los valores de los indicadores (etapa de evaluación, apartado 4.3) sean repetibles y comparables entre todos los demás en el proyecto. Por lo tanto, es mandatorio guardar no solo los datos de la medición, sino también ciertos metadatos como *procedimiento de medición*, *escala*, *tipo de escala*, *modelo de indicador elemental*, *niveles de aceptabilidad*, entre otros.

Una *métrica* es la especificación de un proceso de medición que trasforma un atributo de una entidad en una medida. Un *indicador elemental* es la especificación de un proceso de evaluación, el cual recibe como entrada los valores correspondiente a una métrica en particular y produce un valor indicador (es decir, información contextual). Sin embargo en los distintos trabajos relacionados a la temática, en general, se le ha dado poca relevancia a la especificación de este tipo de conceptos. De la misma manera, se puede observar una falta de consenso importante entre los diferentes estándares y manuscritos relativos a la temática medición y evaluación.

En las siguientes subsecciones se presentan los pasos de la estrategia para medición y evaluación propuesto por Olsina et. al. [3, 16]; tomando como principal enfoque la medición y evaluación de transformaciones especificadas en QVT-R, desde el punto de vista del mantenimiento.

4.1. Definición de Atributos

Como bien se establece en la estrategia de medición y evaluación propuesta por Olsina et. al [16], en primer lugar se debe definir un conjunto de atributos (o requerimientos) agrupados en categorías, subcategorías, etc. De esta manera se obtiene una estructura de árbol que pasa a ser el elemento principal del proceso de medición y evaluación.

En este trabajo en particular se ha definido el árbol de requerimientos exhibido en la Figura 1. El mismo ha sido desarrollado con el principal objetivo de medir y evaluar transformaciones en lenguaje QVT-R, desde el punto de vista del mantenimiento de software [4, 17]. El mismo está compuesto de 3 categorías de más alto nivel las cuales, a su vez, contienen atributos atómicos. Dichas categorías son *Tamaño*, *Complejidad* y *Reutilización*. En la categoría *Tamaño* se ha definido 3 atributos que pretenden medir la dificultad que conlleva la

comprensión y mantenimiento de la transformación bajo estudio, desde el punto de vista del tamaño que presente la misma. Con los atributos *Tamaño de las Relaciones* y *Relaciones de Gran Tamaño* se pretende relevar la dimensión de las componentes más importantes que presenta una transformación, en el caso del lenguaje QVT-R, las relaciones. Claramente, el tamaño de las transformaciones es un aspecto que puede influir a la hora de llevar a cabo las actividades de mantenimiento. Otro atributo de esta categoría es *Relaciones Pequeñas*; teniendo en cuenta lo mencionado previamente, un gran número de relaciones pequeñas también incrementa la dificultad al momento de desarrollar las actividades correspondientes al mantenimiento y/o evolución de software.

Por otra parte, también se ha incluido la categoría *Complejidad*. La misma está compuesta de atributos que pretenden medir desde ciertos aspectos sintácticos, la complejidad de una transformación. Un claro ejemplo de estos atributos son *Cláusulas when*, *Cláusulas where* y *Relaciones con OCL*; es decir, si la mayoría de las relaciones poseen dichas cláusulas y/o expresiones OCL, entonces la complejidad para llevar a cabo actividades de mantenimiento y evolución se ve claramente incrementada. Lo mismo sucede si la transformación presenta muchos anidamientos, este aspecto es medido por el atributo *Relaciones Anidadas*. Si la transformación utiliza muchos metamodelos (*Cantidad de Metamodelos*), claramente, esto amplía el espectro de conocimiento que debe poseer el ingeniero de software para poder realizar cambios en el código.

La última categoría se denomina *Reutilización*, Esta posee dos atributos que pretenden reflejar cuan reutilizable es el código que compone la transformación: *Metamodelos Estándares* y *Contexto Abarcado por cada Dominio*. El primero indica si en la transformación se ha utilizado algún metamodelo que no ha sido especificado por alguna organización de relevancia internacional (OMG, IEEE, ACM, etc.). Notoriamente, si se utilizan metamodelos no estándares en la transformación, el ingeniero de software encargado de realizar dichas tareas puede encontrar ciertas dificultades como falta de documentación, partes de dominios no especificados, ambigüedades en los metamodelos, entre otros. El atributo *Contexto Abarcado por cada Dominio* pretende establecer el contexto necesario que debe tener el ingeniero de software con respecto a los metamodelos utilizados, a la hora de realizar tareas de mantenimiento y evolución de software. Claramente, si el contexto es grande, esto hace a la transformación menos reutilizable y más difícil de mantener, ya que determinadas tareas implicadas en el mantenimiento y evolución (como la comprensión) consumen más tiempo y esfuerzo.

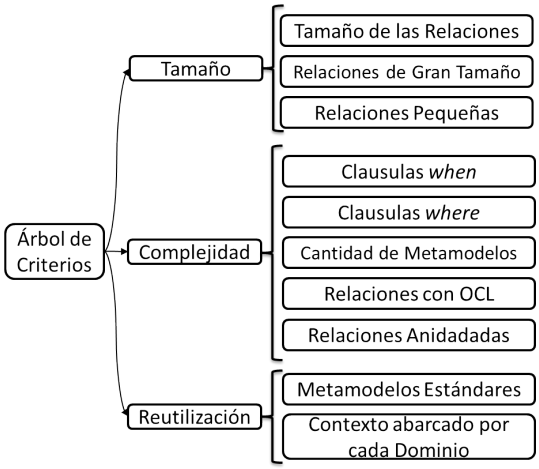


Figura 1. Árbol de Características y Atributos para evaluar transformaciones.

4.2. Especificación de Métricas

Una vez definido el árbol de atributos, es necesario establecer las métricas que permitirán cuantificar dichos atributos. Para diseñar una métrica, se debe definir el método de medición y el procedimiento de cómputo, en conjunto con la escala en donde se refleja la medición. Una *medición* produce una *medida*, es decir, el proceso de *medición* se define como “la actividad que usa una especificación de métrica con el objetivo de producir un valor, resultado de dicha medición”; mientras que, una *me-*

dida es el "numero o categoría asignada a un atributo de una entidad por medio de una *medición*" [3]. Teniendo esto en cuenta, para diseñar una métrica directa, es necesario especificar claramente un procedimiento de medición y su correspondiente escala. El tipo de la escala depende de la naturaleza de las relaciones entre los valores de la misma; es decir, algunos aspectos que se pueden ver reflejados por medio de ciertas preguntas, como: los valores en la escala, ¿mantienen el orden y/o las distancias entre categorías? ¿Existe el cero (ausencia de atributo medido)? entre otras. Entre los tipos de escalas más utilizadas en Ingeniería de Software se encuentran: *nominal, ordinal, intervalos, porcentaje, absolutas*. Esto es importante, ya que cada tipo de escala, determina el uso de operaciones matemáticas y técnicas estadísticas adecuadas para analizar los datos.

Por motivos de extensión de este artículo se han especificado las siguientes métricas: i) Tamaño de las Relaciones (Categoría: Tamaño); ii) Relaciones con OCL (Categoría: Complejidad); ii) Metamodelos Estándares (Categoría: Reutilización). Para más detalles, es posible consultar el reporte interno generado a partir de esta investigación [2]. Como se mencionó en el apartado precedente, la especificación ha sido delineada tomando en cuenta la estrategia de especificación de métricas propuesto por Olsina et. al. [3, 16].

4.2.1. Tamaño de las Relaciones

Atributo: *Tamaño de las Relaciones*.
Definición: Esta métrica proporciona información respecto del tamaño promedio de las relaciones en una transformación. El tamaño promedio es comparado con un valor que refleja un tamaño considerable en las relaciones, con respecto a la comprensión de las mismas. La métrica esta basada en que las relaciones grandes dificultan el entendimiento de las transformaciones, y por ende, las tareas de mantenimiento.

Métrica Indirecta: *Proporción del Tamaño de las Relaciones* (*PTR*).
Objetivo: Determinar el tamaño promedio en Líneas de Código (*LOCs*) de las relaciones en la transformación bajo estudio en proporción a un valor que represente una relación difícil de entender debido a sus LOCs.
Método de Cálculo:
$PTR = (\frac{\#LOCsRel}{\#TotalRel} / LOCsRelGrand) * 100$
Escala Numérica: Continua.
Tipo de Valor: Real.
Tipo de Escala: Proporción.
Métricas Relacionadas:
1) *Número Total de LOCs en las Relaciones* ($\#LOCsRel$); 2) *Número Total de Relaciones* ($\#TotalRel$); 3) *Número de LOCs en una Relación Considerada Grande* ($LOCsRelGrand$).

Atributo: *Cantidad Total de LOCs en las Relaciones*.
Métrica Directa: *Número Total de LOCs en las Relaciones* ($\#LOCsRel$).
Objetivo: sumar las LOCs de todas las relaciones.
Especificación: se cuentan las LOCs de todas las relaciones de la transformación (esto incluye comentarios, LOC con llaves que abren o cierran, etc.), a excepción de los espacios en blanco.
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

Atributo: *Cantidad Total de Relaciones*.
Métrica Directa: *Número total de Relaciones* ($\#TotalRel$).
Objetivo: sumar todas las relaciones en la transformación.
Especificación: se deben contar todas las relaciones definidas en la transformación bajo estudio.
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

Atributo: *Cantidad de LOCs en una Relación Considerada Grande*.
Métrica Directa: *Número de LOCs en una Relación Considerada Grande* ($LOCsRelGrand$).
Objetivo: especificar una cota que indique la cantidad mínima de LOCs que hacen a una relación difícil de entender y mantener debido a su tamaño.
Especificación: de acuerdo al criterio de los auto-

res y determinados expertos en la temática, se ha establecido que 40 LOCs, de acuerdo a lo definido anteriormente, dificultan la comprensión de una relación.
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

4.2.2. Relaciones con OCL

Atributo: *Relaciones con OCL*.
Definición: La métrica está orientada a medir la proporción de relaciones que usan expresiones OCL, sobre el total. Las relaciones que poseen expresiones OCL dificultan el entendimiento por parte del ingeniero de software, ya que el mismo no sólo debe comprender la semántica de las sentencias en el lenguaje de transformación, sino también la semántica de OCL. Claramente, si una transformación posee muchas relaciones con OCL, esto dificulta el proceso de comprensión y mantenimiento de la misma.

Métrica Indirecta: *Proporción de Relaciones con Expresiones OCL* (*PROCL*).
Objetivo: determinar la cantidad de relaciones en la transformación que poseen expresiones OCL en proporción a la cantidad total de relaciones.
Método de Cálculo:
$PROCL = (\frac{\#RelOCL}{\#TotalRel}) * 100$
Escala Numérica: Continua.
Tipo de Valor: Real.
Tipo de Escala: Proporción.
Métricas Relacionadas:
1) *Número Total de Relaciones con Expresiones OCL* ($\#RelOCL$); 2) *Número Total de Relaciones* ($\#TotalRel$).

Atributo: *Cantidad Total de Relaciones con Expresiones OCL*.
Métrica Directa: *Número Total de Relaciones con Expresiones OCL* ($\#RelOCL$).
Objetivo: sumar las relaciones que contengan expresiones OCL.
Especificación: se cuentan las relaciones que posean dentro de su sintaxis cualquier tipo y cantidad de expresiones en OCL.
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

Atributo: *Cantidad Total de Relaciones*.
Métrica Directa: *Número total de Relaciones* ($\#TotalRel$).
Objetivo: sumar todas las relaciones en la transformación.
Especificación: se deben contar todas las relaciones definidas en la transformación bajo estudio.
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

4.2.3. Metamodelos Estándares

Atributo: *Metamodelos Estándares*.
Definición: la métrica indica si la transformación utiliza algún metamodelo no estándar, es decir, si entre todo el conjunto que implican los metamodelos orígenes y destinos, existe al menos un metamodelo que no esté estandarizado. Esto claramente afecta las características de reutilización que pueda llegar a tener la transformación.
Métrica Directa: *Posee Metamodelos Estándares* (*MEst*).
Objetivo: detectar si la trasformación utiliza algún metamodelo estándar.
Método de Cálculo: El experto inspecciona el conjunto compuesto por los metamodelos en origen y en destino. Si se encuentra al menos un metamodelo no estándar, entonces el valor de MEst es 0, de lo contrario es 100. Los metamodelos utilizados se consideran estándares si han sido especificados por alguna organización de relevancia internacional (OMG, IEEE, ACM, etc.).
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Absoluta.

4.3. Evaluación

Claramente, todo procedimiento de medición, es seguido de uno o varios procesos de evaluación. De

manera análoga, las métricas definidas en los apartados precedentes serán útiles si se definen los indicadores adecuados en el proceso de evaluación. Un *indicador* permite especificar como calcular e interpretar los atributos especificados en el árbol en la Figura 1. En general se utilizan dos tipos de indicadores: i) indicadores elementales e ii) indicadores parciales/globales. Los primeros evalúan los requerimientos de más bajo nivel, es decir, los atributos. Cada indicador elemental posee un *modelo elemental* que proporciona una función que mapea las medidas obtenidas a partir de la métrica (el dominio) a la escala del indicador (el rango).

La nueva escala se interpreta utilizando criterios de decisión, también conocidos como *Niveles de Aceptabilidad*, los cuales ayudan a analizar el nivel de satisfacción alcanzado por cada requerimiento elemental; es decir, por cada atributo. Dichos niveles aportan tanta semántica como los mismos indicadores, ya que permiten determinar los grados de satisfacción del atributo bajo análisis. Para este caso de estudio, se han establecido 3 criterios de aceptabilidad en general: i) *Satisfactorio*; ii) *Marginal*; iii) *Insatisfactorio*; por cada uno de estos se debe especificar un intervalo en el rango [0,100] que permite determinar en que criterio de evaluación se encuentra el valor del indicador elemental. De esta manera, es posible determinar niveles de aceptabilidad por cada indicador elemental, de acuerdo al objeto de análisis y su entorno. De acuerdo a cada métrica, cada criterio tiene semántica diferentes, sin embargo, la naturaleza es la misma bajo el enfoque que ha tomado el proceso de medición y evaluación. Es relevante remarcar que en esta investigación en particular, se ha tomado un enfoque orientado al mantenimiento. Por lo tanto, si la evaluación de un atributo resulta en un valor que se encuentra en el intervalo asociado al criterio *Satisfactorio*, desde el enfoque evaluado en este trabajo, esto implica que las tareas de mantenimiento no comprenderán grandes esfuerzos tomando en cuenta dicho atributo. Por el contrario, de manera análoga, si resulta en un valor en el intervalo asociado al criterio *Insatisfactorio*, eso implica que las tareas de mantenimiento conllevarán grandes esfuerzos. En esta instancia del trabajo, los intervalos de aceptabilidad para cada indicador han sido definido considerando distintos criterios de investigadores entendidos en la temática.

Por otra parte, los *indicadores parciales/globales* evalúan requerimientos de mediano o alto nivel, es decir, características o subcaracterísticas (por ejemplo, en el árbol de la Figura 1, la característica *Complejidad*). Para esto es posible utilizar algún modelo de agregación, como por ejemplo LSP (Logic Scoring of Preference) [5, 13]. Un indicador global, representa el grado de satisfacción global del objeto de estudio con respecto a los requerimientos medidos y evaluados. En esta investigación en particular, sólo se utilizan indicadores elementales, es decir, sólo se evalúan los atributos del árbol de requerimientos, para posteriormente llevar a cabo un análisis en conjunto de los indicadores.

En los siguientes tres apartados se muestran los indicadores elementales correspondiente a las métricas definidas en la etapa previa (apartado 4.2).

4.3.1. Tamaño de las Relaciones

Atributo: *Tamaño de las Relaciones*.
Indicador Elemental:
Nombre: *Nivel de Desempeño en Tamaño de las Relaciones* (D_PTR).
Modelo Elemental:
Especificación: el mapeo es de la siguiente manera:

$$D_PTR = \begin{cases} 100 - PTR & \text{PTR} \leq 100 \\ 0 & \text{PTR} \geq 100 \end{cases}$$

Donde PTR es la métrica indirecta definida en el apartado 4.2.
Criterios de Decisión:
Criterio 1: Insatisfactorio;
Rango: $0 \leq D_PTR \leq 25$;
Criterio 2: Marginal;
Rango: $25 < D_PTR \leq 60$;
Criterio 3: Satisfactorio;
Rango: $60 < D_PTR \leq 100$;
Escala Numérica: Continua.
Tipo de Valor: Real.
Tipo de Escala: Proporción.

4.3.2. Relaciones con OCL

Atributo: *Relaciones con OCL.*
Indicador Elemental:
Nombre: *Nivel de Desempeño en Relaciones con OCL* (D_PROCL).
Modelo Elemental:
Especificación: el mapeo es de la siguiente manera:

$$D_PROCL = \begin{cases} 100 - PROCL & \text{PROCL} \leq 100 \\ 0 & \text{PROCL} \geq 100 \end{cases}$$

Donde $PROCL$ es la métrica indirecta definida en el apartado 4.2.
Criterios de Decisión:
Criterio 1: Insatisfactorio;
Rango: $0 \leq D_PROCL \leq 40$;
Criterio 2: Marginal;
Rango: $40 < D_PROCL \leq 60$;
Criterio 3: Satisfactorio;
Rango: $60 < D_PROCL \leq 100$;
Escala Numérica: Continua.
Tipo de Valor: Real.
Tipo de Escala: Proporción.

4.3.3. Metamodelos Estándares

Atributo: *Metamodelos Estándares*.
Indicador Elemental:
Nombre: *Nivel de Desempeño en Metamodelos Estándares* (D_MEst).
Modelo Elemental:
Especificación: el mapeo es directo, es decir, si $MEst$ es 100, entonces el porcentaje de satisfacción del indicador D_MEst es 100, de lo contrario es 0. Por lo tanto,
si $MEst = 100 \rightarrow D_MEst = 100$;
sino, si $MEst = 0 \rightarrow D_MEst = 0$;

Donde $MEst$ es la métrica indirecta definida en el apartado 4.2.
Criterios de Decisión:
Criterio 1: Insatisfactorio; **Valor:** 0;
Criterio 2: Satisfactorio; **Valor:** 100;
Escala Numérica: Discreta.
Tipo de Valor: Entero.
Tipo de Escala: Binaria.

4.4. Análisis de Resultados

Luego de que se han llevado a cabo las actividades que implican especificar métricas y evaluar los datos obtenidos por las mismas, es imprescindible realizar un análisis de los resultados obtenidos en las etapas previas. Es decir, mientras el proceso de medición, produce datos, el de evaluación retorna información contextual respecto de lo que se está evaluando. El resultado de este análisis se refleja en distintos tipos de reportes, documentos, sugerencias, de acuerdo al objeto de estudio. De esta manera se puede proveer un punto de partida para intervenir los puntos débiles que se detecten en el conjunto de requerimientos de calidad.

5. Caso de Estudio

Para mostrar la aplicabilidad del enfoque desarrollado en las secciones previas, en este apartado se presenta un caso de estudio en donde se evalúa una transformación definida para la empresa especializada en optimización multi-objetivo ESTECO [6]. Dicha transformación permite la conversión de muchos flujos de trabajo ingenieriles definidos en el formato propietario de ESTECO al estándar de procesos de negocio BPMN2 [1]. Las reglas de transformación, han sido validadas experimentalmente aplicándolas en la empresa mencionada anteriormente.

De acuerdo a lo establecido en la sección 4, para este estudio, se han utilizado 3 intervalos de aceptabilidad en una escala de porcentaje para los indicadores elementales definidos en el apartado 4.3. Los valores pueden caer en los intervalos: i) *Insatisfactorio*, indicado con color rojo; ii) *Marginal*, en color amarillo; o iii) *Satisfactorio* determinado por el color verde.

La Tabla 1 muestra los resultados de la evaluación para el caso de estudio. Como puede observarse en relación a la categoría *Tamaño* se han utilizado 3 métricas: i) El indicador para el atributo *Tamaño de las Relaciones* posee un valor de 0 %, ya que el promedio del tamaño de las relaciones superan el valor que se ha considerado para determinar

Característica / *Atributo*	***I.E %***
Tamaño	
Tamaño de las Relaciones	0
Relaciones Grandes	78
Relaciones Pequeñas	86
Complejidad	
Cláusulas when	61
Cláusulas where	81
Cantidad de Metamodelos	14
Relaciones con OCL	100
Relaciones Anidadas	80
Reutilización	
Metamodelos Estándares	0
Contexto Abarcado por cada Dominio	65

Tabla 1. Indicadores Elementales para la transformación analizada. Lo valores están especificados en %.

que una relación es calificada como grande. Esto dificulta el mantenimiento, ya sea correctivo, adaptativo o perfectivo, debido a que si las relaciones poseen muchas lineas de código, el esfuerzo para realizar este tipo de tareas se ve incrementado.

ii) Si observamos el resultado con respecto al atributo *Relaciones Grandes*, se puede observar que la transformación no posee un gran número de relaciones extensas.

iii) De forma similar al punto ii), lo mismo sucede con las relaciones, pequeñas. Si la transformación posee muchas relaciones pequeñas aumentan las posibilidades que existan muchas invocaciones a otras relaciones. En el caso de estudio se puede observar que existe un número mínimo de relaciones pequeñas.

Al momento de evaluar la *Complejidad* de la transformación, se puede analizar desde el punto de vista de los 5 atributos: i) La cláusula *when* es aquella que establece las condiciones que se necesitan para satisfacer la relación. Mientras más relaciones posean este tipo de cláusulas, en un contexto de tareas de mantenimiento, la complejidad de las mismas aumenta, ya que se necesita poseer un conocimiento más amplio sobre el dominio. La transformación analizada en el caso de estudio presenta un nivel *Marginal*.

ii) La cláusula *where* define la condición que deben cumplir todos los elementos que intervienen en la relación, y tiene el poder de restringir cualquiera de las variables en la relación y sus dominios. En esta cláusula se pueden encontrar la mayoría de las invocaciones a otras relaciones, es por ello que mientras más relaciones posean esta cláusula más difícil será comprender la transformación. El caso de estudio presenta un nivel *Satisfactorio* respecto de este tipo de cláusulas.

iii) Es evidente que mientras más metamodelos se utilicen en la transformación, más compleja será, ya que se deben conocer todos los metamodelos que participan en la misma. La cantidad de metamodelos dificulta la comprensión y el mantenimiento. En el caso de estudio este indicador elemental resulta en 14 % (*Insatisfactorio*) porque la transformación utiliza 6 metamodelos.

iv) Teniendo en cuenta que la especificación de QVT cuenta con el estándar OCL, existe un gran porcentaje de que las transformaciones posean expresiones OCL. Pero si bien OCL ayuda a la hora de construir una transformación, hay que tener en cuenta que mientras más relaciones posean este tipo de expresiones la complejidad aumentará. En el caso de estudio no se han encontrado expresiones OCL, por tal motivo el indicador elemental es de 100 %

v) De manera análoga a los lenguaje de programación de propósito general, mientras más anidamientos presente la transformación, más difícil es comprender la misma. La transformación analizada posee relativamente poco anidamiento, lo que facilitará su mantenimiento y su comprensión.

Teniendo en cuenta la *Reutilización* se han analizado dos aspectos: i) Es claro que si todos los metamodelos son estándares (especificados por alguna organización reconocida internacionalmente) se facilitará la reutilización de la transformación completa o partes de la misma. Por el contrario, si la transformación utiliza al menos un metamodelo que no es estándar, el ingeniero de software puede verse en diferentes tipos de dificultades, como por ejemplo falta de documentación, asistencia por parte de expertos, entre otras. El caso de estudio posee 2 metamodelos no estándares, es por ello que el nivel de aceptabilidad del indicador correspondiente es *Insatisfactorio*.

ii) Para determinar el contexto abarcado por cada dominio se han considerado todas las clases que participan en una transformación y que pertenecen al dominio origen y dominio destino. Si en la transformación se utilizan todas las clases de los metamodelos que participan en la transformación, más difícil será poder reutilizar la misma. En el caso de estudio analizado el indicador elemental ha resultado en *Marginal* ya que se utiliza un porcentaje medio/bajo (35 %) de clases sobre el total contenido en los dominios.

6. Conclusiones y Trabajos Futuros

Sin lugar a dudas, una de las tareas más complejas y que más tiempo consume en el ciclo de vida de una aplicación es la de mantenimiento [4, 18]. La tendencia al uso de los lenguajes de transformación a nivel metamodelo, indica que este tipo de "programas" no es la excepción en comparación con los lenguajes de propósito general. Las tareas comprendidas dentro del contexto de mantenimiento y evolución de software pueden llegar a ser diversas de acuerdo al objetivo de dicha actividad. En este sentido, es posible que el mantenimiento conlleve corrección de errores, re-estructuración, mejora en desempeño, portabilidad o cualquier otro atributo de calidad, entre otros.

En este artículo se definió un conjunto de métricas que posibilitan la medición y evaluación de transformaciones a nivel metamodelos con el objetivo de asistir al ingeniero de software a analizar y estimar esfuerzos para llevar a cabo tareas de mantenimiento y evolución en las mismas.

En primer lugar se definió un conjunto de atributos agrupados en tres categorías diferentes: *Tamaño*, *Complejidad* y *Reutilización*. Luego, por cada atributo: i) se especificó una métrica que permite obtener un valor (medida) del mismo; ii) se definió un indicador elemental que permite obtener información contextual respecto de la medida, es decir, permite evaluar el resultado de medir dicho atributo.

Para probar del enfoque propuesto en este trabajo se empleó como caso de estudio una transformación desarrollada para una empresa especializada en optimización multi-objetivo (ESTECO [6]). Dicha transformación permite la conversión de muchos flujos de trabajo ingenieriles definidos en el formato propietario de ESTECO al estándar de procesos de negocio BPMN2. De este modo, las reglas de transformación se validaron experimentalmente aplicándolas a un caso de estudio real.

Como resultado del análisis del caso de estudio se puede concluir los siguientes aspectos:

- Desde el punto de vista del *Tamaño*, las relaciones poseen, en promedio, más líneas de código que las estimadas para una relación grande, sin embargo la transformación posee pocas relaciones "pequeñas" y "extensas".
- Desde un enfoque más general, la *Complejidad* de la transformación no muestra muchos atributos como *Insatisfactorios*, sin embargo, se puede notar que hace uso de muchos metamodelos, aspecto que desfavorece el proceso de comprensión.
- Con respecto a la *Reutilización*, es posible destacar lo siguiente: i) la transformación posee un metamodelo no estándar y esto perjudica su reutilización en otro contexto por parte de los ingenieros de software; ii) el contexto del dominio, teniendo en cuenta todos los metamodelos, que debe tener el ingeniero de software para comprender la transformación es *Marginal*, es decir, no está dentro de lo óptimo pero tampoco se considera voluminoso.

Como trabajos futuros se proponen las siguientes aristas principales:

- Someter los "componentes" de la estrategia de evaluación a un proceso de mejora continua; es decir, mejorar de manera continua el árbol de atributos, las métricas y los indicadores elementales propuestos, mediante interacción con expertos en la temática.
- Expandir el árbol de atributos abarcando otras características que reflejen aspectos del mantenimiento no contemplados en esta instancia de la investigación.

- Incorporar a la etapa de evaluación, un método de agregación multicriterio para obtener valores a nivel de características [5, 13].
- Proponer estrategias de extracción automática de métricas como sugiere Kapova et. al. [11] en conjunto con la integración en un entorno de desarrollo como Eclipse o Netbeans.

Referencias

[1] C. Abdelahad. Transformación de Workflows Científicos a Modelos de Negocio en BPMN2 utilizando QVT. Master's thesis, UNSL, 2013.

[2] C. Abdelahad, E. Miranda, and N. Perez. Medición y Evaluación de una Transformación en QVT-R para ESTECO . Technical report, 2014.

[3] P. Becker, P. Lew, and L. Olsina. Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy. *Advances in Software Engineering*, 2012:2, 2012.

[4] K. Bennett and V. Rajlich. Software Maintenance and Evolution: a Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 73–87, New York, NY, USA, 2000. ACM.

[5] J. Dujmović and M. Kadaster. A technique and tool for software evaluation. *Evolution*, 374:246, 2002.

[6] ESTECO. `http://www.esteco.com`, 2014.

[7] J.-M. Favre, J. Estublier, and M. Blay. Beyond MDA: Model Driven Engineering (L'Ingénierie Dirigée par les Modeles: au-déla du MDA) Edition Hezmes-Lavoisier. Technical report, ISBN 2-7462-1213-7. Février, 2006.

[8] M. Fowler. *Refactoring: Improving the Design of Existing Code*. Pearson Education India, 1999.

[9] Giandini, Roxana S and Pons, Claudia and Pérez, Gabriela. A two-level formal semantics for the qvt language. *CIbSE*, 9:73–86, 2009.

[10] F. Jouault and I. Kurtev. Transforming models with ATL. In *Satellite Events at the MoDELS 2005 Conference*, pages 128–138. Springer, 2006.

[11] L. Kapová, T. Goldschmidt, S. Becker, and J. Henss. Evaluating Maintainability with Code Metrics for Model-to-Model Transformations. In *Research into Practice–Reality and Gaps*, pages 151–166. Springer, 2010.

[12] A. G. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[13] E. Miranda, M. Beron, G. Montejano, M. J. Varanda Pereira, and P. Henriques. NESSy: a New Evaluator for Software Development Tools. In J. P. Leal, R. Rocha, and A. Simões, editors, *2nd Symposium on Languages, Applications and Technologies*, volume 29 of *OpenAccess Series in Informatics (OASIcs)*, pages 21–37, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[14] O. M. G. O. Modeling and metadata specifications. `http://www.omg.org/spec`, 2014.

[15] P. Mohagheghi and V. Dehlen. Developing a quality framework for model-driven engineering. In *Models in Software Engineering*, pages 275–286. Springer, 2008.

[16] L. Olsina, P. Lew, A. Dieser, and B. Rivera. Updating Quality Models for Evaluating New Generation Web Applications. *Journal of Web Engineering*, 11(3):209–246, 2012.

[17] M. Petrenko, V. Rajlich, and V. R. Partial Domain Comprehension in Software Evolution and Maintenance. *International Conference on Program Comprehension (ICPC08)*, pages 13–21, 2008.

[18] R. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, Inc., New York, NY, USA, 7 edition, 2009.

[19] QVT. `http://www.omg.org/spec/QVT/1.1`, 2014.

[20] L. Reynoso, M. Genero, and M. Piattini. Measuring ocl expressions: An approach based on cognitive techniques. *Metrics for Software Conceptual Models*, 2005.

[21] M. Van Amstel, S. Bosems, I. Kurtev, and L. F. Pires. Performance in model transformations: experiments with ATL and QVT. In *Theory and Practice of Model Transformations*, pages 198–212. Springer, 2011.

[22] M. F. van Amstel, C. F. Lange, and M. G. van den Brand. Using metrics for assessing the quality of ASF+ SDF model transformations. In *Theory and Practice of Model Transformations*, pages 239–248. Springer, 2009.

[23] A. Van Deursen, J. Heering, and P. Klint. *Language Prototyping: An Algebraic Specification Approach: Vol. V*. World Scientific, 1996.

[24] M. Völter, T. Stahl, J. Bettin, A. Haase, S. Helsen, and K. Czarnecki. Model-Driven Software Development: Technology. *Engineering, Management. Wiley*, 5:6, 2006.

Caracterización de los Riesgos inherentes a la Ingeniería Reversa

Cuevas, Juan Carlos; Gastañaga, Iris; Gimenez Zens, Inés; Mana Franco
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract
La identificación de los riesgos en el proceso de Ingeniería Reversa constituye un actividad primordial a los efectos de desarrollar e implementar estrategias serias para su administración. A tal fin es necesario que, además de identificar estos riesgos, se establezca la importancia relativa de ellos conjuntamente con las causa por las que ciertos riesgos son percibidos como más importantes que otros. Estas acciones son útiles a los fines de orientar las decisiones de los responsables del proyecto en aquellos factores que presentan mayor nivel de riesgo.
En este trabajo se muestran los resultados obtenidos en un estudio empírico realizado con la participación de profesionales de la industria del Software de la ciudad de Córdoba, República Argentina, con experiencia en las actividades propias de la Ingeniería Reversa (IR) destinado a identificar los riesgos inherentes a dicha actividad. Estos riesgos se identificaron y categorizaron de acuerdo a los aportes realizados por los profesionales entrevistados.

Palabras Clave
Riesgos, Identificación de riesgos, Ingeniería Reversa.

1.- Introducción

1.1.- Formulación del problema

El mantenimiento de Software consiste en la modificación de un producto software después de su entrega para corregir fallas, mejorar su rendimiento u otros atributos o para adaptar el producto a un entorno cambiante [IEEE Std 610]. La administración de proyectos de mantenimiento de software es dificultosa debido a la necesidad de comprender el sistema de software antes de realizar cualquier tipo de cambio. Entre las principales razones que fundamentan lo expuesto se incluye: documentación incorrecta, desactualizada o inexistente, complejidad del sistema y falta de conocimiento suficiente sobre el dominio del sistema de software por parte del encargado de su mantenimiento. Una forma de abordar este tipo de problemas es obtener información relevante a partir de los distintos artefactos que componen un sistema de software, como por ejemplo el código fuente. Este proceso es lo que se conoce como IR, el cual implica el análisis del sistema objetivo, la identificación de sus componentes y sus interrelaciones y la creación de representaciones del sistema en diferentes formas o en elevados niveles de abstracción [Chikofsky:1990]. Para ello es fundamental la aplicación de un proceso sistemático para facilitar la tarea de comprensión del sistema software y obtener documentación del mismo que contribuya a mejorar su mantenimiento.

Por otro lado, como en todo proceso, existe la probabilidad de errores, fallas, omisiones, etc. que impactan en diferente medida sobre cada uno de los componentes del proceso, así como sobre el proceso considerado como un todo. Esta situación nos pone ante la necesidad de identificar y gestionar los riesgos. [ISO-IEC 31000:2009] proporciona principios y directrices para la gestión de riesgos y el proceso implementado en el nivel estratégico y operativo. Entre sus principales principios podemos mencionar que la gestión de riesgos: crea valor: integra procesos organizacionales; aborda explícitamente la incertidumbre; debe ser sistemática, estructurada y oportuna; integra factores humanos y culturales; y, facilita la mejora continua de la organización, entre otros. Por su parte [ISO-IEC 31010:2009] aborda las técnicas de evaluación de riesgos. Por lo expuesto la gestión de riesgos está muy ligada a la prevención, elemento fundamental en la gestión de la calidad.

El abordaje de los riesgos vinculados a la IR no es uno de los temas de mayor tratamiento en la bibliografía existente, pero

no por ello deja de ser un tema de interés y por ende es propicio y necesario avanzar en su exploración, sobre todo en ambientes de proyectos de software que son administrados sin la aplicación de métodos ingenieriles y sin suficiente basamento en la filosofía de Calidad.

El presente documento se estructura a partir del análisis de la bibliografía existente, la propuesta de investigación, los resultados y la propuesta de caracterización de los riesgos de IR que se desprenden del análisis realizado.

1.2.- Marco Teórico

Existen diferentes abordajes sobre la **IR:** Un enfoque programable para IR [Tilley et al:1994]; la discusión de la relación del análisis del dominio de la aplicación y la IR [DeBaud:1994] ; una base sobre la cual los usuarios construyen ambientes de IR específicos del dominio [Tilley:1995]. Se discuten tres cuestiones de calidad con respecto a la IR de bases de datos existentes [Chiang-Barron:1995]; se propone un enfoque definido y sistemático para comparar procesos o componentes de procesos [Tortorella-Visaggio:1997]; Se caracteriza las dimensiones de variación para IR de bases de datos [Blaha:1997], se describe IR basado en patrones de Componentes de Diseño [Keller et al:1999]; Se describen circuntancias que impiden la aplicación de técnicas automatizadas y motiva a la IR manual [Swafford:2000]; la precisión y completitud de una IR se trata en [Rugaber et al:2001]; se plantea que las vistas polimétricas contribuyen a comprender los problemas de defectos y estructura de un producto software en las fases iniciales de un proceso de IR [Lanza-Ducasse:2003];

Más allá de ello, no se encuentra abordado el tema de los riesgos en la IR.

La IR en proyectos de software de aerotransporte es un tema frecuente y se aborda en [Rierson-Lingberg:2004]; la IR en el contexto del matenimiento de software es abordada también por [Grubb-Takang:2005] a la vez que se realizan algunos estudios empíricos en IR [Tonella et al:2007] y la revisión del campo de la IR, sus principales logros y áreas de aplicación [Canfora:2007]; Dos técnicas de IR, visualización de software y ubicación característica, pueden ayudar a los sistemas mainframe a adaptarse a los cambios [Van Geet-Demeyer:2010]; entre otros.

De los antecedentes mencionados podemos concluir que: 1) la gestión de riesgos no es tenida en cuenta como un elemento activo del proceso de IR; y 2) entre aquellos antecedentes en los cuales se hace mención al riesgo podemos citar a [Grubb-Takang:2005] quienes, si bien abordan el tema de riesgo en diversos capítulos de trabajo, en el capítulo referido a IR no realizan referencia alguna a la temática. En lo que respecta a la publicación de [Eilam:2005] el abordaje de riesgos está relacionado a los aspectos de seguridad del acceso al producto software, pero no a los riesgos propios del proceso de IR, y en el caso del aporte de [Tonella-Potrich:2005] la única referencia a riesgo no está abordada e nivel de gestión sino de advertencia. Resumiendo, es escasa la mención de riesgos y en aquellos casos en los que se los referencia, su abordaje es de índole genérica y no se profundiza demasiado en los riesgos relacionados a la información de entrada, sus métodos, el personal, la medición, el medio ambiente (del contexto del producto software, como así también de la tecnología propia del producto, el dominio del mismo, etc.) y las herramientas propias de este proceso, limitación ésta que impacta directamente en la gestión de la calidad del proceso.

La única excepción encontrada a lo largo del análisis del material obtenido es la propuesta de [Demeyer et al:2000] que postulan que la IR debe ser considerada como una faceta escencial en el paradigma de orientación a objetos y que como todo proyecto requiere, debido a sus propias incertidumbres, de una evaluación de riesgos. En este material, los autores incluyen factores de riesgo para cada grupo de patrones, así como también postulan

cómo cada patrón contribuye a reducir dichos riesgos.

En lo que respecta a **herramientas y técnicas** relacionadas a los procesos de la IR, también existe una importante cantidad de publicaciones, como por ejemplo: August II: Una herramienta para el paso a paso de la IR del Modelo de Datos [Davis:1995], Uso de ténicas formales e informales en IR para programas en C [Gannod-Cheng:1996], Comparación de cuatro herramientas de IR [Bellay-Gall: 1997], Lecciones aprendidas en IR de Datos [Davis:2001], Herramientas para IR como medio para Conocimiento imperfecto [Jahnke-Valenstein:2000], atributos de la calidad –aplicabilidad, extensibilidad y escalabilidad- [Tilley:1998], Métricas para Ingeniería Reversa [Zhou et al:1999], cuestiones de Calidad para IR en Base de Datos: el método, adquisición de la semántica del dominio en forma minuciosa, y criterio de evaluación de performance [Chiang-Barron:1995], Hacia un abordaje efectivo para IR [Garcia et al:2004], Ingenería reversa con Acoplamiento Lógico [D´Ambros-Lanza:2006], e Ingeniería de Reversa para Requerimientos destacando aspectos conducentes a reducir la complejidad y a mejorar la calidad del software [Fahmi-Choi:2007], entre otros, con igual carencia de un adecuado abordaje de los riesgos de la IR.

Asímismo, en lo referente a la administración de riesgos es conveniente realizar algunas reflexiones tales como que la IR siendo parte del mantenimiento del software o la re-ingeniería involucra riesgos, como en toda actividad de una organización. ISO define al riesgo como "la combinación de la probabilidad de un evento y su consecuencia" [ISO-IEC 73:2009] y el PMBok[1] define al riesgo de la siguiente forma: "Un riesgo es un evento o condición incierta que, si sucede, tiene un efecto en por lo menos uno de los objetivos del proyecto" [PMBok:2013]. La gestión de riesgos ayuda a la toma de decisiones al tener en cuenta la incertidumbre y su efecto sobre el logro de objetivos como así también ayuda a evaluar la necesidad de acciones. PMI [PMBok:2013] postula que los objetivos de la Gestión de los Riesgos de un Proyecto son aumentar la probabilidad y el impacto de eventos positivos, y disminuir la probabilidad y el impacto de eventos negativos para el proyecto.

Ya que el resultado de la IR es único, deberíamos considerar la IR un proyecto en sí mismo y gestionarlo como tal. PMI [PMBok:2013] define a un proyecto como "un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único". Por ello algunos riesgos típicos de los proyectos pueden ser "entradas" para la identificación de riesgos de la IR. Los riesgos de los proyectos pueden ser técnicos (requerimientos, tecnología, complejidad e interfaces, desempeño y confiabilidad y calidad), de gestión (estimaciones, planificación, control y comunicación), externos (subcontratistas y proveedores, regulaciones, mercado y clientes) y organizacionales (áreas del proyecto, recursos, fondos y priorización). El objeto de estudio de esta investigación focalizó fundamentalmente en los riesgos referidos a los aspectos técnicos y de la gestión de proyectos.

Continuando con los riesgos vinculados al software, se puede observar un abordaje fundamentalmente orientado a la producción del mismo, como por ejemplo que la identificación de riesgos en etapas tempranas del desarrollo disminuye los costos a largo plazo y ayuda a prevenir desastres de software [Boehm:1991], también se conoce metodologías generales del abordaje de riesgos tales como el que

[1] Desarrollada por el Project Management Institute (PMI), la Guía del PMBOK® es el conjunto de conocimientos en Dirección/Gestión/Administración de Proyectos generalmente reconocidos como «buenas prácticas», y que se constituye como estándar de Administración de proyectos. La Guía PMBOK® comprende dos grandes secciones, la primera sobre los procesos y contextos de un proyecto, la segunda sobre las áreas de conocimientos específicos para la gestión de un proyecto.

propone un proceso de 7 pasos que describe cómo identificar factores de riesgos, calcular su probabilidad y efecto sobre un proyecto, planear y conducir la administración de riesgos [Fairley:1994], y se plantea que si bien en las disciplinas de ingenierías maduras la administración de riesgos ha sido de rigor por centurias, se preguntan cómo se aplica a los últimos negocios riesgosos: el desarrollo de software [Boehm-DeMarco:1997].

También se analizaron otros abordajes tales como los de: Lister [Lister:1997], Carr [Carr:1997], Carvey, Phair y Wilson [Carvey et al:1997], Moyniham [Moyniham:1997] y otros; y el trabajo que aborda el tema de la Administración de Riesgos en el Mantenimiento de Software [Charette et al:1997] como uno de los pocos trabajos que vinculan el tema de riesgos en la etapa del mantenimiento dentro del ciclo de vida del software.

Lo anteriormente puntualizado generó la necesidad de profundizar el conocimiento del estado del arte en lo referente a IR en general y las prácticas de Gestión de sus Riesgos asociados como contribución a la Calidad en dicho ámbito.

Este proyecto forma parte de un programa de investigación orientado al desarrollo de un framework para la Gestión de la Calidad en IR dentro del contexto de la Cátedra de Servicios del Software del Departamento de Sistemas de Información de la UTN – FRC, con aportes desde la cátedra Proyecto Final y tiene por objetivo la identificación de los riesgos inherentes a la IR contribuyendo así al avance del conocimiento en la disciplina, como así también a las mencionadas asignaturas y otras con las que interactúa en la complejidad del currículo de la carrera, a los docentes de la carrera, investigadores, graduados y profesionales y a la Industria del Software

2.- Elementos del Trabajo y metodología

El objetivo de la investigación estuvo centrado en identificar los riesgos inherentes a la Ingeniería Reversa como herramienta que contribuya a la productividad, eficiencia y calidad en su gestión. Para lo cual se realizó el *estudio exploratorio* de carácter bibliográfico sobre Mantenimiento de Software en general e IR en particular, agregando a dicho estudio, un importente análisis de temas adicionales como Gestión de Riesgos, Herramientas para la IR y Métodos para la IR.

En lo que respecta a la *identificación del universo poblacional* se consideró como tal al conjunto de empresas de la industria del software de la Ciudad de Córdoba que tuvieran experiencia en IR. A tal fin se tomó como base a las empresas integrantes del Córdoba Clúster Technology integrado por 143 empresas, de las cuales se contactó a 72 de ellas y 11 que no integran dicho Clúster, de la misma ciudad, de las cuales solo 56 de ellas contaban con sistemas heredados y estaban dispuestas a colaborar en el trabajo de campo para este proyecto de investigación.

Se desarrolló un *plan de recopilación de la información* que resultó en el desarrollo del procedimiento utilizado.

Se *desarrolló un cuestionario* como herramienta para recopilar la información. El mismo consta de 8 preguntas generales sobre la empresa en la que se desempeña o desempeñó el profesional al momento de llevar a cabo las actividades de IR, otro grupo de preguntas con respecto a los resultados obtenidos de dichas actividades y un conjunto de 11 preguntas sobre los problemas identificados por el profesional entrevistado durante el desarrollo de su/s experiencia/s en relación a las actividades de IR. Al referenciar este grupo de preguntas es necesario realizar dos observaciones:

a.- Estas preguntas están relacionadas a los principales componentes que introducen variabilidad en un proceso, postuladas, entre otros, por [Shewhart:2003] y Kaoru Ishikawa. Se adoptó este enfoque para abordar el tema de los riesgos al carecer de un marco de trabajo estandarizado y reconocido a nivel de la industria y del campo científico, y se identificaron criterios

aglutinantes de riesgo vinculados a cada uno de estos componentes, a saber:

a) Materia prima.
    a. Información de Entrada.
    b. Conocimiento de otros dominios.
b) Mano de Obra.
    a. Aptitudes del personal.
    b. Actitudes del personal.
c) Maquinaria.
    a. Problemas asociados a la Tecnología.
    b. Problemas relativos a las Herramientas.
d) Medición.
    a. Métricas.
e) Medio Ambiente.
    a. Relaciones con el cliente.
    b. Dominio de la aplicación.
    c. Ambiente del producto software.
f) Método.
    a. Método de IR.
    b. Métodos de gestión de Proyectos.

Los resultados esperados se circunscriben a: (1) la *identificación* de los riesgos vinculados al proceso de Ingeniería Reversa (2) su *priorización* en función de una relación entre la severidad de sus consecuencias y su reiteración, y, (3) *aportar* a una gestión más eficiente del software en aspectos tales como funcionalidad, confiabilidad, usabilidad, mantenibilidad, portabilidad, seguridad, costos y satisfacción de los clientes-usuarios, aspectos todos ellos elicitados a partir de las experiencias y percepciones de los profesionales entrevistados.

La población a estudiar se constituyó por profesionales de Sistemas de Información o Informáticos *con experiencia en IR*, de la empresas del sector industrial del Software de la Ciudad de Córdoba, siendo, por ende, la unidad estadística el profesional con experiencia en IR y se recurrió a un muestreo no probabilístico, en particular de la Muestra Disponible, también conocido como Muestreo a la Mano o Muestreo por Conveniencia, quedando la muestra constituida por una parte de la población que se encontró convenientemente disponible.

A los profesionales entrevistados se les solicitó que identificaran los riesgos de acuerdo a la categorización antes descripta, así como también su percepción en base a su experiencia (ya que no contaban con datos específicos al momento de la entrevista) en lo relativo a vulnerabilidad y frecuencia, en ambos casos se valorizó de acuerdo a una escala cualitativa: **M**uy **B**aja-**B**aja-**M**edia-**A**lta-**M**uy **A**lta.

b.- Se partió de la premisa de que un problema, antes de constituirse como tal, era un modo potencial de fallo (existía la posibilidad de que se presentara y tendría un cierto impacto), es decir, era un riesgo. Esta premisa se adoptó debido a que el grupo de investigación estimó que al entrevistado le resultaría más fácil identificar problemas reales que realizar lo propio con los riesgos debido a que la población objetivo no tiene internalizada una cultura de análisis de riesgos en sus prácticas profesionales habituales.

Finalidad de las IR analizadas

En el Gráfico 1 se presentan los resultados obtenidos relativos a la finalidad de los procesos de ingeniería reversa realizados en la empresa en que desarrollan o desarrollaron sus actividades los profesionales entrevistados.

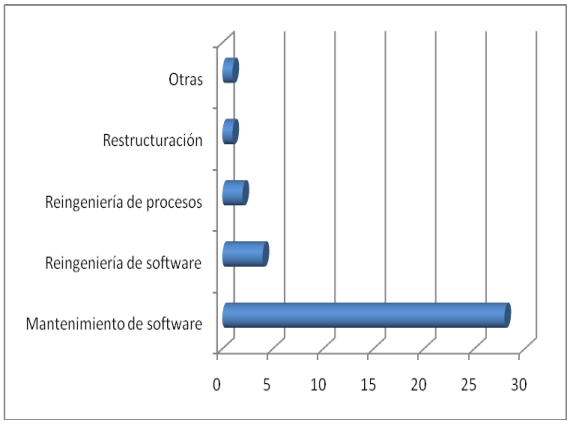


Gráfico 1: Detalle de de la finalidad de las IR realizadas por los profesionales entrevistados, que realizan o han realizado actividades propias de esta disciplina.

3- Riesgos Identificados

El resultado obtenido fue la identificación de 153 riesgos que fueron agrupados de acuerdo a la categorización antes descripta. En este punto es menester destacar que dichos riesgos se identificaron en base a la experiencia, experimentación, y en último término en la percepción de los profesionales entrevistados. El detalle de cantidad de riesgos identificados por categoría se presentan en el Gráfico 2. Esta categorización se utilizó a los fines de establecer, de alguna manera, los aspectos relevantes que hacen al proceso de IR, que introducen variabilidad en su desempeño y por ende impactarán en la salida de dicho proceso y que se consideran necesarios a los fines de poder orientar en forma organizada la identificación de los riesgos.

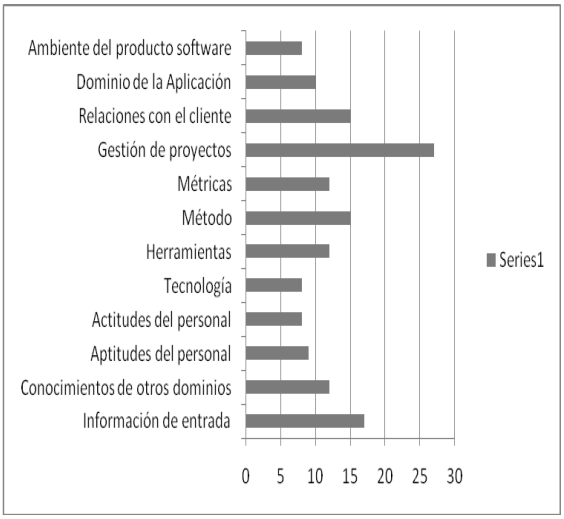


Gráfico 2: Detalle de cantidad de riesgos identificados por categoría.

A continuación, en Tabla 1, se identifican por categoría de riesgos, aquellos con mayor Nivel de Peligrosidad. Este nivel de peligro resulta de la relación entre su vulnerabilidad (estimación de la severidad de las consecuencias) y su frecuencia relativa (referida al número de veces en que se presenta un determinado riesgo en relación a las veces en que el profesional realizó las actividades de IR a los fines de estimar la probabilidad de ocurrencia del incidente o evento).

Información de Entrada
Inexistencia de documentación
Imposibilidad de acceso al código
Falta de personas dispuestas a brindar información detallada
Conocimiento de otros dominios
Falta de experiencia en IR
Desconocimiento de los pasos a seguir
Aptitudes del personal
Falta de Capacitación del IR
Falta de organización
Desconocimiento de herramientas y técnicas
Actitudes del personal
Restar importancia a la documentación
Problemas de comunicación entre los integrantes del proyecto
Tecnología
Complejidad en las tecnologías usadas
Desconocimiento de los lenguajes de programación
Herramientas
Inexperiencia en el uso de herramientas
No disponer de las herramientas
Métricas
Mediciones vinculadas solo a la performance del personal
No obtener métricas
Relaciones con el cliente
Falta de comprensión acabada de los requerimientos
Incumplimiento de fechas
Dominio de la aplicación
Desconocimiento del dominio
desconocimiento en el uso de la aplicación a ser sometida a la IR
Ambiente del producto software
Desconocimiento del entorno del producto software objeto de la IR
Nivel de obsolescencia del producto
Complejidad al evaluar la relación entre los módulos
Dificultad en construir ambientes de desarrollo y testing
Producto inmaduro al momento del despliegue en el cliente
Método
Desconocimiento del método
Falta de definición formal del método
Gestión del Proyecto
Estimaciones incorrectas
Falta de experiencia en la gestión de proyectos
Falta de gestión de riesgos

Tabla 1: Ítems de riesgo identificados por categoría y con mayor nivel de peligrosidad dentro de dichas categorías.

En el análisis de estos riesgos identificados se constató que diferentes profesionales refenciaban los mismos riesgos por lo que se procedió a considerar los que se presentaron con mayor frecuencia y se describen en la Tabla 2.

Categoría de Riesgo	**Principal/es Riesgo/s Identificado/s por Categoría**
Información de entrada	***Inexistencia o falta de documentación***
	Imposibilidad de acceder al código fuente
Conocimientos de otros dominio	***Falta de experiencia en IR***
Aptitudes del personal	***Falta de capacitación***
Actitudes del personal	***Restar importancia a la documentación (*)***
Problemas asociados a la tecnología	***Cambios en la aplicación durante del desarrollo de la IR***
Herramientas	***Falta de Herramientas***
	Desconocimiento de las herramientas
Método	***Desconocimiento del método***
Métricas	***No tomar métricas formales***
Gestión de proyectos	***Falta de seguimiento del proyecto***
Relaciones con el cliente	***Errónea gestión de requerimientos***
Dominio de la aplicación	***Falta de conocimiento del dominio***
Ambiente del producto software	***Falta de conocimiento del entorno/ambiente tecnológico del producto software***

Tabla 2: Principales riesgos, por Categoría, con más reiteración en su enunciación por los entrevistados

4.- Resultados

Para finalizar el análisis y considerando lo postulado por Bohem [Boehm:1991] sobre la necesidad de identificar los principales riesgos dada la complejidad que implicaría la gestión de todos los riesgos identificados, es que en la Tabla 3 se caracterizan los riesgos con más alto Nivel de Peligrosidad de acuerdo a la información provista por los profesionales entrevistados.

Riesgo
Falta de Capacitación del IR
Falta de organización
Desconocimiento de herramientas y técnicas
Inexperiencia en el uso de herramientas
No disponer de las herramientas apropiadas
Mediciones vinculadas solo a la performance del personal
No obtener métricas
Falta de comprensión acabada de los requerimientos
Incumplimiento de fechas
Desconocimiento del dominio
Desconocimiento en el uso de la aplicación a ser sometida a la IR
Desconocimiento del método
Falta de definición formal del método
Estimaciones incorrectas
Falta de experiencia en la gestión de proyectos
Falta de gestión de riesgos

Tabla 3: Listado de riesgos con más alto Nivel de Peligrosidad hallados.

Discusión

En primer lugar, el abordaje de la administración de riesgos vinculados a la IR no es uno de los temas de mayor tratamiento en materia de investigación y desarrollo, pero no por ello debe dejar de ser considerado como conveniente y necesario avanzar en su exploración, sobre todo en ambientes de proyectos de software administrados sin la aplicación de métodos ingenieriles y/o que no se basan en los fundamentos filosóficos de la Calidad.

En un escenario ideal de Ingeniería de Software, los controles y aseguramiento de calidad sobre el producto y los procesos deberían arrojar un resultado tal que la IR no fuese necesaria. Sabemos que no es así y que aún cuando pudiésemos lograrlo hay una cantidad de sistemas de software implementados en negocios y ámbitos cuya criticidad (aplicaciones de bancos o de aerolíneas por ejemplo) hace imposible su reemplazo. Para estos casos la IR será siempre una disciplina necesaria.

En segundo lugar, podemos observar, en función de los resultados obtenidos en el presente estudio, que la principal finalidad de las experiencias realizadas en IR tiene como objetivo el mantenimiento de software, lo que nos abre el interrogante de hasta qué punto la futura mantenibilidad de los desarrollos de productos software es uno de los requerimientos principales de los

nuevos desarrollos, sobre todo en lo referente a aquellos basados en las metodología ágiles.
En tercer lugar, los resultados obtenidos en general de las entrevistas con los profesionales nos muestran que los 36 entrevistados reconocieron 153 riesgos, lo que implica un promedio de 4.25 riesgos identificados por profesional. Esto nos lleva a reflexionar sobre si la cultura de identificación, análisis y gestión de riesgos de los profesionales de nuestra industria es la adecuada y por ende establecer si no estamos ante una gran oportunidad de mejorar aquellas prácticas, como en este caso las vinculadas a la prevención, fuertemente orientadas a la Calidad de nuestros productos software.
En cuarto lugar, tomando en cuenta los resultados del trabajo de [Arnuphaptrairong:2011] y comparandolos con los de este proyecto -***Tabla 4***- se puede establecer que existen riesgos comunes, a la vez que se pueden identificar algunos riesgos específicos de la Ingeniería Reversa como lo son: *Desconocimiento del dominio (del producto software objeto de la IR); Desconocimiento en el uso de la aplicación a ser sometida a la IR y Desconocimiento del método a utilizar para realizar una Ingeniería reversa.*

Riegos en Ingeniería Reversa	**Riesgos en Ingeniería de Software**
Falta de Capacitación del IR	Incomprensión de los requerimientos
Desconocimiento del dominio	Falta de compromiso y soporte de la administración
Desconocimiento de herramientas y técnicas de IR	Falta de una adecuada participación del usuario
Desconocimiento en el uso de la aplicación a ser sometida a la IR	Falla en obtener el compromiso del usuario
Falta de comprensión acabada de los requerimientos	Falla en la gestión de las expectativas del usuario final
Desconocimiento del método	Cambios en los requerimientos
Falta de experiencia en la gestión de proyectos	Falta de una metodología de gestión de proyectos efectiva

Tabla 4: Comparación de riesgos identificados para la Ingeniería Reversa vs Ingeniería de Software.

En quinto lugar, se considera que es necesario profundizar en la investigación y desarrollo en la disciplina de la IR a los fines de prevenir un posible incremento de su utilización ante la creciente necesidad de producir mayor cantidad de software en el menor tiempo posible lo que redunda demasiado frecuentemente en desarrollos de productos software con un alto grado de dificultad en lo que respecta a su mantenimiento.

La caracterización final presentada puede servir de base para una identificación de riesgos de IR constituyendo una primera lista de chequeo para el proceso de identificación que luego cada organización podrá enriquecer a partir de su experiencia.

5.- Conclusión

Este proyecto se focalizó en la identificación de los riesgos propios del proceso de IR basado en el conocimiento empírico de los profesionales entrevistados que forman parte de la Industria del Software de la Ciudad de Córdoba, República Argentina.
La administración de riesgos está muy ligada a la prevención. El concepto de la prevención se fundamenta en la comprensión del proceso que requiere de la acción preventiva a los fines de poder determinar las posibles causas de error, las cuales pueden y deben ser controladas ya que todo proceso está conformado por un gran número de actividades que utilizan recursos, cada uno de los cuales debe tratarse por separado con el fin de reducir y/o eliminar las causas de los problemas. Por ello es necesario analizar los posibles fallos que pueden ocurrir en los diferentes elementos del proceso (información de entrada, tecnología disponible, mano de obra, métodos y entorno) y cómo éstos influyen en el producto resultante.
Con este propósito se decidió desarrollar una metodología que permitiera, en primera instancia, identificar los riesgos propios del proceso de IR. Esta metodología incluyó, además del estudio exploratorio

bibliográfico y de publicaciones científicas antes mencionados, la identificación del universo objeto de estudio, que no es otro que el conjunto de profesionales vinculados a la industria del software de la ciudad de Córdoba con experiencia en las actividades propias del proceso de IR; el desarrollo de un plan de recopilación de la información y el desarrollo de una herramienta a tal fin; el desarrollo de un estudio descriptivo; el análisis de la información recopilada y finalmente la caracterización de los riesgos inherentes a la IR.

El resultado obtenido fue la identificación de 153 riesgos, los cuales se agruparon, en primer lugar de acuerdo a una categorización que surgió de la nececidad de orientar al entrevistado sobre los elementos relevantes del proceso de IR a tener en cuenta a los fines de la identificación de los riesgos, fundamentalmente, por carecer de un framework específico para la IR; en segundo lugar, y por cada categoría, los riesgos con mayor nivel; en tercer lugar, por cada categoría aquellos riesgos con mayor número de reiteraciones por los diferentes entrevistados, para en último lugar identificar los dieciséis riesgos con mayor nivel de riesgo. En este punto es menester destacar que dichos riesgos se identificaron en base a la experiencia, experimentación, y en último término en la percepción de los profesionales entrevistados.

Un resultado secundario de la investigación arroja que la mayoría de los riesgos son procedentes de la falta de metodologías y herramientas específicas lo cual plantea un escenario interesante para el desarrollo futuro y la profesionalización en este ámbito de acción.

Finalmente el grupo de trabajo considera que cada organización debería realizar su propio análisis de riesgos y que los resultados obtenidos proveen, en primer lugar una lista de control que puede ser útil para conducir dichos análisis en proyectos de IR y, en segundo lugar, información adecuada para contribuir a incrementar la cultura de la gestión de riesgos en los proyectos de IR, contribuir a identificar problemas potenciales en los proyectos de IR, reflexionar sobre las consecuencias de una comprensión inadecuada de estos riesgos, la necesidad de conocer su naturaleza y priorizarlos a los fines de poder decidir si se asumen, si se transfieren, en los casos que así se aconseje y la organización esté en condiciones de hacerlo, o si se procederá a desarrollar un plan para mitigar y/o eliminar su probabilidad e impacto, y por último, a futuras investigaciones sobre el tema.

Agradecimientos.

Es el deseo del grupo de trabajo hacer llegar nuestro más sincero agradecimiento a las empresas que nos abrieron sus puertas para poder llevar a cabo este proyecto, y, fundamentalmente, a todos aquellos profesionales que nos brindaron su afectuosa, desinteresada e imprescindible colaboración.

Referencias

[Angyal et al:2012] Angyal, Lázló; Lengyel, Lázló and Charaf, Hassan. “An Overview of the State-of-the-Art Reverse Engineering Techniques”. 7th International Symposium of Hungarian Researchers on Computational Intelligence. Bulgary. 2012

[Arnuphaptrairong:2011] Arnuphaptrairong, Tharwon "Top Ten Lists of Software Project Risks :Evidence from the Literature Survey". Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol I, IMECS 2011, March 16 - 18, 2011, Hong Kong

[Bellay-Gall: 1997] Bemdt Bellay and Harald Gall; "A Comparison of four Reverse Engineering Tools" 0-8186-8162497 - IEEE - 1997.

[Blaha:1997] Blaha, Nichael. “Dimensions of DataBase Reverse Engineering”. 0-8186-8162-4/97. IEEE. 1997.

[Boehm:1991] Barry W. Boehm "Software Risk Management: Principles and Practice" 0740-7459/91/0100/0032. IEEE. 1991.

[Boehm-De Marco:1997] Barry W. Boehm and Tom DeMarco " Software Risk Management". 0740-7459/97. IEEE. 1997.

[Canfora:2007] Canfora, Gerardo and Di Penta, Massimiliano “New Frontiers of Reverse Engineering”. Future of Software Engineering(FOSE'07). IEEE Computer Society. 2007.

[Carr:1997] Marvin J. Carr "Rik Management May Not Be for Everyone". IEEE Software. May-Jun/1997.
[Carvey et al:1997] Paul R. Carvey, Douglas J. Phair and John A. Wilson " An Information Architecture for Risk Assessment and Management". IEEE Software. May-Jun/1997.
[Charette et al:1997] Robert N. Charette, Kevin MacG. Adams and Mary B. White " Managing Risk in Software Maintenance". IEEE Software. May-Jun/1997.
[Chiang-Barron:1995] Chiang, Roger H.L. and Barron, Terence M. "Quality Issues in Database Reverse Engineering: An Overview". 95.
ENGINEERING MANAGEMENT CONFERENCE - 0-7803-2799-3/95/ - IEEE - 1995.
[Chikofsky:1990] Chikofsky, E. and Cross, J. (1990). Reverse engineering and design recovery: A Taxonomy. IEEE Software.
[Crarette:1996] Robert N. Charette "Large-Scale Project Management is Risk Management". 0740-7459/96. IEEE. 1996
[D´Ambros-Lanza:2006] Marco D'Ambros and Michele Lanza "Reverse Engineering with Logical Coupling". Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06) - 0-7695-2719-1/06 - IEEE - Computer Society - 2006.
[Davis:1995] Kathi Hogshead Davis " August-II: A Tool for Step- by-Step Data Model Reverse Engineering" IEEE - 0-8186-7111-4/95 - 1995
[Davis:2001] Kathi Hogshead Davis; "Lessons Learned in Data Reverse Engineering". 1095-1350/01 - IEEE - 2001.
[de la Puerta et al:2003] Linares de la Puerta, Alberto; Minguez Gabiña, Rikardo; Ruiz Ruiz, Javier and Ispizpua Anduiza,Rafael. "State-of-the-art in Reverse Engineering in the Spanish Market". XIII ADM – XV INGEGRAF. Italy. 2003.
[DeBaud:1994] DeBaud, Jean-Marc; Moopen, Bijith and Rugaber, Spencer "Domain Analysis and Reverse Engineering". 0-8186-6330-8/94. IEEE. 1994.
[Demeyer et al:2000] Demeyer, Serge; Ducasse, Stéphane and Nierstrasz, Oscar. "A Pattern Language for
[Eilam:2005] EldadEilam "Reversing - Secrets of Reverse Engineering". Wiley Publishing, Inc - IN - EEUU - 2005.
[Fahmi-Choi:2007] Syed AhsanFahmi and Ho-Jin Choi; "Software Reverse Engineering to Requirements". 2007 International Conference on Convergence Information Technology - 0-7695-3038-9/07 - IEEE - Computer Society - 2007.
[Fairley:1994] Richard Fairley "Risk Management for Software Projects". 0740-7459/94. IEEE. 1994.
[Gannod-Cheng:1996] Gerald C. Gannod and Betty H. C. Cheng " Using Informal and Formal Techniques for the Reverse Engineering of C Programs" - 0-8186-7674-4/9 - IEEE - 1996.
[García et al:2004] Garcia, Vinicius C.; Lucrédio, Daniel and do Prado, Antonio F. "Towards an effective approach for Reverse Engineering". Proceedings of the 11th Working Conference on Reverse Engineering (WCRE'04). IEEE Computer Society. 2004.
[Garcia et al:2004] Vinicius C. Garcia, Daniel Lucr´edio, Antonio F. do Prado, Alexandre Alvaro and Eduardo S. de Almeida "Towards an effective approach for Reverse Engineering". Proceedings of the 11th Working Conference on Reverse Engineering (WCRE'04) - 1095-1350/04 - IEEE - Computer Society - 2004.
[Grubb-Takang:2005] Penny Grubb and Armstrong A. Takang; "Software Maintenance - Concepts and Practice" Second edition. WorldScientific - Singapore - 2005.
[IEEE Std 610-1990] IEEE Standard Computer Dicctionary. A compilation of IEEE standard Computer Glossaries. IEEE – 1991.
[Ishikawa:1994] Ishikawa, Kaoru. "¿Qué es el Control Total de la Calidad? La modalidad Japonesa".Editorial Norma. Colombia. 1994.
[ISO-IEC 73:2009] ISO/IEC 73:2009 Risk Management – Vocabulary. 2009
[ISO-IEC 31000:2009] *ISO 31000:2009,* Risk management – Principles and guidelines,
[ISO-IEC 31010:2009] *ISO/IEC 31010:2009* Risk Management – Risk assessment techniques
[Jahnke-Valenstein:2000] Jens H. Jahnke and Andrew Walenstein "Reverse Engineering Tools as Media for Imperfect Knowledge" 1095-1350/00 - IEEE - 2000.
[Jarzabek:2007] Stanislaw Jarzabek; "Effective Software Maintenance and Evolution. A Reuse-Based Approach". Auerbach Publications. NY - EEUU - 2007.
[Keil et al:1998] Keil, Mark; Cule, Paul E.; Kalle, Lyytinen and Schmidt, Roy C. "A framework for identifying Software Project Risks". COMMUNICATIONS OF THE ACM. November 1998/Vol. 41, No. 11. USA. 1999.
[Keller et al:1999] Keller, Rudolf K.; Schauer, Reinhard; Robitaille, Sébastien and Page, Patrick. "Pattern-Based Reverse-Engineering of Design Components". ICSE´99. Los Angeles. CA. USA. 1999.
[Klint et al:1997] Klint, P.; van den Brand, M.G.J. and Verhoef, C. "Reverse Engineering and System Renovation – An Annotated Bibliography". ACM SIGSOFT. Software Engineering Notes – vol 22 – no 1. 1997.
[Lanza-Ducasse:2003] Lanza, Michele and Ducasse, Stéphane. "Polymetric Views—A Lightweight Visual
[Lister:1997] Tim Lister "Risk Management is Project Management for Adults". IEEE Software - May-Jun/1997.
[Lu et al:2001] Lu, Chih-Wei; Chu, William C.; Chang, Chih-Hung; Chung, Yeh-Ching; Liu,

Xiaodong and Yang, Hongji. “Reverse Engineering” Department Of Computer Science –Leicester - England – 2001.
[Moyniham:1997] Tony Moyniham "Haow Experienced Project Managers assess Risk". IEEE Software. 1997.
[Patel et al:2007] Patel, Reshma; Coenen, Frans; Martin, Russell and Archer, Lawson: “Reverse Engineering of web applications: A Technical Review”. UK. 2007.
[PMBok:2013] Project Management Institute “Project Managament Body of Knowledge” 5th Edición. USA. 2013.
[PreEmptive:2007] PreEmtive Solutions. “Reverse Engineering Vulnerability Assessment”. 2007.
[Rierson-Lingberg:2004] Rierson, Leanna and Lingberg, Barbara. “REVERSE ENGINEERING OF SOFTWARE LIFE CYCLE DATA IN CERTIFICATION PROJECTS“.Federal Aviation Administration (FAA), Washington, D.C. 2004.
[Rugaber et al:2001] Rugaber, Spencer; Shikano, Terry; Stirewalt, R.E. Kurt. “Reverse reverse-engineering”. Automated Software Engineering Conference. 2001.
[Shewhart:2003] Shewhart, Walter A. “Statistical Method from the Viewpoint of Quality Control”. Dover Publications Inc. 2003. ISBN-10: 0486652327 ISBN-13: 978-0486652320
[Spring et al:2004] Spring, Nail; Wheterall, David and Anderson, Thomas. “Reverse Engineering the Internet”. ACM SIGCOMM Computer Communications Review. 2004
[Swafford:2000] Swafford, Dave; Elman, Diana; Aken, Peter and Merhout, Jeff. “Experiences Reverse Engineering Manually”. Preceedings of Fourth European Conference Software Maintenance and Reengineering. IEEE. 2000.
[Systä:2000] Systä, Tarja. “Static and dynamic reverse engineering techniques for JAVA software systems”. University of Tempere. Departament of Computational and Information Sciences. Finland. 2000.
[Tilley et al:1994] Tilley, Scott R.; Wong, Kenny; Storey, Margaret Anne D. and Müller, Hausi A. “Programmable Reverse Engineering”. Intenational Journal of Software Engineering and Knowledge Engineering Vol. 4, No. 4, pp 501-520. 1994.
[Tilley:1995] Tilley, Scott Robert. “Domain-Retargetable Reverse Engineering”. Universidad de Victoria. USA. 1995.
[Tilley:1998] Scott Tilley; "A Reverse-Engineering Environment Framework". TECHNICAL REPORT
[Tiwana-Keil:2004] Tiwana, Amrit and Keil, Mark “The one-minute risk assessment tool”. COMMUNICATIONS OF THE ACM. November 2004/Vol. 47, No. 11. USA. 2004.[Tonella et al:2007] Tonella, Paolo; Tochiano, Marco: Du Bois, Bart and Systä, Tarja “Empirical studies in reverse engineering: State of art and future trends”. Spriger Science + Business Media, LLC 2007.
[Tonella-Potrich:2005] Paolo Tonella and Alessandra Potrich; "Reverse Engineering of Object Oriented Code". SpringerScience + Business Media, Inc. - Boston - EEUU - 2005
[Tortorella-Visaggio:1997] Tortorella, María and Visaggio, Giuseppe. “ CREP- Characterizing Reverse Engineering Process component methodology” 1063-6773/97 IEEE. 1997.
[Van Geet-Demeyer:2010] Van Geet, Joris and Demeyer, Serge. “Reverse Engineering on the Mainframe: Lessons Learned from “In Vivo” Research”. IEEE SOFTWARE *Published by the IEEE Computer Society. 2010.*
[Yeh-Kuo:2002] Yeh, Dowming and Kuo, Wen-Yuan. “Reverse engineering relationship based on propagation of operations”. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR.02). IEEE Computer Society. 2002.
[Zhou et al:1999] Shikun Zhou, Hongji Yang, Paul Luker and Xudong He; "A Useful Approach to Developing Reverse Engineering Metrics". 0-7695-0368-3199 - IEEE - 1999.
Approach to Reverse Engineering”. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 29, NO.9 2003.
CMU/SEI-98-TR-005. Carnegie Mellon University - Software Engineering Institute - PA - EEUU. 1998.
Reverse Engineering”. Funded by the Swiss Government under Project no. NFS-2000-46947.96. 2000.

Datos de Contacto:

Juan Carlos Cuevas. *Universidad Tecnológica Nacional – Facultad Regional Córdoba – Maestro Lopez y Cruz Roja Argentina. Córdoba 500. e-mail:juancarloscue@gmail.com.*
Iris Nancy Gastañaga. *Universidad Tecnológica Nacional – Facultad Regional Córdoba – Maestro Lopez y Cruz Roja Argentina. Córdoba 5000. e-mail:iris.gastanaga@gmail.com.*
Inés Gimenez Zens. *Universidad Tecnológica Nacional – Facultad Regional Córdoba – Maestro Lopez y Cruz Roja Argentina. Córdoba 5000. e-mail: igimenezzens@gmail.com*
Franco Mana. *Universidad Tecnológica Nacional – Facultad Regional Córdoba – Maestro Lopez y Cruz Roja Argentina. Córdoba 5000. e-mail: manafranco@gmail.com*

Testing en el Desarrollo de Software Científico en el Marco de la Integración Continua

Salamon Alicia,
Maller Patricio,
Mira Natalia,
Departamento de Informática
Instituto Universitario Aeronáutico
Av. Fuerza Aérea, 6500, Córdoba, Argentina

Boggio Alejandra,
Pérez Sofía,
Coenda Francisco.
Departamento de Informática
Instituto Universitario Aeronáutico
Av. Fuerza Aérea, 6500, Córdoba, Argentina

as.salamon@gmail.com, pmaller@gmail.com, ncmira@gmail.com, alejandra.boggio@gmail.com, sofiabeatrizperez@gmail.com, franciscocoenda@gmail.com

Abstract

El trabajo presenta una propuesta que provee a los desarrolladores del ámbito científico técnico, una forma de trabajo que incorpora las actividades de la etapa de pruebas dentro del ciclo de vida del desarrollo de software, en el marco de la integración continua, utilizando técnicas y patrones de pruebas basados en una norma de referencia específica del ámbito científico – técnico. De esta manera se intenta mejorar la calidad de los componentes desarrollados y formalizar la práctica de incluir el testing en el marco de la integración continua para adoptarla como referencia en futuros desarrollos. Este desarrollo está basado en el marco del proyecto de investigación PIDDEF 42/11, titulado "Metodología y Framework de Gestión de Líneas Base de Integración de Aplicabilidad en el Desarrollo de Software para el Proyecto UAV"

1. Introducción

La presencia de software crítico en nuestro entorno es cada vez mayor, lo encontramos encajeros automáticos, redes de comunicaciones, equipos médicos, aviones, automóviles, camiones, plantas de energía nuclear, etc., siendo necesaria su fiabilidad y seguridad para evitar muertes y/o perjuicios socio– económicos que pueden ocasionar a las personas.

Sin embargo, hasta en los sistemas más costosos, ampliamente probados y certificados pueden ocurrir fallos, y esto se debe a que estos sistemas son grandes y difíciles de comprender ya que contienen operaciones en tiempo real, algoritmos complejos, numerosas interacciones, etc.

Esta dificultad de probar la ausencia total de errores en los sistemas críticos seguirá vigente mientras no se modifiquen algunas prácticas, hay que fomentar la reutilización de los componentes de software ya probados en otros proyectos reduciendo así la presencia de fallos, como así también promover la publicación de los desarrollos de los sistemas críticos con el fin de compartir las experiencias y generar el aprendizaje colaborativo. Existen diferentes motivos por los cuales no se difunden los desarrollos de los proyectos, algunos de ellos son porque pertenecen al ámbito del Ministerio de Defensa y otros porque es una ventaja competitiva para la empresa que lo comercializa. Por ejemplo en el caso de la industria de la aviación, cada año se incrementa la presencia del software en los componentes del avión con lo que se hace difícil garantizar la ausencia total de errores [1].

No obstante, una de las respuestas de la ingeniería de software es la aplicación de un proceso de integración continua que aporta sistematización a todo el ciclo de vida del producto, contribuyendo en el descubrimiento de errores en etapas tempranas del proceso de desarrollo de software [2].

Por otro lado, los científicos y técnicos no cuentan con conocimientos profundos sobre la ingeniería de software, lo que conlleva a que el desarrollo de software científico–técnico carezca del soporte o fundamentos necesario para implementar las prácticas más básicas de esta disciplina. [3, 4].

Ante lo presentado anteriormente, se ha realizado una investigación en el marco del proyecto PIDDEF 42/11, llamado "Metodología y Framework de Gestión de Líneas Base de Integración de Aplicabilidad en el Desarrollo de Software para el Proyecto UAV", donde se

propone una arquitectura de referencia desarrollada con componentes open source [5] que permiten automatizar el proceso de desarrollo a través de la integración continua y el control de versiones.

Para un mejor relevamiento de la situación problemática en cuestión se llevó a cabo un Workshop con los equipos de desarrolladores de software científico–técnico de la institución, en virtud de su experiencia acumulada en este dominio. Dichos grupos trabajan a diario en el desarrollo de software crítico por lo que conocen cuales son los obstáculos a sortear para conseguir un producto de calidad. El objetivo del Workshop fue elicitar las complicaciones que se les presenta a los desarrolladores en el proceso de desarrollo de software científico–técnico.

Además, se relevó que estos equipos adoptan pautas de trabajo en sus proyectos a partir de la Norma European Cooperation For Space Standarization (ECCS). En la sección ECCS–E– ST-40C de dicha norma se trata la disciplina "Software" que se centra en los procesos de requisitos y en los resultados esperados, haciendo hincapié en la relación sistema-software y en la verificación y validación de elementos software. Esta sección se complementa con otras: ECSS–M–ST–40 (Configuración y gestión de la información), ECSS–Q–ST–20 (Gestión de los procesos) y ECSS–Q–ST–80 (Aseguramiento de la calidad) [6].

En respuesta a los resultados del Workshop, y a partir de los antecedentes investigados, se construyó como solución una arquitectura con el fin de mejorar la calidad del producto como también las buenas prácticas de desarrollo de software. En este trabajo se expone cómo aplicar algunos de los patrones y técnicas de testing de la ingeniería de software en el desarrollo de software crítico mediante la arquitectura de integración continua.

2. Software Científico

El desarrollo de software científico es muy diferente al desarrollo de software comercial, ya que poseen características particulares, hay dos culturas bien definidas [7]. Una de ellas es el dominio en el que se desenvuelven, tomando en cuenta que el ámbito científico es un ambiente muy específico, ya que sólo un grupo muy pequeño poseen un conocimiento acabado acerca de la problemática a solucionar. Desde hace un tiempo se ha tratado de transferir las metodologías y prácticas de la ingeniería de software al ámbito científico pero han sido un mal ajuste ya que no se ha tratado de entender a la comunidad científica y sus limitaciones [8,9].

Segal observó la existencia de dos culturas bien marcadas, donde los científicos saben del dominio acabadamente y escasamente poseen formación en cuanto a las metodologías de desarrollo de software, documentación formal, buenas prácticas de programación, gestión de equipos mientras que los ingenieros de software saben hacer software de calidad, flexible, reutilizable [10,11].

Los problemas antes expuestos aparecen ya que los desarrolladores científicos rara vez tienen formación acerca de la ingeniería de software [3,4].

3. Integración Continua

La integración continua es una práctica en la cual los miembros de un grupo de desarrollo integran (compilación y ejecución) los distintos componentes de un proyecto con una frecuencia especificada. Cada integración se realiza de forma automática (incluyendo sus test) con el fin de detectar los errores de integración lo antes posible. Según Martin Fowler, muchos equipos de desarrollo han encontrado que este enfoque reduce significativamente los problemas de integración y permite que los equipos desarrollen software cohesivo más rápido [2].

Un escenario típico de Integración Continua [12] se compone de la siguiente manera:

- Paso 1: Un desarrollador realiza un commit de su trabajo al repositorio de control de versión a la vez que el servidor de Integración Continua verifica cambios en el repositorio, por ejemplo cada 5 minutos.
- Paso 2: El servidor de integración continua detecta los cambios en el repositorio de control de versión, extrayendo el último commit que se ha realizado y ejecutando una build script que se encarga de integrar los distintos componentes del software en desarrollo.
- Paso 3: El servidor de integración continua genera feedback con los resultados del proceso de building, el cual es enviado a los miembros que se especifique del proyecto.
- Paso 4: Sólo si el paso anterior se ejecutó correctamente. El servidor de integración continua genera feedback con los resultados del proceso de testing, el cual es enviado a los miembros que se especifique del proyecto.
- Paso 5: El servidor de integración continúa revisando cambios en el repositorio de control de versiones.

La Fig. 1. muestra las partes de este escenario que se ha descripto.

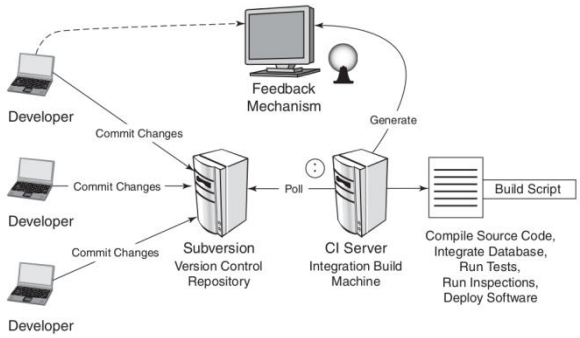


Fig. 1. Escenario típico de integración continua

4. Pruebas de Software o Testing

Como antes se mencionó las pruebas de software son uno de los módulos de la arquitectura de la integración continua que se encuentra dentro en el repositorio y es ejecutado por el servidor de integración continua luego de que el software se construyó correctamente. [2]

Según Cem Kaner [13] define el testing como una investigación técnica de un producto bajo prueba con el fin de brindar información relativa a la calidad del software. Es decir, dar cierto nivel de confianza a los interesados. Para ello Martin Fowler [2] recomienda la automatización de las pruebas como una buena práctica de la integración continua ya que permite capturar muchos bugs a lo largo del desarrollo del producto, pero la excesiva cantidad de ejecución de pruebas hace que la integración continua sea lenta entonces Paul Duvall agrega que es necesario una categorización para las pruebas con el fin de que el sistema de integración continua no demore cada vez más en completar las construcciones a medida que el proyecto avanza [12].

Por otra parte Myers [14] afirma que las pruebas de software requieren más creatividad que su diseño y se debe a que es una tarea extremadamente creativa e intelectualmente desafiante, de hecho muchos ingenieros que diseñan y desarrollan software admiten que encuentran mayor satisfacción en el diseño de las pruebas ya que implica un desafío encontrar errores; es por ello que algunas personas lo llaman "quebrar el software". Pero esta acepción es errónea ya que los ingenieros de prueba realmente no rompen software sino simplemente exponen los errores que ya existen, al refutar o falsear el supuesto de que el software es impecable a través de los resultados que observan, luego de aplicar pruebas estructuradas, de deducción lógica y de experimentos.

Sin embargo, los errores que se encuentran sólo son una parte de las pruebas ya que los ingenieros de pruebas además agregan valor al proceso de desarrollo mediante la validación del software y la detección de errores, facilitándoles a los desarrolladores información para evitar errores y a la administración para evaluar el riesgo.

Para cumplir correctamente el labor de los ingenieros de pruebas de software es necesario poseer ciertas capacidades intelectuales y personales, James McCaffrey define ocho cualidades: [15]

- Pasión por el análisis y las pruebas
- Habilidades técnicas
- Capacidad Intelectual pura
- Capacidad para priorizar y organizar
- Capacidad de adaptación y aprendizaje
- Capacidad para trabajar sin supervisión directa
- Capacidad para comunicarse
- Capacidad para comprender la estrategia de negocios

Como los sistemas de software crítico aumentan su tamaño y complejidad a medida que se van desarrollando, el valor de las pruebas de software se está incrementando y convirtiendo en una actividad cada vez más importante.

Por otra parte, para el desarrollo de sistemas críticos las normas ECSS, por ejemplo, define métodos y técnicas de testeo basándose en diversos patrones de testing.

Sin embargo, dentro de los ámbitos de desarrollo de software científico–técnico no se visualiza una amplia utilización de pruebas, a pesar de que muchos desarrolladores de estos sistemas reconocen el valor e importancia de estos en el desarrollo de sistemas [16]. También se sigue visualizando una problemática en la cual los equipos de desarrollo y en especial los de testing no se encuentran envuelto desde el inicio del proyecto, es decir, desde la definición de los requerimientos. Esto ha podido ser corroborado en las observaciones que se realizaron en grupos de I+D donde la inclusión, en especial del equipo de testing, suele ser tardía [17].

5. Proceso de prueba

Un proceso de pruebas formal, está compuesto por cinco etapas [18].

1. Planeación de pruebas.
2. Diseño de pruebas.
3. Implementación de pruebas.
4. Evaluación de criterios de salida.
5. Cierre del proceso.

A continuación se presenta en la figura 2 el proceso de pruebas identificando sus entregables de entradas y salidas.

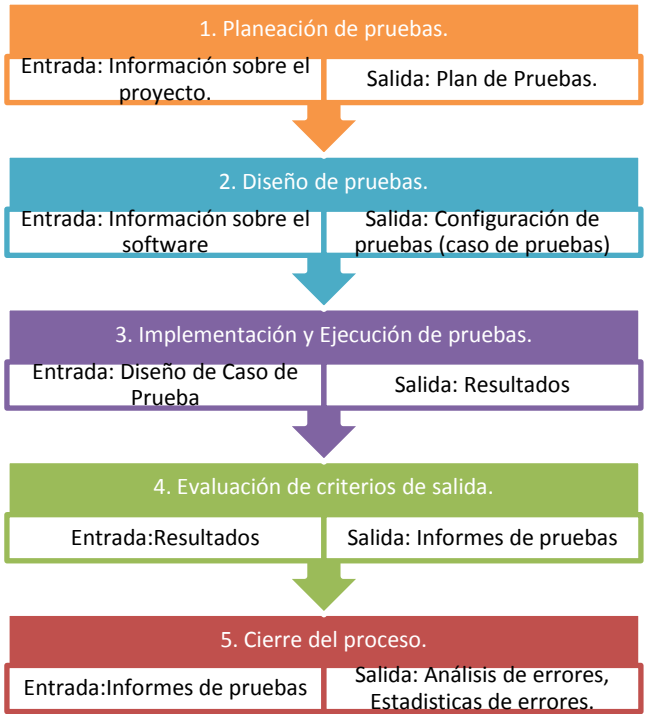


Fig. 2. Proceso de prueba.

1. Planeación de Pruebas: se genera el artefacto o entregable denominado "Plan de Pruebas" el cual debe estar conformado mínimamente por los siguientes aspectos:
    - Alcance de la prueba: en esta etapa se determina que funcionalidades del sistema serán probadas durante el transcurso de la prueba como también el alcance detallado del proceso de pruebas, la prioridad con la que las funcionalidades deben probarse y la profundidad de las pruebas. Este listado de funcionalidades a probar se extrae del análisis de riesgos realizado en una etapa previa donde se tienen en cuenta variables tales como el impacto que podría ocasionar la falla de una funcionalidad y la probabilidad de falla de una funcionalidad.
    - Tipos de Prueba: se define qué tipos de pruebas requeriría el producto. No todos los productos de software requieren la aplicación de todos los tipos de pruebas que existen, es por ello que el líder de pruebas deberá plantearse qué tipos de prueba son aplicables al proyecto en evaluación.
    - Estrategia de Pruebas: como no es viable probar todas las posibles combinaciones de datos, es necesario establecer a través de un análisis de riesgos sobre que funcionalidades se debe centrar mayor atención. Adicionalmente, una buena estrategia de pruebas debe indicar los niveles de pruebas o ciclos que deberán emplearse y la profundidad a aplicar para cada nivel de pruebas definido como también definir los criterios de entrada y salida para cada ciclo de pruebas a ejecutar.
    - Criterios de Salida: se debe especificar de manera formal, bajo qué condiciones se puede considerar que una actividad de pruebas fue finalizada. Los criterios de salida se deben definir para cada nivel de pruebas a ejecutar. Por ejemplo el porcentaje de funcionalidades de alto riesgo probadas con éxito, número defectos críticos y/o mayores aceptados, etc.
    - Otros aspectos: se debe incluir una estimación de tiempos, los roles y/o recursos que harán parte del proceso, la preparación del entorno de pruebas, cronograma, etc.
2. Diseño de Pruebas: una vez elaborado y aprobado el "Plan de Pruebas", el equipo de trabajo debe iniciar el análisis de toda la documentación existente con respecto al sistema para diseñar los casos de prueba. Los entregables claves a revisar son: casos de uso, historias de usuario, arquitectura del sistema, diseños, manuales de usuario, manuales técnicos, etc. El diseño de los casos de prueba, debe considerar la elaboración de casos positivos y negativos. Estos últimos permiten validar cómo se comporta el sistema ante situaciones atípicas y permite verificar la robustez del sistema, atributo que constituye uno de los requerimientos no funcionales indispensable para cualquier software crítico. Por último, es necesario definir cuáles son los datos de prueba necesarios para la ejecución de los casos de prueba diseñados.
3. Implementación y Ejecución de Pruebas: en esta etapa se inicia con la creación de los datos de prueba necesarios para ejecutar los casos de prueba diseñados en la etapa 2. La ejecución de estos casos, puede realizarse de manera manual o automatizada; en cualquiera de los casos, cuando se detecte un fallo en el software, este debe ser documentado y registrado en una herramienta que permita gestionar los defectos. Una vez el defecto ha sido corregido, es necesario realizar un re–test que permita confirmar que el defecto fue solucionado de manera exitosa. Por último, es

indispensable ejecutar un ciclo de regresión que permita garantizar, que los defectos corregidos en el proceso de depuración no hayan ocasionado más defectos en el software.

4. Evaluación de Criterios de Salida: los mismos son necesarios para determinar si es posible dar por terminado un ciclo de pruebas. Por lo que es conveniente definir una serie de indicadores que permitirán comparar los resultados obtenidos contra los indicadores definidos, sí los resultados obtenidos no superan los indicadores definidos, no es posible continuar con el siguiente ciclo de pruebas. Coexisten varios tipos de criterios de salida por ejemplo cubrimiento de funcionalidades en general, cubrimiento de funcionalidades críticas para el sistema, número de defectos críticos y mayores detectados, etc.
5. Cierre del proceso: principalmente en esta etapa se elabora un informe con del análisis de errores encontrados a lo largo del proceso de prueba como también una estadística de los errores más frecuentes dejando lecciones aprendidas para aplicar en futuros proyectos. Además, se comprueba que se hayan cerrado todas las incidencias reportadas, se verifica que los artefactos de software hayan sido entregados y aprobados, etc.

5.1. Técnicas de pruebas de software crítico

Según la Norma ECSS en el documento ECSS–E–HB–40A [6] y SWEBOK [19], describen las siguientes técnicas de testeo.

- Introducción de fallos: consiste en ingresar fallos en el hardware y software de manera intencional con el fin de probar los mecanismos de tolerancia de fallos, para que la misma sea eficaz es necesario contar con una cantidad de tiempo significativa de un analista que posea una buena perspectiva del software y con herramientas automatizadas así como lo propone la integración continua. Esta técnica apunta a la evaluación de la robustez del sistema.
- Prueba de estrés: en este caso se busca sobrecargar el sistema de manera que active sus mecanismos de manejo de errores y recuperación de sí mismo. Por ejemplo son objetos a controlar los buffers internos o variables dinámicas, pilas como también el uso de los recursos del sistema, tiempo CPU, espacio de almacenamiento y memoria. Esta técnica hace hincapié en probar el rendimiento del sistema.
- Partición equivalente y clases equivalentes: se divide el dominio de entrada de un software en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia. Normalmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. Los mismos criterios se aplican a las salidas esperadas. El objetivo de esta técnica es generar resultados en todas y cada una de las clases, de esta manera se reduce el número de casos de pruebas.
- Análisis de valores límites: el propósito es proporcionar casos de prueba para detectar y eliminar los fallos que se producen en los límites de los parámetros para ello es necesario identificar las clases equivalentes. Esta técnica se emplea en conjunto con la técnica antes mencionada en virtud de garantizar la cobertura de las pruebas especificadas. El análisis de valores límites permite detectar fácilmente las fallas de cálculo, tamaño de la matriz, punteros nulos, iteraciones de bucle.
- Previsión de Errores: este tipo de test es diseñado por ingenieros de software cuyo objetivo es anticiparse a las posibles fallas del sistema. Para ello, se utiliza la información histórica de aquellas fallas que se descubrieron en proyectos anteriores.
- Tabla de decisión: la tabla de decisión representa lógicamente los imput y output. Los casos de test se derivan considerando cada combinación posible entre input y output.
- Observación del Usuario: la especialización heurística, también conocida como inspección de usabilidad, es aplicada de manera sistemática a la observación del uso de sistemas en condiciones controladas con la finalidad de determinar cómo los usuarios finales utilizan el sistema y sus interfaces.

5.2. Patrones de Testing

La IEEE en su libro SWEBOK define una serie de patrones de testing [19]. De estos patrones de testing que se proponen a continuación se toman solamente algunos para ser incluidos en la arquitectura propuesta de este trabajo.

- Test Unitario: Consiste en la verificación de manera aislada de partes del software que son factibles de ser testeada. Estas pruebas son atómicas y son escrita normalmente por los mismos desarrolladores.
- Test de Integración: Se ocupa de verificar la interacción entre los componentes de software. En la actualidad una práctica que está tomando mayor auge es la integración incremental de testing, lo que permite ir verificando los resultados de integración a través de cada iteración sin necesidad de hacer una gran integración al final del desarrollo.

- Test de Regresión: Es una re–testeo selectivo de los componentes de un sistema con el fin de verificar que las modificaciones introducidas no produzcan efectos no deseados y que los componentes sigan cumpliendo con los requerimientos para los que fueron desarrollados.
- Test de estrés: En este tipo de test evalúa la carga máxima y minina de trabajo que soporta el sistema y los mecanismos de protección que poseen los sistemas críticos.
- Test de Recuperación: En este tipo de test se evalúa las capacidades del sistema de reiniciarse después de que se produce una falla o algún otro tipo de contingencia.
- Test de Sistema: Este test evalúa el comportamiento del sistema en general. Suele usarse para evaluar los requerimientos no funcionales, como seguridad, velocidad, confiabilidad, etc.
- Test de aceptación: Se evalúa si el sistema los criterios de aceptación a través del chequeo del comportamiento del sistema contra los requerimientos del cliente.
- Test de seguridad: Se verifica que el sistema se encuentre protegido ante ataques externos. Se verifica la confidencialidad, integridad y disponibilidad del sistema y la información.
- Test de Humo: Consiste en realizar pruebas rápidas con el fin de testear que la funcionalidad básica del sistema se encuentra estable.

6. Propuesta

La propuesta de este trabajo es elaborar una solución que incorpore las actividades de la etapa de pruebas del software al proceso de desarrollo del software científico técnico, sugiriendo algunas técnicas y patrones de testing en el marco de trabajo del PIDDEF 42/11. Cabe destacar que los patrones recomendados están contemplados en la norma ESCC y SWEBOK.

6.1. Técnicas y patrones sugeridos

De acuerdo a las características mencionadas de los grupos de desarrollo científico–técnico, como ser la escasa formación en cuanto a la ingeniería de software se ha optado por seleccionar técnicas sencillas de aplicar y que permitan vislumbrar resultados efectivos rápidamente.

Las técnicas que se proponen utilizar son: partición equivalente y clases equivalentes, y análisis de valores límites. Además, se seleccionan los patrones de pruebas de unidad, de integración, regresión, y de humo. Las técnicas y patrones seleccionados deben ser aplicados siguiendo la forma de trabajo que se describe a continuación.

6.2. Forma de trabajo

Se adopta el proceso planteado para la etapa de pruebas:

- Planificar las pruebas a realizar: se debe generar el Plan de Pruebas, en el mismo se especifican los siguientes aspectos: alcance de cada prueba, tipos de pruebas, estrategias de pruebas, criterios de salida, estimación de tiempo de las pruebas, los roles y recursos involucrados en cada una de las pruebas, cronograma, etc.
- Pre–definir los resultados esperados: en esta actividad se utilizan las técnicas de partición equivalente y clases equivalentes y análisis de valores límites y los patrones asociados.
- Construir las pruebas: se construyen los componentes de pruebas para ser ejecutadas en el marco de la integración continua. Cabe aclarar que los componentes de prueba una vez construidos deben ser ubicados en un lugar específico de la estructura de directorio del servidor de integración continua.
- Ejecutar las pruebas: la configuración del servidor de integración continua genera la ejecución del código y de los componentes de prueba asociados al mismo.
- Evaluar las pruebas: determinar si el componente de prueba es adecuado para la validación del código.

6.3. Equipo involucrado

Es importante determinar quien o quienes del equipo de desarrollo poseen un perfil más cercano al descripto por McCaffrey. Se sugiere buscar personas que sean creativos para diseñar las pruebas, que logren el objetivo de validar software encontrando errores y proporcionando a los desarrolladores información para evitar errores futuros [15].

Asimismo existen diferentes roles que participan en un proyecto de pruebas. Una persona puede asumir más de un rol en un proyecto y tomar diferentes roles en proyectos distintos. En la medida de que una persona es capaz de asumir distintos roles va creciendo en conocimiento y experiencia.

Se identifican tres tipos de roles dentro de un equipo de testing: tester de software, líder de testing de software y consultor.

Tester de software es quien diseña, ejecuta e informa el resultado de las pruebas sobre un producto de software. Los testers integran el equipo de pruebas e interactúan con el equipo de desarrollo y la gerencia de proyectos.

Líder de testing de software es quien está a cargo del equipo de testing y tiene conocimiento y experiencia en

el diseño, ejecución y reporte de pruebas sobre un producto de software, pero también es capaz de estimar, planificar y dar seguimiento a un proyecto de pruebas.

Consultor. Es un rol adecuado para aquellas personas capaces de dirigir un equipo de pruebas y orientar sobre mejores prácticas en testing [16].

En el ámbito científico en la mayoría de los casos quien realiza las pruebas es la misma persona que desarrolla el componente a ser probado, pero no es conveniente ya que la persona conoce como está programado, es por ello que es recomendable que los miembros del equipo asuma uno o más roles diferente y logre cumplir con los objetivos de las pruebas.

6.4. Las pruebas en el marco de la integración continúa

La arquitectura de integración continua del PIDDEF 42/11 se compone de un servidor de integración continua (Cruise Control), un sistema de control de versiones (Git) y un sistema de construcción (Ant).

El proceso da inicio con la creación de un repositorio vació donde se va a jerarquizar el desarrollo de software. Para esto se arma una estructura de directorios donde se separa el código fuente de la suite de test y a su vez, esta última se la jerarquiza en componente por componente de a acuerdo a la funcionalidad que se pretende validar y verificar [12]. Los desarrolladores remiten sus modificaciones al repositorio del proyecto, el cual alimenta al servidor de integración continua en su proceso de construcción. Este último presenta los resultados del proceso de construcción a través de su interfaz web.

En este proceso de construcción, se recomienda que los test unitarios sean ejecutados con cada construcción [20], mientras que aquellos componentes que han finalizado sus test no es necesario que los corran nuevamente. En cuanto a los test de integración y de humo se sugiere que sean ejecutados una vez que se ha finalizado con el desarrollo del componente. Estas pruebas deberán ser planificadas y en la medida de lo posible ser ejecutadas fuera del horario laboral, por ejemplo a la noche [20]. Por último, los test de regresión deberían ser ejecutados cuando se produzca un fallo en las pruebas de humo e integración. Este tipo de test se recomienda que sea ejecutado los fines de semana.

Los test unitarios van a permitir evaluar los input–output, permitiéndole a los desarrolladores de software científico–técnico verificar y validar sus modelos [20]. Los test de integración y humo van a permitir corroborar la interacción de los nuevos módulos que se agreguen al sistema y verificar que se siga manteniendo la estabilidad de este [19,20]. Las pruebas de regresión van a permitir determinar que las modificaciones introducidas en el sistema no produzcan fallas no deseadas en otras partes del sistema y que este siga cumpliendo con los requerimientos iniciales [21]. A través de esta planificación se busca evitar que los tiempos de integración se incrementen exponencialmente a medida que el desarrollo avanza.

7. Resultado

El modelo propuesto se ha presentado al grupo de desarrollo que ha participado en el Workshop logrando captar su interés y una amplio involucramiento, en la exposición se vieron comprometidos y con deseos de experimentar la implementación de esta solución, incorporando a su forma de trabajo el proceso formal de pruebas descripto con sus técnicas y patrones. El próximo paso es realizar la experiencia evaluando los indicadores definidos para tal fin.

8. Conclusión

La investigación y desarrollo que se ha llevado a cabo en este trabajo ha permitido obtener un conocimiento mayor acerca de la problemática que viven los equipos de desarrolladores de software científico y especialmente de aquellos grupos que desarrollan sistemas críticos en el cual no se permiten fallas ya que las consecuencias implican el riesgo de vidas humanas y una considerable suma de dinero.

Por lo tanto, a través de este trabajo se busca plantear un modelo de referencia que permita mejorar el desarrollo de componentes de software crítico en el marco de la integración continua donde se busca integrar y testear automáticamente el desarrollo de un proyecto a medida que este avanza. De esta manera se posibilita el descubrimiento de errores a corto plazo sin tener que esperar a realizar las pruebas finales. También se espera aportar una reducción del re trabajo y la optimización de los tiempos de desarrollo.

Este trabajo continuará con la realización de una experiencia con el grupo de desarrollo de software científico – técnico para la aplicación de esta nueva forma de trabajo a través de la arquitectura desarrollada en el proyecto PIDDEF 42/11.

9. Referencias

[1] Sánchez Domínguez J. C. y Rodríguez Dapena P. “¿Cómo verificar la calidad del software en sistemas críticos?”, *Ponencia del IV Congreso Gallego de la Calidad*, Santiago de Compostela, España (2003).

[2] Continuous Integration por Martin Fowler, http://martinfowler.com/articles/continuousIntegration.html

[3] D. Kelly y R. Sanders. “Assessing the Quality of Scientific Software”. *First International Workshop on Software Engineering for Computational Science and*

Engineering. Leipzig, Germany. (2008).
[4] Carver J., Kendall R., Squires S., y Post D. "Software Development Environments for Scientific and Engineering Software: A Series of Case Studies."*IEEE Computer Society*, Washington, DC, USA, (2007).
[5] The Open Source Initiative, http://opensource.org/
[6] Normas ECSS http://www.ecss.nl/
[7] Segal J. "Scientists and software engineers: a tale of two cultures". *PPIG University of Lancaster*, UK, (2008).
[8] Kelly D. "A Software Chasm: Software Engineering and Scientific Computing."*IEEE Computer Society.* Los Alamitos, CA, USA. (2007).
[9] Segal J. "Models of scientific software development." *First International Workshop on Software Engineering in Computational Science and Engineering*, Leipzig, Germany. (2008).
[10] Basili V. et al. "Understading the High-Performance-Computing Community: A Software Engineer's Perspective." *IEEE Software Computer Society*. vol. 25, no. 4, 29-36. Los Alamitos, CA, USA. (2008).
[11] Hannay J. E., MacLeod C., y Singer J. "How Do Scientists Develop and Use Scientific Software?" *IEEE Computer Society*, Washington, DC, USA, (2009).
[12] Duvall P., Matyas S. y Glover Andrew. *Continuous Integration: Improving Software Quality and Reducing Risk*. (2007).
[13] Kaner C., Falk J. y Nguyen H. *"Testing Computer Software"* International Thomson Computer Press. (1993)
[14] Myers, G. *"The Art of Software Testing".* John Wiley & Sons. 2004.
[15] McCaffrey J. *"What Makes A Good Software Tester?"* 2008.
[16] Nguyec-Hoan L., Flint S. y Sankaranarayana R.*"A Survey of Scientific Software Development".* ESEM, Bolzano-Bozen, Italia(2010).
[17] Humphrey W., Snyder T. y Willis R. "Software Process Improvement at Hughes Aircraft". *IEEE Software Computer Society Vol. 8 Issue 4* (1991).
[18] Jacobson I., Booch G. y Rumbaugh.J. *"The Unified Software Development Process."* Addison-Wesley Professional. 1999.
[19] Bourque P. y Fairly R. *"SWEBOK v 3,0 Guide to the Software Engieering Body of Knowledge."* IEEE. Accedido el 2014.
[20] Stephen P. B., Appleton B. y Brown K. *"Software Configuration Management Patterns: Effective Teamwork, Practical Integration Paperback"*. Addison Wesley 2002.
[21] Cardinal, M. *"The Hidden Roles of Software Architects".* MSDN. 2008.

Implementación de Sistemas de Información Basados en Ontologías: Análisis de Tecnologías

Agustín Martínez, Santiago Sosa, Emiliano Reynares, Ma. Laura Caliusco
CIDISI - UTN - Facultad Regional Santa Fe

Abstract

En el campo de estudio de ontologías y web semántica, el mercado actual ofrece una gran cantidad de estándares/lenguajes que permiten la implementación correcta de modelos conceptuales ontológicos, su almacenamiento, razonamiento y la ejecución de consultas sobre los mismos. Aun así, existe una brecha entre dichos estandares/lenguajes y las soluciones de software que dan soporte completo al diseño y desarrollo de sistemas de información basados en ontologías, tanto en los aspectos de rendimiento como en la completitud y la expresividad de la información contenida en dichos sistemas. En este trabajo se realiza un análisis, comparación y evaluación de un subconjunto de herramientas actuales que dan soporte al uso, almacenamiento y razonamiento de ontologías utilizando un caso práctico y observando su evolución con el uso de herramientas simples hasta llegar a las complejas.

1. Introducción

En esta última década, donde la información toma un papel preponderante, las organizaciones han encontrado un obstáculo importante en la gestión del conocimiento, sea propio o de terceros, que proviene de la imposibilidad de realizar dicha gestión con los métodos tradicionales de almacenamiento de datos (conocimiento radicado en las personas, registros archivados en papel, datos en bases de datos relacionales, etc.). En la lucha por mitigar esta dificultad aparecen herramientas de gestión de conocimiento como las ontologías (en torno al auge de la web semántica) que teóricamente desafían a la forma tradicional de manejo de la información permitiendo una explotación rica en beneficios para las organizaciones. Se dice teóricamente porque el soporte real de gestión de conocimiento en base a ontologías no incluye, aún, la totalidad de los contenidos teóricos alcanzados por la comunidad científica, como sucede a menudo en el ámbito de investigación, donde la teoría y los estándares generalmente ven la luz antes que las herramientas que les brinden soporte.

Al ser una tecnología relativamente nueva, muchas de las organizaciones que utilizan sistemas ontológicos hoy en día la ven como un reemplazante de la base de datos estructurada tradicional. Usualmente, se construyen las ontologías para ser utilizadas en sistemas ya implementados, y en el afán de lograr esta adaptación se dejan de lado muchas características que permiten explotar al máximo el potencial de las mismas. Las organizaciones que aprenden el verdadero potencial del concepto centran el desarrollo del sistema en el uso de la ontología logrando así mayor semántica en los resultados y verdadera generación de conocimiento que permanecería oculto en una base de datos tradicional. Es decir, en la ingeniería de software, las ontologías puede ser usadas de dos maneras diferentes: (1) como Artefactos Arquitectónicos, y (2) como Fuentes de Información [1].

El desarrollo de sistemas de información basado en ontologías es un área de investigación que todavía requiere de madurez. Varios avances se han realizado en la definición de estrategias para dar soporte al análisis y modelado de los mismos. Sin embargo, en la literatura se encuentran pocos trabajos que aborden la etapa de implementación de los mismos. El principal objetivo de nuestra investigación fue el estudio de las tecnologías para la implementación de sistemas de información basados en ontologías que ante grandes volúmenes de información tenga el rendimiento esperado de un sistema de información computarizado. Por esto, se considera que el contexto que describe este informe es similar al que encuentra cualquier grupo de desarrolladores (novatos en el tema) a la hora de llevar a la práctica el uso de ontologías y refiere a las dificultades de expresión, estructuración, desarrollo y explotación del conocimiento. Estas dificultades involucran el nivel de expresividad de los lenguajes de construcción y consulta actuales, los formatos de confección de la estructura de las ontologías, las herramientas y lenguajes de programación para "materializar" esas ontologías y cuestiones no menores de rendimiento, escalabilidad y mantenimiento del sistema de base ontológica.

Para cumplir con el objetivo planteado, este trabajo se estructura de la siguiente manera. En la Sección 2 se describen las principales características de un sistema de información basado en ontologías. En la Sección 3 se describe la selección de las herramientas de implementación que existen en el mercado. En la Sección 4 se discute la experiencia de desarrollo que se fue llevando a cabo con dichas

herramientas. En la Sección 5 se presenta una discusión del trabajo para finalmente en la Sección 6 presentar las conclusiones.

2. Sistemas de Información Basados en Ontologías

La utilización de tecnologías semánticas constituye el núcleo de los futuros sistemas de información de las organizaciones. La utilización de las tecnologías semánticas permitirá definir sin ambigüedades el significado de las conceptos del negocio, haciéndolos comprensibles tanto para los expertos humanos como para los sistemas de software.

La información relevante de un dominio dado se encuentra fuertemente relacionada con las reglas de negocio. Las reglas de negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización y son de vital importancia para alcanzar sus objetivos estratégicos [2].

Las reglas de negocio de una organización se encuentran generalmente embebidas en el código procedural de una aplicación de software. Sin embargo, los entornos dinámicos de las organizaciones actuales imponen cambios frecuentes en dichas reglas, implicando intensivas tareas de mantenimiento que requieren la identificación y modificación del código fuente relevante a cada uno de los cambios. Esta situación expone la necesidad de tecnologías que soporten la propagación rápida de las nuevas reglas de negocio en el sistema de software, a fin de incrementar su flexibilidad, extensibilidad y facilidad de mantenimiento.

Encapsular las reglas de negocio en un componente de software específico es un enfoque para tratar el problema anterior. Las bases de reglas aparecen como una solución posible, pero la utilización de diferentes atributos para representar la misma entidad del mundo real y la definición de reglas basadas en nociones incompatibles de un mismo concepto son problemas que pueden afectar la interacción entre las reglas [2].

Las ontologías, utilizadas como artefactos destinados a representar explícitamente la semántica de los datos, podrían ser utilizadas para tratar ambos inconvenientes: encapsular las reglas de negocio y evitar conceptos malentendidos [3][4]. Surgen así los sistemas de información basado en ontologías, los cuales utilizan las ontologías en tiempo de diseño como un componente de datos.

Dado que tales ontologías deben ser desarrolladas y mantenidas junto a los restantes componentes de software, la metodología de desarrollo de la ontología debe ser considerada en el contexto del proceso de desarrollo de software. Además, para que los sistemas de información basados en ontologías sean una realidad es necesario encontrar una forma de implementación de los mismos que haga que su rendimiento sea comparable con el de los sistemas tradicionales.

3. Selección de Herramientas para la Implementación de Sistemas de Información Basados en Ontologías

Como primer paso fue necesario realizar un relevamiento de las diferentes herramientas que existen actualmente en el mercado y seleccionar, dentro de las más difundidas, una cantidad factible tomando como punto principal de decisión las características relevantes a ser analizadas.

Se dividió el análisis de las diferentes soluciones de software que den soporte al desarrollo de sistemas basados ontologías en tres partes: a) análisis del soporte para el almacenamiento; b) para el razonamiento; c) para la consulta. A su vez, en forma paralela, fue necesario tener en cuenta si dichas características eran implementadas por la herramienta sobre memoria principal o memoria secundaria, con el objetivo de examinar el comportamiento de la ontología y el sistema con grandes volúmenes de datos que deban ser procesados.

Con el fin de realizar una comparación completa, se utilizó un caso práctico que necesitaba de las diferentes funcionalidades generales anteriormente definidas (almacenamiento, consulta y razonamiento).

Sobre el dominio del problema indicado en dicho caso práctico, se partió de un enunciado que necesite de una implementación integrada. En primer lugar, se buscó que dicha implementación dé solución a las diferentes tareas en forma sencilla para cualquier desarrollador que desempeñe sus actividades utilizando el paradigma orientado a objetos.

Luego, en forma gradual, se fueron incorporando al sistema diferentes herramientas que alteren características particulares deseables y no deseables del mismo, tanto desde un punto de vista general (ej. rendimiento) como desde un enfoque que tome en cuenta aquellas particularidades propias de un sistema basado en ontologías (ej. expresividad).

A su vez, para cada herramienta se realizó un análisis puntual de fortalezas y debilidades, examinando las potencialidades de cada una en forma general para solucionar otros casos particulares diferentes al enunciado. De esta forma, además de describir la aplicación a casos complejos, se logran encontrar aplicaciones para sistemas más específicos y con necesidades particulares (ej. exclusivamente razonamiento). En la tabla 1 puede verse para cada herramienta, las características que cumple y con qué tipo de memoria trabaja.

Tabla 1 – Herramientas y características

Herramienta	*Característica*			*Ubicación de Memoria*
	Consulta	*Almacenamiento*	*Razonamiento*	
Jena	X	Solo archivo		Primaria
OWLAPI	X	Solo archivo		Primaria
Pellet			X	Primaria
TDB	X	Tripletas		Secundaria
Stardog	X	Tripletas	X	Secundaria

4. Ejecución del Experimento

Para los diferentes casos de desarrollo se tomó una misma ontología compuesta por 2117 axiomas (Modelo conceptual más 286 Individuos) implementada en el lenguaje OWL 2 (Web Ontology Language) y SWRL (Semantic Web Rule Language). Esta ontología fue creada para dar soporte a la asignación de becas de ayuda económica de alumnos pertenecientes a la UTN-Facultad Regional Santa Fe y debido al nivel de expresividad requerido para representar el dominio del problema se optó por utilizar extensiones del lenguaje RDF como lo son OWL y SWRL ya que de lo contrario no hubiese sido posible la implementación.

Los diferentes desarrollos se llevaron a cabo bajo la misma configuración de middleware y hardware. El sistema operativo elegido fue Windows en su versión 7 (Edición Professional con arquitectura de 64 bits), el lenguaje de programación utilizado fue Java y su framework de desarrollo (Java 7 Update 45 CPU JRE y JDK), el entorno de desarrollo integrado (IDE) elegido fue NetBeans en su versión 7.3.1. Para el desarrollo y prueba del archivo OWL que contenía la ontología se utilizó el editor y framework Protegé en su versión 4.3 con razonador Pellet versión 2.2.0.

En cuanto a API's (Java) incluidas en el desarrollo se utilizaron Swing y AWT como soporte para la interfaz gráfica, también Apache jena versión 2.11.1 (y su correspondiente versión de TDB y Pellet), OWLAPI versión 3.4.9 y Stardog SNARL 2.1.2 (Stardog Native API for the RDF Language) para el manejo de la ontología dentro de Java, y finalmente para el manejo de configuración en XML se utilizó la librería Xstream versión 1.4.7.

Cabe aclarar que debido al tiempo de duración de la experiencia la versión de Java se actualizó automáticamente según el esquema programado por Oracle.

En cuanto a hardware diferenciamos el utilizado en la etapa de desarrollo y el utilizado en la etapa de implementación y prueba. En la primera etapa se utilizaron diferentes configuraciones debido a la portabilidad y la necesidad de trabajo a distancia, sin embargo la segunda etapa se llevó a cabo con la misma configuración de hardware de modo de poder obtener métricas representativas del experimento. Las características se detallan a continuación:

- Procesador: AMD Athlon II X2 240
- Memoria principal: 2GB DDR2 PC2 6400 240 pinn DIMM
- Memoria secundaria: 320GB HDD SATA 3 Gb/s 16Mb Cache.

A nivel general la arquitectura del sistema está compuesta por (Ver figura 1): 1) Capa GUI (Interfaz Gráfica), 2) Capa de lógica (Acceso y modificación de la ontología, hilo principal del sistema), 3) Capa de acceso a datos (Configuraciones, consultas y parametros), 4) DTOs (Data Transfer Objects - Objeto de Transferencia de Datos: Objeto que transporta datos entre procesos), 5) APIs.

4.1. Primer Desarrollo

En primera instancia se implementó una solución con las herramientas más recomendadas en la comunidad: los framework Jena[1] (Java) y OWLAPI[2] (Java).

Dichos frameworks son los más utilizados en los últimos años para la introducción al desarrollo de sistemas basados en ontologías [5-7]. Mediante ellos se puede consultar toda la información de la ontología a través de llamados a métodos puramente orientados a objetos, desde listar individuos hasta agregar nuevos axiomas, sin necesidad de conocer muy en profundidad los conceptos complejos relacionados a ontologías.

Este primer desarrollo, permite la integración a Jena del razonador Pellet[3] para realizar las inferencias en base a los axiomas definidos internamente en la ontología, todo en memoria principal cargando la misma desde un archivo OWL en formato RDF.

[1] https://jena.apache.org/
[2] http://owlapi.sourceforge.net/
[3] http://clarkparsia.com/pellet/

4.2.Segundo Desarrollo

En segunda instancia, se reemplazó el uso de algunos métodos de los frameworks para acceder a información de la ontología por consultas SPARQL (acrónimo recursivo del inglés "Protocol and RDF Query Language") con Jena y su motor de consultas ARQ[4], logrando una mejora en el rendimiento en casi veinte veces.

A través de la API de Jena, utilizando la base de conocimiento del razonador Pellet para la OWLAPI, se ejecutaron consultas en SPARQL para simplificar la generación de conocimiento desde la ontología, evitando el uso de instrucciones Java.

Luego, se adaptó la solución para que permita el uso de TDB[5] (Java). Mediante TDB se pueden realizar consultas SPARQL y llevar el almacenamiento de tripletas a memoria secundaria con la particularidad de que no es posible realizar inferencias automáticamente. Al no incluir un razonador, TDB debe trabajar con la memoria principal y un razonador acoplado (ej. Pellet, integrado a Jena).

Con esta tercera variante, se logró un rendimiento similar a la anterior configuración.



Figura 1 – Arquitectura del Sistema

4.1. Tercer Desarrollo

El último paso consistió en utilizar Stardog[6]. Dicha herramienta consiste en un sistema completo de gestión de bases de datos orientadas a grafos, con razonador incorporado que trabaja en memoria secundaria, al igual que el almacenamiento. Debido a esto, fue necesario realizar una adaptación más compleja, eliminando por completo Jena, Pellet y TDB de la implementación, pasando a usar una comunicación directa al servidor Stardog, en forma similar a la comunicación con una base de datos SQL.

Los resultados de esta implementación tuvieron una baja del rendimiento, justificada por el uso de memoria secundaria.

Para hacer un análisis un poco más profundo de la herramienta, se cargaron a Stardog y a Jena ontologías de gran tamaño (archivos OWL varios ordenes de magnitud mayor a la ontología del trabajo práctico). Mientras que Stardog mantuvo aproximadamente su relación tamaño/rendimiento, las soluciones con Jena y Pellet no llegaron a caber en memoria principal y no lograron ejecutarse.

En la tabla 2 pueden apreciarse los diferentes rendimientos que se lograron con cada implementación.

Tabla 2 - Rendimientos

Herramienta	*Rendimiento (Tiempo)*
Jena + OWLAPI + Pellet	2' 35"
Jena + OWLAPI + Pellet + SPARQL	8"
Jena + OWLAPI+ Pellet + SPARQL + TDB	11"
Stardog	20"
Stardog (c/ aumento de individuos)	Estable
Jena (c/ aumento de individuos)	Desbordamiento de memoria

5. Discusión

Puede apreciarse cómo el uso de tecnologías de almacenamiento de ontologías en memoria secundaria está estrechamente relacionado a las medidas de rendimiento del sistema, esfuerzo de los desarrolladores y sobre todo a las competencias de los mismos. Si se extrapolan estos resultados a sistemas de mayor escala y volúmenes de transacciones, se observa cómo el rendimiento de una implementación que trabaje en memoria principal no es medible debido a que tiene un techo (físico y económico) de almacenamiento mucho más bajo que una en memoria secundaria que pueda manejar ese volumen de carga. También, es visible cómo aumenta la brecha entre las habilidades de un desarrollador promedio, que se desempeña puramente en un paradigma orientado a objetos, y otro que necesariamente debe poseer conocimientos sobre ontologías.

En la tabla 3 puede observarse el nivel de competencias que deben tener los desarrolladores al involucrarse en un proyecto que utilice alguna de las herramientas ontológicas que hemos analizado.

[4] http://jena.apache.org/documentation/query/
[5] http://jena.apache.org/documentation/tdb/
[6] http://stardog.com/

Tabla 3 - Niveles de competencia para un proyecto basado en ontologías

Herramienta	*Competencia*	
	P.O.O.	*Ontologías*
Jena + OWLAPI + Pellet	Alto	Bajo
Jena + OWLAPI + Pellet + SPARQL	Alta	Medio
Stardog	Baja	Alta

En la tabla 4 pueden observarse los pros y contras de la utilización de cada una de las herramientas investigadas.

Tabla 4 - Pros y Contras de las Herramientas

Herramienta	*Pros - Contra*
Jena + OWLAPI + Pellet	✓ Poco conocimiento en ontologias ✓ Expresión estructurada del dominio del problema
	× Bajo rendimiento × Imposibilidad de utilizar almacenamiento secundario
Jena + OWLAPI+ Pellet + SPARQL	✓ Potencia del lenguaje de consultas SPARQL ✓ Alto rendimiento
	× Imposibilidad de utilizar almacenamiento secundario
Stardog	✓ Rendimiento medio ✓ Utilizacion de memoria secundaria ✓ Roles ✓ Seguridad ✓ Manejo de transacciones ✓ Protocolos de conexion ✓ Reglas nativas
	× Poco soporte de Built-Ins de Reglas SWRL

Como punto extra, a lo largo del trabajo se hizo evidente la necesidad de lenguajes de programación que acompañen naturalmente la semántica de un sistema de estas características.

Esta necesidad llevó a encontrar lenguajes de programación como Go[7] y Python[8] que manejan el tipado dinámico y la programación imperativa, dos características que permiten expresar el problema utilizando conceptos propios de una ontología en forma directa sin necesidad de adaptar la solución a un paradigma, por ejemplo, orientado a objetos.

6. Conclusión

La realización de este trabajo desencadenó un estudio profundo de las tecnologías de soporte al desarrollo de sistemas basados en ontologías. El primer motivo que dio lugar al estudio fue el bajo rendimiento observado a través de la utilización de las herramientas más difundidas en el mercado hasta ese momento, permitiendo encontrar nuevas tecnologías que ofrecen mejor desempeño y permiten superar las barreras físicas de almacenamiento principal. Otra de las causas que originó el análisis global de los sistemas de soporte, fue la percepción de un desaprovechamiento del potencial de una ontología en contraste con una base de datos tradicional.

En conclusión, se puede afirmar que las tecnologías que existen hoy en día, permiten alcanzar una implementación de sistemas más expresivos y con mayor semántica en los resultados tanto para humanos como para otros sistemas con los cuales deba interoperar. Pero, si bien se ha logrado vencer ciertas barreras físicas y teóricas, aún resta la difusión del campo de estudio y la capacitación de profesionales que lleven a cabo proyectos con este tipo de tecnologías.

[7] http://golang.org/
[8] https://www.python.org/

Referencias

[1] Calero, C. and Ruiz, F. and Piattini, M., eds.: *Ontologies for Software Engineering and Software Technology.* Springer. (2006)

[2] Zacharias, *V.: Rules as simple way to model knowledge: Closing the gap between promiseand reality*. En Proc. of 10th Int. Conf. on Enterprise Information Systems. 2008.

[3] Ruotsal T.: *Methods and Applications for Ontology-Based Recommender Systems*. PhD. Thesis. Aalto University School of Science and Technology. Finland. 2010.

[4] Vieira T.A.S.C., Casanova M.A., Ferrao, L.G.: *An Ontology-Driven Architecture for Flexible Workflow Execution*. En Proc. of WebMedia and LA-Web. IEEE Computer Soc. pp 70-77. 2004.

[5] Ayub, C.; Cian, A.; Reynares, E. and Caliusco, M. L. "Developing an Ontology-Based Team Recommender System using EDON Method: An experience Report" SADIO Electronic Journal Informatics and Operations Research. Vol 13 - No. 1 - June 2014. Special Issue dedicated to ASSE 2013 (Argentine Symposium on Software Engineering). Editores: Pace, A.; Colla, P. Disponible on line: http://www.sadio.org.ar/wp-content/uploads/2014/06/Revista_13.htm.

[6] Reynares, E.; Caliusco, M. L. and Galli, Ma. R. "Approaching the feasibility of SBVR as modeling language for ontology development: An exploratory experiment". Expert Systems with Applications 41 (2014) ISSN 0957-4174.

[7] Gaspoz, C.; Vertossi, V.; Reynares, E., Caliusco, Ma. L. "Applying EDON Methodology and SBVR2OWL Mappings for Building an Ontology-Aware Software". XIX Congreso Argentino de Ciencias de la Computación (CACIC), V Workshop Innovación en Sistemas de Software (WISS), del 21 al 25 de Octubre de 2013, Mar del Plata (Buenos Aires).

Framework Basado en Streams y Grafos para la Extracción de Modelos desde Código Fuente

Paul Mendoza del Carpio
Departamento de Ingeniería de Software
Universidad La Salle
Arequipa, Perú
pmendozadelcarpio@gmail.com

Abstract

Software visualization is an important issue in the domains of software maintenance, reconstruction of software architectures, and reverse engineering. In this context, the use of some supporting tools is relevant for medium and large software applications, where the quantity of source code to review and analyze is higher. This work presents a framework based on streams and graphs for the extraction of models from source code. Some important features of this framework includes parallelization of streams, first elements representation based on graphs, rules definition based on DSL, and generation of UML diagrams in XMI format. Java Web applications from GitHub repositories were used for evaluating the framework implementations, the results are shown.

1. Introducción

La proliferación de hardware *multicore* ha permitido extender la capacidad de procesamiento en aplicaciones hoy en día. Adicionalmente, el diseño e implementación de aplicaciones *multicore* es difícil, dificultad que radica en decisiones acerca de las estrategias de concurrencia y sincronización a emplear [10, 13]. El presente *framework* hace uso de los mecanismos de manejo de concurrencia incorporados en la versión 8 de Java, facilitando la tarea de implementación de aplicaciones *multicore* en el dominio de extracción de modelos.

Los *frameworks* permiten y promueven la enseñanza y uso de nuevos enfoques de desarrollo de software [23]. En este punto la propuesta promueve el uso de elementos de programación funcional como son las expresiones Lambda, presentes en la versión 8 de la plataforma Java, versión liberada recientemente en el presente año de elaboración de este trabajo. El presente *framework* busca al igual que sus similares en otros dominios [2, 6, 9, 12, 20, 25], permitir una simplificada creación de aplicaciones.

El uso de las expresiones Lambda, empleados en esta propuesta, proporciona importantes elementos para los desarrolladores usuarios del *framework* [1]: *streams*, paralelismo, iteradores internos, operaciones perezosas (*lazy operation*), interfaces funcionales, operaciones *MapReduce*.

Las aplicaciones de *streaming* son programas que procesan datos en forma continua tan pronto como los resultados están disponibles [24]. Muchas aplicaciones de *streaming* son empleadas en escenarios que requieren alto rendimiento [14]. El presente *framework* emplea un modelo basado en *streams* como una característica a señalar, la cual le permite abordar escenarios de aplicación con eficiencia [3]. Cada *stream* es una secuencia de datos, y cada operador (*filter*, *map*, *reduce*, *peek*) consume elementos de *streams* entrantes produciendo otros hacia *streams* de salida. Puesto que los operadores pueden ser ejecutados en forma concurrente, ellos exponen paralelismo en forma inherente [14].

El modelo de *streaming* empleado en el *framework* soporta la ejecución concurrente de un productor A con un consumidor B (*pipeline* paralelo) [14]. Es elección del desarrollador que emplea el *framework*, aplicar una ejecución secuencial o paralela, según su criterio o problema a resolver.

Por otro lado, el *framework* emplea nodos y relaciones de grafos, como representación inicial, durante el proceso de extracción de modelos. Luego, los elementos de estos grafos pueden ser manipulados y transformados en nuevas representaciones según el problema a resolver. Cabe señalar que los grafos comúnmente son fáciles de entender y ampliamente empleados en diversos dominios.

2. Motivación

El entender y el modificar sistemas de software pueden convertirse en tareas cada vez más difíciles

conforme el software incrementa su tamaño y a su vez se hace más complejo. Lo anterior, incrementa la importancia de la creación y uso de herramientas que brinden soporte a estas tareas. Un aspecto clave de este soporte es la visualización de software, debido a que ella proporciona una vista más substancial del software [18] y puede reducir significativamente el esfuerzo empleado en estas tareas [22].

La visualización de software tiene un alto grado de valor para los interesados e involucrados en los dominios de mantenimiento de software, ingeniería inversa, y reconstrucción de arquitecturas de software [19]. El mantenimiento de software constituye una de las fases más costosas en el ciclo de vida de software, donde la mayor parte del tiempo es empleado en entender el sistema a mantener [18]. Luego, la visualización también juega un rol importante y es considerada un problema central en el dominio de la ingeniería reversa [19], incluyendo aplicaciones contemporáneas [15].

La comprobación de modelos de software constituye una forma prometedora para demostrar el cumplimiento de la especificación de una aplicación [11]. Visualizar la arquitectura de software es uno de los tópicos más importantes en el campo de la visualización de software. Ello comprende la visualización de la jerarquía y las relaciones entre los componentes de software. La capacidad de tener una visión general del sistema es de considerable importancia para enfocar el interés en partes específicas del software sin perder la visión completa del contexto [18]. Conforme avanzan el diseño e implementación de un sistema basado en una arquitectura de software, es posible encontrar discrepancias entre la arquitectura objetivo y la arquitectura actual en desarrollo, siendo ello aún más probable en casos donde la arquitectura no está completamente especificada, documentada o comunicada. A partir de ello, la extracción de vistas para la reconstrucción de arquitecturas de software desde artefactos de implementación (código fuente) es crítica para la verificación de la arquitectura implementada [5, 8].

3. Extracción de Modelos

El presente *framework* emplea grafos para la representación de elementos y relaciones, y a su vez emplea *streams* para la identificación y procesamiento de los mismos. Sin embargo, en su forma fundamental, el mismo extrae información de dos fuentes principales:

- Código fuente: archivos de código fuente y la estructura de paquetes subyacente.
- Documento XML: archivo XML, elaborado de acuerdo a un esquema propio (*XML Schema*) propuesto en este trabajo, donde el desarrollador usuario del *framework* puede establecer reglas que permitan identificar elementos de interés a formar parte del modelo final.

Para el tratamiento del código fuente y el documento XML, árboles AST (*Abstract Syntax Tree*) y un DSL (*Domain Specific Language*) son empleados respectivamente, seguido se describe cada uno de ellos.

3.1. AST

Se tiene en consideración que el desarrollo de un *parser* dedicado es costoso e induce al uso considerable de *hardcoding* en las travesías del árbol sintáctico [7]. Frente a ello la propuesta realiza la obtención automática de un AST desde el código fuente, para lo cual se hace uso del API empleado por IDEs basados en la plataforma Eclipse: Eclipse JDT (*Java Development Tools*) [4]. En el recorrido del árbol AST, se emplean visitadores (*visitors*, en el catálogo de patrones de diseño GoF) sobre los diferentes elementos presentes en el código fuente: tipos, paquetes, variables, métodos, marcadores, anotaciones, entre otros.

3.2. DSL

El *framework* cuenta con un esquema XML donde se define un DSL externo. Ello facilita la definición de reglas a cumplir para identificar elementos relevantes a considerar en el modelo a extraer. Si bien la extracción del modelo podría representar cada elemento encontrado en su recorrido, la presente característica puede ser de utilidad en caso se requiera mayor análisis de los elementos a considerar. Por ejemplo, en un escenario de reconstrucción arquitectónica, por lo general, sólo se requieren extraer elementos arquitectónicamente significativos, omitiendo en ocasiones elementos de detalle.

La estructura del esquema XML, véase figura 1, incluye fundamentalmente los siguientes elementos:

1. Recursos (*resources*): elemento XML que define valores para el contenido, o propiedades de los archivos de código fuente empleados como entrada. Emplea expresiones regulares para especificar nombres de archivos, valores de rutas XPath (para archivos de configuración XML), anotaciones de Java, valores de atributos en anotaciones, entre otros.
2. Filtros (*filters*): definición de las reglas a cumplir para identificar elementos a ser considerados en el modelo extraído. Internamente este elemento es convertido en un operador *filter*. Permite presentar una lógica compleja, pues incluye elementos XML que representan operadores lógicos de AND (*featureAnd*) y OR (*featureOr*), los cuales pueden ser anidados uno u otro en forma sucesiva.

3. Funciones (*function*): nombre calificado completo de implementaciones de funciones, que pueden ser aplicadas en una operación de mapeo de objetos *Path* (ruta de los archivos a analizar) hacia nodos que representan elementos a considerar en el modelo objetivo.

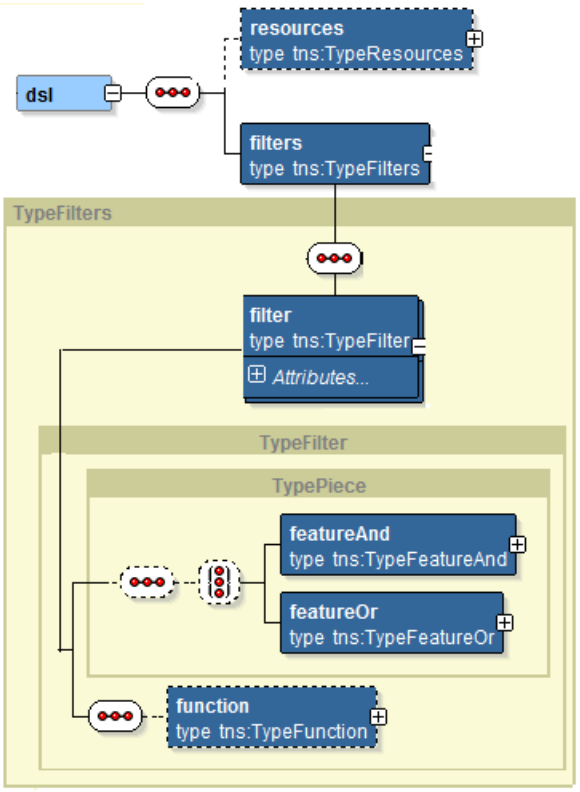


Figura 1. Esquema del DSL.

La figura 2 muestra la estructura básica de un documento XML, donde se indicarían algunas características para identificar elementos del modelo a extraer. Nótese el uso de reglas relacionadas a anotaciones del código fuente (*withAnnotation*) y a valores de rutas XPath en archivos de configuración (*withXPath*), ambos mecanismos son frecuentemente empleados en la declaración de componentes del lado del servidor en aplicaciones Java EE (*Enterprise Edition*).

4. Modelos Basados en Grafos

Los grafos son considerados entre los tipos de visualización más empleados en áreas como mantenimiento de software, ingeniería inversa, y reingeniería [17]. Los grafos cuentan con las características necesarias para representar relaciones entre componentes, considerando los elementos de software como nodos, y las relaciones entre éstos como aristas [17]. Es así que los grafos son empleados en diversos campos para solucionar problemas.

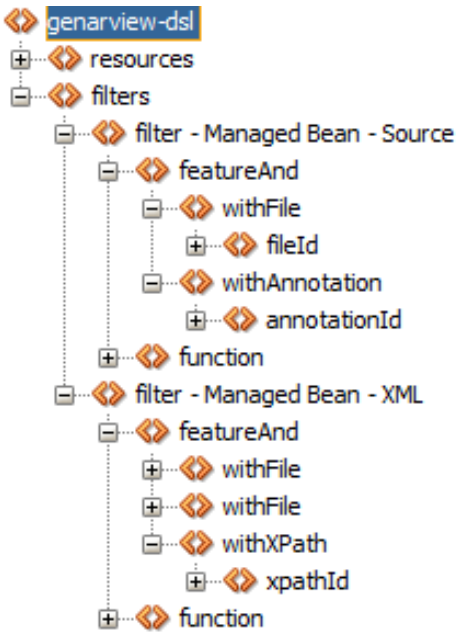


Figura 2. Estructura de documento XML.

Cada elemento (e.g. componente, clase, interfaz), contenedor (e.g. directorio, paquete) y/o relación (e.g. dependencia, asociación, realización) es representado como nodos y relaciones que son almacenados directamente en una base de datos. Para ello se emplea la base de datos basada en grafos Neo4j, la cual es considerada un estándar de la industria en cuanto a almacenamiento de grafos [16], resaltando también por su rendimiento [21]. La estructura de los grafos en su forma fundamental, es mostrada en la figura 3 mediante un modelo conceptual de los nodos y sus relaciones. Se pueden distinguir diferentes tipos de nodos de acuerdo a las etiquetas (*Label*) que presente cada nodo. Asimismo cada nodo puede contener diversos datos, los cuales son almacenados como propiedades (*Property*) de cada uno.

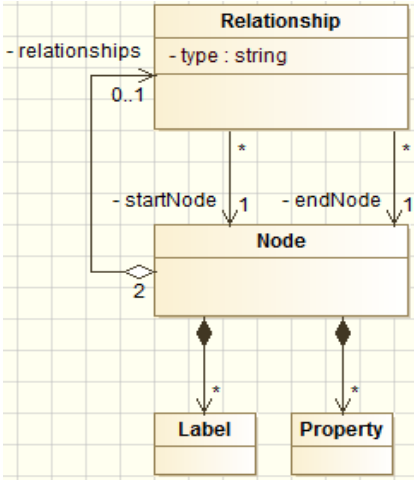


Figura 3. Modelo conceptual de nodos.

Diversos aplicativos pueden contener una considerable cantidad de elementos de código, los cuales al contar con múltiples relaciones entre ellos, pueden presentar un grafo complejo y abrumador como se muestra en la figura 4. Grafos que representan contenedores (e.g. paquetes) pueden mostrar una presentación más simplificada del aplicativo como se muestra en la figura 5. Los aplicativos que fueron diseñados atendiendo tener una complejidad manejable, pueden tener una representación modular y entendible como se muestra en la figura 6. Según lo mencionado anteriormente, se puede apreciar que la representación basada en grafos nos puede aproximar a una primera opinión de la complejidad, modularidad, y relacionamiento, presentes en un sistema.



Figura 4. Grafo de representación de clases.

La disposición de un lenguaje de consulta sobre los grafos permite a la propuesta evitar problemas de confusión y sobrexposición en la visualización, problemas que se pueden presentar en grafos con una considerable cantidad de nodos. Adicionalmente, una ventaja importante que proporciona el uso de Neo4j, radica en su lenguaje de consulta Cypher, el cual permite encontrar patrones inmersos en los grafos en forma fácil [21]. Una implementación en desarrollo puede incorporar una consulta a medida durante el mapeo de nodos hacia relaciones, aquí se puede establecer una función que será pasada a un operador de mapeo en el *pipeline*, y que luego será ejecutada sobre cada nodo a fin de encontrar relaciones a través de la consulta establecida.

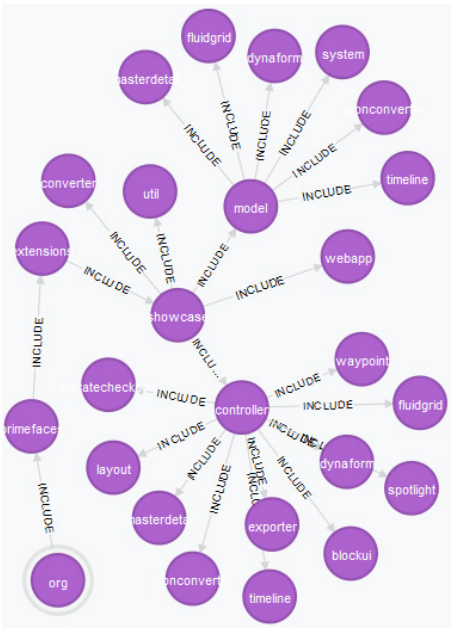


Figura 5. Grafo de representación de paquetes.

5. Construcción del Modelo

Al recorrer las fuentes de información empleadas por el *framework* (véase la sección Extracción de Modelos), se van creando nodos y relaciones de un grafo en Neo4j. Tales nodos y relaciones son creados aplicando operadores de filtro (i.e. *filter*) y mapeo (i.e. *map*) sobre los datos pertinentes en el *stream*. Véase la figura 7, la cual muestra una herramienta de Neo4j para la visualización de nodos, relaciones, y algunos datos de un nodo seleccionado.

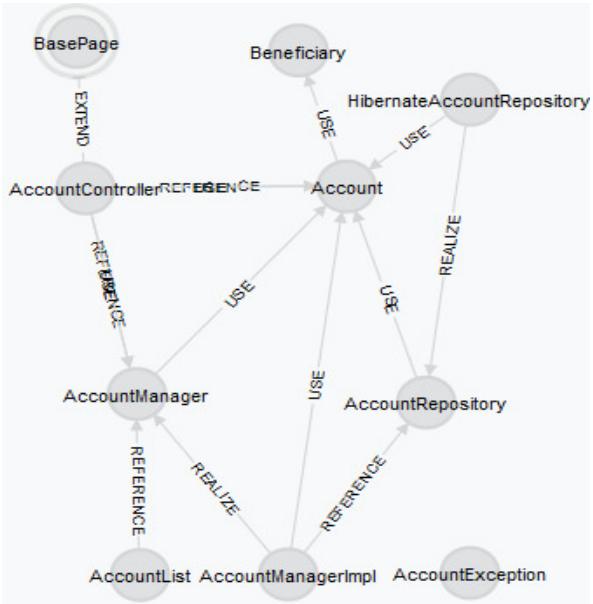


Figura 6. Partición de un grafo.

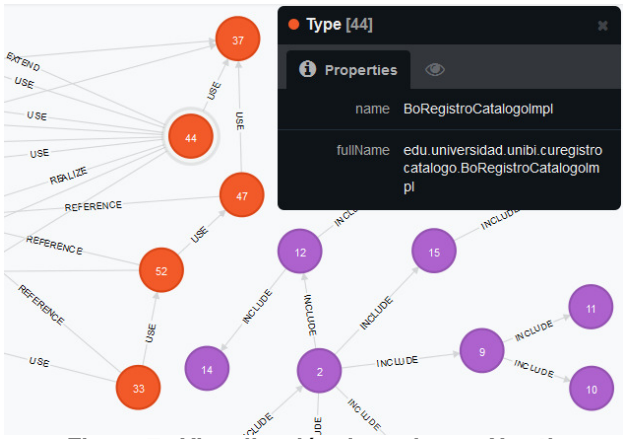


Figura 7. Visualización de grafos en Neo4j.

A partir de tal grafo es posible aplicar operadores para la obtención de otras representaciones. Tales operadores son ejecutados sobre los *streams* de datos, siendo posible emplear paralelismo sobre el proceso. Por ejemplo, el *framework* cuenta con funciones para la creación de elementos de UML a partir de nodos: paquetes, clases, asociación, dependencia, generalización, realización, entre otros. La figura 8 muestra un diagrama de clases, cuyos elementos fueron generados en formato XMI empleando el *framework* en mención.

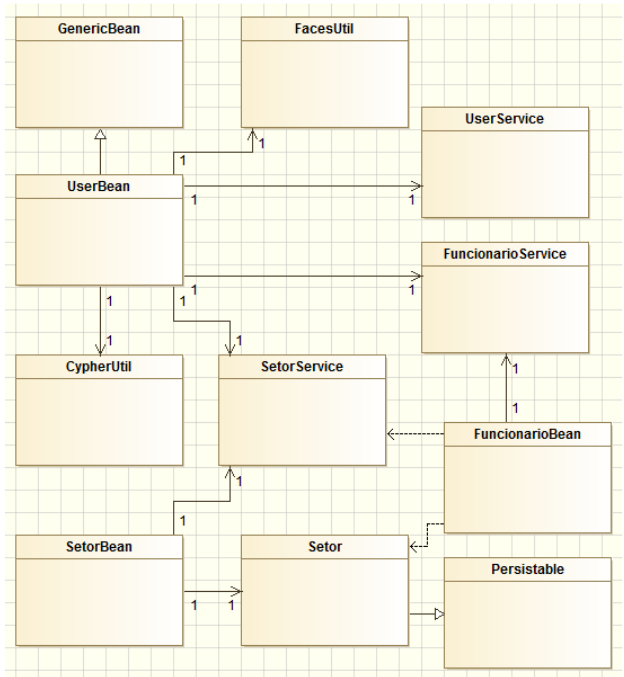


Figura 8. Diagrama UML extraído.

6. Vista lógica

La vista lógica es resumida en un diagrama de paquetes que es presentado en la figura 9.

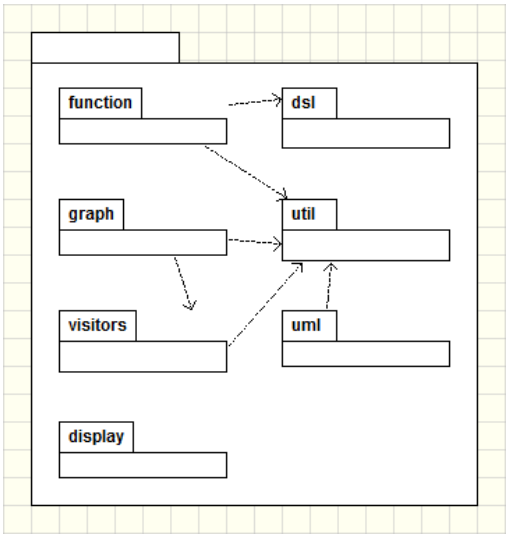


Figura 9. Diagrama de paquetes.

Seguido se tiene una descripción de cada paquete:

- *function*: funciones y predicados para la creación y evaluación de elementos del modelo extraído.
- *graph*: funciones para la creación de nodos y relaciones.
- *util*: utilitarios para el manejo de archivos, grafos, *streams*, y código fuente.
- *visitors*: elementos *visitor* basados en árboles AST (*Abstract Syntax Tree*) para la identificación de elementos y relaciones en los archivos de código fuente.
- *uml*: funciones para la creación de elementos clasificadores de UML, incluye operadores sobre los mismos.
- *display*: funciones y operadores para la visualización de elementos del modelo.
- *dsl*: clases para la lectura y obtención de información contenida en los archivos XML de reglas (véase la sección DSL).

7. Vista de procesos

La vista de procesos es descrita y presentada a continuación mediante tres diagramas que representan tres fases para la extracción de modelos:

I. Generación de contenedores: generación de contenedores para elementos y modelos. Se emplean representaciones de paquetes, considerándolos como contenedores según establece UML.

II. Generación de elementos y relaciones: generación de elementos del modelo y relaciones que se puedan dar entre ellos. Ambos son representados en nodos y relaciones de grafos.

III. Generación del modelo: generación propiamente dicha del modelo, luego de haber identificado

elementos y relaciones. Considera la forma de salida para el modelo.

7.1. Generación de contenedores

Fase que inicia con un *stream* de los *paths* (i.e. rutas) de archivos a analizar. Luego se realiza el flujo de datos a través del *pipeline*, a lo largo del mismo, los tipos de datos en el *stream* pueden cambiar al aplicar un operador. Se tienen los siguientes pasos que son mostrados en la figura 10:

1. Filtrar directorios: filtro de archivos que correspondan a directorios contenedores de archivos.
2. Mapear directorios: mapeo que recibe un *stream* de *paths* y proporciona un *stream* de nodos de grafos, los cuales representan los directorios (en adelante, se denominarán nodos directorio).
3. Mapear directorios incluidos: recibe un *stream* con los nodos directorio y crea relaciones de "incluye" entre los nodos.

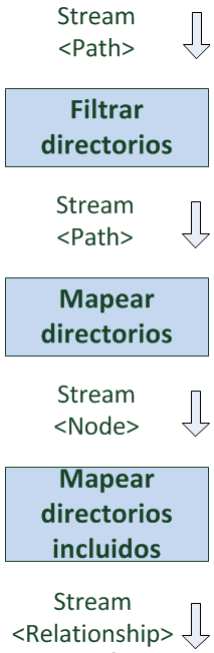


Figura 10. Generación de contenedores.

7.2. Generación de elementos y relaciones

Fase que inicia con un *stream* de los *paths* de archivos a analizar, culmina estableciendo las relaciones entre nodos que representan elementos del modelo. Se tienen los siguientes pasos que son mostrados en la figura 11:

1. Filtrar archivos: filtro de archivos planos.
2. Mapear elementos: mapeo que recibe un *stream* de *paths* y proporciona un *stream* con los nombres calificados (completos) de elementos del modelo a extraer.
3. Mapear nodos elemento: recibe un *stream* con los nombres de elementos y proporciona un *stream* con los nodos que representan elementos del modelo a extraer (en adelante, se denominarán nodos elemento), entre otros se pueden tener clases, componentes, objetos.
4. Mapear elementos contenidos: recibe un *stream* con los nodos elemento y crea relaciones de "contiene" entre los nodos directorio y los nodos elemento del *pipeline* actual.
5. Mapear relaciones: recibe el *stream* de *paths* de archivos ya filtrado, y crea relaciones entre los nodos elemento. Entre otros se pueden considerar relaciones con denominación "depende" (dependencia), "referencia" (asociación), "extiende" (herencia), "realiza" (realización).

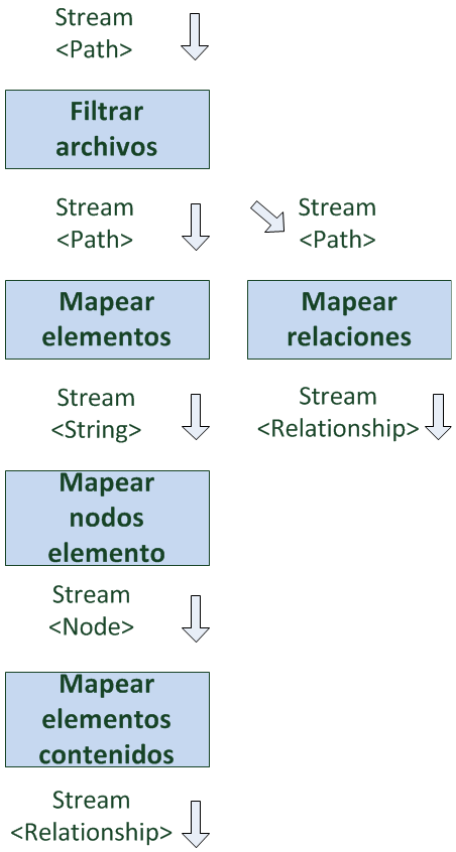


Figura 11. Generación de elementos y relaciones.

Cabe señalar que el uso del DSL ayuda al desarrollador en esta fase para automatizar la identificación de nodos elemento y relacionarlos según una plantilla de características (*features*) establecida en el archivo XML, ello es de utilidad cuando se requiere considerar únicamente elementos relevantes.

7.3. Generación del modelo

Fase final que propiamente genera el modelo objeto a extraer. Inicia con un *stream* de los nodos directorio, y utilizando la información ya generada en las anteriores fases, va generando elementos que forman parte de la salida esperada. En el *framework* se procura el uso de la notación UML, soportando también el uso de otras notaciones o representaciones. Para salidas que emplean UML, los elementos generados corresponden al componente UML2 del proyecto Eclipse MDT (*Model Development Tools*), proyecto abocado al desarrollo de modelos basados en meta modelos estándares de la industria. Se tienen los siguientes pasos que son mostrados en la figura 12:

1. Mapear *package*: mapeo que recibe un *stream* de los nodos directorio, nodos que contienen los archivos procesados en las fases anteriores, y proporciona un *stream* de objetos *Package* del modelo a extraer.
2. Mapear *model element*: recibe un *stream* con los objetos *Package* y proporciona un *stream* con los elementos del modelo objetivo. Entre éstos se puede considerar diferentes clasificadores de UML. Cabe señalar que cada instancia de Package contiene el nombre completo del paquete, el cual es empleado para consultar nodos directorio, y a partir de ellos se puede obtener los nodos elemento, éstos últimos serán empleados para la creación de cada instancia de clasificador.
3. Ciclo: recibe el *stream* de clasificadores y ejecuta una o varias operaciones sobre ellos (se aplica el operador *peek*). Este paso está abocado a establecer relaciones entre los clasificadores o adicionar información sobre ellos. Produce como salida el mismo *stream* de clasificadores original.
4. Reducir *model element*: recibe un *stream* con los clasificadores y proporciona un único elemento representativo, por lo general se tratará del elemento padre contenedor de los demás (el objeto modelo a serializar o visualizar). Para salidas que emplean UML, a partir de este último objeto es posible generar un archivo XMI (*XML Metadata Interchange*), especificación estándar empleada para compartir modelos UML con diversas herramientas de modelado.

Cabe señalar que la salida a generar no sólo puede considerar objetos que representan un modelo en MDT UML2 sino también se puede utilizar otras formas de salida. Es así que se pueden desarrollar implementaciones con salidas de visualización empleando el *toolkit* Draw2d de Eclipse, el cual cuenta con diversas implementaciones de figuras, conexiones, decoraciones, y *layout*.

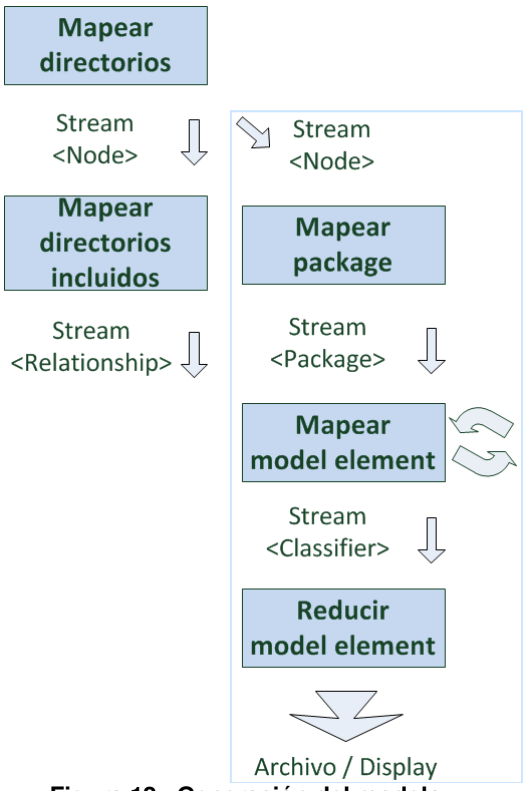


Figura 12. Generación del modelo.

8. Hot spots

Los puntos flexibles del *framework* (i.e. *hot spots*) se encuentran en cada operación de filtro, mapeo o reducción que se ha mostrado en la vista de procesos. El uso de expresiones Lambda en el *framework* permite establecer implementaciones propias o extendidas en los operadores del pipeline (e.g. *filter*, *map*, *reduce*, *peek*). Dependiendo del operador, estas implementaciones pueden ser elementos nuevos presentes en Java 8: predicados (*Predicate*) para *filter*, funciones (*Function*) para *map*, operadores binarios (*BinaryOperator*) para *reduce*, consumidores (*Consumer*) para *peek*.

9. Demostración

A continuación, fragmentos de código fuente con fines usuales en el desarrollo de una implementación son mostrados como demostración del uso del *framework*.

La figura 13 muestra la generación de listados de directorios y archivos a partir de una ruta determinada, Se hace uso de un utilitario para manejo de archivos (*FileUtil*), del operador de filtrado (*filter*), así como de *streams* paralelos (*parallelStream*) a diferencia del común *stream* secuencial.

La figura 14 muestra la creación de nodos y relaciones entre ellos, empleando operadores *map* presentes en el *framework* para la representación de directorios y sus

relaciones de inclusión. Nótese el uso de las funciones *FPackage* y *FRelationshipContainPackage*.

```
List<Path> lstPathSource
        = FileUtil.generateListPath(pathSource);
// Filtrado de directorios y archivos
List<Path> lstSourceDir = lstPathSource.parallelStream()
        .filter(p -> (Files.isDirectory(p)))
        .collect(Collectors.toList());
List<Path> lstSourceFile = lstPathSource.parallelStream()
        .filter(p -> (!Files.isDirectory(p)))
        .collect(Collectors.toList());
```

Figura 13. Código para la generación de listados.

```
// Creación de estructura de paquetes en nodos
FPackage fNodePackage = new FPackage(pathSource);
List<Node> lstNode = lstSourceDir.parallelStream()
        .map(fNodePackage)
        .collect(StreamUtil.<Node>collectorList());
FRelationshipContainPackage fRelationshipPackage
        = new FRelationshipContainPackage();
lstNode.stream().map(fRelationshipPackage)
        .collect(Collectors.toList());
```

Figura 14. Código para la creación de nodos.

La figura 15 muestra la ejecución de operaciones para establecer relaciones entre clasificadores, éstos últimos son proporcionados mediante *streams*. Aquí también se muestra el uso de *streams* paralelos.

```
// Creación de relaciones entre clasificadores
lstClass.parallelStream()
    .peek(new CGeneralization(mapClass))
    .collect(Collectors.toList());
lstClass.parallelStream()
    .peek(new CRealization(mapClass))
    .collect(Collectors.toList());
lstClass.parallelStream()
    .peek(new CAssociation(mapClass))
    .collect(Collectors.toList());
lstClass.parallelStream()
    .peek(new CDependency(mapClass))
    .collect(Collectors.toList());
```

Figura 15. Código para establecer relaciones.

10. Evaluación y Resultados

A fin de evaluar el *framework* y su desempeño en diversos aplicativos, se ha considerado un conjunto de 30 aplicativos Java Web, los cuales fueron obtenidos desde repositorios de GitHub. La tabla 1 muestra las principales características de los aplicativos en consideración: número de directorios (#DIR), número de archivos (#ARC), y número de líneas de código (LOC).

Se realizaron sucesivas pruebas sobre cada aplicativo, obteniendo tiempos promedio en milisegundos empleados para la extracción de un modelo. Tales tiempos incluyen la generación de archivos XMI, cuyos elementos y relaciones fueron visualizados en una herramienta de modelado UML. Las pruebas fueron realizadas considerando implementaciones con ejecución secuencial así como paralela. La tabla 2 muestra los resultados obtenidos: tiempo promedio de ejecución secuencial (SEC), tiempo promedio de ejecución paralela (PAR), y la diferencia del tiempo secuencial menos el tiempo paralelo (DIF). Los tiempos son expresados en unidades de milisegundos.

Tabla 1. Listado de aplicativos.

	APLICATIVO	#DIR	#ARC	LOC
1	03-security-solution	5	24	1789
2	acm-showcase	7	15	1270
3	architect	19	100	7818
4	auction	9	28	1533
5	bestellprozess-java-ee6	9	17	847
6	canchita	7	32	3268
7	cassandra	9	14	782
8	idp	9	20	1850
9	invoice-en	7	15	856
10	IPFIXConfig	1	89	27563
11	javaee6-sample	4	14	2176
12	jeedemo	4	12	1251
13	jsf-booking	7	11	929
14	jsf_eficaz	7	22	1393
15	kbrowser	8	23	5630
16	movieplex7	10	24	2608
17	musico-war	9	12	845
18	order-confirmation-rules	7	14	857
19	plato	24	86	15415
20	primefaces-spring-security-hibernate	5	25	1916
21	project-ui	10	15	1072
22	roo-jsf	5	11	273
23	scrumboard	7	15	1832
24	showcase	25	102	12044
25	socialize-workshop	12	32	1270
26	tcc	12	35	2535
27	ticket2rock	22	94	6838
28	TopicIndex	28	188	69946
29	upcdew-deportivoapp	11	44	5122
30	webservice	16	38	3038

11. Conclusiones y trabajo futuro

El *framework* ha mostrado una correcta aplicación en los diferentes aplicativos mencionados en la sección de evaluación. El uso de *streams* permite a las implementaciones desarrolladas con el *framework*, contar con capacidades de ejecución en forma paralela y/o secuencial. Asimismo, la representación inicial basada en grafos permite representar diversos tipos de nodos para cada posible elemento a considerar en el modelo extraído, como también etiquetar diferentes tipos de relaciones entre ellos.

El uso de *streams* paralelos implica procesamiento para dividir el trabajo entre diferentes hilos y juntar los resultados obtenidos por cada uno de ellos. Sin embargo, para aplicativos con mayores cantidades de archivos y

LOC, el uso de *streams* paralelos es una alternativa de mejora frente al procesamiento secuencial.

Tabla 2. Tiempos obtenidos por aplicativo.

	APLICATIVO	SEC	PAR	DIF
1	03-security-solution	302	192	110
2	acm-showcase	151.22	114.44	36.78
3	architect	1461.8	576.44	885.34
4	auction	240.11	182.33	57.78
5	bestellprozess-java-ee6	148.33	167.67	-19.34
6	canchita	238	167.33	70.67
7	cassandra	142.56	122.44	20.12
8	idp	462.33	131.56	330.77
9	invoice-en	100.78	98.78	2
10	IPFIXConfig	1795.9	735.89	1060
11	javaee6-sample	146.56	149.33	-2.77
12	jeedemo	137.67	238.11	-100.4
13	jsf-booking	156.33	103.11	53.22
14	jsf_eficaz	540.22	110.22	430
15	kbrowser	316.22	192.44	123.78
16	movieplex7	211.56	209.22	2.34
17	musico-war	111.56	103.78	7.78
18	order-confirmation-rules	139.56	99.56	40
19	plato	1443.3	752.11	691.22
20	primefaces-spring-security-hibernate	763.44	191	572.44
21	project-ui	267.89	116.89	151
22	roo-jsf	280.33	269.56	10.77
23	scrumboard	222.89	129.67	93.22
24	showcase	982.89	507.89	475
25	socialize-workshop	658.67	204.22	454.45
26	tcc	769.11	334.44	434.67
27	ticket2rock	1822.4	559.22	1263.2
28	TopicIndex	91968	31833	60135
29	upcdew-deportivoapp	2871.2	1345.9	1525.3
30	webservice	1464.1	701.78	762.33

Se tiene como trabajo futuro el emplear el presente *framework* como componente en una propuesta de modelo para la generación de vistas arquitectónicas de software, tema de la tesis doctoral del autor, la cual se encuentra actualmente en desarrollo.

12. Referencias

[1] Alex Gyori, Lyle Franklin, Danny Dig, and Jan Lahoda. 2013. Crossing the gap from imperative to functional programming through refactoring. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2013). ACM, New York, NY, USA, 543-553. DOI=10.1145/2491411.2491461 http://doi.acm.org/10.1145/2491411.2491461

[2] Arun Kumar, Sheetal K. Agarwal, and Priyanka Manwani. 2010. The spoken web application framework: user generated content and service creation through low-end mobiles. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)* (W4A '10). ACM, New York, NY, USA, , Article 2 , 10 pages. DOI=10.1145/1805986.1805990 http://doi.acm.org/10.1145/1805986.1805990

[3] Buğra Gedik. 2014. Partitioning functions for stateful data parallelism in stream processing. The VLDB Journal 23, 4 (August 2014), 517-539. DOI=10.1007/s00778-013-0335-9 http://dx.doi.org/10.1007/s00778-013-0335-9

[4] http://www.eclipse.org/jdt/

[5] http://www.sei.cmu.edu/architecture/research/previousresearch/reconstruction.cfm

[6] J. Tindle, M. Gray, R. L. Warrender, K. Ginty, and P. Dawson. 2012. Application Framework for Computational Chemistry AFCC Applied to New Drug Discovery. *Int. J. Grid High Perform. Comput.* 4, 2 (April 2012), 46-62. DOI=10.4018/jghpc.2012040104 http://dx.doi.org/10.4018/jghpc.2012040104

[7] Javier Luis Cánovas Izquierdo and Jesús García Molina. 2014. Extracting models from source code in software modernization. *Softw. Syst. Model.* 13, 2 (May 2014), 713-734. DOI=10.1007/s10270-012-0270-z http://dx.doi.org/10.1007/s10270-012-0270-z

[8] Josip Maras, Maja \&\#138;tula, and Ivica Crnkovic. 2009. phpModeler - A Web Model Extractor. In *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering* (ASE '09). IEEE Computer Society, Washington, DC, USA, 660-661. DOI=10.1109/ASE.2009.40 http://dx.doi.org/10.1109/ASE.2009.40

[9] Koji Kagawa. 2010. WappenLite: a web application framework for lightweight programming environments. In *Proceedings of the 9th international conference on Information technology based higher education and training* (ITHET'10). IEEE Press, Piscataway, NJ, USA, 21-26.

[10] Lyle Franklin, Alex Gyori, Jan Lahoda, and Danny Dig. 2013. LAMBDAFICATOR: from imperative to functional programming through automated refactoring. In *Proceedings of the 2013 International Conference on Software Engineering* (ICSE '13). IEEE Press, Piscataway, NJ, USA, 1287-1290.

[11] M. M. Gallardo, C. Joubert, P. Merino, and D. Sanán. 2012. A model-extraction approach to verifying concurrent C programs with CADP. *Sci. Comput. Program.* 77, 3 (March 2012), 375-392. DOI=10.1016/j.scico.2011.10.003 http://dx.doi.org/10.1016/j.scico.2011.10.003

[12] Marcelo Worsley, Michael Johnston, and Paulo Blikstein. 2011. OpenGesture: a low-cost authoring framework for gesture and speech based application development and learning analytics. In Proceedings of the 10th International Conference on Interaction Design and Children (IDC '11). ACM, New York, NY, USA, 254-256. DOI=10.1145/1999030.1999075 http://doi.acm.org/10.1145/1999030.1999075

[13] Marcelo Sousa and Alper Sen. 2013. LLVMVF: A Generic Approach for Verification of Multicore Software. J. Electron.

Test. 29, 5 (October 2013), 635-646. DOI=10.1007/s10836-013-5405-9 http://dx.doi.org/10.1007/s10836-013-5405-9

[14] Martin Hirzel, Robert Soulé, Scott Schneider, Buğra Gedik, and Robert Grimm. 2014. A catalog of stream processing optimizations. ACM Comput. Surv. 46, 4, Article 46 (March 2014), 34 pages. DOI=10.1145/2528412 http://doi.acm.org/10.1145/2528412

[15] Mona Erfani Joorabchi and Ali Mesbah. 2012. Reverse Engineering iOS Mobile Applications. In *Proceedings of the 2012 19th Working Conference on Reverse Engineering* (WCRE '12). IEEE Computer Society, Washington, DC, USA, 177-186. DOI=10.1109/WCRE.2012.27 http://dx.doi.org/10.1109/WCRE.2012.27

[16] Peter Macko, Daniel Margo, and Margo Seltzer. 2013. Performance introspection of graph databases. In *Proceedings of the 6th International Systems and Storage Conference* (SYSTOR '13). ACM, New York, NY, USA, , Article 18 , 10 pages. DOI=10.1145/2485732.2485750 http://doi.acm.org/10.1145/2485732.2485750

[17] Petra Mutzel and Peter Eades. 2001. Graphs in Software Visualization - Introduction. In Revised Lectures on Software Visualization, International Seminar, Stephan Diehl (Ed.). Springer-Verlag, London, UK, UK, 285-294.

[18] Pierre Caserta and Olivier Zendra. 2011. Visualization of the Static Aspects of Software: A Survey. IEEE Transactions on Visualization and Computer Graphics 17, 7 (July 2011), 913-933. DOI=10.1109/TVCG.2010.110 http://dx.doi.org/10.1109/TVCG.2010.110

[19] Rainer Koschke. 2003. Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey. Journal of Software Maintenance 15, 2 (March 2003), 87-109. DOI=10.1002/smr.270 http://dx.doi.org/10.1002/smr.270

[20] Rob King. 2010. Pixaxe: a declarative, client-focused web application framework. In *Proceedings of the 2010 USENIX conference on Web application development* (WebApps'10). USENIX Association, Berkeley, CA, USA, 10-10.

[21] Salim Jouili and Valentin Vansteenberghe. 2013. An Empirical Comparison of Graph Databases. In Proceedings of the 2013 International Conference on Social Computing (SOCIALCOM '13). IEEE Computer Society, Washington, DC, USA, 708-715. DOI=10.1109/SocialCom.2013.106 http://dx.doi.org/10.1109/SocialCom.2013.106

[22] Sarita Bassil and Rudolf K. Keller. 2001. Software Visualization Tools: Survey and Analysis. In *Proceedings of the 9th International Workshop on Program Comprehension* (IWPC '01). IEEE Computer Society, Washington, DC, USA, 7-.

[23] Steffen Zschaler, Birgit Demuth, and Lothar Schmitz. 2014. Salespoint: A Java framework for teaching object-oriented software development. *Sci. Comput. Program.* 79 (January 2014), 189-203. DOI=10.1016/j.scico.2012.04.005 http://dx.doi.org/10.1016/j.scico.2012.04.005

[24] Thomas Heinze, Valerio Pappalardo, Zbigniew Jerzak, and Christof Fetzer. 2014. Auto-scaling techniques for elastic data stream processing. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (DEBS '14). ACM, New York, NY, USA, 318-321. DOI=10.1145/2611286.2611314 http://doi.acm.org/10.1145/2611286.2611314

[25] Wenjun Wu, Thomas D. Uram, Michael Wilde, Mark Hereld, and Michael E. Papka. 2010. A Web 2.0-Based Scientific Application Framework. In Proceedings of the 2010 IEEE International Conference on Web Services (ICWS '10). IEEE Computer Society, Washington, DC, USA, 642-643. DOI=10.1109/ICWS.2010.107 http://dx.doi.org/10.1109/ICWS.2010.107

JInsight: Herramienta de Análisis Estático para la Comprensión de Programas

Arnaldo Ceballos, Hernán Bernardis, Mario Berón, Daniel Riesco
Departamento de Informática - Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
arceballos,hbernardis,mberon,driesco{@unsl.edu.ar}

Abstract

Los sistemas a menudo están compuestos de módulos formados por miles de líneas de código. En general, no suele ser fácil encontrar qué parte del código genera cierta característica dada del sistema. Además, en muchas ocasiones debido a la desactualización, falta de documentación, desconocimiento de la arquitectura del sistema, entre otras tantas posibilidades, está dificultad no es reducida. Una manera de lograr comprender un sistema es aplicar una disciplina dentro de la Ingeniería del Software denominada Ingeniería Reversa, la cual permite identificar los componentes del mismo. Herramientas que lleven a cabo este proceso son de gran interés, debido a que permiten obtener una visión del sistema de manera rápida y precisa. En este artículo se presenta JInsight, una herramienta que ayuda en la tarea de comprensión de programas aplicando conceptos de Ingeniería Reversa. JInsight realiza extracción de información estática y visualiza la información obtenida de diferentes maneras para ayudar al usuario en el proceso de comprensión del sistema. Esta herramienta permite analizar sistemas escritos en lenguaje Java mediante la extracción de información estática y la presentación de la misma mediante visualizaciones atractivas. El usuario tiene la posibilidad de interactuar con la interfaz y las visualizaciones al ofrecer, entre otras características, la posibilidad de navegar entre las diferentes vistas. Esta peculiaridad permite inspeccionar el sistema en distintos niveles de abstracción del sistema.

1. Introducción

La Ingeniería de Software se define como "La aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras y la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos" [23].

El trabajo que se asocia a la disciplina previamente mencionada se puede dividir en las fases mostradas en la figura 1, con independencia del área de aplicación, tamaño o complejidad de proyecto.

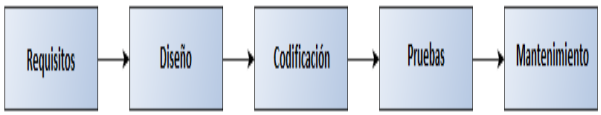


Figura 1. Fases del proceso de desarrollo de un sistema.

La figura previamente mostrada brinda un enfoque lineal para la Ingeniería de Software. Una breve descripción de cada una de las fases se describe a continuación:

- Requisitos. Reunión de requisitos para la construcción de software. El ingeniero de software debe comprender las funcionalidades requeridas.
- Diseño. El proceso de diseño traduce los requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación.
- Codificación. El diseño se debe traducir en una forma ejecutable por la máquina.
- Pruebas. Una vez que se ha generado el código comienza la etapa de pruebas. Dicha etapa tiene como objetivo detectar errores y asegurar que las entradas definidas producen los resultados esperados.

- Mantenimiento. El software indudablemente sufrirá cambios después de ser entregado al cliente. Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo, o porque el cliente requiere mejoras funcionales y/o de rendimiento, entre otras tantas posibilidades.

Dentro de este proceso, la fase de mantenimiento es una de las etapas que mayor costo, en tiempo y recursos, insume. Se estima que la proporción de recursos y tiempos dedicados al mantenimiento de software ronda el rango del 60% al 80% del ciclo total de desarrollo de un sistema [1,2,3,4]. A su vez, Fjeldstad & Hamlen en [9] reportan que entre el 47% y el 62% del tiempo gastado en tareas de mantenimiento, es dedicado a actividades de comprensión. Otras investigaciones estiman que este porcentaje es aún mayor, rondando el rango de entre el 50% y el 90% [10,11,12]. Estos elevados costos en la comprensión de programas son producidos en general porque:

- Las personas encargadas de llevar adelante el mantenimiento de los sistemas raramente son los autores de los mismos (o un período de tiempo significativo ha pasado entre su desarrollo y su mantenimiento).
- Es difícil encontrar documentación completa y actualizada de los sistemas.
- Dificultad para comprender código realizado por otras personas (código sin indentar, poca semántica en las declaraciones de atributos, funciones, procedimientos, etc.).
- Malas prácticas de programación en el código del sistema (uso de sentencias *go to*, poca modularización, baja cohesión y alto acoplamiento de componentes, entre otras tantas).

De lo expuesto en los párrafos previos se puede concluir que es importante mejorar la etapa de mantenimiento dentro del proceso de desarrollo. Para lograr el objetivo antes mencionado es necesario facilitar la comprensión de los mismos [14]. Por este motivo, la elaboración, desarrollo y uso de herramientas que ayuden en este análisis es de gran utilidad.

Este artículo se encuentra organizado como se describe a continuación. La sección 2 brinda el marco teórico sobre el cual se desarrolla esta herramienta abordando los siguientes temas: Proceso Cognitivo, análisis dinámico y estático sobre código fuente, visualización de código y métricas propuestas por diversos autores. En la sección 3 se expone JInsight[1], se describe su arquitectura y la implementación de conceptos teóricos de la sección anterior. La sección 4 muestra un caso de prueba donde se utilizó JInsight, esta tarea se llevo a cabo proporcionando las vistas recuperadas en cada paso de la ejecución. La sección 5 concluye el artículo, brindando las conclusiones y los posibles trabajos a futuro que pueden desprenderse de esta investigación.

2. Marco Teórico

La Comprensión de Programas (CP) [13,14,15,16] es un área de la Ingeniería del Software destinada a elaborar métodos, técnicas y herramientas, basados en un Proceso Cognitivo y un Proceso de Ingeniería, con el objetivo de facilitar el estudio y entendimiento de sistemas de software.

Por Proceso Cognitivo [18,19] se entiende al estudio y análisis de las fases y pasos seguidos por los programadores para comprender un programa. El Proceso de Ingeniería [20] tiene la finalidad de representar la información del sistema de manera que enfatice sus principales aspectos. Este proceso incluye áreas tales como: Visualización de Software, Extracción de la Información y Administración de la información, entre otras.

Uno de los principales desafíos en CP radica en establecer una relación entre los Dominios del Problema y del Programa [21, 22]. El primero se relaciona con el comportamiento del sistema de estudio; mientras que el segundo se centra en las componentes del programa que producen dicho comportamiento. A través del estudio de la literatura de Comprensión de Programas se pudo comprobar que un programador entiende un programa cuando puede encontrar las componentes de software usadas para producir la salida del sistema [21]. En otras palabras, cuando puede relacionar los Dominios del Problema y Programa. El camino adecuado para alcanzar este objetivo, como se mencionó en el párrafo previo, consiste en:

1. Proveer una representación del Dominio del Problema.
2. Proveer una representación del Dominio del Programa.
3. Definir un proceso que permita unir ambas representaciones.

[1] JInsight es una herramienta desarrollada en el marco de una tesis de licenciatura en ciencias de la computación. Es importante mencionar que esta herramienta difiere de otra, con el mismo nombre, destinada para el análisis dinámico. Dicho análisis está no está considerado en la herramienta presentada en este trabajo.

La figura 2 muestra el modelo de comprensión subyacente a los pasos mencionados con anterioridad.

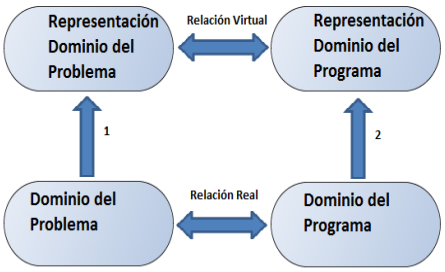


Figura 2. Modelo de Comprensión de Programas.

2.1 Proceso Cognitivo

En un Proceso Cognitivo [28] se encuentran implicadas las estructuras de la información y estrategias de estudio usadas por los programadores para entender programas. Los Procesos Cognitivos utilizan las siguientes componentes:

- Conocimiento,
- Modelo Mental,
- Proceso de Asimilación.

Existen dos tipos de conocimiento: general y específico. El primero es independiente del sistema de software a comprender. Se compone por un conjunto de conceptos y relaciones que conforman la estructura de conocimiento del programador. El segundo, representa aquellos componentes que son los nuevos conceptos proporcionados por el sistema de estudio. Durante el proceso de comprensión, los ingenieros de software adquieren conocimiento específico y general, aunque este último suele venir dado en gran parte por la experiencia del ingeniero de software.

El Modelo Mental es la representación mental que tiene el programador del sistema de estudio. Codifica la comprensión del programa del ingeniero de sistema. El grafo de funciones, comunicaciones de módulos, etc. son posibles modelos mentales.

Finalmente, el Proceso de Asimilación describe como el Modelo Mental evoluciona usando el conocimiento del programador junto con el código fuente del programa y la documentación. Se pueden considerar los siguientes Procesos de Asimilación, dependiendo del conocimiento inicial del ingeniero de software del sistema bajo análisis:

- Top-down,
- Bottom-up,
- Híbrido.

En el proceso de asimilación Top-Down el programador realiza una reconstrucción del conocimiento acerca del dominio del programa e intenta mapear este conocimiento hacia el código fuente.

El proceso de asimilación Bottom-up es el proceso inverso al Top-down, en el cual el programador primero lee declaraciones en el código fuente y luego mentalmente agrupa estas declaraciones en abstracciones de más alto nivel. Se realizan abstracciones hasta que la comprensión del sistema sea lograda.

El proceso de asimilación Híbrido es una combinación de los procesos de asimilación previamente mencionados. Se realiza un proceso Top-down cuando el código es familiar. Por el contrario, cuando el código no resulta familiar, se desarrolla un proceso Bottom-up.

Los pasos mencionados previamente conforman la base para construir verdaderas aplicaciones de Comprensión de Programas. En consecuencia, se deben tener presentes durante el diseño de cada una de las partes constituyentes, es decir, para los métodos de extracción de la información y las visualizaciones de programas.

2.2 Proceso de Ingeniería

El proceso de Ingeniería incluye el estudio de diversas áreas con la finalidad de representar la información del sistema de una manera que enfatice sus principales aspectos.

2.2.1 Análisis dinámico sobre código fuente

El análisis dinámico se refiere al proceso de extraer información de un sistema cuando está en ejecución. Dentro de las técnicas de extracción dinámica más conocidas se pueden mencionar: Slicing Dinámico de Programas [39], Profiling [40] e Instrumentación de Código [14]. Esta información es útil para analizar el funcionamiento del programa y establecer las partes del mismo utilizadas en resolver determinados aspectos del problema. Por ejemplo, si se ejecuta un sistema para cargar los datos de un nuevo cliente, se pueden conocer aquellas funciones, procedimientos y métodos empleados en llevar adelante dicha sub-tarea.

2.2.2 Análisis estático sobre código fuente

El análisis estático se realiza cuando el programa no se encuentra en ejecución. En este artículo se trata el análisis estático, particularmente sobre programas escritos basados en el paradigma orientado a objeto. El

objetivo de este tipo de análisis es el de extraer información acerca de un programa sin ejecutar el mismo para examinar el código fuente, chequear el cumplimiento de reglas específicas, controlar el uso de argumentos, etc.; permitiendo al ingeniero razonar sobre todos los comportamientos posibles que podrían surgir en tiempo de ejecución.

Los tipos de análisis estático que se pueden practicar son manual y automatizado. El examen manual es realizado por el ingeniero mediante la inspección directa del código. Esta inspección se realiza sobre el código fuente, documentación, requerimientos, diseño y cualquier elemento que pueda brindar información del sistema a analizar. Este tipo de análisis tiene la desventaja que consume mucho tiempo y es susceptible a errores de comprensión y omisión de detalles que puedan resultar relevantes.

Por otro lado, el proceso de análisis estático automatizado es llevado a cabo mediante el uso de una herramienta. Este tipo de análisis es más rápido y no requiere del conocimiento por parte del programador que es necesario para un análisis manual, aunque si es preciso que pueda comprender la salida que produce la herramienta para poder realizar el análisis.

El análisis estático tiene las siguientes ventajas:

- Completitud. Debido a que se realiza sobre todo el código fuente del sistema en estudio, permiten obtener un análisis completo del mismo.
- Precisión en el análisis del código fuente. Permite conocer y deducir propiedades y relaciones inherentes a las componentes del sistema como por ejemplo relaciones entre métodos, tipos de estructuras de datos existentes, entre otras.
- Construcción de representaciones completas del sistema. Este tipo de técnicas permiten crear representaciones globales del sistema que dan una visión total del mismo. Además, dependiendo del tipo de representación creada, se pueden aplicar distintos tipos de algoritmos existentes para deducir otras propiedades del sistema. Por ejemplo, se puede construir el grafo de funciones de un sistema y luego, aplicando algoritmos de análisis de grafos, conocer subsistemas dentro del sistema, funciones aisladas, etc.
- Los resultados del análisis son, en general, una descripción de la estructura del sistema, no importa con que entradas o en qué ambiente el programa se ejecuta.

Se puede mencionar como ejemplos de análisis estático los que realizan los compiladores, revisiones formales, inspecciones, análisis de flujo de control, análisis de flujo de datos, análisis de alias, etc.

2.2.3 Visualización de Código

Un aspecto importante involucrado en la Comprensión de Programas es la visualización de software. Esta es una disciplina de la Ingeniería del Software que provee una o varias representaciones visuales de la información de los sistemas permitiendo una mejor comprensión de los mismos. Dichas representaciones, conocidas como vistas, actúan como puente cognitivo entre el conocimiento que posee el programador y los conceptos usados por el sistema. Existen distintos tipos de vistas, dependiendo de la información que se desee visualizar. Las vistas pueden ser estáticas, obtenidas sin ejecutar el sistema, o dinámicas, obtenidas al ejecutar el sistema. Algunos ejemplos de vistas estáticas son:

- Código fuente. El código fuente en sí mismo es una vista. Se pueden aplicar distintas técnicas sobre él para que sea más comprensible. Por ejemplo una indentación adecuada, resaltar palabras claves, permitir reducir y maximizar bloques de código, etc. Este tipo de vista toma mayor relevancia cuando ella forma parte de un conjunto de vistas que muestran la información extraída de este código fuente de distintas maneras.
- Treemap. Muchas técnicas han sido propuestas para mostrar estructuras jerárquicas de manera eficiente, por ejemplo grafos. Estas técnicas son eficientes para grafos pequeños, pero fallan cuando la cantidad de nodos a visualizar de manera simultánea es grande. La principal razón de esta falencia es el uso ineficiente del espacio de visualización. Los treemap [34,35] fueron desarrollados para remediar este problema. El espacio de visualización total es utilizado para representar el contenido del árbol dentro de un espacio rectangular. Son visualizaciones efectivas y compactas. Con el treemap se puede mostrar distintos niveles de abstracción del sistema. Un ejemplo de una vista mediante treemap se puede ver en la figura 3 la cual fue extraída de [35].

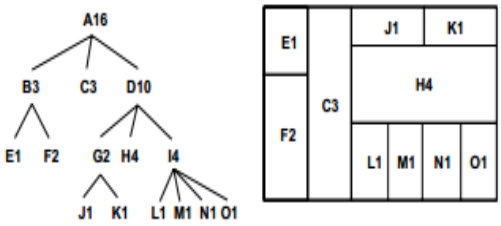


Figura 3. Representaciones de árbol.

- Grafos. Visualizaciones basadas en grafos donde los nodos representan entidades del código fuente, tales como paquetes, clases, métodos y atributos. Los arcos denotan relaciones de dependencia entre ellos, tales como herencia, llamadas de métodos y acceso a campos. Un ejemplo de grafo es proporcionado por la herramienta DA4Java [36], la cual muestra las dependencias entre entidades del código fuente. Esto se puede ver en la figura 4.

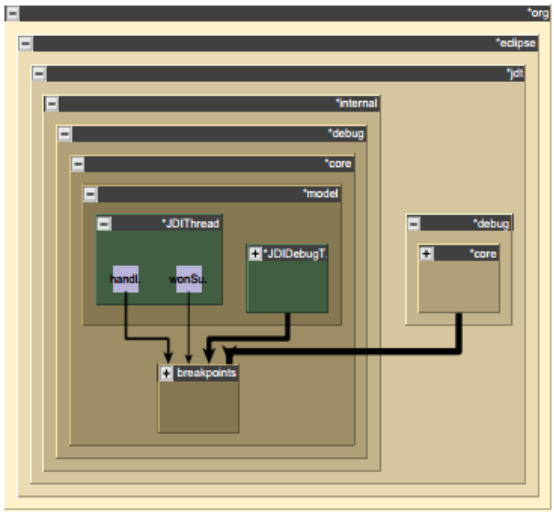


Figura 4. DA4Java. Grafo del sistema.

Muchos sistemas de visualización de software presentan diferentes visualizaciones del programa (Dominio del Programa) pero no contemplan otras interesantes como es la salida del sistema (Dominio del Problema). En este sentido son escasas las estrategias de visualización de este dominio, lo cual indica que elaborar una de ellas representa un desafío importante en el campo de la Comprensión de Programas.

Un medio importante para entender, controlar y mejorar los desarrollos de software son las métricas. Debido a la importancia de ellas en la sección siguiente se aborda dicha temática.

2.3 Métricas

Una métrica es una medida cuantitativa del grado en que un sistema, componente o proceso posee de un atributo dado. Las métricas para sistemas orientados a objetos deben ajustarse a las características que distinguen el software orientado a objetos del software convencional. Estas métricas hacen hincapié en la abstracción, encapsulamiento, ocultamiento de la información, polimorfismo, herencia, cohesión y acoplamiento. Entre los objetivos principales de las métricas se encuentran:

- Comprender mejor la calidad del producto.
- Estimar la efectividad del proceso.
- Mejorar la calidad del trabajo.

A continuación, se presentan métricas desarrolladas por distintos autores, brindando una breve descripción de cada una de ellas [37]. Estas métricas fueron consideradas porque se aplican a sistemas orientados a objetos, particularmente aquellos escritos con el lenguaje de programación Java.

2.3.1 Métricas de Chidambery Kemerer (CK). Son métricas orientadas a clases: clases individuales, herencias y colaboraciones.

- DIT (Depth Inheritance Tree). Esta métrica representa la distancia desde una clase a la raíz del árbol de herencia. Cuanto más alto es el valor, mayor complejidad hay en el diseño y más posibilidades existen de que se reutilicen métodos heredados.
- NOC (Number of Children). Esta métrica representa el número de hijos inmediatos en el árbol de herencia. Cuanto más alto es el valor, mayor es la probabilidad de que una clase sea más reutilizable pero también lo es la probabilidad de que se hayan hecho extensiones no apropiadas de la clase es mayor.
- CBO (Coupling between Object Classes). Esta métrica representa el número de clases acopladas a una clase. Dos clases están acopladas cuando los métodos de una de ellas usa variables o métodos de una instancia de la otra. Si existen varias dependencias sobre una misma clase es computada como una sola. Cuanto más alto más difícil será el mantenimiento, el reuso, y en general el código será más propenso a errores.
- LCOM (Lack of Cohesion). Esta métrica calcula el conjunto de atributos comunes en los métodos de una clase. Cada método de una clase tiene acceso a uno o más atributos. A mayor número de atributos compartidos entre métodos de una clase, mayor cohesión hay en la clase. Es importante conseguir valores bajos.

- WMC (Weighted Methods per Class). Esta métrica calcula la complejidad de una clase mediante la suma de las complejidades de sus métodos.
- WAC (Weighted Attributes per Class). Esta métrica especifica el número de atributos de la clase.

2.3.2 Métricas de Lorenz y Kidd. Dividen las métricas basadas en cuatro categorías: tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaño para una clase individual se centran en cálculos de atributos y de operaciones y promedian los valores para el sistema orientado a objetos en su totalidad. Algunas son:

- PIM (Public Instance Method). Esta métrica representa el número de métodos de instancia públicos.
- NIM (Number of Instance Method). Esta métrica representa la cantidad de métodos de instancia.
- NIV (Number of Instance Variable). Esta métrica representa el número de variables de una instancia.
- NCM (Number of Class Method). Esta métrica representa el número de métodos de clase.

Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones en la jerarquía de clases. Algunas de ellas son:

- NMO (Number of Method Override). Esta métrica representa el número de métodos redefinidos.
- NMI (Number of Method Inheritance). Esta métrica representa el número de métodos heredados.
- NMA (Number of Method Added). Esta métrica representa el número total de métodos que se definen en una subclase.

Las métricas orientadas a valores internos de clase examinan la cohesión y asuntos relacionados con el código. La más conocida es:

- APPM (Average Parameter per Method). Esta métrica representa el promedio de parámetros por método (Número total de parámetros por métodos/Número total de métodos).

Las métricas relacionadas a valores externos examinan el acoplamiento y la reutilización.

3. JInsight

La figura 5 muestra la arquitectura de JInsight. Como se puede observar esta herramienta está compuesta por los siguientes módulos: modelo, vista y controlador. Cada uno de estos y el sistema de entrada que es provisto como entrada a JInsight se describen en las sub-secciones siguientes.

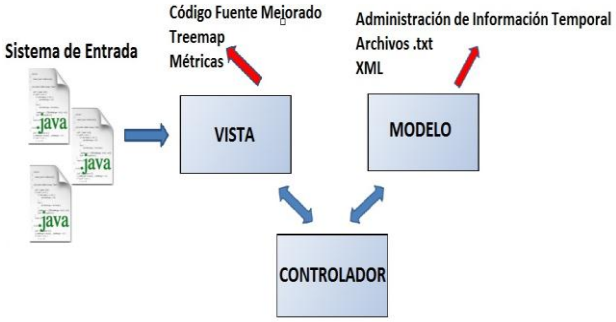


Figura 5. Arquitectura de JInsight.

3.1 Sistema de Entrada

El sistema de entrada corresponde al código fuente a ser analizado. Para analizar archivos pertenecientes a un sistema estos deben ser seleccionados uno a uno. Los mismos deben tener extensión .java.

3.2 Controlador. Módulos para la extracción de la información

Estos módulos extraen información estática del sistema bajo análisis. Para la extracción de la información se utilizan técnicas de compilación tradicionales. Para ello, JInsight hace uso de Antlr [38], una herramienta que recibe como entrada una gramática libre de contexto y construye automáticamente un lexer y parser correspondiente al lenguaje especificado por la gramática, en este caso Java. Esto permite no solo detectar si un sistema es sintácticamente correcto, sino también poder extraer información del mismo. Para lograr dicha extracción, se insertan acciones semánticas en la gramática que será la entrada de Antlr. Estas acciones semánticas se escriben en el mismo lenguaje que usa la gramática generada. Antlr cuando construya el reconocedor (lexer y parser) incorporará al mismo el código dentro de las acciones semánticas en los lugares especificados dentro de la gramática. Las acciones semánticas incorporadas a la gramática de Java para la construcción de la herramienta tienen la función de extraer información de nombres de clases, métodos, atributos y otra información que será de interés para ser visualizada. La figura 6 muestra la regla sintáctica *normalClassDeclaration* de la gramática Java a la cual se le han incorporado acciones semánticas para la extracción de código. Las mismas se encuentran encerradas entre llaves y están escritas en lenguaje Java. En la imagen, dichas acciones, han sido resaltadas con color rojo para una mejor identificación.

```
normalClassDeclaration returns [String nCD]
scope{ Clase clase; }
@init{$normalClassDeclaration::clase= new Clase();
      $compilationUnit::metodos= new ArrayList<String>(); }
  :  modifiers'class' IDENTIFIER {nCD=$modifiers.mod + "class " + $IDENTIFIER.text;
                      $normalClassDeclaration::clase.setNombre(nCD);
                     }
     (typeParameters)?
     ('extends'type {nCD+=" extends "+$type.t;})?
     ('implements'typeList {nCD+=" implements "+$typeList.tl;} )?
     classBody {nCD+=$classBody.cB;}
     {$normalClassDeclaration::clase.setMetodos($compilationUnit::metodos);}
     {$compilationUnit::datosProyecto.setClases($normalClassDeclaration::clase);}
  ;
```

Figura 6. Incorporación de acciones semánticas a la gramática de Java.

3.3 Modelo. Repositorio de la información

Esta componente posibilita el almacenamiento de toda la información extraída del sistema. Esta información inicialmente es almacenada en archivos. Cada archivo contiene información relacionada a un nivel jerárquico. Los niveles jerárquicos que se consideran son:

- Archivos
- Clases
- Métodos

Además, se construye un archivo XML (eXtensible Markup Language) con la información del nivel jerárquico que es visualizado durante la ejecución de JInsight para que sea provisto como entrada al programa que genera la vista treemap. Dicha vista se detalla en la siguiente sección.

3.4 Vista. Perspectivas del sistema a ser visualizadas

La interfaz de la herramienta cuenta con una barra de menú, desde la cual se debe seleccionar los archivos a analizar. Bajo la barra de menú, se ha incorporado una barra de herramientas, la cual provee una lista desplegable que permite seleccionar, entre los archivos escogidos para el análisis, uno sobre el cual se aplicarán las visualizaciones. Además esta barra de herramientas incorpora un botón que permite retornar al nivel jerárquico anterior en todas las vistas, en el caso de ser posible. El área principal de la interfaz se encuentra divido en cuatro partes. Tres de ellas, visualizan de una manera diferente la información recuperada del código fuente. En cada momento de la ejecución de JInsight se visualiza un nivel jerárquico del archivo. Al nivel visualizado se lo denomina *ámbito*.

A continuación se muestra, en la figura 7, una instantánea de la interfaz de la herramienta señalando cada componente de la interfaz.



Figura 7. Visualización de la interfaz de JInsight.

A continuación se describen cada una de los componentes del área principal de la interfaz:

- **Código Fuente Mejorado**. Al traductor construido con Antlr se le incorporaron acciones semánticas que permitan recuperar el código fuente a analizar con la indentación adecuada, aunque el archivo original no la tenga, de forma tal de facilitar la comprensión. Además, se resaltan las palabras claves, nombres de clases y métodos, para obtener una más rápida localización de las partes del código que se está analizando. Estas palabras resaltadas son hipervínculos, los cuales al ser presionados permiten profundizar el ámbito de análisis en todas las vistas proporcionadas en la interfaz. Si la visualización se encuentra en el ámbito *Archivo*, las palabras resaltadas son los nombres de las clases. Al hacer click sobre una de estas palabras se profundiza al ámbito *Clases*, reemplazando el código que está siendo visualizado por el que corresponde a la clase que se encuentra asociada al hipervínculo que se presionó. Además, al hacer click sobre estos hipervínculos, la vista treemap se actualiza para visualizar la estructura jerárquica del código de la clase. El listado de métricas también es actualizado al proporcionar las métricas disponibles para el ámbito Clases. Si se presiona nuevamente sobre una palabra resaltada, que en este caso serían nombres de métodos, se profundiza al ámbito *Métodos*. El código fuente que se visualiza es el del método asociado al

hipervínculo presionado. Tanto la vista treemap como las métricas serían inhabilitadas debido a que no se proporciona información de ellas en este nivel de jerarquía.

- **Estructura Jerárquica**. Se hace uso de la librería Prefuse [41] debido a que permite generar la vista treemap. Esta vista es de gran interés por hacer un uso eficiente del espacio, como se expuso en la sección previa. Esta visualización proporcionada por la librería provee un campo de texto para realizar búsquedas sobre la vista. En este campo de búsqueda se ingresa un texto que en el caso de haber coincidencias con un elemento mostrado en el treemap resalta en él las zonas que coincidan. El treemap muestra sólo un nivel de profundidad sobre el archivo siendo analizado. Si el ámbito es *Archivo* muestra las clases del archivo, si el ámbito es *Clases* muestra métodos de la clase y por último si el ámbito es *Métodos* no se provee información en esta visualización. El ámbito puede identificarse a la derecha del cartel *Ámbito de análisis* que se encuentra en la barra de herramientas.
- **Métricas**. Se hace uso de la librería JFreeChart [42] para visualizar las métricas recuperadas del código fuente. Se escogió JFreeChart debido a que es una librería de código libre disponible para Java. Para visualizarlas se utiliza la vista diagrama de barras proporcionada por esta librería. El listado de métricas posibles a visualizar depende del nivel de jerarquía que se encuentre visualizado. Cuando la aplicación es ejecutada, se visualiza el listado de métricas, asociado al nivel jerárquico Archivo, pero no el diagrama de barras asociado a una de ellas. Debe seleccionarse una métrica de la lista para que el diagrama de barras asociado a esta pueda visualizarse. Al seleccionar una métrica, en el cuadro a la derecha de la lista se proporciona un listado de interpretaciones de los resultados de las métricas que ayudan al usuario en la comprensión del código fuente siendo analizado. Las métricas implementadas a nivel de archivos son: DIT, NOC, NMO, LOC, WAC, WMC y PIM. Las métricas implementadas a nivel de clases son: NOP y LOC.

Para cada vista de la interfaz se proporcionan barras de desplazamiento cuando el tamaño de la imagen no puede visualizarse de manera completa.

En el caso de haber profundizado en el ámbito de análisis, es posible subir a los niveles de jerarquía visualizados previamente al presionar sobre el botón con una flecha de color verde, de aquí en adelante se denominará *ámbito previo*, que se encuentra a la derecha de cuadro que muestra en ámbito de análisis en la barra de herramientas.

- **Documentación a generar para realizar el análisis**. Esta parte de la interfaz sirve para mantener reportes de los análisis que realice el usuario. Está dividido a su vez en dos partes, cada una con un editor de texto y un botón que permite guardar en documentos con extensión .txt lo que se ingrese en cada uno de ellos. El propósito del Área de Documentación que se encuentra arriba es para que el usuario pueda guardar las conclusiones que realiza del análisis. Por otro lado, la que se encuentra abajo es para que pueda guardar las dudas generadas en el análisis. Para guardar en un documento los comentarios o dudas se debe presionar el botón con imagen de disco a la derecha de cada editor. Al hacer click sobre estos, si es la primera vez desde el inicio de la ejecución de la herramienta, se genera una carpeta que corresponde al tipo de documento a generar y en ella se crea el documento con el nombre de la versión seguido de *_conclusiones* para documentos de conclusiones y *_dudas* para documentos de de dudas. La ruta donde se almacenan estos documentos es el directorio donde se encuentra el archivo que está siendo analizado. Esta ruta además es especificada al presionar el botón guardar en del cuadro de diálogo.

La navegación proporcionada por los hipervínculos del área de Código Fuente Mejorado y la posibilidad de volver al ámbito anterior, permiten al usuario realizar distintos procesos de asimilación durante el proceso cognitivo que está llevando a cabo sobre el código fuente bajo análisis. Los usuarios pueden tomar un enfoque top-down al hacer click sobre los hipervínculos o un enfoque bottom-up al volver al ámbito previo al presionar el botón *ámbito previo* o ambos alternando el uso de los botones antes mencionados. Esta tarea la lleva a cabo el ingeniero según su criterio.

Cada una de las visualizaciones puede ser maximizada al tamaño total de la pantalla al arrastrar sus bordes. Esto es importante cuando el código abarca varias líneas, cuando en la estructura jerárquica la raíz tiene varios hijos y en el caso de las métricas cuando la visualización no permite ver el nombre de cada una de las barras y/o las barras en su totalidad.

4. Caso de Prueba

En esta sección se muestra como JInsight es utilizado para extraer información estática de sistemas escritos en Java y como ésta herramienta visualiza la información del sistema de distintas maneras. Se lleva a cabo un caso de prueba sobre el código fuente de un editor de texto plano escrito en Java.

4.1 Caso de estudio: Editor de texto plano

En esta sección se presenta un caso de estudio, el análisis de un Editor de Textos Plano (ETP), con el propósito de mostrar las funcionalidades JInsight. Se deja como trabajo futuro la realización de sesiones de trabajo en donde se puedan analizar los efectos producidos por JInsight para la comprensión de programas.

ETP está formado por cuatro archivos de código fuente Java, cuyos nombres son los siguientes archivos: Busca.java, Buscareemplaza.java, Editor.java y ElegirColor.java.

Para extraer información de estos archivos y visualizarla se deben realizar los siguientes pasos:

1. Hacer click en el botón *Archivo* en la barra de menú.
2. Seleccionar la opción *Abrir*.
3. En el cuadro de diálogo abierto se puede seleccionar uno o más archivos a analizar.
4. Una vez seleccionado el/los archivos presionar el botón *Analizar*.

Cuando los pasos previamente mencionados se han realizado se visualizan las vistas proporcionadas por JInsight, las cuales representan la información extraída del archivo Busca.java. En primer lugar se visualiza Busca.java (ver figura 8) debido a que en el orden alfabético es el primero de los archivos seleccionados. El ámbito que se visualiza es el de *Archivo*. Como no se ha seleccionado ninguna métrica, la zona de métricas se encuentra con fondo gris. Debe seleccionarse alguna de la lista para que el diagrama de barras asociado a la misma sea visualizada, esto puede apreciarse en la figura 8.

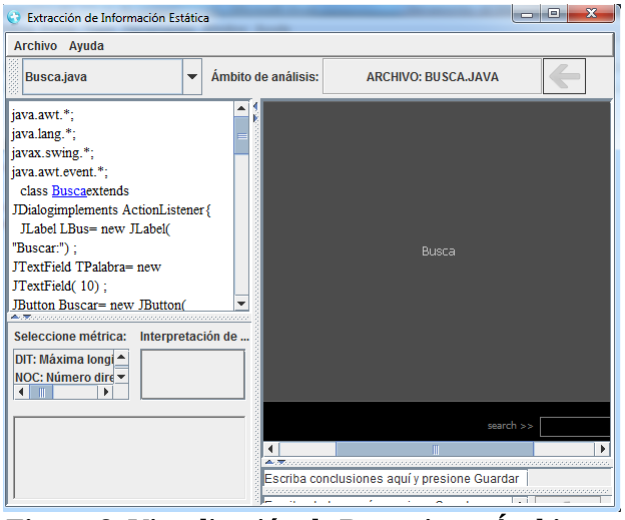


Figura 8. Visualización de Busca.java. Ámbito Archivo.

Posteriormente se selecciona la métrica WAC. Se visualiza el diagrama de barras con la información de la misma como puede verse en la figura 9.

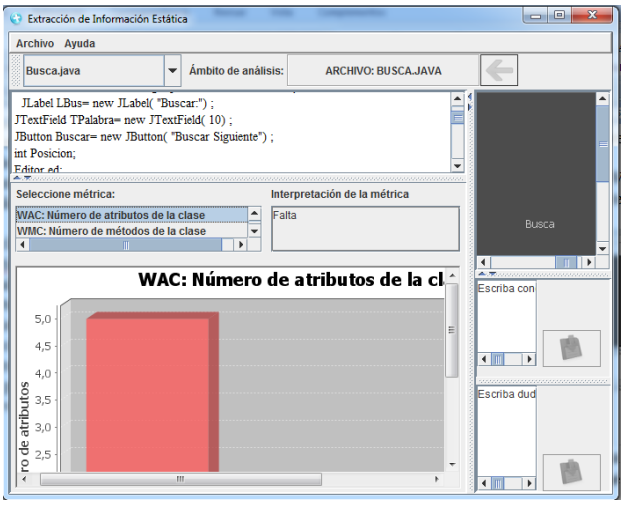


Figura 9. Diagrama de barras de la métrica WAC.

Si se amplía la visualización de las métricas para disminuir el tamaño de la imagen que es capturada (ver figura 10), se puede notar que la escala del eje Número de Atributos se modifica.

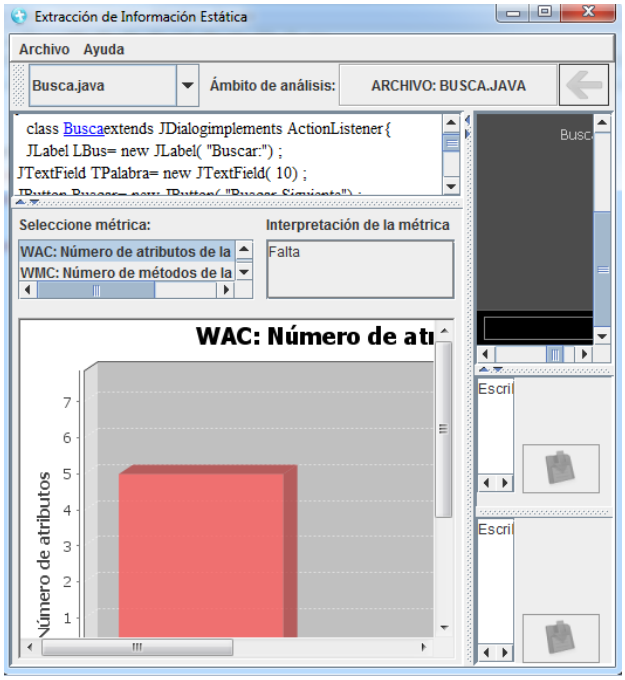


Figura10. Alejamiento de la visualización del diagrama de barras.

Al hacer click sobre el hipervínculo Busca que se encuentra en el área Código Fuente Mejorado, el cual corresponde al nombre de la definición de una clase, se visualiza en esta área el código fuente asociado a la misma y en la vista treemap, se visualiza la jerarquía de la clase.

En la clase Busca se han declarado dos métodos: Busca y actionPerformed. Parte del código de la clase se puede visualizar en la zona de Código Fuente Mejorado y los dos métodos se pueden ver graficados en la zona del treemap. Esto se puede apreciar en la figura 11.

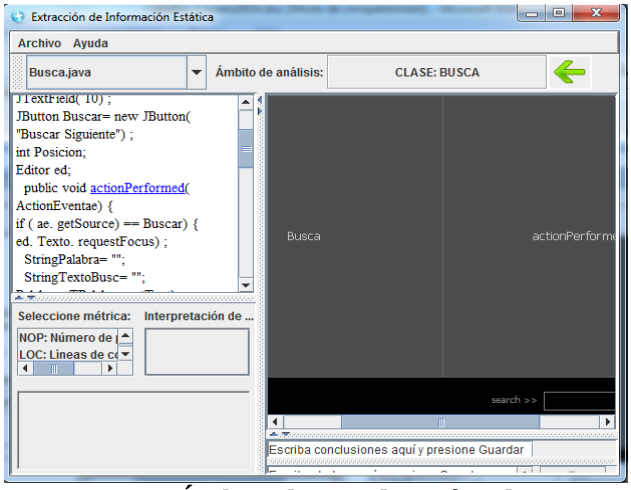


Figura 11. Ámbito de visualización Clase para el archivo Busca.java.

Además se puede notar, en la figura 11, que en la barra de herramientas se habilitó el botón *ámbito previo* apuntando hacia la izquierda. Este botón permite volver al ámbito de análisis previo. Al presionarlo se volvería a visualizar la figura 8.

Al ingresar el texto *act* en el campo de texto sobre la vista treemap el se puede observar que se resalta el área del rectángulo que tiene como prefijo act, como se puede ver en la figura 12.

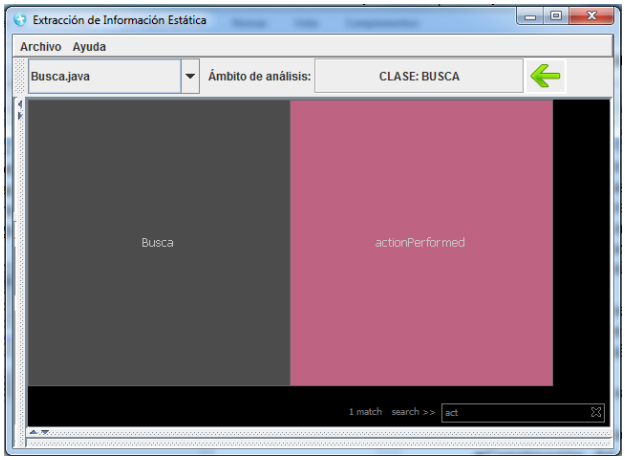


Figura 12. Búsqueda en la vista treemap por el prefijo act.

Además para notar claramente la estructura jerárquica que es representada por el treemap se maximizó la vista al desplazar los bordes hasta abarcar toda el área de visualización de la interfaz. Aquí se puede notar que en la clase Busca del Archivo Busca.java se han declarado dos métodos: Busca y actionPerformed.

Al hacer click sobre el hipervínculo actionPerformed, el cual es un método, se visualiza, en el área Código Fuente Mejorado, el código asociado al método. La zona de métricas se inhabilita en este nivel de jerarquía al igual que la zona de visualización del treemap, ya que no se proporciona información sobre estas vistas en este nivel.

Al cargar comentarios en la zona de Conclusiones y presionar el botón *guardar* se almacena el documento con el texto ingresado en el editor. Posteriormente puede visualizarse, en el mismo directorio donde se encuentra el archivo siendo analizado, que se creó una carpeta con el nombre Conclusiones y dentro de ella un archivo con las conclusiones recién cargadas con el nombre 0_conclusiones.txt, que corresponde a la primera versión.

Al cargar dudas en el editor que se encuentra abajo en el área de Comentarios y luego presionar guardar se crea la carpeta Dudas y el archivo 0_Dudas.txt.

Al presionar sobre la lista desplegable que se encuentra en la esquina superior izquierda se selecciona el archivo BuscaReemplaza.java para visualizar información extraída de este archivo. Esto se puede ver en la figura 13.

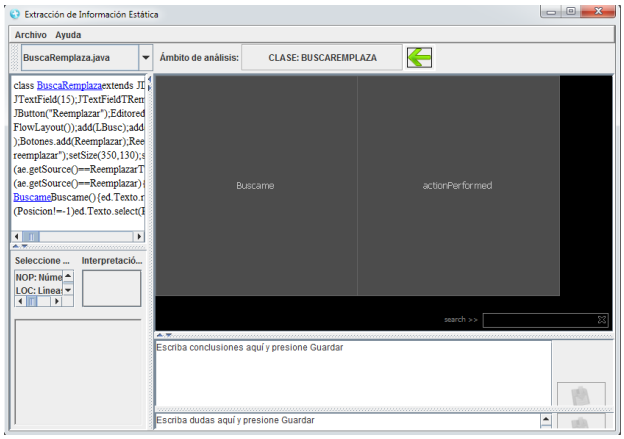


Figura 13. Selección del archivo BuscaReemplaza.java.

De esta manera se ha realizado un recorrido por todas las funcionalidades que actualmente provee la herramienta y se mostró que las mismas no son difíciles de llevar a cabo. Esta característica es relevante porque el ingeniero puede concentrarse directamente en el análisis del sistema de estudio sin distraerse en el aprendizaje de una nueva herramienta.

5. Conclusiones y Trabajos a Futuro

Luego de haber estudiado teoría de Comprensión de Programas, implementado JInsight y haber realizado pruebas con sistemas reales se puede concluir que esta herramienta brinda al usuario los siguientes beneficios en la tarea de comprensión de un sistema:

- Interfaz sencilla y fácil de comprender lo que facilita la usabilidad.
- Fácil compresión de resultados.
- Múltiples vistas.
- Permite evaluar la calidad ofreciendo una gran cantidad de métricas aplicadas a archivos, clases y métodos.
- Por cada métrica ofrece una lista de interpretaciones posibles para ayudar al usuario en el entendimiento de los resultados.
- Se adapta al enfoque que quiera utilizar el programador al permitirle utilizar en cualquier momento un proceso de asimilación top-down o bottom-up por medio de la navegabilidad que proveen los hipervínculos y la posibilidad de volver al ámbito previo visualizado.
- Embellece el código fuente resaltando las palabras claves e indentándolo correctamente.
- Ofrece la posibilidad de filtrar la información sobre la vista treemap al proveer un campo de texto para realizar búsquedas sobre ella. Cada área rectangular visualizada en el treemap está identificada por un nombre. Este nombre puede representar una clase o método, dependiendo del ámbito actual durante la ejecución. Si el texto ingresado en el campo de texto coincide con alguno de estos nombres se resalta el área de estas zonas, permitiendo así reducir el área de concentración.
- Permite focalizarse sobre alguna visualización de la interfaz al permitir desplazar los bordes de la zona hasta el máximo de la pantalla.

Como trabajos a futuro sobre JInsight se plantean las siguientes mejoras:

- Permitir al usuario definir nuevas métricas.
- Construcción del árbol de análisis sintáctico (AST).
- Incorporar análisis dinámico para realizar un análisis más completo sobre el sistema, el cual permita además obtener vistas de la relación del dominio del problema con el dominio del programa.
- Visualizar la tabla de símbolos.
- Permitir la selección de un proyecto completo para su análisis.
- Realizar sesiones de trabajo en donde se pueda comprobar empíricamente como JInsight mejora la comprensión.

6. Referencias

[1] A. Abran and H. Nguyenkim. Analysis of maintenance work categories through measurement. *In Software Maintenance, 1991., Proceedings. Conference* on, pages 104-113, oct 1991.

[2] Ghazi Alkhatib. The maintenance problem of application software: an empirical analysis. *Journal of Software Maintenance*, 4(2):83-104, June 1992.

[3] Lowell Jay Arthur. *Software evolution: the software maintenance challenge*. Wiley Interscience, New York, NY, USA, 1988.

[4] Barry W. Boehm. *Software Engineering Economics*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1981.

[5] E. Burton Swanson and Cynthia Mathis Beath. *Maintaining information systems in organizations*. John Wiley & Sons, Inc., New York, NY, USA, 1989.

[6] John R. Foster. *Cost factors in software maintenance*. PhD thesis, Durham University. United Kingdom, 1993.

[7] James Martin and Carma L. McClure. *Software Maintenance: The Problems and Its Solutions*. Prentice Hall Professional Technical Reference, 1983.

[8] Bennett P. Lientz and E. Burton Swanson. *Software Maintenance Management*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1980.

[9] J. D. Novak and D. B. Gowin. *Learning How to Learn*. Cambridge University Press, 1984.

[10] T. A. Corbi. *Program understanding: challenge for the 1990's*. IBM Syst. J., 28(2):294-306, June 1989.

[11] Panos E. Livadas and David T. Small. Understanding code containing preprocessor constructs. In *IEEE Third Workshop on Program Comprehension*, pages 89-97,1994.

[12] Thomas A. Standish. An essay on software reuse. *IEEE Trans. Software Eng.*, 10(5):494-497, 1984.

[13] Spencer Rugaber. *Program Comprehension*. 1995.

[14] M. Beron, P. Henriques, and R. Uzal. *Program Inspection to interconnect Behavioral and Operational Views for Program Comprehension*. Technical Report, 2007.

[15] T. A. Corbi. *Program understanding: challenge for the 1990's*. IBM Syst. J., 28(2):294-306, June 1989.

[16] Yann-Gaël Guéhéneuc. Taupe: towards understanding program comprehension. In *Hakan Erdogmus, Eleni Stroulia, and Darlene A. Stewart, editors, CASCON*, pages 1-13. IBM, 2006.

[17] Thomas D. Latoza, David Garlan, James D. Herbsleb, and Brad A. Myers. Program comprehension as fact finding. In *Proceedings of Foundations of Software Engineering*, 2007.

[18] H. A. Muller M. A. D. Storey, F. D. Fracchia. *Cognitive design elements to support the construction of a mental model during software exploration*. 1998.

[19] Su Xiaohong Ma Peijun Wang Kechao, Wang Tiantian. *Overview of program comprehension*. 2012.

[20] Pedro Rangel Henriques Maria João Varanda Pereira Mario Berón, Roberto Uzal. *Inspección de código para relacionar los dominios del problema y programa para la comprensión de programas.* 2008.

[21] Germán Montejano Pedro R. Henriques Maria J. Pereira Mario M. Berón, Daniel Riesco. *Estrategias para facilitar la comprensión de programas*. 2010.

[22] Pedro R. Henriques, Maria João Pereira, José Luis Albanes, Mario M. Berón. *Estrategias para relacionar el dominio del problema con el dominio del programa para la comprensión de programas*. 2011.

[23] Boehm W. Barry. *Software engineering*. 1976.

[24] Bing Wu Jane Grimson Vincent Wade Ray Richardson D O Sullivan Jesus Bisbal, Deirdre Lawless. *An overview of legacy information system migration*. 1997.

[25] S.P. Reiss. *Evolving evolution* [software evolution]. 2005.

[26] Michael L. Nelson. *A survey of reverse engineering and program comprehension*. 1996.

[27] Elliot J. Chikofsky and James H. Cross II. *Reverse engineering and design recovery: A taxonomy*. 1990.

[28] Margaret-Anne Storey. *Theories, Methods and Tools in Program Comprehension: Past, Present and Future*. 2005.

[29] Shaochun Xu. *A Cognitive Model for Program Comprehension*. 2005.

[30] Ivo Gomes, Pedro Morgado, Tiago Gomes, Rodrigo Moreira. *An overview on the Static Code Analysis approach in Software Development.*

[31] Gerard Chroust, Helmut Lexen. *Software Inspections - Theorie, New approaches and an Experiment*.

[34] Michael Burch and Stephan Diehl. Visualizing multiple Hierarchies. In *Proceedings of 13th Conference on Visualization and Data Analysis*. 2006.

[35] Jarke J. van Wijk, Huub van de Wetering. Cushion Treemaps: Visualization of Hierarchical Information. *IEEE. Symposium on Information Visualization*. 1999.

[36] Martin Pinzer, Katja Gräfenhain, Patrick Knab, and Harald C. Gall. A Tool Visual Understanding of Source Code Dependencies. *The 16th IEEE International Conference on Program Comprehension*. 2008.

[37] Ana María García Sánchez. *Evaluación de Métricas de calidad del software sobre un programa Java*. 2010.

[38] Terence Parr. *The Definitive ANTRL Reference*. Building Domain-Specific Languages.

[39] Hiralal Agrawal and Joseph R. Horgan. Dynamic program slicing. *SIGPLAN Not.*, 25:246-256, June 1990.

[40] Susan L. Graham, Peter B. Kessler, and Marshall K. Mckusick. Gprof: A callgraph execution profiler. *SIGPLAN Not.*, 17(6):120-126, June 1982.

[41] http://www.prefuse.org.
[42] http://www.jfree.org/jfreechart/.

Aceleradores de hardware para funciones de software en SoCKIT bajo Linux

Esteban Peláez, Sergio D. Calderón R.
Laboratorio de Electrónica Investigación y Servicio
Universidad Nacional de San Luis, FCFMyN
San Luis, Argentina
{epelaez,scalderon}@unsl.edu.ar

Mario Berón
Área de Programación y metodologías de desarrollo de software
Universidad Nacional de San Luis, FCFMyN
San Luis, Argentina
mberon@unsl.edu.ar

***Resumen*—Este trabajo presenta una arquitectura de un procesador con aceleradores de hardware para optimizar algoritmos de beamforming, utilizando una arquitectura híbrida FPGA/CPU. Cada algoritmo es analizado para determinar las operaciones que demandan más cálculo computacional, las cuales se implementan en hardware. De acuerdo a las funciones a optimizar se desarrollan, (i) periféricos que se comunican con el procesador por medio de un bus o (ii) aceleradores que se comunican a través de la memoria (registros de intercambio). El sistema fue implementado en una placa SocKit que consiste de un procesador dual-core ARM CortexTM-A9 MPCore y una FPGA Cyclone V SE, con esta configuración se logra una reducción en el tiempo de ejecución del algoritmo.**

I. Introducción

La aceleración por hardware, es la utilización de hardware especifico para llevar a cabo algunas funciones más rápido que el software ejecutado en una Central Processing Unit (CPU) de propósito general. La utilización de Field Programmable Gate Array (FPGA) como aceleradores de hardware se está convirtiendo en una práctica común en el diseño embebido [1] [2] [3], pero todavía no hay una metodología estándar y un conjunto de herramientas que faciliten su implementación.

El continuo incremento de la complejidad del diseño, donde los sistemas embebidos se integran de varios componentes, algunos de los cuales se implementan en software y otros en hardware, hace que la tarea del diseñador sea cada vez más difícil. Herramientas de diseño que puedan combinar la flexibilidad y el bajo costo de las soluciones de software, con las características de rendimiento y potencia de los enfoques de hardware se están convirtiendo en imprescindible en el mundo del diseño embebido.

Por otra parte, la FPGA surge como una solución alternativa para implementar un sistema con gran exigencia de lógica con un costo más bajo que el diseño de un Circuito Integrado (IC). Además, la FPGA también permite la reconfiguración del sistema con un tiempo de salida al mercado más corto en comparación con un Circuito Integrado para Aplicaciones Específicas (ASIC). Tales beneficios atrae a los investigadores para desarrollar e instrumentar sistema de procesamiento digital usando FPGA, lo cual la convierte en una plataforma ideal para implementar módulos de hardware de procesamiento de imágenes y motores de procesamiento de vídeo mientras que ofrece un rendimiento cercano a las de un IC personalizado. Esto está en línea con la definición de acelerador de hardware en el procesamiento o cálculo específico que se ejecuta por medio del hardware personalizado configurable.

Para ejemplificar el uso de aceleradores de hardware se empleó un sistema de beamforming que utiliza un array de micrófonos para extraer la señal de voz deseada de un ambiente ruidoso y de interferencia dominante. Los algoritmos utilizados en este campo requieren un rápido procesamiento, por esta razón utilizamos un SoC para implementar este sistema [4] [5] [6].

Los algoritmos de array de micrófonos se basan en la diferencia de tiempo de arribo (TDOA) de la señal de audio a cada micrófono, hay dos grupos principales de algoritmos: invariante en el tiempo y adaptativo[7][8]. Los algoritmos del primer grupo, tales como Generalized Cross-Correlation with Phase Transform (GCC-PHAT) o Delay and Sum Beamformer (DSB)[9][10], son mas simples y rápidos, utilizados para implementaciones de tiempo real. Los algoritmos adaptativos son capaces de adaptar automáticamente su respuesta a diferentes ponderaciones o delay temporales, sin embargo, requieren más capacidad de procesamiento y son complejos de implementar.

Un algoritmo GCC-PHAT que se utiliza para realizar el beamforming es presentado como un caso de estudio, en un diseño que utiliza un array de 6 micrófonos omni-direccionales con una interfaz digital I^2S con una precisión de 24 bits, a una frecuencia de muestreo de 48 KHz. Para implementar este diseño se utiliza SoC de la empresa Altera que esta integrado por un ARM-based hard processor system (HPS) y una memoria de interfaz con la FPGA. El documento está organizado de la siguiente manera: la Sección II describe el estado del arte en SoC FPGAs con HPS. Sección III ofrece una explicación de la arquitectura del sistema que hemos desarrollado. Sección IV se describe la implementación del sistema en un SoCKIT FPGA. Sección V ofrece resultados experimentales. En la sección VI final, el documento se cierra con una conclusión y perspectivas.

II. Estado del Arte

Una de las ventajas de combinar lógica programable y un procesador de propósito general, es el uso de la FPGA como acelerador de hardware (Figura 1.a). Esto significa que algunas operaciones intensivas de computación serán llevadas a cabo

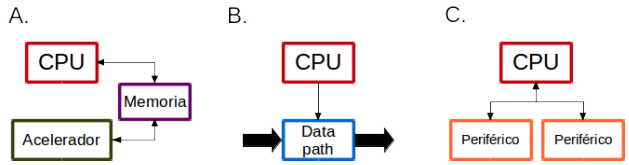


Figura 1. Casos de uso de arquitecturas híbridas

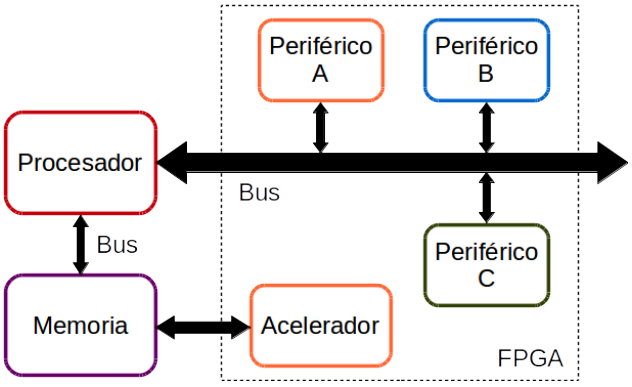


Figura 2. La arquitectura del sistema de hardware de un SoC embebido

por la FPGA para aumentar el rendimiento del sistema, de este modo aligerar el trabajo al procesador. Las aplicaciones típicas de funciones de DSP son enviadas (offloading) hacia la FPGA. No sólo algunas funciones específicas pueden ser implementadas en la FPGA, sino también puede utilizarse como un subsistema independiente (Figura 1.B). Aplicaciones en tiempo real, por ejemplo, pueden ser completamente implementadas en la FPGA, mientras que el HPS continúa ejecutando aplicaciones en tiempo no real. Mediante la aplicación de esta estrategia, se crea una arquitectura plana que contiene un camino lento y un camino rápido. Esto permite añadir interfaces personalizadas al sistema o la adición de periféricos (Figura 1.C), además de las interfaces disponibles estándar en el subsistema del procesador.

Las arquitecturas híbridas FPGA/CPU que se encuentran en el mercado son provista por las empresas Altera y Xilinx[11][12]. Ambas combinan tecnología de 28 nm para fabricar las FPGA con un procesador de propósito general ARM Cortex A9 dual core. La interconexión entre la FPGA y el procesador se realiza en ambos casos usando el protocolo AMBA AXI[13].

Un esquema simple de un SoC Embebido puede estar integrado por un procesador, memorias y periféricos, junto con los bus que los interconectan. Este modelo del sistema de hardware se muestra en la Figura 2.

III. Arquitectura del Sistema

Esta sección describe el principio de funcionamiento del algoritmo GCC-PHAT en el cual se baso la arquitectura del sistema [14]. Éste calcula la dirección de arribo de la fuente de sonido predominante utilizando desplazamiento temporal entre dos micrófonos (TDOA). La diferencia de arribo entre cada uno de ellos se obtiene cuando la correlación cruzada $R(k)$ entre los pares de micrófonos i, j es máxima.

$$\Delta_{ij} = arg_k max R(k) \tag{1}$$

La correlación cruzada se calcula tomando la transformada inversa de Fourier (IFFT) del producto entre la transformada rápida de Fourier (FFT) de uno de los micrófonos y la FFT conjugada del otro, como se muestra en la ecuación 2:

$$R(k) = IFFT((FFT(f(t))) \cdot FFT^*(g(t))) \tag{2}$$

Donde $FFT(f(t))$ es la transformada de Fourier de la señal $f(t)$ del micrófono i, y $FFT^*(g(t))$ es la conjugada de la transformada de Fourier de la señal del micrófono j. $IFFT$ es necesaria para regresar al dominio del tiempo y extraer el valor correspondiente al índice k. Luego se aplica la transformación de fase (PHAT) a la ecuación (2), lo que mejora la robustez contra el ruido. La ecuación 3 define al GCC-PHAT:

$$R(k) = IFFT\left(\frac{(FFT(f(t))) \cdot FFT^*(g(t)))}{|(FFT(f(t))) \cdot FFT^*(g(t)))|^\beta}\right) \tag{3}$$

β es un coeficiente entre $(0, 1)$. Considerando la aproximación de campo cercano, el coseno del ángulo de arribo medido por los micrófonos i y j puede ser calculado como:

$$\cos(\phi_{ij}) = \frac{kv_s}{f_s d_{ij}} \tag{4}$$

Donde f_s es la frecuencia de muestreo, y d_{ij} la distancia entre los micrófonos i, j, y v_s la velocidad del sonido. Debido a que cada par de micrófonos puede proveer un solo ángulo, el resultado desde ellos pueden combinarse como:

$$\cos(\phi) = \frac{(\mathbf{dd}^T)^{-1}\mathbf{k}v_s}{f_s} \tag{5}$$

Donde **dd** es un vector que contiene la distancia entre todos los pares de micrófonos y **k** es el vector que contiene el índice a extraer desde su correlación cruzada. Por lo tanto el ángulo de arribo esta definido por el arco coseno de la ecuación 5.

Para analizar la eficiencia del algoritmo GCC en relación con la viabilidad de realizarlo en hardware, utilizamos un modelo descripto C con todas las operaciones matemáticas necesarias para realizar el calculo del angulo, es decir, sin la utilización de funciones especificas predefinidas. De esta manera se analiza la aplicación puramente secuencial (figura 3). Debido a que nuestro objetivo final es la implementación de aceleradores de hardware, se desarrollo una manera de medir su carga computacional.

Todas las operación se mediran en términos de unidades de multiplicador-acumulador (MA). Una operación MA esta definida como:

$$u = u + v * w \tag{6}$$

La carga del algoritmo GCC-PHAT depende de la cantidad de micrófonos N y el tamaño de la trama Ns (número de

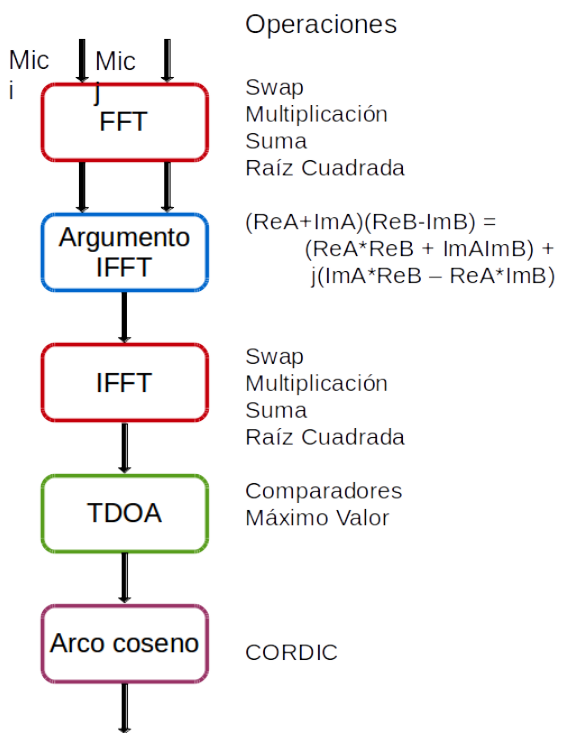


Figura 3. Funciones principales en GCC-PHAT

Cuadro I
CARGA DEL GCC-PHAT EN OPERACIONES DE MULTIPLICADOR-ACUMULADOR

Nro	Operación	Unidades MA
1	FFT	$N_s \cdot \log_2 N_s \cdot N$
2	$IFFT$	$N_s \cdot \log_2 N_s \cdot \binom{N}{2}$
3	$FFT(x) \cdot FFT^*(y)$	$N_s \cdot \binom{N}{2}$
4	$\lvert FFT(x) \cdot FFT^*(y)\rvert$	$4 \cdot N_s \cdot \binom{N}{2}$
5	$\frac{A}{\lvert A\rvert}$	$2 \cdot N_s$

muestras por trama de datos) como se muestra en el Cuadro I.

Analizando la información del Cuadro I se observa que las operaciones que utilizan mayor número de MA son las operaciones 1, 2, siendo la mayor la número 2 por aumentar proporcionalmente a $log_2 N_s$. Pero teniendo en cuenta que las sentencias utilizadas para la $IFFT$ y la FFT son las mismas, se como considerar que las operaciones $IFFT$ y FFT son una sola. La operación candidata a ser implementada por medio de un acelerador de hardware es la FFT.

IV. IMPLEMENTACIÓN

La función FFT[15] se puede descomponer en las siguientes operaciones básicas::

- Acceso a memoria.
- Multiplicación.
- División.
- Suma y resta.
- Desplazamiento de bit

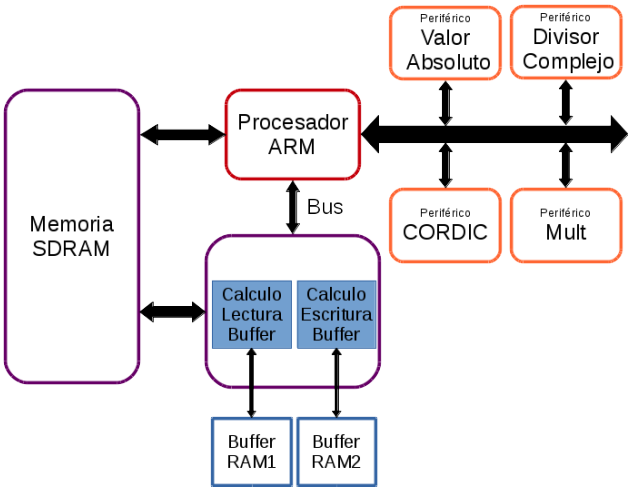


Figura 4. Esquema del sistema que implementa a GCC-PATH

- Iteración con contador

Todas las operaciones mencionadas anteriormente se traducen en construcciones de hardware (figura 4). Una de la operaciones fundamentales en la FFT es el cálculo butterfly, este cálculo reordena los valores a procesar en varios subvectores y viceversa. Esta operación se realiza debido a que la entrada de datos se almacena en forma secuencial pero para realizar el cálculo de la FFT se debe acceder a los datos en un orden preestablecido, esta operación demanda mucho acceso a memoria lo que consume muchos ciclos de reloj, la forma de acelerar esta operación es precalcular la posición en que debe ir el dato al escribirlo, así al acceder a él se puede hacer en forma secuencial. Para mejorar el rendimiento del acelerador, se debe optimizar el código para permitir que las tareas de escritura y lectura sean independientes para poder paralelizarlas. Esta técnica de optimización requiere de buffering rápido de datos, que se puede implementar utilizando las memorias on-chip disponibles en la FPGA.

La aplicación software FFT realiza los cálculos de los datos en el lugar, utilizando el mismo buffer de memoria para cargar los valores de entrada, guardar los resultados de los cálculos intermedios, y almacenar los valores de salida. Los cálculos en el lugar ahorran recursos de memoria, pero el rendimiento suele ser lento cuando se traduce al hardware. La concurrencia completa es difícil de lograr con una única memoria intermedia debido a los stalls de memoria.

El acelerador de la FFT utiliza la memoria on-chip para el buffer de los datos de entrada y salida, de esta manera la parte de cálculo del acelerador puede operar independientemente de la memoria SDRAM. Para lograr la plena concurrencia, la fase de transformación de la FFT debe ser capaz de leer y escribir al mismo tiempo. Esto se obtiene con un doble buffer, también conocido como el almacenamiento en buffer de ping-pong, permite un puerto maestro en el acelerador de FFT para leer los datos de entrada de un buffer, mientras que otro puerto maestro escribe los resultados en otro buffer. Este

tipo de almacenamiento intermedio de datos evita los stalls de memoria.

La figura 4 ilustra en una forma simplificada un esquema de doble buffer utilizado en la FFT optimizada, en esta implementación se requiere un total de 4 buffers. Esto se debe a que la FFT realiza cálculos de los datos de entrada real e imaginario simultáneamente y que son independiente de otros, por lo tanto el acelerador puede realizarlo concurrentemente.

En cuanto a los periféricos que se utilizaron, el divisor, multiplicador, sumador y restador, están diseñados para números complejos, y se comunican directamente al procesador ARM por medio del bus de comunicación Avalon, aunque también se puede configurar para que los periféricos se comuniquen con el procesador por medio de la memoria.

V. Resultados

Para realizar una comparación en el desempeño del algoritmo GCC-PHAT, se ejecutan 1000 iteraciones donde cada iteración ejecuta 256 puntos de radix 2 de datos, en un sistema con 6 micrófonos omni-direccionales de entrada de 24 bits de datos y con una frecuencia de muestreo a 48 KHz y una trama de 256 datos. Esta testbench se ejecuto tanto en la versión software como en la que tiene los aceleradores de hardware obteniendo los datos que se muestran en el Cuadro II.

Cuadro II
Carga del GCC-PHAT in operaciones de Multiplicador-Acumulador

Diseño	% FFT en SW	Tiempo (seg)
Software	94,3	87767457
SW + HW	5,67	5271886

La columna de porcentaje de la FFT en el software, indica la utilización de la misma en el GCC-PHAT, como se aprecia en implementación software la FFT es predominante, pero cuando es implementada por aceleradores de hardware, se reduce considerablemente, porque todo el cálculo se realiza por hardware, y el software solo realiza el control del acelerador.

VI. Conclusión

En este trabajo se exponen las ventajas de utilizar aceleradores de hardware para optimizar el funcionamiento de programas ejecutados en un ARM, como primer medida es necesario utilizar herramientas que determinen cuales operaciones son mas óptimas a ser reemplazadas por aceleradores de hardware. Debido a que el ARM ejecuta una versión de Linux se propone utilizar herramientas de profiler tales como Valgrind, gprof u oprofile, en vez de calcular teniendo en cuanta los MA. El utilizar aceleradores de hardware no es una operación transparente, es necesario modificar el código original, para que no haya dependencia de datos, así de esa forma lograr una plena concurrencia. En trabajos futuros nos enfocaremos en desarrollar una metodología que permita incorporar en forma dinámica los aceleradores de hardware desde el análisis del código hasta las forma de conexionado con el procesador.

Referencias

[1] W. Wang, M. Bolic, and J. Parri, "pvfpga: Accessing an fpga-based hardware accelerator in a paravirtualized environment." in *CODES+ISSS*. IEEE, 2013, pp. 1–9. [Online]. Available: http://dblp.uni-trier.de/db/conf/codes/codes2013.html

[2] M. Genovese and E. Napoli, "Fpga-based architecture for real time segmentation and denoising of hd video," *J. Real-Time Image Process.*, vol. 8, no. 4, pp. 389–401, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1007/s11554-011-0238-1

[3] K. Sridharan and T. K. Priya, "The design of a hardware accelerator for real-time complete visibility graph construction and efficient FPGA implementation," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 4, pp. 1185–1187, 2005. [Online]. Available: http://dx.doi.org/10.1109/TIE.2005.851591

[4] Q. Li, M. Zhu, and W. Li, "A portable usb-based microphone array device for robust speech recognition." in *ICASSP*. IEEE, 2009, pp. 1301–1304. [Online]. Available: http://dblp.uni-trier.de/db/conf/icassp/icassp2009.html

[5] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering." *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 216–228, 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/ras/ras55.html

[6] E. Zwyssig, M. Lincoln, and S. Renals, "A digital microphone array for distant speech recognition." in *ICASSP*. IEEE, 2010, pp. 5106–5109. [Online]. Available: http://dblp.uni-trier.de/db/conf/icassp/icassp2010.html

[7] J. Benesty, J. Chen, Y. Huang, and J. Dmochowski, "On microphone-array beamforming from a mimo acoustic signal processing perspective." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1053–1065, 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/taslp/taslp15.html

[8] S. Fischer and K. U. Simmer, "Beamforming microphone arrays for speech acquisition in noisy environments," *Speech Communication*, vol. 20, no. 3-4, pp. 215–227, 1996. [Online]. Available: http://dx.doi.org/10.1016/S0167-6393(96)00054-4

[9] C. Segura and J. Hernando, "Gcc-phat based head orientation estimation." in *INTERSPEECH*. ISCA, 2012. [Online]. Available: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2012.html

[10] C. Segura, A. Abad, J. Hernando, and C. Nadeu, "Speaker orientation estimation based on hybridation of gcc-phat and hlbr." in *INTERSPEECH*. ISCA, 2008, pp. 1325–1328. [Online]. Available: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2008.html

[11] *Zynq-7000 All Programmable SoC - Technical Reference Manual*, Xilinx, 2013.

[12] *AlteraSoC: Booting and Configuration*, Altera, 2012.

[13] *AMBA AXI Protocol Specification*, ARM, 2003.

[14] F. A. Escobar, X. Chang, C. Ibala, and C. Valderrama, "Accuracy study of a real-time hybrid sound source localization algorithm." in *INTETAIN*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, M. Mancas, N. D'Alessandro, X. Siebert, B. Gosselin, C. Valderrama, and T. Dutoit, Eds., vol. 124. Springer, 2013, pp. 146–155. [Online]. Available: http://dblp.uni-trier.de/db/conf/intetain/intetain2013.html

[15] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

Transformación y obtención de Modelos Conceptuales mediante Léxico Extendido del Lenguaje y Escenarios

Fernández Taurant, Juan Pablo – Marciszack, Marcelo Martín
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

En éste trabajo se presenta una propuesta para la obtención de modelos conceptuales a partir de un dominio de ejemplo representado en BPMN (Notación de Gestión de Procesos de Negocios). Los elementos del dominio modelados con BPMN serán mapeados a una estructura orientada al cliente denominada "Requirements Baseline" (Línea Base de Requerimientos), que modela el vocabulario del sistema mediante el Léxico Extendido del Lenguaje (LEL), y los escenarios para modelar su comportamiento. A partir de estos elementos, y aplicando una serie de transformaciones, se obtendrán modelos que serán utilizados posteriormente en el proceso de validación de requerimientos de software.

Palabras Clave
Léxico Extendido del Lenguaje, LEL, BPMN, Transformación de modelos, Validación de requerimientos de software.

Introducción

Actualmente existen en el mercado una gran variedad de herramientas y metodologías que permiten la gestión de requerimientos de software a lo largo del ciclo de vida de desarrollo.
Independientemente de la herramienta o metodología utilizada, la creación y mantención de un gran número de artefactos y modelos es realizada por el analista en forma manual, generando con gran frecuencia inconsistencias entre los modelos, impactando en la trazabilidad de los requerimientos.
Es por esto que se plantea en este trabajo, una herramienta que permita al analista gestionar los requerimientos en forma asistida y parcialmente automatizada, por medio de la generación de modelos conceptuales como representaciones de máquinas abstractas, que se obtendrán a partir del modelo de negocio aplicando diferentes transformaciones, y que se utilizarán para hacer controles de validez de los modelos iniciales.

Elementos del Trabajo y Metodología

La propuesta será desarrollada en una serie de etapas. La primera consiste en la obtención del modelo de negocio utilizando BPMN[1], a partir del cual se extraerán las actividades seleccionadas previamente por el analista para "mapearlas" luego, dentro de una metodología orientada al cliente propuesta por Leite, que utiliza una estructura denominada Requirements Baseline[2][3]. Esta metodología utiliza el Léxico extendido del lenguaje (LEL) para modelar el vocabulario del sistema, y los escenarios para representar su comportamiento.
A partir de las actividades mapeadas se generarán entradas de LEL y escenarios que contendrán las descripciones del comportamiento del sistema. Cada escenario deberá ser completado por el analista. Para esta etapa se utilizará Baseline Mentor Workbench[4] (BMW) como herramienta de soporte para la gestión de entradas de LEL y la generación de escenarios.
Finalmente, la tercera etapa consiste en la generación de nuevos archivos con un formato XML específico, uno por cada escenario, que serán utilizados por simuladores de máquinas abstractas para validar los modelos generados. Este proceso será soportado por la herramienta BMW agregando la funcionalidad necesaria

para la obtención de los de esquemas de las máquinas abstractas en formato XML a partir de los escenarios definidos.

La estructura de la propuesta puede observarse en la figura 1.

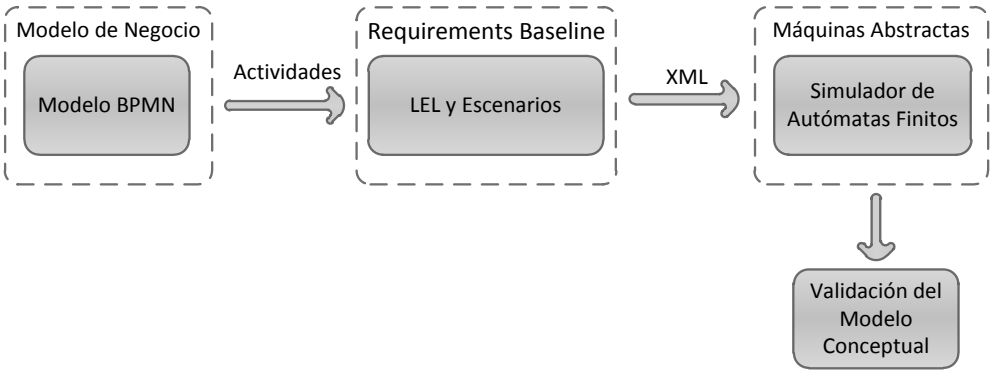


Figura 1

- ***Aplicación de la metodología***

Para el análisis de esta propuesta y su aplicación sobre un caso práctico se utilizará un dominio de ejemplo modelado en BPMN y se tomará el proceso indicado en la figura 2. Todas las actividades serán mapeadas a la Requirements Baseline a excepción de las seleccionadas por el analista con el estereotipo "Manual", indicando que la actividad se espera que será realizada sin la ayuda de algún motor de ejecución de proceso de negocio o alguna aplicación.

También serán mapeados los "*Pools*" o "*Lanes*" (utilizados como mecanismos de organización de actividades en categorías visuales separadas para indicar las diferentes áreas funcionales o responsables) según sea el caso que corresponda al último nivel en la jerarquía de roles como responsable de la realización de un determinado grupo de actividades.

Cada *"Pool"* o *"Lane"*, debe ser introducida dentro del LEL como símbolo. Cada símbolo se describe mediante una noción y un impacto. La noción indica el

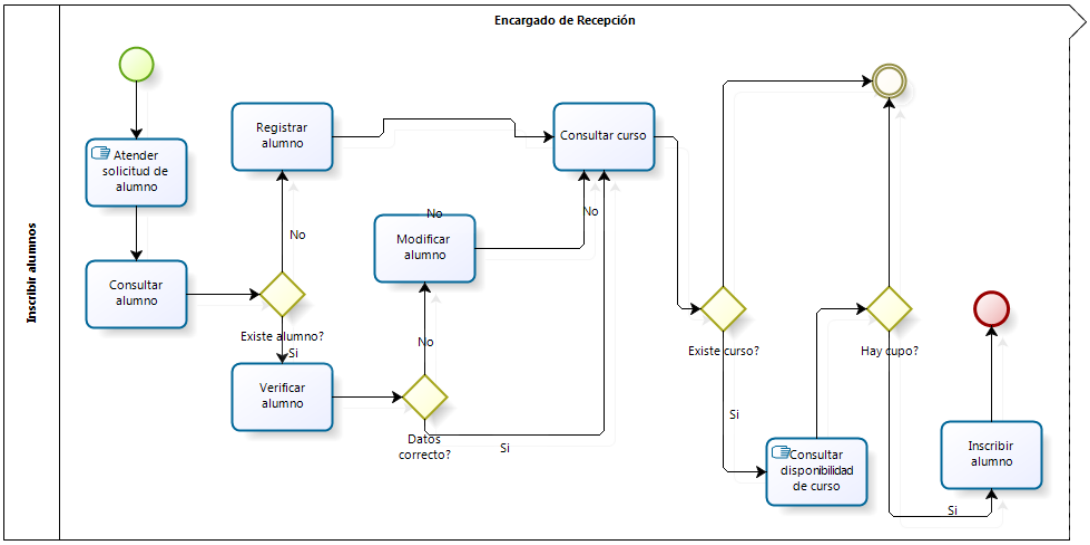


Figura 2

Entrada de LEL – Sinónimos	Identificador del símbolo
Noción	Denota el significado del símbolo
Impacto	Connotación o repercusión en el sistema

Tabla 1 – Definición de una entrada de LEL

significado del símbolo, y el impacto las repercusiones que ese símbolo tendrá en el sistema, como lo indica la tabla 1.
La descripción de los símbolos debe hacerse de manera que se cumplan dos reglas en forma simultánea [5][6]:

• *Principio de circularidad*: acotando el lenguaje en función del dominio mediante la maximización de símbolos del lenguaje del LEL, que se logra utilizando en las definiciones de noción e impacto símbolos ya descriptos dentro del LEL.

Sujeto	*Nociones*: describen quién es el sujeto.
	Impactos: registran acciones ejecutadas por el sujeto.
Objeto	*Nociones*: definen al objeto e identifica a otros términos con los cuales el objeto tiene algún tipo de relación.
	Impactos: describen las acciones que pueden ser aplicadas al objeto.
Verbo	*Nociones*: describen quién ejecuta la acción, cuándo ocurre, y cuáles son los procedimientos involucrados.
	Impactos: describen las restricciones sobre la acción, cuáles son las acciones desencadenadas en el ambiente y las nuevas situaciones que aparecen como resultado de la acción.
Estado	*Nociones*: describen que significa y que acciones pueden desencadenarse como consecuencia de ese estado.
	Impactos: describen otras situaciones y acciones relacionadas.

Tabla 2 – Heurísticas para la definición de los símbolos. [6][7]

• *Principio del vocabulario mínimo*: en donde la tarea es minimizar el uso de símbolos externos al dominio de la aplicación.

Los símbolos deben también clasificarse agrupándolos según sujeto, verbo, objeto o estado. La clasificación se utiliza para mantener homogeneidad en las descripciones [6]. La Tabla 2 indica la heurística a utilizar para la definición de los símbolos según su clasificación.
Bajo las premisas anteriores, cada "*Pool / Lane"* puede ser introducido al LEL en forma de sujeto, ya que en BPMN representan un rol que se encarga de realizar ciertas actividades. Esto equivale a decir en LEL que un *"Pool / Lane*" es un sujeto que ejecuta diferentes acciones, esas acciones corresponden a las actividades que deben ser agregadas como impactos en la definición del sujeto. Los impactos definidos como acciones en los sujetos serán los escenarios candidatos.
Por otra parte, no hay una restricción de uno a uno en cuanto a la relación Impacto-Actividad. El analista podrá agregar nuevos impactos como escenarios candidatos sin necesidad mantener esa relación.
Los escenarios guardan ciertas similitudes con los Use Cases de UML[8]. Se utilizan para describir la acción que desarrolla el sujeto especificando el actor o encargado de realizar la acción, las precondiciones de la acción, los sujetos y objetos intervinientes, un objetivo a cumplir y la descripción de cada episodio utilizando la información que se encuentra contenida en el LEL. Para trabajar con LEL y escenarios utilizaremos la herramienta Baseline Mentor Workbench (BMW). La figura 3 muestra la definición del símbolo correspondiente al proceso BPMN de ejemplo.

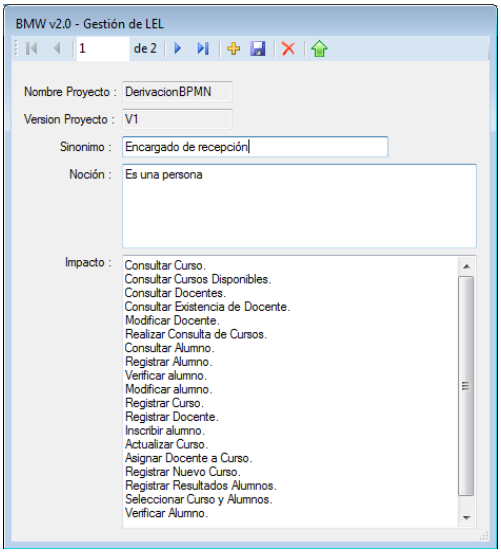


Figura 3

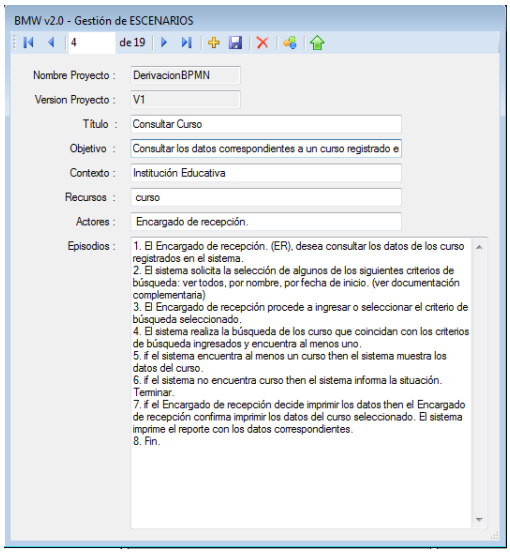


Figura 4

En la figura 4 se muestra el proceso para la descripción del escenario "Consultar Curso" obtenido a partir del impacto del símbolo de LEL del ejemplo. La descripción de cada escenario la deberá realizar el analista siguiendo las pautas de la vista de escenarios propuesta por la Requirements Baseline, con algunas excepciones necesarias para la obtención de la máquina abstracta que se utilizará para validar el escenario. Por un lado se deberá agregar un episodio final cuya descripción se indicará con la palabra "Fin", que se utilizará para detectar el estado final de aceptación de la máquina abstracta. Se utilizará también la palabra "Terminar" al final de la descripción del escenario para los casos en que la secuencia de acciones o de ejecución de episodios se vea interrumpida por una excepción que obligue a finalizar la ejecución del escenario en un estado no deseado, vale decir, en un estado de no aceptación en la máquina abstracta.

Luego de realizar la descripción de los escenarios se pasa a la tercera etapa, donde se genera un archivo XML por cada escenario.

El mecanismo de generación de estos archivos será agregado como funcionalidad extra a la herramienta BMW. Cada archivo XML contiene todas las definiciones de la máquina abstracta correspondiente a la descripción de la secuencia de los episodios del escenario, y que será utilizada por simuladores de autómatas para verificar su validez. Cada episodio estará representado por un estado. La transición entre estados quedará determinada por la secuencia de ejecución de episodios, representada con la letra "S". Las estructuras condicionales de los episodios serán también representadas con bifurcaciones en las transiciones de estados, que se representaran con las letras "S" y "N". Para los episodios que contengan la palabra "Terminar" se agregarán transiciones a estados finales de no aceptación. Asimismo, para el episodio "Fin" se agregará una transición a un estado final de aceptación.

La figura 5 muestra la máquina abstracta correspondiente al escenario de la figura 4.

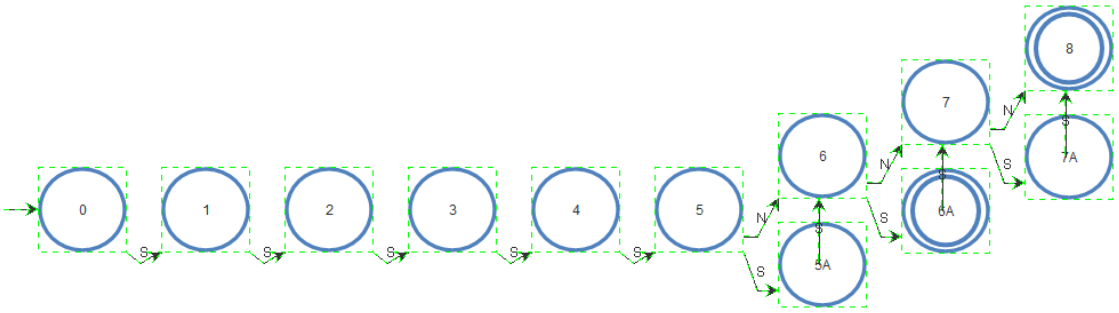


Figura 5

Resultados

Fue posible introducir como símbolo de LEL dentro de la categoría "*Sujeto*" a cada *"Pool / Lane"* que agrupe un conjunto de actividades, y al grupo de actividad como acciones que realiza ese sujeto en forma de impactos con cierto grado de pérdida de información en el pasaje entre metodologías. También es posible integrar al Léxico extendido del Lenguaje la estructura o jerarquía de roles definidos por los *"Pools"* y *"Lanes"* de BPMN.

Discusión

En función de los resultados obtenidos se evidencia la posibilidad de generar modelos que sean de utilidad para el proceso de validación de requerimientos de software mediante el uso de diferentes metodologías, aplicando transformaciones intermedias como mecanismos de traducción entre ellas.
Este trabajo se presenta además como una alternativa al trabajo "Transformación de Modelos en el Proceso de Obtención de Modelos Conceptuales Partiendo de BPMN"[9] que se encuentra dentro de la misma línea de investigación, en el que también se evidencia la posibilidad de transformar modelos de diferentes metodologías con cierta pérdida de información.
En un futuro se seguirá optimizando el proceso de obtención de modelos conceptuales con el objetivo de reducir la pérdida de información asociada a la transformación del modelo BPMN. Esta pérdida se debe a que sólo se tienen en cuenta *"Pools"*, *"Lanes"* y actividades. No se tienen en cuenta estereotipos de actividades, relaciones entre actividades y otros elementos de BPMN.
Adicionalmente se buscará incrementar el nivel de automatización del proceso tratando de evitar en lo posible, la subjetividad introducida por el analista en el modelado.
Los modelos resultantes podrán a su vez, utilizarse para validar los modelos propuestos inicialmente.

Conclusión

Si bien este proceso requiere de la intervención del analista, es posible generar modelos más coherentes y en forma asistida durante el proceso de captura de requerimientos, mejorando así su trazabilidad. Los modelos obtenidos serán representaciones de máquinas abstractas que se utilizarán para validar los modelos iniciales. Los resultados de este trabajo servirán además como elemento de comparación a los obtenidos en trabajos anteriores como "Transformación de Modelos en el Proceso de Obtención de Modelos Conceptuales Partiendo de BPMN". Este trabajo se integra también en el marco de desarrollo del proyecto "Validación de Requerimientos a través de Modelos Conceptuales".

Referencias

[1] Object Management Group. Business Process Modeling Notation (BPMN).
[2][Leite'97] Leite J.C.S.P., Rossi G.,et al. Enhancing a Requirements Baseline with Scenarios. Proceedings of RE 97': International Symposium on Requeriments Engineering, IEEE. Enero 1997.
[3][Leite95] Leite J.C.S.P, Albuquerque Oliveira, A P. A Client Oriented Requirements Baseline. Proceedings of RE 95':
Second IEEE International Symposium on Requeriments Engineering. Inglaterra, Marzo 1995.
[4] Leandro Antonelli "Traceability en la Elicitación de Requerimientos". http://hdl.handle.net/10915/4061
[5][Leite'93] Leite, J.C.S.P., "Eliciting Requirements Using a Natural Language Based Approach: The Case of the Meeting Scheduler Problem", March 1993.
[6][Hadad'96] Hadad,G., Kaplan, G., Maiorana, V., Balaguer, F., Oliveros, A., Leite, J.C.S.P., Rossi, G. Informe Técnico: "Léxico Extendido del Lenguaje y Escenarios del Sistema Nacional para la Obtención de Pasaportes". Proyecto de Investigación, Departamento de Investigación, Universidad de Belgrano, Buenos Aires, 1996.
[7][Leonardi 2001] C. Leonardi, J.C.S. Leite, Gustavo Rossi. " Una estrategia de Modelado Conceptual de Objetos , basada en Modelos de requisitos en lenguaje natural ".Tesis de Maestría Universidad Nacional de la Plata. http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Leonardi.pdf.
[8] OMG. Unified Modelling Language: Superstructure Version 2.0 (online), Julio 2005, http://www.omg.org
[9] Juan Pablo Fernández Taurant, Marcelo Marciszack."Transformación de Modelos en el Proceso de Obtención de Modelos Conceptuales Partiendo de BPMN" http://conaiisi.unsl.edu.ar/2013/141-526-1-DR.pdf

Datos de Contacto

Juan Pablo Fernández Taurant
jtaurant@gmail.com
Universidad Tecnológica Nacional – Facultad Regional Córdoba.
Maestro M. López esquina Cruz Roja Argentina.
X5016ZAA – Córdoba
República Argentina

Marcelo Martín Marciszack
marciszack@gmail.com
Universidad Tecnológica Nacional – Facultad Regional Córdoba.
Maestro M. López esquina Cruz Roja Argentina.
X5016ZAA – Córdoba
República Argentina

Integrando SCRUM y CMMI en PyMES.

Sosa Zitto, Rossana; Pralong, Lourdes; Blanc, Rafael; Álvarez, Claudia.
Universidad Autónoma de Entre Ríos, Facultad de Ciencia y Tecnología,
Sede Concepción del Uruguay.

Abstract

En Argentina la Industria de Desarrollo de Software está formada en gran parte por empresas pequeñas y medianas (Pymes), las cuales constituyen un eslabón fundamental en el sector productivo del país [1].

Para estas empresas la calidad del software es esencial, siendo la mejora de procesos de software una actividad que desean implementar con el objetivo de incrementar la calidad y capacidad de sus procesos y, en consecuencia, la calidad de sus productos y servicios. [2]. Sin embargo, su estructura y la necesidad de inversión en dinero, tiempo y recursos es un impedimento para acceder a certificaciones de calidad [3].Las pymes necesitan prácticas de ingeniería de software adaptadas a su tamaño y tipo de negocio. [2].

El objetivo del presente trabajo es analizar las prácticas de la metodología Scrum y su correspondencia con las dimensiones conceptuales de SEI CMMI. El estudio de casos se realizó en la ciudad de Concepción del Uruguay, Entre Ríos.

Para evaluar la aplicación de las prácticas Scrum que aplican la empresas se utilizó una escala likert de conformidad y se realizó un mapeo para cada empresa objeto de estudio. Por último se evaluó el grado de cumplimiento de cada práctica según la escala anterior.

Palabras Claves

Desarrollo de Software - Metodologías ágiles - SEI-CMMI – Modelos de calidad de software.

1. Introducción

La línea de investigación presentada se desarrolla en el marco del Proyecto de Investigación Anual (PIDA) denominado “Implementación de metodologías ágiles y su impacto en la calidad de software”, acreditado por la Secretaría de Ciencia y Técnica de la Universidad Autónoma de Entre Ríos (UADER) y perteneciente a un equipo de docentes investigadores de la Facultad de Ciencia y Tecnología Sede Concepción del Uruguay de la misma universidad.

Calidad en las pymes

La calidad es un factor competitivo y de diferenciación clave en las Pymes. Dado que la calidad del producto de software desarrollado está estrechamente relacionada con la calidad del proceso utilizado, las Pymes necesitan mejorar sus procesos de desarrollo, para así incrementar la calidad de sus productos. [1]

Este sector se caracteriza por su tamaño, por el volumen de venta anual y por contar con mano de obra calificada; ésta característica le permite disponer de una estructura adaptativa y flexible a los proyectos de software que desarrollen. Su mayor capital es el capital humano. [4].

2. Metodologías Ágiles

Con objeto de gestionar proyectos de una manera eficiente han surgido las metodologías agiles para el desarrollo de software como herramientas que permiten mejorar los procesos productivos. [5]

Para desarrollar sus productos, las metodologías ágiles ofrecen a las PyMEs la posibilidad de contar con procesos livianos y simples, los cuales pueden adaptarse a su estructura o la estructura al proceso.

Las metodologías ágiles tienen algunos cambios significativos en comparación con las tradicionales. Entre los cuales podemos destacar:

- Son adaptativos y no predictivos.
- Están orientados a las personas y no orientados a los procesos.
- Los métodos ágiles son menos orientados a la documentación.
- Los métodos ágiles consideran la participación del cliente durante el desarrollo como esencial. [6] [7]

3. Scrum

Scrum es un marco de trabajo compuesto por un conjunto de prácticas que puede ser usado para administrar y controlar proyectos complejos. [8] y realizar un desarrollo ágil de proyectos.

En principio surgido en la industria del software, pero de suficiente sencillez y flexibilidad como para ser aplicado en contextos muy diversos. Considera al proceso de desarrollo de un proyecto como una sucesión de ciclos cortos de trabajo denominados Sprints, obteniendo de cada uno de ellos un producto funcional que va completándose en forma iterativa.

Esta forma de concebir la gestión de proyectos es radicalmente diferente al modo secuencial en que se afrontan tradicionalmente los mismos, dividiéndolos en etapas, sucesivas y especializadas, planificadas a priori en forma detallada; proponiendo un desarrollo en forma iterativa sobre una versión completa del producto, centrada en el valor para el cliente.

Scrum trabaja en base a roles, uno de los roles clave definidos en Scrum es el denominado propietario del

producto o product owner, que participa activamente en el proceso de desarrollo, facilitando la comprensión por parte del equipo de los aspectos prioritarios y centrales del resultado esperado. Es quien representa al cliente, con una fuerte y continua interacción con el equipo, facilita desde el inicio la clara percepción de la visión del producto y de los aspectos que se consideran de valor sustancial en el mismo.

El Scrum Master o facilitador, es el rol que es responsable de orientar al equipo en la aplicación de las prácticas adecuadas para lograr los beneficios esperados de esta modalidad de gestión, al mismo tiempo que se encarga de remover impedimentos y reducir las fricciones que la dinámica de trabajo pueda producir.

El Development Team es el equipo de trabajo responsable del desarrollo del producto a entregar, para esto trabaja en conjunto con los distintos stakeholders desde la definición de los requerimientos (Product Backlog) hasta la entrega del producto. El tamaño del equipo es una cuestión fundamental para los procesos ágiles, siendo lo común equipos entre 3 y 9 desarrolladores por equipo. Con más de 9 personas la cantidad de relaciones entre los integrantes aumenta exponencialmente y esto es caótico para la comunicación del equipo [9].

Scrum no requiere o provee ninguna práctica específica, dispone de prácticas y herramientas para la gestión de sus diferentes fases. Las principales prácticas son:

- Product Backlog: define los requerimientos del sistema o el trabajo a realizar a lo largo del proyecto. Lo mantiene el propietario del producto.
- Sprint Backlog: es un listado de tareas, personas, tiempos y recursos que el equipo se compromete a realizar para generar el incremento previsto.
- Estimación de esfuerzos: es un proceso iterativo donde las estimaciones del product backlog son ajustadas de acuerdo a la última iteración. Lo hace el equipo de desarrollo y el propietario del producto.
- Gráfico Burn-down: es una herramienta para gestionar y seguir el trabajo de cada sprint y representa gráficamente el avance del mismo.
- Gráfico Burn-up: es una herramienta que permite al propietario del producto gestionar y seguir las versiones previstas del producto, funcionalidades, velocidad estimada, fechas probables de cada versión, margen de error previsto y avance real.
- Planning Poker: es un juego para ayudar al equipo a definir una estimación de tareas en la reunión inicial del sprint. [10]

Además se espera de cada iteración o sprint, un entregable denominado incremento que es considerado un producto potencialmente completo en su totalidad y listo para su utilización. Esto promueve una gran transparencia respecto a los problemas que se interponen en su desarrollo, así como sobre las capacidades del equipo para lograrlo.

Este ciclo corto de trabajo también exige al equipo un alto grado de interacción para concretar el objetivo, favoreciendo que surjan tempranamente las dificultades o impedimentos que serán afrontados diariamente a través de reuniones de seguimiento, breves.

Estas reuniones no sólo apuntan a realizar un control del avance sino también a coordinar esfuerzos para superar obstáculos, compartir estrategias y técnicas para afrontar situaciones y mejorar la cohesión del equipo de trabajo.

Es así que cada sprint se desarrolla en tres fases: una reunión de planificación, un período de trabajo a lo largo del cual se realizan las reuniones diarias de seguimiento, y una reunión de revisión del producto desarrollado en el sprint, denominado incremento, seguido de una reunión de evaluación del proceso de trabajo con miras a mejorarlo en forma continua, denominada retrospectiva.

De acuerdo a un estudio realizado en el año 2011, esta metodología ágil es la más utilizada en Argentina [11].

4. CMMI - DEV Versión 1.3

El modelo propuesto por CMMI para desarrollo utiliza conceptos de gestión e ingeniería para ayudar en la entrega de productos a tiempo y de alta calidad, especialmente para aquellas organizaciones que dependen fuertemente del desarrollo de software.

Este modelo cubre el ciclo de vida de productos y servicios desde la concepción de estos hasta la entrega y mantenimiento.

Con la adopción de este modelo como guía de referencia, las organizaciones pueden lograr:

- Mejorar la satisfacción del cliente.
- Aumentar la calidad de los productos desarrollados.
- Realizar las entregas en los tiempos acordados.
- Minimizar los costos de desarrollo.
- Garantizar un retorno de inversión.
- Mejorar las condiciones de trabajo de los empleados. [6].

CMMI-DEV contiene 22 áreas de proceso. De esas áreas de proceso, 16 son áreas de proceso base, 1 es un área de proceso compartida y 5 son áreas de proceso específicas de desarrollo.

Todas las prácticas del modelo CMMI-DEV se centran en las actividades de la organización desarrolladora. Cinco áreas de proceso se centran en las prácticas específicas del desarrollo: tratando desarrollo de requisitos, solución técnica, integración del producto, verificación y validación.

CMMI para Desarrollo es un modelo de referencia que cubre las actividades para desarrollar tanto productos como servicios.

CMMI da soporte a dos caminos de mejora usando niveles. Un camino permite a las organizaciones mejorar de forma incremental los procesos que corresponden a un área de proceso individual (o grupo de áreas de proceso) seleccionada por la organización. El otro camino permite a las organizaciones mejorar un conjunto de procesos relacionados tratando, de forma incremental, conjuntos sucesivos de áreas de proceso.

Estos dos caminos de mejora están asociados con los dos tipos de niveles: niveles de capacidad y niveles de madurez. Estos niveles corresponden a las dos aproximaciones de mejora de procesos denominadas "representaciones". Las dos representaciones se denominan "continua" y "por etapas." El uso de la representación continua permite alcanzar "niveles de capacidad". El uso de la representación por etapas permite alcanzar "niveles de madurez".

Para alcanzar un nivel particular, una organización debe satisfacer todas las metas del área de proceso o del conjunto de áreas de proceso que son objeto de la mejora independientemente de si es un nivel de capacidad o de madurez.

Ambas representaciones proporcionan caminos para mejorar sus procesos con el fin de lograr los objetivos de negocio, proporcionan el mismo contenido esencial y utilizan los mismos componentes del modelo. [12]

5. Caracterización de las pymes de desarrollo de software

A mediados de la década pasada en la provincia de Entre Ríos, se comenzaron a instalar oficinas de desarrollo de empresas cuya casa central se encuentra en las principales ciudades del país.

El mercado local está formado por pymes, cuyas iniciativas de mejora de procesos debe ser diferente, ya que sus características, son diferentes, con lo cual se necesita un enfoque que se ajuste a ellas [3], [4].

En el presente estudio se relevaron empresas de desarrollo de software de más de dos empleados. Se tomó una muestra representativa que conformaba el 80% del total de empresas instaladas en el departamento. Las mismas poseen las siguientes características.

Tamaño de la empresa: En cuanto a la cantidad de empleados que las mismas poseen en las oficinas locales, del total de las empresas relevadas un 63 % poseen entre 5 y 10 empleados, un 25% posee entre 11 y 20 y un 12 % restante posee menos de 5 empleados.

Ley de Software: respecto a la adhesión a la ley de promoción de software el 63% de las empresas relevadas declaró estar adherida.

Calidad: En cuanto a las conductas referentes a la calidad, el 50% está certificada por normas ISO, no se observaron empresas que adhieran a CMMI.

Mercado: En relación al destino de los productos desarrollados el 71% es a comercio y el 29% a industria. La totalidad de las empresas relevadas desarrollan un producto específico el cual ofrece a diferentes clientes con adaptaciones de acuerdo a las necesidades. Tan solo el 10% realiza software a medida. El 29% de la muestra declaró exportar algún producto de los que produce.

Metodologías de desarrollo: En ninguno de los casos se detectó que una empresa aplique una metodología de desarrollo pura, las empresas relevadas siguen el patrón nacional en cuanto a aplicación de metodologías ágiles, correspondiéndose en su gran mayoría sus prácticas, a prácticas aisladas de Scrum.

6. Experiencia

La investigación que se desarrolla en este trabajo surge como resultado de un análisis pormenorizado de las empresas de desarrollo de software de la ciudad de Concepción del Uruguay, provincia de Entre Ríos.

En el presente trabajo se analiza cuales prácticas se han adoptado de la metodología Scrum y cuales dimensiones conceptuales de CMMI se cumplimentan. Para ello se elabora un mapeo entre ambas y un posterior análisis. Diversos estudios se han realizado al respecto, [12], [15], [16], [17], [18], [19], [20], [21], [22], [23] y sirven de base para el presente ensayo. El análisis se realizó sobre las siguientes dimensiones conceptuales del modelo CMMI: Planificación del proyecto, Gestión de Requisitos, Gestión de Riesgos, Monitorización y control del proyecto, Gestión de la configuración, Medición y Análisis y Verificación.

Se realizó un mapeo para cada dimensión seleccionada y se determina como se implementa a través de prácticas de Scrum en la empresa. A continuación se describe cada una de las dimensiones con sus objetivos más destacables.

Planificación del proyecto: contempla establecer y mantener el presupuesto y el calendario. Identifica los riesgos del proyecto. Es responsable de establecer el plan para la gestión de los datos y del proyecto.

Gestión de Requisitos: Involucra el obtener y entender los requisitos, gestionar sus cambios, mantener la trazabilidad bidireccional e identificar inconsistencias entre el avance del proyecto y los requisitos.

Gestión de Riesgos: básicamente implica identificar los problemas posibles antes de que ocurran.

Monitorización y control del proyecto: involucra las actividades de monitorear los parámetros de planificación del proyecto, monitorear los compromisos, los riesgos del proyecto, la gestión de datos y la participación de los stakeholders.

Gestión de la configuración: consiste en el control de versiones, la gestión de cambios, el control de liberaciones y la gestión de la construcción.

Medición y Análisis: hace referencia al desarrollo de la capacidad de medición, la cual es usada para dar soporte a las actividades de administración.

Verificación: Verifica que el módulo del producto final se desarrolle de acuerdo a los requerimientos especificados.

A continuación se identifican las principales prácticas de Scrum que cubren cada una de las dimensiones anteriormente descriptas.

1. **Monitorización y Control de Proyecto**
   - Reuniones Diarias
   - Reuniones de Revisión
   - Reuniones Retrospectivas
   - Scrum Master
   - Gráficos Burndown y Burnup

2. **Gestión de Requisitos**
   - Backlog del Producto
   - Backlog del Sprint
   - Planificación del Sprint
   - Reunión de Revisión
   - Reuniones retrospectivas
   - Propietarios del Producto
   - Cliente in-situ

3. **Gestión de la Configuración**
   - Reuniones Diarias

4. **Gestión de Riesgos**
   - Reuniones Diarias
   - Revisión de Sprint
   - Scrum Master
   - Propietarios del producto

5. **Medición y Análisis**
   - Reuniones Diarias
   - Reuniones retrospectivas
   - Reunión de Revisión
   - Planificación del Sprint
   - Scrum Master
   - Propietarios del Producto

6. **Verificación**
   - Backlog del Sprint
   - Historias de usuario
   - Stakeholders

7. **Planificación de Proyecto**
   - Planificación del Sprint
   - Backlog del Producto
   - Backlog del Sprint
   - Fases del ciclo de vida Scrum
   - Gráficos Burndown y Burnup
   - Históricos de backlogs

En la etapa de Relevamiento se realizaron entrevistas en profundidad a gerentes y líderes de proyectos.

A partir de los resultados obtenidos se diseñó la Tabla 1, donde se puede observar en qué porcentaje cumplimentan las empresas cada dimensión.

Tabla 1: Grado de cumplimiento por dimensión.

Dimensiones	E1	E2	E3	E4	E5	E6	E7	E8
Planificación del proyecto	50%	50%	75%	25%	50%	25%	75%	75%
Gestión de Requisitos	75%	50%	75%	25%	75%	50%	75%	75%
Gestión de riesgos	50%	50%	50%	0%	50%	25%	75%	75%
Monitorización y Control del Proyecto	50%	50%	50%	25%	50%	25%	75%	75%
Gestión de la configuración	50%	50%	50%	50%	50%	25%	75%	75%
Medición y Análisis	50%	50%	50%	25%	50%	50%	75%	75%
Verificación	75%	75%	75%	75%	50%	50%	75%	75%

Para evaluar las prácticas Scrum que aplican la empresas se utilizó una escala likert de conformidad del 1 al 5, como se puede observar en la tabla 2, siendo 1 equivalente a 0%, 2 al 25%, 3 al 50%, 4 al 75% y 5 al 100% y se realizó un mapeo entre las dimensiones CMMI y las prácticas Scrum para las empresas objeto de estudio.

Por último se evaluó el grado de cumplimiento de cada práctica según la escala anterior.

Tabla 2: Grado de cumplimiento de las empresas según escala Likert.

Dimensiones	5 (100%)	4 (75%)	3 (50%)	2 (25%)	1 (0%)
Planificación del proyecto	0,0%	37,5%	37,5%	25,0%	0,0%
Gestión de Requisitos	0,0%	62,5%	25,0%	12,5%	0,0%
Gestión de riesgos	0,0%	25,0%	50,0%	25,0%	0,0%
Monitorización y Control del Proyecto	0,0%	25,0%	50,0%	25,0%	0,0%
Gestión de la configuración	0,0%	25,0%	62,5%	12,5%	0,0%
Medición y Análisis	0,0%	25,0%	62,5%	12,5%	0,0%
Verificación	0,0%	75,0%	25,0%	0,0%	0,0%

7. Resultados

Del análisis de las empresas relevadas se concluye que ninguna de las empresas aplica metodologías ágiles puras pero si prácticas aisladas de Scrum. Las prácticas ágiles más utilizadas son las reuniones diarias, el 62,5% de las empresas relevadas las utilizan. La principal razón para implantar la agilidad es incrementar el time-to-market, el 100% de las empresas que utilizan prácticas ágiles manifiestan que este es el motivo por el cual las aplican.

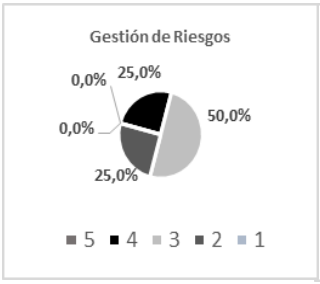


Fig. 1: Gestión de riesgos

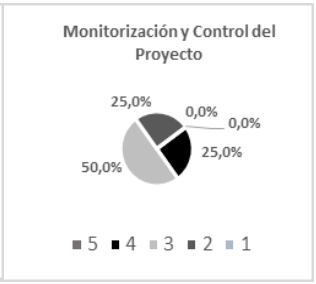


Fig. 2: Monitorización y control del proyecto

De los gráficos anteriores se observa que para las dimensiones Monitorización y Control del Proyecto y para la dimensión Gestión de Riesgos el 25% de las empresas adhiere en un 75% a las prácticas, igual porcentaje adhiere en un 25% y un 50% adhiere 50% a las prácticas.

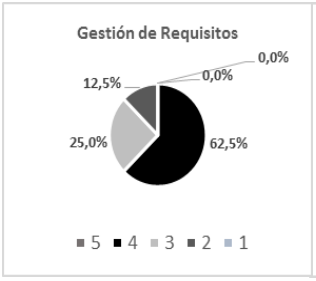


Fig. 3: Gestión de Requisitos

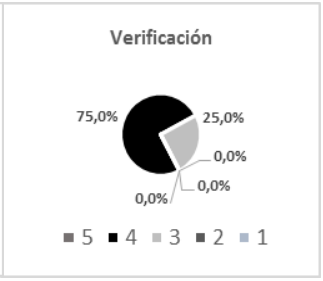


Fig. 4: Verificación

En la Gestión de Requisitos un 62,5% de las empresas adhiere en un 75%.
Para la Verificación se puede observar que el 75% de las empresas adhiere en un 75% a esta dimensión

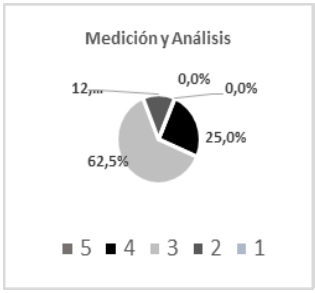


Fig. 5: Medición y análisis

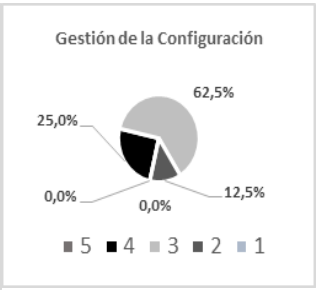


Fig. 6: Gestión de la configuración

En las dimensiones Gestión de la configuración y en Medición y Análisis un 25% de las empresas adhieren en un 75% y un 62,5% lo hace en un 50%.

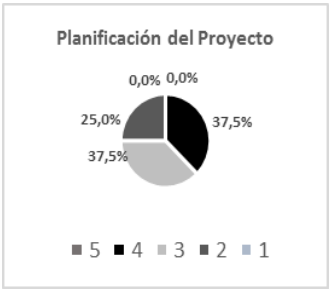


Fig. 7: Planificación del proyecto

Y por último la Planificación del proyecto, en la cual el 37,5% de las empresas tiene una adhesión mayor al 75% y un 37,5% adhiere un 50% a las prácticas Scrum evaluadas, mientras que el 25% restante lo hace en un 25%.

Las dimensiones que se encuentran menos cumplimentadas son Monitorización y control de proyectos y Gestión de Riesgos.
La dimensión que se cumplimenta en un porcentaje más alto es la de Verificación.
Cómo se observa muchas veces las empresas en forma inconsciente al aplicar determinadas prácticas, tales como las de Scrum obtienen niveles de calidad a bajo costo, pero es importante destacar que asegurar y reproducir calidad de forma fiable requiere de procesos bien definidos cuya aplicabilidad posterior debe, además, quedar garantizada. Es necesario tener en cuenta no solo el qué se hace sino también el cómo se hace, para poder mejorar el rendimiento general y para lograr lo anterior Scrum es un marco de trabajo adecuado.
Futuros trabajos podrán ampliar la base de datos sobre la cual se realizó este estudio a fin de poder tener un estado de situación detallado de las empresas en cuanto al cumplimiento de la totalidad de áreas de procesos de CMMI y en cuanto a la aplicación de dicho modelo.

8. Referencias

[1] Martínez, N., Ramón, H., Bertone, R Aplicabilidad de Competisoft a partir de un método ágil como Scrum. Un caso práctico. (2012).

[2] Navarro J., Garzás J. Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías Ágiles y Software Libre. (2010).

[3] Llaneza, M., Dapozo, G., Greiner, C., Estayno, M. Análisis comparativo de modelos de calidad orientado al desarrollo de software en pymes. (2013).

[4] Sosa Zitto, R.; Blanc, R.; Pralong, L.; Alvarez, C.; Galáz, S. El desafío de producir software de calidad aplicando prácticas de CMMI para las pymes de Concepción del Uruguay, Entre Ríos. Conaiisi (2013).

[5] Barrios W., Godoy M. Fernández M., Mariño S. Ferreira F., Zarrabeitia C.; SCRUM: Experiencia de Aplicación en una Empresa de Desarrollo de Software del NEA. (2012).

[6] Paolini, Andrés. Implementación de área de proceso de gestión de riesgos de CMMI v1.3 3 utilizando Metodologías ágiles. (2013).

[7] Taghi Javdani Gandomani1, HazuraZulzalil. Compatibility of Agile Software Development Methods and CMMI. (2013)

[8] Schwaber, Ken. Agile Project Management with Scrum. Redmond: Microsoft Press, (2004).

[9] Nazareno, R.; Leone, H., Gonnet, S. Trazabilidad de procesos ágiles: un modelo para la trazabilidad de procesos Scrum. (2013).

[10] Rivadeneira, Silvia. Metodologías ágiles enfocadas al modelado de requerimientos. (2012.)

[11] Álvarez, M. Escobar F., Nardin, A. Ricci, Aparicio, Bioul, G. Análisis de la implementación de prácticas ágiles en Argentina. (2011).

[12] CMMI® para Desarrollo, Versión 1.3. CMMI-DEV, V1.3. (2010).

[13] Richardson, C.G.V. Wangenheim, Why are small software organizations different?, (2007).

[14] Pantaleo Guillermo, Calidad en el desarrollo de software, Editorial Alfaomega. (2011).

[15] Hanan Bin Liaqat, M. Rizwan Jameel Qureshi and Shahid Bhatti, "An Enhanced Software Project Management Methodology Accompanying Agile and CMMI,". (ICIIT). (2010).

[16] Foegen M. Scrum and CMMI – Does it fit together?. (2010).

[17] Foegen M. Croome D. How Scrum helps with CMMI. (2011).

[18] Samina J., Ali J. SCXTREME Framework: A Customized Approach of Process Improvements in Agile Blend with CMMI Practices in Pakistan. (2013).

[19] Marcal A., de Freitas B., Furtado Soares F., Belchior A., Blending Scrum practices and CMMI Project Management Process Areas. (2008).

[20] Glazer, H.; Dalton, J.; Anderson, D.; Konrad, M.; Shrum, S.; "CMMI® or Agile: Why Not Embrace Both!". (2008).

[21] Glazer, H.: Love and Marriage: CMMI and Agile Need Each Other. (2009).

[22] Potter N., Sakry M. Implementing Scrum (Agile) and CMMI® together. (2009).

[23] Potter N. Adding Practices to Scrum to Achieve your goals. (and comparison with CMMI level 3). (2013).

Diseño de GUI: características deseables para su construcción

M. Claudia Albornoz, Mario Berón, Germán Montejano
Universidad Nacional de San Luis, San Luis, Argentina
{albornoz, mberon, gmonte}@unsl.edu.ar

Resumen

Actualmente, se interactúa constantemente con Interfaces Gráficas de Usuario. Las Interfaces no sólo se encuentran en una computadora, sino también en objetos de uso diario y común; como celulares, cajeros automáticos, electrodomésticos, etc. Es evidente que las Interfaces están presentes cotidianamente en la vida de las personas. Por esto es fundamental que la Interfaz se encuentre bien diseñada. El mal diseño de la Interfaz puede condenar a cualquier aplicación al fracaso; aun cuando la aplicación sea eficiente, cumpla con los requisitos del usuario, contenga todas las tareas que se espera debe hacer, si la Interfaz no puede comunicarse con el usuario o el usuario no logra comprenderla, la aplicación no será exitosa.

Lo expuesto previamente permite entender el rol fundamental que desempeñan las interfaces de usuario en las aplicaciones. Por esta razón, construir las mismas siguiendo criterios claros de diseño es una tarea central en el desarrollo de toda aplicación.

Desafortunadamente, la literatura aborda de manera aislada esta temática. Esta peculiaridad hace que el desarrollador de software no use, o use sin percibir, los criterios troncales de diseño que una interfaz gráfica debería reunir para que cumpla eficientemente su función y sea de calidad.

Este artículo tiene como principal objetivo mostrar una revisación sistemática de la literatura y así proponer una solución al problema antes mencionado mediante el establecimiento de criterios fundamentales de diseño. Estos criterios surgen a partir de una extensiva y minuciosa investigación en la materia; éste conjunto de criterios proporciona un marco de trabajo propicio para ayudar al Ingeniero de Software a lograr una Interfaz Gráfica de Usuario de calidad.

1. Introducción

En la vida diaria se interactúa constantemente con interfaces, ya sea con la Interfaz del celular, con la de un cajero automático por citar algunos ejemplos. Es común utilizarlas, a veces el usuario no es consciente de ello, de modo tal que a veces ni siquiera sabe qué es una Interfaz.

En términos informáticos, se puede definir Interfaz Gráfica de Usuario como: '*Conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas gráficas e imágenes. Con formas gráficas se refiere a botones, íconos, ventanas, etc. los cuales representan funciones, acciones e información*' [1].

La Interfaz es la encargada de hacer posible la comunicación entre la persona y la aplicación. Para que esto suceda la Interfaz debe estar bien diseñada, de no ser así la persona se verá frustrada en su accionar y optará por descartarla [2]. El equipo de desarrollo no sólo debe ocuparse de la corrección del sistema en términos de funcionalidad (y que cumpla con los requerimientos del cliente), sino también en cómo ve el usuario ése sistema. No solo importa que haga lo que debe hacer sino además que el usuario tenga una comunicación fluida, amigable con el sistema. La Interfaz hace que el producto sea o no competitivo. Se considera que casi la mitad del código de una aplicación está dedicada a la Interfaz.

Uno de los temas centrales en el desarrollo de Software es: ¿Cómo diseñar y crear software que sea fácil de usar y además sea funcional? Para proporcionar una respuesta adecuada a esta pregunta, no se debe ver a la Interfaz como parte secundaria del producto, sino como principal; y todo el equipo debe estar familiarizado con ello. Se sabe que cada integrante tiene un rol específico: especialistas en Interfaz (si es que hay en el equipo), documentadores, desarrolladores de código, etc. deben apuntar a que la Interfaz se integre al producto y a otros productos que formen parte del proyecto [3].

El aporte obtenido con esta investigación consiste en la recopilación de características deseables agrupadas según los siguientes criterios: (a) factores humanos, (b) reglas, técnicas y principios que respondan a cuestiones computacionales, y (c) factores que hacen a la estética y aspectos comunicacionales en el diseño gráfico. Estos criterios y sus características asociadas serán utilizados en investigaciones futuras para la elaboración de estrategias de evaluación de la calidad de las GUI y para la detección de anomalías en los criterios que las GUI deben poseer. Está demás decir que investigaciones como la presentada en este artículo son muy útiles para los investigadores que deseen incursionar en temáticas relacionadas con las GUIs debido a que proporcionan un amplio panorama del estado del arte.

Los criterios presentados surgen de un extenso estudio (en las conclusiones se detallan la bibliografía consultada) de artículos científicos alojados en sitios web de importantes Instituciones Académicas; librerías y repositorios digitales reconocidos en el ambiente académico (Google Schoolar, CiteSeerX, IEEExplore, Digital Library), y bibliografía disponible en la Biblioteca de la Universidad Nacional de San Luis.

El artículo está organizado de la siguiente manera: la próxima sección describe ejes fundamentales en el diseño de GUI: i) Factores Humanos, ii) Reglas, técnicas y principios del diseño, iii) Características deseables de las GUI y iv) Interacción persona-computadora. Por último, se encuentran las Conclusiones y trabajos futuros.

2. Ejes fundamentales en el Diseño de GUI

Al desarrollar una aplicación gran parte del trabajo consiste en programar la interfaz, que es la parte visible de la misma. Hasta hace algunos años, no se le daba mayor importancia, actualmente es distinto; la interfaz juega un rol fundamental en toda aplicación.

Teniendo presente la observación planteada en el párrafo anterior, se realizó un extensivo estudio de la bibliografía existente considerando criterios, factores y ejes que influencian el diseño de interfaz gráfica y sobre diseño en sí mismo. Este estudio mostró que, al menos en las fuentes de información investigadas, no se encuentran propuestas de criterios unificados para construir/evaluar GUI (la bibliografía trata éstos criterios por separado). Es por esta razón que en este artículo se intentará mostrar una recopilación de la bibliografía que aborda diferentes características de la GUI para, en el futuro, proponer un conjunto de ejes fundamentales que se deben considerar al momento de diseñar, construir y evaluar una GUI. En las siguientes sub-secciones se describen características, reglas y otros criterios que fueron posibles extraer luego de la extensiva investigación realizada.

2.1. Factores Humanos

Es fundamental conocer cómo la mente humana procesa la información, es por esto que al momento de diseñar una GUI, el desarrollador no sólo debe conocer características del usuario final sino también factores como los que explica Ian Sommerville en su obra "Ingeniería del software 7/e" [4]:

- Las personas tienen memoria limitada a corto plazo: se pueden recordar alrededor de siete elementos de información. Si al usuario se le presenta mucha información al mismo tiempo, no podrá asimilarla.
- Todas las personas cometen errores, en especial cuando se trata de manejar demasiada información o se trabaja bajo presión. Si la aplicación no responde como se espera aumenta el estrés de los usuarios.
- Las personas poseen un amplio rango de capacidades físicas. Algunas ven, u oyen mejor que otras; algunas son daltónicas; otras tienen problemas de motricidad.
- Las personas difieren en los gustos de interacción: algunas prefieren trabajar con menús, otras con imágenes, otras con texto; incluso otras eligen emitir comandos al sistema.

No alcanza con conocer las características del usuario final que utilizará la aplicación, sino también cuestiones de la psicología cognitiva del ser humano. Tal como lo explica el Prof. Juan Manuel Cueva Lovelle de la Universidad de Oviedo (España) en su material de estudio [5], *Cognición* es el proceso por el que se adquieren conocimientos, mientras que la *Psicología cognitiva* estudia como comprender y representar la manera en la que los seres humanos interaccionan con el conocimiento y en particular con la computadora cuando esta tecnología es usada para aprender. En la interacción persona-computadora, la persona recibe la información generada por la computadora, y responde proporcionando una entrada a la computadora. Toda ésta interacción se realiza por medio de la Interfaz. El ser humano recibe la información a través de sus sentidos: vista, oído y tacto.

A partir de lo mencionado en el párrafo anterior, se comienza a indagar sobre las limitaciones de los sentidos intervinientes. El sentido visual (el sentido de mayor incidencia), tiene como restricciones:

- Tamaño del texto: el tamaño de la letra no debe ser pequeño (12pts como mínimo), el espaciado debe ser proporcional entre líneas, las fuentes no deben ser complicadas, no utilizar sólo mayúsculas.
- Color: elegir combinaciones de colores compatibles (evitar azul-amarillo, rojo-verde, rojo-azul, verde-azul), evitar colores brillantes, llamativos en grandes porciones de la pantalla. Mostrar contraste entre el color del texto y color de fondo, la información importante no debe recaer en el color (hay personas con problemas de percepción del color).
- Formas y figuras: la distribución de los elementos es una decisión que toma el diseñador en función a su intuición o del espacio disponible; se debe considerar la información respecto a los procesos psicológicos que subyacen en la percepción organizada de escenas.

A parte de los sentidos, que son los que perciben la información, interviene la Memoria Humana[5].

En éste punto se puede concluir que [5]:

- La persona percibe información a través de los sentidos: Vista, oído, tacto. Guarda, manipula y utiliza ésta información.
- Se debe recordar que cuando el usuario no sabe qué debe hacer o falta ayuda, disminuirá su rendimiento.
- Contemplar capacidades y limitaciones del usuario ayudará en el buen diseño de sistemas interactivos.

Es evidente que el diseño de la GUI ha tomado protagonismo, tanto que ha logrado acercar ciencias Humanas como la Psicología a ciencias Exactas como la Informática. Es muy importante conocer cómo el ser humano percibe la información, cómo trabaja la mente humana, para poder mostrar la información de la manera más clara, más eficiente así la interacción es más fluida. Nunca el diseñador debe programar para sí mismo, en base a sus propias características; siempre debe hacerlo pensando en el usuario más inexperto. La Interfaz es la

encargada de guiar al usuario en el camino para concretar una tarea. Es fundamental que al momento de diseñar la GUI se intente crear un ambiente cómodo de trabajo, esto ayuda al usuario a no cometer errores, a no sentirse fatigado, tanto psicológica (en el caso de verse obligado a recordar varios pasos para concretar una tarea) como físicamente (en el caso de utilizar colores o tipos de letra que provoquen un esfuerzo visual).

El lector interesado en profundizar los conceptos tratados aquí, puede consultar [5, 6, 21, 22, 23, 24,35].

2.2. Reglas, técnicas y principios del diseño

En la sección anterior se expuso respecto a los factores humanos que se deben tener en cuenta al momento de diseñar una GUI. Pero esto no es lo único, como en cualquier otra construcción, se debe conocer y aplicar técnicas, principios, reglas que ayudan al buen diseño. Sommerville [4] establece una serie de principios, o reglas del diseño que se deben tener en cuenta para construir una GUI:

- *Familiaridad del usuario*: la GUI debe utilizar palabras, términos que sean de uso común.
- *Mostrar Uniformidad*: la GUI debe ser uniforme, sus componentes deben tener el mismo formato.
- *Ofrecer Mínima sorpresa*: el comportamiento del sistema no debe provocar sorpresa o desconcierto.
- *Proveer Recuperación de los errores*: la GUI debe ofrecer al usuario la posibilidad de recuperarse de los errores, recuperar estados anteriores.
- *Guía de usuario*: la GUI debe proveer ayuda al usuario, retroalimentación.
- *Diversidad de usuario*: la GUI debe mostrar características de interacción que se ajusten a los diferentes tipos de usuario.

Otro de los autores con gran conocimiento en el tema es Galitz, Wilbert quien explica en [7] técnicas y principios para lograr un 'buen diseño' de Interfaz:

- *Los elementos deben estar organizados clara y significativamente*.
- *Se debe mostrar un punto de inicio*.
- *Mostrar de forma ordenada los datos y contenidos.*
- *Cada elemento se debe poder diferenciar o distinguir*.
- *La información importante se debe enfocar y enfatizar*.
- *Presentar la información simple y significativamente*.
- *Utilizar tipografía legible*.

En la actualidad, a causa del avance tecnológico donde las personas interactúan constantemente con interfaces, importantes universidades han incluido en los planes de estudio de carreras informáticas asignaturas específicas a Interfaces de Usuario [8]. Como ejemplo, se puede citar el material de estudio propuesto por el Prof. Ph.D. Javier Rodeiro Iglesias en asignaturas que dicta en la Universidad de Vigo (España) [9]. Este material muestra una recopilación de reglas de diseño:

- *Otorgarle el control al usuario*.
- *Bajar la carga de memoria para el usuario.*
- *Mostrar Consistencia*.

Las reglas mencionadas coinciden con las propuestas por otro autor en "Diseño de Interfaces Hombre-Máquina" [10], allí también hace mención de las guías de diseño de interfaces que afectan a la presentación, el comportamiento y la interacción de los elementos de la interfaz, y son reglas e indicaciones a seguir en cuanto a la apariencia y comportamiento de estos. Las guías de diseño abarcan tres áreas del diseño de la Interfaz: *física* (el hardware de la Interfaz), *sintáctica* (presentación de la información, secuencia y orden de las acciones del usuario para realizar una tarea, ej. imprimir) y *semántica* (significado de los objetos y acciones, ej. las palabras Exit y Cancel). Un aspecto que se debe cuidar en las guías de diseño es el soporte para diversos lenguajes, con vistas al uso del producto a nivel internacional. Cada entorno de desarrollo tiene sus guías de diseño, ej: Microsoft (Windows).

Leopoldo S. Gómez, en su trabajo "Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación." [11] propone una serie de Principios del Diseño de GUI, algunos de ellos son:

- *Anticipación:* la aplicación debe intentar anticiparse a las necesidades del usuario.
- *Valores por Defecto*: los valores por defecto deberían ser opciones inteligentes y sensatas.
- *Eficiencia del Usuario:* se debe dar mayor prioridad a la productividad del usuario. Esperar la respuesta del sistema por un período prolongado, se convierte en pérdida económica para la organización.
- *Interfaces Explorables:* se debe permitir que el usuario pueda salir ágilmente de la GUI, dejando una marca del estado de avance de su trabajo, para que pueda continuar en otra oportunidad. La GUI debe poder 'deshacer' las acciones, debe contar con una manera fácil de finalizar la aplicación.
- *Uso de Metáforas:* las buenas metáforas crean figuras mentales fáciles de recordar (una metáfora es una representación del mundo real; ej: la tijera = cortar).
- *Reducción de Latencia:* el uso de tramas (multi-threading) permite colocar la latencia en segundo plano. Las técnicas de trabajo multitarea posibilitan el trabajo ininterrumpido, realizando las transmisión y computación de datos en segundo plano.
- *Protección del Trabajo:* se debe asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía, o alguna otra razón inevitable.

Se puede observar que diversos autores proponen las mismas técnicas y reglas. Además algunas de ellas coinciden con algunos de los factores humanos que se

deben tener en cuenta al momento de diseñar una GUI. El hecho de respetar y aplicar reglas, técnicas y principios, acercan al desarrollador al mejor diseño de GUI. Cada una de estas reglas o técnicas han sido formuladas con el fin de lograr el mejor diseño, o la mejor interacción entre la persona y la aplicación. No son enunciados arbitrarios, caprichosos y sin sentido; por el contrario. En cualquier área de desarrollo se deben tener presentes todas las reglas propuestas con anterioridad para su diseño, ya que las mismas reflejan el estudio precedente sobre el tema. El usuario se verá beneficiado si el diseñador las pone en práctica al momento de desarrollar la GUI.

El lector interesado en profundizar los conceptos tratados en los párrafos anteriores, puede consultar la siguiente bibliografía [7, 9,10, 11, 12, 13].

2.3 Características deseables de las GUI

Aplicar Reglas, Técnicas y Principios al momento de diseñar conduce a una Interfaz bien diseñada; a que ésa Interfaz cuente con las Características Deseables que el usuario espera encontrar. Wilbert Galitz [7] prioriza la Consistencia. En este sentido, el autor afirma que la Interfaz debe reflejar consistencia con el mundo real, al igual que una consistencia interna (los elementos siempre significan lo mismo).

En "Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación"[11] se establece, además de una serie principios y prototipos, características necesarias en el diseño de una GUI. Como características o pautas a evaluar propone:

- Visibilidad del estado del sistema.
- Semejanza del sistema con el mundo real.
- Control y libertad para el usuario.
- Consistencia y estandarización.
- Prevención de errores.
- Reconocimiento de acciones y opciones.
- Flexibilidad y eficiencia en el uso.
- Estética y diseño minimalista.
- Reconocimiento de errores, diagnóstico, recuperación
- Ayuda y documentación.

El grupo de investigación de Ingeniería Reversa /Comprensión de Programas /Visualización de Software de la Universidad de San Luis [14], propone una lista de características de una GUI, algunas de ellas son:

- *Ayuda*: la GUI debe brindar ayuda y asistencia al usuario (Hepl, Ayuda On Line, etc.).
- *Robustez*: debe saber responder ante diversas situaciones; inesperadas o destructivas (eliminar).
- *Funcional*: debe permitir realizar más de una tarea al mismo tiempo.
- *Efectividad*: si el usuario encuentra entre las opciones las tareas que necesita realizar.
- *Consistencia*: implica que todo elemento (imagen, ícono, sonido, etc.) siempre represente lo mismo.
- *Flexibilidad*: que permita al usuario personalizar (ej: elegir el idioma con cual trabajar)
- *Portabilidad* (Gráfica): la GUI siempre debe verse igual en cualquier Sistema Operativo.
- *Estética armoniosa*: la GUI debe mostrar un diseño visualmente agradable, sencillo, que facilite el trabajo sin distraer. Incluye: i) Fuente, ii) Color, iii) Otros elementos (imágenes, tablas, sonidos, recuadro) se deben mostrar ordenados, armoniosos
- *Vocabulario claro y preciso* (claridad): que el usuario pueda comprender los términos que se emplean (términos técnicos, vulgares, sofisticados, etc).

Se han descrito una serie de características muy importantes, pero en los últimos tiempos ha surgido una característica fundamental: la *Usabilidad*.

Son muchos los artículos que exponen respecto a la Usabilidad, por ejemplo en [14] se la define como la característica *'que hace que la aplicación sea fácil de utilizar y fácil de aprender'*. *Fácil de utilizar* se refiere a que la aplicación realiza lo que se supone debe hacer, eficiente e intuitivamente. *Fácil de aprender* se relaciona con la rapidez (o no) en que se realiza una tarea, cuantos errores se cometen y si es del agrado del usuario que está utilizando la aplicación. Se refiere a la capacidad de la GUI de ser comprendida, aprendida, usada y ser atractiva para el usuario, en condiciones específicas de uso.

El término Usabilidad tiene diversas definiciones, y todos se los considera válidos. Nielsen, considerado el padre de la Usabilidad, la define como [15]: "*La usabilidad se podría definir como la facilidad de uso. Más específicamente, hace referencia a la rapidez con que se puede aprender a utilizar algo, la eficiencia al utilizarlo, cuál es su grado de propensión al error, y cuánto le gusta a los usuarios. Si una característica no se puede utilizar o no se utiliza, es como si no existiera*".

Generalmente, se la define como la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso [16]. Para el estándar ISO 9241, que trata los requerimientos ergonómicos, la usabilidad es "*el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso*" [17]. A modo de definición, un producto cumple con la característica de Usabilidad si es:

- *Útil:* capacidad que tiene una aplicación para ayudar a cumplir tareas específicas.
- *Fácil de Usar*: se refiere a la efectividad o eficiencia, medida como velocidad o cantidad de errores.
- *Fácil de Aprender*: es la medida del tiempo requerida para trabajar con cierto grado de eficiencia usando la aplicación, y alcanzar un cierto grado de retención de estos conocimientos luego de un tiempo de no usarla.
- *Buena Apariencia*: es una medida de las opiniones,

sentimientos y actitudes generadas en el usuario por la aplicación; una medida de su seducción.

Se considera difícil medir la Usabilidad por ser una característica abstracta, es por esto que en "Calidad de software e ingeniería de usabilidad" [20] se propone un conjunto de Atributos de la Usabilidad, los mismos pueden ser medidos a través de pruebas de usabilidad. Las pruebas se aplican a la aplicación para determinar si la misma alcanza un nivel aceptable de usabilidad.

En la actualidad la Usabilidad es una característica exigible en toda GUI. Todas las características aquí mencionadas son deseables en cualquier Interfaz. Si la GUI cuenta con la mayoría de ellas se puede considerar que es una GUI bien diseñada. Cada una de éstas características ayudan a evaluar la GUI, ésta evaluación es necesaria cuando se cuenta con más de una aplicación para realizar una tarea específica. Pero además de ayudar a seleccionar la mejor opción también ayuda a mejorar el diseño al momento de desarrollar un sistema. La tarea del desarrollador no sólo se reduce a cumplir con los requerimientos del usuario, sin considerar los atributos y características que debe poseer una GUI. El lector interesado en ampliar sus conocimientos en ésta área puede consultar [18, 19, 20, 30, 32, 34].

2.4 Interacción Persona-Computadora

Con la progresiva digitalización de los contenidos y la informatización de los servicios, a menudo el usuario no trata directamente con una persona sino con una aplicación informática, y no consulta documentos en papel sino en formato digital. El profesional Informático para ofrecer un buen servicio debe dominar nuevos campos, como la interacción persona-computadora [25]. Por esto, en los últimos tiempos ha surgido una disciplina que cada vez toma mayor importancia, la Interacción Persona-Computadora (IPC), (o HCI: Human Computer Interaction). Se considera que ésta disciplina es el punto de encuentro de las ciencias humanas, como la psicología, la pedagogía; con la tecnología o las ciencias exactas como la informática. La IPC se centra en estudiar la interacción entre usuarios y sistemas informáticos; su objetivo es ofrecer bases metodológicas, prácticas y teóricas para el diseño y evaluación de aplicaciones interactivas[26]. Esta tarea se realiza para que puedan ser usadas de forma eficiente, eficaz, segura y satisfactoria.

A modo de definición, se puede decir que: *la interacción persona–computadora es el intercambio observable de información, datos y acciones entre un humano y la computadora y viceversa* [30]. Está constituida por la comunicación, la conversación, el diálogo que pueda fluir en ambas direcciones, entre una persona y el sistema que se está utilizando. La interfaz de usuario, es el medio por el cual la interacción persona-computadora es establecida, manejada y controlada. La misma conforma la parte del software y hardware que permite, que ése intercambio de información y sus distintas secuencias o hilos de diálogo, se produzcan.

Desde una concepción más amplia, la IPC [30] se la considera como un área dentro de las Ciencias de Computación que se encarga del diseño, evaluación e implementación de sistemas de computación interactivos, para el uso humano, incluyendo el estudio de todos los fenómenos concernientes a ello. La IPC se convierte en un área interdisciplinaria, que puede nutrirse de nociones provenientes de la Psicología, por el análisis empírico del comportamiento del usuario y la aplicación de teorías de procesos cognitivos; de la Antropología y Sociología, por el estudio de la interrelación entre la tecnología, el trabajo y las organizaciones; del Diseño Industrial, por el estudio y diseño de productos de interacción; del Diseño Visual, por el uso de formas, colores y otros paradigmas visuales y sus diferentes aplicaciones; entre otros.

Los primeros trabajos de la IPC se focalizaron en educar a trabajadores de oficina con capacidades o talentos particulares, luego se fueron perfeccionando para ayudar a niños, personas con capacidades diferentes y ancianos [31].

La Usabilidad, característica mencionada en la sección anterior, es un tema central en la IPC y no sólo se la define como atributo de calidad, sino también como metodología de diseño y evaluación. En éste sentido se habla de Ingeniería de Usabilidad y Diseño Centrado en el Usuario, que abarcan un conjunto de procesos y metodologías que aseguran el cumplimiento de los niveles de usabilidad requeridos en la aplicación.

Si bien la IPC se focaliza en la interacción persona-computadora a través de la Web, ésta disciplina aporta, y mucho, al desarrollo de la GUI. Es interesante observar la evolución que se ha dado en éste tema: desde las primeras aplicaciones, donde el usuario debía ser personal idóneo y la interacción se hacía efectiva sólo si la persona sabía qué es lo que debía hacer; hasta la actualidad donde se ha desarrollado toda una disciplina que se dedica a estudiar cómo mejorar ésa interacción. Tal como se mencionó al inicio de ésta sección, gran cantidad de servicios se tramitan a través de una computadora, gran parte de la información en la actualidad se ha digitalizado. Es por esto que la interacción debe ser posible para todo tipo de usuario (experto o no), si la GUI no está bien diseñada difícilmente la comunicación será fluida. El lector interesado en profundizar los conceptos tratados en ésta sección puede leer [27, 28, 29, 33]

3. Conclusiones y trabajos futuros

El presente artículo tiene como objetivo mostrar una revisación sistemática de la literatura existente en cuanto a diseño de GUI; ésta tarea tiene como finalidad:

i) cubrir la ausencia, encontrada hasta el momento en la extensa bibliografía consultada, respecto de los criterios o ejes de diseño de GUI que deben tener en cuenta para que las mismas sean eficientes y de calidad, y ii) proporcionar las bases para establecer un conjunto de ejes fundamentales que ayuden al profesional informático a: lograr el mejor diseño de la GUI, aplicar mecanismos de evaluación de GUI, proporcionar a investigadores iniciales un acabado estado del arte respecto a los criterios fundamentales que se deben tener en cuenta al diseñar/construir una GUI.

Para alcanzar el objetivo mencionado, se hizo una investigación profunda respecto a los ejes fundamentales que avalan un buen diseño de GUI y se concluyó que entre los mismos se encuentran: Factores Humanos; Reglas, Técnicas y Principios de Diseño; Características Deseables de las GUI y la nueva disciplina IPC que ofrece una metodología para hacer efectiva la Interacción entre personas y computadoras.

Es necesario hacer un estudio riguroso de todos los criterios aquí propuestos al momento de programar una GUI, ya que ellos permiten generar una GUI bien diseñada. La formulación de estos criterios no se hizo de manera caprichosa y sin sentido, sino que los mismos se obtuvieron a partir de un extenso análisis de bibliografía referida al tema; donde se pudo verificar que hasta el momento no se encuentran disponibles trabajos de investigación que establezca o proponga ejes troncales para el diseño de GUI. Por consiguiente se considera que el resultado de esta investigación es un humilde aporte al tema que propende a cubrir la falta de aportes en esta temática. El conocimiento de estos criterios permite enfocar mejor al diseño de la GUI, más orientado a los requerimientos del usuario. Las Interfaces de usuario, como vínculo de inmersión del hombre en el entorno actual de trabajo tecnológico, realzan su importancia en el desarrollo de nuevos productos, más interactivos, eficaces y eficientes.

Como trabajos futuros se considera seguir ahondando en el tema de los ejes aquí mencionados; sin descartar considerar un nuevo eje que incida en el Diseño de Interfaces Gráficas de Usuario. Además se considera investigar la manera de evaluar éstos ejes en una GUI.

En el trabajo actual se realizaron aportes propios sobre una base de más de 100 artículos científicos descargados de librerías digitales reconocidos en el ambiente académico (IEEExplore, CiteSeerX, Google Schoolar) –de los cuales 32 abordan el tema de GUI de una aplicación particular–; 6 capítulos de libros y 16 sitios web de Universidades prestigiosas que incluyen en los planes de estudio de carreras informáticas asignaturas referentes a las GUI (materiales de estudio).

Para finalizar, es importante mencionar que este trabajo es una investigación realizada para optar por el Título de Posgrado "Especialista en Ingeniería de Software" carrera dictada en la Universidad Nacional de San Luis.

4. Referencias

[1] http://www.alegsa.com.ar/Dic/gui.php

[2] Mercovich, Eduardo. "Ponencia sobre Diseño de Interfaces y Usabilidad: cómo hacer productos más útiles, eficientes y seductores." *Buenos Aires: SIIGARPH* (1999).

[3]Wills, C. "User interface design for the engineer". In *Electro/94 International. Conference Proceedings. Combined Volumes.*(1994, May) (pp. 415-419). IEEE.

[4]Sommerville, Ian. "Ingeniería del software 7/e". *Pearson Educación*, 2005.

[5]Material de estudio del Prof. Juan Manuel Cueva Lovelle de la Universidad de Oviedo – España http://di002.edv.uniovi.es/~cueva/asignaturas/doctorado/2004/2FactorHumano.pdf

[6]iSaltiveri, Toni; Vidal, Jesús & Cañas Delgado, José. "Diseño de sistemas interactivos centrados en el usuario". *Editorial UOC*, 2011.

[7]Galitz, Wilbert O. "The essential guide to user interface design: an introduction to GUI design principles and techniques". *Wiley.com*, 2007.

[8]Abascal, J.& Moriyón, R. "Tendencias en interacción Persona-Computador". *Revista Iberoamericana de Inteligencia Artificial* 6.16 (2002).

[9]Material de estudio del Prof. Javier Rodeiro Iglesias– "Diseño de Interfaces Hombre-Máquina" http://trevinca.ei.uvigo.es/~jrodeiro/Teaching/diu/DIU-Tema1.pdf

[10]Rodríguez, Patricio. "Diseño de Interfaces Hombre-Máquina" (HMI). *Instituto de Electricidad y Electrónica* – Universidad Austral de Chile

[11]Gómez, Leopoldo Sebastián M. "Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación".

[12]www.escet.urjc.es/~intgraf/documentos/Principios-Disenno-Delphi.pdf

[13]Albornoz, M. Claudia. "Diseño de Interfaz Gráfica de Usuario". *XVI Workshop de Investigadores en Ciencias de la Computación.* 2014.(pág 540-544)

[14]Albornoz, M. Claudia, Miranda, Enrique & Berón, Mario. "Evaluación de Interfaces Gráficas de Usuario Usando LSP". *1° Congreso Nacional Ingeniería Informática /Sistema Información* – CoNaIISI 2013.

[15]Nielsen, J., & Loranger, H. (2006). "*Prioritizing web usability"*. Pearson Education.

[16]International Organization for Standardization ISO 9126: Software Engineering. Product quality, Geneva, Switzerland. ISO 2001

[17]International Organization for Standardization ISO 14598-1.Information Technology - Evaluation of Software Products. General Guide. ISO 1998.

[18]Moracho, J. G. (2007). "Pensando en el usuario: la usabilidad". *Anuario ThinkEPI*, (1), 172-177.

[19]http:// www.bohmann.dk/observations/2001oct12.html

[20]Mascheroni, M, Greiner, C., Petris, R., Dapozo, G., & Estayno, M.(2012). "Calidad de software e ingeniería de usabilidad". *In XIV Workshop de Investigadores en Ciencias de la Computación*.

[21]Card, S; Moran, T y Newell, A. “The Psycology of Human-Computer Interaction”. *Lawrence Erlbaum Assoc*, 1983
[22]G. Salvendy, John Wiley & Sons. “Handbook of Human Factors and Ergonomics”. *Second Edition*, 1997.
[23]Oborne, David. “Ergonomics at Work: Human Factors in Design and Development”. *Third Edition*, 1996.
[24]Smith, A. (1997)“Human-computer factors: a study of users and information systems”. *McGraw-Hill Higher Education*.
[25]Turró, M.“Evolución y tendencias en la interacción persona - ordenador”. El profesional de la información, 14(6), 414-422. 2005
[26]Hassan Montero; Yusef M.; Fernández F. “La experiencia del usuario”- no solo usabilidad: revista de diseño y tecnología (ISSN 1886-8592) http://www.nosolousabilidad.com/articulos/experiencia_del_usuario.htm
[27]Granollers, Toni, et al. "Integración de la IPO y la ingeniería del software: MPIu+ a." *III Taller en Sistemas Hipermedia Colaborativos y Adaptativos*, Granada España (2005).
[28]Mor, E; Muriel Garreta, D; Galofré, M. "Diseño Centrado en el Usuario en Entornos Virtuales de Aprendizaje, de la Usabilidad a la Experiencia del Estudiante." 2007 *SPDECE*.
[29]Londoño L., Felipe. “Interficies de las Comunidades Virtuales” - Capítulo 1 del libro: INTERFACES DE LAS COMUNIDADES VIRTUALES - ISBN: 958-8231-39-6 - *Editorial Universidad de Caldas, 2004* - Universitat Politécnica de Catalunya. España.
[30]Díaz, Javier F., Harari, Ivana, & Amadeo, Ana P.. "Guía de recomendaciones para diseño de software centrado en el usuario."*Editorial: Editorial de la Universidad Nacional de La Plata (EDULP) (2013). Institución de origen: Facultad de Informática.* ISBN: 978-950-34-1030-1
[31]Olson, G, & Olson, J. (2003) "Human-computer interaction: Psychological aspects of the human use of computing." *Annual review of psychology* 54.1 p: 491-516.
[32] Ruiz, A., & Price, Y. W. (2008, August).” GUI testing made easy”. In *Practice and Research Techniques, 2008. TAIC PART'08. Testing: Academic & Industrial Conference* (pp. 99-103). IEEE.
[33]Othman, M. K. (2006, June). “Graphical user interface design for Collaborative Virtual Environments”. In *Computing & Informatics, 2006. ICOCI'06. International Conference on* (pp. 1-6). IEEE.
[34] Miyoshi, T., & Murata, A. (2001). “A method to evaluate properness of GUI design based on complexity indexes of size, local density, aliment, and grouping”. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on* (Vol. 1, pp. 221-226). IEEE.
[35] Oshana, R. (2013). “Human Factors and User Interface Design for Embedded Systems”. *Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications*, 417.

Aprendizaje Móvil para el Aprendizaje Organizacional en el Proceso de Desarrollo de Software

Figueroa, Liliana
Universidad Nacional de Santiago del Estero, Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías
Rios, Miriam
Universidad Nacional de Santiago del Estero, Departamento de Matemática, Facultad de Ciencias Exactas y Tecnologías
Maldonado, Marilena
Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías
Universidad Nacional de Santiago del Estero
López, Gustavo
Departamento de Matemática, Facultad de Ciencias Exactas y Tecnologías Universidad Nacional de Santiago del Estero.

Abstract
En el ámbito de las organizaciones de desarrollo de software es necesario contar con recursos humanos, con conocimientos y experiencias para la transmisión del conocimiento, por los cual, resulta de interés contar con espacios de intercambio formal e informal de aprendizaje, de formación del conocimiento, salvando sobre todo las barreras de distancia y tiempo; donde el Aprendizaje Móvil constituye una alternativa tecnológica a la cual se podría recurrir para estos entornos de trabajo. En estas organizaciones basadas en conocimientos, el Aprendizaje Organizacional permite a los miembros de los equipos de proyectos, renovar los conocimientos actuales y mejorar sus procesos en tiempo y forma, y de esta manera brindar una mejor respuesta a los cambios del entorno de trabajo.

Palabras Clave
Aprendizaje organizacional, aprendizaje móvil, desarrollo de software.

Introducción

Actualmente el Aprendizaje Organizacional permite no solo renovar los conocimientos actuales y generar nuevos, sino mejorar continuamente los procesos y rutinas en las organizaciones, para responder adecuadamente a los cambios del entorno.

En particular, en el ámbito de las organizaciones, cuando desarrollan software es necesario contar con recursos humanos, con conocimientos y experiencias; de esta manera se tratan de organizaciones basadas en conocimientos. Así, los conocimientos y experiencias que los miembros de los equipos de proyectos crean y adquieren durante el desarrollo del software constituyen entonces un valioso activo para estas organizaciones.

En este contexto resulta de interés crear espacios de intercambio formal e informal de aprendizaje, de formación del conocimiento, salvando sobre todo las barreras de distancia y tiempo; entonces el Aprendizaje Móvil constituye una alternativa tecnológica a la cual se podría recurrir para estos entornos de trabajo.

A partir de esta realidad, la propuesta de este trabajo es plantear al Aprendizaje Móvil como una estrategia de Aprendizaje Organizacional en el proceso de desarrollo software.

Para alcanzar el objetivo propuesto se ha organizado este trabajo en las siguientes sesiones en donde se abordara la nueva propuesta de aprendizaje y su vínculo con el Aprendizaje Organizacional, luego se elaborarán conclusiones y aportes personales a esta nueva estrategia.

Aprendizaje Móvil

Aprendizaje Móvil es la capacidad de usar tecnología móvil para lograr una experiencia de aprendizaje en diversos ámbitos, puede utilizarse en las empresas para capacitar a los empleados, para hacer labor de campo y para realizar campañas

promocionales. Además, promete apoyar la enseñanza virtual por medio de nuevos dispositivos portátiles que fomenten nuevos sistemas de comunicación entre el hombre y la tecnología; y ayuda a eliminar la formalidad que existe en el método tradicional de aprendizaje y a combatir la resistencia al cambio tecnológico que existe por parte de los adultos que ahora necesitan involucrarse a esta nueva época que se vive en materia de aprendizaje móvil.

Definiciones de Aprendizaje Móvil.

Se podría decir que el m-learning es una evolución de e-learning, que es a su vez una evolución de la formación a distancia. El m-learning es una metodología de enseñanza y aprendizaje valiéndose del uso de pequeños y maniobrables dispositivos móviles, tales como teléfonos móviles, celulares, agendas electrónicas, tablets PC, pocket pc, ipods, ipads y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica.

Conceptualmente se puede afirmar que se denomina m-learning a la difusión de contenidos formativos mediante dispositivos móviles. Los usuarios buscan contenidos "just in time, just for me" que se ajusten de forma muy concreta a su perfil, los cuales pueden utilizarse en el momento en que se requiera.

El Aprendizaje Móvil tiene varias definiciones, dependiendo del enfoque donde se ubica dentro de los ambientes de aprendizaje:

El m-learning es el descendiente directo del e-learning para varios investigadores [1][2], dado que el e-learning es el aprendizaje apoyado por recursos y herramientas electrónicas digitales y mlearning es el e-learning que se apoya de dispositivos móviles y transmisión de wireless; o simplemente, es cuando el aprendizaje toma lugar con dispositivos móviles.

El aprendizaje, según [3] lo describe como un proceso de acercamiento al conocimiento, donde los participantes en cooperación con sus compañeros y profesores, construyen en forma conjunta la interpretación de su mundo. Esta definición da a las tecnologías móviles un rol especial porque incrementa sus posibilidades de comunicación y conversación.

Según [4], menciona que es el que se da a través de enseñanzas que no están limitadas por el ambiente de aprendizaje, sino que lo complementa, enriquece y estimula para provocar un aprendizaje flexible y móvil, que le ayuda al estudiante a aprender desde diferentes escenarios y contextos.

M-learning es una manera de apoyar al aprendizaje en un medio ambiente donde diversos elementos como la espontaneidad, la personalización, la informalidad, la contextualización, la portabilidad, la conveniencia, la adaptabilidad, la integración y la disponibilidad, juegan un papel relevante [6].

Otros investigadores, Grupo de e-learning 360 citado por [2] lo definen más a partir del proceso y mencionan que el m-learning es cualquier actividad que permite a los individuos ser más productivos cuando consumen, interactúan con o crean información, mediada a través de un dispositivo digital compacto, que el individuo lleva consigo de manera constante, que tiene una conectividad confiable y que le cabe en el bolsillo.

Estas cinco definiciones dan luz sobre las diferentes formas en que puede ser definido el término de Aprendizaje Móvil, pero en ellas podemos vislumbrar concordancia en que intervienen dos elementos: movimiento y aprendizaje. Estos dos elementos son transmitidos a través de ciertos recursos: los dispositivos móviles.

Dispositivos Móviles

Un dispositivo móvil es un procesador con memoria que tiene muchas formas de entrada (teclado, pantalla, botones, etc.), también formas de salida (texto, gráficas, pantalla, vibración, audio, cable). Algunos dispositivos móviles ligados al aprendizaje son las laptops, teléfonos celulares, teléfonos inteligentes, asistentes personales digitales (Personal Digital Assistant, PDA, por sus siglas en inglés), reproductores de audio portátil, iPods, relojes con conexión,

plataformas de juegos, etc.; conectados a Internet, o no necesariamente conectados, cuando ya se han "archivado" los materiales.

Con esta visión podemos ver cómo las comunicaciones de redes inalámbricas, sus tendencias y productos están revolucionando e incrementando en el día a día, el cómo nos comunicamos y las posibilidades de acceso que tenemos. Aún así, es conveniente también señalar que las redes móviles, al ser nodos móviles, pueden tener variaciones y los enlaces entre nodos, ya sean unidireccionales o bidireccionales, se pueden interrumpir, aunado a esto tenemos que tener presente que el ancho de banda disponible en una interface inalámbrica es inferior al que utiliza cables, y además pueden darse interferencias en las señales electromagnéticas.

En este abanico de posibilidades, y de retos, también, [6] mencionan que los recursos a través de dispositivos móviles pueden aportar al ámbito de los procesos de enseñanza y aprendizaje muchas potencialidades y que se requieren concebir nuevos métodos, prácticas y diseños que contemplen las características tecnológicas particulares que tienen los dispositivos. Estas características se dan por la esencia de la "portabilidad" del dispositivo, pero esto también puede llegar a ser una potencialidad por: la posibilidad de conexión para comunicaciones espontáneas y colaborativas, capacidad de proveer información de dispositivo a dispositivo, localización de información inmediata, capacidad de recursos con sonido, grabación, cámaras, videoclips.

Ventajas del Aprendizaje Móvil

Entre las ventajas del aprendizaje móvil se pueden mencionar:

Mayor flexibilidad para el acceso de contenidos al facilitar el aprendizaje en cualquier momento y desde cualquier lugar a través del celular.

Personalización de experiencias de aprendizaje al permitirle al usuario escoger el dispositivo, el lugar y el tiempo que mejor se acople a su ritmo de vida.

Aprendizaje significativo a través del diseño de ambientes de instrucción que propicien experiencias de acuerdo a la realidad del usuario.

Desarrollo y fortalecimiento de habilidades profesionales. Además del contenido se adquieren otras habilidades como el uso de tecnología de punta, la comunicación efectiva, trabajo en equipo y búsqueda de información.

La posibilidad de 'aprender en movimiento', literalmente en movimiento: podemos acceder a contenidos formativos durante nuestros desplazamientos.

Mayor efectividad del aprendizaje al promover una atención activa por periodos más largos.

La posibilidad de recurrir a la instrucción que necesitamos en cualquier momento y en cualquier lugar.

Las diferentes experiencias llevadas a cabo han demostrado que el Aprendizaje Móvil es realmente efectivo en:

Aprendizajes basados en la resolución de problemas y mejora de determinadas habilidades.

Aprendizajes al aire libre o para trabajos de campo.

Aprendizajes en instituciones culturales. En estos entornos se ha demostrado que las tecnologías multimedia e inalámbricas son una eficaz herramienta, ya que proporcionan al usuario información de interés en función del lugar en el que se encuentre.

Actualización profesional.

Aprendizaje Organizacional

El Aprendizaje Organizacional es la habilidad de una organización para lograr un mejor entendimiento de sus prácticas a partir de la experimentación, la observación, el análisis y la colaboración; y una organización que aprende es una organización que adquiere nuevos conocimientos y es capaz de innovar con la rapidez necesaria para adaptarse y sobrevivir en un entorno dinámico.

En estos últimos años, los avances en TIC han sido aplicados, tanto en las instituciones educativas como en las organizaciones; debido a que factores como tiempo, costo y disponibilidad han comenzado a impactar cada vez más en las mismas; y en la manera en que llevan a cabo en todos sus procesos de negocios y en particular la formación de sus empleados.

Además y considerando que, la formación es fundamental en el desarrollo social del hombre, y a lo largo de la historia, la formación casi siempre se ha llevado a cabo con el modelo aprendiz-maestro, mediante conocimientos adquiridos, aprendiendo en el lugar del trabajo mediante la práctica directa. Luego, fue el surgimiento de las instituciones académicas lo que provocó la división entre el entorno laboral y el formativo. Es un hecho que las organizaciones pretenden crear contextos de aprendizaje que se asemejen al laboral, para dar mayor validez a los conocimientos adquiridos por los usuarios.

Entonces las empresas aplican modelos de formación y capacitación basados en nuevos desarrollos tecnológicos y en dispositivos móviles cuya reciente aplicabilidad se debe también al incremento del gasto en formación que año a año deben afrontar las organizaciones actuales. Pero, a pesar de que se ha producido un avance vertiginoso en dicha materia, todavía las organizaciones tienen que afianzar las ventajas que dicho sistema supone.

Este modelo demanda grandes retos, ya que a diferencia de la e-learning el recurrir en la formación a las tecnologías de comunicación móvil permitirá que los empleados tengan acceso prácticamente en cualquier momento y en cualquier lugar, cuando estos lo quieran, el aprendizaje organizacional no tiene una ubicación fija y de esta manera se va logrando que la organización adquiera nuevos conocimientos y experiencias que le permita adaptase a las nuevas exigencias del ambiente cambiante.

Además, las tecnologías de comunicaciones móviles también pueden emplearse para sistematizar y organizar los saberes internos de la organización, a fin promover la interacción espontánea y la formación continua; de esta manera, se podría construir la infraestructura necesaria para la innovación. Por otro lado, la flexibilidad y escalabilidad que caracteriza a las tecnologías del m-learning contribuyen con la generación, intercambio y aplicación de conocimientos de forma permanente.

De esta manera el objetivo del Aprendizaje Organización recurriendo a tecnologías móviles es desarrollar y compartir una inteligencia colectiva a través de la comunicación efectiva con el propósito de desarrollar nuevos conocimientos y así poder innovar. Se lograra entonces:

Producir una amplia variedad de soluciones a problemas organizacionales.

Clarificar la visión, el propósito, los valores y el comportamiento organizacional.

Relacionarse mejor entre empleados.

Que el personal se sienta apoyado y se comprometa con la organización.

Un desempeño organizacional efectivo y una ventaja competitiva superior.

Equilibrar las demandas de los inversores.

Aumentar la habilidad de enfrentar los cambios.

Reducir la probabilidad de ocurrencia de los mismos errores.

Comprender los riesgos y la diversidad con mayor profundidad.

Expandir las capacidades profesionales de todos los empleados.

Que la organización no perezca y logre innovar.

Desde hace algún tiempo, la propuesta en materia de Aprendizaje Organizacional es que la formación se realice bajo una propuesta de ubicuidad, basada en la movilidad y el tiempo, donde es fundamental: formarse en cualquier lugar y en cualquier momento, donde y cuando el empleado lo necesite. Esto hace que la formación sea mucho más efectiva. Se fomenta un formación más directa, basada en la demanda, y muy útil en contextos de la empresa. Lo más significativo es que se permite el acceso a los materiales

formativos en el momento en que se lleva a cabo la tarea práctica.
Se libera así al empleado, en muchas ocasiones lo ideal es dar la formación en el contexto donde desarrolla sus actividades. Ese hecho, acelera la comprensión y práctica de los conocimientos adquiridos, mientras aumenta la motivación y la involucración de los empleados en la formación.
De la misma manera que el e-learning fomentó la formación frente al ordenador, donde no debía abandonar su puesto de trabajo para asistir a la capacitación; con el m-learning el trabajador queda liberado de la formación frente a un ordenador. Para ello es fundamental:
Realizar un análisis de necesidades formativas.
Implantar el m-learning como un elemento más dentro de un programa de formación mixta.
Crear nuevos contenidos adaptados a dispositivos móviles. No valen los contenidos del e-learning ni de la formación presencial. Se trata de una formación más directa, basada en la demanda.
Implantar en forma gradual el m-learning a través de los planes formativos de las organizaciones.
Facilitar la integración entre el entorno laboral y el proceso de aprendizaje. Se fomenta el aprendizaje social gracias al contacto con otras personas.
Para un trabajador móvil y en particular para todo aquel empleado que no pueda o no desee mantener una ubicación fija por mucho tiempo, es indispensable contar con soporte tecnológico que le permita tener acceso a la capacitación en todo momento, sin importar el medio de comunicación, la hora o el lugar. Por lo tanto la nueva estrategia de aprendizaje, que se viene presentado desde un comienzo en este trabajo, emerge para satisfacer las necesidades individuales de estos empleados, permitiéndoles acceso a información específica desde cualquier lugar, en cualquier momento y en cualquier dispositivo portátil, ya sea para su utilización inmediata o posterior. En este sentido la información que puede ser mostrada sobre un formato visual o auditivo es restringida, ya sea un texto o una combinación entre ellos. Este tipo de unidades son utilizados en la actualidad dentro de modalidades autos formativos centrados en un aprendizaje breve y conciso, es decir contenidos que no admiten ninguna duda sobre lo que se quiere obtener de ellos. Entre los tipos de contenidos mínimos de información encontramos los siguientes: Contenidos casuales, Contenidos de protocolo, Contenidos de mensaje social, relacional o comunicativo, Contenidos específicos y Contenidos de control.

Proceso de Desarrollo de Software

Hoy en días se requiere cada vez más, que el software sea desarrollado en tiempos cada vez más breves, por lo cual debe ser posible reciclar procedimientos, procesos, diseños, estrategias, alternativas y experiencias que permitan mejorar los procesos de desarrollo de software. Por otro lado se da una realidad que la demanda de software aumentará de manera significativa, promocionada por los cambios económicos mundiales, las nuevas aplicaciones tecnológicas y el auge de los dispositivos móviles, en otras palabras no es posible que la oferta de desarrollo de software pueda cubrir toda esta demanda, entonces ¿cómo se obtendrá la capacidad para poder satisfacer esta demanda? La posible solución no es aumentar el número de desarrolladores, porque esto podría traer más problemas que ventajas. Lo que se debe lograr es cambiar radicalmente los métodos de desarrollo de software para que las personas implicadas no realicen tareas repetitivas, y puedan recurrir a experiencias y conocimientos de proyectos ya desarrollados.
De esta manera, el Aprendizaje Organizacional en este contexto constituye una alternativa para que el conocimiento y las experiencias puedan ser retroalimentados entre los diferentes miembros del equipo de desarrollo; y los

flujos de conocimientos puedan ser dinámicos, abiertos y distribuibles.
El autor [7] afirma que los procesos de desarrollo de software se apoyan fuertemente en el conocimiento y la creatividad tanto de los individuos como de los equipos de desarrollo de software, y sostiene que el principio básico en la Ingeniería de Software es que la calidad general del software puede mejorarse cuando el conocimiento se hace disponible y se lo utiliza de manera competente. Para este autor, la necesidad de fomentar el desarrollo de las prácticas de Ingeniería de Software en las organizaciones es un agregado a la demanda de gestionar sistemáticamente el conocimiento y las destrezas en todas las etapas del ciclo de vida del software. Concluye que desarrollar maneras efectivas de gestionar el conocimiento relativo al software es de interés para los desarrolladores.
Para [8], el aprendizaje en y alrededor de los proyectos software no es una opción, sino una obligación ("a must") para la sobrevivencia organizacional. Agregan que la reutilización de conocimiento es apropiada en Ingeniera de Software, pero que el conocimiento anterior debe reutilizarse para mejorar las operaciones futuras y no para repetir errores cometidos en el pasado.

Conclusiones

Cada vez hay una mayor tendencia al uso de dispositivos móviles para el manipular la información, para integrar actividades cotidianas, para manejar tecnología de punta y para buscar información. Con esta tecnología los usuarios están aprendiendo a adaptar y realizar sus tareas movilizándose, es decir, a ser y a estar móviles.
Además, a las organizaciones les interesa y se preocupan por adquirir nuevos conocimientos en el momento justo en el que se necesite para ser aplicado en el contexto adecuado por quien lo requiera, cuando toma decisiones, diseña, planifica, diagnostica y evalúa. Por lo tanto, surge la necesidad de gestionar este conocimiento de una forma especial que brinde apoyo al aprendizaje individual a partir de las experiencias y practicas realizadas.
Es entonces, un desafío para las organizaciones que hoy sus empleados se capaciten en cualquier momento y en cualquier lugar, y puedan adquirir y compartir conocimiento.
Por lo tanto para las organizaciones, conformar un plan o estrategia de capacitación basada en M-learning requiere no sólo contar con tecnología apropiada, sino que también se requiere de crear una formación apoyada en la demanda de cada uno de los empleados, con aplicaciones diferentes basadas especialmente en dispositivos móviles; es por esta razón que aun las organizaciones imponen en algunos casos resistencia a la hora de recurrir a estas estrategias de formación.
Si bien es cierto que hay organizaciones más innovadoras en el uso de tecnologías y han experimentado y apostado en un plan de formación basada en entornos móviles, de las cuales podemos encontrar sus experiencias y ventajas en sitios web; aun quedan cuestiones por hacer frente, y que no está solo vinculado con el uso de tecnológicas, sino que son parte también de la cultura de la organización y de las políticas y estrategias que esta fija.
Una situación quizás más compleja se presenta en organizaciones de nuestro medio, en donde si bien se ha dado un "contagio" por el uso de telefonía móvil, incrementando la demanda en forma personal, aun no se la pueda considerar como una propuesta para que las organizaciones puedan aprender y tengan la información y el conocimiento que describan actividades y tareas que deben realizarse sistemáticamente, conformando de esta manera una estrategia continua de formación y capacitación. Entonces, se requiere primero un cambio cultural que fomente la capacitación en el mismo lugar de trabajo y desde las mismas experiencias compartidas, y por otro lado, también los profesionales informáticos deben iniciarse

en el desarrollo de aplicaciones para estos entornos tecnológicos.

Si bien a lo largo de este trabajo, hemos destacado los avance, tanto en la tecnología móvil como en sus aplicaciones y sus impactos en el contexto de las organizaciones, aun quedan algunas cuestiones que no permiten potenciar la mejora en los procesos de aprendizaje continuos a través de esta tecnología, que permita fortalecer al conocimiento de los procesos de software.

Referencias

[1] Pinkwart, N.; Hoppe, H. U.; Milrad, M.; Perez, J. (2003). Educational scenarios for the cooperative use of Personal Digital Assistant. Journal of Computer Assisted Learning, 19, 3, (383-391).

[2] Quinn, C. (2007). Mobile magic: Think different by design. Ciclo de conferencias de la Escuela de Graduados en Educación y Centro de Innov@te del Tecnológico de Monterrey, 18 de septiembre. Podcast. Disponible en: http://podcastuv.itesm.mx.

[3] Sharples, M. (2005). Learning as conversation: Transforming education in the mobile age. Proceedings "Seeing understanding, learning in the mobile age". Budapest, 28-30 de Abril, (147- 152).

[4] Salz, P. A. (2005). When will we ever learn? Mobile Communications International, 1, 129.

[5] Laouris, Y.; Eteokleous, N. (2005). We need an Educationally Relevant Definition of Mobile Learning. mLearn 2005 4th World conference on mLearning. Conference theme: Mobile technology: The future of learning in your hands. Disponible en: http://www.mlearn.org.za/CD/papers/Laouris%20&%20Eteokl eous.pdf

[6] Kukulska-Hulme,A.; Traxler, J. (2005). Mobile learning a handbook for educators and trainers. London, England: Routledge.

Pastore, R. (2002). Elearning in Education: An Overview. Proceedings of Society for Information Technology and Teacher Education International Conference 2002. (275-276). Chesapeake, VA: AACE.

[7] Aurum, A.(2003) Supporting structures for managing software engineering knowledge, en Aurum, A. et al. (eds.) Managing Software Engineering Knowledge, Berlin, Springer, 2003,pp. 69-72.

[8] Desouza, K., Dingsoyr, T., Awazu Y. (2005): Experiences with conducting project postmortems, Proceedings of the 38 Hawaii International Conference on System Sciences, 2005, p. 233

Análisis para la identificación de clusters en información recopilada de empresas de desarrollo de software sobre técnicas de priorización de requerimientos

De Federico, Sara, Sincosky, Noelia, Lascano, Alfredo, Avogradini, Mariela, Moschetti, Diana

Universidad Tecnológica Nacional Facultad Regional Rosario
Departamento Ingeniería en Sistemas de Información

Abstract

Dentro de un proyecto de investigación, la obtención de datos reales y ricos en información sobre el tema de investigación es una base sólida desde donde continuar con las metas subsiguientes hacia el objetivo final. El uso adecuado de estos datos y la obtención de toda la información explícita e implícita contenida marca un punto de inflexión determinante para el alcance de la mayor cantidad y los más ajustados resultados. Este trabajo muestra algunas de las tareas realizadas para el análisis de la información obtenida de un grupo de encuestas realizadas a empresas de software de la ciudad y alrededores, y dentro del marco del proyecto de investigación "Creación y Simulación de Metodologías de Análisis, Clasificación e Integración de Nuevos Requerimientos a Software Propietario" Res. CD 680/12 de la Universidad Tecnológica Nacional, Facultad Regional Rosario, orientadas a la identificación de clusters sobre técnicas de priorización de nuevos requerimientos a implementar en software en etapa de mantenimiento.

Palabras Clave

Software, mantenimiento, priorización, nuevos requerimientos, métodos, procedimientos.

Introducción

La integración de nuevos requerimientos a un producto de software que ya ha sido implementado cuenta con multitud de procedimientos y técnicas referentes para la selección de un pool de nuevos requerimientos que han sido solicitados de forma que se garantice la funcionalidad y tenga en cuenta las necesidades y el impacto sobre los usuarios pasivos (que no han sido solicitantes de estos requerimientos nuevos incorporados) [1,2,3]. La permanencia de productos de software que han logrado una posición importante en el mercado ha dado pie a la investigación de cómo mantenerlo, logrando al mismo tiempo captación de nuevos clientes y la conservación y satisfacción de los clientes actuales. Es muy conocido el riesgo y las consecuencias catastróficas que devienen de lanzamiento de una nueva versión que ha caído en funcionalidad y usabilidad del producto, provocando muchas veces un retroceso a la versión anterior, en desmedro tanto económico como de imagen para la empresa proveedora del software. [4,5].

Los estudios actuales sobre el mantenimiento del software (S3M o SMmm) [6] se enfocan en un control profundo y documentado de las etapas del mantenimiento del software, un conjunto de normas de calidad contemplan este modelo [7,8,9]. Estas normas hacen hincapié en las capacidades de los intervinientes en el proceso, y destaca los elementos necesarios para garantizar la calidad en el SMmm, que constan en documentación exhaustiva y chequeo permanente de las etapas y resultados de la aplicación de instrumentos de medición sobre el mantenimiento.

El objetivo del proyecto es la obtención de metodologías que permitan resultados tangibles sobre el impacto y consecuencias de la implementación de nuevos requerimientos o en un software, proveyendo la posibilidad de efectuar cambios en los criterios de selección de estas nuevas funcionalidades para alcanzar

un rendimiento acorde a las necesidades de la empresa. Las grandes empresas desarrolladoras de software usan procedimientos con criterios de selección que cubren sus objetivos. Este tipo de clasificación debe ser referente para la creación de una metodología que se formalice y aplique en pequeñas empresas, habilitando procesos que documenten, regulen, controlen y midan los factores que afectan a la selección de los requerimientos, y las etapas posteriores de desarrollo e integración al sistema. Se procedió a la realización de un instrumento de captación de información que permitió la obtención de una muestra representativa de las empresas de la ciudad. Este constó de una encuesta de catorce preguntas cerradas, abiertas, booleanas y de múltiple Choice. Se obtuvo un volumen interesante de información acerca del tema, y la mayoría de las empresas participó, aun existiendo las dificultades comunes como falta de tiempo, responsabilidades o situaciones implícitas a las empresas de desarrollo de software, imposibilidad de brindar demasiada información respecto de los procesos internos de negocio, etc.

Primer análisis de la información obtenida

Se procedió a hacer un análisis descriptivo de las respuestas. Como resultado se obtuvieron las siguientes conclusiones acerca de la priorización de solicitudes de nuevos requerimientos:

- Casi la totalidad de las empresas encuestadas cuentan con productos bajo mantenimiento, con muy alta proporción de recepción de solicitudes de cambio, que pueden rechazarse. Estas son clasificadas con diferentes escalas y categorías, y utilizando técnicas variadas.
- Los criterios más usados para la clasificación es el de *prioridad*, cr*iticidad*, *urgencia*. También es utilizado el *control temporal* e *importancia*, dentro del contexto de la empresa. Muy pocas empresas aluden a la experiencia o la subjetividad de los superiores. Una sola empresa comenta sobre un *software de ingreso de requerimientos* sin referencias específicas.
- Más de un tercio de las empresas no realizan métodos de análisis del impacto o seguimiento, nombrando algunos el *testing* o *verificación*. Se destaca un pequeño porcentaje que utiliza *encuestas de aceptación o satisfacción* para captar el impacto sobre los clientes. Se mencionan la *herramienta de monitoreo del ciclo de vida,* el *registro histórico de cambios y evaluaciones* y la *encuesta de incidencia real sobre la modificación.*
- Finalmente en la sección de problemas relacionados con la implementación de nuevos requerimientos la mayoría de las empresas afirman su existencia, pero en este punto se observa la mayor dispersión destacándose el *no uso de la nueva implementación,* seguido de la *no aceptación del presupuesto,* y el *desarrollo interrumpido del requerimiento.*

Conclusiones preliminares

Del primer análisis se extrajo una serie de conclusiones preliminares que marcan las características más importantes a tener en cuenta para los pasos futuros de la investigación. El eje principal es la evidencia del uso individualizado de procedimientos para la priorización. Se observa una marcada tendencia al uso de métodos personalizados y no estandarizados para la clasificación de nuevas funcionalidades. Un alto porcentaje de las empresas sufren incidentes y problemas en la incorporación de nuevos requerimientos. Existe una tendencia a la priorización de las solicitudes por nivel de criticidad determinado de cada empresa, acentuándose los tiempos, la antigüedad del cliente, importancia del requerimiento a nivel funcional. Se destaca la falta de técnicas para el análisis del impacto de una implementación de nuevo requerimiento y seguimiento dentro del mismo proceso. Se consideran el *testing* y la *verificación* como métodos que miden el impacto y hacen el seguimiento. Hay inconstancia y una atenuación en cuanto a la formalidad en el tratamiento de estas implementaciones. Algunas empresas usan algún tipo de

mecanismo para obtener información sobre la implementación. La figura 1 muestra las variaciones obtenidas de una única pregunta

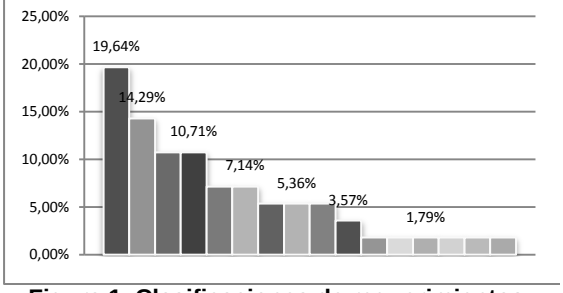


Figura 1. Clasificaciones de requerimientos.

abierta, sobre clasificaciones de requerimientos que se utilizan en la empresa.

La gran cantidad y variedad de información presente en las respuestas, sin tendencias ni medidas centrales, llevó a la decisión de continuar la búsqueda de ejes y líneas de comportamiento en la muestra. Por ello se inició una segunda etapa de análisis, centrada en la identificación de agrupamientos de los datos.

Segundo análisis: identificación de clusters

El análisis de detección de agrupamientos o *clustering* es una tarea que consiste en lograr agrupar objetos por su similitud, que está determinada de diferentes maneras, dependiendo de los tipos de las variables que los representan. Es utilizado en diversos campos de las ciencias, como en la recuperación de información, y en marketing y la minería de datos [10]. Existen una amplia variedad de algoritmos de obtención de clusters, aplicables según los tipos de variables. La elección del algoritmo correcto es fundamental para obtener buenos resultados, pero el análisis es iterativo; generalmente los primeros resultados son ineficientes, pero de su estudio se logran encontrar otras variables no definidas inicialmente en la muestra, que permiten el uso de otros algoritmos de agrupación. Se elige un software libre de minería de datos que cuenta con un conjunto importante de algoritmos de agrupación [11].

Clustering de variables booleanas

La muestra obtenida tiene en su mayoría variables booleanas y dos variables nominales. Por ello primeramente se seleccionan los algoritmos jerárquicos de agrupamiento con las variables booleanas. Se utiliza el *algoritmo aglomerativo*, que consiste en tomar de a un objeto e ir agregando los siguientes, formando los diferentes grupos. La similitud que se utiliza para agrupar los objetos es la medida de la *distancia* entre ellos en un mismo cluster y entre otros clusters. La distancia provista para el algoritmo aglomerativo es la llamada *distancia Euclidea o Euclideana:*

$$\|a-b\|_2 = \sqrt{\sum_i^k (a_i - b_i)^2}$$

El software cuenta con opciones de elección criterios de medición de las distancias, mínima distancia (Single linkage), máxima distancia (Complete linkage) y UPGMA ("Unweighted Pair Group Method with Arithmetic Mean" o Average link).

Los resultados de aplicación de los algoritmos muestran en todos los casos un desgranamiento en un número demasiado grande de clusters, con un máximo de 47 clusters. En la figura 2 se observa un gráfico KKlayout (Kamada Kawai Layout) que grafica los clusters unidos por *"resortes"* definidos por fuerzas de atracción

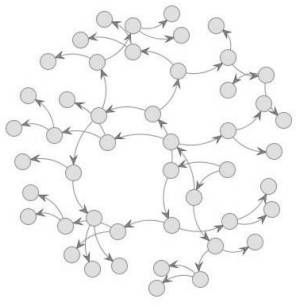


Figura 2. Algoritmo aglomerativo single linkage.

entre los grupos. Estas fuerzas de atracción son medidas en función de las similitudes de los objetos y de los clusters.

El resultado no brinda información relevante, ya que el software no logra obtener agrupaciones importantes, por eso se toma la decisión de disminuir la cantidad de variables booleanas. Se divide la muestra en tres partes, *políticas de priorización de requerimientos, tipos de clasificación de requerimientos y problemas relacionados a la implementación de requerimientos.* Los resultados muestran nuevamente una gran dispersión de objetos y

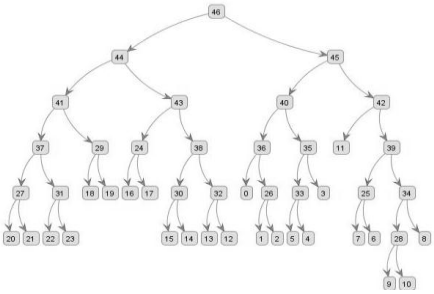


Figura 3. Complete linkage de Problemas relacionados a la implementación.

demasiados agrupamientos. Las figuras 3 y 4 son algunos de los árboles y *dendrogramas* resultantes de la aplicación del algoritmo en todas sus opciones para las tres categorías.

Clustering usando software alternativo

En un intento de encontrar alguna tendencia de agrupamiento, se procesan los datos binarios con un software alternativo de cálculo simbólico. La muestra resultante muestra 3 clusters de datos, caracterizados por la cantidad de respuestas positivas, siendo el tercer grupo un intermedio, que abarca las encuestas no decisivas.

Algoritmos de clustering por variables numéricas ordinales

Analizando los gráficos, el resultado del procesamiento del software de cálculo simbólico y el número de clusters obtenidos, es indicativa la necesidad de obtener una medida que permita una diferenciación en pocos grupos definidos. Entonces se construyen las siguientes variables: *afirmaciones sobre priorización, afirmaciones sobre tipos de*

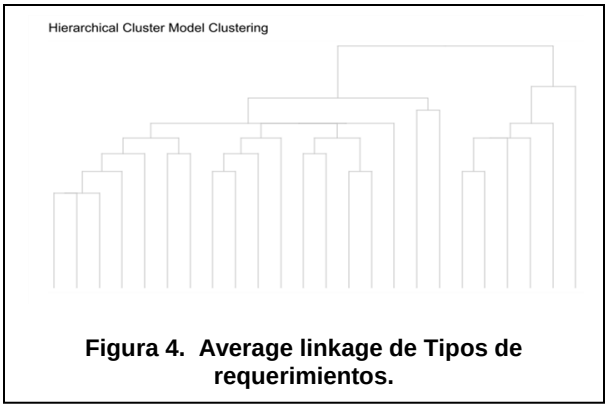


Figura 4. Average linkage de Tipos de requerimientos.

requerimientos y afirmaciones sobre problemas ocurridos. Como cada pregunta booleana está construida a partir de extractos de repuestas abiertas, de tal forma que al responder "true" implica participación, y a su vez son puntos que estarán incluidos en la o las nuevas metodologías productos de la investigación, estas nuevas variables miden el nivel de acercamiento a las mismas.

Se construye una nueva tabla con estas variables, y se selecciona el algoritmo K-means, que tiene como objetivo la división de *n* observaciones en *k* grupos en el que cada observación pertenece al grupo de tal forma que se minimiza la suma de los cuadrados dentro del mismo[12], obtenidos por la siguiente fórmula:

$$arg_s min \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

El algoritmo busca un punto central dentro de cada grupo, llamado "*centroide*". Este punto es equidistante de todos los objetos dentro del cluster [12]. Se aplica el algoritmo k-means a las variables construidas. Se toma cada una de las variables contra la variable nominal "*tipo de organización" (empresa comercial, servicios interna, servicios tercerizada)* y la variable booleana *"hace clasificación de requerimientos",* finalmente se corre el algoritmo con todas las variables juntas.

Resultados del algoritmo k-means

Los resultados obtenidos muestran distintivamente una subdivisión en tres grupos,

ratificando el resultado obtenido en el software alternativo. La primera corrida, que compara los tipos de clasificación con el tipo de organización muestra una clara relación entre las variables, mostrando la mayor cantidad de afirmaciones para las empresas del tipo Comercial. En la figura 5, que muestra una medición de los objetos con respecto a los centroides obtenidos, se observa la influencia del tipo de organización sobre la cantidad de afirmaciones, y la tabla de los centroides resultantes. De la misma forma se observa una influencia del uso de clasificaciones con respecto a la cantidad de problemas que sufrieron durante la implementación. Se puede

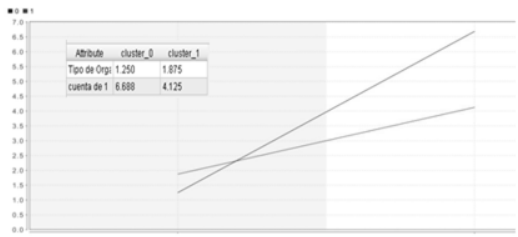


Figura 5. K-means de tipos de clasificaciones.

observar en la gráfica 6 la relación entre el tipo de organización y la ocurrencia de problemas en la implementación, el hecho de poseer algún método o procedimiento para priorizar no asegura la reducción de estos problemas si se excede en una clasificación exagerada.

De la observación de las gráficas k-means se puede inferir que las metodologías utilizadas por las empresas comerciales son más complejas e influyen en la cantidad de problemas que puedan surgir durante la implementación del nuevo requerimiento.

Metas a seguir

De la información obtenida en el primer análisis y los grupos y tendencias visualizados en el análisis clustering, se puede delinear una meta a futuro para el cumplimiento de los objetivos del proyecto, que es la construcción de una nueva metodología de clasificación, priorización, selección e implementación de un

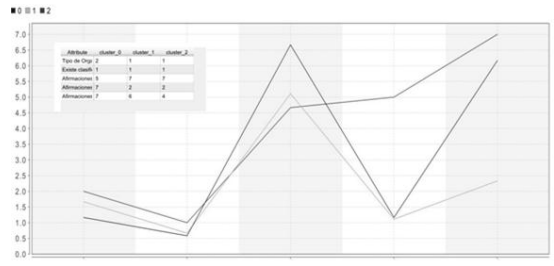


Figura 6. K- means de afirmaciones contra tipo de organización y clasificaciones.

nuevo requerimiento, que contenga procedimientos claros de ejecución, métodos de medición de valores representativos de los efectos y consecuencias que provoca esta implementación. Teniendo en cuenta los procesos más utilizados mundialmente para la priorización de requerimientos, la metodología puede brindar pautas para la toma de decisiones, para un uso óptimo de recursos humanos y materiales de la empresa, siendo un nuevo requerimiento una inversión monetaria de alto riesgo que puede generar pérdidas a corto y largo plazo. La construcción de la nueva metodología incluye los siguientes pasos:

- Creación de criterios de clasificación de Tipos de requerimientos y Tipos de clientes.
- Creación de métricas de medición de la solicitud y estimaciones preliminares, por ejemplo: Historial del cliente, Nivel de riesgo, cantidad de departamentos y personal involucrados, cantidad horas/hombre estimadas.
- Creación de valores e índices de interés para la empresa para medición del impacto por ejemplo: ingresos obtenidos, niveles de beneficio, solicitudes presupuestadas canceladas, niveles de uso de los requerimientos incorporados.
- Creación de métricas de medición para el seguimiento durante el ciclo de vida del nuevo requerimiento: horas reales insumidas, personal ocupado, problemas o situaciones con el cliente durante el desarrollo, impacto sobre el sistema, bugs ocurridos, nivel de dificultad en el

desarrollo, documentación para el historial del cliente, etc.
- Creación de una o varias metodologías de asignación de prioridades a una cola de requerimientos tratando de maximizar los valores e índices de interés para la empresa.
- Formulación de los posibles métodos y procedimientos necesarios para la incorporación de una metodología al proceso del mantenimiento del software.

A su vez las nuevas metodologías se simularan en un entorno de desarrollo comercial, los criterios de clasificación serán los más votados y que permitan evitar la mayoría de los problemas que se comentaron.

Conclusiones

De la primera etapa del proyecto se obtuvo una muestra que posee información altamente calificada, representativa del mercado del software de la ciudad, clara, detallada y característica de las empresas. La precisión en la construcción de las metas a seguir como líneas de la investigación futura permiten concluir que los datos obtenidos son de muy alto nivel y conforman una base sólida para la construcción de las etapas siguientes, dando posibilidad a la obtención de resultados positivos y fundamentados. La muestra de datos es una fuente de recurrencia permanente para revisión y control de los pasos futuros.

Agradecimientos

Agradecemos a todos los involucrados que participaron desinteresadamente en esta recolección de datos, los compañeros de trabajo, de estudios, los empleados de las empresas, de alta participación en el mercado local y mundial, que atendieron solícitamente a las peticiones, los superiores y jerárquicos que aceptaron compartir información referente. También agradecemos a los directivos del Departamento ISI y Secretaría de Ciencia y Técnica de la Universidad Tecnológica Nacional Facultad Regional Rosario, por permitir la realización de este proyecto.

Referencias

[1] S. Dart, A. Christie, A. Brown: “A Case Study in Software Maintenance”, Technical Report CMU/SEI-93-TR-08 ESC-TR-93-185 CMMI Examples, November 1993

[2] G. Canfora. A. Cimitile: “Software Maintenance”, University of Sannio, Faculty of Engineering at Benevento Palazzo Bosco Lucarelli, Piazza Roma 82100, Benevento Italy, 29 November, 2000

[3] P. Berander, A. Andrews: “Requirements Prioritization”, School of Engineering at Blekinge Institute of Technology in Sweden - Software Engineering at Washington State University. 2002

[4] K. Erdil: et al, “Software Maintenance As Part of the Software Life Cycle” Department of Computer Science Tufts University, 2003

[5] C. López Pérez: “Modelo de Madurez de la Capacidad del Software”, InforMAS Revista de Ingeniería Informática del CIIRM, Ilustre Colegio de Ingenieros en Informática de la Región de Murcia, España. 2004

[6] A. April, J. Desharnais: “Software Maintenance Maturity Model (SMmm): A Software Maintenance Process Model”, Department of Software Engineering École de technologie supérieure 1100 Notre-Dame West, Montréal, Québec, Canada. 2006

[7] L. Gray: “Using MIL-STD-498 and ISO/IEC 12207 for OO and RAD”, Ada PROS Inc, Fairfax, Virginia, USA, 1995

[8] L. Gray.:“A Comparison of IEEE/ EIA 12207, ISO/IEC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers”, Abelia Corporation, Fairfax, Virginia, USA. 1999

[9] R. Franz.: “The Webok 2nd edition”, IEEE Communications Magazine, pag 12, 2011.

[10] Garre, M., Cuadrado, J., Sicilia, M.,Rodriguez, D., Rejas, R., "Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software", Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.3, No1,2007

[11] RapidMiner Documentation, 2014 http://rapidminer.com/documentation/

[12] Larrañaga, P., Inza, I.,Mouhajid, A., "Clustering", Departamento de Ciencias de Computacion e Inteligencia Artificial, Universidad del Pais Vasco-Heuskal Erriko Universitatea, 2007

Datos de Contacto:
Sara Ester De Federico, Universidad Tecnológica Nacional, Facultad Regional Rosario, Zeballos 1341 Rosario, Santa Fe, Argentina, saraedf@gmail,com

Cuadrantes de Testing Ágil

Fernando Martín Córdoba
Departamento Ingeniería en Sistemas de la Información
Universidad Tecnológica Nacional, Facultad Regional Villa María
Av. Universidad 450, Villa María, Córdoba
mcordoba@frvm.utn.edu.ar

Abstract

Agile testing is based on the Agile Manifesto, aimed at delivering quality software to meet the needs of the interested companies using different levels and types of testing adapting agile development methodologies. This paper seeks to make a base Agile Testing to guide future work. The research was based on two questions: How can the different Testings in Agile methodology be arranged? And, How to implement Agile testing?. These questions had their origin in the interest to go deeper on the issue of implementing the use of Agile Testing, since the trend of agile development is gaining ground in the Software Development Industry, though many companies are not implementing this methodology yet. Data was collected in order to know if city companies were implementing agile testing. There was an unfavorable response to this research which led to emphasize this paper on the above mentioned topic.

Due to this, experience from the academic to the city companies will be transmitted.

Palabras Clave

Testing ágil, pruebas ágiles, cuadrante ágil, automatización de pruebas, TDD, BDD, pruebas unitarias, pruebas de componentes, pruebas de aceptación, pruebas funcionales, Pruebas exploratorias, pruebas de seguridad.

1. Introducción

Las técnicas de pruebas ágiles son relativamente nuevas, pero ya existe una propuesta [1]. Hacia el año 2001 se presentó el manifiesto ágil [2], que introduce los valores y los principios que deben adoptar los miembros de un equipo ágil para llevar a cabo el desarrollo adecuado y rápido de una aplicación. Entre ellos, se destaca la prioridad de satisfacer al interesado por medio de las entregas tempranas de software; la colaboración continua entre el equipo interesado y el equipo desarrollador y la mejora continua en el proceso, porque se responde al cambio y a las nuevas expectativas del interesado. Algunas organizaciones enfrentan un cambio cultural al tratar de tomar estas metodologías, por lo cual esta idea no se puede aplicar a todas las organizaciones. Aquellas que tienen un pensamiento independiente, no tienen establecida una estructura jerárquica y tienen políticas prioritarias de calidad, son más adecuadas para asumir un pensamiento ágil [3].

Dentro de un equipo ágil existe comunicación continua entre los desarrolladores e interesados, definiendo los requisitos, las pruebas y el comportamiento esperado del producto. Además, se manifiesta el concepto de roles, donde se intenta que cada miembro posea un buen número de roles para eliminar la dependencia entre individuos, promoviendo siempre a todos el principio de calidad en sus actividades diarias. En este marco, en un equipo ágil todos los individuos tienen habilidades de testers como establece Talby [4].

Un tester ágil se caracteriza por tener, además de habilidades técnicas de testing, destreza para integrar las necesidades del interesado al equipo de desarrollado, definiendo las historias del usuario en requisitos y, también, transformándolas en parámetros de prueba cuando se codifiquen. El rol del tester sobresale por colaborar con todos los integrantes del equipo.

Los conceptos y definiciones que giran alrededor del testing ágil son:

Equipo ágil: conjunto de personas expertas en áreas de software y en el dominio del negocio, que trabajan en pequeños ciclos de tiempo, llamados sprint. Los sprint tienen fases: análisis, diseño, implementación y pruebas.

Metodología ágil: proceso de desarrollo de software que busca construir aplicaciones de manera rápida teniendo en cuenta al interesado para la mayoría de los pasos que el método contiene. Las metodologías ágiles enfatizan en las comunicaciones cara a cara en vez de la documentación.

Prueba: procedimiento para validar una funcionalidad de la aplicación de software. Existen varios tipos de

prueba, entre ellas de funcionales, de exploración, de regresión entre otras.

Xunit: entorno utilizado para automatizar las pruebas unitarias. Actualmente se vienen desarrollando frameworks para su implementación.

Sistema de seguimiento de fallos: herramienta utilizada para el seguimiento de errores manejada por los equipos de desarrollo para conocer qué fallos se deben corregir y cuáles no.

Es importante dejar en claro que la codificación y las pruebas constituyen un solo proceso integrado y, por ello, el tiempo empleado en codificación puede ser el mismo a emplear en las pruebas. Es por ello que la experiencia de los testers se hace de vital importancia a la hora de definir requisitos del cliente y su estimación de codificación y pruebas, porque el nombre ágil no significa entregar productos rápidamente con la funcionalidad solicitada, sino que, además, deben ser confiables. El diseño de pruebas, como tal, y el uso de las herramientas adecuadas se vuelven vitales para asegurar el principio de calidad. La planeación y el diseño de pruebas es importante a la hora de distribuir tiempo entre:, la prueba en particular, arreglar bugs y realizar otro tipo de pruebas como pruebas de aceptación, pruebas de regresión, pruebas por unidad, pruebas por componente y pruebas de exploración.

El factor de éxito de las pruebas ágiles es saber automatizar las pruebas de regresión, las cuales aseguran calidad y precisión sobre todo el comportamiento del sistema.

2. Antecedentes

El testing es una actividad cognitiva y no mecánica ni repetitiva que involucra varias funciones mentales como el lenguaje, la imaginación, percepción, entre otros. Un tester va desarrollando estas habilidades a medida que avanza en el testing y obtiene conocimiento y experiencia.

El testing lo podemos observar de 4 aspectos diferentes: Orientado al negocio, Orientado a la tecnología, Soporte al equipo y Crítica al producto (La palabra "crítica" no en un sentido negativo. Una crítica puede incluir elogios y sugerencias de mejora) [5].

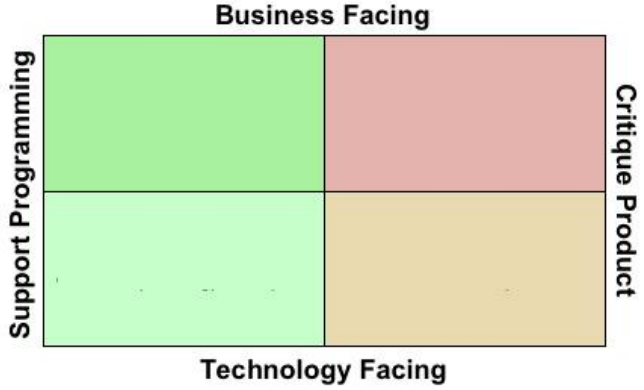


Figura 1 - Cuadrantes de Metodología Ágil [6]

Cada cuadrante de los mostrados en las figuras 1 y 2 representan diferentes pruebas que se pueden ejecutar. Uno de los ejes divide la matriz en los test que apoyan al equipo del lado izquierdo, y las pruebas que critican al producto del lado derecho. El otro eje divide las pruebas orientadas al negocio en la parte superior, y en la inferior las orientadas a la tecnología.

Los cuadrantes nos ayudan a identificar los diferentes tipos de pruebas que necesitamos para utilizar con el fin de ayudar a la codificación.

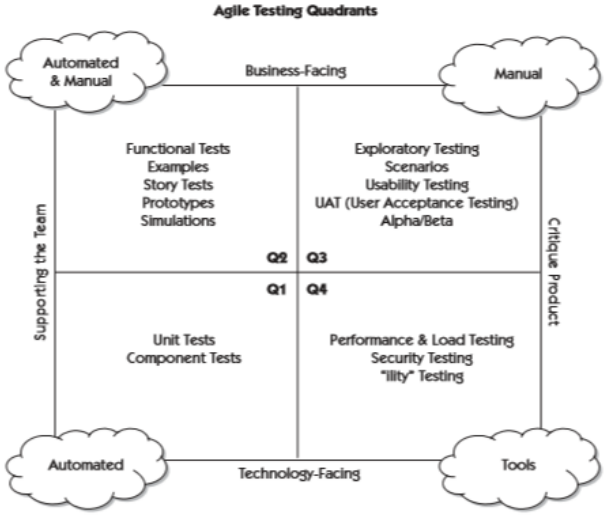


Figura 2 - Matriz de Metodología Agil

2.1 Cuadrante 1

El cuadrante inferior izquierdo. (Soporte al Equipo y Orientado a la Tecnología) representa el desarrollo basado en pruebas (TDD) o desarrollo basado en comportamiento (BDD), que es una práctica de desarrollo ágil. Actualmente se está trabajando con el grupo de Investigación de la Facultad en estas dos metodologías, buscando las diferencias y similitudes entre ambas, y se implementó la práctica de una historia de usuario relacionada con mensajes de bienvenida a los usuarios que se loguen en el sistema que forma parte del Sprint 1. Obteniendo como resultado la Figura N° 3 [7]



Figura N°3: Corrida Prueba

Las Pruebas Unitarias verifican la funcionalidad de un pequeño subconjunto del sistema, tal como un objeto o método. Pruebas de Componentes verifican el comportamiento de una parte más grande del sistema, tal como un grupo de clases [8]. Ambos tipos de pruebas suelen automatizarse con un miembro de la familia xUnit (Junit para Java – Nunit para .net – PyUnit para Python, entre otras).
Cuando falla una prueba de unidad, el programador puede identificar el problema rápidamente y resolver el problema con la misma rapidez.
Las pruebas unitarias y de componentes entran en este cuadrante, están automatizadas y se escriben en el mismo lenguaje de programación que la aplicación. Un experto en negocios probablemente no los pueda entender mediante la lectura directamente, pero estas pruebas no están destinados para uso del cliente. De hecho, la calidad interna no se negocia con el cliente.

2.2 Cuadrante 2

Las pruebas en el Cuadrante 2 también apoyan el trabajo del equipo de desarrollo, pero a un nivel más alto. Están ubicadas en el eje (orientadas al negocio y apoyo al equipo), definen la calidad externa y las características que los clientes quieren.
Con el desarrollo ágil, estas pruebas se derivan de ejemplos proporcionados por el cliente, ellos describen los detalles de cada historia. Están escritos en forma de negocios para que puedan entender fácilmente utilizando el lenguaje de dominio del negocio.
La mayor parte de las pruebas comerciales que apoyan el equipo de desarrollo deberían ser automatizadas. Se deben ejecutar con frecuencia para darle al equipo realimentación en caso de cambio inesperado. Todas estas pruebas se deben ejecutar en el marco de una integración continua automatizada.

Las pruebas de los cuadrantes 1 y 2 están escritos para ayudar al equipo a entregar lo solicitado por los clientes. Verifican que la lógica de negocio y las interfaces de usuario se comporten de acuerdo a los ejemplos proporcionados por los clientes.

2.3 Cuadrante 3

Cuando los programadores escriben código de prueba, podrían no estar entregando lo que el cliente realmente quiere.
Ahí es donde las pruebas para criticar el producto y las pruebas orientadas al negocio entran en juego, para ver si cumple con las expectativas o no. Cuando hacemos pruebas de negocio para criticar el producto, tratamos de emular la forma en que un usuario real va a utilizar la aplicación. Tenemos que utilizar nuestros sentidos, nuestro cerebro y nuestra intuición para comprobar si el equipo de desarrollo proporciona valor de negocio requerido por los clientes.
A menudo, los usuarios y los clientes realizan este tipo de pruebas. Pruebas de aceptación del usuario ofrece a los clientes la oportunidad de ver qué cambios podrían querer en el futuro, y es una buena manera de reunir nuevas ideas para nuevas historias. La prueba de aceptación podría ser un paso necesario en la aprobación de las historias terminadas.
Las pruebas de usabilidad es otro ejemplo de un tipo de prueba puede incluir la navegación de una página a otra o incluso algo tan simple como el orden de tabulación.
El testing exploratorio es central en este cuadrante. Durante las sesiones de pruebas exploratorias, el probador diseña y realiza pruebas de forma simultánea, utilizando el pensamiento crítico para analizar los resultados. Esto ofrece una mejor oportunidad de aprender acerca de la aplicación. No estamos hablando acerca de pruebas ad hoc, que es improvisada. El testing exploratorio es un enfoque más reflexivo y sofisticado que las pruebas ad hoc. Se guía por una estrategia y opera dentro de limitaciones definidas. Desde el inicio de cada historia, los testers empiezan a pensar en escenarios que quieran probar, analizar los resultados de las pruebas, y a medida que aprenden, pueden incursionar en nuevas áreas para explorar. El testing exploratorio funciona de la misma manera que los usuarios finales. Testers usan su creatividad y su intuición. Como resultado, con este tipo de pruebas se encuentran muchos errores graves.
El grupo de Investigación del Dpto. de Ingeniería en Sistemas de la Facultad, definió y trabajo con el Testing Exploratorio, se realizaron pruebas de este tipo en la página web http://www.nba.com (está página fue elegida buscando una que tuviese gran cantidad de accesos, que sea internacional y gozara de prestigio) se detectaron con este tipo de prueba, varias fallas, las mismas fueron documentadas a través de la herramienta Jing (software libre para capturar imágenes y filmar video) se utilizó en

todos los casos esta última opción para tener una evidencia más clara y precisa de las fallas detectadas. Es importante mencionar que las pruebas exploratorias siguen un procedimiento de los ítems a probar, ya que de no hacerlo así era después difícil recordar toda la secuencia de pasos para repetir dicho error. Las distintas personas del equipo escogían que parte iban a explorar y el tipo de profundidad, si eran solo errores de Interfaz o errores en el desarrollo de la página, de funcionalidad, etc. En todos los casos se indicaba el ambiente en el cual se hizo el desarrollo, tipo y versión de Sistema operativo, de Browser y versión con el que se probó. A continuación se adjunta la figura N°4 con evidencia de dos fallas detectadas en el sitio antes mencionado, se detallan a continuación los pasos que se siguieron: 1° se accedió a la URL: http://www.nba.ccom/global, luego se accedió al link "Login" situado en la parte superior izquierda, luego se accedió al link que dice "JOIN NOW, IT´S FREE!" al acceder al formulario se completó la primer parte con los datos pertinentes, al pasar a la segunda parte, donde pide completar el dato de fecha de nacimiento "Date of Birth" se produce la falla que esa fecha puede ser superior a la fecha actual (Figura N°4). Otra falla en el mismo campo es que al querer corregir algún dato y desplazarse con los cursores, el mismo se iba siempre al final del campo y no permite desplazarse, sucedió con los navegadores Internet Explorer y Chrome, no sucediendo lo mismo con Mozilla Firefox.

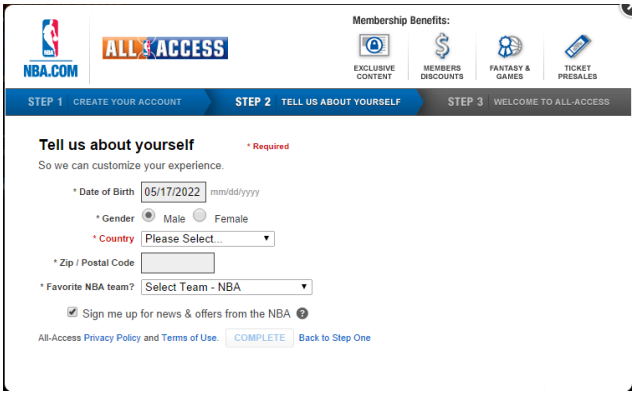


Figura N° 4 – Fallo detectado con Testing Exploratorio

2.4 Cuadrante 4

Los tipos de pruebas que caen en el cuarto cuadrante son también críticas. Son pruebas orientadas a la tecnología y destinadas a criticar las características del producto, como el rendimiento, robustez y seguridad.
Los integrantes del equipo del cliente a menudo asumen que el equipo se hará cargo de temas tales como la velocidad y la seguridad.
Si conocemos los requisitos de rendimiento, seguridad, la interacción con otros sistemas y otros atributos no funcionales antes de comenzar la codificación, es más fácil de diseñar y programar con eso en mente. Algunos de estos pueden ser más importantes que la funcionalidad en sí misma. Las herramientas de automatización se pueden utilizar para crear los datos de prueba, establecer escenarios, conducir pruebas de seguridad, y ayudar a dar sentido a los resultados. La automatización es obligatorio para algunos esfuerzos, como las pruebas de carga y rendimiento.
No dejar actividades esenciales, como la carga o la facilidad de uso de pruebas para el final, cuando tal vez sea demasiado tarde para rectificar problemas.
La información producida durante la prueba para revisar el producto debe ser alimentado de nuevo en el lado izquierdo de la matriz y se utiliza para crear nuevas pruebas para impulsar el desarrollo futuro.
Las cortas iteraciones de desarrollo ágil dan al equipo la oportunidad de aprender y experimentar con los diferentes cuadrantes de prueba. Se puede utilizar los cuadrantes de pruebas ágiles para ayudar a asegurar que todas las pruebas necesarias se realizan en el momento adecuado.

3. Trabajos Futuros

Se propone como trabajo futuro de Investigación "Tipos de Pruebas para Testing Ágil", tomando como base el presente trabajo, focalizar en los tipos de pruebas que se identificaron como de mayor utilidad Pruebas Unitarias y de Componentes como por ejemplo TDD, BDD y por otro lado Testing Exploratorio aplicado a desarrollos de la Facultad.

4. Conclusión

En este paper se presentó un resumen del Trabajo de Investigación, que está en una etapa inicial, abordando el área de Testing y especialmente el ágil, ya que notamos que aún hay mucho camino por recorrer, las empresas de desarrollo de software deben aún realizar el período de transición hacia una metodología ágil, así lo indica la tendencia de la Industria en nuestra Región.
Y retomando las preguntas que se realizaron cuando iniciamos la investigación, podemos decir:
1. ¿Cómo se pueden organizar los diferentes Testing en metodología ágil?
Una de las formas que se estudiaron y se hizo foco en el trabajo fue la organización en cuadrantes, lo que permite tener un mejor contexto global de las diferentes tipos de pruebas y su impacto entre la tecnología, el negocio, el equipo de trabajo y el producto en sí. De esta manera dependiendo el foco que quiera obtener la Empresa de desarrollo, puede seguir el o los cuadrantes que más se adapten a sus necesidades. Probar todo el sistema en un 100% es una tarea difícil de alcanzar por los tamaños de los mismos, es por ello que buscamos a través de los cuadrantes guiar a los equipos a que elijan el mejor camino y utilicen sus esfuerzos en lograr aplicar los tipos

de pruebas que mejor le resulten. En nuestro caso estos tipos de Testing fueron BDD (Desarrollo Basado en Comportamiento) del Cuadrante 1 y Testing Exploratorio del Cuadrante 3. Tienen diferentes visiones pero se complementan y se adaptan a las metodologías ágiles.

2. ¿Cómo implementar testing ágil?

Continuando con la pregunta anterior, BDD nos demandó más tiempo, requiere tener un conocimiento más amplio y un cambio en la forma de programar, pero es posible llevarlo a cabo. En el Testing Exploratorio, hay que ser persistente, paciente, intuitivo, ordenado, pero es una técnica que se puede complementar con otras y obtuvimos buenos resultados (encontrar fallas) que es el fin en sí mismo de realizar un Testing exitoso.

También podemos decir que es un cambio progresivo, que demanda su tiempo y esfuerzo, va a depender en gran medida del involucramiento de toda la Organización. Las pruebas Unitarias son el puntapié inicial para comenzar con este apasionante tema.

Este enfoque ágil también permite incorporar la calidad desde el principio, no esperando hasta el final. El equipo de pruebas es parte integral del equipo. No es un grupo de personas que viene a posteriori a validar lo que hemos hecho, pertenecen integralmente al equipo que va a construir el producto desde el principio

5. Referencias

[1]S. Shaye 2008. Transitioning a team to agil test methods, agile conference 2008.

[2] Manifiesto for Agile Software Development, http://agilemanifiesto.org.

[3] L. Crispin amd J. Gregory v2009. Agile Testing: a practical guide for testers and agile teams,Ed. Addison-Wesley Professional, Boston.

[4] D. Talby, A. Karen, O. Hazzan, Y., and Dubinsky, 2006. Agile Software Testig in a large-scale project, IEEE Software, Vol. 23. N°4 pp. 30-37.

[5] Marick, Brian. Everyday Scripting with Ruby: For Teams, Testers and you. Pragmatic Bookshelf, 2007.

[6] Marick, Brian. "My Agile testing Project" http://www.exampler.com/old-blog/2003/08/21/

[7] "Testing Ágil: Presente y Futuro" Trabajo de Investigación. Dpto. Sistemas de la Información UTN Fac. Reg. Villa María

[8] Meszaros, Gerard. XUnit Test Patterns: Refactoring Test Code, Addison- Wesley, 2007

Un Análisis sobre la Asignación de Recursos en Procesos de Negocio

Claudio Barua, Daniel Romero y Cynthia Villalba
Núcleo de Investigación y Desarrollo Tecnológico
Facultad Politécnica, Universidad Nacional de Asunción, Paraguay
{cbarua, dromero, cvillalba}@pol.una.py

***Resumen*—Cada vez más empresas están implementando sistemas de flujo de trabajo y de gestión de procesos de negocio, debido a que estos sistemas permiten mejorar la productividad y reducir los costos operativos. Para el modelado y la ejecución de procesos de negocio, BPM y *Workflow* son dos estándares de *facto* utilizados hoy en día. Un aspecto importante en estos sistemas es la gestión de los recursos y el impacto de éstos en el proceso, área en la cual no se han puesto de acuerdo los investigadores y que presenta desafíos a ser abordados en trabajos futuros.**

En este artículo se discute la asignación de recursos en sistemas BPM y *Workflow*. Se presenta una revisión de los principales trabajos relacionados a asignación de recursos, los cuales describen aspectos como: tipo de recursos (humanos o no humanos), disponibilidad (limitados o no limitados), categoría (primarios o adicionales) y la asignación de esos recursos en las tareas. Además, se presenta un análisis comparativo de cuatro herramientas relacionadas con BPM y *Workflow*. El análisis considera la metodología utilizada en la asignación de los tipos de recursos durante el modelado y la ejecución de los procesos.

I. Introducción

La implementación de los sistemas de flujo de trabajo y de gestión de procesos de negocio ha tenido un aumento considerable en los últimos años debido a su potencial para aumentar la productividad y ahorrar costos [6], [9], además de mejorar aspectos como: la capacidad de dirección de las organizaciones, los tiempos de gestión, la administración de los recursos, la interacción con los clientes, la visibilidad del negocio y la calidad de la documentación.

Un flujo de trabajo o *Workflow* consiste en un conjunto de tareas coordinadas y diseñadas para llevar a cabo un proceso complejo, por ejemplo, el tratamiento de pacientes en un hospital [23], [24]. Durante este proceso los documentos, informaciones y tareas pasan de un participante a otro para una acción, de acuerdo con un conjunto de reglas de procedimiento. En el Modelo de Referencia [10] de la *Workflow Management Coalition (WfMC)*[1], expresa que un Sistema de Gestión de Workflow (WfMS) es "un sistema que completamente define, gestiona y ejecuta los flujos de trabajo a través de la ejecución de un programa cuyo orden de ejecución es impulsado por una representación computarizada de la lógica de flujo de trabajo". Los recursos que ejecutan las tareas de un sistema de gestión de flujo de trabajo pueden ser personas, máquinas o software, de ahí que la asignación y planificación de recursos son fundamentales en estos sistemas, principalmente debido a que esos recursos pueden ser limitados.

[1] www.wfmc.org

De acuerdo con van der Aalst et al. [23] [24], la Gestión de Procesos de Negocio (BPM[2]) es un conjunto de disciplinas empresariales que combinan el conocimiento de la tecnología de la información junto con las ciencias de gestión mediante un conjunto de métodos, técnicas y herramientas que permiten el diseño, ejecución, gestión y análisis de procesos de negocio operacionales. En este contexto, un sistema BPM (BPMS) puede ser visto como una extensión de un sistema *Workflow*, ya que un WfMS se centra principalmente en la automatización de procesos de negocio, mientras que un BPMS tiene un alcance más amplio que va desde la automatización y el análisis de los procesos hasta la gestión de operaciones y la organización del trabajo [24].

En este trabajo se identifican las diferentes formas en que se modelan los recursos en sistemas BPM y *Workflow*. Se considera la clasificación de recursos dada en [16] y [18], las cuales distinguen recursos humanos y no humanos, que pueden ser limitados o no, consumibles o duraderos. Se abordan diferentes trabajos que presentan *frameworks*, modelos y algoritmos que permiten mejorar la eficiencia en la asignación de recursos. Además, se analizan las metodologías utilizadas por algunos sistemas BPM y *Workflow* para realizar las asignaciones de recursos a las distintas actividades a ser ejecutadas en los procesos de negocio.

El resto del artículo está organizado como sigue. En la Sección II se describen los trabajos relacionados al modelado y asignación de diferentes tipos de recursos en procesos de negocio. La Sección III presenta cuatro herramientas que permiten el modelado y ejecución de procesos de negocio, y cómo los recursos son considerados en las mismas. En la Sección IV se expone un análisis comparativo de las herramientas examinadas. Finalmente, en la Sección V, se presentan las consideraciones finales y los trabajos futuros.

II. Estado del Arte en la Asignación de Recursos en Procesos de Negocio

Para el modelado y ejecución de los sistemas BPM y *Workflow* se utilizan distintos lenguajes, cada uno con caraterísticas diferentes. Los más utilizados son: BPMN[3] (Business Process Model and Notation), estandarizado por la *Object Management Group* (OMG[4]); XPDL[5] (XML Process Definition Language)

[2] BPM, *Business Process Management*
[3] www.bpmn.org
[4] www.omg.org
[5] www.xpdl.org

de la WfMC; WS-BPEL (Web Services Business Process Execution Language), o simplemente BPEL, de la *Organization for the Advancement of Structured Information Standards* (OASIS[6]); y YAWL[7] (Yet Another Workflow Language) dado en [8]. Estos lenguajes se centran en el control de flujos más que en la perspectiva de los recursos [17].

La *Workflow Patterns Initiative*[8] tiene como objetivo proporcionar una base conceptual para la tecnología de proceso. Entre las áreas analizadas se encuentran: la perspectiva de flujo de control, que capta los aspectos relacionados con el control de flujo de las dependencias entre las distintas tareas; la perspectiva de datos, que trata el paso de la información, el ámbito de las variables, entre otros; la perspectiva para el manejo de excepciones, que está relacionada con las diferentes causas de las excepciones y las diversas acciones que deben ser tomadas como resultado de las excepciones que se producen; y la perspectiva de los recursos, que tiene en cuenta los recursos asignados a las tareas.

Rusell et al. [18] [19] extendieron la *Workflow Patterns Initiative* para incluir patrones desde la perspectiva de recursos. Esta perspectiva se centra en el modelado de los recursos y su interacción con los sistemas de información consciente de los procesos. Los recursos se dividen en dos grupos: humanos y no humanos. Los recursos no humanos fueron agrupados en dos categorías: recursos duraderos, aquellos cuya capacidad no se ve afectada por la cantidad de trabajo realizado, y recursos consumibles, aquellos que se consumen, parcial o totalmente, durante la realización del trabajo. Además, identificaron un conjunto de patrones de recursos que capturan las diversas formas en que son representados y utilizados los recursos en los flujos de trabajo. Delimitando estos patrones, de forma independiente a las tecnologías específicas de flujo de trabajo y del lenguaje de modelado, pudieron ofrecer un análisis completo desde la perspectiva de los recursos.

El modelado de recursos está fuera del ámbito de BPMN, como se indica en la especificación de la OMG [15]. Es por ello que Awad et al. [1] extendieron el metamodelo de BPMN para incluir las restricciones en la asignación de recursos. La extensión fue orientada a los seres humanos como recursos que llevan a cabo el trabajo en un proceso. Eligieron el lenguaje OCL (*Object Constraint Language*) para expresar formalmente estas restricciones.

En otro trabajo, Stroppi et al. [22] propusieron una extensión al lenguaje BPMN 2.0 para apoyar la definición de requerimientos desde la perspectiva de los recursos en los modelos de procesos de negocio, con el propósito de mejorar la comunicación de los requisitos entre los analistas de negocios y desarrolladores técnicos. La extensión propuesta considera tres aspectos desde la perspectiva de los recursos: la estructura de recursos, la distribución de trabajo y la autorización.

Los sistemas BPM y *workflow* actuales, en general sólo consideran los recursos humanos debido a que los mismos modelos de procesos de negocio se abstraen de representar otros tipos de recursos. Además, estos sistemas sólo permiten recursos individuales, dejando de lado recursos múltiples o tipos de recursos no humanos utilizados en los procesos de negocio y necesarios para cumplir con una tarea. Para atenuar esta limitación y extender la perspectiva de recurso actual, Ouyang et al. [16] desarrollaron un modelo de datos conceptual que tiene en cuenta dos categorías de recursos: humanos y no-humanos, subdividiendo nuevamente a estos últimos en aplicaciones (software o web-service) y no-aplicaciones (consumibles o duraderos, como en [18]). Para capturar la forma en que múltiples recursos son asignados a una misma tarea introdujeron en el modelo desarrollado el concepto de recursos primarios y adicionales. Según este modelo, sólo los recursos humanos y aplicaciones pueden ser asignados a recursos primarios para llevar a cabo la gestión de una tarea y asegurarse de que todos los recursos adicionales están disponibles. Sin embargo, cualquier recurso puede ser parte de los recursos adicionales.

En [13], Li et al. analizan los recursos restringidos en las especificaciones de un *workflow*. En el trabajo, los autores proponen cuatro algoritmos para la verificación de la consistencia de los recursos en la especificación del *workflow*. Estos algoritmos permiten realizar análisis estático en las especificaciones del flujo de trabajo en tiempo de diseño del modelo, pero no tienen en cuenta el análisis dinámico de los flujos de trabajo en tiempo de ejecución. Además, realizan un análisis de las especificaciones de flujo de trabajo con tiempo, donde el método de verificación de la consistencia de los recursos es revisado con respecto a la información de tiempo en la especificación del flujo de trabajo.

Senkul et al. [20] desarrollaron un *framework* para los flujos de trabajo cuya exactitud está dada por un conjunto de restricciones en la asignación de recursos. Además, desarrollaron técnicas para la planificación de tareas en este tipo de *workflow*. El *framework* integra Transacciones Lógicas Concurrentes (CTR[9]) con Programación Lógica con Restricciones (CLP[10]) que llamaron Transacción Lógica Concurrente con Restricciones (CCTR[11]). En este proceso, el usuario especifica el flujo de trabajo usando CTR y un conjunto de posibles agentes para cada tarea, con sus costos. Entonces, un algoritmo de transformación devuelve un nuevo flujo de trabajo en CCTR com restricciones de asignación de recursos que satisfacen las restricciones originales.

En otro trabajo similar, Senkul y Toroslu [21] presentaron una arquitectura para modelar y planificar los flujos de trabajo bajo las restricciones de asignación de recursos y las limitaciones temporales y de causalidad. La arquitectura incluye un lenguaje de especificación con la capacidad de expresar los recursos y las restricciones en la asignación de los mismos. Además, se desarrolló un algoritmo que devuelve un nuevo flujo de trabajo y una asignación de recursos que garantizan las restricciones iniciales.

Por otra parte, Delias et al. [5], a través de una estrategia a la que llamaron Optimización Conjunta de Recursos en Conflicto (Re.Co.Jo.Op.[12]), abordaron el problema de la asignación de recursos junto con la programación de flujo de trabajo, esto es debido a que para ellos el resultado final es, al mismo tiempo, una planificación de procesos y un sistema de reserva

[6] www.oasis-open.org
[7] www.yawlfoundation.org
[8] http://www.workflowpatterns.com/

[9] CTR, *Concurrent Transaction Logic*
[10] CLP, *Constraint Logic Programming*
[11] CCTR, *Concurrent Constraint Transaction Logic*
[12] Re.Co.Jo.Op., *Resource Conflicts Joint Optimization*

de recursos. Además, optimizaron dos factores críticos en la gestión de recursos, el conflicto de recursos y la utilización de los mismos. El método desarrollado puede ser aprovechado para evaluar la cantidad mínima de recursos necesarios para la promulgación del flujo de trabajo apropiado en la fase de diseño del sistema. Para aplicar este método en tiempo de ejecución, propusieron una arquitectura basada en agentes intermediarios que manejan las decisiones de asignación de los recursos registrados.

Liu et al. [14] desarrollaron un *closed-loop workflow framework* para tratar el problema de la asignación de recursos y mejorar la productividad de la gestión de recursos de flujos de trabajo. Los autores propusieron un enfoque basado en asociación de reglas de minería de datos para determinar la lógica entre los recursos y las actividades del flujo de trabajo, lo que ayudaría en la toma de decisiones en el momento de la asignación de recursos permitiendo una gestión más inteligente y precisa de los mismos.

Xu et al. [25], a su vez, propusieron un enfoque que optimiza el uso de recursos en una empresa explorando características estructurales de los procesos de negocio y adaptando las estructuras de estos procesos de negocio para asignar mejor los recursos disponibles. El trabajo propuso un modelo basado en roles para especificar la relación entre los procesos de negocio, las tareas, las funciones y los recursos. Sobre la base del modelo presentado, se estableció un marco general para pre-analizar y optimizar la asignación de recursos y la mejora de procesos de negocio, y con ello adaptarlos a la mejor coincidencia. Los autores analizaron un conjunto de reglas heurísticas y diseñaron algoritmos basados en esas reglas para optimizar la asignación de recursos.

Pesic y Van der Aalst [17] presentaron un trabajo sobre la forma en que el sistema de administración de *workflows* distribuye el trabajo basado en la estructura de la organización y las capacidades de las personas. En este trabajo, los autores proporcionan un modelo básico que puede ser visto como un modelo de referencia de los sistemas de gestión de flujo de trabajo existentes y como un punto de partida hacia un modelo de referencia más exhaustivo para la distribución del trabajo. El modelo presentado es dinámico y está basado en Redes de Petri Coloreadas [11]. Además demuestran cómo los mecanismos más avanzados de distribución de trabajo, conocidos como patrones de recursos, pueden ser modelados y analizados. Los autores consideran el trabajo realizado como una extensión de la *Workflow Patterns Initiative* ya que agregaron cuatro patrones avanzados al modelo existente, además, complementa los patrones de recursos, proporcionando modelos ejecutables para el mecanismo de distribución de trabajo.

Un punto muerto o *deadlock* en un flujo de trabajo ocurre cuando la ejecución de un proceso se detiene y no puede continuar, pero aún no se ha llegado al punto final. Para evitar que los puntos muertos puedan ocurrir debido a las restricciones en los procesos de negocio, Kohler y Schaad [12] analizaron las restricciones de seguridad relacionadas a la separación de tareas en un flujo de trabajo, determinando el número mínimo de recursos necesarios y proporcionando una asignación óptima de usuarios de acuerdo a la actividad para la ejecución de un flujo de trabajo libre de *deadlocks*. Según los autores, el objetivo del método presentado es evitar situaciones que lleven a un punto muerto durante la ejecución de un proceso de negocio. Utilizaron la teoría de grafos en un enfoque de tres pasos para determinar la cantidad mínima requerida de usuarios que permita la ejecución satisfactoria de los procesos de negocio programados. Además, demostraron que esto va en paralelo con la determinación de una asignación óptima de usuario-actividad.

III. Herramientas de Administración de Procesos de Negocio

En esta sección se analiza la metodología utilizada por algunos sistemas BPM y *Workflow* para realizar la asignación de recursos a las distintas actividades a ser ejecutadas en los procesos de negocio modelados.

Se examinan dos sistemas BPM, Bizagi Studio y Bonita Open Solution, y dos sistemas Workflow, YAWL y Together Workflow Server. En trabajos futuros se prevé ampliar el estudio realizado considerando otras herramientas. Los sistemas evaluados son distribuidos sin costo y, a excepción de Bizagi Studio, son de código abierto.

Bonita y Bizagi (a partir de su version 9) utilizan el lenguaje BPMN para la diagramación y modelado de los procesos, es decir, para definir la notación y semántica del Diagrama de Procesos de Negocio [2] [3]. El lenguaje BPMN provee una notación común para que las personas relacionadas con los procesos puedan expresarlos gráficamente de forma más clara, estandarizada y completa. A su vez, YAWL emplea su propio lenguaje de especificación de procesos basado en patrones de flujo de trabajo [7] y Together Workflow Server utiliza XPDL como formato nativo de definición de flujos de trabajo [4].

A continuación se describen brevemente los sistemas mencionados.

Bizagi Studio. En el modelador de Bizagi se pueden asignar participantes, del tipo Rol o Entidad, que realizarán las diferentes tareas modeladas. Sin embargo, no se puede asignar participante a una tarea en particular. Por otra parte, en el BPMS Bizagi la asignación de recursos a una actividad se realiza en el módulo participantes, en tiempo de diseño del modelo considerando: reglas de asignación, método de asignación y precondiciones. Para realizar una asignación de recurso humano a una tarea deben configurarse previamente las características del recurso, por ejemplo: área al que pertenece, cargo, roles, habilidades, entre otros. Los métodos de asignación de recursos utilizados son: por carga, el usuario con menor carga de trabajo; secuencial, uno después de otro cíclicamente; primero disponible, el primer usuario disponible dependiendo de la zona horaria asociada; y todos, cualquiera puede decidir realizar la actividad [2]. En el sistema BPM Bizagi no se tiene en cuenta los recursos no humanos (computadoras en una oficina, libros en una biblioteca, camas en un hospital, etc). Tampoco se tiene en cuenta la cantidad limitada de los recursos disponibles.

Bonita Open Solution. Bonita permite asignar actores (recursos humanos) a un carril (*lane*), así como a tareas específicas definidas con tipo de actividad humana. En ambos casos, se puede asignar un grupo determinado de actores o actores individuales. Cuando se selecciona un grupo de actores, es

posible utilizar filtros para seleccionar un subconjunto de usuarios disponibles en el grupo. En las tareas con tipo de actividad humana es posible asignar actores que reemplazarán a aquellos definidos en el *lane*.

Sin embargo, para realizar una simulación del proceso diseñado se puede agregar recursos discretos, humanos o no humanos, para asignar a la tarea. También es posible definir el tiempo de duración del recurso, lo que permite simular el tiempo de espera por el recurso para completar una tarea. Además, en el momento de agregar al sistema los recursos de simulación, Bonita permite definir recursos ilimitados que pueden ser útiles en la simulación para detectar cuellos de botella creados por otras condiciones distintas a la disponibilidad de esos recursos. Cuando un recurso se define como ilimitado, la simulación utiliza tantos recursos como sea necesario para completar la tarea sin esperar la disponibilidad de los mismos. Una vez completada la simulación, Bonita presenta un informe detallado de todas las tareas realizadas, los recursos empleados, el tiempo esperado por un recurso, el costo, entre otras detalles.

YAWL. El sistema YAWL ofrece apoyo a los patrones de recursos, es decir, permite capturar los requerimientos de los recursos. Los recursos asignables a las diferentes tareas de un workflow pueden ser primarios (humanos) o secundarios (humanos o no humanos). La asignación de estos recursos se puede realizar a través del editor de YAWL una vez conectado con el administrador de recursos del servidor, o directamente desde el sistema de administración de workflow de YAWL. Para realizar las asignaciones es necesario definir primeramente los recursos a ser utilizados en los procesos de flujo de trabajo. En YAWL los recursos pueden ser asignados por el Sistema (dinámicamente, basado en ciertas configuraciones como participantes, roles, disponibilidad del recurso, etc) o por el Usuario (manualmente, por el participante o el administrador). Para que el sistema asigne los recursos al workflow, se debe indicar la estrategia de asignación a ser utilizada: *Fastest to Allocate, Fastest Resource, Round Robin, Cheapest to Starter, Shortest Queue, Random Choice*, entre otros.

En el momento de la asignación de un recurso secundario a un proceso de flujo de trabajo se verifica la disponibilidad de ese recurso y en tiempo de ejecución del proceso (en el WfMS), se registra en un log y se informa al usuario de la disponibilidad o no del recurso asociado, aunque no es posible modificar el recurso no disponible asignado a la tarea en proceso.

Together Workflow Server. Together Workflow Server (TWS) utiliza XPDL como formato nativo de definición de flujo de trabajo. Soporta los procesos de flujo de trabajo automatizados, manuales y mixtos, y tiene algoritmos extensibles para la asignación de item de trabajo. Las actividades están automatizadas mediante un sistema de herramienta de agentes, también extensible, permitiendo que la invocación de la lógica externa sea definida en diferentes formatos (clases Java, EJB, ejecutables nativos, scripts en lenguajes de secuencias de comandos arbitrarios o en Servicios Web). Las interacciones humanas son gestionadas a través de los items de trabajo, que son exclusivamente manuales [4].

En Together Workflow Editor (TWE) se pueden definir participantes del proceso, que son los recursos que realizarán las tareas. Estos pueden ser humanos, roles, unidades organizacionales, recursos, conjunto de recursos o sistemas. Por otra parte, a través del sistema de administración de usuarios de TWS es posible crear grupos y agregar nuevos usuarios a esos grupos. Una vez definidos los usuarios se puede asociar cada usuario con los participantes del proceso en ejecución, definidos en TWE, para la realización de la tarea, ya que los participantes especificados en el editor no están sincronizados con el administrador de usuarios del servidor.

Cuadro I: Comparación entre diferentes BPMS y WfMS

Criterios de comparación	**Bizagi**	**Bonita**	**YAWL**	**TWS**
Permite representar recursos durante el modelado	Si	Si	Si	Si
Permite asignar recursos humanos a una tarea	Si	Si	Si	Si
Permite asignar recursos no humanos a una tarea	No	No	Si	Si
Permite asignar recursos limitados a una tarea	No	No	No	No
Permite asignar recursos consumibles a una tarea	No	No	No	No
Permite simulación de los procesos de negocio	Si	Si	Si	No
Posee documentación	Si	Si	Si	Si
Lenguaje de modelado utilizado	BPMN	BPMN	YAWL	XPDL
Licencia del software	Freeware	Open Source	Open Source	Open Source

Como se puede notar, cada una de las herramientas analizadas permite la asignación de recursos, pero ninguno considera recursos adicionales que pueden ser necesarios para la realización de una tarea. Tampoco tienen en cuenta la cantidad limitada de estos recursos.

IV. Análisis Comparativo de BPMS y WfMS

La representación de los recursos y el tratamiento de los mismos en los diferentes procesos de negocio es muy variado en los sistemas estudiados. Con el objetivo de describir una visión general de los mismos, en el Cuadro I se presenta un resumen de los principales comportamientos considerados en el análisis. A continuación, se describe brevemente cada uno de los criterios de comparación seleccionados.

Permite representar recursos durante el modelado. En general, todos los sistemas analizados permiten representar recursos humanos durante el modelado, ya sean como participantes o como actores que llevarán a cabo la tarea. Sin embargo, no todos permiten modelar recursos no humanos, necesarios para llevar adelante las actividades del proceso.

En el modelador de Bizagi sólo se pueden agregar, modificar o borrar participantes, pero no permite asignarlos a tareas específicas.

A través del modelador de procesos de negocio incorporado en Bonita, es posible asignar actores (individual o perteneciente a un grupo) a una tarea de actividad humana. Estos actores son los que realizarán las actividades.

En el sistema YAWL es posible cargar los recursos, ya sean humanos o no humanos, en el servidor de recursos. En el editor

de YAWL, una vez conectado con el servidor de recursos, se puede indicar al administrador de recursos como manejarlos, es decir, si la asignación de recursos será realizada por el usuario o por el sistema. Una vez que el flujo de trabajo se encuentra en ejecución en el sistema YAWL, el administrador del sistema puede reasignar un item de trabajo a otro recurso humano, además puede asignar recursos secundarios (humanos o no humanos) al item mencionado.

Por último, en TWE, durante el modelado del proceso de negocio, se pueden definir los participantes del proceso (humanos o no humanos), aunque estos participantes no son asignados directamente a tareas específicas. Por defecto, las actividades definidas en un *lane* están asignadas a participantes (responsable del *lane*) pero es posible modificar esa asignación en el administrador de usuarios del TWS.

Permite asignar recursos humanos a una tarea. A través del sistema Bizagi es posible asignar participantes a una tarea específica (como se mencionó en párrafos anteriores), y el participante es el que realizará la actividad modelada.

En Bonita se puede definir actores que realizarán las tareas pertenecientes a un *lane*. Además, a cada tarea de actividad humana se puede asignar actores que reemplazarán a aquellos definidos en el *lane*. En ambos casos, se puede asignar un grupo determinado de actores o actores individuales.

En YAWL, cada item de trabajo es asignado a un recurso humano específico, ya sea durante el modelado en el editor o durante la ejecución del flujo de trabajo en el sistema.

Por último, TWS permite asignar actores a las tareas modeladas. Como se mencionó en párrafos anteriores, las actividades definidas en el *lane* están asignadas a participantes pero es posible modificar esa asignación en el administrador de usuarios del TWS, asociando cada usuario con los participantes del proceso en ejecución (definidos en TWE) para la realización de la tarea.

Permite asignar recursos no humanos a una tarea. El modelador de procesos de Bizagi y Bizagi Studio no permiten asignar recursos que no sean participantes (humanos) a las diferentes tareas de un proceso de negocio.

Bonita tampoco permite asignar recursos no humanos a las tareas modeladas en el proceso de negocio. En este punto es necesario mencionar que para realizar una simulación sí es posible definir recursos no humanos que se utilizarán durante el proceso, como se indica en el apartado acerca de simulaciones.

YAWL sí permite definir recursos secundarios, humanos o no humanos, que pueden ser asignados a cada elemento de trabajo. Sin embargo, durante la ejecución no se tiene en cuenta si estos recursos secundarios son limitados o no.

Al crear participantes en el editor de Together Workflow, es posible definir recursos no humanos y asignarlos a una tarea en particular, pero no está claro cómo lo utiliza TWS al ejecutar los procesos.

Permite asignar recursos limitados a una tarea. Ninguno de los sistemas analizados disponen de opciones para indicar que los recursos son limitados. En este sentido, los sistemas que permiten realizar simulación de los procesos utilizan tantos recursos como sean necesarios para completar la tarea sin esperar por la disponibilidad de los mismos.

Permite asignar recursos consumibles a una tarea. Nuevamente, ninguna de las herramientas evaluadas permite asignar recursos que puedan ser consumidos (por ejemplo, tinta, papel, aceite) durante la realización del proceso de negocio. Estos recursos generalmente tienen una capacidad asociada que sirve para estimar el tiempo de vida del recurso.

Permite simulación del proceso modelado. La simulación de los procesos de negocio empresariales permite mejorar los procesos modelados antes de su puesta en ejecución en un BPMS o WfMS. La simulación no sólo debe responder a preguntas estratégicas, sino también debe servir para tomar decisiones tácticas y operativas.

Bizagi soporta la simulación a través de BPSim[13] un estándar que permite la mejora de los procesos de negocio modelados en BPMN.

Bonita Studio tiene su módulo de simulación incorporado en el sistema y permite administrar los recursos de simulación (humanos o no). Aquí se deben indicar el nombre, la cantidad y el costo del recurso a ser utilizado durante la simulación. A su vez, en la ficha Simulación correspondiente a cada tarea, durante el modelado, se debe asignar el recurso a ser utilizado en esa actividad. Además, en el momento de agregar al sistema los recursos de simulación, Bonita permite definir recursos ilimitados que pueden ser útiles en la simulación para detectar cuellos de botella. Una vez completada la simulación, Bonita presenta un informe detallado de todas las tareas realizadas, por ejemplo, los recursos empleados, el tiempo esperado por un recurso y el costo.

En YAWL para realizar la simulación se debe establecer un vínculo entre el sistema de flujo de trabajo YAWL y el marco de trabajo para la minería de procesos ProM[14]. Este software es un marco de trabajo genérico de código abierto para la implementación de herramientas de minería de proceso en un entorno estándar.

Por su parte, TWS no dispone de un módulo para realizar simulaciones de los flujos de trabajo modelados. Sin embargo, la simulación puede realizarse a través de un software de terceros que permita importar flujos de trabajo modelados en XPDL.

Tiene documentación. Todos los sistemas examinados tienen la documentación necesaria para facilitar el diseño del modelado y la ejecución de los procesos. Esto facilita la incorporación de los mismos dentro de las empresas, y sirve de ayuda a los ingenieros responsables de la administración de los procesos.

Lenguaje de modelado utilizado. Como se mencionó en párrafos anteriores, Bonita y Bizagi utilizan BPMN, YAWL tiene su propio lenguaje y TWS utiliza XPDL como lenguaje de modelado para representar los procesos de negocio.

Licencia del software. Todos los sistemas analizados son de distribución gratuita y permiten su utilización de forma libre; en particular, Bonita, YAWL y TWS, son de código abierto.

[13] *Business Process Simulation*, www.bpsim.org

[14] ProM, www.promtools.org, www.processmining.org

De acuerdo a las comparaciones realizadas, ninguno de los dos sistemas BPM analizados permite asignar recursos no humanos a una tarea a ser llevada a cabo durante el proceso de negocio. Es más, ninguna de las cuatro herramientas evaluadas permite asignar recursos limitados o recursos consumibles a una tarea, lo que demuestra la ausencia de esta característica necesaria en sistemas reales.

V. Consideraciones Finales y Trabajos Futuros

La estandarización de la asignación de recursos en sistemas de procesos de negocios es una tarea pendiente. En ese sentido, varios aspectos deben tenerse en cuenta, por ejemplo, los sistemas deben considerar que los recursos pueden ser humanos o no humanos, duraderos o consumibles. Además, los recursos asignados a las tareas podrían ser complejos, y no sólo individuales como se considera actualmente. La ejecución de las actividades de los procesos de negocio puede ser realizada por un recurso (por ejemplo, una secretaria o un software) o esta ejecución puede consumir recursos (por ejemplo, las camas de un hospital o el aceite de una máquina). También se debe considerar que la disponibilidad para utilizar un recurso puede ser limitada o ilimitada. Todos estos factores contribuyen a la complejidad de la administración de recursos en los sistemas de procesos de negocio.

En este trabajo se presentó una revisión de los principales artículos de investigación que consideran aspectos relacionados a la asignación de recursos en procesos de negocio. Además, se describió cómo cuatro herramientas disponibles en el mercado (Bizagi, Bonita, YAWL y TWS), permiten el tratamiento de recursos. Ninguna de las herramientas analizadas permite asignar recursos limitados ni consumibles a una tarea a ser llevada a cabo en un proceso de negocio. Por otra parte, sólo dos de las cuatro herramientas estudiadas permiten asignar recursos no humanos, aunque no está claro el manejo de esos recursos durante la ejecución del flujo de trabajo. Todo esto podría ser motivo para no completar el proceso modelado. Por último, se presentó un análisis comparativo sobre el modelado y la asignación de recursos, así como el lenguaje de modelado utilizado en estas herramientas.

En trabajos futuros se pretende extender la lista de herramientas considerando otros sistemas BPM y *Workflow*, por ejemplo, jBPM, Intalio, Apache ODE, JFlower, Activiti, Pegasus, ProcessMaker, y OpenWFEru.

Referencias

[1] Ahmed Awad, Alexander Grosskopf, Andreas Meyer, and Mathias Weske. Enabling resource assignment constraints in BPMN. Technical report, Technical report, Business Process Technology-Hasso Plattner Institute, 2009.

[2] Bizagi. Bizagi BPM Suite. Descripcion Funcional, 2014.

[3] BonitaSoft. Bonita BPM, 2014.

[4] Together Teamsolutions Co. *Together Workflow Server: User Manual*, 2011.

[5] Pavlos Delias, Anastasios Doulamis, Nikolaos Doulamis, and Nikolaos Matsatsinis. Optimizing resource conflicts in workflow management systems. *Knowledge and Data Engineering, IEEE Transactions on*, 23(3):417–432, 2011.

[6] Flor Díaz. Gestión de procesos de negocio BPM (Business Process Management), TICs y crecimiento empresarial. ¿Qué es BPM y cómo se articula con el crecimiento empresarial? *Revista Universidad y Empresa*, 10(15):151–176, 2008.

[7] Yawl Foundation. YAWL User Manual, 2009.

[8] Yawl Foundation. Features of YAWL, 2014.

[9] Paul Harmon and Celia Wolf. The state of business process management 2014. Technical report, BPTrends.com, 2014.

[10] David Hollingsworth. The workflow reference model. Technical report, Workflow Management Coalition, 1995.

[11] Kurt Jensen. *Coloured Petri Nets: basic concepts, analysis methods and practical use*, volume 1. Springer, 1996.

[12] Mathias Kohler and Andreas Schaad. Avoiding policy-based deadlocks in business processes. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 709–716. IEEE, 2008.

[13] Hongchen Li, Yun Yang, and TY Chen. Resource constraints analysis of workflow specifications. *Journal of Systems and Software*, 73(2):271–285, 2004.

[14] Tingyu Liu, Yalong Cheng, and Zhonghua Ni. Mining event logs to support workflow resource allocation. *Knowledge-Based Systems*, 35:320–331, 2012.

[15] OMG. Business process model and notation 2.0. Technical report, Object Management Group, 2010.

[16] Chun Ouyang, Moe Thandar Wynn, Colin Fidge, Arthur HM ter Hofstede, and Jan-Christian Kuhr. Modelling complex resource requirements in business process management systems. *ACIS 2010 Proceedings*, 2010.

[17] Maja Pesic and Wil MP van der Aalst. Modelling work distribution mechanisms using colored petri nets. *International Journal on Software Tools for Technology Transfer*, 9(3-4):327–352, 2007.

[18] Nick Russell, Arthur HM Ter Hofstede, David Edmond, and Wil MP van der Aalst. Workflow resource patterns. Technical report, BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.

[19] Nick Russell, Wil MP van der Aalst, Arthur HM ter Hofstede, and David Edmond. Workflow resource patterns: Identification, representation and tool support. In *Advanced Information Systems Engineering*, pages 216–232. Springer, 2005.

[20] Pinar Senkul, Michael Kifer, and Ismail H Toroslu. A logical framework for scheduling workflows under resource allocation constraints. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 694–705. VLDB Endowment, 2002.

[21] Pinar Senkul and Ismail H Toroslu. An architecture for workflow scheduling under resource allocation constraints. *Information Systems*, 30(5):399–422, 2005.

[22] Luis Jesús Ramón Stroppi, Omar Chiotti, and Pablo David Villarreal. A BPMN 2.0 extension to define the resource perspective of business process models. In *XIV Congresso Iberoamericano in Software Engineering (CIBSE), Rio de Janeiro, Brasil*, 2011.

[23] Wil MP Van Der Aalst. Business process management demystified: A tutorial on models, systems and standards for workflow management. In *Lectures on concurrency and Petri nets*, pages 1–65. Springer, 2004.

[24] Wil MP van der Aalst. Business process management: A comprehensive survey. *ISRN Software Engineering*, 2013, 2013.

[25] Jiajie Xu, Chengfei Liu, and Xiaohui Zhao. Resource allocation vs. business process improvement: How they impact on each other. In *Business Process Management*, pages 228–243. Springer, 2008.

Desarrollo de software robusto y flexible en el Paradigma de la Programación Orientada a Objetos

Passerini, Nicolás
Universidad Tecnológica Nacional (F.R. Buenos Aires) – UNQ – UNSAM – INSSJP
Spigariol, Lucas
Universidad Tecnológica Nacional (F.R. Buenos Aires / F.R. Delta)
Casanovas, Inés
Universidad Tecnológica Nacional (F.R. Buenos Aires/F.R.G.P.) – UBA –FUNDESCO

Abstract

Existen herramientas que extienden parcialmente el paradigma de objetos, intentando ofrecer soluciones a situaciones para las que resulta poco adecuado. Por otro lado, la enseñanza de programación en las carreras relacionadas con sistemas de información muchas veces se queda en un planteo básico del paradigma de objetos que no hace referencia a conceptos más avanzados. Este trabajo realiza una aproximación al problema analizando la viabilidad de construir herramientas de desarrollo que permitan un mayor aprovechamiento de las posibilidades de la Programación Orientada a Objetos, principalmente de sus conceptos avanzados en cuanto a formas flexibles de generalización de responsabilidades, para lograr un software sólido y más fácil de modificar, acorde a la actual complejidad del panorama profesional de sistemas. Adicionalmente, propone aplicar estas mejoras con objetivos pedagógicos en la enseñanza universitaria.

Siendo los datos mayormente cualitativos y el análisis interpretativo, los resultados se presentan mediante descripciones, perspectivas y reflexiones. El énfasis fue puesto en las propiedades que caracterizan al objeto de estudio, identificando patrones que facilitan el entendimiento de la problemática.

En cuanto a la perspectiva pedagógica, este trabajo plantea contenidos, enfoques y recursos que eventualmente permitan seguir recreando materias básicas de enseñanza de objetos, como así también crear nuevas materias que continúen o profundicen la temática.

1. Introducción

La Programación Orientada a Objetos constituye en la actualidad el paradigma más utilizado para la construcción de sistemas de software. Si bien existen diferentes herramientas, lenguajes y variantes en las implementaciones concretas en el marco de este paradigma, son numerosos proyectos de desarrollo de sistemas en el ámbito industrial que hacen un uso limitado y parcial de las herramientas, conceptos y facilidades que ofrece el paradigma para construir software de calidad. Esto tiene su impacto tanto en los productos finales de software que se construyen como en los tiempos y modalidades de su proceso de desarrollo, y se manifiesta de múltiples formas, que desde la perspectiva analítica del presente trabajo, se describen desde la tensión entre flexibilidad y robustez. Polarizando los extremos, por un lado, existen en soluciones que con la excusa de la solidez y confiabilidad optan por modelos estáticos y se tornan muy poco flexibles frente a los requerimientos de cambio y adaptaciones permanente que caracteriza hoy la dinámica de los sistemas de información, los que se puede categorizar como un uso imperativo del paradigma de objetos. Por otro lado, hay soluciones que pretendiendo ser modernas e innovadoras, se tornan poco sólidas por no contar con el respaldo metodológico ni las herramientas de desarrollo adecuadas que le brinden robustez, lo que se define como un uso ingenuo del paradigma de objetos.

Generalmente, los desarrollos de baja calidad coinciden en basarse sobre un modelo formado por unos pocos conceptos básicos del paradigma de objetos, entre los que se destacan la centralidad de las clases como única herramienta organizadora de los objetos y la herencia simple como modo jerárquico de relación entre las clases. En este enfoque, las clases toman un rol preponderante por sobre los objetos, y en consecuencia cada objeto pertenece a una única clase, las clases se limitan a crear objetos, un objeto no puede cambiar de clase y todo el comportamiento del objeto está definido por su propia clase. A su vez, la herencia simple aparece como la forma predilecta para organizar la jerarquía de clases y como herramienta exclusiva para definir el código.

Aun asumiendo que los profesionales de sistemas aplican correctamente este modelo, lo cual es también muy discutible con una rápida mirada del ambiente, se puede decir que funciona y permite desarrollar soluciones eficaces, pero existen muchas situaciones para las que resulta poco adecuado, insuficiente y poco proclive a modificaciones. Para esto, hay una serie de herramientas que extienden y profundizan el paradigma de objetos, que intenten abarcar un mayor conjunto de posibilidades, entre las que se encuentran las de traits, mixins, prototipado, open classes, combinación con programación funcional y programación orientada a aspectos.

Por otro lado, los planes de estudios de las carreras relacionadas con los sistemas de información, en su amplia mayoría se quedan en un planteo básico del paradigma de objetos sin hacer referencia a este otro tipo de conceptos más avanzado. De esta manera, las universidades forman profesionales poco preparados para aprovechar en su toda su potencialidad las posibilidades del paradigma de objetos.

.

2. Objetivos

El presente trabajo pretende realizar una primer aproximación al campo para analizar la viabilidad de construir herramientas de desarrollo que permitan un mayor aprovechamiento de las posibilidades de la Programación Orientada a Objetos, principalmente de sus conceptos avanzados en cuanto a formas flexibles de generalización de responsabilidades, en vistas a lograr un software sólido y más fácil de mantener y modificar, acorde a la actual complejidad del panorama profesional de sistemas. Busca describir el panorama actual y abrir posibilidades en relación a herramientas conceptuales que permitan hacer evolucionar el diseño y la construcción de software basados en modelos alternativos de la programación orientada a objetos, como así también, de nuevas variantes dentro del paradigma de objetos que continúen expandiendo el horizonte de los metamodelos disponibles para el diseñador de software.

Por otra parte, a partir de una revisión de los contenidos y metodologías que se desarrollan en las carreras universitarias de sistemas, en particular las que abordan la programación en el marco del paradigma orientado a objetos, se propone analizar la pertinencia de introducir las herramientas de desarrollo propias de modelos alternativos, ya sea en tanto software educativo como facilitador de los procesos de aprendizaje o como herramientas de desarrollo de propósito general. De esta manera, se busca fortalecer en el ámbito universitario la formación de profesionales de sistemas que estén a la vanguardia de la tecnología existente y que dispongan de herramientas para implementar estos conocimientos en ambientes de trabajo reales, acordes a los requerimientos profesionales de la actualidad.

3. Metodología utilizada

La metodología utilizada, en primer lugar, se basa en un trabajo más sistemático de exploración bibliográfica acerca de los avances en los conceptos del paradigma de objetos y las discusiones teóricas en relación a las perspectivas y posibilidades del paradigma. Complementariamente, se llevó adelante una recuperación de experiencias sobre implementaciones de nuevas características que incluyen los lenguajes de programación vigentes del paradigma de objetos.

Se procedió inductivamente a la revisión, validación y organización del material de relevamiento recolectado para proceder a su análisis destacándose la incorporación y formalización de nuevos concepto, la recopilación de antecedentes y el estudio de herramientas experimentales que aún no han tenido una plena aplicación profesional. Este proceso inductivo no fue puramente lineal sino que se trabajó en forma iterativa incremental a medida que surgían nuevas categorías de análisis. Esto implica un modelo metodológico emergente a partir de un plan inicial, no prescriptivo sino orientativo. Durante el transcurso de la investigación, nuevas perspectivas emergieron haciendo necesaria una visión interpretativa del problema y su complejidad. El análisis se hizo como una continua reflexión de la evolución de la problemática en estudio a la luz del surgimiento de nuevos conceptos y situaciones desde los datos examinados en un proceso concurrente de análisis e interpretación [1]. Siguiendo a Schatzman & Strauss [2] el énfasis fue puesto en las propiedades del objeto de estudio, identificando patrones que faciliten el entendimiento de la problemática.

La validez y confiabilidad de un estudio cualitativo están definidas por la pertinencia y calidad de las fuentes de datos. Se incluyó literatura proveniente de fuentes académicamente confiables en cuanto a la evaluación de autenticidad, credibilidad y rigor de sus publicaciones [3]. Se emplearon procedimientos sistemáticos y documentados de triangulación y convergencia de teorías de diferentes fuentes minimizando los posibles sesgos. Al ser los datos mayormente cualitativos y el análisis interpretativo, los resultados se presentan mediante descripciones, perspectivas y reflexiones [4].

De manera transversal, otro de sus ejes abordó la utilización de herramientas informáticas en el ámbito académico, y que desde lo metodológico, incluye también una perspectiva pedagógica y de análisis del contexto. Teniendo en cuenta las características del actual estado de la industria de desarrollo de software y las herramientas que se utilizan tanto en el ámbito profesional como el académico, se planteó la posibilidad de compatibilizar las

herramientas desarrolladas con las utilizadas en esos ámbitos.

4. Contribuciones esperadas

El trabajo se enmarca una visión integral del avance del conocimiento de la disciplina, proponiéndose realizar aportes en el campo específico de la ciencia que den un marco conceptual para desarrollos tecnológicos factibles de ser transferidos al medio profesional del desarrollo de sistemas y que produzcan mejoras visibles en los entornos de trabajo que se enmarcan en el contexto del paradigma de objetos. Las herramientas conceptuales y metodológicas que se releven permitirán proponer avances, mejores y nuevas herramientas para un mejor aprovechamiento de los modelos alternativos del paradigma de objetos en desarrollos industriales.

Son numerosos los lenguajes orientados a objetos que han manifestado un alto crecimiento en cuanto a su grado de adopción por parte de la industria, que reconoce su utilidad como una herramienta para mejorar tanto la calidad de los sistemas construidos como su productividad al hacerlo. Sin embargo, este crecimiento se ve en parte limitado por la ausencia de entornos de desarrollo adecuados para la producción de sistemas de gran envergadura. El análisis de esta situación favorecerá el desarrollo de herramientas concretas que en su conjunto podrían producir un salto cualitativo en el grado de utilización de las bondades del paradigma como así también de la adopción de modelos alternativos dentro del paradigma de objetos.

Por último, retomando la perspectiva pedagógica ya planteada, aportará elementos para llevar adelante de mejor manera las materias que tratan sobre el paradigma de objetos en el ambiente universitario. Esto se concreta mediante las mejoras que se quiere realizar sobre las herramientas de desarrollo que se utilizan como recurso pedagógico actualmente y también con la incorporación de las nuevas herramientas y conceptos.

5. Análisis de las herramientas

La visión tradicional del paradigma de objetos con frecuencia se encuentra con trabas que le impide proponer modelos adecuados para determinados tipos de problemas. Muchas de estas alternativas han demostrado ser superiores en algunos aspectos a la visión tradicional, pero todas también muestran debilidades.

Si bien muchas de las propuestas avanzadas del paradigma de objetos ideas tienen décadas de desarrollo en el ámbito científico, su uso en la industria es incipiente, más allá de algunas pocas excepciones. Recientemente varios lenguajes que incluyen estos conceptos han incrementado sus niveles de adopción en la industria, entre ellos Python, Ruby, Groovy, Scala, Pharo y Newspeak. Por ejemplo, como explica Wilkens al referirse a JavaScript[5], este lenguaje de programación en crecimiento, manteniendo la idea central de objetos, aborda una manera diferente la relación entre objeto y clase.

El concepto clave y a la vez muchas veces ausentes, es el de polimorfismo. Como se afirma en el trabajo titulado precisamente *Difficulties in Learning Inheritance and Polymorphism*[6], para poder manejar la complejidad inherente a herencia y polimorfismo, se propone un *modelo formal* de la ejecución de un programa que explica el enlace dinámico.

La incorporación de modelos alternativos en la industria del software presenta desafíos importantes dado que la mayor parte de las herramientas utilizadas no tienen las características que estos requieren, dado que han sido concebidas teniendo en cuenta la visión más tradicional. Esta carencia puede notarse, por ejemplo, en los entornos de desarrollo, lenguajes de modelado, herramientas conceptuales y tecnológicas para realizar refactorings o patrones de diseño.

Son necesarias herramientas que facilitan la construcción de un software más flexible frente a los cambios, más sencillo de adaptar a los nuevos requerimientos, que ofrezca una variedad de alternativas a los profesionales de sistemas a la hora de diseñar e implementar soluciones informáticas, sin que ello signifique perder robustez.

6. Herramientas de desarrollo y depuración de errores

Los lenguajes dinámicos ofrecen la posibilidad de modificar el comportamiento de los objetos en tiempo de ejecución. Algunas metodologías, como el Desarrollo Dirigido por las Pruebas (TDD), proponen hacer uso exhaustivo de esta posibilidad, esta técnica es muy común en lenguajes dinámicos que cuentan con entornos de depuración (debuggers) con amplias posibilidades de programación, como el dialecto de Smalltalk llamado Pharo. Sin embargo, otros lenguajes dinámicos más flexibles que Pharo carecen de ese tipo de entornos. Como afirma un integrante del equipo desarrollador de dicho lenguaje de programación, "los test unitarios son actualmente una importante actividad en los proyectos de software. Los test automáticos ayudan al desarrollador a asegurar la calidad del software y detectar posibles errores o defectos en el código de las aplicaciones"[7].

Adicionalmente, la utilización de modelos más flexibles requiere de una profunda revisión de las herramientas que se utilizan para desarrollar. Por

ejemplo, para el desarrollo orientado a objetos sin clases, el proyecto Self propuso en un entorno de trabajo (IDE) totalmente diferente a los editores de código más usuales. Sin embargo, poco se ha avanzado en este terreno. En este sentido, sus creadores lo definen como "un lenguaje de programación basado en un pequeño número de ideas simples y concretas, como prototipos, comportamiento y variables"[8].

Estudiando la problemática específica de la depuración de errores en lenguajes de programación dinámicos, una necesidad imperiosa en vistas a la robustez del código es el diseño de herramientas que simplifiquen la tarea de comprender los potenciales errores que surgen al programar con estos modelos o al ejecutar este tipo de programas. A su vez, el propósito es lograr que dichas herramientas permitan la modificación del programa en tiempo de ejecución para comprobar la correctitud de una modificación realizada sobre él, sin necesidad de reiniciar la prueba, y de esta manera optimizar tiempo de desarrollo y aprovechar toda la capacidad de los lenguajes dinámicos.

Como antecedentes en la materia, existe una herramienta denominada POT (Transacciones Puras en Objetos) que, se propone incorporar el concepto de transacción como parte del metamodelo de un lenguaje orientado a objetos. Así lo explican sus creadores: "Enfrenta dos problemas de rutina con las IU: la sincronización entre la vista y el modelo, y la posibilidad de modelar operaciones atómicas. Las variantes de MVC basadas en enlaces, como MVVM y MVB, proporcionar una solución interesante al problema de sincronización. La combinación de estas ideas con STM en gran medida amplía su aplicabilidad y permite resolver ambos problemas en un manera transparente, es decir, sin necesidad de cambios en el código de dominio."[9].

Desde esta perspectiva, todo mensaje que se envía a un objeto se realiza dentro de un contexto transaccional. El contexto acepta las operaciones usuales de abrir transacción, commit y rollback. La presencia del rollback permite retrotraer los objetos en un contexto a su estado antes de comenzar la transacción, simplificando visiblemente el manejo de errores y excepciones. En el trabajo se propone una implementación basada en programación orientada a aspectos. Una extensión de esta idea, incorpora el concepto en una herramienta de construcción de interfaces de usuario. La capacidad de rollback provista por POT permite una implementación muy sencilla de ventanas con operaciones de "cancel" y "undo".

7. Refactorings

Un concepto que es muy valorado por las metodologías de desarrollo de concepción más reciente es el de refactoring, que consiste en una modificación que se realiza sobre un programa que produce un cambio estructural sin alterar el comportamiento visible del programa. Como garantiza que el comportamiento del sistema no cambia por las modificaciones en el código, puede ser aplicado en forma segura aún en etapas avanzadas del desarrollo, por ejemplo durante el mantenimiento del sistema. Por este motivo, esta técnica permite corregir errores de diseño o bien preparar el programa para un agregado de funcionalidad posterior.

Considerando que existen escasas estrategias de refactoring específicas para los modelos alternativos y que las propias de la visión tradicional son inadecuadas o subutilizan las posibilidades de dichos modelos, es un desafío poder definir nuevas estrategias de refactoring que sean apropiadas y permitan optimizar las tareas de construcción de software.

Una línea de trabajo interesante consiste en realizar "el refactoring de una jerarquía de clases, mediante el análisis de las dependencias entre las clases intervinientes en la relación de herencia de un sistema basado en clases y su transformación para obtener una mejor estructura que una jerarquía de clases, focalizando en obtener componentes con servicios ofrecidos y requeridos en un sistema basado en componentes"[10]. Este trabajo apunta a permitir reemplazar las relaciones de herencia por delegación, produciendo un programa más flexible y que permite cambiar su comportamiento dinámicamente. Las herramientas de análisis de una jerarquía de herencia desarrolladas en este trabajo sientan las bases para estudiar otros refactorings. Por otro lado, sientan bases que podrían utilizarse para realizar análisis similares en modelos alternativos del paradigma de objetos.

8. Modelo integrador

Uno de los modelos alternativos del paradigma de objetos propone definir un modelo de lenguaje con un único concepto unificador de las capacidades de clases y traits y un álgebra propia basado en un conjunto de operaciones que permitan combinar estos elementos para formar nuevos, dejando el control de la composición al desarrollador.

Partiendo de la idea original de los modelos que tienen por un lado traits, en los que es el desarrollador quien controla su composición mediante operaciones definidas en un álgebra, tales como suma, exclusión o aliasing, y por otro lado cuentan con clases que definen comportamiento estáticamente, plantea la posibilidad de hacer evolucionar ese modelo para construir un concepto unificador de clases y traits, con un álgebra acorde que con diversas operaciones posibilite la definición de nuevos elementos dinámicamente y de esta manera favorecer el desarrollo de software flexible y reutilizable.

Este camino parte de entender que muchos lenguajes orientados a objetos definen una forma distinta de especificar el comportamiento de los objetos. "Aunque muchas de estas variantes parecen converger a conceptos muy difundidos, como herencia simple de clases o prototipos, continúan surgiendo nuevas construcciones en un intento de resolver sus limitaciones propias. La mayoría de estas construcciones se desarrollan para integrarse en un lenguaje ya existente, sin embargo algunas de ellas presentan una solución radical preferible a simplemente tratar de enmendar las herramientas adoptadas por el modelo original existente."[11]

8. Patrones de diseño

A partir de su publicación original en el año 1995, el concepto de patrón de diseño se ha convertido en una herramienta fundamental para muchos aspectos del desarrollo de sistemas. Actualmente muchas de las metodologías y técnicas de estudio del diseño de software hacen uso intensivo de patrones. Sin embargo, esta técnica ha sido desarrollada casi exclusivamente en modelos de objetos basados en un modelo de clasificación rígida y herencia simple. Muchos de los patrones pensados para ese modelo no son aplicables a otras variantes del paradigma de objetos.

Ante la constatación de que los patrones pensados para la visión tradicional del paradigma de objetos no son aplicables en los modelos alternativos y la falta de patrones de diseño específicos para ser usados en ellos, es necesario poder formular nuevos patrones que aprovechen las capacidades específicas de los modelos alternativas y de esta manera incrementar el aprovechamiento de la potencialidad de los lenguajes de programación.

9. Ambiente universitario

La dicotomía entre teoría y práctica, tan frecuente en los sistemas educacionales de América Latina no parece haber perdido vigencia. Retomando lo que llamó Paulo Freire denominó *educación bancaria* [15], muchas veces en la actualidad universitaria se tiene una concepción de enseñar que se reduce a transferir el concepto vacío de contenido y no transmitir la comprensión precisa de este contenido. En la enseñanza de la programación es conveniente desarrollar una dinámica que combine lo teórico con lo práctico, donde la práctica permita comprender mejor el sentido de los conceptos teóricos y a la vez que dichos conceptos permitan plasmar soluciones prácticas adecuadas a problemas reales. A su vez, se debe reconocer la distancia que separa al estudiante de la programación, que no es menor ni trivial, aunque tampoco es una dificultad insalvable: requiere de mediaciones pedagógicas. En este contexto, como estrategias docentes para facilitar el uso de herramientas de desarrollo propias de los modelos alternativos del paradigma de objetos en el ámbito universitario, una experiencia emblemática la constituye el software educativo denominado *Ozono*.

Propone un método de enseñanza del paradigma de objetos enfatizando el concepto de *recorrido*, en particular alterando el orden típico en que los conceptos se van introduciendo a lo largo del curso. Para sustentar esta estrategia pedagógica se desarrolló un ambiente de programación orientado a objetos que no contiene el concepto de clase. Utilizando esta estrategia y la herramienta que la soporta se demuestra que es posible lograr un aprendizaje mucho más rápido y robusto del paradigma. El modelo de prototipos propuesto por este trabajo es sencillo y adecuado para la enseñanza. En su primera versión se denominó *Object Browser*. Posteriormente, fue abierta a la comunidad a través de una licencia *open source*, permitiendo la colaboración de numerosas personas, entre alumnos y docentes universitarios, y pasó a llamarse *Loop* Como consecuencia de nuevos cambios y la incorporación de mejoras y herramientas complementarias, tomó su actual denominación de Ozono.

Una de las motivaciones iniciales fue que "los alumnos puedan experimentar rápidamente la aplicación de los conceptos de objetos, mensajes y relaciones, haciéndolos andar en una computadora y pudiendo aprender de los resultados que obtienen" [16].

Otra de las claves de la herramienta, desde el punto de vista de los docente y creadores de Ozono es que "contar con un propio entorno de programación nos permite seleccionar los conceptos de programación que queremos introducir en cada paso del proceso de aprendizaje, proporcionando una excelente terreno para la introducción gradual de los conceptos" [17].

Desde una mirada pedagógica, "lo que consideramos que brinda la mayor utilidad a la herramienta es el momento, el enfoque y la forma de uso que se hace de ella, acorde a una ponderación de la importancia relativa de los temas de la asignatura desde la percepción del campo profesional, a las decisiones acerca del orden y gradualidad en que se presentan los temas y a la importancia que se le da a la resolución de problemas reales utilizando un lenguaje concreto como articulación entre teoría y práctica" [18].

Si bien el software fue objeto de premios (tercer puesto en los premios a la innovación ESUG 2011) y es utilizado actualmente en la Universidad Tecnológica Nacional (FRBA y FRD), en la Universidad Nacional de Quilmes y en la Universidad Nacional de San Martín, sigue teniendo mejoras pendientes y nuevos requerimientos por parte de los docentes.

11. Conclusiones

El aporte del presente trabajo consiste en la definición de un modelo particular, alternativo e innovador de construcción de software en el paradigma de objetos como un campo disciplinar sobre el cual es oportuno y necesario tanto profundizar analíticamente como trabajar en el desarrollo de nuevas herramientas informáticas que permitan enriquecer la formación y el ejercicio laboral de los profesionales de sistemas.

La recuperación de trabajos existentes, la delimitación de categorías teóricas y la organización del campo en diferentes aspectos brinda la base para continuar con un análisis de factibilidad de desarrollo de un software cuya finalidad es mejorar la forma, precisamente, de desarrollar software. En este sentido, se ofrece un primer acercamiento a la problemática, estimando que un proyecto de investigación más ambicioso pueda profundizar en la producción teórica en la medida en que mediante una forma de análisis y diseño iterativo incremental, se puedan ir desarrollando las herramientas pertinentes, estableciendo de esta manera un diálogo productivo entre teoría y práctica. Esto permitirá que rápidamente se pueda comenzar a construir aplicaciones usando estas mismas ideas, para validar su utilidad en contextos realistas y para impulsar la detección potenciales desarrollos de nuevas herramientas conceptuales y tecnológicas específicas para ese contexto.

En cuanto a la perspectiva pedagógica, este trabajo orienta a plantear contenidos, enfoques y recursos que eventualmente den lugar a seguir recreando materias básicas de enseñanza de objetos, como así también crear nuevas materias que continúen o profundicen la temática introduciendo, por ejemplo herramientas de testeo o chequeo de tipos. Los conceptos que apuntan hacia una mayor flexibilidad en los desarrollos, como los mencionados en relación al manejo de transacciones, el álgebra integradora de clases, o las estrategias de refactoring, en última instancia orienten hacia la reformulación de planes de estudios que sean acordes a los avances científicos y tecnológicos en el área. A su vez, las nuevas herramientas de tecnología educativa o las mejoras que se puedan realizar sobre las existentes a partir de las ideas planteadas, podrán ser utilizadas por los numerosos docentes que llevan adelante su tarea educativa en materias vinculadas con el desarrollo, y la sistematización del conocimiento asociado a los modelos de objetos permitirá una mejor incorporación de estas ideas en la propuesta curricular de las carreras universitarias relacionadas.

En definitiva, se trata de develar nuevos horizontes, pero no azarosamente o corriendo detrás de efímeras modas o espejismos dibujados por los vaivenes del mercado, sino mediante una reflexión técnica que sugiere los caminos firmes por donde seguir andando.

12. Referencias

[1] Marshall C. y Rossman G., *Designing qualitative research*. Sage Pub., CA, 2006.

[2] Schatzman L. y Strauss A., *Field research strategies*. Prentice Hall, NJ, 1973.

[3] Le Compte M. y Goetz J., *Ethnography and qualitative design in educational research*. Academic Press, Fl., 1984.

[4] Miles M. y Huberman A., *Qualitative data analysis: a sourcebook of new methods*. Sage Pub, CA, 1994.

[5] Wilkens, L. "Objects with Prototype-Based Mechanisms", en *Journal of Computing Sciences in Colleges*, 17, 3, February 2002.

[6] Liberman, N., Beeri, C. y Kolikant, Y. (2011). "Difficulties in Learning Inheritance and Polymorphism*"*. en *ACM Transactions on Computing Education (TOCE*), 11, 1, February 2011 .

[7] Reichhart, S., *Assessing Test Quality*. Institut fur Informatik und angewandte Mathematik, Universitat Bern, 2007.

[8] Ungar, D. y Smith, R., "Self, the power of simplicity" en *Lisp and symbolic computation.* Kluwer Academic Publishers. Marzo 1991.

[9] De Jesus, R., Fernandes, J., Gassman, L., Passerini, N. y Tesone, P., "Pure Transactional Observable Objects*". 1er Congreso Nacional de Ingeniería Informática y Sistemas de Información*. Córdoba 2013.

[10] Arévalo, G. y Passerini, N., "An automatic approach for migrating inheritance relationships into composition", *XVII Congreso Argentino de Ciencias de la Computación*, 2011, pp. 827-836.

[11] Bergh, A., Cesario, L., Sanzo, A., Scarcella, N. y Matos, M. "Chunk's Combinatory Algebra Definition", 2012.

[12] Griggio, C., "Generate UML diagrams from Smalltalk code for Pharo *". European Smalltalk User Groups. Conference*, 2010. Barcelona.

[13] Freire, P., *Pedagogía del Oprimido.* Siglo XXI Editores, 2009.

[14] Lombardi, C., Passerini, N. y Cesario, L., "Instancias y clases en la introducción a la programación orientada a objetos". *Smalltalks 2007 – Primera Conferencia Argentina de Smalltalk*, 2007.

[15] Griggio C., Leiva G., Polito G., Decuzzi G., Passerini N. *A programming environment supporting a prototype-based introduction to OOP*, IWST '11 Proceedings of the International Workshop on Smalltalk Technologies, ACM New York, NY, USA 2011.

[16] Spigariol, L. y Passerini, N., "Enseñando a programar en la orientación a objetos". *1er Congreso Nacional de Ingeniería Informática y Sistemas de Información*, Córdoba, 2013.

A Graph Suite Generator for Real-World Quorum Protocol Analysis*

Robert Schadek and Oliver Theel
Carl von Ossietzky University of Oldenburg
Department of Computer Science
D-26111 Oldenburg, Germany
{*robert.schadek,theel*}*@informatik.uni-oldenburg.de*

Abstract

Applying quorum protocols in real-world networks often turns out to be difficult, because existing quorum protocols usually require specific logical topologies that are different from typical topologies found in real-world networks. Therefore, often inefficient mappings are needed from the logical topologies to the real-world network. We present a protocol that is able to work directly on real-world networks. As our protocol is not bound to one specific topology, a performance analysis becomes more difficult as all topologies have to be analyzed. As this is impossible, we will present criteria as to when graphs can be considered equal regarding to our analysis. Additionally, we will present a graph generator that is able to create a representative suite of graphs, based on presented properties, that will be used in further analyses of our protocol.

1. Introduction

Quorum protocols have a wide range of applicability. They have been used for data replication protocols and mutual exclusion [2], [11]. Quorum protocols use a set of rules to construct quorums Q on a static number of nodes N of a distributed system. These quorums are then used to execute the desired operation. The idea behind replication is to increase the availability of the stored object by having multiple replicas of it available. Nodes are available with a certain probability $p \in [0, 1]$ and each of these nodes hosts a replica of the replicated data object. If a node is not available, then the hosted data is also not available. Quorum systems usually support read and write operations to manipulate the stored replicated data. For each read and write operation, a set of nodes is created it is used to execute the read respective write operation. These sets are called read quorums (QRs) and write quorums (QWs). It is a desired property that read and write operations on the replicated data behave as they would do on the single replica. This is known as one-copy serializability (1SR) [1]. It means that even though possible several operations might be executed in parallel, the history of the data appears as if executed in a serial manner. Upholding this property is the task of the quorum protocol. Quorum protocols usually achieve 1SR by constructing the QWs in a way that they have at least one node in common with every other QW. Furthermore, every QR is constructed in a way such that at least one replica of the QR is also a part of every QW. This way, locking a QW will prevent every other write operation from locking an additional QW and thereby prohibit parallel writing as well as creating a QR.

All this happens on a logical level. This means that nodes and replicas are nothing more than elements in a set and creating QRs and QWs is nothing more than set operations. Some quorum protocols use logical topologies (LT) for these set operations. These quorum protocols are called structured quorum protocols. These LT arrange the nodes in graphs to increase the efficiency of the quorum protocol. LT have been used with great success in the scope of quorum systems. A very prominent and efficient example for such a protocol is the Triangle Lattice (TL) protocol [12]. The TL protocol uses a triangulated grid as shown in Figure 1.

If such a quorum protocol should be used on a physical network topologys (PNTs), it usually has to be mapped to the PNT as the LT used by the structured quorum protocol will likely not match the PNT. By mapping, we mean a bijection from one graph to another. Changing the PNT to match the LT may be impossible or simply unwanted. As an example, consider the LT with 16 nodes used by the TLprotocol in Figure 1 and the PNT shown in Figure 3. In order to use the QRs and QWs created by the TLprotocol, additional nodes may be required in the PNT to actually connect the nodes of the quorums. Therefore, depending of the LT and the PNT, this required mapping step may decrease the availability of operations of the quorum protocol.

The quorum protocol, we propose works on any planar graph. Together with a technique that transforms non-planar graphs into planar graphs, this protocol is applicable to every two-dimensional network topology.

In order to compare the performance of our protocol to the performance of other structured quorum protocols, we need a representative set of graphs. If we only choose one specific graph topology for the comparison, then our quorum protocol or the competing quorum protocols may be optimal for that topology and therefore a performance comparison

. * This work has been partially supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

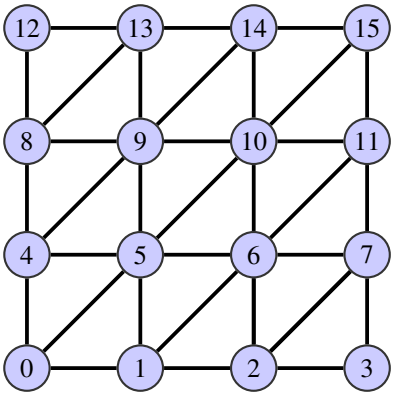


Figure 1: Optimal TLprotocol graph with 16 nodes.

based on only this graph would not be representative. Additionally, the chosen topology might not be represented in real-world networks. Therefore, the graphs we are creating for our graph suite, also needs to consist of graphs that can be found in real-world networks like buses, star-topologies, grids, trees and combinations of them. Furthermore, we need random undirected graphs to strengthen the statement of our quorum protocol analyse. As the number of possible graphs is uncountable, we cannot iterate all graphs and analyse our protocol based on them. To minimize the number of possible test cases, we partition graphs into different classes. Defining when graphs are equal or belong into the same class is difficult and depends highly on the comparison operator. Based on the graph equality property and the general graph properties discussed later, we will present an idea for an algorithm for a graph generator that will be used to create a meaningful graph suite which we will use for the analysis and comparison of our protocol.

We will present such a definition of equality for graphs in this paper. The graph suite generator will allow us to compare our protocol against other quorum protocols in a representative manner.

The paper is structured as follows: The system model used in this paper will be discussed in Section 2. Our protocol is presented in Section 3. The concept of mappings will be discussed in Section 4. In Section 7 we will present certain graph properties we found suitable for the comparison of graphs. Additionally, we will present our equality criterion for graphs in Section 7. In Section 6 we will discuss network topologies that are present in real-world situations. The resulting graph generator design will be discussed in Section 7. Related work in discussed in Section 8. Section 9 gives a conclusion and an outlook onto further research.

2. System Model

Graphs describe the topology of the LT and PNT. A graph G is defined by a set of vertices V and a set of edges E that connects the vertices. A vertex $v_i \in V$ is defined as $v_i = (x, y)$, where x and y are coordinates in a plane and i is the index of the vertex. An edge $e_{i,j} \in E$ is defined as $e_{i,j} = (v_i, v_j)$, where v_i and v_j are vertices. Two vertices v_1, v_2 are called connected in a graph G if:

$$\exists e \in G : e = (v_1, v_2) \tag{1}$$

Unless stated otherwise, edges are commutative, meaning that if edge $e_{i,j}$ exists, so exists edge $e_{j,i}$ if both edges are of same graph.

A graph is planar if no edge intersects on the plane with another edge. The graph in Figure 3 is not planar, because the edges $e = (0, 10)$ and $e = (2, 4)$ intersect. The graph in Figure 1 is planar, as no edges intersect.

A path $P_{v,v'}$ connects two vertices v and v' in graph G through potentially multiple edges. A path is a sequence of edges. For instance, a path $P_{1,8}$ of Figure 2 is $< e_1 = (1, 14), e_2 = (14, 8) >$. A shorter notation just lists the vertices that are part of the path. For the given example the short path notation is $P_{1,8} =< 1, 14, 8 >$.

A path partitions a graph into three partitions S_1, S_2 and S_3 if partitions S_2 and S_3 are divided by partition S_1, the path itself, such that:

$$\nexists P_{v,v'} : v \in S_2 \wedge v' \in S_3 \wedge P_{v,v'} \cap S_1 = \emptyset \tag{2}$$

The partitions S_2 and S_3 can be empty, only the path i.e. S_1 must contain elements. As an example for a partitioning path, in Figure 3, consider the path $P_{1,3} = (1, 4, 10, 3)$.

3. The Crossing Protocol

The idea behind the Crossing Protocol is to create intersections by crossing a given PNT. Read quorums have to cross the PNT vertically or horizontally. Write quorums have to cross the PNT vertically and horizontally. Every vertical crossing has to intersect with every horizontal crossing. Therefore, every read quorum has to intersect with every write quorum and every write quorum has to intersect with every other write quorum. The fundamental idea behind this is, that a vertical crossing basically partitions the graph in half vertically and a horizontal crossing partitions the graph in half horizontally. Therefore, horizontal crossings intersect with vertical crossings and vice versa. This approach is similar to the TL Protocol. In contrast to the TL protocol which requires a specific LT as shown in Figure 1, the Crossing Protocol works on any planar PNT.

A crossing is a path that connects specific nodes. The specific nodes are the nodes on the border of the topology. The border marks a path on the topological outside of the border. Further more, the border is partitioned in four subsets, the top-set (T), the bottom-set (B), the left-set (L) and the right-set (R). Together these are called the TBLR sets. A horizontal crossing has to connect a node from the left-set to a node from the right-set. A vertical crossing has to connect a node from the top-set to a node from the

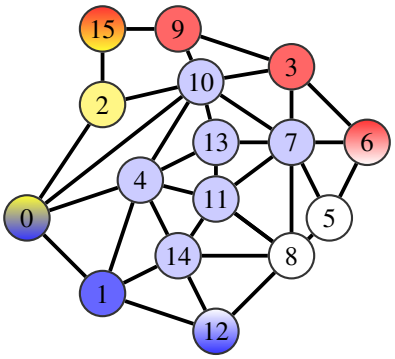


Figure 2: A graph with T nodes marked red, B nodes marked blue, L nodes marked yellow and R nodes marked white. Additionally, the graph is made planar by the removal of edge $(2, 4)$.

bottom-set. The union of a vertical and horizontal crossing is called a complete crossing. The TBLR-sets are defined as follows: U is a placeholder for either T, B, L and R respectively.

$$U \neq \emptyset \tag{3}$$
$$\forall v, v' \in U \times U : P_{v,v'} \in U \tag{4}$$
$$T \cap L \neq \emptyset \tag{5}$$
$$T \cap R \neq \emptyset \tag{6}$$
$$B \cap L \neq \emptyset \tag{7}$$
$$B \cap R \neq \emptyset \tag{8}$$

As an example consider the topology in Figure 2. The red nodes belong to the top-set $T = \{15, 9, 3, 6\}$, the blue nodes belong to the bottom-set $B = \{0, 1, 12\}$, the yellow nodes belong to the left-set $L = \{0, 2, 15\}$ and the white nodes belong to the right-set $R = \{12, 8, 5, 6\}$.

Every vertical or horizontal crossing is a read quorum. A vertical read quorum VR for a given graph G is defined as:

$$\exists v \in VR : v \in T \tag{9}$$
$$\exists v' \in VR : v' \in B \tag{10}$$
$$\exists P_{v,v'} \in G \tag{11}$$

A horizontal read quorum HR for a given graph G is defined as:

$$\exists v \in HR : v \in L \tag{12}$$
$$\exists v' \in HR : v' \in R \tag{13}$$
$$\exists P_{v,v'} \in G \tag{14}$$

In order to find a read quorum, a path must be found that connects either a top node to a bottom node or a path that connects a left node to a right node. Formally a write quorum is defined as:

$$WQ = VR \cup HR \tag{15}$$

This means that every union of a vertical crossing and a horizontal crossing is a valid write quorum. Selecting the shortest crossing is important as this crossing minimizes the cost of the read operation. Existing path finding algorithm can be used for this purpose. An efficient algorithm, for example is the Floyd-Warshall algorithm [4] with path reconstruction, as it computes all shortest paths between all nodes. The graph in Figure 3 is non-planar. The edge $(0, 10)$ and the edge between nodes $(2, 4)$ intersect. Non-planarity is an unwanted property as it averts the intersection property of our protocol. As an example, consider the vertical crossing $< 0, 10, 3, 6 >$ and the horizontal crossing $< 2, 4, 14, 12 >$. These crossings do not intersect and are therefore undesired by the Crossing Protocol. To use such a graph, we need to make it planar. Obviously, a graph is made planar by ignoring always one of the intersecting edges. This way we preserve the topology of the graph.

4. Mapping

As explained earlier, existing structured quorum protocols are usually limited to a specific LT. In order to fairly compare our protocol with these protocols, they have to be applied to the PNT our protocol is using. In order to apply structured quorum protocols to arbitrary topologies, either their LT must match the PNT or the nodes of the LT have to be mapped to the PNT. As it is highly unlikely that the LT matches the PNT, the mapping becomes a necessity. The mapped quorum protocol obviously has the disadvantage that it does not have any knowledge of the actual used topology. In order to use the quorums created by the quorum protocol on the PNT, additional nodes may be required to connect the nodes of the quorum in order to achieve 1SR. The number of nodes in the LT and the PNT must be equal. Formally, a mapping can be defined as:

$$M(G, G') = \{(v, v'), \dots (v_n, v'_n)\} \tag{16}$$
$$\forall p \in M : v \in G, v' \in G' \tag{17}$$
$$\forall p \in M : M \setminus p \cup p_v = \emptyset \tag{18}$$
$$\forall p \in M : M \setminus p \cup p_{v'} = \emptyset \tag{19}$$

This says that every node v of graph G is uniquely mapped to a node v' of graph G'. The challenge is to find a mapping that requires as few additional nodes as possible to connect the quorum created for graph G on graph G' and to actually connect the nodes of the quorum with as few additional nodes as possible. Finding a good mapping is a difficult problem as the number of possible mappings grows factorial with the number of nodes in the graph. This is best shown by considering a simple example. A hand optimized mapping from the graph in Figure 1 to the graph in Figure 2 is: $\{(0, 12)$,$(1, 8)$,$(2, 5)$,$(3, 6)$,$(4, 14)$,$(5, 11)$ $(6, 7)$,$(7, 3)$,$(8, 0)$,$(9, 2)$,$(10, 10)$,$(11, 9)$ $(12, 1)$,$(13, 13)$,$(14, 4)$,$(15, 15)\}$. If the TL protocol yields the diagonal as a write quorum and the mapping is as shown above, then finding the shortest connecting path between the four nodes $0, 5, 10, 15$ is a complex problem. If mapped to the graph in Figure 3, with the mapping shown above , the nodes $\{12, 11, 10, 15\}$ have to be connected, as none of these nodes is directly connected.

5. Graph Measures

As we cannot just randomly create graphs and hope that these graphs give meaningful result in our analysis, we have to create graphs that obey certain properties.

The first criterion is the number of vertices of the graph. Empty graphs and graphs with one vertex are not considered as there is no reason to compare quorum protocols on them. Obviously any quorum protocol will have the same availability.

The vertex degree is the number of edges connected to the vertices of the graph. The vertex degree is an indicator on how important a certain vertex is. We only consider undirected graphs. The maximum vertex degree is defined by the number of vertices of the graph. The minimum is one, as we do not consider unconnected vertices.

The connectivity of a graph is defined by the number of vertices or edges that have to be removed before the graph either becomes partitioned or the vertex count becomes one. As mentioned, there are two connectivity measures. Vertex connectivity and edge connectivity [3]. The vertex connectivity $k_v(G)$ describes how many vertices can fail until the graph is partitioned. The Edge Connectivity $k_e(G)$ describes how many edges can fail until the graph becomes partitioned. These two measures are important in the scope of analyses since they define the maximum number of faults, of either vertices or edges, before the graph becomes partitioned and the operations of the quorum protocols potentially unavailable.

The diameter of a graph is the longest path between any two vertices in the graph. This measurement is computed by finding the longest shortest path between any two vertices of the graph. It can be used as an indication of the quorum sizes.

The betweenness is a measure that defines for every vertex of a graph how often it is part of any shortest path in the graph. Through this measure, it is also possible to get an idea of how important an vertex is in order to connect the nodes of the graph efficiently.

Another measure criterion is the number of intersection edges. This measure is easily computed by testing all edges for intersection with all other edges. Currently, our protocol only works on planar graphs. Therefore this measure can be a first indicator on how good our protocol will perform on that graph.

5.1. Graph Homomorphism

A graph is usually defined as homomorph (or equal) if all adjacent vertices are mapped to adjacent vertices in another graph [7]. This means that only the edges and vertices are of interest, but not the position of the vertices. Moving the vertices would therefore not change the graph topology. As presented, our protocol does not modify the topology of the PNT, it only ignores certain edges when creating quorums. A strict interpretation would mean that the coordinates of all vertices must remain the same and all edges must be the same. This requirement will define many graphs as unequal even though all vertices have the same edges and all edge intersections are the same. These two interpretations of equal are in conflict.

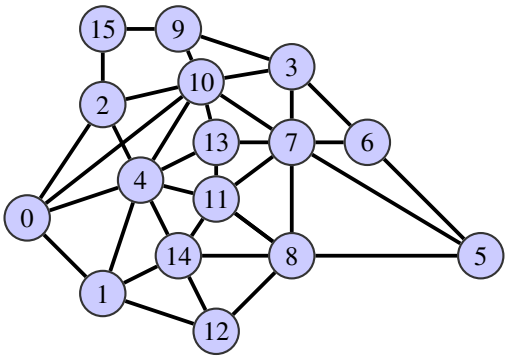


Figure 3: Another Graph

A better notion for equality of graphs is to require the graphs to be homomorphic, preserve all edge intersections and have the same border nodes. This way, we can reduce the number of graphs to be considered significantly. As an example, consider the graphs in Figure 3 and the graph in Figure 3. The only difference is the position of node five and therefore, we can consider them equal in the scope of our analysis.

6. Real-World Topologies

In order to measure the performance of our protocol in a cohesive way we also have to take a look at networks that are found in the real-world. The following network topologies are the most common network topologies found in local-area networks. Combinations of these topologies can be found in many wide-area networks.

6.1. Bus Networks

The bus network topology connects all participating nodes of the network through a common medium. The 10Base2 ethernet standard operates on bus networks [9]. If this medium is unavailable, then no communication is possible. Therefore, the bus is a single-point of failure network. In contrast, the failure of a node does not have any effect on the communication of other nodes.

6.2. Line Networks

The line network is a hybrid between a bus network and a mesh network. Every node of the network is connected

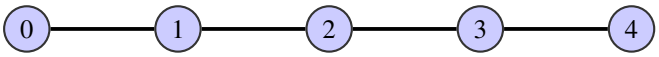


Figure 4: A Line Network

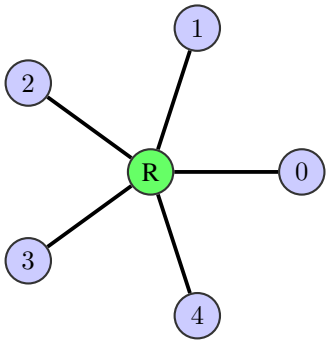


Figure 5: A Star Network

to a maximum of two other nodes. The start and end nodes only have one connection. If an information is to be sent from the start to the end node, the information not only has to pass all connections between all nodes of the network, it also has to pass all nodes in between. Therefore, the number of failure points in a path from the first node to the last node is $|V|+|V|-1$. On the other hand, the failure of connections or nodes may not interfere with the communication of other nodes. For instance, the failure of node 1 in Figure 4 will not prohibit the communication between nodes 2 to 4.

6.3. Star Networks

In a star network, every node in the network is connected to one special node. This special node is logically placed in the center of the topology and connects the other nodes of the network. Thus, making this special node the single point of failure for the network. This special node is what is commonly known as a router. The failure of a node has no influence on other nodes, other than that the failed node is no longer reachable. If a connecting edge fails, only the node connected through this edge is no longer available in the network. As an example, consider node 2 in Figure 5 to be unavailable. Even though this node is unavailable, nodes 1 and 4 are still able to communicate. In contrast, if node R in Figure 5 fails, no more communications between any nodes is possible. The star network topology is used in the 10Base-T standard and its successors [9].

6.4. Ring Networks

A ring network is a line network with one additional connection. This connection is connecting the start and the end a line network. Depending on whether the graph is directed or undirected, the failure of a node will partition the nodes. Even though not found as often in todays networks topologies, the ring topology is historically important and heavily used in theory.

6.5. Mesh Networks

Mesh networks can be described as a combination of the previously presented network topologies. Its topology must not follow a specific design and may even look somewhat chaotic or at least irregular. The most useful way to describe the mesh networks is by the previously present graph measures of Section .

7. Graph Suite Generation

Creating a representative set of graphs is a combinational problem, because graphs are to be created that exhibit a multitude of properties. All these properties have to be combined to multiple graphs with different values. Additionally, to the number of vertices of a graph all presented measurement criterion are part of the state space to consider in the graph creation. To not always create the same graph, a random element is added to the graph creation. This will not eliminate the problem of creating equal graph, but is a trade-off to prevent the problem from interfering constantly.

The idea behind the graph suite generation algorithm is to consider the combinations of the presented measures and their value ranges and then create a graph that matches one concrete value of any of the graph measurements. For example, the graph generator could be executed to create a graph with 16 vertices with a vertex connectivity of 4. The algorithm then places 16 vertices randomly in a plane. The next step is to add edges to each vertex until the range of the vertex degree is reached. The edges are randomly, recursively added and after every addition, the required criteria are checked. If any of the measures is exceeded, the newly created edge is removed again. For many measures the value is inside of a value range or an average value. For instance, fixing the vertex degree to a single natural number will highly limit the number of graphs. Therefore, the value of the analysis based on these graphs is limited.

Some of the measures are continuous values, they cannot be iterated as easily, for instance, the vertex connectivity. Therefore, these values are discretized. Certain measures like the edge and vertex connectivity are closely related and generating graphs where there is no correlation between these values is is hardly possible. Therefore, only one of these measures is used at a time or the required combination of the measures is check beforehand for their feasibility and usefulness. This can also be done at design time of the particular class of graphs.

Additionally, to the mesh network, bases on the measures of Section , real-world topologies, as presented in Section 6, will be generated and analyzed. Also, combinations of these real-world topologies will be generated and

analyzed. Considering all the measures criteria for the real-world topologies is not required as these measures are often derived from their definition. For instance, a star network of star networks, where the R nodes are connected, will always have an edge connectivity of one as no new paths are introduced to bypass edges.

These multiple newly generated graphs are then stored together as a new graph suite. This suite is subsequently used for performance analysis of the Crossing Protocol in comparison to other existing structured quorum protocols.

8. Related Work

Network and graph generation is not a new topic, as many different generators already exists. Most of the graph generators use the same graph measures as presented here. Many of the generators usually only use a subset of the measures presented in this paper. As an example, consider [8]. This graph generator mostly measures the vertex and edge degree as well as the diameter of all vertex pairs. Other generators or measurement tools, as presented in [6] and [10], concentrate on topologies with much more vertices than considered in our analysis. These generators are mostly used to simulate Internet-like topologies. Therefore, these tools have limited usefulness in the scope to our analysis, as the graph we analyze are much smaller. This makes these tools very difficult to apply for our problem. Some generators only focus on specific graphs like [5] which focuses on acyclic graphs. Overall most published graph generators are purpose build, which makes them difficult to use for other scenarios. Additionally, the combination of real-world network topologies and randomly generated mesh networks are not found in existing graph generators.

9. Conclusion

We presented a graph suite generator approach that is suitable for a performance analysis of the Crossing Protocol. The graph suite generator is based on creating graphs that present certain graph properties. These properties were analyzed and selected. Additionally, we presented a short overview of the Crossing Protocol that will be using the graph generator for further analysis. We described why existing quorum protocols have to mapped to a PNT in order to be used on them and we described the mapping process. Additionally, we presented the restrictions and drawbacks of this mapping approach. We discussed why we need a graph suite to benchmark our protocol against existing quorum protocols and presented real-world topologies that are needed to be included in this suite.

In future research, we will use the presented graph suite generator to benchmark our protocol. In order to do a fair comparison to the mapped quorum protocols, we need to find good mappings. As described earlier, the growth rate of the number of mappings is factorial with the number of nodes. Therefore, we will need to develop heuristics for finding good mappings in a reasonable time.

References

[1] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison Wesley, Feb. 1987. ISBN-13 978-0201107159.

[2] S. B. Davidson, H. García-Molina, and D. Skeen. Consistency in partitioned networks. *ACM Computing Surveys*, 17(3):341–370, Sept. 1985.

[3] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[4] R. W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345–, June 1962.

[5] S. Gupta. A scalable acyclic graph generator. *CoRR*, abs/1204.6535, 2012.

[6] H. Haddadi, M. Rio, G. Iannaccone, A. W. Moore, and R. Mortier. Network topologies: Inference, modeling, and generation. *IEEE Communications Surveys and Tutorials*, 10(1-4):48–69, 2008.

[7] P. Hell and J. Nesetril. *Graphs and Homomorphisms*. Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford, 2004.

[8] B. Quoitin. Topology generation based on network design heuristics. In *Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology*, CoNEXT '05, pages 278–279, New York, NY, USA, 2005. ACM.

[9] C. Spurgeon. *Ethernet: The Definitive Guide: The Definitive Guide*. O'Reilly Media, 2000.

[10] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: degree-based vs. structural. In *SIGCOMM*, pages 147–159, 2002.

[11] C. Wu. *Replica Control Protocols that Guarantee High Availability and Low Access Cost*. PhD thesis, University of Illinois, Urbana-Champaign, 1993.

[12] C. Wu and G. G. Belford. The triangular lattice protocol: A highly fault tolerant and highly efficient protocol for replicated data. In *Proceedings of the 11th Symposium on Reliable Distributed Systems (SRDS'92)*. IEEE Computer Society Press, Oct. 1992.

Aplicación de Mapa Cognitivo Compartido en Equipos de Desarrollo de Software Científico-Técnico

Salamon Alicia,
Maller Patricio,
Mira Natalia,
Departamento de Informática
Instituto Universitario Aeronáutico
Av. Fuerza Aérea, 6500, Córdoba, Argentina

Boggio Alejandra,
Pérez Sofía,
Coenda Francisco.
Departamento de Informática
Instituto Universitario Aeronáutico
Av. Fuerza Aérea, 6500, Córdoba, Argentina

as.salamon@gmail.com, pmaller@gmail.com, ncmira@gmail.com, alejandra.boggio@gmail.com, sofiabeatrizperez@gmail.com, franciscocoenda@gmail.com

Abstract

El presente trabajo realiza una adaptación de la metodología Strategic Options Development and Analysis (SODA) en el campo de la mejora continua de procesos de software. La hipótesis de este trabajo es que los mapas conceptuales compartidos, como representación gráfica de los modelos cognitivos de los decisores, son una herramienta que puede ser utilizada en situaciones donde el problema tenga impacto potencial y no haya situaciones de conflicto pre-existentes entre los decisores. Se presenta una experiencia en la que líderes técnicos y de gestión de equipos de software debían identificar y priorizar áreas de mejora siguiendo los lineamientos de SODA y aplicándolo en una intervención grupal para consensuar objetivos comunes y elicitar una visión compartida.

1. Introducción

Las encuestas son herramientas de uso común para investigar áreas de mejora relacionadas a la Ingeniería del Software en equipos de científicos-técnicos que desarrollan software. Este trabajo, para investigar dichas áreas de mejora en lugar de utilizar encuestas, explora un nuevo marco de trabajo para la elicitación y estructuración de la información generada a partir de los miembros del equipo de desarrollo de software científico-técnico utilizando diferentes técnicas como Strategic Options Development and Analysis (SODA), Rejilla de Kelly, Card Sorting y otros. Este enfoque pone énfasis en el consenso del equipo y en el establecimiento de prioridades para identificar las áreas críticas a mejorar.

Este trabajo expone una experiencia de intervención en un área de ingeniería que nuclea desarrolladores de software científico-técnico. Esta área desarrolla e integra software embebido y de simulación bajo estándares internacionales.

Algunos autores presentan estudios realizados en los cuales indican que las diferencias en los procesos de construcción de software empresarial y software científico son tales que los modelos y prácticas utilizadas en el primero no son fácilmente transferibles al segundo sin un esfuerzo importante de adaptación (tayloring) [1]

Algunas diferencias entre el software científico y software empresarial pueden ser apreciadas en la Tabla 1. [2-6]

Tabla 1. Diferencia entre software científico y software empresarial

Software científico	Software Empresarial
El software es un medio para un fin. Es un componente de otro producto mayor.	El software es un fin en sí mismo. Es un producto único.
Se definen los objetivos al comienzo del desarrollo pero no siempre de manera clara y concisa. Puede o no seguir una metodología de desarrollo de software.	Se definen los objetivos de manera clara y concisa al comienzo del proyecto. Se sigue una metodología de desarrollo de software.
Generalmente los desarrolladores y usuarios son los mismos, es difícil encontrar un mercado externo para sus	Generalmente los desarrolladores y los usuarios del sistema son distintas personas

productos.	y existe un amplio mercado externo para sus productos.
Las pruebas y aseguramiento de la calidad son costosas. Se utiliza simulación para llevarlas a cabo ya que las pruebas del experimento físico la mayoría de las veces no se realiza por el costo (económico, social, ambiental) que ello significa.	Las pruebas y aseguramiento de la calidad son más fáciles de llevar a cabo que en el software científico.

En investigaciones a nivel internacional se han utilizado las encuestas, siendo una herramienta de uso común cuando se plantea identificar áreas susceptibles de mejoras en el desarrollo de software, aún así, no se puede negar que el uso de esta técnica como única herramienta presenta serios inconvenientes.

En este sentido, resulta fundamental enfocarse en las dificultades que se presentan en los procesos de desarrollo como una forma para mejorar posteriormente los mismos.

Se detecta la necesidad de implementar una metodología que permita la identificación y priorización de áreas de mejora de procesos, de manera de alcanzar mayor eficiencia, velocidad y calidad en los desarrollos.

Este trabajo describe los resultados de aplicar una versión modificada de la metodología SODA y completar la matriz de constructos personales de Kelly con un grupo de líderes en esta organización, con una adaptación en la formulación de los mapas cognitivos.

En el contexto de mejora continua, es necesario considerar la utilización combinada de diversos enfoques para lograr en una primera etapa la comprensión holística del dominio del problema, y posteriormente abordar la estructuración del mismo. La técnica SODA [1,7] permite contemplar los aportes del equipo de decisores del área, para trabajar luego en la selección y priorización de opciones. Sin embargo, la generación de mapas cognitivos, uno de los puntos más complejos de SODA, presenta una limitación en el tiempo requerido en una actividad que a priori no representa grandes divergencias en los mapeos cognitivos individuales.

La hipótesis de este trabajo es que los problemas de mejora continua constituyen un subconjunto de los problemas complejos, en el contexto de lo Complejo no hay relación evidente entre causas y efectos y sólo se puede avanzar prestando atención a los patrones emergentes [8]

Por lo tanto se deben crear entornos que permitan que los patrones emerjan, por lo que es importante elevar los niveles de interacción y comunicación del equipo; y por último promover que las ideas emerjan a través de la discusión, o marcando desafíos, fomentando la diversidad, y que pueden ser tratados con herramientas de menor costo y complejidad que los mapas cognitivos individuales, tales como las herramientas de la Investigación Operativa Soft [9,10] y como el Repertory Grid de Kelly [11], y aun así conservar las salidas esperadas de una estructuración del problema bajo los lineamientos de la metodología SODA.

La estructura de presentación de este trabajo es, en una primera instancia, a modo de marco teórico, en el cual se hace referencia a los recursos metodológicos utilizados durante la experiencia. Posteriormente se realiza una breve descripción de la justificación para modificar la utilización de mapas cognitivos, a ésta le sigue la descripción de la aplicación concreta en el taller realizado con los líderes decisores y se consignan las instancias desarrolladas en el mismo. Finalmente se presentan algunas conclusiones sobre la experiencia que delimita el alcance del trabajo.

2. Recursos metodológicos

Para abordar la problemática, se utilizó como marco conceptual SODA, un marco para la estructuración de problemas elaborado en la segunda parte de la década del 70. El enfoque de SODA se basa en el subjetivismo, donde el saber y la experiencia de los integrantes del equipo constituyen un elemento clave para el desarrollo de decisiones confiables y sustentables. La estructuración de problemas está guiada por la "Teoría de las Construcciones Personales", teoría cognitiva que sostiene que los individuos se esfuerzan por dar sentido a su mundo con el fin de manejarlo y controlarlo. [12-14]. Las herramientas clásicas de SODA son los mapas cognitivos, a nivel individual y colectivo.

Se realizó una dinámica de elicitación utilizando card sorting para indagar acerca de la forma en que cada uno de los responsables de la gestión evaluaría áreas de mejora en el proceso de desarrollo de software. Esta tarea se apoya en las teorías de clasificación que se centran en las categorías y en las relaciones jerárquicas entre conceptos. Esto es, las estructuras mentales de los individuos presentan un ordenamiento jerárquico, de manera que cada grupo está incluido en otro de orden superior [15] y, a su vez, aglutina a varios de orden inferior [16,17]. En esta etapa, y aquí la diferenciación con la utilización tradicional de mapas cognitivos, no se construyeron los mapas individuales, sino que se diseñó una dinámica para favorecer la construcción grupal del mapa.

Se utilizó la técnica Repertory Grid creada por Kelly [11] está diseñada para interpretar el modo en que una persona da sentido a sus experiencias. Repertory Grid es un instrumento de evaluación de las dimensiones y estructura del sistema de construcción personal que utiliza la valoración de elementos asociados a constructos bipolares. [18,19], se utiliza para identificar tanto

coincidencias como disidencias en un grupo de investigadores. Dicha grilla permitió limpiar conceptos y organizarlos en una matriz (Kelly afirma que las personas buscan encajar las experiencias del mundo en constructos cognitivos). Se define como un constructo a una categoría descriptiva que permite clasificar los acontecimientos y que representa la precepción de la realidad. Se generó una matriz de datos conocida como Matriz de Kelly. La técnica en cuestión utilizada es un instrumento de evaluación de las dimensiones y estructura del significado personal que se deriva de la "Teoría de los Constructos Personales", estas dimensiones o elementos son los aspectos relevantes en el contexto que se está evaluando. Esta matriz empleada: Matriz de Kelly, es un sistema bidimensional de elementos y constructos donde los mismos están interceptados entre sí. Los elementos están ubicados en la parte superior de la matriz: columnas, y los constructos en las filas.

A partir de la teoría de Kelly, los constructos bipolares fueron un importante mecanismo para organizar, simplificar e interpretar el "entorno".

En la matriz de la Tabla 2 se observan los constructos bipolares emergentes (positivos) a la derecha y los constructos bipolares opuestos (negativos) a la izquierda de cada fila.

Tabla 2. Matriz con constructos bipolares

	Elemento 1	Elemento 2	
Constructo Negativo 1			Constructo 1 Positivo 1
Constructo Negativo 2			Constructo 2 Positivo 2

Como se mencionó anteriormente, los subsistemas están relacionados entre sí, en este caso se realizó a través del Método de Contexto Mínimo donde Kelly plantea la técnica de la tríada: evaluando tres elementos describir una característica común a dos de ellos (medida de similitud), y que a su vez los diferenciara del tercero (medida de contraste), de esta manera se tomaron dichas medidas como valores de bipolaridad, reflejando la posibilidad de moverse en la misma dimensión entre dos valores identificados, ayudando así a validar los constructos expuestos.

3. Técnica de mapa cognitivo compartido

Aunque la técnica propuesta por SODA claramente son los mapas cognitivos, la utilidad de la aplicación en su máxima expresión ha sido cuestionada en investigaciones. La necesidad real de un mapa cognitivo compartido fue analizada por Langfield-Smith [20], concluyendo que la capacidad para mantener ese nivel de creencias colectivas es transitoria, y por lo tanto no resulta crítico para el proceso de toma de decisiones. Un componente clave, sin embargo, se refiere al construccionismo ejercitado por el grupo en la creación del mapa cognitivo, como un aspecto a ser preservado [21]. En términos de economías, un mapa cognitivo colectivo derivado a partir de la agregación de los mapas cognitivos individuales puede contener unos 1000 nodos, lo que representaría en la mayoría de los casos una sobreingeniería de la actividad de mejora. Siendo en estos casos que involucran problemas como los descriptos anteriormente en los que los participantes para la construcción del mapa comparten el conocimiento homogéneo del dominio, un subgrupo factible a tratar con la opción de Repertory Grid entre otras técnicas para construir el mapa grupal, sin pasar primero por la construcción de los mapas individuales y luego construir el mapa grupal [22].

Nuestra fundamentación para sustituir la generación de mapas cognitivos individuales por una dinámica de grupo tiene su base en el tipo de objetivos y grado de impacto de los proyectos de mejora continua. La mejora continua, si bien cumple con las características de los problemas complejos, tiene tres características que son intrínsecas al dominio y que generan un pre-acuerdo entre decisores:

- La mejora implica una ganancia sobre la situación actual, y el retorno de inversión puede estimarse.
- Los decisores pueden variar su grado de compromiso con el proyecto, y en general, sus responsabilidades primarias incluyen tangencialmente la mejora.
- Hay una baja fragmentación de la visión.

El mapa cognitivo compartido como herramienta de soporte a la estructuración tiene como objetivo principal el relevamiento, acuerdo y consenso que potencialmente puede lograrse a partir de la conducción del grupo en una situación concreta y particular a fin de obtener una visión compartida acerca de dicha situación [23-26].

4. Experiencia

La experiencia se desarrolló durante una jornada completa, en 3 fases específicas y de diferente duración.

En la primera fase se realizó una intervención informal a través de una discusión y debate acerca de la situación actual y características de los procesos de desarrollo de software extraídos de la industria, así como experiencias anteriores en el área de desarrollo de software científico-técnico. El objetivo principal de esta fase fue promover la interacción del grupo para que construyan consenso como observadores del problema. Seguido, los entrevistados expresaron como participantes cómo ven los procesos de desarrollo de software en los que son actores y cumplen algún rol, utilizando lenguaje

coloquial con marcada tendencia técnica, como era de esperarse. Para esta actividad los facilitadores de este taller trataron de atemperar los liderazgos más fuertes y alentar los más tranquilos, de forma que el equipo de trabajo fuera una estructura plana y sin atractores. Se capturaron las frases/conceptos/ideas más representativas. Para esta actividad se decidió utilizar la técnica de card sorting [27], a fin de categorizar los contenidos agrupando y asociando entre sí un número predeterminado de tarjetas (4 por persona) etiquetadas con las diferentes categorías temáticas (surgidas de la interacción y el acuerdo entre los participantes como grandes agrupaciones de aspectos: Metodología, Soporte, Capacitación y Gestión). Así, cada participante completó las 4 tarjetas mencionadas consignando en ellas características/percepciones de la situación de los procesos de desarrollo. De esta manera, partiendo del comportamiento de los propios participantes, se organizó la información conforme al modelo mental de los mismos. Con esta aplicación del concepto de clasificación se logró manejar y reducir la información compleja acerca de la realidad del desarrollo de software científico-técnico, con la finalidad de mejorar la comprensión del entorno, generando así categorías cognitivas, es decir, un conjunto de concepciones intelectualmente organizadas que referencian la realidad del sujeto, tales como las dificultades que el desarrollador asume, considera y percibe críticas en su proceso de construcción, y que conforman así los grupos cognitivos.

En la segunda fase del taller se utilizó la Repertory Grid de Kelly, que permitió limpiar conceptos y organizarlos en una matriz de datos. Los mismos debieron ser representativos de lo que está sujeto a análisis, homogéneos y lo más precisos posibles. Esta matriz empleada: Matriz de Kelly, es un sistema bidimensional de elementos y constructos: los elementos están ubicados en la parte superior de la matriz: columnas, y los constructos en las filas.

Como se mencionó anteriormente, los subsistemas están relacionados entre sí a través del Método de Contexto Mínimo y la técnica de la tríada: la misma consistió en presentar al individuo tres elementos y pedirle que describiera una característica común a dos de ellos, y que a su vez los diferenciara del tercero, de esta manera se tomaron dichas medidas como valores de bipolaridad, reflejando la posibilidad de moverse en la misma dimensión entre dos valores identificados, ayudando así a validar los constructos expuestos. Por ejemplo, la rotación en los cargos directivos caracteriza a las actividades de Gestión y Metodología, no en la misma medida que las actividades de Soporte.

Paso siguiente se completó la matriz con valores. Esta matriz construida en el taller representó los constructos obtenidos conjuntamente entre todos los participantes. Se organizaron en las columnas los elementos: Metodología, Soporte, Capacitación y Gestión identificados en la fase 1, en las filas se ubicaron las ideas extraídas también en la fase anterior, mostrando las proposiciones positivas a la derecha y las negativas a la izquierda, luego se completaron los espacios faltantes (ya sea a la derecha o a la izquierda). (Tabla 3). Posteriormente, de acuerdo a una escala, se categorizó cada elemento en función de la valoración realizada sobre cada constructo (Tabla 3), es decir, se consensuó la medida en la que dicha característica se presenta en un elemento en la realidad de los participantes.

Tabla 3. Matriz grupal resultante obtenida

	M	C	S	G	
Falta de experiencia	6	1	6	1	Mayor capacitación
Conocimientos básicos	1	6	2	6	Capacitación especializada
Objetivos estratégicos confusos	1	6	1	6	Objetivos estratégicos claros
Ausencia procedimientos y pautas de trabajo	5	5	1	6	Procedimientos y pautas de trabajo claras
Falta de planificación de las actividades	5	1	6	1	Llevar metodologías de ejecución
Inestabilidad de los plazos	6	2	6	1	Estabilidad de los plazos
Escases de herramientas y recursos	4	5	5	5	Documentar de manera ágil
Resistencia al cambio por parte de las personas	4	2	2	5	Personas proactivas
Alta rotación de cargos directivos	4	1	1	6	Poca rotación de cargos directivos
Presupuestos bajos	1	6	6	2	Presupuestos más altos

M=Metodología, C=Capacitación, S=Soporte y Gestión

La tercera fase abarcó la construcción de un mapa cognitivo colectivo. Como era de esperarse, el resultado presentó un alto nivel de consenso. A partir de las ideas (constructos) obtenidos en la fase anterior se generó el mapa cognitivo grupal: se ubicaron los conceptos y se relacionaron entre sí, nuevamente a partir de la discusión y consenso grupal. Se expresaron en forma de acciones deseables positivas. Para la construcción y elaboración del mapa se utilizó el enfoque "escalonado hacia abajo", es decir, trabajando desde las opciones hacia la meta: La mejora de la calidad de productos software. Esto es, a partir de la matriz de Kelly se tomaron los valores positivos de la derecha y se ordenaron a partir de la

interacción con los participantes, llegando a la meta del equipo.

Como resultado se posicionaron en las colas las potenciales acciones/aspectos a considerar y sobre los cuales accionar para lograr la mejora objeto de realización de este taller. (Figura 1) En este sentido, conocimiento compartido de los objetivos estratégicos, enfoque en la capacitación, aplicación de presupuestos acordes, etc. son las acciones a considerar sobre las cuales la ejecución de acciones terminaría impactando en la situación inicialmente planteada.

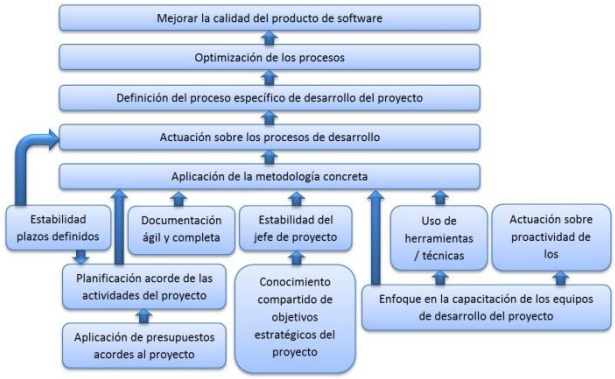


Fig. 1. Mapa cognitivo grupal obtenido

5. Conclusión

La situación problema identificada en este trabajo, está dentro de la categoría de problemas complejos, pero con las particularidades de las iniciativas de mejora continua.

En ese contexto se realizó una experiencia de intervención basado en SODA adaptado, donde se realizó una modificación en el empleo de los mapas cognitivos. De esta manera, las expectativas de identificación y priorización de las áreas de mejora más críticas o sensibles fueron cumplidas y consensuadas, y toda la actividad pudo llevarse a cabo en una jornada.

La adaptación de las técnicas para abordar este tipo de problema permite la utilización de técnicas y herramientas acorde a los niveles de inversión y retorno esperado de los proyectos.

6. Referencias

[1] Rosenhead J. y Mingers J. "Análisis Racional Reestudiado para un Mundo Problemático: Métodos para estructurar problemas en condiciones de complejidad, incertidumbre y conflicto." *Iveplan Instituto Venezolano de Planificación*, Caracas, Venezuela. (2001).

[2] Segal J. Scientists and software engineers: a tale of two cultures. *PPIG University of Lancaster*, UK, (2008).

[3] Segal J. Some problems of professional end user developers. *IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC)*, Coeur dAlene, Idaho, USA. (2007).

[4] Chalmers A. What is this called science?. *Open University Press*. Milton Keynes, UK. (1982).

[5] D. Kelly and R. Sanders. Assessing the Quality of Scientific Software. *First International Workshop on Software Engineering for Computational Science and Engineering*. Leipzig, Germany. (2008).

[6] Carver J., Kendall R., Squires S., and Post D. Software Development Environments for Scientific and Engineering Software: A Series of Case Studies. *IEEE Computer Society*, Washington, DC, USA, (2007).

[7] Ion G. "Cognitive mapping and strategic options development and analysis (SODA)". *Wiley Encyclopedia of Operations Reasearch and Management Science, edited by Cochran J. J. Copyright John Wiley & Sons, Inc.* (2010).

[8] Snowden D. y Boone M. "A Leader's Framework for Decision Making" *Harvard Business Review*. 2007.

[9] Zanazzi J., Conforte J., Boaglio L., Dimitroff M. y Carignano C. "Problemática de trabajo en equipo. Metodología para el análisis conjunto". *I ERABIO (Encuentro Regional Argentino Brasilero de Investigación Operativa) - XXI ENDIO*. Posadas, Argentina. (2008).

[10] Zanazzi J., Salamon A., Cabrera G., Gonzalez A. y Pedrotti B. "La investigación operativa soft en la estructuración de problemas vinculados con la orientación vocacional" *XXII EPIO - XXIV ENDIO*. Rio Cuarto, Córdoba. Argentina. (2011).

[11] Kelly G. A. "The psychology of personal constructs" *Norton and Company*, N. York. (1955).

[12] Eden C. y Ackermann S. "Using repertory grid for problem construction". *European Journal of operational research, vol. 35 (9), pp779-790*. (1998).

[13] Stewart V. y Stewart A. "Business applications of repertory grid." *McGraw-Hill Book Company*. Berkshire England. (1981).

[14] Lemke F., Clark M. y Wilson H. (2010): "Customer experience quality: an exploration in business and consumer contexts using repertory grid technique". *Journal of the Academy of Marketing Science, Disponible on line*, (2011).

[15] Saez Martínez F. J. "Las configuraciones cognoscitivas como herramienta de análisis de la estructura sectorial". *Revista europea de dirección y economía de la empresa, vol. 14 Nº3, pp. 111 – 134.* (2005).

[16] Rosch E. "Principles of categorization". Publicado en Rosch y lloyd "Cognition and categorization". *Lawrence Erlbaum Associates pp. 24-48.*, New Jersey, (1978).

[17] Anderson J. R. "Cognitive Psychology". *Foreman.* Nueva York. (1985).

[18] Feixas G. y Cornejo J. M. "Manual de la técnica de la rejilla mediante el programa record v.2.2". Paidos, Barcelona. (1996).

[19] Alexander P., Van Loggerenberg J., Lotriet H. y Phahlamohlaka J. "The use of the repertory grid for collaboration and reflection in a research context" *Group Decision and Negotiation, vol. 19, pp. 479-504.* (2010)
[20] Langfield-Smith K. "Exploring the need for a Shared Cognitive Map". *Journal of Management Studies, vol. 29(3), pp. 249-368.* (1992).
[21] Martínez Sánchez B. "Estructuración cognitiva del mundo vocacional. Departamento de Psicología Evolutiva y de la Educación". *Tesis Doctoral. Universitat de Valencia, Servei de Publicacions.* (2005).
[22] Kerievsky J. "Stop Over-Engineering!" *Software Development, vol. 10, Nº 4.* (2002)
[23] Alles M. A. "Dirección estratégica de recursos humanos. Gestión por competencias." *Ediciones Granica S.A.* Argentina. (2000)
[24] Alles M. A. "Desempeño por competencias." *Ediciones Granica S.A.* Argentina. (2002)
[25] Levy-Leboyer C. "La Gestión de las Competencias". *Gestión 2000.* Barcelona. (1997):
[26] Gasalla J. M. (2004): La nueva dirección de personas. Ed: Pirámide Madrid.
[27] Rugg G. y Mcgeorge P. "The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts". *Expert Systems, vol. 22, Nº 3.* (2005)

Validando aplicaciones para Ciudades Inteligentes - Recolección de Residuos Urbanos

Raúl Luft, Darío Sosa, Diego A. Godoy, Eduardo O. Sosa, Juan D. Benitez
Centro de Investigación en Tecnología de la Información y Comunicaciones
Departamento de Ingeniería y Ciencias de la Producción
Universidad Gastón Dachary
Av. López y Planes 6519- Posadas, Misiones, Argentina. Teléfono: +54-376-4438677
{rluft,suedsosa,diegodoy,eduardo.sosa,juan.benitez}@citic.dachary.edu.ar

Resumen

En este trabajo se presenta el diseño de un prototipo de software que utilizando los datos generados por una red de sensores inalámbricos, permita determinar, aquellos contenedores de residuos urbanos que ameriten ser recogidos; y en base a esta información, calcular una ruta óptima de recolección.

La arquitectura del sistema está conformada por tres capas: infraestructura, procesamiento y vista. La capa de infraestructura se compone de una serie de nodos sensores ubicados dentro de los contenedores. Los nodos capturan los movimientos y la iluminancia del interior del contenedor y posteriormente envían tramas con esta información a una estación base. La capa de procesamiento, realiza el análisis y procesamiento de las tramas recibidas y calcula una ruta óptima de recolección. La capa de vista utiliza una interfaz web para brindar información a los usuarios sobre la ubicación de los contenedores a recoger.

Palabras claves: *Ciudades Inteligentes, Internet de las Cosas, Redes de Sensores Inalambricos, Recoleccion de residuos urbanos.*

1. Introducción

Dado que en la actualidad el mayor porcentaje de la población mundial reside en ciudades en vez de en zonas rurales, los espacios urbanos se encuentran cada vez más poblados, por lo que se deben afrontar problemas relativos a la gestión de recursos, movilidad urbana, tráfico, eficiencia energética y servicios públicos

Ante este escenario, se incrementarán las necesidades energéticas, se generará una gran cantidad de polución y de residuos.

Gracias al avance de las tecnologías de la información y las comunicaciones (TICs), el abaratamiento de equipos y servicios de telecomunicaciones, y su disponibilidad, hace pensar que la aplicación de las mismas proveería grandes beneficios a los espacios urbanos.

De la aplicación del concepto anterior surge la concepción de Ciudad Inteligente. Se define así a una ciudad que utiliza las tecnologías de la información y las comunicaciones, para hacer que sus actividades, elementos y servicios públicos, sean más eficientes e interactivos.[1]

Uno de los servicios públicos indispensables es la recolección de residuos. El objeto de este servicio es retirar los residuos generados por la comunidad y gestionarlos de manera adecuada para evitar perjuicios a la salud y al ambiente.

La ciudad de Posadas, Misiones, posee actualmente una población de 324.756 habitantes[2] por tanto puede ser considerada una ciudad con una población importante, que genera una gran cantidad de residuos urbanos.

Para llevar a cabo la recolección de residuos, la municipalidad de esta ciudad, cuenta con camiones especializados, contenedores de distintos tipos y capacidades, y personal calificado para realizar la tarea.

Existen rutas de recolección preestablecidas que cubren la totalidad de las zonas de la ciudad. Dichas rutas poseen distintas frecuencias de recolección. En zonas céntricas la frecuencia es de dos veces por día, en zonas cercanas al centro la recolección se realiza una vez al día, mientras que en las zonas más alejadas se realiza tres veces por semana.

En el trayecto mencionado, los recolectores pueden encontrarse con contenedores con niveles variables de desechos, sin embargo, deben revisar cada uno de los mismos, produciendo pérdida de tiempo y desgaste físico. Por otro lado, si en una ruta, los primeros contenedores que se recolectan están al nivel máximo de llenado, la capacidad del camión se completará antes de finalizar el recorrido, provocando mayor cantidad de viajes, mayor gasto en combustible y por lo tanto mayores emisiones de CO_2, elemento considerado causante directo del efecto invernadero.[3]

Tanto la capacidad de los camiones, las distancias del recorrido, el crecimiento industrial y urbano, sumado a un posible mal diseño de rutas, generan un servicio ineficiente de recolección de residuos.

Ante esta situación, es sumamente interesante contar con un sistema basado en un nuevo paradigma que consiste en utilizar Redes de Sensores Inalámbricos para la detección de los contenedores de residuos que merecen ser recogidos y calcular la ruta óptima para llegar a cada uno de ellos, lo cual podría maximizar el rendimiento laboral, energético y ecológico.

El presente trabajo está estructurado de la siguiente manera: en la sección 2 y 3 se dará una introducción a la Internet de las Cosas y las Ciudades Inteligentes respectivamente. En la sección 4 se darán los detalles de construcción de la solución, dispositivos de hardware y herramientas de software utilizadas. Seguidamente en la sección 5, se presentarán las pruebas en laboratorio y en campo y los resultados obtenidos. Para finalizar se presentarán las conclusiones y trabajos futuros respectivamente.

2. Internet de las Cosas

En septiembre de 1991, Mark Weiser publicó un artículo titulado "The Computer for the 21st Century"[4], en el que describió su idea de lo que él llamaba "Computación Ubicua". La idea consistía en desarrollar entornos donde existieran objetos de la vida cotidiana con capacidad de cómputo y comunicación inalámbrica, que interactúen con otros objetos para facilitar las tareas a las personas.

El principio de la Computación Ubicua consiste en crear dispositivos cada vez más inteligentes que se mezclen en la vida diaria de las personas hasta que lleguen a ser indistinguibles. De esta manera, las personas se centrarían en las tareas que deban realizar y no en las herramientas que deben utilizar para llevar a cabo las mismas.

Para lograr un entorno ubicuo, los objetos no deben interferir con las actividades para las que son usadas, deben proporcionar un uso cómodo, sencillo y útil; además deben cumplir las siguientes propiedades:

- Deben comunicarse con otros objetos: Los objetos deben comunicarse no sólo con el usuario, sino también con los demás objetos integrados que existan a su alrededor.
- Deben tener memoria: Los objetos deben tener capacidad de memoria y utilizar la misma para mejorar la interacción con los demás dispositivos.
- Deben ser sensibles al contexto: los objetos deben adaptarse a posibles situaciones, como la situación geográfica, dispositivos a su alrededor, preferencias de los usuarios, etc. y actuar dependiendo del entorno que los rodea.
- Deben ser reactivos: Deben reaccionar ante determinados eventos que pueden percibir en su entorno mediante sensores, o a través de la interacción con otros dispositivos.

La visión de Weiser estaba bastante alejada de su época, debido a que no existía la tecnología necesaria para llevarla a cabo, sin embargo, dos décadas después gracias a los avances tecnológicos en las comunicaciones y en los dispositivos, se puede decir que su idea empieza a aplicarse. De esta manera, surgió un nuevo concepto a partir de la Computación Ubicua, denominado "Internet del Futuro" o "Internet de las Cosas" (IoT por sus siglas en inglés Internet of Things).

La Internet de las Cosas (IoT) consiste en que las cosas tengan conexión a Internet en cualquier momento y lugar.

La Internet de las Cosas implica que todo objeto puede ser una fuente de datos y que el comportamiento de todo puede ser monitorizado a través del tiempo y el espacio.

En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible. Dado su tamaño y coste, los sensores son fácilmente integrables en hogares, entornos de trabajo y lugares públicos. De esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la Red. Esto está empezando a transformar la forma de hacer negocios, la organización del sector público y el día a día de millones de personas.

2.1. Tecnologías facilitadoras.

Para entender el Internet de las Cosas desde un punto de vista más técnico, es necesario comprender las tres capas que lo posibilitan hoy en día [5]. Estas tres capas son la miniaturización del hardware, la infraestructura y el desarrollo de aplicaciones y servicios.

2.1.1. La miniaturización del hardware

La miniaturización de los componentes permite desarrollar ordenadores cada vez más potentes y rápidos que sus predecesores.

En 1965, Gordon Moore afirmó que el número de transistores por unidad de superficie en circuitos integrados se duplicaba cada año y que la tendencia continuaría durante las siguientes dos décadas. Más tarde, en 1975, modificó su propia ley al corroborar que el ritmo bajaría, y que la capacidad de integración se duplicaría aproximadamente cada 24 meses. Esto es lo que se conoce como Ley de Moore [6]. Con una mayor

cantidad de transistores aumenta la velocidad de cálculo de los ordenadores.

Moore también sugirió una disminución de costes, ya que los componentes con base de silicio utilizados en los ordenadores crecerían en rendimiento a la vez que se volverían más económicos de producir y abundantes en nuestra vida diaria.

Esto se ha venido cumpliendo hasta el día de hoy, pero no sólo la capacidad de cómputo de los procesadores ha avanzado rápidamente, sino también la capacidad de almacenamiento, el ancho de banda para las comunicaciones, etc. En resumen, cada vez se fabrican dispositivos más baratos, más pequeños y más potentes. Y no parece que vaya a parar este crecimiento, sino todo lo contrario, la previsión para los próximos tiempos es que siga ocurriendo lo mismo.

El avance de la miniaturización de los componentes electrónicos ha permitido el desarrollo de sensores cada vez más pequeños y baratos, que son integrados en objetos cotidianos, conectando el mundo físico con el mundo digital.

Los sensores son piezas fundamentales en el Internet de las Cosas, debido a que poseen tres propiedades que posibilitan que se lleve a cabo. Primero, permiten capturar información tanto del entorno como del objeto en el que se encuentran integrado, para un análisis posterior. Por ejemplo, las compañías de seguros pueden utilizar sistemas de sensores para recuperar información de las pautas de conducción de sus asegurados.

Segundo, los sensores pueden actuar como desencadenantes de una acción, permitiendo la automatización de determinadas funciones. Este puede ser el caso de la activación de una alarma por la detección de una persona no autorizada o el frenado automático de un coche ante la inminente colisión con otro vehículo.

Por último, los sensores también tenderán a ser localizables en todo momento, con lo que se expande el rango de aplicaciones. Por ejemplo, la localización de los paquetes en los sistemas de gestión de logística permite la determinación de su posición exacta a lo largo de un trayecto y la comunicación con las cintas transportadoras para establecer el destino de las mercancías.

2.1.2. Segunda Capa: La infraestructura

Gracias a la infraestructura móvil, en la actualidad Internet se encuentra en todas partes, sin embargo la capacidad de las infraestructuras de telefonía móvil es limitada y la proliferación de los teléfonos inteligentes (smartphones) está saturando la capacidad de las redes. Si a esto se suman millones de nuevos dispositivos conectados a Internet, la tecnología 3G o LTE no serán suficientes. La combinación del uso del móvil, la conexión Wi-Fi y la fibra óptica cobran vital importancia para subsanar la saturación de las infraestructuras móviles.

Ante esta situación algunos gobiernos han comenzado a tomar conciencia y están implementando algunas medidas. Por ejemplo el gobierno de China ha propuesto transformar las cabinas de teléfono públicas en “puntos de acceso” (Wi-Fi hotspots) con conexión a internet para hacer del país una red gigante de acceso a conexión Wi-Fi. Los hotspots son zonas que se encuentran en lugares públicos, como cafeterías, aeropuertos y bibliotecas, donde se permite conectar ordenadores o teléfonos móviles a Internet a través de una conexión Wi-Fi.

. En nuestro país ya existen varios lugares donde se brinda conexión Wi-Fi gratuita, por ejemplo en Capital Federal [7] se provee de esta tecnología en lugares públicos. Otro ejemplo se ha implementado en la ciudad de Posadas, provincia de Misiones [8], donde el gobierno a provisto de puntos de acceso distribuidos en la zona céntrica de la ciudad para brindar acceso gratuito a Internet

La Internet de las Cosas pretende la interconexión de objetos y localizaciones del mundo real, por ello, el direccionamiento de objetos también es otro problema debido a la escasez de direcciones globales. IPv6 ha ampliado la capacidad de direccionamiento global; esto es un gran avance porque se prevé el agotamiento de direcciones IPv4 públicas en los próximos años [9]. Sin embargo, esta no sería la única alternativa de direcciomiento que se utilizaría para localizar objetos debido a los miles de millones de objetos que deberían estar conectados.

2.1.3. Tercera Capa: Las Aplicaciones y Servicios

La tercera y última capa la forman las aplicaciones y los servicios que ponen en uso la gran cantidad de información creada a partir del IoT y donde se encuentra el mayor potencial de creación de valor. Estas aplicaciones conllevan la creación de nuevos modelos de negocio e iniciativas empresariales muy interesantes desde el punto de vista de la innovación.

Las aplicaciones que se pueden desarrollar y los servicios que se pueden brindar a través del IoT, son prácticamente infinitas, algunas de estas aplicaciones se pueden agrupar de la siguiente forma:

- Automatización de vehículos.
- Logística de transportes.
- Control Domótico de casas y edificios.
- Monitorización Ambiental.
- Control de Almacenes y Logística.
- Cuidado de la Salud.

- Agricultura y Ganadería de Precisión.
- Monitorización de Estructuras.

El verdadero valor de negocio va a venir de la mano de estas aplicaciones y servicios que utilicen la nueva información que se está generando en el IoT. Van a surgir nuevos métodos de análisis inteligente de la información y las tareas manuales van a ser sustituidas por automatismos que operen en base a dicha información, prescindiendo de la intervención humana. El desarrollo de software óptimo para este tipo de tareas va a enfrentarse al elevado ritmo de las oportunidades. Emprendedores y empresas ya establecidas desarrollarán nuevas aplicaciones que exploten nuevas fuentes de información y cambien radicalmente los modelos de negocio.

3. Ciudades inteligentes

Se considera "Inteligente" a aquellas localidades donde la estrategia consiste fundamentalmente en mitigar los problemas generados por el crecimiento de la población urbana y la rápida urbanización, empleando para ello todas las posibilidades ofrecidas por la tecnología emergente de las TIC´s; con el objetivo que la vida cotidiana de los ciudadanos sea cada vez más simple y conveniente, en concomitancia directa con los efectos de la contaminación ambiental, control de tránsito vehicular y la utilización eficiente de la energía.

La tecnología ha transformado, en un período de tiempo increíblemente corto, el orden del conocimiento y la innovación a escala global, provocando con ello nuevos desafíos y aparición de oportunidades hasta hoy desconocidas y desaprovechadas. Las ciudades no pueden competir en este nuevo entorno, a menos que sean más innovadoras y capaces de responder más eficazmente a las preferencias y necesidades de sus ciudadanos.

El concepto de "Ciudad Inteligente" ha ido ganando popularidad con el paso del tiempo en todos los continentes, extendiéndose rápidamente en el contexto europeo y en algunos países de Asia, lugares donde estas tecnologías se despliegan en el mundo real no sólo con el fin de analizar su viabilidad, sino también para considerar su impacto socio-económico.

De esta manera las ciudades conforman, hace un buen tiempo, un tecno-paisaje urbano en nuestra sociedad, conformando un nuevo modelo productivo basado en el conocimiento, y en la utilización del mismo como aporte de soluciones técnicas que se ocupan de una mejor calidad de vida de los ciudadanos de una forma sustentable. Sin embargo no se observa actividad incipiente sobre el tema en las ciudades latinoamericanas.

En Argentina, no existe un proyecto integrador de las experiencias sobre el tema, como lo es el ejemplo español con la Red Española de Ciudades Inteligentes[10], donde los alcaldes de las diferentes ciudades españolas han convenido en aunar esfuerzos en la ruta hacia un objetivo común.

Hoy en día hay clara idea de la necesidad, y de la utilidad de potenciar y evolucionar hacia lo que denominamos las ciudades inteligentes.

4. Construcción de la Solución.

Durante esta etapa se realizó una entrevista al Ing. Ricardo Charón, encargado de la Dirección de Servicios Públicos de la Municipalidad de la ciudad de Posadas. En la misma se analizaron diferentes temas relacionados al servicio de recolección de residuos urbanos, como ser:

- Recursos utilizados para llevar a cabo este servicio (camiones y contenedores de distintos tipos y capacidades).
- Frecuencia de recolección en zonas céntricas de la ciudad, en zonas cercanas al centro y en zonas más alejadas.
- Problemática de la recolección de residuos, en donde se destacó el costo (en dinero y tiempo) que significa realizar el recorrido de todos los contenedores de la ciudad, tengan o no residuos.

Esta entrevista permitió obtener un conocimiento sobre el funcionamiento actual del servicio de recolección de residuos urbanos y en base a esto se propuso un prototipo de software que permita realizar la monitorización del estado de contenedores de residuos urbanos, a través de una Red de Sensores Inalámbricos.

4.1. Arquitectura del sistema

El sistema propuesto utilizará sensores de movimiento y sensores de luz para determinar el nivel de llenado de los contenedores. Luego se transmitirá la información generada por estos sensores hacia una estación central, por medio de la infraestructura provista por la red de sensores inalámbrica.

En la estación base se evaluará la información recibida y se actualizarán los datos de llenado de los contenedores (cantidad de movimientos e iluminancia). Posteriormente se realizará el cálculo de ruta óptima, seleccionado aquellos contenedores que ameritan ser recogidos.

Por último, se visualizará en un mapa la ruta óptima, tomando como referencia la posición geo-referencial de

los contenedores (latitud-longitud), indicando el punto de inicio, los contenedores a recoger y el fin del recorrido. En la figura 1 se puede apreciar la arquitectura del sistema.

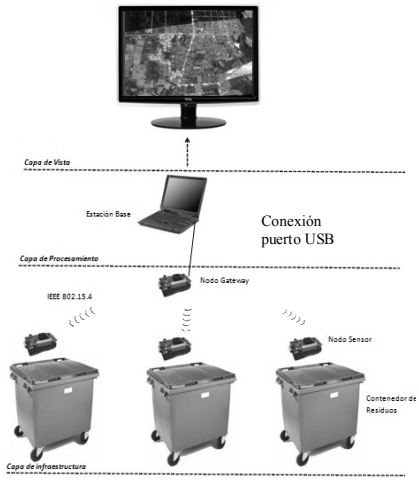


Figura 1 - Arquitectura del Sistema Propuesto

El detalle de la funcionalidad de cada capa se describe a continuación.

4.1.1. Capa de infraestructura

La capa de infraestructura se compone de una serie de nodos sensores ubicados dentro de los contenedores de residuos urbanos y de un nodo gateway.

En esta capa, se encuentra los siguientes subsistemas:

Subsistema de detección de movimientos del nodo PIR, Subsistema de detección de luz del nodo iluminancia y el Subsistema sumidero de datos.

Los nodos con sensores PIR, de aquí en adelante: *nodo sensor pir*, capturan los movimientos que ocurren en el interior de los contenedores (causados por la introducción de objetos). Luego se generan tramas de datos en las que incluyen, la dirección MAC del nodo y el valor actual del contador de movimientos.

Los nodos con sensores de luz, de aquí en adelante: *nodo sensor iluminancia*, capturan el valor de iluminancia en el interior del contenedor. Posteriormente se generan tramas de datos en la que se envía, la dirección MAC del nodo y el valor de iluminancia detectado.

Las tramas generadas son enviadas hacia el nodo gateway inalámbricamente vía radio.

El nodo gateway, recibe las tramas provenientes de los nodos que se encuentran dentro del radio de cobertura, y posteriormente las envía a la estación base. La comunicación entre el nodo gateway y la estación base se realiza mediante un puerto serie. Para esta arquitectura, se ha utilizado un puerto USB.

4.1.2. Capa de procesamiento

La capa de procesamiento, se compone de una notebook que actúa como estación base.

En esta capa se encuentran los siguientes subsistemas: El Subsistema de captura y almacenamiento de datos de la estación base y el Subsistema de cálculo de ruta óptima.

El subsistema de captura y almacenamiento de datos de la estación base, analiza la trama recibida para determinar si es válida. Si esto se cumple, se examina la sección de datos (payload).

Si la trama proviene de un nodo sensor pir, se extrae de la trama la dirección MAC y el contador actual. Luego se actualiza el contador asociado al contenedor y se almacena la trama leída en la base de datos.

Si la trama proviene de un nodo sensor iluminancia, se extrae de la trama la dirección MAC y el valor de iluminancia recibida. Posteriormente se actualiza en la base de datos, el valor de iluminancia del contenedor asociado.

En ésta capa también se realiza el cálculo de ruta óptima, seleccionando de la base de datos, aquellos contenedores cuyo contador y valor de iluminancia superan un valor predeterminado. Posteriormente se ejecuta el algoritmo A* (A estrella) para determinar el orden en que se recogerán los contenedores.

4.1.3. Capa de visualización

La última capa de la arquitectura es la de visualización, la cual utiliza una interfaz web para brindar información a los usuarios, sobre la ubicación de los contenedores a recoger.

En esta capa se encuentra el Subsistema de visualización de ruta.

4.2. Componentes de hardware utilizados

Los nodos sensores utilizados en la capa de infraestructura son de la marca iSense, y están fabricados por la empresa Coalesenses[11].

Se decidió utilizar el módulo de seguridad y el módulo ambiental, al considerar que son los que mejor se adaptan para resolver la problemática planteada. El módulo de seguridad dispone de un sensor pasivo infrarrojo (PIR) y un acelerómetro, mientras que el módulo ambiental dispone de un sensor de luz y un sensor de temperatura.

El hardware de la plataforma iSense presenta un esquema modular.

El "nodo sensor pir", está formado por el módulo principal (core module), el módulo de seguridad, y el módulo de energía (Figura 2).

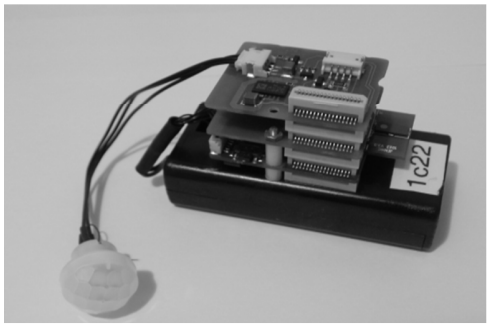


Figura 2: Nodo sensor pir

El "nodo sensor iluminancia", está formado por el módulo principal (core module), el módulo ambiental y el módulo de energía.

4.3. Herramientas de software utilizadas

El hardware iSense utilizado en la capa de infraestructura, proporciona un conjunto de herramientas de software que incluyen: Sistema operativo, librerías y códigos fuentes desarrollados en lenguaje C++.

Este conjunto de herramientas provee un variado número de servicios y protocolos listos para usar, lo cual permite el desarrollo rápido de aplicaciones pequeñas pero complejas.

Entre las herramientas de software utilizadas se encuentran: herramientas de compilación (make, cmake, g++), compilador ba-elf (para micro controladores Jennic), plataforma de desarrollo Eclipse, firmware iSense (librería para desarrollo iSense), iShell (herramienta para análisis, programación y operación sobre nodos).

5. Pruebas Realizadas y Resultados Obtenidos

En esta sección se describen las pruebas realizadas en laboratorio y en campo.

5.1. Prueba en Laboratorio

Para evaluar el nivel de llenado, se llevó a cabo una prueba de laboratorio, en la que se utilizó una caja de cartón cuyas dimensiones fueron: paredes laterales de 44 cm de ancho, paredes frontal y trasera de 47 cm de ancho y profundidad o alto de 35 cm. En la Figura 3 se puede ver una imagen de la caja y componentes utilizados, y en la figura 4 una imagen del interior de la caja con el nodo dentro.

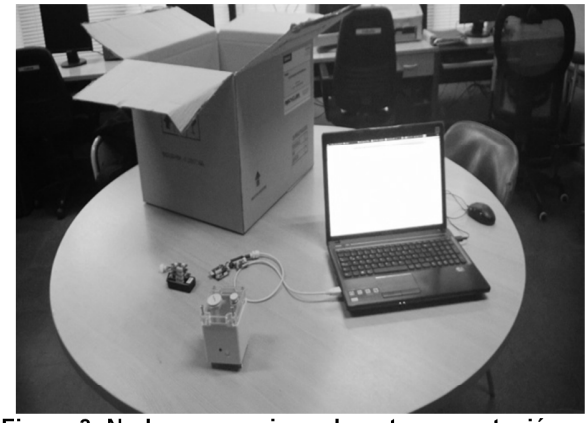


Figura 3: Nodo sensor pir, nodo gateway, estación base y caja para pruebas

Antes de iniciar las pruebas, se ubicó el nodo en la parte superior de una de las paredes laterales de la caja y con el sensor pir apuntando hacia el fondo, a 45º de la pared lateral, para que el área de detección cubra la mayor superficie posible.



Figura 4: Ubicación del nodo sensor pir en la caja

Una vez ubicado el nodo sensor, se encendió el mismo y se espero un minuto para que el sensor se inicialice y reconozca la temperatura ambiente. Luego se inició la aplicación de "captura y almacenamiento" en la estación base y se comenzaron a introducir objetos en la caja para verificar si el sensor los detectaba.

Por cada trama recibida, se verificó que la misma sea válida. Esta validación se llevó a cabo, evaluando la longitud de la trama recibida y verificando si los

caracteres de inicio y fin de trama coincidian con los definidos en la capa de infraestructura. Se considera válida una trama si inicia con los caracteres 1002 y si termina con los caracteres 1003 y si la longitud es de 16 caracteres.

Finalizadas la pruebas se obtuvieron los siguientes resultados:

- Se recibieron algunas tramas incompletas o defectuosas que no pasaron la validación.
- No se detectaron todos los objetos que pasaron delante del sensor. Esto pudo ocurrir por dos motivos:
    - El objeto pasó delante del sensor fuera del lapso de monitoreo.
    - El objeto que se introdujo no irradió el calor suficiente para que el sensor lo detecte.
- Se comprobó que si el sensor se encuentra obstruido por algún objeto, no detecta ningún movimiento.

5.2. Prueba en Campo

Las pruebas de campo se realizaron en el barrio Villa Sarita de la ciudad de Posadas. Se eligió este barrio por estar cercano al centro y por utilizar los mismos contenedores de residuos del centro de la ciudad.

Se consideró una ruta constituida por 23 contenedores, de los cuales se seleccionaron cuatro contenedores que se encontraban próximos entre sí, tal como se indica en la figura 5.

Para cada uno de los contenedores de la ruta, se ingresó en el software que se ejecuta en la capa de procesamiento, las direcciones de latitud y longitud.



Figura 5: Ubicación en el mapa de los contenedores

Los contenedores utilizados en las pruebas (figura 6), presentan las siguientes características:

Están fabricados de polietileno de alta densidad de formulación especial, con una densidad de 0,95 gr./cm3.

- Altura total: 129,5 cm.
- Ancho: 126,5 cm.
- Volumen útil: 1.080 litros
- Fondo: 107 cm.
- Peso: 56 kg.
- Carga máxima: 500 kg.



Figura 6: Contenedor de residuos utilizado en las pruebas de campo

Los elementos utilizados en estas pruebas fueron: un nodo sensor pir, un nodo gateway y una notebook que funcionó como estación base. Las pruebas se realizaron de un contenedor por vez. Antes de iniciar las mismas se cumplieron los siguientes pasos:

- Se verificó que el contenedor esté vacío.
- Se ubicó el nodo sensor pir en uno de los extremos del contenedor, con el sensor apuntando hacia abajo, para que el mismo no detecte movimientos que ocurrían fuera del contenedor.
- En el software del nodo sensor pir, se estableció un intervalo de silencio de 2 segundos y un temporizador para informar el valor del contador cada 20 segundos.

Finalizado los pasos anteriores, se encendió el nodo sensor pir y se esperó un minuto para que el mismo reconozca la temperatura del interior del contenedor. Durante este tiempo, no se realizó ningún movimiento

delante del sensor. Luego se ejecutó la aplicación de captura y almacenamiento en la estación base.

Posteriormente, se comenzaron a introducir bolsas de residuos en distintos extremos del contenedor (figura 7), para verificar si el sensor los detectaba, controlando cada 20 segundos la interfaz a fin de constatar que el contador se incremente.

Después de introducir varias bolsas de se constató lo siguiente:

- Se recibieron algunas tramas incompletas o defectuosas que no pasaron la validación.
- El contador asociado al contenedor tenía un valor superior a la cantidad de objetos introducidos. Esto se produjo porque el nodo sensor pir captó movimientos ajenos al interior del contenedor.



Figura 7: Inserción de residuos en el contenedor

Esta secuencia de pruebas se repitió en los tres contenedores restantes. Para simular distintos nodos, se asignó al nodo sensor pir, diferentes direcciones MAC (d1, d2, d3, d4).

En la figura 8 se puede visualizar la pantalla de captura de tramas al finalizar las pruebas sobre los cuatro contenedores.

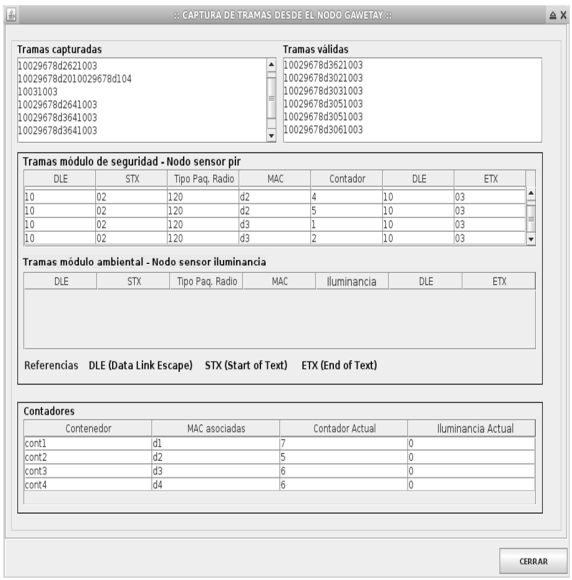


Figura 8 - Captura de tramas

5.3. Resultados de la prueba

Al concluir las pruebas de detección de objetos, se constató que el gateway recibía tramas de datos (válidas e inválidas), únicamente cuando tenía una visión directa con el nodo sensor pir a unos 70 metros de distancia aproximadamente. Por este motivo el nodo gateway se ubicó en la intersección de las calles Nicomedes Castro y Moritán como se puede ver en la figura 9.

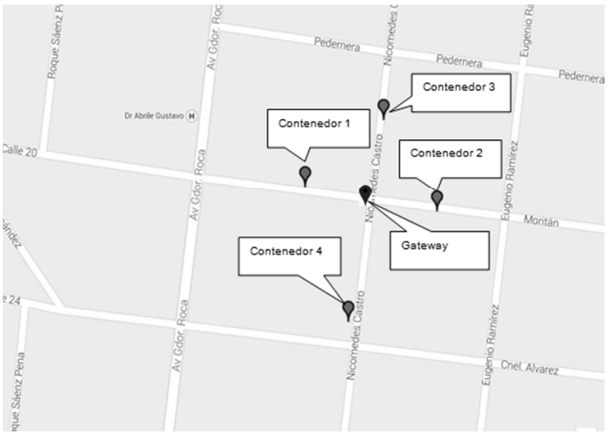


Figura 9: Ubicación en el mapa del gateway

Por último, se ejecutó el módulo de cálculo de ruta, se verificó el resultado del algoritmo A * en base a las

distancias indicadas en la grilla de contenedores a recorrer y se visualizó la ruta óptima en la interfaz web.

En la figura 10 se visualiza el mapa con la ruta completa y en la figura 11 se puede ver la porción del mapa (con un nivel de acercamiento superior), donde se indican los contenedores a recorrer. El punto A corresponde al origen del recorrido (Dirección de Servicios Públicos). Los puntos B, C, D y E corresponden a los contenedores 1, 2, 3 y 4. El punto F corresponde al final del recorrido (Estación de tratamiento de residuos).

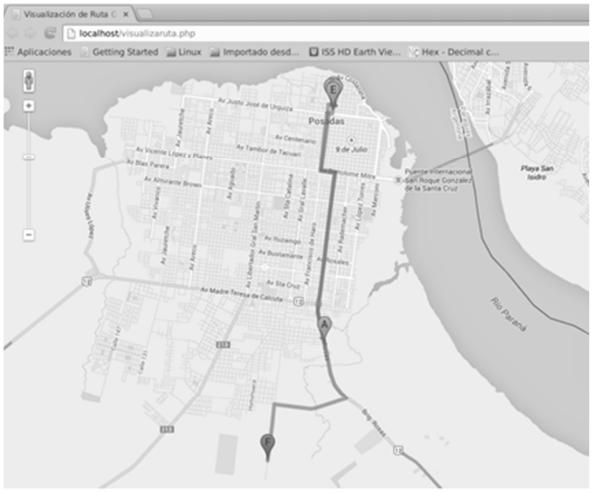


Figura 10: Visualizacion de ruta óptima.

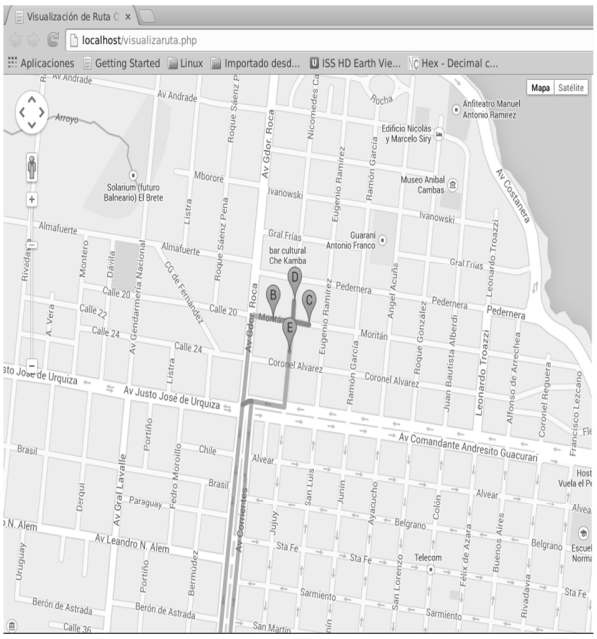


Figura 11 - Contenedores a recoger

6. Conclusión

Con las pruebas realizadas, se logró probar la comunicación entre un nodo sensor y un nodo gateway, utilizando el estándar IEEE 802.15.4.

Además, se concluyó que si bien el nodo sensor pir permite detectar los objetos que ingresan al contenedor, no sirve para determinar un nivel de llenado por sus características de fábrica.

Por último, se observó que al trabajar en un entorno inalámbrico, se pierden muchas tramas debido a interferencias y ruidos presentes en el ambiente. Por esta razón, es fundamental que un sistema que utilice una red de sensores inalámbricos, valide las tramas recibidas, a fin de trabajar con datos correctos.

7. Trabajos Futuros

Como trabajos futuros se adicionará otro tipo de sensores a fin de mejorar la estimación de esta medida, necesaria para llevar a cabo el trabajo. Se prevee también realizar las interfaces web necesarias para registrar contenedores y utilizar algoritmos de búsqueda para mejorar la estimación del camino óptimo de recolección de contenedores.

8. Bibliografia

[1] Ariel, *Smart Cities: Un primer paso hacia la Internet de las cosas*. Barcelona: Fundación Telefónica, 2011.

[2] Indec. (2010) Censo 2010. [Online]. http://www.sig.indec.gov.ar/censo2010/

[3] Gavin A Schmidt, "Attribution of the present day total greenhouse effect," *Journal of Geophysical Research: Atmospheres*, 2012.

[4] The Computer for the 21st Century. [Online]. http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf

[5] Fundación de la Innovación Bankinter, *El internet de las cosas. En un mundo conectado de objetos inteligentes*., 2011.

[6] Ley de Moore. [Online]. http://es.wikipedia.org/wiki/Ley_de_Moore

[7] Buenos Aires Ciudad. [Online]. http://www.buenosaires.gob.ar/modernizaci%C3%B3n/ejes-de-trabajo-y-objetivos/ciudad-inteligente

[8] Misiones Cuatro. [Online]. http://misionescuatro.com/ampliar.php?id=15455

[9] Valladares Tomas y LOPEZ David,., 2011, ch. Capitulo II.

[10] Red Española de Ciudades Inteligentes. [Online]. http://www.redciudadesinteligentes.es/

[11] Coalesenses GmbH. Bridging the gap between virtuality and reality. [Online]. http://www.coalesenses.com

Análisis del Impacto del Tráfico, de las Configuraciones de Red y de Seguridad en Redes Wi-Fi

H. Facchini, S. Pérez, L. Bisaro,
F. Hidalgo, G. Cangemi
GridTICs- Dpto de Electrónica-UTN Mendoza
Rodríguez 273, Ciudad Mendoza
CP (M5502AJE) República Argentina
{higiniofac, santiagocp}@frm.utn.edu.ar,
{lbisaro, fabianhdlg, gncelectronic}@gmail.com

Nora Costa
Facultad de Ingeniería
Universidad de Mendoza
Paseo Descotte 750, Ciudad Mendoza
CP (M5502AJE) República Argentina
ncosta@um.edu.ar

Abstract

The actual throughput of a WLAN 802.11 reaches, by the inherent limitations of the shared medium used, a rate significantly lower than the nominal value of speed, in correspondence with the theoretical values of the access technical of CSMA/CA. In addition, the total traffic presents a significant overhead on the actual data traffic. This study evaluates the performance in detail and real cash of Wi-Fi networks, in a controlled laboratory environment. It is proposed a new methodology of evaluation of the performance, based on the actual transfer of a file, using a passive and independent station to capture the traffic, which approximates the study to the usual conditions of practical behavior. We compared the results obtained in the various cases, verifying the differential impact they have on the performance the type of traffic, the topology of the network, and security implemented, in that order.

Resumen

El rendimiento real de una red WLAN 802.11 alcanza, por las limitaciones inherentes del medio compartido que se utiliza, una tasa significativamente más baja que el valor nominal de velocidad, en correspondencia con los valores teóricos de la técnica de acceso CSMA/CA. Además, el tráfico total presenta una sobrecarga significativa respecto al tráfico real de datos. Este estudio evalúa el desempeño efectivo real en detalle de las redes Wi-Fi, en un entorno de laboratorio controlado. Se propone una nueva metodología de la evaluación de la actuación, basado en la transferencia real de un archivo, usando una estación pasiva e independiente para capturar el tráfico, que aproxima el estudio a las condiciones normales de comportamiento práctico. Se han comparado los resultados obtenidos en los diversos casos, verificando las distintas repercusiones que tienen sobre el desempeño del tipo de tráfico, la topología de la red y la seguridad implementada, en ese orden.

1. Introducción

Debido al significativo despliegue que están tomando las aplicaciones Wi-Fi, resulta cada vez más importante analizar el comportamiento de estas redes y el impacto de las topologías de red, los tipos de tráficos y los esquemas de seguridad aplicados sobre ellas.

En el tráfico total de estas redes hay una importante sobrecarga sobre el tráfico de datos real propiamente dicho, respecto a las redes cableadas. Además, la ubicación del nodo origen y del nodo destino de la transferencia, en una red mixta de redes cableadas e inalámbricas, implica escenarios sustancialmente diferentes en términos de prestaciones.

A partir de diversos trabajos, se sabe que, en general, las comunicaciones 802.11 tienen un rendimiento bruto real de a lo sumo el 45-55 % que dista mucho de la tasa nominal de velocidad definida en la norma aplicada. Esta prestación es muy inferior al rendimiento en las comunicaciones cableadas. Sin embargo, no hay muchos estudios que respalden dicho enunciado sobre las redes Wi-Fi, combinando en un mismo trabajo el análisis del impacto en el rendimiento de acuerdo a diversas configuraciones de topologías de red, a los tipos de tráfico de los enlaces wireless y los contextos de seguridad.

El presente trabajo se desarrolla con los siguientes apartados: II. Security Wi-Fi, III. Trabajos de Investigación, IV. Propuesta y Adopción de Tres Escenarios de Experimentación, V. Resultados Experimentales, VI. Principales Conclusiones, y VII. Aportes y Trabajos Futuros.

2. Seguridad Wi-Fi

Es claro el conocimiento sobre la creciente demanda e implantación de todo tipo de redes wireless LAN (WLAN) en entornos corporativos, de pequeñas empresas, y en el ámbito familiar. La tecnología Wi-Fi (Wireless Fidelity) es una de las tecnologías líderes en la

comunicación inalámbrica, y el soporte para Wi-Fi se está incorporando en cada vez más aparatos: smartphones, tablets, y otros dispositivos móviles, ya que este tipo de redes ofrece un gran abanico de ventajas frente a las tradicionales redes cableadas. Facilita la instalación, tiene amplia cobertura y movilidad, es sencilla su ampliación, etc. Precisamente, es gracias a estas características que las redes inalámbricas deben el gran apogeo que viven en este momento

Sin embargo, la seguridad ha sido y es un aspecto que cobra especial relevancia cuando hablamos de redes inalámbricas. Para tener acceso a una red cableada es imprescindible una conexión física al cable de la red. Sin embargo, en una red inalámbrica desplegada en una oficina, un tercero podría acceder a la red sin ni siquiera estar ubicado en las dependencias de la empresa. Bastaría con que estuviese en un lugar próximo donde le llegase la señal. Es más, en el caso de un ataque pasivo, donde sólo se escucha la información, ni siquiera se dejan huellas que posibiliten una identificación posterior.

El canal de las redes inalámbricas, al contrario que en las redes cableadas privadas, debe considerarse inseguro. Cualquiera podría estar escuchando la información transmitida. Y no sólo eso, sino que también se pueden inyectar nuevos paquetes o modificar los ya existentes (ataques activos). Las mismas precauciones que tenemos para enviar datos a través de Internet deben tenerse también para las redes inalámbricas.

Conscientes de este problema, el IEEE publicó un mecanismo opcional de seguridad, denominado WEP [1-2], en la norma de redes inalámbricas 802.11. Pero WEP, que inicialmente se desplegó en numerosas redes WLAN, tiene vulnerabilidades que lo convirtieron en una protección limitada [3-4].

Para solucionar sus deficiencias, el IEEE comenzó el desarrollo de una nueva norma de seguridad, conocida como 802.11i [2, 3, 5], que permitiera dotar de suficiente seguridad a las redes WLAN.

Algunas empresas en vistas de que WEP (de 1999) era insuficiente y que no existían alternativas estandarizadas mejores, decidieron utilizar otro tipo de tecnologías como son las VPNs (Virtual Private Networks) para asegurar los extremos de la comunicación (por ejemplo, mediante IPSec), o recurrir a soluciones propietarias. La idea de proteger los datos de usuarios remotos conectados desde Internet a la red corporativa se extendió, en algunos entornos, a las redes WLAN. De hecho, como hemos comentado antes, ambos canales de transmisión deben considerarse inseguros. Pero la tecnología VPN es quizás demasiado costosa en recursos para su implementación en redes WLAN.

No ajena a las necesidades de los usuarios, la asociación de empresas Wi-Fi decidió lanzar un mecanismo de seguridad intermedio de transición hasta que estuviese disponible 802.11i, tomando aquellos aspectos que estaban suficientemente avanzados del desarrollo de la norma. En 2003, Wi-Fi Alliance propuso WPA [6-7], que incluía un subconjunto de las características que agregaba 802.11i. WPA fue una solución interina de segunda generación diseñada para corregir las vulnerabilidades WEP por anticipado de la ratificación de IEEE 802.11i.

Concurrentemente con la ratificación del agregado IEEE 802.11i en 2004, Wi-Fi Alliance introdujo WPA2 [8]. Inicialmente fue una certificación opcional, pero en 2006 se volvió un requerimiento obligatorio para todo nuevo equipamiento. WPA2 se construyó sobre las características de WPA, e introdujo un encriptación más fuerte.

Las Tablas I y II resumen la evolución histórica de los mecanismos de seguridad en IEEE 802.11.

El éxito del mercado de redes wireless 802.11 depende de la satisfacción de usuario y la interoperabilidad. El número rápidamente creciente de dispositivos Wi-Fi hogareños, de empresas y móviles quiere más que interoperabilidad, confiabilidad, facilidad de uso, y aparatos y APs (Access Points) de altas prestaciones. Estos dispositivos 802.11 deben proveer también los más altos niveles de seguridad disponibles usando mecanismos basados en estándares.

3. Trabajos de Investigación

La introducción de la seguridad en las WLANs ha significado siempre un desafío y el potencial riesgo de una degradación en las prestaciones, dependiendo en parte de las opciones elegidas para la encriptación, autenticación, certificados digitales, entre otros aspectos.

Tabla 1. Evolución histórica de los mecanismos de seguridad 802.11

Fecha	Hito
Setiembre de 1997	Se ratifica el estándar IEEE 802.11, incluyendo WEP
Abril de 2000	Se lanza el programa Wi-Fi CERTIFIED con soporte para WEP
Mayo de 2001	Se crea el Grupo de Tareas IEEE 802.11i
Abril de 2003	Se introduce WPA con: Autenticación IEEE 802.1X, Cifrado TKIP (Temporal Key Integrity Protocolo) y Soporte de EAP-TLS (EAP-Trasnport Layer Security
Setiembre de 2003	Es obligatorio WPA para todo equipamiento Wi-Fi CERTIFIED
Junio de 2004	Se ratifican agregados al estándar IEEE 802.11
Setiembre de 2004	Se introduce WPA2 con: Autenticación IEEE 802.1X, Cifrado AES y Soporte para EAP-TLS
Abril de 2005	Soporte para 4 nuevos tipos EAP adicionados:EAP-TTLS/MSCHAPv2 (EAP-Tunneled TLS Microsoft Challenge Handshake Authentication Protocol Version 2); PEAPv0/EAP-MSCHAPv2 (Protected EAP Version 0); PEAPv1/EAP Generic Token Card (EAP-GTC); y EAP-SIM (EAP-Suscriber Identity Module
Marzo de 2006	Es obligatorio WPA2 para todo equipamiento WiFi CERTIFIED
Enero de 2007	Se lanza elprograma Wi-Fi Protected Setup
Noviembre de 2007	Se crea el Grupo de Tareas IEEE 802.11w
Mayo de 2009	Soporte para EAP-AKA y EAP-FAST
Enero de 2012	Soporte para Protected Managment Frames con WPA2

Tabla 2. Aspectos técnicos de los mecanismos de seguridad 802.11.

	Métodos 802.11i		
	WEP	WPA	WPA2
Protocolo de seguridad	WEP	TKIP	CCMP
Cifrado	Asignación Manual de Claves. Claves compartidas usando el algoritmo RC4 (Rivest Cipher 4). El cifrado es por stream.	TKIP basado en RC4	Modo Contador con CCMP(Cipher Block Chaining Message Authentication Code Protocol) con 128 bits. Se usa el algoritmo AES que es de cifrado por bloque
Longitud de Clave	40 o 104 bits	128 bits para la encriptación, 64 bits para la autenticación	128 bits
Vida de la Clave	24 bit IV	48 bit IV	
Generación de Clave	Concatenación	Función de mezcla de dos fases	No necesitada
Integridad de los Datos	CRC-32	Michael	CBC-MAC
Integridad del Encabezado	Ninguna	Michael	CBC-MAC
Protección de Reproducción	Ninguna	Número de Paquete	
Gestión de la Clave	Ninguna	Basada en EAP	
Autenticación	Clave Abierta o Compartida	802.x o Pre-Shared Key (PSK)	
Integridad de los Datos	Función Hash Lineal	Función Hash Criptográfica	
Detección de Reproducción	No	SI	

La finalización del estándar IEEE 802.11i (WPA2) se convirtió en motivo de diversos trabajos de investigación, ensayos experimentales y estudios basados en la comparación del comportamiento de configuraciones de red: sin seguridad, o con seguridad WEP o WPA. En el tiempo se han sucedido una gran cantidad de trabajos de investigación, entre los cuales destacamos los referidos en [9-25]. Como una secuencia histórica, presentan un proceso continuo y dinámico; además, muestran en general, resultados diversos. En algunos casos, se observan diferencias en las prestaciones según las opciones de seguridad; aunque estas también pueden ser importantes, dependiendo de las configuraciones de red, las implementaciones y del contexto de trabajo.

En [13] se investigaron los efectos de los frameworks de seguridad de las VPN (Virtual Private Network) y IEEE 802.1x sobre la performance de la red. El patrón general encontrado fue que a mayor nivel de seguridad más baja las prestaciones de la red. Mientras que las VPN ofrecen seguridad extremo a extremo (comparado con 802.1x, el cual provee solamente seguridad del cliente al AP), son más complejas de implementar y tienen un impacto mayor sobre el rendimiento que 802.1x.

En [14] se extendió esta investigación evaluando los efectos de la longitud de paquete sobre las prestaciones de la red con diferentes arquitecturas de seguridad. Este estudio mostró que la encriptación WEP reduce significativamente el rendimiento de la red, cuando la red está congestionada.

Mientras que en [15] se introdujo el problema del ancho de banda en las redes wireless. Se analizó las prestaciones de redes 802.11g, considerando qué puntos son responsables para la decisión de una clase específica de seguridad, ponderando factores según el tipo de red. La evaluación se realizó usando los métodos WEP 64, WEP 128 y WPA. Se determinó que WPA alcanzaba mejores prestaciones que su predecesor.

Los autores de [16] desarrollaron un ambiente de prueba para conducir automáticamente los test de interoperabilidad de sistemas, cuando se introdujeron los nuevos estándares WPA y WPA2/IEEE802.11i, cuyo objeto fue plantear sugerencias a partir de las regulaciones de la Wi-Fi Alliance. Identificaron algunas categorías de test y derivaron requerimientos comunes para una solución de software automatizada. Como en muchos trabajos, en [17-18] se muestran que no sólo debía resolverse la interoperabilidad de los sistemas, sino además que se iniciaba el proceso de compilación y comparación de ventajas y desventajas entre los métodos de seguridad provistos por WEP, WPA y RSN.

En [19] se presenta otro trabajo que actualiza la problemática que plantea la evaluación de los efectos de los estándares de seguridad 802.11i sobre la performance de las redes wireless. Se obtuvo midiendo las prestaciones y los retardos, mientras el tráfico pasaba por una red de prueba. El tráfico utilizado consistió alternativamente de tráfico producido por un generador sintético, y como así también, obtenido por aplicaciones típicas de transferencia de archivos. Es interesante destacar que en el trabajo se establecieron y evaluaron diversos niveles de seguridad, a saber: sin seguridad, seguridad WEP con encriptación de 40 y 104 bits, WPA con autenticación PSK y encriptación RC4, WPA con autenticación EAP-TLS y encriptación RC4, WPA2 con autenticación PSK y encriptación AES, y WPA2 con autenticación EAP-TLS y encriptación AES. Haciendo las aclaraciones del contexto de sus ensayos experimentales, se concluyó que todos los niveles de seguridad, tienen rendimiento, latencia y tasas de error similares para todas las transferencias de tráfico.

4. Propuesta y Adopción de Tres Escenarios de Experimentación

Para estas pruebas se propusieron y adoptaron 3 escenarios de experimentación con distintas topologías físicas de trabajo. La selección de estos escenarios de mínima complejidad se basó en la búsqueda de los límites superiores de los rendimientos posibles a esperar para cada uno de los casos. El incremento del número de estaciones implica, necesariamente, rendimientos por debajo de los límites superiores encontrados.

Los rendimientos se midieron desde el inicio a la finalización de la transferencia de un archivo de texto. Se

excluyó en el estudio, el impacto de cualquier variante de autenticación, por su baja influencia en los rendimientos brutos y efectivos, especialmente durante la transferencia.

4.1. Escenario de Experimentación 1

La topología de experimentación del Escenario 1 es una red tipo Infraestructura, con un Access Point, una estación (computadora) conectada mediante una placa inalámbrica, y la otra estación (computadora) mediante una conexión cableada tipo FastEthernet de 100Mbps sobre cable UTP categoría 5e, de acuerdo a la Figura 1.

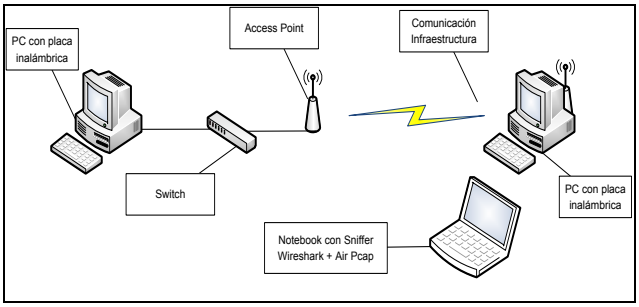


Figura 1. Escenario de experimentación 1.

4.2. Escenario de Experimentación 2

La topología de experimentación del Escenario 2 es una red tipo Infraestructura, con un Access Point, y las dos estaciones (computadoras) conectadas mediante una placa inalámbrica cada una, de acuerdo a la Figura 2.

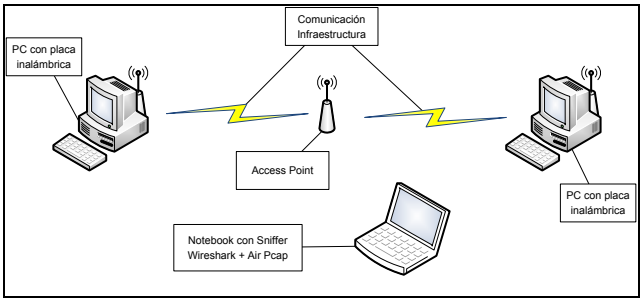


Figura 2. Escenario de experimentación 2.

4.3. Escenario de Experimentación 3

La topología de experimentación del Escenario 3 es una red Ad-Hoc, es decir, 2 estaciones (computadoras) conectadas directamente sin Access Point, de acuerdo a la Fig. 3.

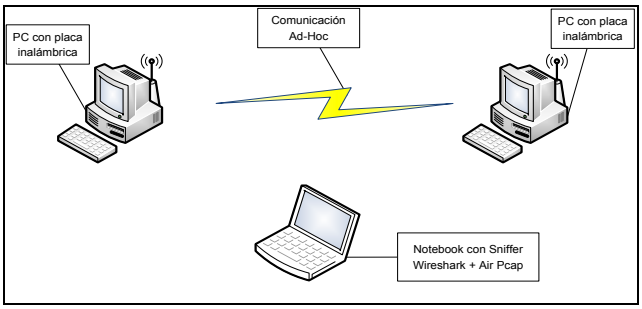


Figura 3. Escenario de experimentación 3.

4.4. Características Comunes de los Escenarios de Experimentación

En este apartado se describen los equipos, el software asociado y las configuraciones comunes a todos los Escenarios de Experimentación. Se utilizaron los siguientes dispositivos:

- Access Point: Cisco WAP4410N Wireless-N, y
- Estaciones: Computadoras de escritorio con Procesador AMD ATHLON X2, con memoria RAM de 2 GB DDR3 de 1166 MHz, Sistema Operativo Windows 7 Pro, placa de red Ethernet con conexión a 100Mbps y Placa de red inalámbrica Cisco Linksys WUSB600N.

En las computadoras fuente y destino de la transmisión de datos, se habilitaron y utilizaron dos tipos de herramientas de tráfico de archivos:

- Servidor FTP Filezilla [26], para generar transferencias de datos reales bajo el protocolo TCP,
- Servidor TFTP Tftpd32 [27], para generar trasferencias de datos reales bajo el protocolo UDP.

Y una estación independiente donde se corrió el software sniffer Wireshark [28] con la placa AirPcap [29] (Figura 4) para capturar el tráfico en el aire, y posteriormente obtener las lecturas de las variables a analizar.



Figura 4. Kit de AirPcap para captura de tráfico Wi-Fi.

Además, se realizaron distintas subconfiguraciones en cada escenario de experimentación, variando el tamaño del archivo de texto (objeto de la transferencia entre las estaciones), la norma 802.11 y los criterios de seguridad en la comunicación inalámbrica, a saber:

- Sin seguridad, es decir, sin cifrado de datos, para la norma 802.11n (a 144Mbps) y 802.11g (a 54 Mbps), y tamaño de archivo de texto de 2, 10 y 100 MB,
- Con seguridad tipo WEP, con cifrado WEP, de claves de 40 bits tipo hexadecimal, sólo para la norma 802.11g (a 54 Mbps), y tamaño de archivo de texto de 2, 10 y 100 MB. En la norma 802.11n no se soporta WEP.
- Con seguridad tipo WPA2, con cifrado AES, con claves de 10 dígitos ASCII de longitud, para la norma 802.11n (a 144Mbps) y 802.11g (a 54 Mbps), con tamaño de archivo de texto de 2, 10 y 100 MB.

De los archivos de la captura con Wireshark se obtuvieron las siguientes métricas:

- Tiempo de Transmisión [seg],
- Rendimiento, velocidad o throughput promedio efectivo de datos [Mbps],
- Rendimiento promedio bruto o real [Mbps] en ambos sentidos y sólo en el sentido de la transferencia,
- Coeficiente o Índice de rendimiento efectivo (cociente entre el rendimiento efectivo y el rendimiento bruto que indica la proporción del rendimiento bruto que efectivamente se utiliza para la transferencia de los datos),
- Cantidad total de paquetes en ambos sentidos y sólo en el sentido de la transferencia,
- Tamaño de paquete promedio [bytes] en ambos sentidos y sólo en el sentido de la transferencia,
- Tamaño de paquete máximo en ambos sentidos [bytes],
- Tamaño de paquete mínimo en ambos sentidos [bytes], y
- Tiempo intertrama promedio, máximo y mínimo.

5. Resultados Experimentales

Se ha verificado que el cambio de Escenario, del tipo de tráfico o de la seguridad configurada, tiene un impacto significativo sobre el rendimiento Wi-Fi. Para cada combinación de contexto, se han experimentado distintos valores para las métricas analizadas. Por cuestiones de espacio, se presentan en las Tablas 3,4 y 5 las métricas principales para el caso de la transferencia del archivo de 10 Mbytes, en 802.11n a 144 Mbps, para FTP (TCP) y FTP (UDP), sin seguridad y con seguridadWPA2.

Tabla 3. Tabla comparativa en el escenario 1.

	Escenario 1 TCP		Escenario 1 UDP	
Métricas / Seguridad	Sin seguridad	Seguridad WPA2	Sin seguridad	Seguridad WPA2
Tiempo de Transmisión [seg]	2,1	2,5	4,3	3,7
Cantidad de Tramas capturadas en el canal	16284	17117	16974	14310
Cantidad de Tramas en ambos sentidos	16198	17040	16649	13881
Cantidad de tramas en el sentido de la transferencia	9931	10713	9977	9213
Tamaño de paquete promedio sentido transf. [bytes]	1295,7	1314,5	1290,1	1282,1
Tamaño de paquete promedio ambos sentidos [bytes]	822,5	857,1	782,4	873,9
Rendimiento bruto sentido transferencia [Mbps]	48,3	45,4	24,3	25,8
Rendimiento efectivo de datos [Mbps]	39,4	33,8	19,4	22,3
Índice de Rendimiento efectivo	0,81	0,74	0,80	0,86
Tiempo intertrama Promedio Global	0,00026	0,00029	0,00051	0,00052

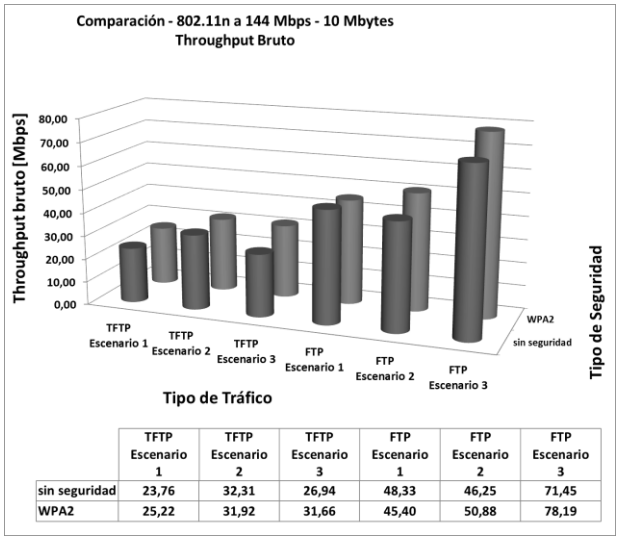
Tabla 4. Tabla comparativa en el escenario 2.

	Escenario 2 TCP		Escenario 2 UDP	
Métricas / Seguridad	Sin seguridad	Seguridad WPA2	Sin seguridad	Seguridad WPA2
Tiempo de Transmisión [seg]	3,4	3,9	6,1	5,9
Cantidad de Tramas capturadas en el canal	24953	30188	32320	30954
Cantidad de Tramas en ambos sentidos	24860	30094	32101	30754
Cantidad de tramas en el sentido de la transferencia	13989	17337	18170	17255
Tamaño de paquete promedio sentido transf. [bytes]	1398,7	1422,5	1364,7	1369,6
Tamaño de paquete promedio ambos sentidos [bytes]	819,0	853,3	793,2	790,6
Rendimiento bruto sentido transferencia [Mbps]	46,2	50,9	33,2	32,9
Rendimiento efectivo de datos [Mbps]	24,8	21,6	13,7	14,2
Índice de Rendimiento efectivo	0,54	0,43	0,41	0,43
Tiempo intertrama Promedio Global	0,00027	0,00026	0,00038	0,00038

Tabla 5. Tabla comparativa en el escenario 3.

	Escenario 3 TCP		Escenario 3 UDP	
Métricas / Seguridad	Sin seguridad	Seguridad WPA2	Sin seguridad	Seguridad WPA2
Tiempo de Transmisión [seg]	1,3	1,2	3,2	2,8
Cantidad de Tramas capturadas en el canal	15838	14748	17859	18278
Cantidad de Tramas en ambos sentidos	15458	13701	17761	18172
Cantidad de tramas en el sentido de la transferencia	9981	10254	11482	11330
Tamaño de paquete promedio sentido transf. [bytes]	1134,4	1188,1	950,6	960,6
Tamaño de paquete promedio ambos sentidos [bytes]	745,6	852,8	629,8	618,3
Rendimiento bruto sentido transferencia [Mbps]	71,5	78,2	27,7	32,7
Rendimiento efectivo de datos [Mbps]	66,2	67,3	25,9	30,5
Índice de Rendimiento efectivo	0,93	0,86	0,93	0,93
Tiempo intertrama Promedio Global	0,00016	0,00017	0,00036	0,00030

En la Figura 5 se observa una comparación del rendimiento bruto y en la Figura 6 del rendimiento efectivo, para los tres Escenarios.



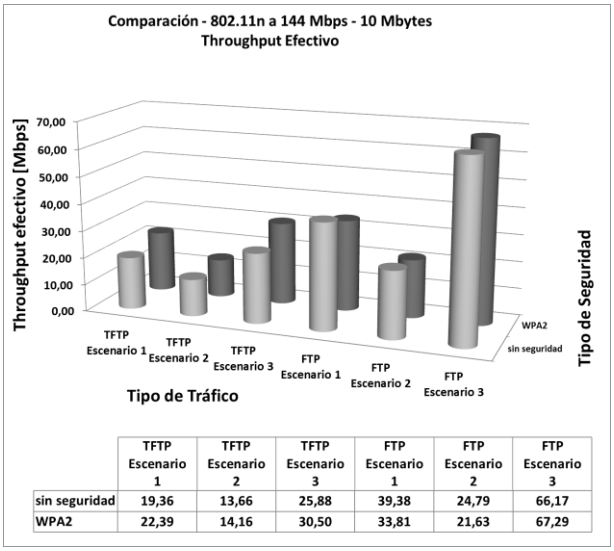
	TFTP Escenario 1	TFTP Escenario 2	TFTP Escenario 3	FTP Escenario 1	FTP Escenario 2	FTP Escenario 3
sin seguridad	23,76	32,31	26,94	48,33	46,25	71,45
WPA2	25,22	31,92	31,66	45,40	50,88	78,19

Figura 5. Comparación de rendimiento bruto 802.11n 144 Mbps.

5.1. Rendimiento Bruto

De la Figura 5 se desprende que:

- Las velocidades brutas son muy bajas con respecto al nominal de 144 Mbps.
- El mejor de los casos se da para FTP con WPA2 en el Escenario 3, con el 54% de la velocidad nominal (78,19 Mbps). Y el peor de los casos se da para TFTP sin seguridad en el Escenario 1, con el 16% de la velocidad nominal (23,76 Mbps). Es decir, una disparidad entre los valores extremos del 38%.
- El cambio de Escenario influye en el rendimiento bruto de TFTP y FTP. Entre el Escenario 1 y el Escenario 2 de TFTP se alcanza la diferencia máxima del 75%.
- Con FTP, en promedio, se obtiene casi el doble de rendimiento bruto que en TFTP.
  1. Entre los Escenarios 1 hay una diferencia del 84%,
  2. Entre los Escenarios 2 del 50%, y
  3. Entre los Escenarios 3 del 160%
- Las dependencias con respecto a la seguridad implementada son menores que las dependencias respecto al tipo de Escenario o al tipo de tráfico TFTP o FTP. No se observa un comportamiento definido. Ateniéndonos a los resultados de las mediciones se observa que la dependencia mayor del tipo de seguridad implementada se da en el Escenario 3 de TFTP. Hay una diferencia del 17,5% en el rendimiento bruto cuando el tráfico no tiene seguridad (26,04 Mbps) o tiene seguridad WPA2 (31,66 Mbps). Mientras que la dependencia menor se da en el Escenario 2 de TFTP con el 1,2% (32,31 Mbps sin seguridad y 31,92 % con seguridad WPA2). Es decir, una diferencia de 16,3% entre extremos.



	TFTP Escenario 1	TFTP Escenario 2	TFTP Escenario 3	FTP Escenario 1	FTP Escenario 2	FTP Escenario 3
sin seguridad	19,36	13,66	25,88	39,38	24,79	66,17
WPA2	22,39	14,16	30,50	33,81	21,63	67,29

Figura 6. Comparación de rendimiento efectivo 802.11n 144 Mbps.

5.2. Rendimiento Efectivo

Analizando el rendimiento efectivo de datos en la Figura 6, se desprende que:

- El rendimiento efectivo es lógicamente menor que el rendimiento bruto, y entonces, más bajo respecto al valor nominal.
- El mejor de los casos se da para FTP con WPA2 en el Escenario 3, con el 46% de la velocidad nominal (67,29 Mbps). Y el peor de los casos se da para TFTP sin seguridad en el Escenario 2, con el 9% de la velocidad nominal (13,66 Mbps). Es decir, una disparidad de los valores extremos del 37%.
- El cambio de Escenario influye en el rendimiento efectivo de TFTP y FTP. Entre el Escenario 2 y el Escenario 3 de TFTP se alcanza la diferencia máxima del 100%.
- Con FTP, en promedio, se obtiene casi el doble de rendimiento efectivo que en TFTP.
  1. Entre los Escenarios 1 hay una diferencia del 76%,
  2. Entre los Escenarios 2 del 64%, y
  3. Entre los Escenarios 3 del 136%.
- Las dependencias con respecto a la seguridad implementada son menores que las dependencias respecto al tipo de escenario. La dependencia mayor de la seguridad se da usando TFTP en el Escenario 3 con el 17,8%, y la dependencia menor se da usando

FTP en el Escenario 3 con el 1,7%. Es decir, una diferencia de 16,1%.

5.3 Visión General

En resumen, se concluye que:

- El rendimiento bruto y efectivo en Wi-Fi es altamente dependiente de la configuración de la red, del tipo de tráfico y de la seguridad utilizada.
- Para los Escenarios, dispositivos y herramientas propuestos se ha encontrado un impacto diferenciado según sea la configuración específica utilizada. El rendimiento bruto o efectivo puede cambiar hasta el:
  - 160% según el tipo de tráfico (TCP-FTP o UDP-TFTP) utilizado,
  - 100 % según el Escenario sobre el que se haga la transferencia, y
  - 18% según la configuración de seguridad.
- El rendimiento bruto (efectivo) está, en el mejor de los casos, en el orden del 55% (50%), y en el peor de los casos en 16% (9%) del rendimiento nominal.
- El índice de rendimiento efectivo varía entre el 41% y el 93% según el Escenario, el tipo de tráfico y la seguridad.

Figura 6. Comparación de rendimiento efectivo 802.11n 144 Mbps.

6. Principales Conclusiones

El uso de un ambiente controlado para las experimentaciones favoreció la obtención de diversas y detalladas métricas sobre las que se han basado las siguientes conclusiones. Aunque dos ensayos sucesivos y similares en estos ambientes inalámbricos no son exactamente repetibles cuantitativamente, los comportamientos de la red en los ensayos generaron valores promedio bastantes equilibrados.

De los resultados de los ensayos experimentales se observa que:

1. Para las herramientas de transferencias de archivos utilizadas, en general, TCP mostró mejor comportamiento que UDP.
2. La cantidad de enlaces inalámbricos entre las estaciones origen y destino de las transferencias influye directamente en las prestaciones. El rendimiento de los Escenarios 3, 1 y 2 bajó, en ese orden.
3. No se observa una tendencia predominante, ni tan significativa, como en los aspectos anteriores, que justifique relegar los esquemas de mayor seguridad como WPA2.

Debe destacarse que los ensayos brindaron un mejor comportamiento para TCP que con UDP, como en el artículo presentado en [14]. De la cuenta de paquetes, presentadas en las Tablas I, II y III, se desprende que hay diferencias, pero no sustanciales en cuanto a la cantidad y tamaño promedio de las tramas en ambos sentidos y en el sentido de la transferencia, de la estación receptora en TCP respecto a UDP. Se observa que la cantidad de confirmaciones en DCF es comparable. Y además, que el tamaño de la ventana deslizante con créditos de TCP se ha mantenido bajo, y no es este el motivo del mayor rendimiento TCP. Si se observa una diferencia importante en los tiempos intertramas promedio, que acompañó inversamente la relación de rendimientos entre TCP y UDP. UDP presentó, en general para las aplicaciones utilizadas, mayor tiempo intertrama que TCP.

Por otro lado, es conveniente destacar la importancia conceptual del índice efectivo de rendimiento, como el porcentaje de tramas utilizadas para transportar los datos respecto a la cantidad total de tramas. A mayor valor del índice efectivo de rendimiento mayor rendimiento efectivo. Esto implica que puede llevar a conclusiones erróneas comparar los comportamientos de Wi-Fi usando sólo como parámetro la métrica de rendimiento bruto.

Finalmente, se observa que en la mayoría de las mediciones, para un mismo escenario y tipo de tráfico, la variación de la configuración de seguridad no introdujo un impacto muy significativo, lo que manifiesta la mejora que se ha alcanzado en las implementaciones de los procesos de cifrado. El parámetro seguridad no tiene tanta influencia en las prestaciones, aun cuando los tamaños de las tramas (de 1564 bytes sin seguridad, 1572 bytes con seguridad WEP y 1580 bytes con seguridad WPA2) y el procesamiento crece con el tipo de seguridad implementada.

Finalmente, se efectuaron ensayos experimentales para las transferencias del archivo de texto, usando el generador de tráfico sintético IP Traffic [30], lo que permitió variar el tamaño de segmento. En estos ensayos se observó que al disminuir el tamaño de segmento bajó el rendimiento. También se sustituyeron las placas de red Wi-Fi USB por PCI, obteniéndose una mejoría de rendimiento. Este último aspecto no es concluyente, pero muestra que la variación del tipo de equipamiento de red y/o del fabricante, también es una variable más en el comportamiento general de la red.

7. Aportes y Trabajos Futuros

Las contribuciones principales de este trabajo, teniendo en cuenta la metodología, escenarios y recursos utilizados, son:

7.1 Aportes

1. La verificación, con resultados cuantitativos detallados, que los diversos esquemas de seguridad, topologías de red y tipos de tráfico en Redes WLAN 802.11 generan un impacto diferenciado en las prestaciones el tráfico de datos bruto y efectivo.
2. La verificación de que el rendimiento máximo de las comunicaciones con enlaces Wi-Fi se relaciona con una tasa bruta real (efectiva) inferior al 55% (50%) de la tasa nominal de velocidad, y que en algunos casos puede disminuir al 16% (9%). Esto valores serán menores si aumenta el número de estaciones.
3. Y que la proporción del tráfico bruto que es verdaderamente de datos puede variar entre el 46% al 90%.
4. La especificación de tres escenarios experimentales y la metodología para establecer mecanismos comparativos entre los distintos ensayos, usando un tipo de tráfico real con la transferencia de archivos txt, y un método de medición pasivo.

7.2 Trabajos Futuros

Los trabajos futuros tienen en cuenta la profundización de los estudios de comportamiento de las redes Wi-Fi. Inicialmente, una línea de investigación en estudio es la evaluación cuantitativa del comportamiento de Wi-Fi con los mismos escenarios propuestos, pero modificando las características del tráfico a multimedial y/o en zonas de saturación, aumentando el número de estaciones, pero sobre 802.11e. Otro aspecto de interés es la determinación cualitativa y cuantitativa de las interferencias y ruido de otras fuentes Wi-Fi y no Wi-Fi en ambientes de laboratorio. Para ello, se analiza la modificación de los escenarios a ambientes más complejos, pero controlados. Finalmente, otro curso de investigación complementario que se tiene como alternativa es el desarrollo de un relevamiento y compilación de las aplicaciones 802.11, y de las características y del comportamiento del tráfico generado, y su influencia sobre las prestaciones de la red

8. REFERENCIAS

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std 802.11, Institute of Electrical and Electronics Engineers, 2012.

[2] Wi-Fi CERTIFIED Wi-Fi Protected Setup™: Easing the User Experience for Home and Small Office Wi-Fi® Networks (2014), http://www.wi-fi.org/file/wi-fi-certified-wi-fi-protected-setup-easing-the-user-experience-for-home-and-small-office-wi, September 2014.

[3] B. Brown, "802.11: the security differences between b and i", IEEE Potentials, vol. 22, October - November 2003, pp. 23–27.

[4] H. Boland and H. Mousavi, "Security issues of the IEEE 802.11b wireless LAN", in Canadian Conference on Electrical and Computer Engineering, 2004, vol. 1, May 2004, pp. 333–336.

[5] Amendment 6: Medium Access Control (MAC) Security Enhancements, ANSI/IEEE Std 802.11i, IEEE, 2004,2012.

[6] The State of Wi-Fi® Security: Wi-Fi CERTIFIED™ WPA2™ Delivers Advanced Security to Homes, Enterprises and Mobile Devices (2012), http://www.wi-fi.org/file/the-state-of-wi-fi-security-wi-fi-certified-wpa2-delivers-advanced-security-to-homes, September 2014.

[7] B. Potter, "Wireless security's future", IEEE Security & Privacy Magazine, vol. 1, July - August 2003, pp. 68–72.

[8] WPA2 Q&A, Wi-Fi Alliance Document,

[9] B. Bing, "Measured performance of the IEEE 802.11 wireless LAN", in Conference on Local Computer Networks, LCN '99, Lowell, Massachusetts, USA, 1999, pp. 34–42.

[10] L. Chandran-Wadia, S. Mahajan and S. Iyer, "Throughput performance of the distributed and point coordination function of an IEEE 802.11 wireless LAN", in IEEE International Conference on Computer Communications (ICCC), August 2002, New York, USA, 2002.

[11] A. Vasan and A. Shankar, "An empirical characterization of instantane-throughput in 802.11b WLANS", Department of Computer Science, University of Maryland, USA, September 2002.

[12] G. Xylomenos and G. Polyzos, "TCP and UDP performance over a wireless LAN", in Eighteenth Annual Joint Conference of the IEEE Computer and Communication Societies. Proceedings. IEEE, vol. 2, New York, USA, March 1999, pp. 439–446.

[13] R. Hunt, J. Vargo and J. Wong, "Impact of security architectures on wireless network performance", in 5th IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2003), Singapore, October 2003.

[14] N. Baghaei and R. Hunt, "Security performance of loaded IEEE 802.11b wireless networks", Computer Communications, Elsevier, U.K., vol. 27, no. 17, 2004, pp. 1746–1756.

[15] E. Arruda, P. Fonseca, M. Leitao and P. de Barros, "Security versus Bandwidth: The Support of Mechanisms WEP e WPA in 802.11 g Network", Instituto de Estudios Superiores de Amazonia (IESAM), Journal of Information Systems and Technology Management, vol. 5, no. 2, 2008.

[16] S. Bohn, S. Groß, R. Nüßgen and P. Schwann, "An Automated System Interoperability Test Bed for WPA and WPA2", 0-7803-9412-7/, IEEE Radio and Wireless Symposium, S. Diego, California, USA, 2006.

[17] H. Bulbul, I. Batmaz and M. Ozel, "Wireless Network Security: Comparison of WEP (Wired Equivalent Privacy) Mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) Security Protocols", e-Forensics 2008, ICST 978-963-9799-19-6, Adelaide, Australia, January 21-23, 2008.

[18], A. Lashkari, M. Mansoori and A. Danesh, "Wired Equivalent Privacy (WEP) versus Wi-Fi Protected Access (WPA)", 2009 International Conference on Signal Processing Systems, Singapure, 2009.

[19] A. Gin and R. Hunt, "The Performance of the IEEE 802.11i Security Specification on Wireless LANs", Proceedings Fifth New Zealand Computer Science Research Student Conference, New Zealand, 2007.

[20]T. Hayajneh, S. Khasawneh, B. Jamil and A. Itradat, "Analyzing the Impact of Security Protocols on Wireless LAN with Multimedia Applications", SECURWARE 2012: The Sixth International Conference on Emerging Security Information, Systems and Technologies, Roma, Italia, 2012.
[21] S. Kolahi, S. Narayan, Y. Sunarto, D. Nguyen and P. Mani, "The Impact of Wireless LAN Security on Performance of Different Windows Operating Systems, in Proceedings of the IEEE Symposium of Computers and Communications(ISCC), pp. 260-264, Morocco, 2008.
[22] E. Tews and M. Beck, "Practical attacks against WEP and WPA", In Proceedings of the second ACM conference on Wireless network security (WiSec '09), Zurich, Switzerland, 2009, pp. 79-86.
[23] A. Gin and E. Hunt, "Performance Analysis of evolving Wireless IEEE 802.11 Security Architectures", Proceedings of the 8th International Conference on Mobile Technology, Applications, and Systems, vol. 2, San Francisco, California, USA, 2008, pp. 101-106.
[24] G. Begh and A. Mir, "Quantification of the Effect of Security on Performance in Wireless LANs", Proceeding of the 3rd International Conference on Emerging Security Information, Systems and Technologies, Athens/Glyfada, Greece, June 2009, pp. 57-62.
[25] N. Baghaei and R. Hunt, "Security performance of loaded IEEE 802.11b wireless networks", Computer Communications, Elsevier, U.K., vol. 27, no. 17, 2004, pp. 1746–1756.
[26] https://filezilla-project.org/
[27] http://tftpd32.jounin.net/tftpd32.html
[28] L. Chappell, "Wireshark Network Analysis: The official Wireshark Certified Network Analyst Study Guide", 1st Edition, Protocol Analysis Institute, Chappell University, 2010.
[29] http://www.airpcap.fr/

Virtualization Extensions into a Microkernel based Operating System

Pablo Pessolani
Departamento de Ingeniería en Sistemas de Información
Facultad Regional Santa Fe
Universidad Tecnológica Nacional
Santa Fe, Argentina
ppessolani@frsf.utn.edu.ar

Abstract

Since it emerged in the late '60s to the present virtualization technology has progressed and the expression "Virtual Machine" (VM) has changed its meaning, considering the virtualized component and how and where the virtualization is done. The software component for virtualization is the hypervisor that allows to create, to run and to end VMs.

On the other hand, in the mid-1980's emerges the microkernel technology as the minimal software layer that provide basic mechanisms to build Operating Systems (OS). In microkernel based OSs' components (filesystem management, process management, networking services, etc.) run isolated in user-mode without supervisor privileges providing fault isolation for reliability and reducing the trusted computing base (TCB) for security.

Despite design differences among microkernels and hypervisors, they share some common features and goals. This contribution explores the opportunities to include virtualization support on Minix OS presenting several approaches. It also details the design, the implementation and the evaluation of Mhyper, a microkernel/hypervisor prototype resulting from adding virtualization extensions to Minix.

Keywords: microkernel, hypervisor, virtualization.

1. Introduction

Virtualization technology developed in the late '60s [1] has emerged in the 1990s as a consequence of its attractive features such as the ability to consolidate multiple VMs on a single physical computer, increasing energy efficiency and providing insulated execution environments for the safety and the security of critical applications.

The formal requirements to be met by virtualization technology were presented by Popek and Goldberg [2] in 1974. Over time, as the technology progressed, the expression "*Virtual Machine*" has changed its meaning, considering the virtualized component and how and where the virtualization is done. The hypervisor or virtual machine monitor (VMM) is a key piece of software for virtualization that allows creating, running and finishing VMs.

In the mid-1980's emerge the microkernel technology as the minimal software layer that provide mechanisms needed to implement OSs [3, 4]. In microkernel based architectures the OS components (Filesystem, Process Management, Networking Services, etc.) run isolated in user-mode without supervisor privileges providing fault isolation for reliability and reducing the trusted computing base (TCB) for security [5].

Despite design differences among microkernels and hypervisors, several researchers explored the convergence of both technologies [4,6,7] because they share some common features and goals and their integration is presented as a promising approach. Modern microkernel-based OS already have abstractions and mechanisms required to support virtualization.

In this article there are proposed several virtualization approaches as extensions into a microkernel based OS (*Minix* [8]). Furthermore, it presents the design, the implementation details and the evaluation of Mhyper, a microkernel/hypervisor prototype. It is the result of adding virtualization extensions to Minix in all its layers to support new abstractions.

The chosen system for the development was *Minix* 3, a multi-server OS, in which system calls made by user processes are messages transfers to one of several servers that also operate in user-mode. By its layered architecture, microkernel based OSs offer several ways for virtualization that will be described in the Minix 3 Virtualization Options Section, and *Minix* 3 seems to provide the right capabilities required for this project. The most valuable features of virtualization systems would be kept:

- *Security Isolation*: A process within a VM can only send messages to processes that belong to its own VM. All message transfers are controlled by the microkernel-hypervisor that enforces a global scheduling policy, verifying the source and destination of every message. Memory access is controlled by the microkernel-hypervisor and therefore does not admit access outside the assigned memory range. As user-space OS components are confined within a VM, each guest OS has its own set of users (including root), PIDs, filesystems, network addresses, etc.
- *Resources Isolation:* The microkernel-hypervisor can schedule the execution time for processes that belong to VMs, partitioning the time in two levels, VMs and processes. Furthermore, each VM has its own memory address space and its real/virtual storage device to work on.

–*Fault Isolation*: The confinement of all user-space OS components inside a VM avoid that a failure in their processes or assigned resources will affect other VMs.

The paper is organized as follows. The Background Section starts off with a brief introduction on virtualization technologies, microkernel technologies and a description of *Minix* microkernel-based OS. The review of other research projects that involves the join of microkernels and hypervisors features is presented in the Related Works Section. Minix 3 Virtualization Options Section describes several approaches for virtualization in a multi-server OS. Mhyper Design Section details the resulting architecture, virtualization alternatives and design decisions made for the project. A description of the successful implementation of the virtualization extensions into *Minix* is presented in the Mhyper Implementation Section. Before concluding, the analysis of comparative benchmarks is exposed in the Mhyper Evaluation Section. And finally, the Conclusion and Future Works Section presents the implications of the presented project and proposes future research derived from it.

2. Background

This section provides an overview of technologies that will allow the reader to understand the design and implementation of the proposed system.

2.1. Virtualization Technologies

2.1.1. Hardware/Platform Virtualization. Modern CPUs have an instruction set (ISA: Instruction Set Architecture) where a subset of them are privileged instructions. Hardware virtualization technology takes advantage of this CPU's feature to encapsulate VMs. The OSs running inside these VMs (named Guests) have the illusion of running on bare hardware with highest privileges. Their kernels assume that they have all the privileges required to manage the underlying hardware, but really they run in user-mode. If the OS-Guest kernel executes a privileged instruction in user-mode as read/write an I/O port, or enable/disable interrupts the CPU triggers a protection fault. The hypervisor or VMM traps this fault and emulates the behavior of the hardware or performs the operation of the real hardware on behalf of the Guest OS.

If the VMM runs on and emulate the underlying raw hardware it is said that it does full-virtualization. A Type 1 VMM runs directly on the hardware; a Type 2 VMM runs on another operating system. These technologies are used in products as *Vmware's vSphere ESXi*, *Oracle's VirtualBox* and *Microsoft Virtual Server*.

2.1.2. Para-virtualization. Hardware or Platform virtualization suffers performance penalties because each privileged instruction the kernel of the Guest OS executes triggers a CPU Trap that requires two CPU mode switch (User-mode to Supervisor-mode and reverse) and the emulation of the instruction made by the VMM.

Para-virtualization avoids these showers of traps and the emulation of hardware behavior modifying the interface among Guest OSs and the VMM. The VMM offers virtualization service to Guests OSs named *Hypervisor Calls*, therefore it requires modifications into the kernel of the guest OS to take advantage of this technology. These technologies are used in products as *Xen*, *Denali* and *Hyper-V*.

2.1.3. Operating System Level Virtualization. OS level virtualization is a partitioning method where the kernel of an OS allows for multiple isolated user-space instances, instead of just one. The virtualization system builds a Container/Jail that isolate a set of applications to its own execution environment. Therefore, each Container/Jail has its own root filesystem, its own sets of users (including root), its own set of Inter-Process Communications (IPC) facilities, etc.

Generic applications use OS services through System Calls. OS level virtualization software traps system calls and modifies their behavior to simulate an isolated execution environment for the application. Therefore, multiple instances of those execution environments can be executed in a secure way, but sharing the same OS.

OS level virtualization technology usually imposes a small system overhead, because programs in Container/Jail use the system call interface and do not need hardware emulation and does not require hardware assistance to perform efficiently. These technologies are used in products as *LXC*, *OpenVZ*, *Virtuozzo*, *Linux-VServer*, *Solaris Zones* and *FreeBSD Jails*.

2.1.4. Process/Application Virtualization. Process or Application Virtualization is a technology that encapsulates application software from the underlying OS on which it is executed. The virtualization system builds a sandbox that isolates the application from the underlying OS. The term "*virtualization*" refers to the application, not to the abstracted artifact as the OS or hardware. Examples of products that provide this technology are *Citrix XenApp*, *P-apps*, *VMware ThinApp*, and *Microsoft Application Virtualization*.

2.1.5. Operating System Virtualization. An OS is a software layer that manages computer hardware resources providing service for programs and it sits between applications and hardware.

In OS virtualization technology the Guest-OS does not rely directly on the hardware, but uses the services offered by the Host-OS. It seems appropriate to refer to the Guest-OS as a Virtual Operating System (VOS).

As in para-virtualization technology, the OS-Guest must be modified, but instead of be served by a hypervisor it is served by a Host-OS. *User Mode Linux (UML)*, *Nomad Linux* and *Minix Over Linux* (*MOL*) are based on this approach.

2.2. Microkernel Technology

From the beginnings until today there are many OSs implemented as a monolithic kernel. This means that most system functions are in a single program (the kernel) that manages CPU, memory, processes, filesystems, device drivers, networking, etc. This approach has several performance benefits as:

–All kernel functions and global variables can be shared by any kernel component with full access to them.

–System calls are implemented as CPU traps.

But, their main drawbacks are:

–The complexity involved in maintaining that big code without affecting other components.

–The security and safety risks that involve sharing the same address space between dissimilar modules.

Microkernels were developed in the 1980s as a response to those problems, minimizing the kernel and implementing servers outside in user-space.

In the microkernel basic memory management, IPC mechanisms and basic scheduling are implemented. All servers and device drivers run in user-mode. Since each server and drivers has allocated their own address spaces, their objects are protected from one another.

Microkernels have benefits as [9]:

–The system becomes more flexible and extensible.

–A clean microkernel interface enforces a more modular system structure.

–A smaller kernel can be more easily maintained and should be less prone to error resulting in a more reliable and robust OS.

–It reduces the trusted computing base (TCB) [10].

The main drawback of microkernels is that a simple system call (as *getpid()*) may require a pair of message transfers. A complex system call requires a series of message transfers between a user process, one or more servers and one or more drivers with a negative impact on performance. Each message transfer involves context switches (insertion/removal of processes from the ready queue) multiple invocations to the scheduler, and in some kernels and, in some OSs, the copy of the message between the sender's and the receiver's address spaces.

Some of the performance advantages of monolithic over microkernel OS is that the former rely on obtaining the benefits of cache-coherent shared-memory SMP that has become the default on modern PCs.

On SMP systems, the cache consistency is maintained by the hardware, therefore discharging the software of doing that task. In shared memory OS kernel objects reside as global data structures and functions.

As trends suggest, chips with tens to hundreds of cores will appear within five years [11]. As increase the number of cores, maintain the cache consistency will result in a bus bottleneck.

From this reasoning emerge multi-kernel OS, a new OS programming model that treats a multicore machine as a network of independent cores as if it were a distributed system. It does not assume shared memory, but rather implements IPC as message-passing [12]. With many-cores computers, a core can be assigned to critical processes as the Filesystem Server, the Process manager, the Network manager and frequently used device drivers sharing the other cores among other non-critical process and user's processes [13]. Microkernel based OS can be adapted for many-cores as it is shown in [14].

2.2.1. Minix 3. *Minix* [8] is a complete, general-purpose, time-sharing, multitasking, microkernel-based and multi-server OS developed from scratch by Andrew S. Tanenbaum broadly used in Computer Science degree courses.

Though it is copyrighted, the source has become widely available for universities for studying and research. Its main features are:

–*Microkernel based*: Provides process management and scheduling, basic memory management, IPC, interrupt processing and low level I/O support.

–*Multilayer system*: Allows modular and clear implementation of new features.

–*Client/Server model*: All system services and device drivers are implemented as server processes with their own execution environment.

–*Message Transfer Interprocess Communications (IPC):* Used for process synchronization and data transfers.

–*Interrupt hiding*: Interrupts are converted into message transfers.

Minix 3 is a new open-source OS [8] designed to be highly reliable, flexible, and secure. It is loosely based somewhat on previous versions of *Minix*, but is fundamentally different in many key ways. *Minix* 1 and 2 were intended as teaching tools; *Minix* 3 adds the new goal of being usable as a serious system for applications requiring high reliability.

The *Minix* 3 kernel is very small (about 5000 lines of source code) and it is the only code that runs under kernel privilege level. User processes, system servers and device drivers are isolated one from another running with lower privileges (Figure 1). These features and other aspects greatly enhance system reliability and it can be characterized as a multi-server OS [15].

A user-mode application make system calls using IPC, packaging the request code and input arguments in a message, then it receives the results in another message in the same way as RPC does. Two servers handle application's requests from system calls; they are the Process Manager (*PM*) and the Filesystem Server (*FS*). *FS* and *PM* also make requests to device driver processes or to other servers using IPC. The Information Server (*IS*) gets information from the other servers and tasks to present it on the system console once the user presses "*Fx*" keys on console keyboard. The Reincarnation Server (*RS*) is used to convert a conventional user process into a Server or Task and to assign it specified privileges for resource access and IPC functions. It is also acts as a watchdog process that controls servers and tasks and restarts them if it detects a failure.

The *Minix* kernel is like a message broker where user processes, servers and tasks exchange messages through it. Processes on Minix 3 communicate using the following IPC primitives:

–*send():* to send a message to a process.

–*receive():* to receive a message from a specified process or from any process.

–*sendrec():* to send a request message and to receive the reply from a process.

–*notify():* a non-blocking send of a special message type to a process.

These IPC functions use *endpoints* as the source or the destination of messages because each process has an associated *endpoint* that identifies it.

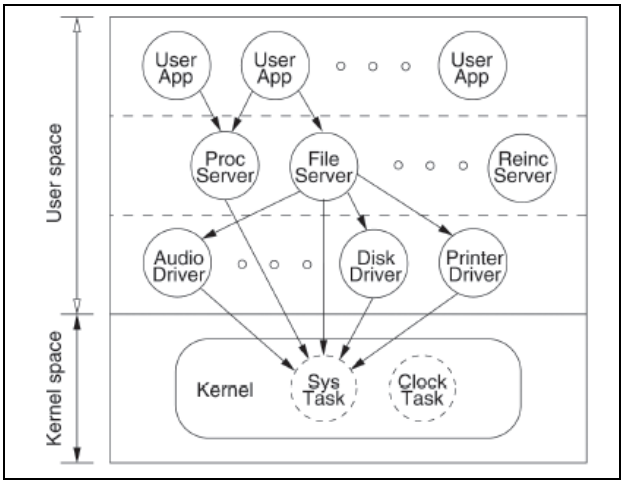


Figure 1. The structure of Minix 3 (from [16]).

A special task named *SYSTASK* (also known as *SYSTEM* Task) runs in kernel or supervisor mode, sharing the same address space with the microkernel, therefore it can access microkernel and *CLOCK* task global variables (Figure 1). If a *Device Driver Task* (*DDT*) needs to do a privileged operation as setting an interrupt handler or writing in an I/O port, it must make a *kernel call* to the *SYSTASK* that perform the required operation after checking the task's capabilities.

The *Minix* microkernel controls message transfers among processes and which kind of IPC primitives a process can use [17]. Moreover, the *SYSTASK* controls which kernel calls a process can request, and the memory address spaces and I/O ports that a DDT can manage through *SYSTASK*. These features ensure security [17] and fault isolation [18] between processes and allow strict access control to resources as memory, devices and I/O.

The project described in this article is based on the following facts:

–In para-virtualization systems, device drivers use hypervisor-calls to perform privileged operations, as DDTs use kernel-calls in *Minix*.

–The isolation characteristics of *Minix* are comparable with those of virtualization systems.

Therefore, adding virtualization support to *Minix* requires:

–To create execution partitions that encapsulate processes called Virtual Machines (VMs).

–To only allow messages transfers between processes within the same VM (except *SYSTASK* and *CLOCK*).

–To modify *SYSTASK* and *CLOCK* so they can handle requests coming from processes of all VMs (as promiscuous tasks), adding to them the hypervisor's features (para-virtualization).

–To modify the scheduler so that the performance of a VM does not affect processes of other VMs partitioning the time among VMs.

–To add new tasks, servers and utilities which allow the administrator to configure and to manage VMs.

This project was developed using *Minix* 3.1.2a that uses a segmented memory model (on PC hardware x386 and above) for memory management, sharing and protection.

3. Related Works

As microkernels and hypervisors have different goals, it does not mean that they cannot be combined to meet both sets of goals. There are several research papers that proof this assertion relating microkernel and hypervisors, but [19] and [20] are distinguished ones because the opposite approaches they propose and discussions that they caused in the scientific community and industry. The foundation concepts of each approach and their implications on the expected benefits of virtualization in embedded discusses in [21]. The resulting question is: "*Is it more convenient to include features of a microkernel into a hypervisor [19] or to include features of a hypervisor into a microkernel [20]?*".

In [19] it is argued that hypervisors are a specialized form of microkernels but done right. Obviously, in [20] it is refuted the statements presented in [19] as poorly justified. Without going into that discussion, the title of this paper suggests that the position of the author of this article is closer to [20].

The NOVA micro-hypervisor [22] implements a thin microkernel and user-mode environment for core functions of the system implemented in separate layers. NOVA uses the virtualization capabilities of newer hardware platforms to implement full-virtualization, therefore unmodified guest OSs are supported. The VMM also implements device emulation for devices made available to each guest.

Another proof of the existence of the convergence of a hypervisor on a microkernel (microvisor) is OKL4 [23]. The OKL4 microvisor supports para-virtualization and use para-virtualized guests. It does not require any instruction emulation because privileged operations are done by explicit hypercalls. The OKL4 microvisor implements partitions known as secure cells for partitioning VMs in the architecture. In [24] a virtualization solution based on the Fiasco microkernel as a hosts run a slightly modified Linux as the guest OS.

PikeOS [25] primarily implements what is called a separation kernel and relies on para-virtualization. Like a hypervisor, a separation (or partitioning) kernel securely isolates environments for upper-level guests. Separation means that a fault in one system cannot impact other applications and the hypervisor is unaffected.

Although Linux it's not a microkernel based OS, it has KVM [26] as a full virtualization solution inside the kernel (as a loadable kernel module). It works on x86

hardware containing virtualization extensions. KVM adds the kernel hypervisor features, but as Linux is monolithic it increases the TCB even more (Linux has +2,500,000 SLOC).

A software-based virtualizable architecture called Fluke [27] allows recursive virtual machines (virtual machines running on other virtual machines). It is implemented by a microkernel running on generic hardware. Fluke does not attempt to emulate an actual hardware architecture closely (as it is followed by a traditional VM architecture), but is instead designed along the lines of a process model and implemented by a microkernel.

VLK (vCPU Layer in Kernel) [28] is an open source virtualization layer combining RTOS and Linux on multicore embedded systems. Another work related to the embedded systems is presented in [29]. It describes the changes made to the L4 API and implementation and of a Wombat server, a para-virtualized Linux system running on top of L4.

A different kind of virtualization approach is used by User-mode Linux [30] (UML). It is a port of the Linux kernel running on top of a Linux host. It works on virtual hardware constructed from resources provided by the host OS, therefore the Linux host is like an assisted VMM (para-virtualization). Processes running inside the Linux Guest have no access to any host resources other than those explicitly provided to the VM.

The project presented in this paper is closer to the UML approach, but as Minix is a microkernel based OS with layers of servers and tasks in user-mode, it offers several virtualization options that are described in the next section.

4. Minix 3 Virtualization Options

As a multilayer client-server OS, *Minix 3* presents several opportunities to implement a variety of virtualization modes. Information sharing between processes is done by IPC message transfers and by the copy of data blocks between address spaces. Messages could be intercepted, modified, redirected [31], deleted or even created, and the level where these operations are done defines the kind of virtualization that would be implemented. Figure 2 presents the structure of *Minix* and the layers available for virtualization.

In the layer below user's applications are the *FS*, *PM*, *RS*, *and IS* that forming the server layer. Below it, there are unprivileged DDTs forming the task layer. At the lowest layer lie the microkernel and two privileged tasks, *CLOCK* and *SYSTASK*. Eventually, a lower layer could exist like a real-time sub-kernel or a hardware hypervisor.

4.1. Server Level Virtualization

A user-level application makes a system call using a message transfer to a Server (which depends on the system call) then, the Server replies to the application with a message transfer too. As the Servers are user-mode processes, the Server Level Virtualization approach allows having multiple instances of Servers dedicated to different sets of user-level applications. Each instance of a Server belongs to a VM. A VM in this environment can be considered as a way of grouping and confining processes. Each user-mode process belongs to a VM. The virtualization is done by intercepting system call requests coming from user-mode processes and redirecting them to the VM Servers.

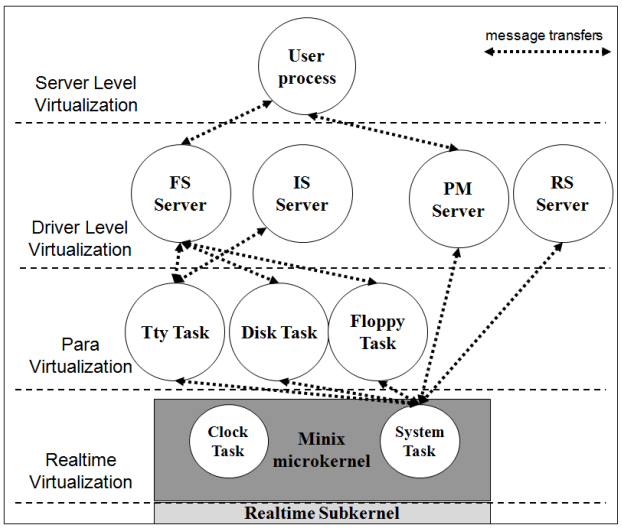


Figure 2. Minix 3 virtualization options.

The use of system calls interception in monolithic kernels is known as feather-weight virtualization [32]. Figure 3 presents *Mhyper* server level virtualization architecture that use the possibilities offered by IPC.

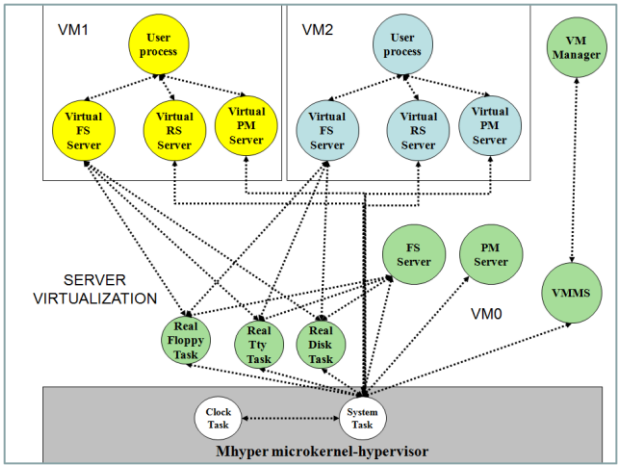


Figure 3. Mhyper server-level virtualization.

As each set of user-mode processes have an assigned VM, all processes belonged to the VM have their own root filesystem, their own PID space, their memory management policy, etc. For instance, VM1's *FS* might manage a filesystem installed on a real hard disk or installed on a virtualized disk such as an image file managed by VM0's *FS*. An experimental (but fully operational) *FS* for each VM was presented in [33].

The application isolation property is achieved because no message can be sent by any user-space process to servers outside its assigned VM. Message transfers

between processes within the same VM have the same behavior as in *Minix*. Fault isolation is only limited to Server faults and security isolation only covers attacks and vulnerabilities on applications and Servers. It also lacks of resource isolation because the *Minix* scheduler (embedded within the microkernel) schedules all processes (including Servers) as regular *Minix* processes without considering the VM they belong to.

The FS exposes files and directories to user-space applications as abstractions. On the other hand a disk device driver exposes disk blocks to the FS as abstractions. A regular file in *Minix* has a name and an associated i-node that keeps file attributes as the UID, GID, access permissions, file size, creation-access-modification times, the number of links, and data block mapping.

An Object layer could be inserted between Servers and Drivers as was proposed in [34] where the FS do not map disk blocks, instead of it a file is linked to a storage object controlled by the Object server. This approach separates file naming and metadata from storage. The FS does file naming and metadata management and the details of storage management are left to the object server.

4.2. Device Driver Level Virtualization

As DDTs are user-mode processes, the Driver Level Virtualization approach allows having multiple instances of DDTs, one assigned to each VM. For example, the *FS* of the stock *Minix* distribution installs on a hard disk has its root filesystem on a device with mayor number 3. The mayor number determines the task number that will handle the requests for that device. If multiple instances of this FS server are running, all of them will try to work on the same device (mayor number 3). To avoid this issue multiple instances of the Disk tasks could be installed. Each driver instance, should manage an assigned real or even virtualized device (as disk image files) and each VM could have its own *FS* server and its own disk device driver. Therefore, when the *FS* sends a message to the disk device driver, making a "*task call*", then the message is redirected to the disk device driver assigned to the FS's VM. There is no need to change or to rebuild the applications or the servers because the message redirection mechanism is transparent to that process level.

The application and server isolation are achieved because no message can be sent from any user-space process or server outside its assigned VM. Message transfers between processes within the same VM have the same behavior as in *Minix*.

On Driver Level Virtualization fault isolation is limited to Server and Tasks faults. Security isolation covers attacks and vulnerabilities on applications, servers and DDTs. Resource isolation is not its strength because the *Minix* schedules all processes as regular *Minix* processes without considering the VM they belong to.

4.3. Para-Virtualization

Usually, device drivers need to access to I/O ports, enable/disable interrupts, install interrupt handlers, copy data among different process address spaces, etc. As device drivers in monolithic OS are embedded (statically linked or by loading a module) in kernel space and have supervisor privileges there is no need to request service to any other component. The para-virtualization approach requires the modification of device drivers of the Guest OS to request services to the hypervisor ("*hyper-calls*").

As a microkernel OS, *Minix* servers and DDTs run in user-mode and cannot execute privileged instructions. They use "*kernel calls*" to request privileged operations to the *SYSTASK* using the IPC mechanism. This functionality is very similar to a para-virtualized hypervisor except that the standard *Minix SYSTASK* only works with one set of server and DDTs. With proper modifications to the microkernel, *CLOCK* and *SYSTASK Minix* could be converted in a microkernel-hypervisor that can manage multiple sets of processes (VMs). The modified *CLOCK* and *SYSTASK* are "*promiscuous*" process types because they handle requests coming from processes of all VMs (same as a hypervisor).

For example, suppose that the system has two Ethernet cards from the same provider (for ease of explanation). Each card has its own set of I/O ports, DMA channel, IRQ and memory address space. *VM1* can use the driver configured to use the first Ethernet card, and *VM2* can also use the same driver configuration. In a conventional system this is a conflict situation, but *Mhyper's SYSTASK* can dynamically modify the I/O ports, the IRQ, the DMA channel and the memory addresses that come from VM2's driver request to their counterpart of the second Ethernet card. On the other hand, when an IRQ occurs, it notifies the driver of the assigned VM. More sophisticated settings can be made that are presented in the *Mhyper Implementation Section.*

At present, the requests from drivers to *SYSTASK* in *Minix* are made through message transfers, but in [35] a direct call mechanism using traps (same as hardware virtualization) was proposed. This approach keeps *Minix* reliability running DDTs in user-mode, but avoiding the message transfer, process queuing and de-queuing, and scheduling overhead.

The para-virtualization approach has complete fault and security isolation because it covers all process levels up to the microkernel. Resource isolation is achieved using a modified k scheduler that handles two scheduling levels: VM scheduling and process scheduling. Each VM has an allocated number of tokens to consume during a time period. A process belonging to a VM can only be chosen by the process scheduler if the VM it belongs to has got at least one token. Therefore a time consuming process of one VM does not affect performance of processes running within other VMs.

4.4. Realtime Virtualization

A real-time subkernel for *Minix* 2.0 was presented in [36]. Then, it was ported to *Minix* 3.0 resulting in two branches, *RTMINIX3*[37] and *Realtime Minix 3*[38].

The real-time subkernel virtualizes time and IRQs for the non-real-time *Minix* Guest. Realtime tasks can share the hardware with *Minix* applications, but real-time IRQs, timers and soft-IRQs (also known as delayed IRQs) managed by real-time tasks have priority over every IRQ, process or task running within *Minix* Guest.

5. Mhyper Design

Mhyper's primary goal is to explore opportunities to include virtualization support on the *Minix* microkernel without significant compromises in performance.

For the *Mhyper* project the following design principles were adopted:

- –The system will offer several kinds of virtualization options.
- –The VMs only will run *Minix* as Guest OS.
- –Minix user-space software and applications will not require modifications.

The *Minix* OS can be seen as a microkernel acting as a (para-virtualized) hypervisor for a single VOS (*Minix*) running on it. To allow several VOSs running concurrently, the microkernel and other system components should be modified to support isolated resource partitions called VMs. The following are some design and implementation issues required to add virtualization support to *Minix*:

- –The microkernel should identify to which VM a process belongs to.
- –User-space processes should not notice that they are running within a VM.
- –A Guest-OS should be confined inside a VM emulating the behavior of a standard *Minix* allowing the execution of user-space applications, servers and tasks.
- –Each server within a VM will request services to tasks within the same VM (exceptions to this rule are detailed later).
- –A DDT will request services to *SYSTASK* on resources allocated to the VM it belongs to.
- –The *SYSTASK* should only admit requests from drivers on resources allocated to the VM they belong to.
- –CPU time must be fairly shared among all active VMs, and among processes within a VM.
- –The system should provide services (system calls) and tools that allow the administrators managing and monitoring the status of each VM, its processes and resources.
- –Each VM should have an allocated memory address space that it should be managed by its own PM. Any VM must not access the memory space allocate to other VM (exceptions to this rule are detailed later).
- –There should be a user-space tools that allow system administrators to create, to configure, to load, to start, to pause, to resume, and to finish VMs.

The *Mhyper* para-virtualized architecture that meets these requirements is shown in Figure 4.

Mhyper starts running *VM0*, a pseudo-VM that has all system resources allocated to it. As *VM0* has the same servers and tasks than stock *Minix*, if no other VM is started, *Mhyper* behavior is like *Minix*. *VM0* is a service VM which has particular security, resource and fault isolation properties and it should be used by system administrators to manage and monitor other VMs.

There are only one *SYSTASK* and only one *CLOCK* on the whole system. They were modified to handle requests and events from processes of all VMs (they are "*promiscuous*" tasks).

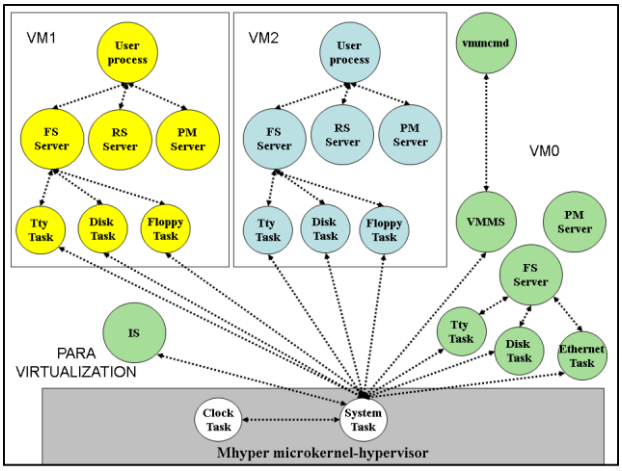


Figure 4. Para-virtualized Mhyper.

A new server was added to manage VMs called the *Virtual Machine Manager* (*VMM*) server. A command called ***vmmcmd*** is an administration tool to create, to configure, to load, to start, to pause, to resume, and to finish VMs using the system calls served by *VMM*.

To keep the security, resource and fault isolation properties of virtualization, the memory address space allocated for a VM cannot be accessed by any other VM. (VM0 is an exception). The *VM0's PM* has not got this memory address range allocated to its management until the VM finishes and release it. Furthermore, the *SYSTASK* keeps registers of the address space range allocated for each VM to control the validity of kernel calls that operate on memory.

A virtualization system with security isolation must not reveal any object or process it have inside each VM. A process within a VM should not be able to deduce that is running inside a VM or the existences of other VMs on the same host. An exception to this rule occurs with *VM0* where a special kind of processes runs within it to support some servers and tasks on other VMs. Those processes are "*promiscuous"* processes and "*proxies*" (detailed in *Mhyper Implementation Section*).

Promiscuous processes, virtual processes and proxies hide their real identities for all VMs (except *VM0*). An example of a promiscuous process is *VM0's IS* which can collect information of resources, processes, servers and tasks of all VMs to present them on the system console.

The *Minix* scheduler, embedded within the microkernel, maintains a set of priority queues, one queue allocated for each priority level (total 16). On each context switch the CPU scheduler selects the highest priority process from those queues and allocates CPU to it. To ensure performance isolation, each VM should have its own set of priority queues and the scheduler should be modified to support two level scheduling (VM level and process level). *Mhyper* scheduler considers those issues and other related to the latency and other

unwanted outcomes of a scheduling policy without particular considerations of the kind of processes that run within VMs.

6. Mhyper Implementation

This section presents a detailed description of *Mhyper* implementation. It describes the changes and additions made at all levels of *Minix* to support virtualization.

Before describing the implementation, it must be noted that unlike a monolithic system where process descriptors or Process Control Blocks (*PCB*) are located inside the kernel. In *Minix*, process descriptor fields are scattered in different software components such as the kernel, *PM* and *FS*. Each component only has the fields that it needs and manages being compliant with the security principle of least privileges.

6.1. Mhyper's Microkernel, CLOCK and SYSTASK

The *Minix* microkernel has a set of global variables, structures and tables to manage resources, processes, scheduling, etc. There is a process descriptor table where the status and attributes of each process are stored. *Mhyper* allocates an instance of those microkernel's global variables, structures and tables for each VM. A *vmid* field was added to process descriptor structure to identify the VM that a process belongs to. Therefore, there may be several processes with the same *endpoint* but different *vmid*.

The kernel IPC functions use the *vmid* field to control that message can only be exchanged between processes within the same VM. The exceptions to this rule are *SYSTASK*, *CLOCK*, promiscuous processes and proxy servers. Particularly, the *SYSTASK* and *CLOCK* process descriptors only exist within *VM0*, but they dynamically change their *vmid* fields according to the VM of the operation they are doing. The change of the *vmid* field for those tasks is automatically done by the microkernel in the modified *receive()* IPC primitive. When the *SYSTASK* receives a request from a process belonging to *VMx*, the microkernel sets *SYSTASK*'s *vmid=VMx*. Then, once the *SYSTASK* calls *receive()* again the microkernel resets the *vmid* to *VM0* (its default VM) until it receive the next request.

The following new kinds of processes were added to *Minix* to support different virtualization modes:

- *Virtual (also known as a stub process)*: It is only a process descriptor allocated in the process table of *VMx* (≠*VM0*) that appears as a real running process for the other processes within *VMx*. Hence, each virtual process has its own *endpoint* that can be referenced on IPC primitives.
- *Proxies*: There is only one proxy process allocated for each VM (≠*VM0*) running within *VM0*. A message sent to a *virtual* process on *VMx* will be received by the *proxy* process assigned to *VMx*. A message sent by a *proxy* appears as it was sent by a *virtual* process within *VMx*.
- *Promiscuous*: It is a process on *VM0* that can interchange messages with processes on other VMs. On each VM there is a process descriptor allocated for the process with its same *endpoint* that represents it within the VM. That descriptor is flagged as a promiscuous process on all VMs except VM0.

Processes of these types are transparent to other processes within a VM because they cannot distinguish between them from regular processes.

These types of processes seem that break the isolation property of virtualization, which is so appreciated. In fact, they partially do that (limited to *VM0*), but there is no need to use these facilities for virtualization. They were included to allow the source code reuse of servers and tasks used in the Server or Driver virtualization approaches.

The following new IPC primitives were added to the microkernel to be used by proxies and by promiscuous processes (therefore, can only be used by processes within *VM0*):

- *sendvm()*: A *promiscuous process* with endpoint ***ep*** within *VM0* send a message to another process on *VMx* as if it is within *VMx* with endpoint ***ep***.
- *notifyvm()*: A *promiscuous process* with endpoint ***ep*** within *VM0* send a *NOTIFY_MESSAGE* message to another process on *VMx* as if it is within *VMx* with endpoint ***ep***.
- *recvvm():* A *promiscuous process* with endpoint ***ep*** within *VM0* will receive messages sent to a process with endpoint ***ep*** on *VMx*.
- *sendrecvm()*: A *promiscuous process* with endpoint ***ep*** within *VM0* send a request message to another process on *VMx* as if it is within *VMx* with endpoint ***ep*** and then it will wait to receive the reply message.
- *sendas()*: A *proxy process* with endpoint ***ep0*** within *VM0* send a message to another process on *VMx* as if it was a *virtual process* in *VMx* with endpoint ***epx***.
- *recvas():* A *proxy process* with endpoint ***ep0*** within *VM0* will receive messages sent to a *virtual process* on *VMx* with endpoint ***epx***.

As DDTs within a VM can request to be notified on interrupts triggered by their owned devices, the microkernel was modified to register DDT's endpoints and *vmid*s for future notifications on interrupts. On every device interrupt, the microkernel will send a *NOTIFY_MESSAGE* message to the registered {*endpoint*,*vmid*}.

Considerable changes were made to *SYSTASK* to allow it to handle for requests of tasks and servers of all VMs and to control that those requests match the assigned resources for them. If a task or a server with endpoint ***ep*** within VM *vmid* requests *SYSTASK* to start an alarm (timer), once that alarm expires a *NOTIFY_MESSAGE* message is sent to process identified as {***ep***, *vmid*}.

As was mentioned earlier, *SYSTASK* is like a para-virtualized hypervisor that execute privileged operation on behalf of user-space processes like servers and tasks. The data block copy between the address spaces of different processes and the virtual to physical address conversions are these kinds of operations. Two new service functions were added as kernel calls to support

the copy of data blocks that involve processes on different VMs. They are:

- *sys_vcopyvm():* It copies data blocks between the address spaces of two processes, at least one of them within a VM≠*VM0*. The requester must be a proxy or a promiscuous process (within VM0).
- *sys_umapvm():* It converts the virtual address of a process within a VM (≠*VM0*) to a physical address.

All IPC primitives (the original and the added ones) were modified to check that proxy processes can only exchange messages or copy data with processes on VM0 and processes within the VM that it represents. In the same way, they check that can only exchange messages or copy data with processes on VM0 and processes within those VMs where it has authorization.

As was presented in [9], the duties of the *CLOCK* task in *Minix* include:

- Maintaining the time of day.
- Preventing processes from running longer than they are allowed to.
- Accounting for CPU usage.
- Handling the alarm system call made by user processes, but requested by the *PM* server.
- Providing watchdog timers for parts of the system itself.
- Doing profiling, monitoring, and statistics gathering.

Several modifications and additions were required to adapt the *CLOCK* task to the virtualization environment. They include:

- Maintaining the uptime for each VM.
- Adding and removing tokens of VMs' buckets.
- Accounting VMs' CPU usage.
- Handling the alarm system call made by user processes, but requested by the PM servers of all VMs.
- Providing watchdog timers for all VMs
- Doing profiling and statistics gathering for all VMs.

A split device driver model can also be implemented on *Mhyper* considers a *front-end* (*FE*) DDT within *VMx* (≠*VM0*) that communicates with a *back-end (BE)* DDT within *VM0* through a virtual task in *VMx*. The *front-end* converts the requests to *VMx* into requests to the *back-end*. There is not direct communication between the *front-end* and the *back-end*; it is done through a virtual task (a stub) on *VMx* that represents the *back-end*. The drawback of this model is that it requires another level of message transfers (between *FE* and *BE*).

The *Minix* scheduler was modified by *Mhyper* to meet the performance isolation requirement of virtualization. As in *Linux-VServer* [39] a token bucket filter (*TBF*) was installed over the standard *Minix* scheduler. Each VM has a token bucket that accumulates tokens at a specified rate; every timer tick, the VM that owns the running process is charged one token. Once a VM runs out of tokens, the scheduler switches to the next VM with tokens in its bucket to schedule their processes.

When a process switches to the *Ready* state it is queued in the VM's priority queue that corresponds to its current priority. The *CLOCK* and *SYSTASK* are the highest priority processes in *Minix* and *Mhyper* and they are scheduled without considering the bucket state. Tasks, servers and user-space applications are scheduled to consider their priorities, time slices and the tokens reminding in their VM's buckets.

Linux, as a monolithic OSs, handles IRQs through Interrupt Service Routines (ISR) embedded within the kernel that consume CPU cycles in the context of the running process. On *Minix*, IRQs are handled by independent processes called DDTs that have their own memory, state and priority. If an interrupt occurs and it is handled by a DDT within a VM that has its bucket empty, the DDT would not be scheduled until its VM's bucket has enough tokens. That behavior would cause unwanted delays responding to hardware devices. To minimize the likelihood this happens, the TBF was modified considering a threshold of tokens on VM's buckets. The token allocation algorithm inserts tokens into the VM's bucket at a specified rate. User-space processes, servers and tasks, while running, consume tokens (one token per timer tick) from the VM's bucket until the mentioned threshold. If the running process is a task it can still run consuming the remaining tokens, otherwise the scheduler switches to the highest priority ready-to-run process on other VM. The token threshold acts as a reservation of tokens allowing DDTs handle hardware interrupts on VMs with CPU-hungry processes.

Mhyper (as non-SMP *Minix*), has one *IDLE* task (within *VM0*) that will only be scheduled when there is not any process *ready to run* on any VM. This avoids to waste CPU cycles in useless processes (an idle task within each VM) as occurs with the idle tasks of a Guest-OS within VMs in hardware virtualization. This unwanted issue happens because the hypervisor doesn't know anything about the processes running within its VMs.

6.2. Mhyper's Device Driver Tasks

Depending on the virtualization level implemented and the partition of devices, there is a need to modify some stock *Minix* DDTs. In the para-virtualization approach, DDTs don't need to be modified, but it should require considerable changes to *SYSTASK*. An exception is the RAM disk DDT (named ***mem***) that could run within any VM without modifications, and requires tiny changes on *SYSTASK* to supervise memory limits.

Device Driver Level virtualization requires the modification of some *DDTs* to share a device among VMs; i.e. the *TTY* task (on *VM0*) was modified to be a promiscuous task partitioning the system console in multiple virtual terminals or *vty*s (see Figure 5).

A promiscuous task really runs within *VM0*, but it can handle requests coming from processes within other VMs. A *virtual* process representing the *promiscuous* task is installed within the VM using the same *endpoint* number and it will be the destination of the requests.

Another example of a promiscuous task is the hard disk task (*AT_WINI)* of VM0. It allows that each VM could allocate its own physical hard disk. Beforehand,

the stock Minix disk driver was modified to support concurrent client's requests because it only supported requests from a single client (the FS).

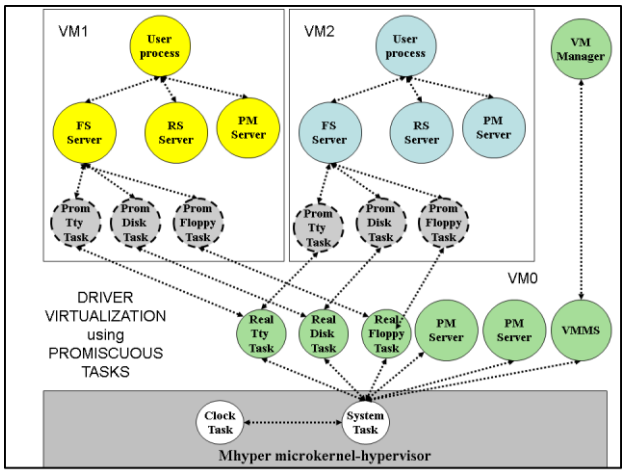


Figure 5. Mhyper with promiscuous drivers.

The *CMOS* task of VM0 that manage the real-time clock (RTC) it is also promiscuous. Some requests are serviced asynchronously as GETTIME/SETTIME that gets and sets the RTC time and date. The *FS* requests *CMOS* on behalf of a user-space process as *readclock* to get the time from the RTC. As this operation could be too slow, it replies *FS* to suspend *readclock*. Therefore, *FS* could still run while the slow operation is completed. Once the CMOS operation completes, it notifies *FS* to request the operation result.

6.3. Mhyper's Servers

New system calls were added to *Minix* to support VMs, but they were not implemented as extensions to any existing server as the *PM* or *FS*. Instead of that, a new server was created called Virtual Machine Manager (*VMM*) server. VMM handles requests related to VMs management. It is the interface between user-space tools (commands) and the *SYSTASK* and the *PM* run on *VM0*. The *VMM* runs on *VM0* and cannot be executed within any other VM.

The ***vmmcmd*** command is used to manage VM operations as creating, loading, starting, ending, etc. VMs. ***vmmcmd*** sends its requests to *VMM* using the added system calls.

The stock version of *Minix* uses the *IS* to present information and statistics about the system resources, processes and statistics in the system console. The *IS* was modified to behave as a *promiscuous* server, allowing it to present information about processes and resources within all VMs. *IS* really runs in *VM0* but it can request information to DDTs, the *PM*, the *FS* and the *RS* of all active VMs through its stubs. As the reader has probably noticed, the modified *IS* makes weaker the isolation property of virtualization, but it must be considered that it is limited to the privileged *VM0*. Another approach to build *IS* through para-virtualization can be made allocating its own needed resources on each VM without interacting with processes of other VMs. This option is more secure and *Mhyper* might provide it, but this approach requires considerable changes on *SYSTASK* to partition the system resources.

6.4. Mhyper's Management tools

As was mentioned earlier, the ***vmmcmd*** command allows the administrator to load, to start, to pause, to resume and to finish VMs. It also allows presenting VM's configuration, status and statistics. ***vmmcmd*** gets the VM's resource configuration parameters (memory size, bucket size, disk device, boot image file, etc.) from a VM's configuration file located in *VM0* but currently, a web based management tools is in the development stage to offer a user interface like commercial products too.

Before starting a VM, the boot image must be loaded in memory. To load the boot image and to create the VM, the ***vmmcmd*** command forks and its child process executes the boot image loader (***loadvmimg***). It allocates the memory space for the new VM and loads the *Minix* boot image from a specified boot image file located in *VM0* into its data segment.

Once the VM was loaded, ***loadvmimg*** remains running in VM0 waiting for requests coming from processes within the VM loaded once it is started. Therefore, ***loadvmimg*** behaves as a Proxy server that represents servers and tasks inside the loaded VM, allowing them to perform operations on servers and drivers inside *VM0* (see Figure 6).

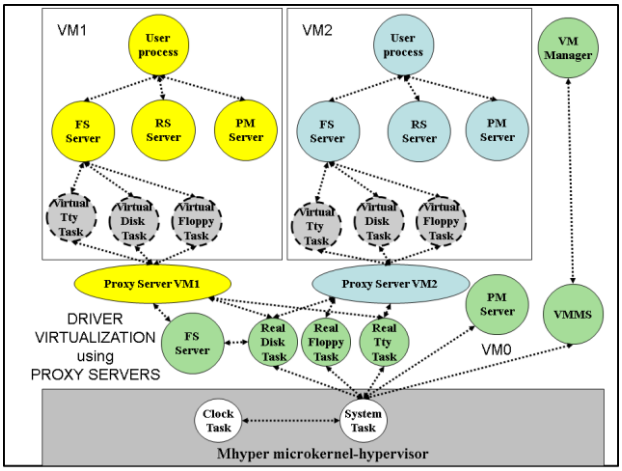


Figure 6. Mhyper with proxy server drivers.

Some critical servers and DDTs need to be loaded at boot-time, others can be loaded at start-up time by a script or later from the prompt using the *service* command that request the RS. The RS forks and the call the *exec()* system call with the path of the server or device driver object program. The *SYSTASK* was modified to load virtual and promiscuous processes within the VM's allocated memory, but they are not really started. Only their process descriptors are created and set within the VM.

To test the Virtual/Proxy model, a device driver that handles old floppy disks was built. A virtual floppy driver task within *VMx* (a *stub*) maps a virtual diskette to a *VM0's* diskette image file. The requests from the FS

within *VMx* to the virtual disk driver on *VMx* (a *stub*) are received by the proxy (***loadvmimg)*** that performs the requested operations on the diskette image file using POSIX standard system calls.

The drawbacks of this device driver model are:

–A single process (*proxy*) must handle requests coming from all Virtual Device Drivers of its represented VM which may be a bottleneck for some workloads.

–It requires another level of message transfers (between Proxy Servers and DDTs or Servers on VM0) but, as the memory address space of *VMx* is "*inside*" the Data Segment of the Proxy Server, it can access *VMx's* process memory address without restriction. This fact allows copying data or making system calls directly between the requester process on *VMx* (as FS) and the task or server process on VM0 (as FS).

To avoid that a process on *VM0* could stop ***loadvmimg*** releasing all its resources and surely hanging the system without finishing its contained VM, the *PM* of *VM0* was modified to prevent that a signal can be sent to a proxy process (using the *kill* system call). The ***loadvmimg*** finishes only once the VM is finished.

7. Mhyper Evaluation

The goal of this work was to run unmodified Minix applications within a VM and on top of Mhyper. An important consideration for the project is the overhead of the extensions in terms of execution time should be comparable with those of native Minix and with a commercial product like VMware Server.

The benchmarks were done on a PC with Intel(R) Core(TM) 2 CPU E7400 @ 2.80GHz with 1 GB of RAM and a Hitachi hard disk of 42 GB. The software used was Minix and Mhyper version 3.1.2a, and VMWare Server 1.0.6.

The following performance benchmarks were chosen because the commands are available on every Minix distribution and they are frequently used:

–***tar***: A ***tar*** of the /usr/src directory with all its files resulting in a .tar file of 60 Mb.

–***compress***: Compress the mentioned .tar file to 28 Mb .tar.Z file.

–***make world***: Builds the complete Mhyper system (libraries, kernel, drivers, servers, etc.).

The testing scenarios to evaluate execution times were:

–*MINIX*: Minix on the bare hardware.

–*MHYPER VM0*: Mhyper's VM0 on the bare hardware with one disk partition allocated to it.

–*MHYPER VM1*: A VM running on Mhyper with another disk partition allocated to it.

–*MHYPER (VM1+VM2):* Two VMs running concurrently on Mhyper with their own disk partition.

–VMWARE MINIX: Minix on VMware with direct access to the physical hard disk on Windows XP.

In the results presented in Figure 7 it becomes evident the following important findings:

1.The imposed overhead as a result of adding the virtualization extensions into Minix causes a 3-5% reduction in performance (Figure 7: 1 vs. 2).

2.The performance of two VMs running concurrently on Mhyper makes evident its fairness on resource scheduling (Figure 7: 4 vs. 5).

3.If a VM runs on Mhyper outperform 2.3 times a VM running Minix on VMware (Figure 7: 3 vs. 6).

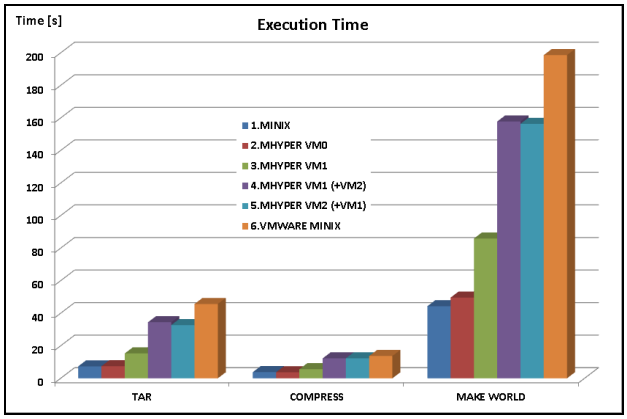


Figure 7. Performance benchmarks results.

The results suggest that the chosen approach and its implementation are correct, even though optimizations were not analyzed yet.

8. Conclusions and Future Works

As the technology progressed, new features were added to OS to support the new facilities. Therefore, it seems more reasonable to incorporate features of a hypervisor within an OS microkernel than modifying a hypervisor to behave as a microkernel.

The research contributions of the project presented in this article are:

–It presents several approaches that meet resource, security and fault isolation virtualization requirements with different degrees of compliance.

–*Minix* is a multi-layer OS that allows different approaches to add virtualization support, according to the needs and requirements. Each level of virtualization presents a trade-off between the isolation properties of virtualization and the reuse of drivers and servers. *Mhyper* takes advantage of those *Minix* features and presents how to use them for several virtualization models.

–*Mhyper* also takes advantage of the reliability features of *Minix*, but it improves them as a consequence of the added virtualization features as security, fault and resource isolation.

–The performance of applications running within a VM on *Mhyper* is better than running them within a VM on VMware, and might be improved with future optimizations.

–As *Minix* is widely used in OS design and implementation courses, the added extensions allow

instructors to include virtualization topics on laboratory assignments.

Mhyper proves that microkernel OS technology and particularly *Minix* is well adapted to include new features as the support for several virtualization approaches.

9. Acknowledgments

The author gratefully acknowledges helps received from Antoine Leca and PhD. Silvio Gonnet.

References

[1] Galley S., "¨*PDP-10 Virtual Machines*". In Proc. ACM SIGARCH-SIGOPS Workshop on Virtual Computer Systems, MA, 1969.

[2] Popek G.; Goldberg R., "*Formal Requirements for Virtualizable Third Generation Architectures*". Communications of the ACM 17 (7), 1974.

[3] M. Acceta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young. "*Mach: A New Kernel Foundation for Unix Development*", Usenix Conference, pp. 93-112, 1986.

[4] Gien M., Grob L. , "*Micro-kernel Architecture Key to Modern Operating Systems Design*", 1990.

[5] Hohmuth M.; Peter M., et al. "*Reducing TCB size by using untrusted components: small kernels versus virtual-machine monitors*"; EW 11; 2004.

[6] Ford B., Hibler M. et al, "*Microkernels meet recursive virtual machines*"; Second Symposium on OSDI, 1996.

[7] Biemueller S.; Dannowski U., "*L4-Based Real Virtual Machines - An API Proposal*". In Proceedingsof the MIKES 2007.

[8] Tanenbaum A., et al, "*Minix 3: Status Report and Current Research*". ;login: The USENIX Magazine. 2010.

[9] Liedtke j.; "*Toward Real Microkernels*"; Communications of the ACM Volume 39 Issue 9, Sept. 1996.

[10] H. Hartig. "*Security Architectures Revisited*". 10th Workshop on ACM SIGOPS European Workshop, EW 10, pages 16{23, 2002}.

[11] Agarwal A.; Levy M., "*Thousand-core chips: the kill rule for multicore*". 44th Annual Conference on Design Automation, 2007.

[12] Baumann et. al, "*The Multikernel: a new OS architecture for scalable multicore systems*", 22nd Symposium on Operating Systems Principles (2009).

[13] Wentzlaff D., at al, "*An operating system for multicore and clouds: mechanisms and implementation*"; Proceedings of the 1st ACM Symposium on Cloud Computing 2010.

[14] Linnenbank N.; "*Implementing Minix on the Single Chip Cloud Computer*"; Master Thesis, Vrije Universiteit, 2011.

[15] Herder J. "*Towards, a True Microkernel Operating System*". Vrije Universiteit. Master degree thesis 2005.

[16] Herder J.; Gras B., Tanenbaum A., "*Modular system programming in Minix 3*", Login:. April 2006.

[17] "*Protection matrixes in Minix*". http://wiki.erights.org/wiki/Protection_matrixes_in_Minix.

[18] Jorrit N. Herder, Herbert Bos, Ben Gras, Philip Homburg, and Andrew S. Tanenbaum, "*Fault Isolation for Device Drivers*", DSN 2009, Estoril, Lisbon, Portugal, 2009.

[19] Hand S.; Warfeld A.; et al, "*Are virtual machine monitors microkernels done right?"* 2005.

[20] Heiser G., Uhlig V.; and LeVasseur J., "*Are virtual machine monitors microkernels done right?*" NICTA Technical Report PA005103, 2005.

[21] Armand, F. and Gien, M., "*A Practical Look at Micro-Kernels and Virtual Machine Monitors*"; 6th IEEE CCCN Conference, 2009.

[22] U. Steinberg and B. Kauer, "*NOVA: A microhypervisor-based secure virtualization architecture*", In 5th EuroSys Conf., Paris, France, Apr 2010.

[23] G. Heiser and B. Leslie, "*The OKL4 microvisor: Convergence point of microkernels and hypervisors*", ACM 978-1-4503-0195-4/10/08, 2010.

[24] Steffen Liebergeld, "*Lightweight Virtualization on Microkernel-based Systems*", January 2010.

[25] Brygier J.; Fuchsen R.; H Blasum, "*PikeOS: Safe and secure virtualization in a separation microkernel*", SYSGO White paper, 2009.

[26] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "*KVM: The Linux Virtual Machine Monitor*", Linux Symposium, 2007.

[27] B. Ford, M. Hibler, J. Lepreau, P. Tullmann, G. Back, and S. Clawson. "*Microkernels meet recursive virtual machines*". OSDI, October 1996.

[28] Ning Li; Kinebuchi, Y.; Mitake, H.; Shimada, H.; Tsung-Han Lin; Nakajima, T., "*A Light-Weighted Virtualization Layer for Multicore Processor-Based Rich Functional Embedded Systems*", ISORC, April 2012.

[29] C. van Schaik and G. Heiser, "*High-performance microkernels and virtualization on arm and segmented architectures*"; 1st International Workshop on Microkernels for Embedded Systems , 2007.

[30] Dike, Jeff. "*A user-mode port of the Linux kernel*"; 4th Annual Linux Showcase & Conference, 2000.

[31] Pessolani, Pablo A. "*Modifying the Behaviour of Minix System Calls through the Redirection of Messages*", CACIC 2011.

[32] Yang Yu and Fanglu Guo and Susanta Nanda and Lap-chung Lam and Tzi-cker Chiueh, "*A feather-weight virtual machine for windows applications*", VEE , 2006.

[33] Blas, M. J.; Prola, G.; Sarli, J., "*Diseño e Implementación de un File Server Alternativo para Minix*", CoNaIISI 2013.

[34] David C. van Moolenbroek; Raja Appuswamy; Andrew S. Tanenbaum, "*Towards a Flexible, Lightweight Virtualization Alternative*", System and Storage Conference SYSTOR 2014.

[35] Pessolani P.; González C., "*Enhancing Minix 3 Input/Output Performance Using a Virtual Machine Approach*", WARSO, CACIC 2010.

[36] Pessolani P., "*MINIX4RT: A Real-Time Operating System Based on Minix*", Master Thesis, UNLP, 2006.

[37] YS Vivek, L Bhuvan, G Ashrith, and DSVN Aditya. "*Realtime Minix"*, 2008.

[38] Banco Zandbergen, "*A more Real-time Minix 3*", October 6, 2009.

[39] Stephen Soltesz and Herbert Pötzl and Marc E. Fiuczynski and Andy Bavier and Larry Peterson, "*Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors*", EuroSys 2007.

Optimizando JavaScript a través de la reescritura de código

Julieta Alvarez, Alexis Ferreyra, Ricardo Medel, Emanuel Ravera
Departamento Ingeniería en Sistemas de Información
Universidad Tecnológica Nacional – Facultad Regional Córdoba
{juli.cba, alexis.ferreyra, ricardo.h.medel, ravera.emanuel}@gmail.com

Abstract

Con el aumento en la complejidad de las aplicaciones para dispositivos móviles y web la performance de los programas escritos en JavaScript ha captado la atención de investigadores y desarrolladores de compiladores y runtimes. Los enfoques utilizados hasta el momento tienen limitaciones que nos proponemos superar por medio de la reescritura de código en tiempo de compilación, reemplazando patrones de código ineficientes por otros de conocida eficiencia.

En este artículo reportamos los avances iniciales de nuestro trabajo. Hemos descubierto y analizado empíricamente tres patrones que nos permiten mejorar la performance hasta 167 veces en ciertos casos. Luego se diseñó e implementó PumaScript, un dialecto de JavaScript que provee las extensiones necesarias para realizar introspección y reescritura de código a través de metafunciones. Los resultados obtenidos nos permiten ser optimistas respecto de la utilidad de nuestro enfoque.

1. Introducción

La gran expansión del uso de la web y, en particular, de los runtimes web para distintos tipos de dispositivos en los últimos años, ha producido, entre otros efectos, el crecimiento de la complejidad de las aplicaciones. En este contexto, la performance de JavaScript ha captado un gran interés por parte de la industria y la comunidad científica.

Se ha trabajado mucho en la mejora de la performance de JavaScript. La mayoría del trabajo realizado se enfocó en mejorar los runtimes, en la mejora de los compiladores "Just in Time" (JIT por sus siglas en inglés). Por ejemplo, el motor Google Chrome V8 [1]. Aunque éste no fue el único enfoque, ya que Mozilla asm.js [2] enfrentó el problema implementando un subconjunto de JavaScript que luego es interpretado de manera eficiente. Otros ejemplos, como el caso de Intel, incluyen el agregado de instrucciones SIMD [3] o la ejecución en paralelo de threads dentro del runtime.

La línea de trabajo elegida por nuestro grupo de investigación y desarrollo ha sido la de probar la factibilidad de mejorar la performance de JavaScript reescribiendo el código en tiempo de compilación. La performance en JavaScript es muy sensible a la manera en que el desarrollador escribe su código. Por ejemplo, usando la sentencia "for in" para iterar un arreglo (array) es normalmente 50 veces más lento que utilizar la notación C estándar para iterar el arreglo.

Adicionalmente, similares situaciones se dan cuando se utilizan diferentes API (*Application Programming Interface*) es un conjunto de funciones o métodos que ciertas bibliotecas ofrecen para ser utilizado por otro software, ofreciendo una capa de abstracción.) propias del runtime. Por ejemplo, usar la API "document.getElementByTagName" para encontrar los elementos del DOM es 100 veces más rápido que utilizar la API "document.querySelectorAll" y se obtienen los mismos resultados.

Creemos que existe una oportunidad para atacar ciertos casos que escapan al alcance de las soluciones propuestas previamente. Nuestra solución apunta a optimizar el comportamiento de los compiladores Just in Time (JIT) utilizado introspección del programa y técnicas de reescritura de código.

Para poner a prueba esta hipótesis identificamos algunos patrones que son potenciales candidatos para las optimizaciones. A partir de los mismos creamos benchmarks básicos, de manera que se pueda probar que esos patrones identificados de manera heurística realmente pueden generar mejoras en la performance.

Luego de haber comprobado, a través de estos benchmarks, que se podían generar importantes mejoras de performance se continuó con la implementación y el diseño de PumaScript [4], un dialecto simple de JavaScript que provee las extensiones necesarias para brindar la capacidad de realizar introspección y reescritura de código. Usando este lenguaje se desarrollaron las metafunciones que permiten la identificación y reescritura automática de los patrones de su forma más lenta a una manera más eficiente.

2. Identificación de patrones en JavaScript y benchmarks

En la etapa inicial de nuestro proyecto identificamos tres patrones de código que pueden ser escritos de una manera diferente y obtener el mismo resultado pero en forma más eficiente. La **<Tabla 1>** muestra la lista de patrones identificados, incluyendo un nombre identificatorio, el código original y el código optimizado.

	Nombre	Código Original	Código Optimizado

1	For in vs For notación C	var array = [1,2,3]
		var i=0;
		for(i in array){
		array[i] +=1;
		}
			var array = [1,2,3]
			for(var i=0; i< array.length; i++){
			array[i] +=1;
			}

2	Selector jQuery ID vs Selector ID Nativo	$("#test");	$(document.getElementById("test"));
3	parseInt vs Bitwise	var number = Math.random(); parseInt(number);	var number = Math.random(); number | 0;

Tabla 1. Patrones de JavaScript.

Luego de haber identificado estos patrones, creamos y ejecutamos los benchmarks para poder medir la diferencia de performance entre el código original y el optimizado. A fin de obtener valores significativos, cada prueba consistió en 10.000 iteraciones de los benchmarks simples. Los benchmarks fueron ejecutados sobre el siguiente hardware:

- **PC:** Core-i5-3320M, 4GB de RAM.
- **Tablet:** Intel Atom Z2560 dual-core 1.6GHz, 2GB RAM.

La <**Tabla 2.a**> y <**Tabla 2.b**> muestra las mejoras de performance obtenidas en la ejecución de los benchmarks en su versión normal y optimizada para los diferentes tipos de hardware y web runtimes.

N°	PC - Chrome- v 36.0.1985.143	PC - Mozilla- v30.0
1	55,54x	167,60x
2	2,15x	1,49x
3	12,91x	33,30x

Tabla 2.a. Mejoras de Performance PC.

N°	Tablet Android - Browser Nativo - 4.4.2	Tablet Android - Chrome - 4.4.2 - v 36.0.1985.135
1	27,96x	25,26x
2	1,98x	1,79x
3	11,24x	4,75x

Tabla 2.b. Mejoras de Performance Tablet.

Como se puede ver en las tablas expuestas las mejoras de performance varían notablemente. Nótese que usando el runtime Mozilla en una PC se obtiene una mejora de apenas 1,49 veces en el patrón 2, pero una mejora de 167,60 veces en el patrón 1.

3. Diseño e implementación de PumaScript

PumaScript es un superconjunto de JavaScript que agrega a este lenguaje la infraestructura necesaria para la reescritura e introspección de código. La funcionalidad agregada son metafunciones que trabajan de manera muy similar a un sistema de macros para lenguajes tipados. Como cualquier otro sistema de macros, las metafunciones pueden desagregar expresiones de la función llamadora en la misma línea.

Una metafunción toma como parámetro el árbol de sintaxis decorado de los argumentos actuales y debe retornar un árbol de sintaxis abstracto de destino que se usa como reemplazo en la función llamadora. Las metafunciones de PumaScript pueden decidir no expandir cierta ocurrencia de la función llamadora retornando "null" en lugar del árbol de sintaxis abstracta.

Una metafunción, además, puede ejecutar otras funciones normales o más metafunciones.

Adicionalmente, todas las metafunciones tienen acceso a funciones intrínsecas especiales que le permiten realizar la introspección y la reescritura de código de cualquier porción del árbol de sintaxis abstracta.

La <**Figura 1**> ilustra una metafunción simple que reescribe las invocaciones a la función "sum" en una expresión de suma directa.

```
/** @meta */
function sum(a, b)
{
    return pumaAst( $a +  $b);
}

// La siguiente invocación a "sum" será reescrita a la expresión "5 + 6"
sum(5, 6);
```

Figura 1. Metafunción Simple y su invocación.

En la <**Figura 2**> se observa un diagrama de alto nivel del flujo de procesamiento y ejecución de

PumaScript. En primer lugar se utiliza la librería Esprima [5] para realizar el parsing de la sintaxis de JavaScript. Luego el runtime de PumaScript se utiliza para ejecutar el árbol de sintaxis abstracta siguiendo la semántica de JavaScript. Dentro de esta ejecución se tienen en cuenta las reglas adicionales y la nueva semántica agregada por PumaScript. Una vez que el programa se ejecuta, PumaScript es el encargado de descartar las metafunciones y el árbol de sintaxis resultante es procesado por Escodegen [6] que imprime el programa en JavaScript estándar.

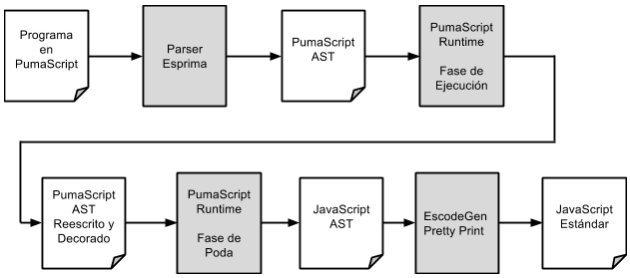


Figura 2. Flujo de procesamiento y ejecución de un programa PumaScript.

4. Metafunciones de PumaScript

Las metafunciones de PumaScript se escriben como funciones de JavaScript agregándoles el comentario *@meta* antes de la declaración de función. Este método es utilizado para evitar la introducción de nuevos requerimientos de sintaxis y permite que PumaScript sea completamente compatible con JavaScript.

Las metafunciones se comportan de manera similar a funciones de JavaScript, pero con tres diferencias que se enumeran a continuación.

a. Todos los parámetros en una metafunción se evaluarán en referencia al árbol de sintaxis decorado al momento de la ejecución. Por ejemplo, cuando se llama a la metafunción "foo(a, b)" con los argumentos "2 * x" y "3 * z" para los parámetros a y b respectivamente, éstos van a tomar los valores del árbol de sintaxis al momento de la ejecución.

b. Todas las metafunciones deben retornar un árbol de sintaxis válido o el valor "null". Si se retorna el valor "null" entonces la función llamadora no será reescrita. En cambio, si el valor retornado es un árbol de sintaxis válido entonces la expresión de la función llamadora será reemplazada por éste.

c. Todas las metafunciones tienen acceso a funciones y objetos intrínsecos que pueden ser utilizados para realizar la introspección de código o la reescritura de cualquier porción del programa. Algunos ejemplos de éstas incluyen funciones tales como "pumaAst", que genera un árbol de sintaxis correcto, o "pumaFindByType", para realizar búsquedas dentro del árbol.

5. Reescribiendo Código para que sea performante

La **<Figura 3.a>** muestra la implementación de una metafunción que reescribe el selector por Id de JQuery [7] en la API nativa "document.getElementById".

```
1  /* @meta */
2  function $(valueExp){
3    var regex = /^#\b[a-zA-Z0-9_]+\b$/;
4    var argValue = {};
5
6    if(valueExp.type === 'Literal' && regex.test(valueExp.value)){
7      valueExp.value = valueExp.value.substring(1);
8      return pumaAst($(document.getElementById($valueExp)));
9    }
10   else if(OPTIMISTIC_REWRITE){
11     argValue = evalPumaAst(valueExp).value;
12     if(regex.test(argValue)){
13       console.log('WARNING: Reescritura optimista en la linea: (' + valueExp.loc
14       return pumaAst($(document.getElementById($valueExp.substring(1))));
15     }
16   }
17   return null;
18 }
```

Figura 3.a. Metafunción para reescribir selectores JQuery por Id.

En la **<Figura 3.a>** puede verse que en la línea 6 se valida que el argumento sea un literal y se realiza el test sobre la expresión regular usada para buscar selectores por Id. Luego se remueve el carácter "#" del comienzo del argumento y se devuelve el nuevo árbol de sintaxis usando la API "document.getElementById" pasándole como argumento el string modificado que se recibió.

La función intrínseca "pumaAst" es utilizada crear el árbol de sintaxis que se devuelve a partir del template en el que las variables locales toman los valores actuales.

La **<Figura 3.b>** muestra diferentes tipos de llamadas a la metafunción. Si se ejecuta la rama del literal entonces se puede estar seguro que la reescritura es correcta. En cambio si se continúa por el camino de la sentencia "else" puede ocurrir el caso en que no se esté completamente seguro de lo que se va a reescribir, porque por ejemplo la variable "element_id" es un argumento dentro de una función.

El código de la metafunción utiliza la variable bandera "OPTIMISTIC_REWRITE" para habilitar la funcionalidad en donde se analiza que al evaluar la expresión en ese contexto se obtenga un selector por Id.

```
1 // Llamada con Literal
2 var myElement = $('#Element_Id_1');
3
4 // Llamada con una expresión no trivial
5 var element_id = "5";
6 var myOtherElement = $('#Element_Id_' + element_id);
7
8 // Llamada que no se incluye selector por Id
9 var myOtherClassElements = $('.Class_Id_' + element_id);
```

Figura 3.b. Diferentes maneras de llamar al selector de JQuery para Id.

La línea 11 de la **<Figura 3.a>** utiliza la función intrínseca "evalPumaAst" que evalúa cualquier árbol de sintaxis válido en el contexto de ejecución. Para el caso de la línea 6 de la **<Figura 3.b>** parecería seguro reescribir la expresión de JQuery [7] en su API más eficiente.

Por último, la línea 9 de la **<Figura 3.b>** muestra un caso similar, donde claramente se puede observar que el selector no concuerda con un selector por Id. Esta metafunción tiene la capacidad de poder identificar este caso de esta manera porque sin importar qué valor tenga la variable "element_id" se va a generar un selector inválido.

6. Trabajos Relacionados

Compiladores Just in Time. El porcentaje más alto de mejoras de performance en el lenguaje JavaScript ha venido de la mano de la creación de runtimes más avanzados. Anteriormente existían runtimes que interpretaban el árbol de sintaxis, tales como Mozilla Rhino [8] y los motores de Microsoft Internet Explorer. Estos dieron paso a una generación más avanzada, tales como el motor Google Chrome V8 [1], el Mozilla SpiderMonkey [9, 10] y el renovado motor Microsoft Chakra [11].

Todos estos compiladores JIT han basado sus motores en la reutilización de técnicas que fueron aplicadas a otros runtimes, tomando como caso más notable al compilador Java Hotspot.

A pesar de que la introducción de los compiladores JIT ha mejorado la performance de JavaScript en un orden de magnitud, como se puede ver en este artículo aún existen casos que escapan a estas mejoras, tal como la reescritura de la sentencia "for in" en una "for" en modo C. A todo lo anteriormente mencionado se le agrega que los compiladores JIT no son buenos en optimizaciones relativas al uso de API, tal y como se demuestra en los ejemplos de JQuery vs. Nativo expuestos en la sección *"Identificación de patrones y benchmarks" de este trabajo*.

Minificadores de Código y Pre-Compiladores. Los minificadores de código han sido utilizados por la comunidad de JavaScript desde sus inicios. Algunos ejemplos simples como JSMin [12] y JSZap [13] proveen optimizaciones para el uso de menores recursos de red pero no proveen mejoras medibles en performance.

Existen también precompiladores más elaborados, como el Google Closure Compiler [14], el cual es capaz de realizar ciertas optimizaciones, tales como la remoción de variables no utilizadas. Pero esas optimizaciones son provistas a fin de reducir el tamaño del código fuente, no de mejorar la performance.

7. Conclusiones y Trabajos Futuros

En este trabajo pudimos demostrar fácticamente que los benchmarks seleccionados son mejorados en su performance a través de la re-escritura automática de código. Esto nos permite seguir investigando nuevos benchmarks que permitan la mejora de un programa completo JavaScript, superando las limitaciones que hoy tienen los compiladores JIT y minificadores de código.

El trabajo futuro se centrará en aplicar estas mejoras de manera automática sobre software real. Esto se logrará implementando estas etapas sobre el código del desarrollador y antes de que llegue a producción en un sistema de integración continua. Consideramos de gran importancia tener una mayor cantidad de benchmarks, a fin de poder generar una suite de mejora de performance abarcadora.

Dado que los casos de optimización presentados, como la reescritura de la sentencia "for in" en "for", forman parte del bagaje básico de programadores JavaScript, creemos que existen altas posibilidad de transferir parte de los beneficios de performance detectados en la reescritura de nuestros benchmarks sintéticos al software real. Sin embargo, aún debemos demostrar que aplicar las transformaciones de reescritura presentadas aquí representa posibilidades significativas de optimización en un porcentaje importante sobre software real.

Adicionalmente, en el ámbito de la mejora de performance de lenguajes de alto nivel, no se ha demostrado que exista una solución única que triunfe sobre todas las demás. Nuestra técnica, de mejora de performance basada en reescritura de código antes de la puesta en producción del software, es complementaria al resto de las técnicas utilizadas como minificadores de código y técnicas de compilación JIT en runtimes.

Finalmente, por su cercanía al código escrito por los desarrolladores, las técnicas descritas en este trabajo y el lenguaje PumaScript pueden tener aplicaciones en procesos de ingeniería de software ayudando a los desarrolladores a identificar malas prácticas o potenciales lugares dentro del código donde mejorar la performance.

8. Referencias

[1] v8. Google Inc. *La máquina virtual V8 de JavaScript*. https://code.google.com/p/v8 , 2013.

[2] David Herman, Luke Wagner, and Alon Zakai. *Especificación de asm.js*. http://asmjs.org/spec/latest/, 2013.

[3] *SIMD en JavaScript*. Intel Co., https://software.intel.com/sites/billboard/article/simd-javascript-faster-html5-apps, 2014.

[4] Repositorio y Wiki en GitHub. *PumaScript.* https://github.com/emravera/puma, 2014.

[5] *Documentación sobre Esprima.* http://esprima.org/doc/index.html, 2014.

[6] Documentación sobre Escodegen. Documentación y código. https://github.com/Constellation/escodegen, 2014.

[7] Documentación de la API de JQuery. http://api.jquery.com/, 2014.

[8] Documentación de Mozilla Rhino JavaScript Runtime. Mozilla Foundation. https://developer.mozilla.org/en-US/docs/Rhino_documentation, 2014

[9] Documentación de Mozilla SpiderMonkey. Mozilla Foundation. https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey, 2014.

[10] Mason Chang, Edwin Smith , Rick Reitmaier , Michael Bebenita , Andreas Gal , Christian Wimmer , Brendan Eich , Michael Franz, *Tracing for web 3.0: trace compilation for the next generation web applications*, Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, March 11-13, 2009, Washington, DC, USA

[11] Microsoft Chakra JavaScript Runtime. Advances in JavaSccript Performance in IE 10 http://blogs.msdn.com/b/ie/archive/2012/06/13/advances-in-javascript-performance-in-ie10-and-windows-8.aspx

[12] D. Crockford. JSMin: The JavaScript minifier. http://www.crockford.com/javascript/jsmin.html.

[13] Martin Burtscher , Benjamin Livshits , Benjamin G. Zorn , Gaurav Sinha, *JSZap: compressing JavaScript code*, Proceedings of the 2010 USENIX conference on Web application development, p.4-4, June 23-24, 2010, Boston, MA

[14] Google closure compiler. https://developers.google.com/closure/compiler.

Despliegue de una arquitectura de Cloud Computing híbrida Open Source

Maria A. Murazzo, Nelson R. Rodríguez, Susana B. Chávez, Francisca A. Valenzuela, Adriana E. Martin, Miguel J. Guevara

Universidad Nacional de San Juan – Facultad de Ciencias Exactas, Físicas y Naturales – Departamento e Instituto de Informática
Complejo Universitario Islas Malvinas (CUIM) – Rivadavia – San Juan – Argentina
marite@unsj-cuim.edu.ar - nelson@iinfo.unsj.edu.ar - schavez@iinfo.unsj.edu.ar - franciscaadriana.valenzuela@gmail.com - arianamartinsj@gmail.com – migueljoseguevaratencio@gmail.com

Abstract

Cloud Computing está transformando los modos tradicionales de cómo las empresas utilizan y adquieren los recursos de Information Technology (IT). Luego de un auge inicial de los Cloud Públicos las empresas han comenzado a montar Cloud híbridos que ofrecen las ventajas de Cloud Computing sumado a la privacidad de los datos que consideren estratégicos. Una solución de Cloud híbrido permite la integración de ambos sistemas. Las empresas líderes en soluciones cloud han entendido esta evolución y han comenzado a ofrecer soluciones hibridas. Más aun, muchas de esas empresas están dando el siguiente paso ofreciendo soluciones basadas en estándares Open Source que permitan alto grado de interoperabilidad y portabilidad.

1. Introducción

El fenómeno emergente de Cloud Computing representa un cambio fundamental en la forma en que los servicios de tecnología de la información son creados, desarrollados, desplegados, escalados, actualizados, mantenidos y facturados. La Informática de hoy refleja una paradoja: por una parte, las computadoras resultan cada vez más poderosas y el coste por unidad disminuye, sin embargo la computación resulta más pervasiva dentro de la organización y se incrementa la complejidad de administrar toda la infraestructura de las arquitecturas, los datos y el software [1].

Los sistemas basados en Cloud Computing han tomado relevancia en los últimos años, basta con comparar por medio de Google Treads las tendencias de búsqueda de términos y se puede observar que "Cloud Computing" ha sobresalido exponencialmente.

Cloud Computing es un paradigma que permite el cómputo a gran escala mediante la abstracción de la complejidad de la plataforma de desarrollo y la infraestructura sobre la que se monta, esto se logra a través de la encapsulación del hardware, la compartición de recursos y la virtualización.

De esta manera es posible crear una plataforma agnóstica y escalable para el desarrollo de aplicaciones. Como característica principal es la capacidad de proporcionar al usuario diferentes modelos que se ajusten a los requerimientos de su aplicación, lo cual se realiza mediante interfaces de simple interacción.

Algunas empresas han montado sus propios Cloud privados para la gestión y administración de sus recursos, pero han sido ineficientes al momento de la escalabilidad exigiéndoles la migración a una arquitectura de Cloud hibrida.

En este sentido la consultora Gartner (www.gartner.com), en su evento *Gartner Symposium/ITxpo 2013*, resaltan 10 tecnologías en las que confluyen aspectos relacionados con lo social, la movilidad, el cloud computing y la información y; considera que 2014 será decisivo en la migración del cloud privado al cloud híbrido.

Por su parte, la consultora IDC (www.idc.com) prevé que el gasto global en TI crezca un 5% en 2014 en comparación al 2013, alcanzando una inversión de 2,140 billones de dólares. El gasto provendrá principalmente de las tecnologías de la tercera plataforma: movilidad, redes sociales, cloud computing y Big Data, las cuales crecerán un 15% cada año y abarcarán el 89% del crecimiento en el gasto de TI. En particular, posiciona al Cloud Computing como la segunda tecnología más estratégica para las organizaciones y los usuarios finales.

Con respecto al tipo de cloud (público, privado o híbrido), no son pocos los que apuntan a una solución híbrida como el modelo cloud de mayor potencial en el entorno organizacional. Gartner e IDC aseguran que el 2014 será el año de la cloud híbrida, y la señalan como la fórmula más atractiva para mejorar y agilizar los procesos de negocio, al mismo tiempo que plantea grandes retos para los proveedores de soluciones y los departamentos o socios TI de las empresas.

Prueba de ello son los recientes movimientos de los principales proveedores de cloud privada como IBM SmartCloud o VMware, que ofrecen nuevas soluciones y adaptan sus productos para la integración de cloud públicas y privadas allanando el camino a las organizaciones en la adopción de una solución híbrida.

De acuerdo con Jim Whitehurst, Presidente y CEO de Red Hat, 2014 va a ser el año en el cual el Cloud será híbrido, debido a que las implementaciones de infraestructuras "tradicionales" están perdiendo terreno frente a las implementaciones híbridas, que consisten en la interoperabilidad de las soluciones de Cloud con la infraestructura que el cliente ya posee.

Por otra parte, los productos de código abierto continuarán desarrollándose y reduciendo la participación del mercado de los productos propietarios, por lo que en 2014 habrá más opciones de soluciones abiertas e interoperables para la empresa.

En función de todo lo analizado, no caben dudas que es necesario investigar e implementar tecnologías estratégicas para cualquier organización, que desee posicionarse como líder dentro del mercado vertical donde brinde sus servicios o productos. Es por ello, que el presente trabajo brinda las pautas necesarias para instalar, configurar, gestionar y poner a punto una Arquitectura de Cloud Hibrida Open Source.

2. Marco Teórico

Internet usualmente se visualiza y conceptualiza como una gran nube (cloud) donde todo está conectado y donde se suministran todos los servicios requeridos. A este esquema de trabajo se lo denomina Cloud Computing, el cual permite un aprovisionamiento de recursos en forma escalable, elástica y bajo demanda y potenciado con tecnologías de virtualización [2].

Atendiendo a la definición dada por el NIST (National Institute of Standards and Technology) [3], cloud computing es un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio.

Según IBM [4], Cloud Computing es un modelo de aprovisionamiento rápido de recursos IT que potencia la prestación de servicios, facilitando la operativa del usuario final y del prestador del servicio. Además, todo ello se realiza de manera fiable y segura, con una escalabilidad elástica que es capaz de atender fuertes cambios en la demanda no previsibles a priori, sin que esto suponga apenas un incremento en los costos de gestión.

La característica básica de este modelo es que los recursos y servicios informáticos como, infraestructura, plataforma y aplicaciones, son ofrecidos y consumidos como servicios a través de Internet sin que los usuarios tengan que tener ningún conocimiento de lo que sucede detrás. Cloud Computing es un esquema que a veces se expresa como Everything as a Service [5].

El NIST clasifica los modelos de la computación en nube en dos grandes categorías: **Modelos de servicio**, que se refieren a los servicios específicos a los que se puede acceder en una plataforma de computación en la nube (software-as-a-service o SaaS, platform-as-a-service o PaaS e infrastructure-as-a-service o IaaS); y el **Modelos de despliegue**, que se refieren a la posición (localización) y administración (gestión) de la infraestructura de la nube (Pública, Privada, e Híbrida).

En el **Modelo de Servicio**, el **IaaS** ofrece una plataforma de virtualización donde se le ofrezca a los clientes recursos computacionales para que puedan desplegar y ejecutar software.

El **PaaS**, brinda a los clientes la capacidad de desplegar sus aplicaciones en la infraestructura del cloud, utilizando diferentes lenguajes y herramientas de programación que el proveedor del servicio soporte.

Con respecto al **SaaS**, llamado CloudApp, proporciona a los clientes el uso de aplicaciones del proveedor, que se ejecutan en su infraestructura. El usuario no se preocupa donde está instalado el software, qué tipo de sistema operativo utiliza o el lenguaje en el que cada aplicación está escrita.

Con respecto al **Modelo de Despliegue**, en el **Modelo Público** el proveedor de los servicios de Cloud Computing es dueño de la infraestructura física y pone a disposición del cliente los servicios del cloud a través de

Internet.

Por su parte, el **Modelo Privado**, es una emulación del público, pero en una red privada, ya que ofrece los mismos servicios que un Cloud público con la ventaja de que el usuario cuenta con sus propios recursos, lo que le permite tener el control de seguridad y calidad de servicio sobre ellos.

Por último, en el **Modelo Híbrido**, se combinan los modelos público y privado, siendo su principal ventaja permitir aumentar la capacidad del cloud privado con los recursos del cloud público para poder mantener niveles de servicio adecuados, frente a rápidas fluctuaciones de carga de trabajo.

3. Arquitectura de Cloud Computing

La arquitectura de cloud computing es similar a la arquitectura de red, desde un nivel físico a un nivel aplicación debido a que se usan protocolos similares a los que se usan en internet.

La arquitectura de cloud computing se la puede representar como un conjunto de capas que se encuentran acopladas para lograr darle funcionalidad al sistema [6]. La misma presenta las siguientes capas, de arriba hacia abajo:

- Aplicación: incluye servicios basados en web y SaaS.
- Plataforma: incluye componentes de aplicación como servicio.
- Infraestructura: incluye software de plataforma como servicio.
- Virtualización: incluye infraestructura virtual como un servicio
- Recursos físicos: incluye elementos como servidores, almacenamiento y red.

Esta arquitectura puede ser montada en cualquiera de los Modelos de despliegue, sin embargo, la mejor opción para cualquier organización en la actualidad es poder implementar una *Arquitectura Híbrida* [7] que permita la combinación de recursos procedentes de infraestructuras físicas, virtuales y de cloud pública y privada.

Es importante destacar que una arquitectura híbrida significa implementar una infraestructura fluida, interconectada e interoperable basada en estándares, capaz de aprovechar la innovación, con independencia de dónde provenga, que le permita mantener el control de los recursos y de su destino. Las características que hacen a esta solución híbrida tan atractiva para cualquier organización son: integración con los sistemas existentes, escalabilidad, personalización en función de las necesidades particulares de la organización, seguridad, orquestación y automatización.

La interoperabilidad entre cloud pública y privada debe estar perfectamente sincronizado, también con el entorno local, de forma que ambas cloud puedan operar conjuntamente las soluciones que están en la cloud privada y las que están en la cloud pública.

El cloud híbrido implica que tanto el cloud privado corporativo como el cloud público utilizado trabajen de forma conjunta, automáticamente, para adaptarse a lo que la organización necesite en cada momento. Además implica no tener que pensar en gastos de capitalización, sino en un abono mensual, y en poder expandir el cloud privado si ésta llega a sus límites.

En un cloud híbrido, los entornos de TI fuera y dentro de las instalaciones no solo están conectados e integrados con una gestión común; también están diseñados para ejecutar las aplicaciones existentes y nuevas, de la misma forma. Con un cloud híbrido, las organizaciones pueden aprovechar un modelo de gestión, organización, conexión en red y seguridad común para dentro y fuera de las instalaciones [8].

Un aspecto muy importante a la hora de instalar una Arquitectura Híbrida, es tener en cuenta que se va a contar con recursos procedentes de infraestructuras físicas, virtuales y de cloud público y privado que en la mayoría de los casos serán heterogéneos.

Por lo tanto se debe incorporar esta diversidad de tecnologías sin dependencia del proveedor y con una interoperabilidad y portabilidad plena. De lo contrario, los beneficios del cloud como eficiencia, agilidad y productividad, se verán afectadas por la carga de gestionar la interoperabilidad entre los diferentes recursos.

En resumen, se puede decir que una Arquitectura Cloud Híbrida Open Source, significa “montar” una infraestructura fluida, interconectada e interoperable basada en estándares, capaz de aprovechar la innovación, con independencia de dónde provenga, que le permita mantener el control de su TI y de su destino.

4. Herramientas Seleccionadas

En los últimos años el número de soluciones open source dedicadas a la administración y gestión de entornos de Cloud Computing ha crecido exponencialmente, grandes compañías como Citrix, Red Hat, IBM, RackSpace, HP, Cisco, entre otras, están apostando fuertemente por este tipo de soluciones.

Un licenciamiento gratuito permite reducir los costos de la infraestructura. Pero este hecho debe estar balanceado con los costes de soporte y de desarrollo para personalizar el código de la solución. El licenciamiento en open source normalmente es menos complicado que en soluciones propietarias.

Adicionalmente, las soluciones open source permiten tener un control mayor sobre las fases de pruebas y evaluación de tecnologías Cloud.

Las principales soluciones que ofrece el mercado son:

- *OpenStack:* Es un software libre y open source distribuido bajo los términos de la licencia Apache. es una de las iniciativas de nube abierta que cuenta con más apoyos, entre los que están Dell, Cisco y HP, así como RackSpace.

- *Eucalyptus:* plataforma open source para la implementación de cloud privada e híbrido sobre arquitecturas de cluster.

- *CloudStack:* software open source de Cloud Computing para crear, controlar y desplegar infraestructuras cloud. Originalmente fue desarrollado por cloud.com y posteriormente por Citrix.

- *OpenNebula:* conjunto de herramientas open source que permiten manejar la infraestructura de virtualización de un centro de datos para construir cloud privadas, públicas e híbridas.

Para implementar una Arquitectura de Cloud Computing Hibrida Open Source, se ha decidido trabajar con OpenStack como IaaS y OpenShift como PaaS.

OpenStack (www.openstack.com*)* es la base de *OpenCloud* (www.opencloud.com*)*, una iniciativa que busca la interacción estandarizada de las plataformas cloud de diferentes empresas. Es un proyecto global para crear una plataforma de cómputo open source en el cloud, que cumpla con las necesidades de los proveedores de servicios en clouds públicos, privados e híbridos, independientemente de su tamaño, que sea fácil de implementar y escalable.

El proyecto fue fundado en octubre de 2010 por la empresa Rackspace Cloud y por la agencia espacial estadounidense, NASA. Actualmente es gestionado por la Fundación OpenStack y cuenta con el apoyo de más de 150 empresas, entre las que destacan Rackspace, Oracle, AMD, Cisco, Canonical, IBM, HP, Dell Red Hat, Suse Linux, VMware, Yahoo y KIO Networks en América Latina.

OpenStack está escrito totalmente en Python y es el tercer proyecto open source en importancia a nivel mundial; y se lo describe como un "sistema operativo de cloud", ya que se desempeña como el coordinador de los recursos que sustentan a los servicios [9].

Un cloud de OpenStack se compone de cuatro nodos con diferentes funciones dependiendo de los componentes de OpenStack que tenga instalados, como muestra la figura 1.

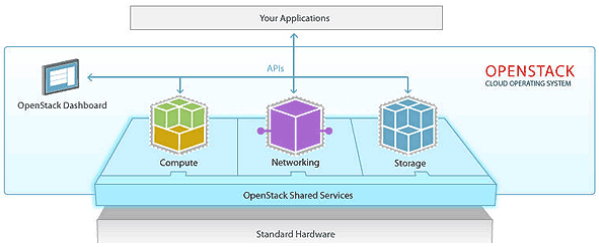


Figura 1: Arquitectura de OpenStack

- **Controller node:** encargado de la gestión de las instancias con Nova y de la orquestación del resto de componentes.
- **Compute node:** encargado de la ejecución de las instancias como máquinas virtuales sobre un hipervisor. Los protocolos de red de almacenamiento (por eje., iSCSI) y el plugin de red Switch.
- **Network node:** encargado de la comunicación de las redes internas con el exterior lo cual permite network connectivity-as-a-service, mediante Quantum que está basado en OpenFlow.
- **Storage node:** encargado de la gestión del almacenamiento. Los principales componentes son Glance, Cinder y Swift.

Cada uno de estos nodos está formado por diferentes componentes que trabajan en conjunto. Esta integración es lograda a través de APIs que cada servicio ofrece y consume. Gracias a estas APIs, los servicios pueden comunicarse entre ellos y además se posibilita que un servicio sea reemplazado por otro de similares características siempre que se respete la forma de comunicación. Es decir, OpenStack es extensible y se ajusta a las necesidades de quien desee implementarlo [10].

Openshift (www.openshift.com*)* es un PaaS presentado por Red Hat en 2011 [11]. Esta plataforma soporta una importante variedad de lenguajes de programación como: Java, Ruby, PHP, Perl y Python;

además incluye los frameworks de Spring, Seam, Weld, CDI, Rails, Rack, Symfony, Zend Framework, Twisted, Django y Java EE. Por otro lado soporta sistemas de archivos distribuidos y almacenamiento de datos SQL y NoSQL. Está basado en los estándares de interoperabilidad Deltacloud y esto permite a los diseñadores ejecutar sus aplicaciones en cualquier proveedor de Cloud Público.

Deltacloud y Libcloud son abstracciones que facilitan el desarrollo de aplicaciones independientemente de las especificaciones de un proveedor de cloud. Sin embargo esto tiene un precio: al ser abstracciones no pueden exponer las características que diferencian a las distintas plataformas y por lo tanto puede representar una pérdida significativa de funcionalidad o de rendimiento. En otras palabras, cuando se trabaja directamente sobre un proveedor de cloud se aprovecha el conjunto completo de sus características, obteniendo beneficios significativos que se traducen en productos más ricos, sólidos y rápidos. Ambos enfoques tienen su lugar y se necesitan en el mercado.

OpenShift, se presenta en tres versiones diferentes [12]: OpenShift Origin, OpenShift Online y OpenShift Enterprise.

Las aplicaciones pueden moverse entre las distintas versiones de OpenShift, siempre y cuando los componentes usados estén disponibles. Esto le da a los desarrolladores y empresas una opción muy buena de **cloud híbrida**, donde es posible que desarrolladores y pequeños equipos trabajen sobre la versión Online con aquellas aplicaciones que permiten datos en el cloud público. Sin embargo, si se comienza a desarrollar una aplicación que tiene requerimientos de datos más estrictos, se puede cambiar de versión manteniendo el mismo patrón de desarrollo que se utiliza para la plataforma Online.

5. Arquitectura propuesta

La arquitectura de cloud que se instaló involucra un sistema operativo de cloud, OpenStack, que administre la capa IaaS y de virtualización, mientras que para la capa PaaS se instalará OpenShift.

La distribución de OpenStack que se usó fue la versión **IceHouse**, la cual es la última versión estable de la plataforma y se encuentra en la novena distribución, liberada el 16 de abril de 2014.

Openstack está formado por distintos componentes que interactúan para proporcionar los servicios de computación, red y almacenamiento, ofreciendo un interfaz sencillo para que el usuario tenga acceso a sus recursos.

En el caso de la versión IceHouse los componentes son 9, tal como se muestra en la figura 2 y, se ajustan al Esquema Conceptual que se muestra en la figura 3.

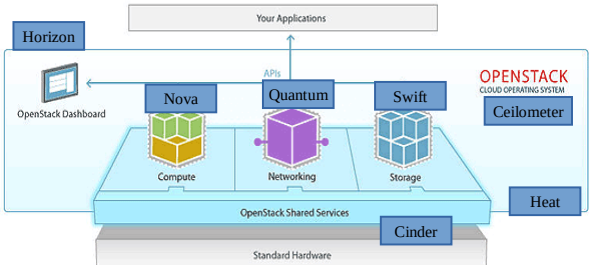


Figura 2: Arquitectura de OpenStack Havana

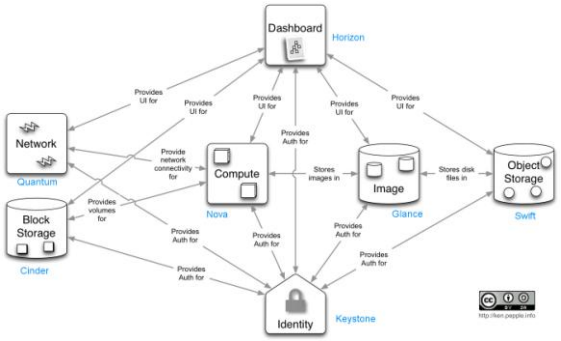


Figura 3: Esquema conceptual de OpenStack IceHouse

Para realizar la instalación de OpenStack IceHouse, se montaron los cuatro nodos en una sola computadora, la cual va a trabajar como servidora y debe tener una instalación previa de Ubuntu 14.04 de 64 bits.

El repositorio que se usó para acceder a OpenStack IceHouse, se encuentra en GitHub por lo tanto se debe tener instalado el paquete git: apt-get install git.

Una herramienta que ofrece OpenStack es DevStack (devstack.org), la cual ofrece un conjunto de script bash que permiten instalar OpenStack de forma automática.

Para realizar la instalación se debe clonar el repositorio git y elegir la rama estable de la versión **IceHouse,** como se muestra a continuación:

git clone https://github.com/openstack-dev/devstack.git

cd devstack

git checkout stable/idehouse

Branch stable/idehouse set up to track remote branch stable/idehouse from origin.

Switched to a new branch 'stable/idehouse'

./stack.sh

Antes de ejecutar el script se debe personalizar las distintas opciones de configuración, el sitio de DevStack, se provee una guía para realizar esta tarea.

Una vez terminada la instalación, se puede acceder a Horizon mediante Dashboard, accediendo a la url http://ip_flotante/:, en este punto se despliega la pantalla de autenticación, como muestra la figura 4.

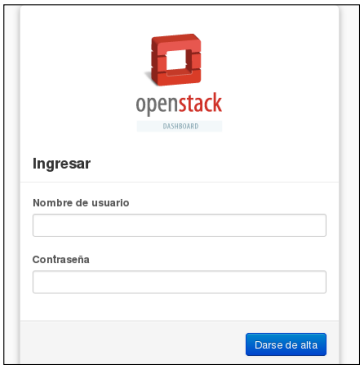


Figura 4: Pantalla de autenticación del Dashboard

Se han creado dos usuarios:

- El usuario administrador se llama **admin**, con contraseña **devstack**.
- Un usuario sin privilegios llamado **demo**, con contraseña **devstack**.

Una vez validado el usuario aparece la pantalla de administración, como muestra la figura 5.



Figura 5: Pantalla del administrador del Dashboard

Mediante esta interface se pueden configurar:

- *Imagen:* Imagen de sistema pre configurado que se utiliza como base para crear instancias. Dentro del cloud habrá diferentes imágenes para cada tipo de instancia que se quiera utilizar.
- *Instancia:* Clon de una imagen que se crea a demanda del usuario en uno de los nodos del cloud.
- *IP privada:* Dirección IP con la que se crean las instancias y que se utiliza para comunicación interna.
- *IP Flotante:* Dirección IP publica que puede asociarse a diferentes instancias con el fin de acceder a ellas desde fuera.
- Grupo de seguridad: Reglas de iptables que controlan el acceso a las instancias mediante la dirección IP flotante.
- Par de claves ssh: Utilizadas para acceder por ssh a las instancias desde fuera del cloud.

Una vez que que se ha terminado de trabajar con OpenStack se deben parar los servicios, ejecutando el script unstack.sh.

```
# cd devstack
# ./unstack.sh
```

Cuando se enciende nuevamente la máquina se debe volver a ejecutar los servicios ejecutando el script rejoin-stack.sh.

```
# cd devstack
# ./rejoin-stack.sh
```

Paralelamente al trabajo de instalación de OpenStack, se comenzaron las investigaciones de OpenShift para lo cual se monto provisionalmente sobre otra máquina, la cual trabaja con un IaaS simulado.

La versión elegida para el proyecto de investigación y desarrollo es OpenShift Origin y el sistema operativo utilizado es Ubuntu 14.04.

Realizando investigaciones en la web oficial y en la comunidad de OpenShift Origin, se descubrió que montar el PaaS sobre Ubuntu directamente y de manera nativa, es complicado y muy inestable pues los repositorios y compatibilidades están hechas para distribuciones de linux basadas en Red Hat y no en Debian. En pocas palabras el mencionado sistema fue hecho casi exclusivamente para ser montado sobre una maquina corriendo Red Hat Enterprice Linux o CentOs (hambas distribuciones bajo el sello de Red Hat) rompiendo un poco con la idea de “software libre” y “código abierto” que tanto llaman la atención del servicio elegido.

De todas maneras, aunque no se puede realizar una instalación paso a paso del sistema, si es posible realizar

un montado de OpenShift sobre Ubuntu a partir de una máquina virtual pre configurada que Red Hat facilita y recomienda.

Lo primero que se debe hacer es descargar la imagen de la VM tal como su guía de montado indicaba, esta tarea se puede ver en la figura 6. El sistema de virtualización que se escogió fue KVM gracias a tu facilidad de uso y configuración así como también las ventajas de emular directamente a nivel de kernel.

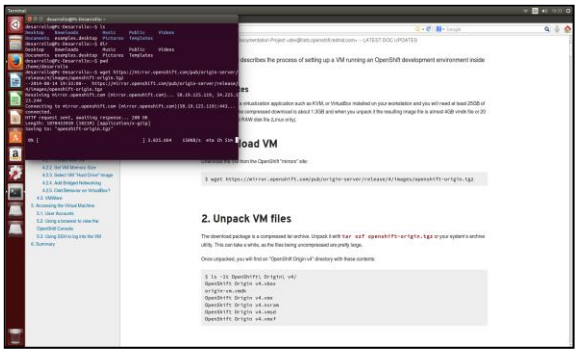


Figura 6: Captura de pantalla del sitio de OpenShift

Fué necesario realizar la instalación de mDNS pues OpenShift lo utiliza internamente para la resolución de nombres de dominios para las aplicaciones así como también de los nodos conectados al broker.

El paquete de mDNS en Ubuntu es distinto al utilizado en RHEL o CentOs, el paquete a instalar mediante el gestor de aplicaciones es: lib32nss-mdns. Siendo el medio de instalación: sudo apt-get install –y lib32nss-mdns.

Una vez instalado el mDNS fue necesario configurar el archivo “nsswitch.conf” del sistema para que el cliente pueda resolver las direcciones sin problemas: sudo nano /etc/nsswitch.conf.

Es necesario verificar que los servicios de libvirt se encuentren en ejecución. La guía formal del montado de la VM indica que se debe utilizar el comando “systemctl” pero dicha función no se encuentra disponible en Ubuntu. En lugar de systemctl se utilizó un comando propio de los paquetes de “startup”.

A continuación resta crear la VM con KVM y luego encenderla. Posteriormente se tiene acceso a la consola para desarrollo, como lo muestra la figura 7.

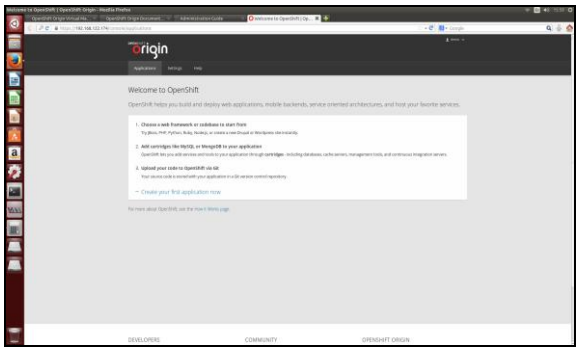


Figura 7: Consola web de OpenShift

6. Trabajos Realizados y Futuros

Una tarea que le llevó al equipo de trabajo mucho tiempo es decidir cuál sería el IaaS que se usaría para implementar el cloud. De entre todas las opciones existentes se seleccionó OpenStack debido a que va más allá de un IaaS, debido a que la forma que tiene de manejar los componentes, lo convierte en un sistema operativo de cloud, donde la capa de virtualización está incluida como un componente de la plataforma.

Hasta el momento se ha montado el servidor del cloud privado y se han configurado *los cuatro nodos*. También se ha analizado la forma en la cual trabaja el *Dashboard*, para poder configurar los componentes del cloud.

Otra tarea que se está realizando es el análisis de la instanciación y configuración de los clientes. Además se está investigando la forma en la cual se comunicara el cloud privado con Internet.

Por último, resta la instalación de OpenShift sobre OpenStack, para realizar esta tarea es necesario contar con la infraestructura de OpenStack ya totalmente configurada para poder empezar con la fase de producción.

7. Conclusiones

Cloud Computing ha dejado de ser una “buzzword”, para pasar a ser uno de los puntos de interés de los especialistas en IT. La posibilidad de instalar, configurar y gestionar, en el ámbito académico una arquitectura de cloud privado open source, que se pueda extender a un cloud híbrido, permitirá contar con herramientas de investigación actuales en temas de gran actualidad. El contar con un cloud de estas características se plantea como una paulatina eliminación de los recursos físicos propios y su sustitución por recursos virtuales con escalada elástica.

OpenStack es una plataforma de cloud computing íntegramente open source (no open core), cuyos objetivos de creación fue la necesidad de contar con estándares abiertos y cubrir las necesidades de los proveedores e integradores de cloud públicas y privadas, con independencia de su tamaño.

La idea es que sea muy escalable y sin excesiva complejidad. Su concepción transparente y abierta (no solo en su código fuente, sino en su gobernanza a través de una Fundación) protege a los clientes del vendor lock-in.

Además, gracias a su ecosistema de empresas, desarrolladores e integradores cooperando y compitiendo entre sí al tiempo, todo va mucho más rápido. No menos importante es su énfasis en la interoperabilidad, que proporciona la libertad para federarse entre proveedores y la capacidad de moverse entre clouds.

Este conjunto de características técnicas y comunitarias, junto a su esquema de licenciamiento permisivo y colaborativo (no-copyleft), han convertido a OpenStack en el proyecto de software libre más dinámico de los últimos años y con mayor apoyo por parte de la industria.

En función de todo esto, no cabe dudas que la instalación de un cloud privado con vistas a convertirlo en híbrido requiere de todas las características ofrecidas por OpenStack, el que se está convirtiendo en la implementación de referencia, para un cloud en fase de producción.

Con respecto a la elección de OpenShift como PaaS, el principal problema que tiene es que el hecho de ser open source, no hace que aparezca una comunidad que contribuya y colabore a su alrededor.

Para solucionar este problema, Red Hat se está migrando hacia un nuevo modelo de contribución que utiliza CI (Continuous Integration) pública. La naturaleza del nuevo modelo de contribución será similar a la forma en que hoy funciona el modelo de desarrollo del kernel Linux, el cual es en base a requerimientos de nuevas características en modo “Git pull.”

Sin embargo a pesar de este esfuerzo, OpenShift está atado a la plataforma de RedHat y es complicado de implementar y difícil de configurar sobre otros cores de Linux.

8. Bibliografía

[1] Sean Marston , Zhi Li , Subhajyoti Bandyopadhyay , Juheng Zhang , Anand Ghalsas (2010) “Cloud computing — The business perspective”, *Decision support systems*, Volume 51, Issue 1, April 2011, Pages 176–189, Elsevier.

[2] Armbrust, Fox, Griffith, Joseph, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica, Zaharia. “A View of Cloud Computing” *Communications of the ACM*, Vol. 53 No. 4, Pages 50-58.

[3] Mell, Grance. "The NIST definition of cloud computing." *NIST Special Publication* 800 – 145, 2011.

[4] Zhu, Fang et al. "IBM Cloud Computing Powering a Smarter Planet", *Libro Cloud Computing*, Volumen 5931/2009, Páginas 621- 625.

[5] Rodríguez, Chávez, Martin, Murazzo, Valenzuela. “Interoperabilidad en cloud computing”. *WICC 2011*. Rosario Argentina.

[6] Murazzo, Rodríguez, Villafañe, Gallardo. “Desarrollo de aplicaciones colaborativas para cloud computing”. *CACIC 2013*, Mar del Plata, Argentina.

[7] Li, Wang, Du. “Applications integration in a hybrid cloud computing environment: modelling and platform”. *Enterprise Information Systems*, 7(3), 237-271. 2013.

[8] Balachandran. “Development of A Decision Support System for Hybrid and Cloud Computing”. *Intelligent Decision Technologies: Proceedings of KES-IDT 2013*, Vol. 255, p. 187. Courier Dover Publications. Junio 2013.

[9] OpenStack. “Introduction to OpenStack, Chapter 2. Brief Overview”. URL: http://docs.openstack.org/training-guides/content/module001-ch002-brief-overview.html.

[10] OpenStack. “Introduction to OpenStack, Chapter 4. OpenStack Architecture”. URL:http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html

[11] Red Hat. Red Hat Launches OpenShift Platform as a Service. URL: http://www.eweek.com/c/a/Cloud-Computing/Red-Hat-Launches-OpenShift-Platform-as-a-Service-721913/

[12] Openshift: https://www.openshift.com/

Sistema de Comunicación para Renderización Paralela de Volúmenes en Tiempo Real

C. F. Perez-Monte
GridTICs
Universidad Tecnológica Nacional (UTN) - Facultad Reg. Mendoza
Mendoza, Argentina
cristian.perez@gridtics.frm.utn.edu.ar

M. F. Piccoli
LIDIC
Universidad Nacional de San Luis (UNSL)
San Luis, Argentina
mpiccoli@unsl.edu.ar

C. Luciano
Department of Mechanical and Industrial Engineering
University of Illinois at Chicago (UIC)
Chicago, IL, USA
clucia1@uic.edu

S. Rizzi
Argonne Leadership Computing Facility
Argonne National Laboratory (ANL)
Lemont, IL, USA
srizzi@alcf.anl.gov

Resumen

Para resolver problemas de gran complejidad como la renderización de volúmenes se utilizan sistemas de cómputo paralelos con una elevada potencia de cómputo. En este trabajo se presenta la implementación de un sistema de comunicaciones orientado al mejor esfuerzo, energéticamente eficiente y tolerante a fallas para resolver la renderización fotorrealista en tiempo real. A fin de conseguir estos objetivos se propone un sistema que utiliza en capa de Transporte UDP, en capa de Red IPv6 (unicast - multicast) y en capa de Enlace de Datos Gigabit Ethernet con control de flujo. Al final exponemos algunos resultados experimentales.

1. Introducción

La renderización fotorrealista de volúmenes en tiempo real requiere una elevada potencia de cómputo para su ejecución debido a la resolución, velocidad de cuadros y efectos de iluminación requeridos. Esta demanda puede ser resuelta mediante la utilización de un sistema paralelo compuesto por numerosos nodos conectados a través de una red. La aplicación de técnicas de paralelización permite distribuir el procesamiento entre los nodos, usando una red con buen desempeño (performance) para las comunicaciones.

En este trabajo planteamos un modelo de sistema con diferentes roles para los nodos conectados y un modelo de distribución de tareas, constituyendo ambos el punto de partida para el diseño del protocolo de comunicación propuesto. Para lograrlo, primero revisamos los conceptos previos, necesarios para el desarrollo del trabajo. Luego se detallan los aspectos de diseño y de implementación considerados en el modelo del sistema y del protocolo desarrollado. Finalmente, se explican los beneficios logrados mostrando algunos resultados experimentales, las conclusiones y futuros trabajos.

1.1. Renderización en Tiempo Real y Fotorrealista

Un modelo virtual es una representación de un objeto 3D, el cual puede estar descrito ya sea a través de polígonos, de un volumen u otra técnica. En nuestro caso el modelo es un volumen, el cual constituye una matriz tridimensional compuesto de voxels, cada uno con un nivel de intensidad proporcional a la densidad del tejido orgánico. Renderización es un término frecuentemente utilizado para describir el proceso donde a un modelo 3D se le realiza un procesamiento, según las condiciones de entrada especificada, para generar una imagen 2D desde un punto de vista del observador. Tanto la ubicación del observador como su orientación visual forman parte de las condiciones de entrada más importantes, pudiendo existir otras como la ubicación de la fuente de iluminación. En la renderización en tiempo real, estas condiciones pueden variar constantemente debido al cambio de posición del observador o las luces, implicando una generación continua de imágenes 2D.

La renderización fotorrealista tiene como objetivo generar una imagen, cuya percepción visual es lo más cercana a la realidad. Para lograr este objetivo se usan técnicas de iluminación y sombreado, las cuales requieren un procesamiento elevado.

1.2. Renderización Paralela Distribuida

En renderización paralela distribuida existe una clasificación, la taxonomía de Molnar [2] [8], la cual básicamente establece dos tipos de técnicas basadas en la manera en cómo distribuye el trabajo entre los diferentes nodos de un sistema distribuido. Estas dos técnicas se conocen con el nombre de Sort first o 2D y Sort last o DB, sus principales características son:

- La renderización paralela distribuida Sort first o 2D propone la división de la pantalla en áreas no solapadas, asignando una a cada nodo o unidad de cómputo disponible. De esta forma, cada nodo renderiza una sección determinada de la pantalla completa. Generalmente los algoritmos de renderización utilizados en este caso son del tipo Image Order, existen algunos desarrollos en sistemas con memoria compartida [10] y en sistemas GPU-CPU con memoria distribuida [1]. Si bien en ambos casos se logra un buen desempeño, la resolución de las imágenes no es muy alta. Finalmente con resolución de imágenes muy superiores, así como también volúmenes de gran tamaño, se logran excelentes resultados [12].
- En la técnica Sort last o DB, la renderización es hecha dividiendo sectores de volúmenes o primitivas, aplicar la renderización y finalmente realizar la composición de los resultados parciales y obtener la imagen final. Este método es especialmente adecuado para algoritmos de renderizado Object Order para resoluciones de la pantalla de salida no demasiado altas y grandes tamaños de volúmenes [4] [7]. Ésta es una técnica altamente escalable con una desventaja importante, la necesidad de interacción entre los diferentes nodos para realizar la composición.

Por último existe, fuera de la taxonomía de Molnar, una tercera técnica de renderización distribuida denominada Alternate Frame Rendering (AFR), muy utilizada en ambientes con varias GPUs en una única PC [9]. Esta se basa en la distribución de cuadros de pantalla.

De las tres técnicas antes citadas, 2D y AFR pueden ser aplicadas en nuestro sistema, constituyendo una buena solución para la aplicación en la renderización fotorrealista en tiempo real. Otras técnicas específicas propias de algunos algoritmos de renderización también pueden utilizarse.

Todas estas técnicas pueden combinarse y definir nuevas políticas aplicables a ambientes distribuidos-paralelos para resolver problemas con gran demanda computacional. El estudio de estas políticas de renderización exceden el alcance de la presente publicación pero el protocolo propuesto está adaptado para funcionar con cualquiera de ellas.

2. Sistema de Renderización Distribuida en Tiempo Real

El sistema propuesto está compuesto por 3 tipos de nodos, cada uno de los cuales tiene un rol definido. Los roles que puede asumir un nodo en el sistema son:

- ***Nodo Administrador:*** Es el nodo que gestiona el procesamiento del sistema al administrar las variables de entrada recibidas desde un dispositivo de entrada para transmitirlas por red a los *Nodos Procesadores*. Adicionalmente debe anunciar a los *Nodos Procesadores* qué Nodo o conjunto de *Nodos Integradores* les corresponde. Opcionalmente puede determinar o no qué procesa cada *Nodo Procesador* de acuerdo a políticas de implementación.
- ***Nodo Procesador:*** Es el nodo que realiza el renderizado y envía el trabajo finalizado por la red al *Nodo Integrador*. Puede asumir, en el caso de que el *Nodo Administrador* no lo haga, la función de auto asignarse la tarea a procesar.
- ***Nodo Integrador:*** Es el nodo que recibe todas las tareas procesadas y las integra pudiendo ser el encargado también de realizar la visualización en tiempo real de la escena renderizada. Pueden existir múltiples *Nodos Integradores* pero existe una limitación, cada *Nodo Integrador* tendrá asignado determinados segmentos de la imagen que no podrán

estar asignados simultáneamente a otros *Nodos Integradores*.

Considerando los distintos tipos de nodos, el sistema de renderización distribuido consta de un *Nodo Administrador* para coordinar las tareas a realizar por los demás nodos, el cual puede estar separado del resto del sistema en una red alejada permitiendo el control del sistema remotamente; varios *Nodos Procesadores*, quienes realizan la renderización propiamente dicha, y los *Nodos Integradores*, responsables de reunir y realizar la visualización en tiempo real del volumen. Estos dos últimos deben estar ubicados físicamente cercanos para su conexión a través de una red de alta performance y baja latencia. En la siguiente sección analizamos el modelo de comunicación propuesto para poder gestionar este sistema.

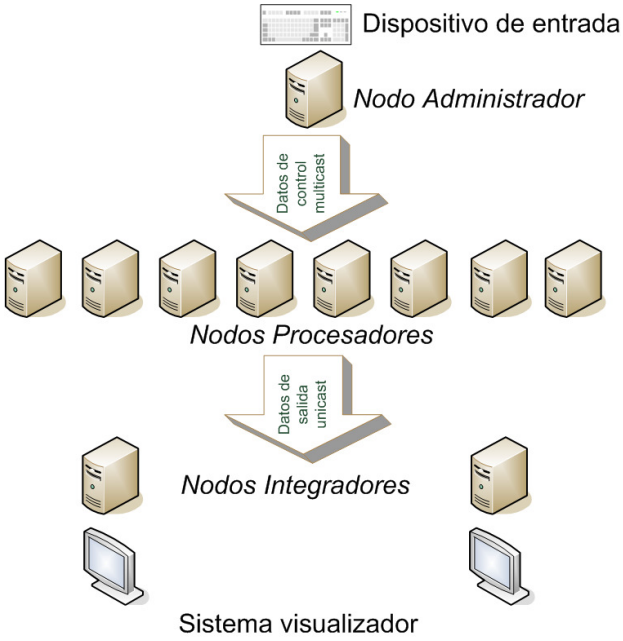


Figura 1. Sistema visualizador

2.1. Modelo de Comunicación

Los nodos en el sistema interaccionan entre sí de una manera específica (ver figura 1), estableciendo de este modo dos tipos de comunicaciones, las cuales son:

- **Información de control:** Esta comunicación tiene lugar desde el *Nodo Administrador* a los *Nodos Procesadores* para enviar la información de las variables de entrada (también denominadas variables de estado del sistema) e información de pertenencia de los segmentos de tareas que le permite conocer a los *Nodos Procesadores* a cuáles *Nodos Integradores* enviar cada segmento. Dado los escasos requisitos necesarios de ancho de banda, esta información puede ser transferida a través de enlaces WAN a redes remotas. La utilización de multicast disminuye los efectos que la latencia de los enlaces WAN pueda ocasionar en el funcionamiento del sistema.
- **Información de salida:** Esta comunicación se lleva a cabo entre los *Nodos Procesadores* y los *Nodos Integradores*. Cada una de ellas contiene la información del streaming multimedia distribuido, el cual está compuesto por los sucesivos cuadros en el tiempo generados por cada uno de los *Nodos Procesadores*. Los requisitos de gran ancho de banda y baja latencia requieren de un enlace LAN exclusivo para la comunicación de salida entre los *Nodos Procesadores* y los *Nodos Integradores*.

En las siguientes secciones se detallan las principales características del protocolo y el modo de funcionamiento.

2.2. Segmentación de Datos

Generalmente en entornos de computación masivamente paralela como el usado en el ambiente de las arquitecturas GPUs (compuesta por grupos de procesadores SIMD) es común usar modelos en los cuales todas las tareas a procesar se dividen en bloques, los cuales son asignados a diferentes procesadores SIMD de la arquitectura [6]. Generalmente es común en sistemas distribuidos contar con nodos con GPU, por ello es importante utilizar un modelo de comunicación el cual mantenga sobre el sistema completo un esquema similar al utilizado por cada nodo. Es por ello que una tarea renderizada (información de salida) que está compuesta por cuadros sucesivos en el tiempo, se puede dividir en segmentos, cada uno de los cuales representa una porción de un cuadro en un momento determinado del tiempo. Denominaremos CS (Compute Segment) a cada una de las estas unidades mínimas de transmisión.

Se puede definir a este CS como una unidad mínima de procesamiento a ser asignada a un *Nodo Procesador* de la red. Esto es equivalente a los bloques de CUDA o work-group de OpenCL (Unidad mínima que procesamiento que puede asignarse a un Streaming Processor, SM) La diferencia entre ellos está en que un bloque es asignado a un sólo SM de la GPU, en cambio un mismo CS de un mismo cuadro puede ser procesado (de forma diferente o no) en más de un nodo, por lo cual el protocolo debe poder contemplar esta situación. Métodos de renderización iterativos o tolerantes a fallas nodales pueden justificar este modo de funcionamiento. Para los casos de sistemas en los cuales cada *Nodo Procesador* utiliza su GPU para procesar y conseguir una máxima performance, los CS deberán tener dimensiones múltiplos de los work-group de OpenCL, por ello el protocolo permite que dicho tamaño sea configurable adaptándose a la arquitectura utilizada.

Los *Nodos Integradores* en cambio sólo pueden ser asignados para recibir CS específicos. Un mismo CS no puede ser recibido por más de un *Nodo Integrador*. Este tipo de arquitectura permite que los *Nodos Integradores*, en caso de ser más de uno, realicen también la visualización en forma distribuida.

La segmentación presenta un nuevo paradigma para la computación masiva sin estado interno que denominaremos computación del mejor esfuerzo que guarda similitudes con la entrega del mejor esfuerzo presente en protocolos de comunicaciones de red sin conexión como IP. En el caso de la computación, ésta se realiza sin garantía de sincronismo mientras que en el caso de la comunicación se hace sin garantía de entrega.

2.3. Estado del Sistema

El estado completo del sistema es definido por el *Nodo Administrador* en cada paquete de Información de control, de forma tal que no hay estados internos almacenados en los *Nodos Procesadores*. Para una correcta gestión y sincronización de la información, cada paquete de Información de control contiene tres diferentes tipos de contadores: el primero varía en función del tiempo, el segundo se incrementa mientras las variables de estado no varíen y se resetea cuando estas cambian y el tercer contador se incrementa cuando las variables de estado se modifican.

Existen básicamente dos estados bien definidos en el sistema, ellos son:

- **Estado Estable:** Estado en el cual las variables de estado que se envían constantemente como información de control no han variado durante un tiempo lo suficientemente largo permitiendo que todos los *Nodos Procesadores* posean el mismo estado.
- **Estado Inestable:** Las variables de estado varían lo suficientemente rápido de forma tal que se asignan diferentes variables de estado a cada uno de los *Nodos Procesadores*.

2.4. Modo de Funcionamiento

El modo de funcionamiento a través del tiempo consta de las varias etapas en donde el estado completo del sistema, compuestos por las variables de entrada y sus correspondiente variables de tiempo, son administradas por el *Nodo Administrador*. A continuación se describe la función de cada tipo de nodo en el sistema.

Los *Nodos Procesadores* se anuncian al *Nodo Administrador*, luego éste envía al grupo multicast de todos los *Nodos Procesadores* los datos de control (variables de entrada capturadas por un dispositivo de entrada con información del tiempo al momento de la captura). El envío se realiza periódicamente con lapsos de tiempo inferiores al tiempo de refresco del sistema visualizador asegurando así que siempre los *Nodos Procesadores* tengan tarea para realizar. El *Nodo Administrador*, también a través de datos de control, le indica a los *Nodos Procesadores* a cuál *Nodo Integrador* le corresponde cada CS de pantalla, no pudiendo haber más de un mismo *Nodo Integrador* asignado a un CS determinado.

Cada *Nodo Procesador* posee una copia completa del volumen a renderizar. En la actualidad el crecimiento del tamaño de la memoria de video de las placas GPU es suficiente para almacenar una copia completa de casi cualquier volumen médico, además el envío de volúmenes a través de la red reduciría el ancho de banda disponible de la red, por lo cual se considera desaconsejable dada la capacidad de almacenamiento de los nodos en la actualidad. Luego cada *Nodo Procesador* recibe la información de control, eligiendo la más actual posible y descartando el resto, y a partir de la cual realiza el procesamiento, generando el tráfico con la información de salida, el cual está compuesto por grupos de CS correspondientes a cada cuadro renderizado. Este tipo de comunicación se lleva a cabo mediante envíos unicast a los respectivos *Nodos Integrador*. El envío de esta información se realiza tan rápido como cada *Nodo Procesador* es capaz de hacerlo, el límite lo impone el procesamiento o el ancho de banda de la red mediante la cual se conectan los *Nodos Procesador* con los *Nodos Integrador*.

Los *Nodos Integrador* tienen como función reunir el streaming multimedia distribuido para finalmente visualizarlo en un sistema compuesto de tantas pantallas como *Nodos Integrador* existan.

3. Características e Inter-operatividad del Protocolo

El sistema de comunicación diseñado tiene las siguientes características:

- **Mejor esfuerzo:** El protocolo, al igual que el sistema en su conjunto, aplica la filosofía del "mejor esfuerzo computacional". Esto significa que ante limitaciones reales (ancho de banda de enlace, *Nodos Procesador* con diferente potencia de cómputo, nodos ocupados, etc.), el sistema lleva a cabo la tarea aprovechando los recursos y obteniendo los mejores resultados posibles. Razón por la cual el protocolo IPv6-UDP es orientado a datagramas, siguiendo un modelo de entrega con la misma filosofía. Así como el protocolo de comunicación es orientado a no-conexión y sin garantía de entrega, la computación es sin estado interno y sin garantía de sincronismo.
- **Tolerante a saturación de enlace:** Relacionado con el concepto anterior, el protocolo es capaz de aprovechar al máximo el ancho de banda de red y

ante su saturación, el protocolo es capaz de continuar funcionando normalmente. La pérdida de información no repercute en el sistema ya que cada datagrama incluye toda la información necesaria sobre el estado del sistema.

- **Energéticamente eficiente:** El procesamiento de información tiene un considerable consumo de energía, principalmente en los *Nodos Procesador*. Como el sistema de comunicación realiza el máximo esfuerzo por transportar los datos de salida, no siempre es posible lograrlo, resultando conveniente procesar sólo la cantidad de datos que el enlace entre los *Nodos Procesador* y los *Nodos Integrador* sea capaz de transportar.
- **Económicamente eficiente:** En cuanto a la posibilidad de aprovechar hardware de diferentes generaciones, logrando su integración en sistemas heterogéneos.
- **Tolerante a fallas de *Nodos Procesador*:** Cuando la cantidad de *Nodos Procesador* crece, las posibilidades de fallo de alguno de ellos se incrementa, razón por la cual ante ciertas políticas de renderización, el protocolo está preparado para tolerar la falla de un nodo y continuar trabajando.
- **Jerárquico:** Siguiendo metodología de encabezados opcionales luego de capa de transporte y acompañando los lenguajes de computación de procesamiento universal para plataformas heterogéneas, se utiliza una jerarquía compatible con los niveles de abstracción de software de las mismas.

En las próximas secciones se demuestra la capacidad del sistema para cumplir con estas características.

3.1. Arquitectura de la Pila de Protocolos

La arquitectura del stack de protocolos que se utiliza para cumplir con los objetivos antes mencionados es la siguiente:

- **Capa de enlace de datos:** En el caso de los enlaces entre los *Nodos Procesador* y *Nodos Integrador* se utiliza el protocolo Ethernet, con control de flujo (opcional) [5]. Si se tiene control de flujo se permite, ante saturación del enlace, a los *Nodos Procesador* sólo procesar aquello que son capaces de enviar por la red, reduciendo el consumo energético.
- **Capa de red:** Aplica el protocolo IPv6 [3] (el estándar de red de alcance mundial en su versión más reciente), siendo adecuado para computación distribuida. Además usa multicast para reducir el tráfico de información de control.
- **Capa de transporte;** Protocolo UDP [11]. Implementa un protocolo con modelo de entrega de mejor esfuerzo y tolerante a la saturación del enlace.

La arquitectura propuesta está diseñada utilizando protocolos de gran difusión permitiendo su implementación en sistemas heterogéneos de bajo costo habituales en entornos de investigación universitaria. La utilización de protocolos de red de capa de enlace de datos como Infiniband o 10 Gbit Ethernet pueden ser utilizados en un futuro cuando sean de menor costo y mayor difusión.

3.2. Descripción Detallada del Protocolo

En el protocolo se distinguen dos tipos de comunicación, las cuales:

- **Información de Control Multicast:**
  Está constituida por datagramas que tienen como función principal el envío de los datos de entrada. Incluyen 3 cabeceras, de las cuales dos son opcionales para el envío de los datos. Ellas son:
  - **Cabecera de control (CCm):** Es obligatoria, contiene la información de cada CS de control**.**
  - **Cabecera de control (CCGo):** Es opcional de índole general. Se usa para anunciar a los *Nodos Procesador*. Esta información puede referirse a los propios *Nodos Procesador* (qué CS debe procesar cada uno) o sobre los *Nodos Integrador* (qué CS tienen asignado cada *Nodo Integrador*). En CCGo se indica cuántas cabeceras CCEo hay.
  - **Cabecera de control específica por nodo (CCEo):** Es opcional con características específicas, contiene información específica para cada *Nodo Procesador* o *Nodo Integrador*.

  El datagrama de la información de control se muestra en la tabla 1.
- **Información de Salida Unicast**
  Como esta información se divide en CS, cada uno de ellos tienen, además, 3 cabeceras compuestas de varios campos, estos son:
  - **Cabecera de información general del sistema (CIGS):** Esta cabecera contiene una descripción general de las variables del sistema completo. Dicha información se repite en cada CS permitiendo la tolerancia a fallas de nodos o de red.
  - **Cabecera de información general del cuadro (CIGC):** La información enviada en cada CS, generalmente no varía de CS a CS de un mismo cuadro.
  - **Cabecera específica de CS (CES):** Es información específica de cada CS.

  Los requisitos del sistema incluyen entre otros que todos los cuadros del streaming de salida tengan la misma resolución y que todos los segmentos de un mismo cuadro tengan el mismo tamaño.
  El datagrama de la información de salida se muestra en la tabla 2.

Tabla 1. Información de control multicast.

	Campos	Descripción
Cabecera IPv6 + Cabeceras opciones	Definidos por el protocolo	Definidos por el protocolo según RFC 2460
	Dirección IPv6 de destino	Dirección IPv6 de destino multicast del grupo de los *Nodos Procesador* según RFC 3306
Cabecera UDP	Definidos por el protocolo	Definidos por el protocolo según RFC 768
CCM	Input Var Type	Descriptor de las variables de entrada
	Time stamp	Campo de marca de tiempo de variable de entrada
	Control Counter	Campo de número de dato de control enviado
	No Var Counter	Campo de número de dato de control enviado con idéntica variable de entrada.
	Next Header	Siguiente cabecera
CCGo	Node Type	Tipo de nodo (procesador o integrador)
	Node Counter	Cantidad de nodos
	Node Info	Tipo y cantidad de información del nodo
	Next Header	Siguiente cabecera
CCEo	ID Node	Identificador único de nodo: 64 bits de menor peso de dirección IPv6 de nodo procesador o integrador
	Segment Info	Campo de información específica para nodo: Mapa de bits de información de CS asignados para recibir por parte de nodo integrador o para procesar por parte de nodo procesador
	Next Header	Siguiente cabecera
Cuerpo	-	Datos de variables de entrada

Tabla 2. Información de salida unicast.

	Campos	Descripción
Cabecera IPv6 + Cabeceras opciones	Definidos por el protocolo	Definidos por el protocolo según RFC 2460
	Dirección IPv6 de destino	Dirección IPv6 de destino unicast de nodo integrador correspondiente al CS transmitido.
Cabecera UDP	Definidos por el protocolo	Definidos por el protocolo según RFC 768
CIGS	Frame Dim	Campo de cantidad de dimensiones del streaming de salida (generalmente es 2)
	Frame Resolution	Campos de resolución total del cuadro en cada dimensión
	Pixel type	Campo de cantidad de bit por pixel y tipo de representación
	Next Header	Siguiente cabecera
CIGC	Segment Dim	Cantidad de dimensiones del CS
	Segment Resolution	Tamaño del CS en cada dimensión
	Total Segment	Cantidad total de CS del cuadro
	Next Header	Siguiente cabecera
CES	ID Segment	Identificador de CS
	Representation Segment	Representación del CS. Campo de CS comprimido o no. Compresión con pérdida o no de calidad. Tipo de compresión
	Segment Size	Tamaño comprimido del CS
	Time stamp	Campo de marca de tiempo de variable de entrada: Cada CS tendrá marcas de tiempo que lo identifiquen como perteneciente a determinada variable de entrada y por lo tanto perteneciente a determinado cuadro en el tiempo.
	Segment Counter	Campo de número de CS generado con igual identificador de CS.
	Render Type	Tipo de método usado para la renderización
	Iteration Counter	Según corresponda nivel de procesamiento ejercido como cantidad de iteraciones del CS.
	No var counter	Campo de número de CS procesado con idéntica marca de tiempo e identificador de CS, útil para políticas específicas de renderizado.
	Next Header	Siguiente cabecera
Cuerpo	CS	Datos del segmento

3.3. Requisitos de Ancho de Banda de Red Teóricos y Latencia Máxima del Sistema

En la presente sección se determinan los requisitos mínimos de ancho de banda del sistema para que la red no sea limitante del rendimiento y la latencia que existe desde el momento que una variable de entrada es ingresada por el dispositivo de entrada hasta que la misma se hace efectiva en el sistema visualizador. Por último el rendimiento que poseerá el sistema en caso de que la red sea limitante del rendimiento.

Siendo:

m : Número de *Nodos Procesador.*

l : Número de *Nodos Integrador.*

R_x : Resolución horizontal.

R_y : Resolución vertical.

B_s : Cantidad de bytes por cada pixel.

R : Rendimiento en cantidad de cuadros por segundo.

BW : Ancho de banda de red.

T_s : Latencia de renderizado.

T_{BW} : Tiempo de transferencia del cuadro.

T_n : Latencia de red.

T_{nap} : Latencia de red entre *Nodo Administrador* y *Nodo Procesador.*

T_{npi} : Latencia de red entre *Nodo Procesador* y *Nodo Integrador.*

T_t : Latencia de interactividad total.

El caso extremo de ancho de banda utilizado por el canal de comunicaciones en la información de salida estará dado en su funcionamiento con la técnica AFR o políticas que la utilicen y será:

$$BW = \frac{R_x \times R_y \times B_s \times R \times m}{l}$$

La latencia total del sistema estará comprendida por la suma de la latencia en la renderización de cada cuadro de la pantalla sumada a la latencia de la red y al tiempo de transferencia del cuadro de pantalla en la red. El tiempo de latencia de red está determinado por la suma de la latencia *Nodo Administrador – Nodo Procesador* y *Nodo Procesador – Nodo Integrador*, en la cual esta última, por ser pequeña, puede despreciarse. El tiempo de transferencia del cuadro en la red puede determinarse según el ancho de banda de la red usada mientras que el tiempo de renderizado depende del hardware de los Nodos Procesador.

$$T_t = T_s + T_{BW} + T_n = T_s + T_{BW} + T_{npi} + T_{nap}$$

$$T_t \approx T_s + T_{bw} + T_{nap} = T_s + \frac{R_x \times R_y \times B_s}{l \times BW} + T_{nap}$$

Si bien la información de salida puede transferirse a través de redes de baja velocidad, el ancho de banda en este caso será un limitante en el rendimiento máximo del sistema expresado en cuadros por segundo (R), siendo este:

$$R = \frac{BW \times l}{m \times R_x \times R_y \times B_s}$$

Los requisitos antes expuestos no son limitantes del funcionamiento del sistema, sino umbrales a partir del cual el sistema estará limitado por un recurso (potencia computacional) u otro recurso (ancho de banda de red).

3.4. Tráfico de Salida Desperdiciado

Se le denomina tráfico desperdiciado al tráfico generado por los *Nodos Procesador*, el cual no puede ser procesado por los *Nodos Integrador*.

Existen básicamente dos tipos de tráfico desperdiciado:

- **Tráfico desperdiciado por *Nodo Integrador***
  Este tráfico generalmente es por falta de sincronización. Como los *Nodos Procesador* trabajan en un modelo asincrónico de mejor esfuerzo, puede ocurrir envíen CS, los cuales son recibidos tardíamente. En tal caso, los Nodos Integrador los descartan. Ocurre lo mismo si los buffers de recepción del *Nodo Integrador* no tienen el tamaño suficiente o trabaja a una velocidad reducida.
- **Tráfico desperdiciado por pérdidas de enlace:**
  Es todo aquel tráfico generado por los *Nodos Procesador* y no recibido por los *Nodos Integrador*. Un caso donde ocurre esto es cuando se produce la saturación del enlace en un medio sin control del flujo, por lo cual la pérdida de una trama que contiene un CS o parte del mismo hará que dicho CS no pueda ser reconstruido en el *Nodo Integrador*.

Si bien el sistema es capaz de funcionar con tráfico desperdiciando, el mismo genera un desperdicio de potencia computacional de los *Nodos Procesador* por lo cual es de suma importancia su reducción.

4. Resultados

Los resultados experimentales se realizaron en un cluster de computación gráfica de renderizado distribuido compuesto por 2 *Nodos Procesadores*, un *Nodo Administrador* y un *Nodo Integrador* conectados mediante una red Ethernet Gigabit. Se utilizó un MTU estándar de 1528 bytes. El tamaño del CS incluido en la carga del datagrama UDP fue definido en 48 Kbytes.

Se realizaron modificaciones en parámetros de kernel en el *Nodo Integrador* para incrementar los buffers de recepción evitando Tráfico desperdiciado por *Nodo Integrador*.

Las características de la arquitectura de cada uno de los *Nodos Procesadores* del sistema son:

- ***Nodo Procesador* 1:** Intel Core 2 Duo 4 GB de RAM GPU GTX560 1GB de RAM
- ***Nodo Procesador* 2:** AMD Fx6200 4 GB de RAM GPU GTX650 1 GB de RAM

La elección de estos nodos fue con la intención de demostrar el funcionamiento con hardware CPU de diferentes fabricantes y hardware GPU de diferente generación.

Las figuras 2 y 3 muestran respectivamente las potencias de cómputo y Ancho de banda de los nodos 1 y 2.

Las figuras 4 y 5 muestra la potencia de cómputo combinada y anchos de banda medidos sobre el *Nodo Integrador* con diferentes composiciones del sistema, valor de n y control de flujo en Ethernet activado o no.

n es la cantidad de iteraciones del método de renderizado, la cual permite incrementar el nivel del realismo en forma previa al envío de cada cuadro y la potencia de cómputo de combinada de todos los *Nodos Procesadores* intervinientes se presenta en MOPS (millones de operaciones /segundo).

La potencia de cómputo es definida como la cantidad de pixels procesados por la cantidad n de operaciones realizadas por cada uno de ellos por unidad de tiempo. Dicho parámetro es más objetivo que cuadros por segundo (FPS) a la hora de medir el rendimiento de los *Nodos Procesadores*.

En primer lugar se de la observación de los resultados de las figuras 2 y 3 es posible determinar que los nodos 1 y 2 en funcionamiento independiente no poseen pérdidas de potencia de cómputo ni tráfico.

En segundo lugar, a partir de la figura 4 es posible determinar que sin control de flujo existe un ancho de banda desperdiciado por pérdidas de enlace lo que

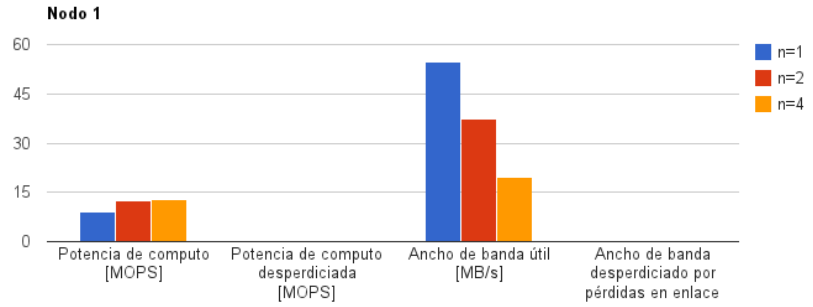


Figura 2. Potencia de cómputo y Ancho de banda de Nodo 1

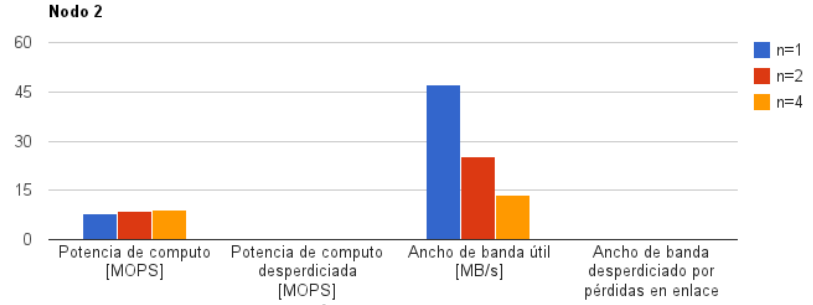


Figura 3. Potencia de cómputo y Ancho de banda de Nodo 2

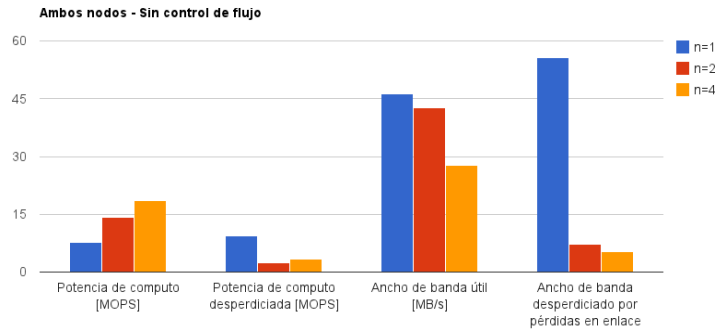


Figura 4. Potencia de cómputo y Ancho de banda de Ambos nodos sin control de flujo

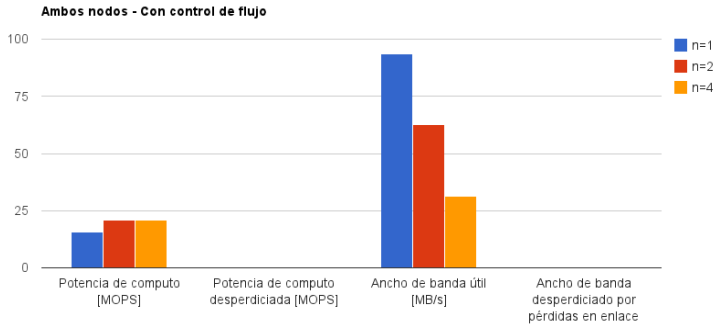


Figura 5. Potencia de cómputo y Ancho de banda de Ambos nodos con control de flujo

repercute en Potencia de cómputo desperdiciada.

Por último, es posible determinar que en estado estable, según se observa en la figura 5, con la configuración con control de flujo activada no hay pérdidas de CS y, por lo tanto, no existe desperdicio en potencia de cómputo, siendo así, toda la energía utilizada en el procesamiento de *Nodos Procesadores* aprovechada. También se observa que la potencia de cómputo global está limitada en el caso de n=1 por el ancho de banda de red mientras que para n superiores está limitada por la potencia computacional de los mismos. Así se demuestra su funcionamiento de computación de mejor esfuerzo.

En los casos de saturación de enlace, para n=1 y con funcionamiento de los dos nodos en simultáneo, se pudo comprobar una reducción:

En la velocidad del streaming con respecto a la suma del tráfico de los dos nodos separados.

En la potencia combinada con respecto a la potencia de cómputo de los dos nodos separados.

Si bien en los resultados no ha podido mostrarse debido a que no es un parámetro cuantitativo, se puede señalar que en estado inestable con fuertes variaciones del estado del sistema se pudieron observar pérdidas de CS por Tráfico desperdiciado por pérdidas en el enlace, las cuales en todos los casos fueron inferiores al 2% del tráfico total

Si bien la limitación de la potencia de cómputo combinada la impone el ancho de banda de la red utilizada, no hay desperdicio de energía. El ancho de banda utilizado es muy cercano al máximo de la red por lo cual se puede observar un mejor esfuerzo por conseguir el mayor rendimiento posible.

En las pruebas en las cuales se ha desactivado el control de flujo de ethernet, se puede observar que en los casos de saturación de enlace existe una pérdida considerable de tráfico. Esto es debido a que existe una gran fragmentación del tráfico IPv6 en tramas ethernet, ya que el MTU es mucho menor al tamaño del paquete IPv6, por lo cual, la pérdida de una única trama ethernet generará la pérdida de todo un CS. Incrementar el MTU o decrementar el tamaño del CS son posibles soluciones en estos casos.

En estas últimas pruebas, si bien el sistema funcionó con menor performance debido a la gran pérdida de tramas, mostró tolerancia a falla de CS, simulando

además un ambiente similar al caso en el cual hay fallas de *Nodos Procesador*, pudiendo demostrarse su funcionamiento con tolerancia a fallas de nodos procesador y saturación de enlace.

5. Conclusiones y Trabajos Futuros

En este trabajo planteamos un protocolo, el cual fue diseñado para un sistema de renderizado distribuido/paralelo en tiempo real. El protocolo cumple con los requisitos necesarios para el control y transmisión de streaming multimedia distribuido. Del análisis de los resultados derivados de la segmentación de datos, se pudo observar que la técnica usada es apropiada tanto para la transmisión de datos en tiempo real como para distribuir las tareas de procesamiento y de visualización. Además el protocolo propuesto no sólo logra conseguir el mejor esfuerzo, sino también ser tolerante tanto a saturación de red como fallas de los *Nodos Procesador*.

Entre las posibles extensiones de este trabajo se encuentra el análisis del comportamiento cuanto la conexión de los nodos es a través de Internet, por ejemplo el *Nodo Administrador* está ubicado en redes geográficamente alejadas de la red en donde se ubican los *Nodos Procesador* y los *Nodos Integrador*. Además se analizará la posibilidad de generalizar el protocolo de manera de poder realizar la gestión remota de cualquier tipo de procesamiento masivamente paralelo en aplicaciones de tiempo real.

6. Agradecimientos

Se agradece el asesoramiento y soporte con infraestructura de los integrantes de los Grupos de Investigación GridTICs y LICPaD de UTN-FRM y LIDIC de UNSL.

El trabajo es sostenido económicamente gracias al financiamiento de los proyectos 25J084 “SARA Operation”, PICT2010/29 “Procesamiento para visualización utilizando algoritmos paralelos en GPU y distribuidos en red” ambos de UTN-FRM, PROICO-30310 de UNSL y otras fuentes de financiamiento de grupo GridTICs.

Este proyecto ha sido realizado gracias al aporte económico de UTN a través de una beca de doctorado y al soporte académico de la carrera de posgrado de UNSL.

7. Referencias

[1] Bajaj C., Ihm I., Park S., Song D. (2000). “Compression-based ray casting of very large volume data in distributed environments.” In HPC ’00: Proceedings of the The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region-Volume 2. Pp 720–725.

[2] Cox M. (1995) “Algorithms for Parallel Rendering”. PhD thesis, Department of Computer Science, Princeton University.

[3] Deering S., Hinden R. (1998) “Internet Protocol, Version 6 (Ipv6) Specification” IETF Standards Track.

[4] Engel K., Hadwiger M., Kniss J. M., Rezk-Salama C., Weiskopf D.(2006) "Real-Time Volume Graphics." A K Peters, Ltd.

[5] IEEE Standards for Local and Metropolitan Area Networks: (1997) “Supplements to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Specification for 802.3 Full Duplex Operation and Physical Layer Specification for 100 Mb/s Operation on Two Pairs of Category 3 Or Better Balanced Twisted Pair Cable (100BASE-T2)," IEEE Std 802.3x-1997 and IEEE Std 802.3y-1997 (Supplement to ISO/IEC 8802-3: 1996; ANSI/IEEE Std 802.3, 1996 Edition).

[6] Kirk D., Hwu W. (2010) “Programming Massively Parallel Processors, A Hands on Approach” , Elsevier, Morgan Kaufmann.

[7] Marchesin S., Mongenet C., Dischler J. (2008). “Multi-gpu sort-last volume visualization“. In Eurographics Symposium on Parallel Graphics and Visualization (EGPGV08).

[8] Molnar S., Cox M., Ellsworth D., Fuchs H. (1994) “A Sorting Classification of Parallel Rendering.” IEEE Computer Graphics and Algorithms. Pp 23-32.

[9] Monfort J., Grossman M. (2009) “Scaling of 3D Game Engine Workloads on Modern Multi-GPU Systems” - The 1st ACM Conference on High Performance Graphics (HPG-09).

[10] Palmer M., Totty B., Taylor S. (1998). “Ray casting on shared-memory architectures: Memoryhierarchy considerations in volume rendering”. IEEE Concurrency, Vol 6, nro 1. Pp 20–35.

[11] Postel J., (1980) “User Datagram Protocol” IETF Standard protocol.

[12] Schwarz N., Leigh J., (2010) “Distributed Volume Rendering for Scalable High-resolution Display Arrays”, Grapp – International Conference on Computer Graphics Theory and Applications.

Impacto de la implementación de 802.11e para tráfico heterogéneo en MANET

Maria A. Murazzo, Nelson R. Rodríguez, Miguel J. Guevara

Universidad Nacional de San Juan – Facultad de Ciencias Exactas, Físicas y Naturales – Departamento e Instituto de Informática
Complejo Universitario Islas Malvinas (CUIM) – Rivadavia – San Juan – Argentina
marite@unsj-cuim.edu.ar - nelson@iinfo.unsj.edu.ar - migueljoseguevaratencio@gmail.com

Abstract

En los últimos años, el despliegue de las tecnologías inalámbricas en el sector de las telecomunicaciones ha experimentado un crecimiento exponencial. Asociado a lo anterior, se ha producido un notable aumento en la demanda del número de aplicaciones multimedia, tales como voz sobre IP (VoIP), audio conferencia, video conferencia, audio y video bajo demanda (AVoD). Estas aplicaciones requieren que los servicios suministrados a los usuarios ofrezcan garantías de calidad de servicio (QoS) debido a que son sensibles a la variabilidad de los recursos. Por otro lado, las aplicaciones clásicas, flexibles a la variabilidad de los recursos deben convivir con las anteriores, siendo necesario proveer un soporte adecuado para el tratamiento conjunto, pero diferenciando el tráfico que generan. El presente trabajo tiene como objetivo fundamental analizar el comportamiento de la implementación de QoS en la sub capa MAC mediante 802.11e en ambientes con tráfico heterogéneo, conformado por flujos provenientes de aplicaciones multimediales y clásicas.

Palabras Claves: Manet, QoS, trafico heterogéneo, NS2, 802.11e

1. Introducción

A diferencia de las redes inalámbricas tradicionales, las redes MANET son redes de comunicaciones formadas espontáneamente por un conjunto de dispositivos móviles, que son capaces de comunicarse entre sí, sin poseer una infraestructura ni una administración centralizada.

Además, debido a que el alcance de transmisión de los dispositivos inalámbricos es limitado, pueden llegar a ser necesarios nodos intermedios para transferir datos de un nodo a otro a través de la red. Es por ello que los nodos cumplen funciones de fuentes, routers y destinos.

La topología de las MANET es dinámica debido a que los nodos se mueven, se incorporan y abandonan la red continuamente; por esta razón, las condiciones de tráfico son altamente variables, haciendo extremadamente complicado dar buena QoS a las aplicaciones que la requieran.

Por otro lado, las variaciones en el canal de radio y las limitaciones de energía de los nodos pueden producir frecuentes cambios en la topología y en la conectividad. Consecuentemente, las MANET deben adaptarse dinámicamente para ser capaces de mantener las conexiones activas a pesar de estos cambios.

Todas estas características hacen que las MANET sean Auto Creadas (Self Creating), Auto Organizadas (Self Organizing) y Auto Administradas (Self Administering) [1].

2. El problema de la administración de QoS

El término QoS se refiere a la garantía de proveer un servicio con cierto grado de fiabilidad en la transmisión de información, a través de una red. Los principales parámetros que se utilizan para medir la misma, que puede ofrecer una red son: la disponibilidad de la red, el ancho de banda, el retardo de los paquetes, la variación del retardo, la tasa de errores y la tasa de pérdida de paquetes. En función de los requisitos de las aplicaciones, algunos parámetros serán más importantes que otros en el momento de definirla [2].

QoS es la capacidad de la red de proveer mejor servicio a cierto tipo de tráfico. En una MANET la implementación de una solución de QoS permite un comportamiento determinístico de la misma, el cual se puede traducir en una priorización de cierto tipo de tráfico. Una adecuada implementación de un ambiente de provisión de QoS es necesaria para eficientizar el uso y la asignación de los recursos de la red, mediante la identificación del tráfico con características críticas y restrictivas.

La conducta del tráfico y los requerimientos de QoS, varían de aplicación en aplicación, como la voz (por ejemplo, VoIP), y el video (por ejemplo, video-streaming), es decir, diferentes aplicaciones demandan diferentes desempeños y recursos de la red basados en: el ancho de banda necesario, la sensibilidad del retardo extremo a extremo, la pérdida de paquetes o la sobrecarga de la red [3].

Una adecuada implementación de soporte de QoS implica la clasificación o diferenciación de los flujos de tráfico en función de sus requerimientos. Pero además, es necesario que esta solución de QoS sea integral para todo el tráfico (elástico e inelástico), generando un ambiente adecuado para manejar flujos con diferentes requerimientos sin que haya un impacto en la arquitectura de soporte, ni en las aplicaciones.

El poder ofrecer niveles de QoS en MANET sigue siendo un tema abierto para la comunidad investigadora, y supone un reto interesante, dadas las dificultades que conlleva. Para lograr esto es necesario realizar la evaluación de múltiples parámetros, tales como: ancho de banda mínimo esperado, porcentaje máximo de pérdidas de datos, retardo máximo y variación máxima del mismo (jitter).

Si bien la administración de QoS está prácticamente resuelta en redes fijas, las características de las redes MANET hacen necesario un nuevo estudio para afrontar este problema [4].

Es por ello, que se puede encarar la problemática de la administración de QoS en redes MANET de dos formas: sobredimensionar la red de recursos o gestionar adecuadamente los recursos existentes. La primera solución tiene como problema principal el costo de implementación y la escalabilidad pues siempre que se generen degradaciones en la performance de la red se deberán volver a redimensionar; la segunda solución es la más adecuada y se puede implementar en distintas capas dependiendo de las prestaciones que necesite el tráfico.

Para realizar una adecuada administración de QoS se pueden implementar las siguientes soluciones:

- Ruteo con QoS: es parte de la capa de red y su función es buscar una ruta con los recursos suficientes sin hacer reserva de ellos de acuerdo al modelo de QoS.
- Señalización de QoS: este enfoque se desarrolla por encima de la capa de red y actúa como centro de control en el soporte de QoS. La funcionalidad de la señalización está especificada por el modelo de QoS a la que está sujeta. Este enfoque se apoya en el ruteo QoS haciendo reserva de recursos en la ruta seleccionada por el protocolo de ruteo.
- Control de Acceso al Medio con QoS: la capa de Control de Acceso al medio (MAC) es un componente esencial de las MANET. Todos los modelos antes mencionados dan por sentado en su funcionamiento que existe un protocolo MAC el cual se encarga de controlar el uso del medio, reservar recursos, y proveer una comunicación confiable.

En este trabajo se decidió trabajar con una implementación de QoS en la subcapa MAC, mediante el estándar 802.11e. Este estándar es una mejora de 802.11 para soportar requerimientos de QoS. El problema de 802.11 es que PCF (Point Coordination Function) y DCF (Distributed Coordination Function), no son aptos para brindar métodos de priorización de tráfico, por lo que no pueden brindar ningún tipo de QoS.

Si bien el método PCF al ser centralizado es adecuado para brindar QoS tiene muchas deficiencias. Una de ellas es que asigna la oportunidad de transmisión a los nodos que están sincronizados con este mediante round-robin, lo que impide administrar distintos tipos de servicios que requieren QoS en distintos niveles.

Otra falla importante es que el AP (Access Point) debe disputar el acceso al medio al igual que las demás estaciones, lo que podría retrasar la asignación del medio a una estación que requiera transmitir con urgencia. Además cuando una estación tiene asignado el medio y está transmitiendo, esta puede seguir haciéndolo aún cuando el tiempo de transmisión permitido haya expirado, por lo cual si existiese en ese momento una estación con mayor prioridad para transmitir tendría que esperar de todas formas a que la estación que está actualmente transmitiendo finalice.

El estándar IEEE 802.11e define un nuevo método de acceso llamado HCF (Hybrid Coordination Function) el cual combina los modos de acceso implementados por el primer estándar PCF y DCF. HCF tiene dos modos de operación llamados EDCA (Enhanced Distributed Channel Access) y HCCA (Controlled Channel Access).

EDCA se puede utilizar únicamente durante el periodo de contienda, mientras que HCCA puede operar en ambos periodos (con y sin contienda) pero el estándar 802.11e recomienda su uso solamente en el periodo sin contienda.

El estándar 802.11e, además de los métodos de acceso nuevos, agrega otras mejoras en la capa MAC,

siendo la más importante el agregado de un nuevo campo en el encabezado de la trama. Este campo sirve para indicar cuál es la prioridad que poseen los datos que son enviados en esa trama [5]

3. Tráfico Heterogéneo

En los últimos años, se ha producido un notable aumento en la demanda de aplicaciones multimedia sobre medios inalámbricos y específicamente sobre MANET, tales como: voz sobre IP (VoIP), audio conferencia, video conferencia, audio, etc. Estas aplicaciones requieren que los servicios suministrados a los usuarios ofrezcan garantías de QoS en cuanto al ancho de banda, retardo de extremo a extremo (latencia), variabilidad en el retardo (jitter) o tasa de error. En concreto, muchas aplicaciones multimedia son sensibles a la variabilidad en la tasa de transmisión, al retardo y a la variabilidad en el retardo, aunque toleran ciertas pérdidas de datos.

Por otro lado, existen las aplicaciones clásicas, flexibles al retardo y a la disponibilidad de ancho de banda, como pueden ser: correo electrónico, transferencia de archivos, navegación web y login remoto, que en general tienen requisitos distintos a las aplicaciones multimedia. En las mismas, la variabilidad en el retardo o en la disponibilidad de ancho de banda, aunque actúan en su detrimento, no impiden que se realice el servicio.

Ambos tipos de aplicaciones generan flujos de tráfico, el cual se puede clasificar en base a dos grandes categorías: tráfico elástico e inelástico.

El tráfico elástico se puede ajustar a los cambios en el rendimiento de la red, sin dejar de satisfacer las necesidades de las aplicaciones. Éste es el tipo de tráfico soportado por las aplicaciones clásicas que corren sobre TCP/IP, (Transmission Control Protocol/Internet Protocol), se adapta a la congestión reduciendo la velocidad de entrega de datos a la red.

Cuando se trata de tráfico elástico, es necesario contar con un scheduler de red que distinga entre uno y otro tipo de flujo en función de los recursos que demanda, de esta manera se realizara una adecuada administración de los mismos, sin interferir en el rendimiento de la aplicación. Sin esta administración, los routers gestionan a ciegas los paquetes, sin importarles el tipo de aplicaciones que los produjo o que está esperándolos, o si forman parte de una trasferencia grande o pequeña. Si se produce la congestión, es improbable que se asignen los recursos de modo que se satisfagan las necesidades de todas las aplicaciones. Cuando además se incorpora el tráfico inelástico, la experiencia es menos satisfactoria.

El tráfico inelástico no se adapta a las variaciones del rendimiento de la red, este tipo de tráfico es el que es generado por las aplicaciones multimediales. La mayoría del tráfico inelástico exige un mínimo de rendimiento consistente, esto es difícil de cumplir en una red congestionada, con retardos variables en las colas de los router y con una alta tasa de pérdidas de paquetes.

El tráfico inelástico introduce la necesidad de contar con un mecanismo para otorgar un tratamiento preferente a las aplicaciones que tengan requisitos más exigentes; por ejemplo, las aplicaciones de voz son mas sensitivas al retardo y las de video a la perdida de paquetes, ya sea de forma proactiva o reactiva.

Tanto las aplicaciones multimedia, como las clásicas utilizan IP como protocolo a nivel de la capa de red. Este protocolo proporciona un servicio de mejor esfuerzo (best-efort) para todos los datos enviados, sin diferenciar el tipo de tráfico o sus requisitos de QoS, es decir, ofrece un servicio de interconexión de redes independiente de los requisitos del usuario, con un tratamiento equitativo para todos los paquetes de datos [6].

Un requisito de cualquier ambiente de comunicaciones es poder administrar todo tipo de aplicaciones, brindándole los recursos necesarios para que puedan funcionar bien. Esta problemática, está prácticamente resuelta en redes cableadas mediante los protocolos de reservación de recursos. Esta solución es muy poco práctica en ambientes MANET debido a que la topología dinámica generaría constantes recálculo de la ruta de la reserva y además, la sobrecarga de los mensajes de reservación.

El hecho de poder ofrecer ciertos niveles de QoS en redes MANET sigue siendo un tema abierto para la comunidad investigadora, y supone un reto muy interesante dado las dificultades que conlleva. Específicamente, los servicios de tiempo real necesitan especial atención debido a que la naturaleza dinámica de estas redes hace difícil aplicar una gestión tradicional de QoS [7].

4. Planteo del problema

Para brindar la satisfacción a los usuarios es necesario contar con una plataforma para soportar aplicaciones clásicas (tráfico elástico) y aplicaciones multimediales (tráfico inelástico). El objetivo principal de esto es lograr que el usuario pueda correr todo tipo de aplicaciones, las cuales poseerán diferentes requerimientos de recursos.

Esta heterogeneidad del tráfico hace muy difícil aplicar una solución de QoS que satisfaga todos los

requerimientos de las aplicaciones y no resienta su desempeño.

En este contexto, se ha trabajado en la implementación de QoS en la subcapa MAC mediante el estándar 802.11e para tráfico CBR (Constant Bite Rate) [8], [9]. Este tipo de tráfico es extremadamente sensible a las variaciones de retardo y a la sobrecarga de la red. Bajo estas restricciones se determino que la implementación de 802.11e era la solución más adecuada, sin embargo en un ambiente de comunicaciones tradicional existe la coexistencia de tráfico que demandan de la red diferentes niveles de recursos.

Para realizar una adecuada administración de QoS en ambientes con tráfico heterogéneo, es necesario implementar una solución que tenga en cuenta el tipo de tráfico generado por cada aplicación, dándole priorización al que demande más recursos; todo ello sin afectar la performance de las demás aplicaciones. Para ello se trabajara con aplicaciones con flujos CBR para el tráfico inelástico y aplicaciones con flujos ftp, para el tráfico elástico.

De todos los parámetros de QoS, el tiempo de respuesta a solicitudes hechas desde dispositivos móviles se ha convertido en una de las necesidades fundamentales a la hora de comunicarse o conectarse. El tiempo de respuesta está fuertemente ligado al retardo de los paquetes de datos dentro de la transmisión. Pues si los paquetes tienen un alto retardo, el tiempo de respuesta se incrementará, provocando una disminución de la performance de la red, y por consiguiente una poca satisfacción del usuario.

El retardo en las MANET debería ser el menor permisible, o al menos tender a cero en conexiones preferenciales. Ahora bien, ¿qué tal si durante una transmisión de gran importancia la cantidad de paquetes de ruteo supera la de datos?, en tal caso la sobrecarga de información de ruteo genera un retraso en el envío y recepción de los paquetes de datos. En otras palabras, la sobrecarga de paquetes de ruteo aumentaría el retardo de los paquetes de datos. Por esta razón, además de controlar el retardo en las transmisiones, también es necesario controlar la sobrecarga de paquetes de ruteo, pues no solo la distancia o cantidad de saltos producen retardo en la recepción de paquetes sino también las tareas de administración de rutas, envíos de petición, confirmación de recepción, descubrimiento de rutas, etc.

Teniendo en cuenta estos aspectos, el estándar 802.11e [10] no congestiona la red con paquetes de señalización, ni con paquetes de descubrimiento de rutas, sino que plantea una forma de administración general, la cual se basa en los tiempos de espera. El que tiene mayor prioridad de transmisión es el que menos tiempo debe esperar. Esto hace, que no sea necesario inundar la red con paquetes de ningún tipo, pues cada nodo por separado sabrá si tiene que transmitir o no de acuerdo al tiempo que tenga que esperar, de este modo cada nodo transmite de forma independiente de los demás, lo cual evita la necesidad de sincronizarse con los demás nodos para reservar recursos y asignar prioridades de forma conjunta.

Otra diferencia importante es que se provee QoS a los paquetes o flujos de datos y no a los nodos que están transmitiendo, a diferencia del ruteo QoS el cual se encarga de hacer reserva de recursos de acuerdo a las capacidades de los nodos; por esta razón con 802.11e no hay necesidad de sincronizarse con los nodos de la ruta seleccionada.

Por otra parte, al ser tan dinámica, las MANET tienen caídas de enlaces muy frecuentemente, lo cual hace que se deban realizar procesos de retransmisión. Dichos procesos en el estándar 802.11e se realizan ejecutando algoritmos de backoff sin la necesidad de reenvíos de paquetes de manera global.

5. Diseño de los escenarios

Teniendo en cuenta los parámetros que se escogieron, y las razones por los cuales fueron seleccionados, se plantearon distintos tipos de transmisiones y también distintas cantidades de nodos. De esta forma se verá cómo cambia el comportamiento de la red a medida que se van agregando nodos y por consiguiente transmisiones.

El simulador elegido para este trabajo es Network Simulator 2 (NS-2) en la versión 2.34. NS-2 es uno de los simuladores más utilizados y los resultados son aceptados por la comunidad internacional de investigadores. Es una herramienta con licencia de distribución libre y ampliamente difundida en el ámbito académico.

Para la topología de la red, se consideró una MANET donde todos los nodos poseen las mismas características y ninguno se diferencia de otro en cuanto a canal de transmisión, interface de red, modelo de propagación de señal, etc. Todos los nodos están dispersos en una zona de 500 metros cuadrados dentro de la cual las estaciones se mueven libremente.

Se trabajo con escenarios de 20 y 100 nodos. Este cambio en la granularidad se plantea para analizar la

reacción de los estándares 802.11 y 802.11e al aumento de nodos y conexiones (por consiguiente aumenta la cantidad de transmisiones de paquetes de datos y de ruteo). Este cambio en la cantidad de nodos, junto con la movilidad que se produce en los mismos debe generar distintos niveles de sobrecarga de paquetes de ruteo, pues existen mayor cantidad de rutas y saltos por estudiar para decidir el mejor camino.

Además, el hecho de que los nodos se estén moviendo continuamente genera que la actividad de descubrimiento de rutas se realice repetidas veces, pues los cambios en la topografía hacen que los nodos que sirven de intermediarios queden fuera del alcance del conjunto de nodos que forma la ruta actual, lo cual produce que las estaciones involucradas en ella deban realizar tareas de descubrimiento de caminos, aviso de caída de rutas, etc. de forma repetida.

La cantidad de transmisiones en las simulaciones es proporcional a la cantidad de nodos. Se usó la relación de n/2 cantidad de transmisiones activas, donde n es la cantidad de nodos que se crearon. De esta forma, y siempre que n sea un número par todos los nodos estarán involucrados en al menos un flujo de datos punto a punto.

Como el objetivo es analizar el impacto del estándar 802.11e sobre tráfico heterogéneo, se considero que todos los nodos corren el mismo tipo de aplicaciones, las cuales obedecen a flujos ftp con paquetes de 1000B y CBR de video.

El protocolo de ruteo seleccionado para realizar las simulaciones es AODV (Ad-hoc On-demand Distance Vector Routing), debido a que el retardo que se produce por el rearmado de las tablas de ruteo tiende a estabilizarse cuando la granularidad de la red es alta (superior a los 10 y 15 nodos) [11], lo cual es propicio para los escenarios que se manejan en las simulaciones.

Otro factor importante es la movilidad, pues afecta de forma directa el rendimiento de los protocolos de ruteo de las MANET. Existen distintos tipos de comportamientos en el mundo físico en cuanto a lo que movimiento se refiere, para emular el movimiento de los usuarios dentro de una MANET existen los “Modelos de movilidad”, los cuales definen la velocidad y las trayectorias que los nodos recorrerán durante la simulación.

Para este trabajo se usaron dos modelos de movilidad, Random Waypoint Mobility Model (RWPM) y Random Walk Mobility Model (RWKM) [12].

6. Resultados obtenidos

6.1. Análisis del retardo extremo a extremo para tráfico heterogéneo

El primer resultado que se muestra en la figura 1, es el retardo con 20 nodos con movilidad Random Walk. La diferencia es muy notable cuando se implementa QoS a nivel MAC. Se registró un retardo promedio sin QoS es de 12,96 segundos (±8,35 segundos), mientras que cuando se implementó QoS se registró un promedio de 0,099 segundos (± 0,222 segundos). Para este escenario las aplicaciones que generan tráfico inelástico se verán afectadas en su performance.



Figura 1: Retardo con 20 Nodos, con RWPM

La figura 2 muestra los resultados cuando el tipo de movimiento es RWPM. En este caso, el retardo promedio cuando no se implementó QoS es de 5,98 segundos (±3,45 segundos), de lo contrario es de 0,009 (±0,024 segundos). Al igual que en el caso anterior, 802.11e no brinda ningún tipo de satisfacción a las aplicaciones que generan tráfico inelástico.

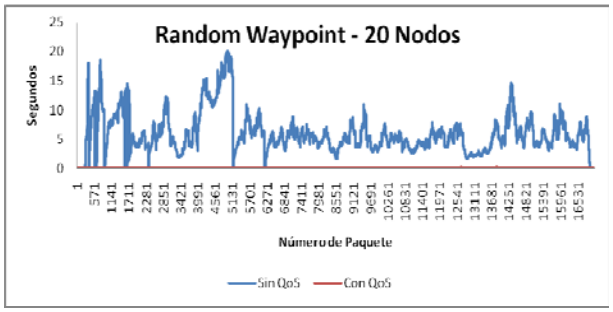


Figura 2: Retardo con 20 Nodos, con RWPM

La figura 3 muestra una comparativa de los retardos para las redes con QoS con distintos tipos de movimiento. Se puede notar una gran diferencia entre aquella red cuyo movimiento es Random Waypoint al de Random Walk.

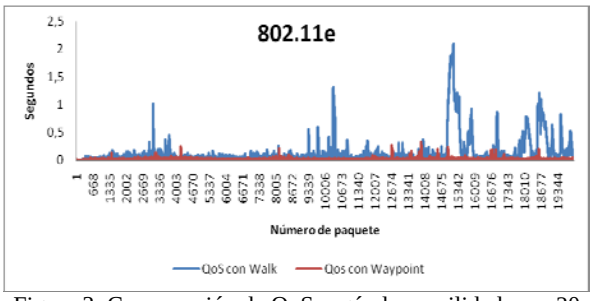


Figura 3: Comparación de QoS según la movilidad para 20 nodos

El retardo es más inestable y siempre es mayor cuando la red se simula con el modelo RWKM. Este resultado, es determinante para asegurar que el tipo de movimiento degrada la calidad de las aplicaciones pues las hace incurrir en más retardo.

En el caso de la figura 4, el retardo promedio registrado fue de 13,39 segundos (±7,05 segundos) sin QoS. Mientras que cuando se aplica QoS el tiempo de retardo promedio es de 1,146 segundos (±0,926 segundos). Si bien estos resultados indican que la red posee mejor desempeño cuando implementa 802.11e, comienza a generar inestabilidad sobre el final de la simulación, la cual provoca un jitter (diferencia de retardo) que es muy perjudicial para las aplicaciones que generan tráfico inelástico.

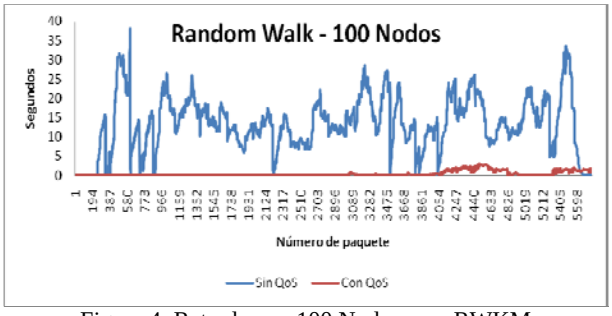


Figura 4: Retardo con 100 Nodos, con RWKM

Para el caso de la figura 5, el tiempo promedio cuando no se aplica QoS es de 11,15 segundos (±7,05 segundos), mientras que cuando se aplica QoS es de 0,737 segundos (±0,921 segundos). En contraposición al caso anterior, aquí la el retardo promedio es menos y no presenta inestabilidad en el.

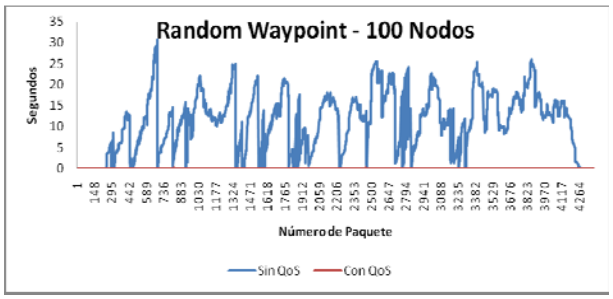


Figura 5: Retardo con 100 Nodos, con RWPM

La figura 6 muestra la comparativa entre los tipos de movimiento cuando se aplica QoS. En este caso, el tipo de movimiento Random Walk tiene un promedio de tiempo más alto, pero su desviación estándar es muy parecida a la de Random Waypoint. Desde el punto de vista del movimiento, se puede ver que Random Waypoint presenta mayor inestabilidad (jiter) en la zona marcada de la figura, esto tiene graves consecuencias cuando las aplicaciones son video interactivo y transferencia de archivos muy grandes.

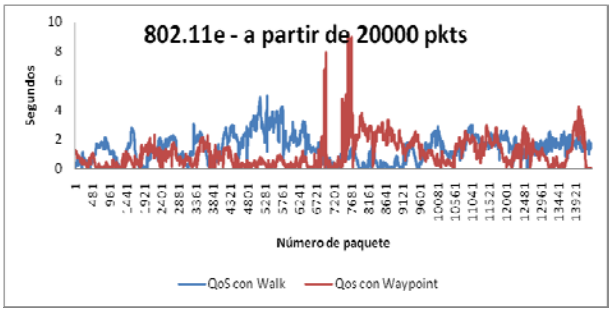


Figura 6: Comparación de QoS según la movilidad para 100 nodos

6.2. Análisis de la sobrecarga para tráfico heterogéneo

En la figura 7 se observa la sobrecarga de paquetes de ruteo cuando el tipo de movimiento es de Random Walk. Para el caso de la red sin QoS se registró una cantidad promedio de 73 paquetes de ruteo (±58 paquetes), lo cual indica una gran inestabilidad en la sobrecarga de la red. Para su contraparte se registró una cantidad promedio de 18 paquetes (± 7 paquetes).

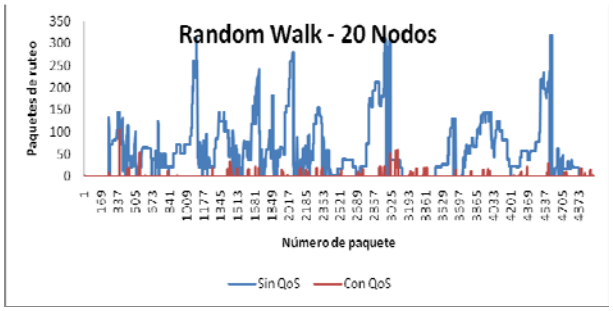


Figura 7: Sobrecarga con 20 Nodos, con RWKM

En la figura 8, en donde el movimiento es Random Waypoint, se obtuvieron los siguientes datos: cantidad promedio de 51 paquetes (±34 paquetes) sin QoS a nivel de MAC, y 11 paquetes (±2 paquetes) en promedio cuando se implementó 802.11e.

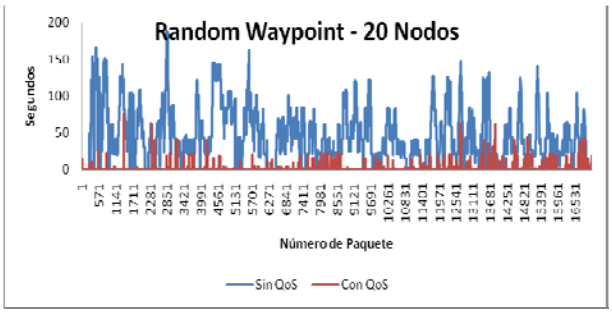


Figura 8: Sobrecarga con 20 Nodos, con RWPM

En la figura 9, se muestra la sobrecarga para las redes con QoS. Se observa que el tipo de movimiento impacta en la sobrecarga cuando se implementa 802.11e. para el caso Random Walk presenta más inestabilidad (véase la zona marcada) lo cual impactara en una reducción del ancho de banda y en el consiguiente aumento del retardo extremo a extremo.

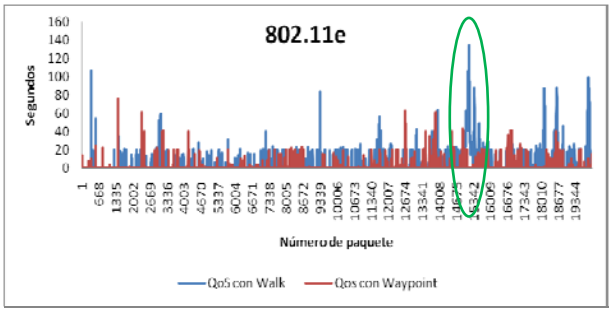


Figura 9: Comparación de QoS según la movilidad para 20 nodos

En la figura 10, con el movimiento del tipo Random Walk, se puede observar que la sobrecarga es muy inestable cuando no se aplica QoS. Para el caso de cuando se implementa QoS, la sobrecarga se vuelve inestable a medida que se envían paquetes. El promedio de paquetes de ruteo en el caso de la red sin QoS es de 2873 paquetes (± 1343 paquetes), mientras que para la red con QoS es de 1556 paquetes (±1153 paquetes). Estos resultados, generan un aumento de consumo de ancho de banda y una degradación en la performance de las aplicaciones.

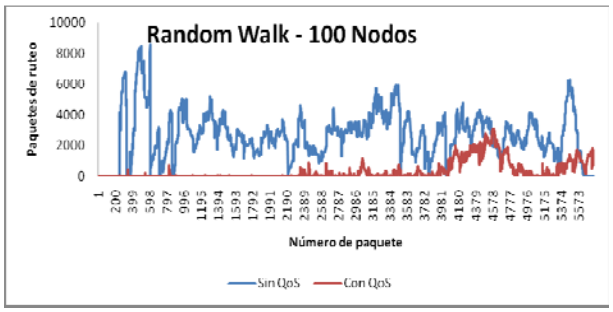


Figura 10: Sobrecarga con 100 Nodos, con RWKM

En el caso de las redes con movimiento Random Waypoint, figura 11, el promedio de paquetes de ruteo que se registró cuando el estándar 802.11e está implementado fue de 1441 paquetes (±1163 paquetes), mientras que cuando no se implementó se obtuvo un promedio de 2952 paquetes (±1551 paquetes). En este caso se puede ver, que existe una mayor estabilidad aunque la cantidad de paquetes de ruteo es mayor que para el caso anterior.

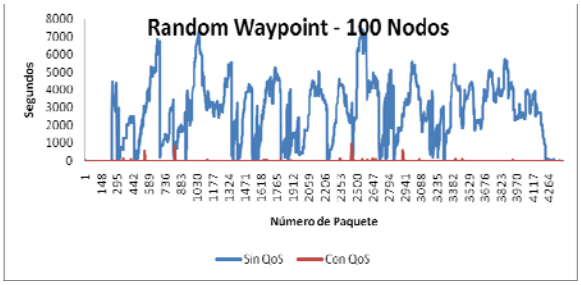


Figura 11: Sobrecarga con 100 Nodos, con RWPM

En la figura 12 se muestra la diferencia entre las redes con QoS implementado según el tipo de movimiento. Como se observa en la figura, existen momentos en la simulación en donde la cantidad de paquetes de ruteo es muy alta, provocando picos en el consumo de ancho de banda, los cuales generaran un retardo perjudicial para aplicaciones que generan tráfico inelástico.

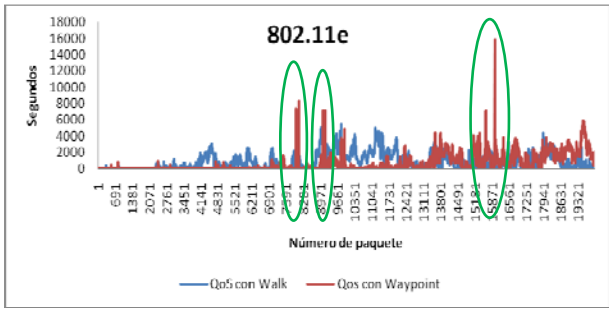


Figura 12: Comparación de QoS según la movilidad para 100 nodos

7. Conclusiones

En líneas generales se puede concluir que la implementación del estándar 802.11e produce mejores resultados en las redes que lo implementan, en lo que se refiere al retardo extremo a extremo. Sin embargo, teniendo en cuenta que la estabilidad del retardo es crucial cuando se trabaja con tráfico heterogéneo y considerando que el tráfico elástico se monta sobre TCP, la implementación de QoS a nivel de MAC no asegura que el tiempo de respuesta se comporte de forma estable. Esto se debe a las variaciones de retardo que se generan en el descubrimiento de rutas y rearmado de las mismas en ambientes con alta tasa de dinamicidad tengan un impacto negativo en el desempeño de las aplicaciones cuyo tráfico es inelástico.

Con respecto a la sobrecarga, el estándar 802.11e marca una gran diferencia entre las redes que lo implementan de aquellas que no lo hacen. Mientras que la granularidad sea baja la sobrecarga se mantiene estable y no alcanza grandes valores cuando se implementa el estándar de QoS a nivel de MAC. Cuando la granularidad de la red comienza a aumentar, la sobrecarga se vuelve inestable y alta. Las redes con alta granularidad, sin importar el tipo de movimiento que tengan los nodos, no son aptas para tráfico heterogéneo, debido a que existe una gran sobrecarga de ruteo, debido a la necesidad de calcular en forma reactivas las nuevas rutas. Esta sobrecarga no solo genera retardos en la transmisión del trafico sino consumo de ancho de banda y por consiguiente una disminución de la performance de la red.

Las aplicaciones del tipo FTP afectan drásticamente el rendimiento de aquellas que funcionan al estilo CBR. Entornos incluso con baja granularidad poseen alta sobrecarga de información de ruteo cuando se ejecutan, con el consiguiente aumento en el consumo de ancho de banda, lo cual genera un aumento del retardo extremo a extremo. Por esta razón, se puede concluir que este tipo de entorno es inapropiado para ambientes en donde coexistan tráfico elástico e inelástico.

En cuanto a la movilidad, tiene un efecto directo y drástico en los retardos de los paquetes de datos. El modelo de movilidad junto con la granularidad fueron los causantes de la gran inestabilidad que se detectó en los retardos en las simulaciones. En general, el modelo Random Walk fue el que generó mayores y más inestables retardos que su contraparte, el Random Waypoint Mobility Model.

Como conclusión final, se puede afirmar que para realizar una adecuada administración de QoS para tráfico heterogéneo en ambientes MANET, la implementación del estándar 802.11e, no es una solución adecuada pues genera una elevada e inestable sobrecarga en la red, lo cual implica un mayor consumo de ancho de banda y una disminución del throughput degradando la cantidad de recursos existentes.

8. Bibliografía

[1] Barbeau y Kranakis (2007). “Principles of Ad-Hoc Networking”. John Wiley and Sons.

[2] Murazzo, Rodríguez, Vergara, Carrizo, González, Grosso (2013). “Administración de QoS en ambientes de redes de servicios convergentes”. WICC 2013, Parana, Entre Rios, Argentina.

[3] Chakrabarti y Mishra (2001). “QoS Issues in Ad Hoc Wireless Networks”. IEEE Communications Magazine.

[4] Marrone, Robles, Murazzo, Rodriguez, Vergara (2011). “Administración de QoS en MANET”. WICC 2011, Rosario, Santa Fe, Argentina.

[5] Marrone, Barbieri, Robles (2011). “Tecnologías wireless y movilidad en IPv4/IPv6". XV Escuela Internacional de Informática, CACIC 2011.

[6] Stallings (2010). “Redes e Internet de Alta Velocidad. Rendimiento y Calidad de Servicio”, pág 16. ISBN: 978-84-205-3921-8, Prentice Hall.

[7] Murazzo, Rodríguez, Sheffer, Guevara (2014). “Soporte de QoS para tráfico heterogéneo en ambientes MANET”. WICC 2014, Tierra del Fuego, Argentina.

[8] Murazzo, Rodríguez, Villafañe, Grosso, Dávila (2013). “Análisis del impacto de implementación de 802.11e en redes MANET con tráfico CBT”. JAIIO (AST) 2013, Córdoba, Argentina.

[9] Murazzo, Rodríguez, Villafañe (2013). “Análisis de las prestaciones de 802.11e en redes MANET”. CACIC 2013, Mar del Plata, Buenos Aires, Argentina.

[10] Wietholter, Hoene (2003). “Design and verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26”. Techical University Berlin Telecommunication Networks Group.

[11] Murazzo, Rodríguez, Martínez (2009). “Evaluación y simulación del rendimiento de los protocolos de ruteo para MANET bajo restricciones de QoS”. WICC 2009, San Juan, Argentina.

[12] Mohapatra (2009). "Implementation and Comparison of Mobility Models In NS-2”. National Institute of Technology, Rourkela.

Simulación de Algoritmos de Control de Congestión en TCP bajo User Mode Linux

Rattalino Damián José, Crespo Aldo Abel
Universidad Nacional de La Pampa, Facultad de Ingeniería

Abstract

La experimentación con protocolos y servicios de aplicación bajo TCP/IP requiere de una costosa infraestructura de laboratorio que debe ser actualizada y/o renovada periódicamente, considerando la alta dinámica científica-tecnológica del área. Por lo general, es necesario crear topologías con múltiples redes y/o subredes pobladas de computadoras, y cuando este es el caso, la experiencia de laboratorio forzosamente debe ser grupal, con las desventajas que ello implica. Una alternativa superadora, que permite mejorar la calidad de enseñanza, es a través de la utilización de tecnologías para la virtualización de computadoras y dispositivos de interconexión, posibilitando la construcción de complejos escenarios de red sin otro requerimiento más que el de disponer de una computadora de medianos recursos. En esta dirección, el presente trabajo muestra como experimentar con algoritmos de control de congestión en TCP, utilizando topologías virtuales construidas en User Mode Linux (UML) [1]. UML es una tecnología que permite virtualizar computadoras y redes de computadoras sobre Linux.

Palabras Claves: TCP/IP, UML, Redes, Algoritmos Control de Congestión, Netem, Kprobes.

1. Introducción

Para el dictado de un curso típico de Redes de Computadoras, se requiere de las bases teóricas del área, para luego entrar a otra, en que el dictado se puede transformar es una conjugación muy estrecha entre teoría y práctica. El límite entre ambos modos de enseñanza suele ser el estudio exhaustivo de las funciones de la capa de enlace de datos respecto del modelo de referencia OSI. Luego, todo lo relativo al estudio de protocolos y servicios de aplicación de red puede abordarse desde la perspectiva simultánea antes mencionada. Mediante esta técnica, el alumno puede asimilar conceptos y reafirmarlos desde la experimentación. Si la experimentación es individual, el proceso de enseñanza mejora notablemente, pues desde la óptica docente, permite unívocamente determinar puntos de incertidumbre, que deben ser aclarados para que el proceso de aprendizaje converja al entendimiento definitivo.

Un laboratorio de experimentación basado en tecnologías para la virtualización de computadoras y dispositivos de interconexión que permita la construcción de complejas topologías de red, es la única vía para lograr la modalidad de enseñanza antes descrita. Ha de notarse que las empresas más renombradas en la producción de dispositivos relacionados a las redes de cualquier envergadura, proveen soluciones propietarias a sus clientes para la virtualización de sus productos. De esta forma, un administrador, antes de realizar modificaciones en la configuración real de una infraestructura de red, lo hace sobre una réplica virtual, y si la nueva configuración responde a sus expectativas, finalmente las aplicara en la infraestructura real.

Si bien existen múltiples opciones *open source* [2] respecto a tecnologías de virtualización para construcción de redes de computadoras, en el ámbito de este trabajo se utilizará User Mode Linux, de ahora en más por su acrónimo UML. UML es un núcleo Linux, modificado para ser compilado sobre GNU/Linux. Las computadoras o dispositivos virtuales se soportan sobre un núcleo Linux compilado para la arquitectura UML, que se ejecuta como un proceso de usuario sobre el núcleo Linux convencional en la computadora anfitriona. De esta manera, es posible crear redes virtuales que respondan a topologías arbitrariamente complejas.

Absolutamente todo lo relativo al conjunto de protocolos TCP/IP puede ser experimentado con UML, y en esta

dirección, este articulo muestra como experimentar con algoritmos de control de congestión. El control de congestión aplicable a flujos de bytes sobre conexiones TCP es uno de los tópicos que en un curso típico de redes de computadoras se lo aborda por lo general, desde una perspectiva teórica, y este artículo contribuye con un novedoso método de experimentación que no requiere de la utilización de complejos simuladores de eventos discretos, tal como NS [3] y OPNET [4], ni de la variada cantidad de dispositivos necesitados para construir una topología real. A tal fin, se virtualizará con UML una topología de red apta para él estudió de algoritmos para el control de congestión, posteriormente se llevaran a cabo los experimentos considerando las características típicas de una infraestructura de red WAN en lo que se refiere a retardos y variaciones de retardo *(jitter)*.

Este trabajo está organizado del siguiente modo. La sección 2.0 tratará sobre aspectos teóricos de algunos de los algoritmos de control de congestión escogidos para la simulación con UML, en la sección 3.0 se describirá la topología de experimentación implementada con UML y las herramientas utilizadas para llevar a cabo el análisis experimental, en la sección 4.0 se presentaran los resultados experimentales y finalmente en la sección 5.0 se presentaran las conclusiones del artículo.

2.0 Control de Congestión

2.1 Introducción

El termino congestión hace referencia a un fenómeno no deseado, generado en una red, o en una parte de su infraestructura, cuando distintas fuentes inyectan un volumen de tráfico superior al que los dispositivos de conmutación de paquetes pueden procesar. En octubre de 1986 se evidenció por primera vez un colapso debido al fenómeno de congestión y a partir de ese evento, la situación se agravaría en la medida que más computadoras se conectaban a Internet.

2.2 TCP Tahoe

En 1988 y a raíz de la problemática descrita, Van Jacobsen propuso en [5], un mecanismo de control de congestión basado en el concepto de ventanas de congestión, *cwnd*, y en la definición de tres algoritmos: arranque lento *(slow start)*, prevención de congestión *(congestion avoindace)* y retransmisión rápida *(fast retransmit)*.

A cada extremo en una conexión TCP se le asigna una ventana de congestión, *cwnd*, indicativa del número de segmentos que puede enviar a la red sin esperar por reconocimientos explícitos desde el receptor (ACK). Al inicio, en la fase de operación de arranque lento *(slow start)*, *cwnd* se configura en uno o dos segmentos de máximo tamaño (MSS) y su valor se incrementará en un segmento por cada reconocimiento recibido desde su par receptor (ACK). Durante la fase operativa de arranque lento, *cwnd* experimentará un crecimiento exponencial, tal como se ilustra en la Figura 1.

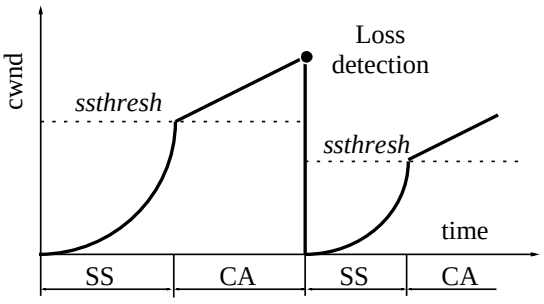


Figura 1: TCP Tahoe

Cuando *cwnd* alcanza el valor de la variable "umbral de arranque lento" o *slow start threhold* (*ssthresh)*, el mecanismo de control de congestión conmuta al modo *congestion avoindance* y *cwnd* se incrementa linealmente a razón de un segmento por grupo de reconocimientos (ACKs), que se corresponden con el valor actual de la ventana de congestión (Figura 1).

Se abandona el estado de *congestion avoindace* cuando el emisor TCP detecta una pérdida por expiración de *timeout* o por

la recepción consecutiva de tres reconocimientos duplicados (ACKs). En ambos casos el emisor reconfigura la variable *ssthrehold* al valor *cwnd/2*, retransmite el segmento perdido *(fast retransmit)*, y conmuta a la fase de operación *slow start* (SS) actualizando *cwnd=1*.

2.3 TCP Reno y TCP NewReno

TCP Reno surgió en 1990 como el sucesor de TCP Tahoe y fue implementado por primera vez en el sistema operativo 4.3BSD Unix. Si bien mantuvo las características de su antecesor, incorporó el algoritmo de recuperación rápida o *fast recovery* [6]. El algoritmo de recuperación rápida (FR) evita el ingreso a la fase de arranque lento (SS), cuando se detectan tres reconocimientos duplicados (ACKs) en la fase prevención de congestión (CA) y bajo esta circunstancia el emisor ejecuta las siguientes acciones: reconfigura la variable *ssthresh* al valor *cwnd/2*, retransmite el segmento perdido y actualiza la ventana de congestión a *cwnd=ssthresh+3*, finalmente abandona la fase de recuperación rápida para regresar a la fase de prevención de congestión (CA).

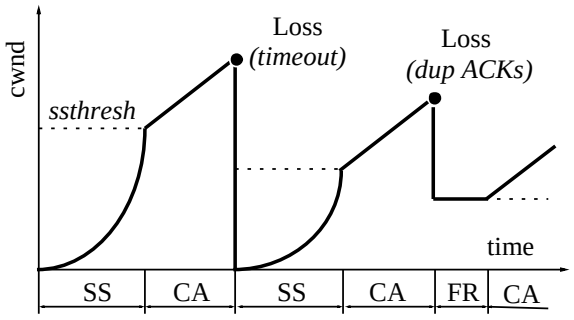


Figura 2: Dinámica de cwnd en Reno

Una de las vulnerabilidades de TCP Reno se manifiesta cuando en la fase *fast recovery* (FR) se origina un evento de perdida de múltiples segmentos para el valor actual de *cwnd*. Bajo esta circunstancia TCP Reno alternará su operación entre las fases CA/FR, y provocará una reducción sucesiva del valor de la variable umbral de arranque lento *(ssthresh)*, y de la ventana de congestión *(cwnd)*. Como consecuencia se produce una importante disminución en el desempeño *(thrughput)* de la conexión TCP.

Atentos a la problemática descrita, Floyd y otros en [7] y [8] introdujeron un simple refinamiento al algoritmo *fast recovery* (FR). En la fase *congestion avoindance* (CA), al recibir tres ACK´s duplicados, ingresa a la fase *fast recovery*, pero a diferencia de TCP Reno, no abandona la fase *fast recovery* hasta que se produzca la confirmación *(acknowledged)* de todos los segmentos de la ventana de congestión *(cwnd)*. Más formalmente, TCP NewReno utiliza una variable de estado adicional para registrar el número de secuencia del último segmento TCP enviado antes de entrar a la fase *fast recovery*.

TCP NewReno resuelve el problema de bajo desempeño ante eventos de múltiples pérdidas en la ventana de congestión *(cwnd)* pero como contrapartida emplea un tiempo excesivo en el proceso de recuperación, pues le toma un *Round Trip Time* (RTT) recuperar cada uno de los segmentos perdidos en la ventana de congestión. Diversas estrategias se formularon para mejorar el desempeño de TCP NewReno en la fase *fast recovery* y una excelente referencia que investiga exhaustivamente numerosas propuestas puede encontrarse en [9].

2.4 TCP BIC

El desempeño del protocolo TCP basado en TCP Reno y TCP NewReno no es satisfactorio en redes de alta velocidad, con elevados RTT entre los host que se comunican, y el factor limitante es el comportamiento conservativo de ajuste de la ventana de congestión que gobierna la tasa de transmisión del emisor [10]. Por tal razón surgieron una serie de propuestas basadas en reformular la vía con la cual TCP adapta la ventana de congestión.

Un intento exitoso para crear un control de congestión que pueda escalar en

redes con alto valor BDP *(bandwidth delay product)* fue propuesto por Xu y otros en [11]. BIC *(Binary Increase Congestión control)* extiende TCP NewReno con una fase adicional denominada de convergencia rápida *(Rapid Convergence or RC)*. La fase RC, tiene como objetivo descubrir mediante una rápida búsqueda binaria, el valor óptimo de la ventana de congestión en función de los recursos actuales de la red. El proceso de búsqueda binaria es como se ilustra en la Figura 3.

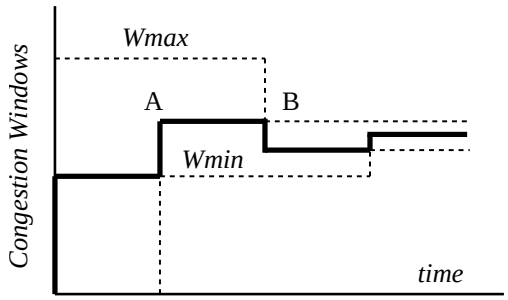


Figura 3: Búsqueda binaria de cwnd

Inicialmente se establecen las variables *Wmin=1* y *Wmax* en un valor arbitrariamente alto. Si la red entrega correctamente todos los segmentos emitidos (el emisor recibe todos los ACKs correspondientes al último RTT), la ventana de congestión se actualiza al valor promedio entre *Wmax* y *Wmin* (A en la Figura 3) y *Wmin* adquiere el tamaño de la ventana de congestión previa. Cuando se detecta una pérdida, BIC reduce *Wmax* al valor actual de la ventana de congestión (B en la Figura 3) y como en TCP NewReno ingresa a la fase de recuperación rápida (FR).

Para incrementar la tasa de convergencia, en ambientes de bajo nivel de pérdidas, BIC reduce el coeficiente de decremento multiplicativo desde 0.5 a 0.125, para lograr mayor velocidad de respuesta.

Las características del algoritmo de búsqueda binaria se traducen en tiempos de convergencia muy bajos que pueden provocar degradación en conexiones TCP. Si la ventana de congestión se incrementa drásticamente, un grupo significativo de segmentos pueden perderse. Por esta razón BIC implementa un algoritmo de arranque lento restringido [12] que limita el incremento en cada paso de *slow start* a un valor máximo de 100 segmentos. En la misma dirección, BIC impone límites al crecimiento de la ventana de congestión en la fase de convergencia rápida cuando el rango de búsqueda binaria es muy amplio, y no permite incrementar la ventana de congestión más allá de un valor predefinido S_{max}.

En el caso opuesto, cuando el rango de búsqueda binaria es pequeño, y por tanto cercano al valor óptimo, BIC define el valor constante S_{min} como un quantum de incremento de la ventana de congestión. Cuando el valor actual de la ventana de congestión alcanza el valor optimo, o lo excede, BIC conmuta a la fase de arranque lento restringido, no limitado por el valor umbral de *ssthresh (slow start threshold)*. El propósito de esta acción es descubrir un nuevo límite superior para luego reiniciar la fase de búsqueda binaria. La Figura 4 ilustra la dinámica de la ventana de congestión de una conexión para TCP BIC.



Figura 4: Dinámica de cwnd en BIC

2.5 TCP CUBIC

Rhee y Xu en [13] observaron que TCP BIC presenta problemas cuando dos o más flujos de bytes con diferencias significativas en sus RTT compiten por el ancho de banda en un enlace de características restrictivas *(bottleneck link)*. Concretamente en [13] se demostró que flujos de bytes con pequeños RTT escalan rápidamente su ventana de congestión al

valor óptimo y consumen los recursos de la red, en detrimento de flujos con RTT mayores. A raíz de las observaciones experimentales, Rhee y Xu [13] propusieron el mecanismo de control de congestión CUBIC, que constituye una mejora de TCP BIC e implementa una función para el crecimiento de la ventana de congestión en forma independiente del RTT para cada conexión en particular.

TCP CUBIC define una ventana de congestión *(w)* como una función cubica del tiempo transcurrido desde el último evento de congestión (Δ):

$$w = C\left(\Delta - \sqrt[3]{\frac{\beta * w_{max}}{C}}\right)^3 + w_{max}$$

Donde C es una constante predefinida, β es un coeficiente multiplicativo decreciente en la fase de recuperación rápida, y w_{max} es el tamaño de la ventana de congestión inmediatamente antes de que se detecte una perdida. La función presenta un rápido crecimiento cuando w es pequeña respecto de w_{max}, y es muy conservativa cuando w se aproxima al valor de w_{max} tal como se ilustra en la Figura 5.

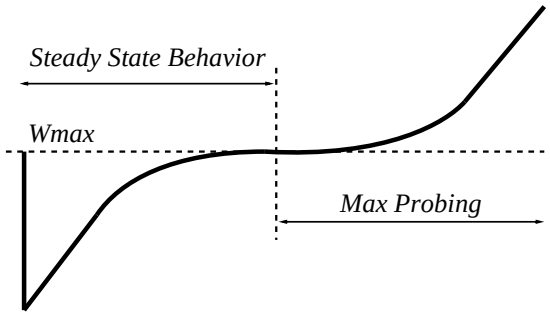


Figura 5: Descripción función cubica

La Figura 6 muestra la dinámica de la ventana de congestión de TCP CUBIC. De la observación se puede deducir que, durante el paso inicial "1" se desconoce w_{max} y para descubrir su valor se emplea la sección derecha de la función cubica en la Figura 5 *(Max Probing)*, esta etapa finaliza al detectarse una perdida. A continuación, en la fase 2, se emplea la sección izquierda de la función cubica, que implica una tasa de crecimiento en la ventana de congestión menor al de la fase 1. La etapa 3 incluye ambas etapas de la función cubica y finaliza cuando se produce una perdida; en este caso particular se actualiza el valor umbral de w_{max}

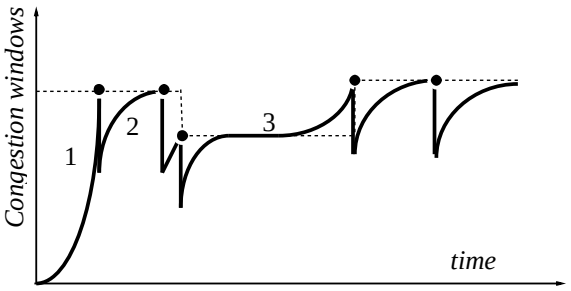


Figura 6: Dinámica de cwnd en Cubic

Por último, TCP CUBIC es el mecanismo de control de congestión más utilizado desde que fue incluido por defecto en las distintas versiones de núcleos Linux desde 2006.

3.0 Análisis Experimental

Es importante destacar que el objetivo de la experimentación no está dirigido a realizar valoraciones sobre medidas de desempeño entre distintos algoritmos de control de congestión. El propósito central del artículo es convalidar un novedoso e inédito método, para experimentar con algoritmos de control de congestión, basado en la construcción de topologías virtuales utilizando *User Mode Linux.*

El método propuesto es de fácil e inmediata implementación, y puede simular el comportamiento de enlaces WAN con sus características inherentes (retardos, *jitter*, tasa de pérdidas y restricciones de ancho de banda). Permite ahorrar tiempo en la obtención de resultados y es apto para obtener medidas de desempeño para cualquiera de los algoritmos de control de congestión incluidos como módulos en las distintas versiones de núcleos Linux.

A continuación en la sección 3.1 se presentara el laboratorio de experimentación a implementar bajo *User*

Mode Linux, se describirán los elementos constitutivos de la topología seleccionada y se expondrán los objetivos finales de la experimentación. En la sección 3.2 se abordaran aspectos de configuración del núcleo Linux en la computadora anfitriona, y del núcleo Linux en los dispositivos y computadoras virtuales.

3.1 Topología UML

La Figura 7 ilustra la topología a implementar en UML y aunque existe una extensa variedad de literatura sobre como ejecutar el escenario de red virtual, la referencia obligada es la de Jeff Dike, autor de la tecnología [14].

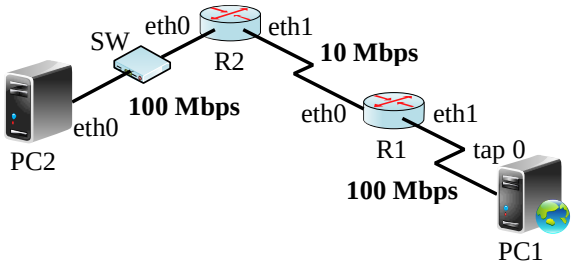


Figura 7: Topología virtual en UML

Los dispositivos de conmutación de paquetes R1 y R2 *(routers)*, la computadora PC2 y el dispositivo de conmutación de tramas, SW, constituyen componentes virtuales UML, soportados por el sistema operativo Linux en el host anfitrión PC1.

La vinculación entre los dispositivos virtuales y el host anfitrión (PC1) se define durante la creación de la interfaz de red *eth1* en R1, que al mismo tiempo genera una interfaz TAP en el host anfitrión PC1.

Una interfaz TAP es un enlace virtual punto a punto entre dispositivos Ethernet, que en vez de recibir tramas desde el medio físico, lo hace desde un programa en espacio de usuario; y en vez de transmitir tramas vía el medio físico, las envía al mismo programa en espacio de usuario. Es importante destacar que el núcleo Linux del host anfitrión debe tener habilitado el modulo *TUN/TAP Support* en la sección de configuración, soporte de dispositivos de red *(Network Device Support).*

Antes de llevar a cabo el trabajo experimental se deben realizar una serie de tareas que incluyen, la configuración de la capacidad de los enlaces tal como lo expresa la Figura 7, la utilización de herramientas de software para simular el RTT de una red de área amplia (WAN), la selección de un generador de trafico TCP para establecer una conexión entre los hosts extremos de la topología, y repetir la experiencia con los algoritmos de control de congestión TCP NewReno, TCP BIC y TCP CUBIC.

3.2 Consideraciones de configuración

Tal como se anticipó en la sección 1.0 de este artículo, *User Mode Linux* requiere de un núcleo Linux, configurado y compilado bajo herramientas GNU, para la arquitectura UML. Este núcleo, servirá de soporte para computadoras y dispositivos virtuales y se ejecutará como un programa en el plano de usuario del sistema operativo de la computadora anfitriona que posee su propio núcleo.

Para experimentar con algoritmos de control de congestión se necesitará disponer de un conjunto de variables en el núcleo Linux, y la vía para realizarlo es a través de *tcpprobe*, un módulo que posibilita el acceso a las variables de estado en una conexión TCP. El mecanismo opera insertando una sonda de prueba no invasiva *(kprobe)* en el buffer de recepción *tcp_recv*. Mediante este procedimiento se puede capturar en forma dinámica todas las variables necesarias de una conexión TCP, para cualquier algoritmo de control de congestión.

La tecnología de virtualización UML inhabilita el módulo *tcpprobe* y esto obliga a que el emisor TCP en la conexión entre los hosts extremos de la Figura 7 sea obligatoriamente PC1, que se soporta sobre el núcleo Linux del host anfitrión. En

síntesis, el núcleo Linux del host anfitrión (PC1) debe ser compilado con el módulo *tcp_probe* habilitado.

Para formar tráfico *(traffic shaping)* en el ambiente virtual, y limitar el ancho de banda de acuerdo a los valores descritos en la Figura 7, es necesario habilitar en ambos núcleos Linux, en la sección de configuración"*QoS and/or fair queueing"* el módulo TBF *(Token Bucket Filter).* TBF es una sencilla disciplina de filas, que no distingue entre clases de tráfico *(clasless)* y que consiste de un *buffer* al que ingresan piezas virtuales de información o *tokens*, a una tasa constante o *token rate* (Figura 8). Tal como se ilustra en la Figura 8, la salida de un *token* desde el *bucket* habilita la partida de un segmento TCP. Fijando la tasa de *tokens* se controla el ancho de banda de un enlace [15].

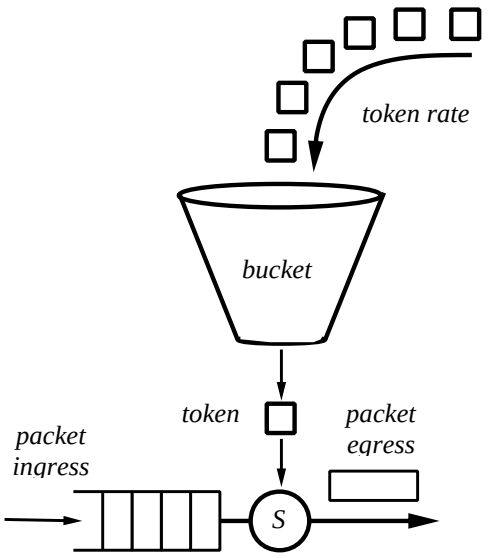


Figura 8: Modelo de disciplina TBF

Para emular el comportamiento de un enlace WAN en el ambiente virtual, es necesario habilitar en el núcleo Linux (soporte de los dispositivos virtuales), la opción *Network Emulator* o *NetEm*[16]. Para configurar *NetEm* se necesita del comando, tc *(traffic control),* que es parte del paquete de software *iproute2* [17].

Aunque NetEm posibilita una gran variedad de opciones respecto de su funcionalidad, en el marco de este artículo sólo se utilizará para generar retardos y deviaciones del retardo (*jitter),* para replicar en el ambiente virtualizado las condiciones de un enlace WAN cuyas características topológicas son similares a las de la Figura 7 y sus características temporales son: RTT promedio 50ms con ± 6 ms de *jitter*.

El objetivo perseguido es replicar en el ambiente virtual las condiciones de una red de área amplia (WAN). No se inducirán pérdidas mediante NetTem, pues se pretende que los algoritmos de control de congestión funcionen en base la limitación originada por la restricción del ancho de banda del enlace WAN (10 Mbps), considerando que el medio físico utilizado es fibra óptica y el nivel de pérdidas de segmentos es despreciable.

Un aspecto a tener en cuenta para lograr granularidad en los retardos introducidos mediante NetEm es relativo a la opción de configuración CONFIG_HZ en el núcleo Linux de los dispositivos virtuales. Esta opción determina la frecuencia con que el núcleo Linux atiende a los eventos generados por procesos en el plano de usuario. El valor por defecto de CONFIG_HZ en núcleos compilados bajo GNU para la arquitectura UML es 100 Hz, y sólo se puede modificar mediante la aplicación del código referenciado en [18]. Si se busca granularidad en el orden de 1 ms, el valor de CONFIG_HZ debe ser modificado a 1000 Hz.

Una cuestión relativa a los dispositivos de conmutación de tramas *(switchs)* utilizados para construir topologías virtuales. UML puede emplear una de dos herramientas de software para implementarlos, *uml_switch* [19] o *vde_switch* [20]. En el trabajo experimental se utilizó *vde_switch* al comprobar que no presentaba perdidas de tramas en condiciones de tráfico intenso. Por el contrario la utilidad *uml_switch* ante

idénticas circunstancias presentó perdidas de tramas (capa 2) y este factor afectaría el comportamiento de los algoritmos de control de congestión experimentados.

Para generar tráfico entre ambos extremos de la topología de la Figura 7 se utilizó Iperf [21], con PC1 como emisor y PC2 como el receptor de segmentos TCP. A continuación en la sección 4, se presentaran las características de hardware y del sistema operativo de la computadora anfitriona y las características de software de las computadoras y dispositivos virtuales; y se presentaran los resultados de la simulación.

4.0 Resultados Experimentales

Las pruebas experimentales para la simulación de los algoritmos de control de congestión se realizaron en una computadora con las siguientes características: procesador AMD (FX)-8128 de 8 núcleos con 8 GB de memoria RAM, ejecutando un sistema operativo GNU/Linux Ubuntu v10.04 basado en la versión del núcleo 2.6.32.28. Las computadoras y dispositivos virtuales UML, se ejecutaron sobre un sistema operativo GNU/Linux Slackware v12 soportado por la versión de núcleo Linux 3.2.59, y 1GB de memoria para cada dispositivo virtual.

Las simulaciones se realizaron, en un todo de acuerdo con la topología ilustrada en la Figura 7, considerando un RTT de 50 ms, un *jitter* de ± 6 ms, sin introducir perdidas ni duplicaciones de segmentos. Los segmentos generados por Iperf son de máxima longitud (MSS) para un MTU de Ethernet de 1500 bytes.

La Figura 9 muestra los resultados de la simulación para el algoritmo de control de congestión TCP New Reno. A diferencia de lo analizado en la sección 2.3, el valor de inicial de la variable umbral de arranque lento es elevado *(ssthresh)* y la ventana de congestión crece exponencialmente en la fase *slow start* hasta que se detecta la pérdida de un segmento por la recepción de tres ACKs duplicados, a continuación se actualiza el valor de variable *ssthresh*, TCP NewReno reingresa a la fase de arranque lento *(slow start)* y aproximadamente a los 10s el crecimiento de la ventana de congestión *(cwnd)* alcanza el valor de *ssthresh* y conmuta a la fase de prevención de congestión (CA). En esta fase de operación *cwnd* se incrementa de acuerdo a lo expresado en la sección 2.3 y en ausencia de perdidas permanece en CA.

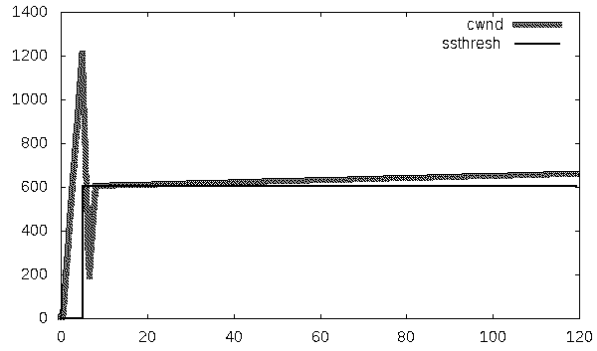


Figura 9: Simulación NewReno

La Figura 10 ilustra el resultado de la simulación para TCP BIC. Tal como se observa en la gráfica, el algoritmo tiene una fase de descubrimiento rápido al inicio, para detectar la capacidad de la red para la conexión. La variable *ssthresh* se corresponde con el parámetro w_{min} estudiado y descrito en la sección 2.4, y su dinámica es de rápida convergencia, mostrando su naturaleza de búsqueda binaria.

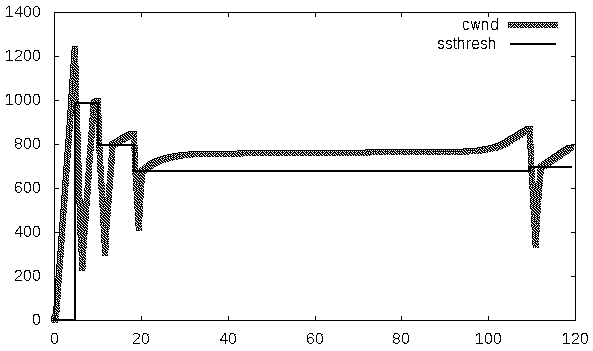


Figura 10: Simulación BIC

El comportamiento dinámico del algoritmo de control de congestión TCP Cubic se muestra en la Figura 11, tal como se describió en la sección 2.5, la simulación muestra una etapa inicial de descubrimiento del límite del ancho de banda de la red, al que se aproxima utilizando la parte derecha de la función cubica que sustenta el funcionamiento del algoritmo. Una vez que se produce una condición de pérdida, el algoritmo disminuye su ventana de congestión *(cwnd)* y se aproxima a *ssthresh* utilizando la parte izquierda de la función cubica, tal como fue tratado en la sección 2.5.

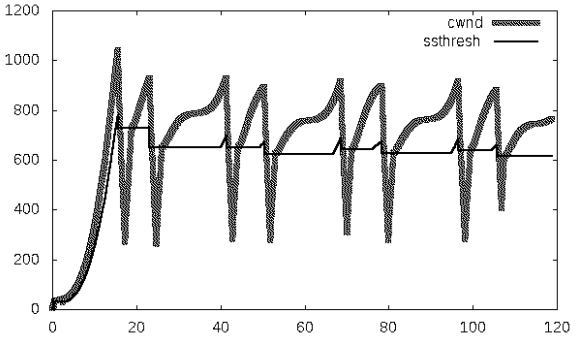


Figura 11: Simulación Cubic

5.0 Conclusiones

A pesar de la extensa variedad de artículos relativos al tópico control de congestión, no existen al respecto antecedentes de implementaciones ejecutadas íntegramente en ambientes virtualizados. Ello significa que el método de experimentación basado en *User Mode Linux*, para la simulación de algoritmos de control de congestión es inédito, y de gran utilidad para el estudio, análisis, y comparación de cualquiera de los algoritmos implementados en las distintas versiones de núcleos Linux.

El método propuesto es de fácil e inmediata implementación, y puede simular el comportamiento de enlaces WAN con sus características inherentes (retardos, *jitter*, tasa de pérdidas y restricciones de ancho de banda).

Permite ahorrar tiempo en la obtención de resultados y es apto para obtener medidas de desempeño.

El esquema propuesto no requiere de la utilización de complejos simuladores de eventos discretos, ni de la variada cantidad de dispositivos necesitados para construir una topología real; solo se necesita disponer de una computadora de medianos recursos para obtener idénticos resultados.

Finalmente, y para concluir, los resultados obtenidos vía simulación bajo UML para los mecanismos de control de congestión TCP NewReno, TCP BIC y TCP Cubic, se corresponden exactamente con los principios teóricos que rigen la dinámica de los algoritmos tratados en la sección 2 del artículo.

6.0 Referencias

[1] The UML Kernel Home Page; http://user-mode-linux.sourceforge.net

[2] O. S. Initiative, http://opensource.org/docs/osd

[3] NS, http://www.isi.edu/nsnam/ns/

[4] OPNET, http://www.riverbed.com/

[5] Jacobson, V., “Congestion avoidance and control”. In Symposium Proceedings on Communications Architectures and Protocols (Stanford, California, United States, August 16 - 18, 1988). V. Cerf, Ed. SIGCOMM ‘88. ACM, New York, NY, 314-329.

[6] Jacobson, V., "Modified TCP Congestion Avoidance Algorithm", end2end-interest mailing list, April 30, 1990.

[7] S. Floyd and T. Henderson, “RFC2582 the New Reno modification to TCP’s fast recovery algorithm”, RFC, 1999

[8] S. Floyd, T. Henderson, and A. Gurtov, “RFC3782 the NewReno modification to TCP’s fast recovery algorithm”, RFC, 2004.

[9] Alexander Afanasyev, Neil Tilley, Peter Reiher, and Leonard Kleinrock. “Host-to-Host Congestion Control for TCP”. IEEE Communications Surveys & Tutorials, 12(3):304–342, 2010.

[10] Sangtae Ha, Yusung Kim, Long Le, Injong Rhee, and Lisong Xu, “A step toward realistic evaluation of high-speed TCP protocols”, 2006.

[11] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control (BIC) for fast long-distance networks, Proceedings of IEEE INFOCOMM 2004, vol. 4, March 2004, pp. 2514-2524.

[12] Floyd, S., "Limited Slow-Start for TCP with Large Congestion Windows", RFC 3742, March 2004

[13] L. Xu, and I. Rhee, "CUBIC: A New TCP-Friendly High-Speed TCP Variant".

[14] Dike Jeff, "User Mode Linux", Publisher: Prentice Hall, April 12, 2006

[15] QoS (Quality of Service) concepts, http://lartc.org/howto/lartc.cookbook.fullnat.intro.html

[16] Stephen Hemminger, "Network Emulation with NetEm", In Linux Conf AU, 2005

[17] IPROUTE2, Linux Foundation Page, http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2

[18] Config clock in User Mode Linux, http://ehc.ac/p/mrvopensource/linux-ppc-2.6/ci/7281ff952c7b3cbefb14847460e5fe73a2d240e4

[19] Switch daemon for UML - uml_switch, http://manpages.ubuntu.com/manpages/lucid/man1/uml_switch.1.html

[20] VDE - Virtual Distributed Ethernet, http://vde.sourceforge.net/

[21] http://sourceforge.net/projects/iperf/, Iperf - The TCP/UDP, Bandwidth Measurement Tool, 2013.

Datos de Contactos:

Damián Rattalino, Facultad de Ingeniería de la Universidad Nacional de La Pampa, Código Postal 6360, damianwt@hotmail.com

Abel Crespo, Facultad de Ingeniería de la Universidad Nacional de La Pampa, Código Postal 6360, crespoa@ing.unlpam.edu.ar

DNSCurve, una alternativa para la seguridad en el Sistema de Nombres de Dominio.

Ernesto Sánchez
C.I.D.I.A.
Universidad Nacional de Salta
Salta, Argentina, 4400
esanchez@cidia.unsa.edu.ar

Daniel Arias Figueroa
C.I.D.I.A.
Universidad Nacional de Salta
Salta, Argentina, 4400
daaf@cidia.unsa.edu.ar

Sergio Rocabado
C.I.D.I.A.
Universidad Nacional de Salta
Salta, Argentina, 4400
srocabad@cidia.unsa.edu.ar

Verónica Aguero
C.I.D.I.A.
Universidad Nacional de Salta
Salta, Argentina, 4400
Vero_m13@hotmail.com

Gustavo Molina
C.I.D.I.A.
Universidad Nacional de Salta
Salta, Argentina, 4400
gustavo_molina323@hotmail.com

Abstract

To cover the DNS protocol safety aspects, DNSCurve is presented as an alternative. It uses the hierarchical structure of itself, to propagate trust by incorporating public / private keys, allowing encrypt and authenticate data packets exchanged between Resolvers and Authoritative Servers. The protocol was designed by Daniel J. Bernstein and it is based on the use of elliptic curve cryptography, more precisely, Curve25519. This paper presents results on the evaluation of this alternative, as part of the research project No. 1223/0: "Extensiones de Seguridad para el Sistema de Nombres de Dominio" (Research Council, UNSa). Aspects related to the implementation of DNSCurve, and evaluation of performance through the use of a software prototype developed for the simulation of traffic will be exposed. That prototype allowed to obtain statistics based on DNS queries. Their analysis and conclusions about the performance obtained on a server with these features will also be included. Currently, DNSCurve has not received yet the diffusion and impulse from organizations like ICANN, Verisign, IANA, among others, so it was very difficult to consider a protocol to implement on a large scale. However, its implementation is suggested as part of a set of solutions, in order to cover all aspects of safety Domain Name System.

1. Introducción

Al igual que otros protocolos que se diseñaron en los primeros tiempos de Internet, el Sistema de Nombres de Dominio carece de medidas de seguridad como la integridad, autenticación y confidencialidad. Tales aspectos tuvieron que ir añadiéndose a posteriori, en algunos casos, se rediseño el protocolo y en otros como es el caso de DNS, por medio de extensiones de seguridad.

Por su naturaleza de sistema público y por considerarse parte esencial para el funcionamiento de Internet, es que organizaciones y particulares, han puesto especial énfasis en la implementación de políticas y prácticas destinadas a dotar de seguridad al servicio proporcionado por DNS.

A la fecha, las dos alternativas presentes son; las Extensiones de Seguridad para DNS (DNSSEC) y DNSCurve. Con respecto a la primera, tales extensiones, proveen autenticación del origen e integridad de los datos intercambiados a través del protocolo DNS. Las mejoras que ofrece DNSSEC radican principalmente en el uso de una jerarquía de "firmas criptográficas" que permite proteger el flujo de información intercambiado entre Servidores Autoritativos, Servidores DNS Recursivos y Clientes DNS. [1]. En base a lo expuesto anteriormente, y según se describe en el RFC 4033, se observa una de las limitaciones presentes en el diseño de estas extensiones de seguridad, la misma radica en la ausencia de confidencialidad del flujo de datos intercambiados en el proceso de resolución de nombres, ya que una respuesta DNSSEC es autenticada pero no encriptada. Así también, y en base a las investigaciones realizadas, se puede considerar como una limitación, que para garantizar la autenticación e integridad, es necesario la construcción de una "cadena de confianza", proceso mediante el cual cada nodo involucrado en el proceso de resolución, debe "confiar" en el nodo inmediatamente superior, hasta llegar a los nodos raíz.

Por otro lado, la alternativa DNSCurve [2] se presenta como un nuevo protocolo de seguridad para DNS, que utiliza la estructura jerárquica del mismo, para propagar la confianza mediante la incorporación de claves públicas, permitiendo encriptar y autenticar los paquetes de datos intercambiados entre Resolvers y Servidores Autoritativos [3]. El protocolo fue diseñado por Daniel J. Bernstein y está basado en el uso de criptografía de curva elíptica, más precisamente, Curve25519.

El propósito de DNSCurve es cubrir los siguientes aspectos de seguridad:

- Confidencialidad, mediante la encriptación de todas las consultas y sus correspondientes respuestas, en un proceso de resolución de nombres.
- Integridad, mediante la encriptación criptográfica de todas las respuestas DNS.
- Disponibilidad, mediante una rápida detección y posterior eliminación de paquetes DNS falsos.

Tomando como base estas dos alternativas, desde el Centro de Investigación y Desarrollo en Informática Aplicada (C.I.D.I.A.) y en el marco del proyecto de investigación Nº 1223/0: "Extensiones de Seguridad para el Sistema de Nombres de Dominio" del Consejo de Investigación de la Universidad Nacional de Salta, se desarrollaron tareas de estudio del estado del arte tanto de DNSSEC como de DNSCurve, implementación de ambientes de pruebas que permitan realizar estimaciones de consumos de recursos tales como ancho de banda, uso de memoria y procesador y aspectos relacionados a la integración de ambas, con la finalidad de obtener una arquitectura DNS segura.
El presente trabajo está centrado en mostrar los aspectos relacionados a la implementación de DNSCurve, y evaluación de comportamiento mediante la utilización de un prototipo de software desarrollado para la simulación de tráfico, lo que permitió obtener datos estadísticos basados en consultas DNS, análisis y conclusiones sobre el rendimiento obtenido en un servidor con estas características.

2. Aspectos generales del protocolo DNSCurve

En un servidor DNS Resolver con característica DNSCurve, todo el tráfico saliente, así como el tráfico entrante, es automáticamente encriptado haciendo uso de criptografía de curva elíptica (Curve 25519). Cuando el Resolver, contacta a un Servidor Autoritativo, comprueba si este contiene una clave pública DNSCurve con el formato correcto. Si la clave secreta por parte del Resolver es c, entonces la correspondiente clave pública es Curve25519(c), del mismo modo ocurre para el servidor autoritativo, considerando que su clave privada es s, la correspondiente clave pública es Curve25519(s). Las claves se componen de la cadena de caracteres "uz5", seguido de una cadena de 51 bytes codificados con la clave pública del servidor. Las claves públicas para un servidor autoritativo, se transportan sobre registros NS, de esta manera, un Resolver sabe que el servidor consultado soporta DNSCurve. De esta manera, el Resolver, puede de manera simple, encriptar y autenticar los paquetes que este envía al Autoritativo. Por su parte, el Resolver envía al servidor consultado, un paquete expandido que contiene su clave pública DNSCurve, un número (nonce), de 96 bits de longitud y una "cryptobox" que contiene el mensaje original, el cual es básicamente la consulta realizada. Para la creación de la "cryptobox", el Resolver usa su clave privada, el nonce del paquete y su clave pública. El servidor que recepciona la consulta, abre esta caja, utilizando su clave privada, más la clave pública del Resolver y el nonce del paquete. Como respuesta, el servidor consultado, envía un paquete expandido que contiene un nonce (de 96 bits) diferente al recibido, más su "cryptobox". De igual manera que lo hizo el Servidor Autoritativo, el Resolver, utiliza su clave secreta DNSCurve, la clave pública del Autoritativo y el nonce, para desencriptar la "caja" recibida. Si la apertura de esta caja fallara, podría significar que el mensaje ha sido falsificado, por lo tanto,

el Resolver, simplemente descarta el paquete y queda a la espera de una respuesta legítima.

2.1. Otros aspectos del protocolo

Las claves utilizadas por el Resolver, son generadas al momento que se inicia el programa DNSCurve, las cuales son usadas para la comunicación con varios Autoritativos. De manera similar, los servidores consultados, luego de que estos calculan una firma once, esta puede ser reusada para varios clientes. Del mismo modo, después de calcular el secreto compartido con un Resolver en particular, se lo puede usar para todos los paquetes que se fueran a intercambiar. Sin embargo, lo que se recomienda es regenerar las claves con cierta frecuencia, para evitar el rastreo de las mismas. Una particularidad de las claves, es que éstas, estan asociadas a los servidores y no a una zona.

3. Resultados

3.1. Implementación de DNSCurve

Durante las etapas de investigación se montaron diferentes escenarios de pruebas, con errores de funcionamiento en los primeros casos, debido a la escasa documentación de instalación y configuración encontrada al respecto. Finalmente se seleccionan dos escenarios, siendo el primero el ambiente donde se realizó el seguimiento de un proceso de consulta/respuesta y análisis de paquetes de datos. El segundo escenario permitió realizar un análisis de rendimiento (consultas procesadas y longitud de paquetes DNSCurve) comparando un servidor DNS estándar frente a un servidor DNSCurve.

A continuación se describen ambos escenarios junto a los resultados obtenidos.

3.2. Escenario de prueba para análisis de traza DNSCurve.

- Plataforma de hardware Servidor:
  - Memoria: 384 MB.
  - Procesador: Pentium (R) Dual Core CPU T4400 @2.20GHz.
  - Espacio en disco: 5,5 GB.
- Plataforma de software Servidor:
  - Linux Ubuntu Versión 10.10 (Maverick)
  - Núcleo Linux 2.6.31-14-generic
  - GNOME 2.32.0
  - Software DNS djbdns.
  - DNSCurve desde repositorios Hauke Lampe.

Con el escenario de prueba montado se procedió a realizar el seguimiento de una consulta enviada desde el cliente DNS hacia el Servidor DNSCurve, para el dominio dnscurve.org. Para la captura y análisis de tráfico se utilizó la herramienta Wireshark [6].

1. Stub Resolver cliente DNS envía consulta DNS estándar a servidor DNSCurve
2. Servidor DNSCurve selecciona servidor raíz y envía la misma consulta tipo A
3. El servidor raíz consultado, devuelve servidores autoritativos para la zona .org y registros adicionales.
4. El servidor DNSCurve elige al azar uno de los servidores con autoridad para la zona org y repite la consulta.
5. El servidor consultado, responde con información de servidores con autoridad para la zona dnscurve.org
6. A continuación el servidor DNSCurve envía la consulta a uno de los servidores elegido al azar.
7. El servidor Autoritativo con características DNSCurve, devuelve su clave pública contenida en un registro NS.
8. En esta instancia el Servidor Recursivo DNSCurve configurado, calcula la clave compartida usando la clave pública del servidor incluida en el NS y su clave privada. Se envía un paquete de consulta DNSCurve donde se incluye la clave pública, un nonce de 12 bytes del servidor recursivo, y la cryptobox. Dentro de la cryptobox se encuentra la misma consulta tipo A para el dominio dnscurve.org.
9. El servidor autoritativo DNSCurve recibe el paquete, extrae la clave pública del cliente (Servidor Recursivo DNSCurve), y haciendo uso de su clave privada calcula la clave compartida. Utilizando el nonce del cliente y la clave compartida desencripta exitosamente el paquete, y luego extrae la consulta DNS estándar. Prepara la respuesta y la encripta utilizando la clave compartida y un nuevo nonce. En el paquete incluye el nonce del cliente, el nonce calculado por el servidor y la respuesta cifrada, la cual finalmente es enviada al cliente.

Cabe resaltar que al momento de realizar las capturas de tráfico intercambiado y en particular para los puntos 8) y 9), al viajar la consulta y su correspondiente respuesta encriptadas, Wireshark no puede acceder a los contenidos, mostrando al paquete DNS como malformado.

3.3. Escenario de prueba para análisis de rendimiento servidor DNSCurve.

- Plataforma de hardware Servidor:
  - Procesador: Intel Core i3-3110M CPU 2,40 GHz.
  - Memoria RAM: 8 GB.
  - Dentro de este servidor se alojaron dos sistemas virtualizados a través de VirtualBox. Cada sistema, se configuró con 1 GB de memoria RAM.
- Plataforma de software Servidor: El servidor se configuró con sistema operativo base, Ubuntu 12.04 LTS 64 bits. Los sistemas virtualizados, se instalaron con sistema operativo Ubuntu Server 12.04 i386. El primer sistema virtualizado se configuró con un servidor TinyDNS. El segundo sistema, contaba con CurveDNS 0.87.

3.3.1. Parámetros de configuración.

Para las consultas a generar se tomaron como parámetros de configuración los siguientes datos, donde el porcentaje de errores de nombres, se utilizó para simular aquellos casos en que se reciben consultas cuyo dominio es inexistente.

Tabla 1. Matriz de configuración del tráfico a generar.

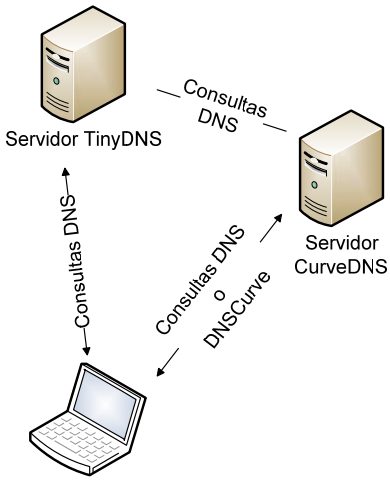
Tipo de Registro de Recurso	% del Total de Consultas	% error de nombre
1	0,70	0,05
2	0,05	0,10
5	0,08	0,10
6	0,03	0,12
15	0,10	0,06
16	0,04	0,07

Se establecieron tres modelos de tráfico para realizar los estudios comparativos:

- Consultas DNS a un servidor DNS estándar.
- Consultas DNS a un servidor que resuelve DNSCurve.
- Consultas DNSCurve a un servidor que resuelve DNSCurve.

A continuación, se muestran las trazas de los modelos de tráfico:

Fig. 1. Trazas de los modelos de tráfico



3.3.2. Análisis de rendimiento.

El análisis de rendimiento se realizó en base a las consultas procesadas, para cada modelo, se crearon cinco configuraciones diferentes de prueba. Cada configuración se distingue de la otra por la cantidad de consultas por segundo. La cantidad de consultas por segundo que se usaron para este estudio comparativo son 250, 500, 750, 1000 y 1250.

Se ejecutaron 5 pruebas para cada configuración de cada modelo de tráfico, obteniéndose los siguientes resultados:

Tabla 2. Cantidad total de consultas para cada modelo de tráfico.

qps	Cantidad de Consultas enviadas	Cantidad de consultas sin errores de nombre
250	15000	14094
500	30000	28188
750	45000	42282
1000	60000	56376
1250	75000	70470

qps	Cantidad de consultas al servidor DNS estándar procesadas	Cantidad de consultas al servidor CurveDNS procesadas
250	13949	13283
500	27262	17733

750	40561	22930
1000	54370	27415
1250	67073	30587

qps	Cantidad de consultas DNSCurve al servidor CurveDNS procesadas
250	13523
500	18281
750	22062
1000	25388
1250	29629

La primera observación que puede realizarse en base a los resultados obtenidos, es que el servidor DNS estándar alcanza a procesar el 95% de las consultas recibidas, mientras que a un servidor CurveDNS alcanza un 39%, con una baja en el rendimiento superior al 50% a medida aumenta la cantidad de consultas por segundo. Al analizar las causas de la caída en el rendimiento, se pudo observar que al recibir las consultas y no disponer de sockets para el reenvío de las mismas, tales consultas se descartan.

Al aumentar la cantidad de consultas enviadas por segundo (4000qps), se puede presentar una generalización del comportamiento del servidor CurveDNS, de manera tal que a medida que las consultas recibidas se incrementan de manera logarítmica, la cantidad de consultas perdidas se incrementan de manera exponencial.

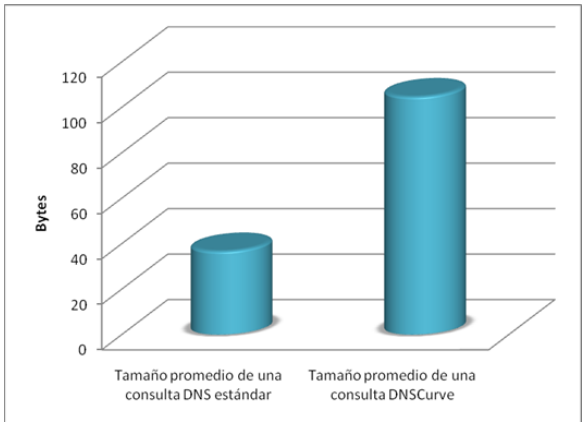
3.4. Comparativa longitud de consulta DNS estándar frente a consulta DNSCurve.

Con el propósito de realizar estimaciones de consumo de ancho de banda, se analizaron longitud de un paquete DNS estándar frente a un paquete DNSCurve. Para tales pruebas se hizo uso de un generador de consultas desarrollado en lenguaje Python, y un conjunto de librerías [4], que permitieron la creación de las claves públicas y privadas, así como la cryptobox Para el desarrollo del generador, se tomó como base el prototipo creado por Harm H. A. Van Tilborg [5].

El desarrollo se divide en dos partes: “dnsQueryGenerator” y “dnsQuerySender”. La primera, genera un archivo que contiene las consultas; para ello se necesita un archivo de configuración, que contiene los parámetros necesarios para generar las consultas, esto es, minutos de ejecución, cantidad de consultas por segundo, porcentaje de consultas DNSCurve, porcentaje de consultas erróneas para un determinado tipo de registro de recurso, entre otros; y otro archivo que contenga un listado de dominios. La segunda parte recibe como entrada este archivo de consultas generado, lee las mismas y las envía.

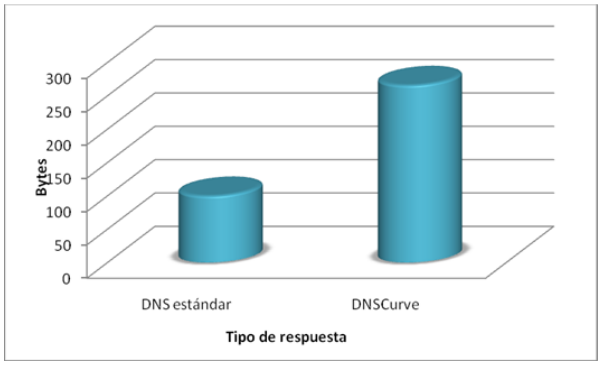
Mediante la ejecución de un script que realiza la lectura de un determinado archivo de consultas generado por el “dnsQueryGenerator”, se calcularon las longitudes de cada consulta presentes en este archivo, posteriormente se calculó el promedio del tamaño de las consultas, obteniendo los siguientes valores, luego de generar 75000 consultas.

Fig. 2. Longitudes promedio de consultas DNS y DNSCurve



Utilizando el mismo script, se pudo obtener valores para las longitudes de las respuestas enviadas por los servidores, obteniendo los siguientes datos.

Fig. 3. Longitudes promedio de respuestas DNS y DNSCurve



4. Discusión

En términos generales, la alternativa DNSCurve es producto de la investigación y desarrollo personal, a diferencia de la alternativa DNSSEC, que actualmente se encuentra impulsada a escala global por organizaciones como la ICANN, Verisign, IANA, entre otros. A la fecha, han disminuido las fuentes que consideran DNSCurve, así como las arquitecturas que le dan soporte, por lo que encontramos muchas dificultades para considerarlo un protocolo a implementar a gran escala. Sin embargo si se sugiere su implementación como parte de un conjunto de soluciones, con el propósito de abarcar todos los aspectos de seguridad en el Sistema de Nombres de Dominio.

5. Conclusión.

En particular y en base a los resultados obtenidos, del estudio realizado sobre el comportamiento para responder a las consultas enviadas, se pudo observar que el rendimiento es muy bajo, cayendo por debajo del 50% en relación al obtenido de un servidor DNS estándar, por la gran cantidad de consultas que quedan sin respuesta, producto de los costos de procesamiento criptográfico. En general, a medida que las consultas recibidas se incrementan de manera logarítmica, la cantidad de consultas perdidas se incrementan de manera exponencial.

Con respecto a la longitud del paquete DNSCurve generado y teniendo como punto de referencia el tamaño de un paquete DNS estándar, que en proporción es aproximadamente un tercio en relación a un paquete DNSCurve, se introduce un retardo en la transmisión del paquete; que no afecta de manera significativa el rendimiento, por lo que se podría no tener en cuenta.

En términos generales, DNSCurve aún no ha sido adoptada por la comunidad de Internet como una alternativa a implementar a escala global, a diferencia de la alternativa DNSSEC, que desde el año 2010 ya la implementan los servidores raíces [7], y son cada vez más los dominios de nivel superior que la han incorporado [8]. Sin embargo, consideramos que DNSCurve debe ser incorporado como parte de un conjunto de soluciones, con el propósito de abarcar todos los aspectos de seguridad en el Sistema de Nombres de Dominio.

6. Referencias

[1] ARENDS, R., AUSTEIN, R., LARSON, M., MASSEY, D., AND ROSE, S. *RFC 4033: DNS Security Introduction and Requirements*, Marzo 2005.

[2] DNSCurve: Usable security for DNS. http://dnscurve.org. Fecha de consulta: Setiembre 2013

[3] Sabbir Ahmmed. *DNS Privacy & Confidential DNS*. Seminar Report: Network & Communications Privacy. Alemania. Mayo 2014.

[4] NaCl: Networking and Cryptography library. Computer Aided Cryptography Engineering. http://nacl.cace-project.eu/. Fecha de consulta: Diciembre 2013

[5] Harm H. A. Van Tilborg. *Shaping DNS security with curves*. Technische Universiteit Eindhoven, Países Bajos. 2010.

[6] WIRESHARK. GNU GENERAL PUBLIC LICENSE Version 2, June 1991 .Copyright (C) 1989, 1991 Free Software Foundation, Inc. http://www.wireshark.org/download.html.

[7] Root DNSSEC. Information about DNSSEC for the Root Zone. http://www.root-dnssec.org/. Fecha de consulta: Julio 2014.

[8] Internet Society. DNSSEC Statistics. http://www.internetsociety.org/deploy360/dnssec/statistics/. Fecha de consulta: Julio 2014.

Implementación y Operación de un Cluster HPC Utilizando Laboratorios de Computadoras en Horarios de Inactividad

Ernesto J. Sale Sebastián Rodríguez
Grupo de Investigación en Tecnologías Informáticas Avanzadas
Facultad Regional Tucumán
Universidad Tecnológica Nacional
Rivadavia 1050 - C.P. 4000 - S.M. de Tucumán
email: {ernesto.sale,sebastian.rodriguez}@gitia.org

Abstract

Este trabajo presenta un reporte preliminar de un método que logra facilitar a alumnos, docentes e investigadores el acceso a clusters HPC a costos accesibles, utilizando recursos existentes en las instituciones donde desempeñan sus actividades. Los laboratorios de informática de las universidades constituyen un reservorio de potencia de procesamiento. Estos pueden ser utilizados para conformar un cluster de cálculo de alto desempeño, en los horarios que no se encuentran ocupados. De esta manera se puede al mismo tiempo reducir los costos de poseer un cluster HPC, ampliando el abanico de servicios y posibilidades que la institución brinda a sus miembros y aumentar la utilización de los laboratorios, mejorando su amortización y por ende justificando mejor la inversión.

Palabras Clave cluster HPC, cluster de alto rendimiento, CAR, procesamiento distribuido, utilización recursos existentes entorno educativo,cluster by night, cluster de noche, cluster en laboratorio.

1. Introducción

A medida que la investigación avanza en diversos campos, encarando tareas más complejas, trae consigo mayores requisitos de procesamiento de datos. Por otro lado, el avance en las herramientas informáticas y la recolección de datos hacen posible enfocar los problemas desde nuevos ángulos, siempre y cuando se posea la capacidad de procesamiento adecuada. Esta es la realidad de aplicaciones tales como predicción de clima, predicción de consumo eléctrico de poblaciones, simulaciones de plegamiento de proteínas, diseño de automóviles, predicción financiera, simulación de terremotos, exploración petrolera y data mining, entre muchas otras, las cuales manejan cantidades de datos y/o algoritmos que requieren recursos de computación a gran escala. Entre los trabajos que apoyan estas afirmaciones podemos enunciar [22, 4, 3].

Esto presenta una oportunidad para que centros de investigación e instituciones educativas universitarias se desarrollen y mantengan actualizados en la provisión de altas capacidades de cálculo, y a la vez la amenaza de quedar relegados como meros espectadores en el avance de los conocimientos en diversas áreas como las mencionadas previamente.

Originalmente esta capacidad era provista por equipos y sistemas que resultaban costosos, a medida del cliente, fabricados por empresas como Cray, Convex, Tera Computer, SGI, IBM y HP.

En 1994, desde el Goddard Space Flight Center de la NASA, hicieron su aparición los clusters del proyecto Beowulf [26, 29], que integran PCs, Linux y una red de comunicación altamente difundida (Ethernet) para formar una computadora paralela, y que de este modo bajaron dramáticamente la barrera para la posesión de una supercomputadora [21].

No obstante, hay numerosas instituciones que siguen impedidas de superar esta barrera por no contar con los recursos necesarios (dinero, equipos, PCs, personal capacitado) para construir y operar un High Performance Computing cluster (en adelante llamado cluster HPC) dedicado.

Sin embargo, muchas mantienen vastos laboratorios de computación, donde están congregados todos los ingredientes físicos necesarios para la operación de un cluster HPC tipo Beowulf, que podrían ser utilizados a tales fines en los horarios que no están siendo ocupados para sus funciones tradicionales, por ejemplo, durante la noche o los días feriados.

Si bien la capacidad de un sistema como el sugerido sería inferior a la de un sistema con recursos dedicados, es justo lo indicado para numerosas instituciones que, por razo-

nes de volumen de trabajo y/o de presupuesto, no pueden justificar el costo de tener un cluster HPC dedicado, y como consecuencia sus alumnos, docentes e investigadores no pueden desarrollar tareas de I+D+i que requieran alta capacidad de procesamiento. Por otro lado, quienes realmente necesiten hardware dedicado, pueden utilizar este sistema como una herramienta de medición que les permita justificar objetivamente dicha necesidad. Adicionalmente serviría para capacitar los recursos humanos necesarios para administrar y mantener el cluster.

Para lograr la implementación de un sistema como el mencionado, es preciso resolver algunos problemas. El acceso físico a las máquinas es uno de ellos, dado que típicamente los laboratorios e incluso los edificios que los contienen se encuentran cerrados por las noches y los días Domingo y feriados todo el día. Otro problema es la configuración de los equipos, que debe ser ajustada con cuidado para no afectar el normal funcionamiento del laboratorio para su función principal. A esto se puede agregar restricciones derivadas de políticas de los administradores de los laboratorios (por ejemplo, no modificar el contenido del disco rígido) y de los administradores de la red (por ejemplo, impidiendo acceso desde Internet), entre otros.

Este reporte presenta un enfoque para transformar un laboratorio de computadoras en un cluster HPC durante sus horarios de inactividad, y los resultados parciales de su implementación en uno de los laboratorios de la Facultad Regional Tucumán de la Universidad Tecnológica Nacional.

El objetivo de este trabajo es proponer una solución que:

1. Permita el empleo de equipamiento e infraestructura existentes.
2. Sea escalable.
3. Sea reutilizable (en otras instituciones educativas, por ejemplo).
4. No requiera conocimientos avanzados de sistemas operativos distribuidos para su configuración, administración y operación.
5. Posea una interfaz para el usuario sencilla de utilizar.
6. Sea útil para la investigación y la docencia.
7. Utilice software gratuito y de código abierto, posible de mejorar.
8. Posea robustez y presente un desempeño estable y predecible.
9. Sea lo más independiente posible de la administración del laboratorio y lo menos invasiva posible de su configuración preestablecida.

La estructura de este documento presenta una breve reseña sobre clusters HPC y los trabajos relacionados en la sección 2; una descripción estructural y funcional de la solución propuesta en la sección 3; la plataforma de evaluación utilizada para prueba del concepto es descrita en la sección 4; los resultados obtenidos en la sección 5; y cerrando el presente reporte, las conclusiones y líneas de trabajo futuro en la sección 6.

2. Trabajos Relacionados

En este trabajo sólo serán evaluadas las soluciones gratuitas y de código abierto, buscando respectivamente el menor costo posible y brindar un aspecto más de utilidad a la docencia e investigación habilitando la posibilidad de que estos sistemas sean evaluados y mejorados.

2.1. Reseña de Clusters HPC

Los *clusters* son grupos de computadoras (llamadas nodos) interconectadas y configuradas con un propósito específico, que típicamente es alto desempeño (también conocidos como HPC, para aplicaciones de cálculo intensivo) o bien alta disponibilidad (o HA, para gran tolerancia a fallas). Los clusters HPC pretenden acelerar un cálculo complejo realizando una partición de este, de modo que cada nodo pueda ejecutar una parte. Así un cluster de 10 nodos podría idealmente resolver un sistema de ecuaciones en un décimo del tiempo que tardaría un único nodo. En la realidad, hay varias restricciones que impiden alcanzar ese ideal. Una de ellas es que no todos los cálculos pueden ser resueltos en paralelo. Cada problema a resolver, cada cálculo a realizar, cada programa a ejecutar tiene distinto grado de paralelismo. Esto hace que cada caso tenga un distinto factor de aceleración al ser procesado en un cluster. Otra es la red de interconexión y sus características como ancho de banda y latencia. También están el tipo de procesador utilizado en los nodos, su relación con cantidad y ancho de banda de la RAM, el sistema de E/S y el *overhead* causado por el sistema operativo subyacente, entre otras. Por todo esto, para intentar comparar este tipo de sistemas se utilizan 2 métricas. Ambas son expresadas en operaciones de punto flotante (típicamente de precisión simple) por segundo (FLOPS). La primera es el desempeño máximo teórico (Rpeak), que se calcula como la suma de las capacidades máximas teóricas de cada procesador. La segunda es un valor arrojado por un *benchmark* que se convirtió en un estándar *de facto* para este tipo de medición, llamado Linpack.

De hecho hay un sitio, top500.org[18] que periódicamente actualiza una lista de los 500 sistemas de computadoras más poderosos del planeta, que utiliza el resultado de Linpack[6] como único parámetro de comparación. Los

clusters HPC son usados en diversas áreas de la investigación científica y comercial para realizar cálculos complejos tales como predicción de clima, simulaciones de plegamiento de proteínas, diseño de automóviles, predicción financiera, simulación de terremotos, exploración petrolera, data mining, etc.

Los clusters HPC de tipo Beowulf[16, 23], nombrados como el héroe de un poema que cuenta con la fortaleza de muchos guerreros y su misión es destruir a cierto monstruo[20, 10], se caracterizan por estar integrados por componentes de uso masivo (*commodity components*). Los componentes incluyen las computadoras, que son típicamente PCs pero pueden ser dispositivos de propósito más específico como celulares o consolas de video juegos[31]; el hardware de red, típicamente ethernet; el software de base, que tiene como principal participante a sistemas operativos GNU/Linux[25, 21, 29]; y *middleware* que se encarga de tareas administrativas. Es en la utilización de este tipo de hardware y software precisamente donde se inclina la balanza de la relación precio/performance hacia los clusters Beowulf[29] en comparación con las soluciones alternativas.

2.2. Soluciones para clustering en general

Para software de clustering se evaluarán únicamente aquellos sistemas que sean gratuitos, de código abierto y basados en Linux, por su amplio soporte de hardware, por la vasta disponibilidad de documentación referida al tema de clustering HPC, por permitir modificación, depuración y subproyectos, y por tener el menor impacto en el presupuesto de la institución. Se consideraron las implementaciones SSI, debido principalmente al interés en contar con balance de carga, migración y *checkpointing*[1] de procesos. Entre ellas se destacaron OpenMosix [8], OpenSSI [9] y Kerrighed [5].

Renaud Lottiaux y otros comparan en un trabajo [27] realizado en 2004 los 3 sistemas operativos mencionados. En la actualidad, sin embargo, el proyecto OpenMosix está cerrado desde Marzo de 2008, y su continuación, LinuxPMI, tiene problemas de estabilidad [15]. El proyecto OpenSSI en su versión de desarrollo más reciente al momento de la redacción (1.9.6, lanzada en Febrero de 2010) utiliza un kernel demasiado viejo, 2.6.12, presentando problemas de drivers para el hardware. Su versión estable (1.2.2) fue lanzada en Marzo de 2005. Para este trabajo, Kerrighed parece ser el mejor candidato porque su última versión al momento de esta redacción (3.0.0 lanzada en Junio de 2010), está basada en el kernel de Linux 2.6.30 y soporta arquitecturas de 64 bits y es por esto el que más probabilidades tiene de soportar el hardware de las máquinas de cualquier laboratorio empleado.

2.3. Enfoques para reutilizar recursos

Debido a que este trabajo propone la confección de un cluster HPC utilizando recursos existentes, y habiendo evaluado el software específico de clustering, cabe evaluar el estado del arte en reutilización de recursos de computación. Entre las técnicas existentes, destacaremos algunas a continuación.

El modelo de **computación voluntaria**, utilizado por ejemplo en el proyecto SETI@home [14], basado en BOINC [1], voluntarios anónimos instalan un cliente en su computadora o dispositivo móvil que se conecta mediante Internet a un servidor que maneja la paralelización de las tareas, donando así poder de procesamiento.

La ventaja principal es que se cuenta con un potencial muy grande de usuarios, por ende de capacidad de cálculo, a un costo relativamente bajo. Las principales desventajas son la dependencia de los usuarios, que deben ser convencidos para instalar el cliente, hay que confiar en su "buen comportamiento", esto es, que no interfieran con el proceso devolviendo resultados incorrectos (voluntaria o involuntariamente), el impacto en la performance o el consumo de energía de sus estaciones de trabajo asociado, el hecho de que se donan sólo los ciclos de CPU inactivos, que su equipo debe estar encendido y la dependencia de internet. Todo estos factores hacen además difícil de predecir el rendimiento del sistema. Como desventaja adicional hay que mencionar que no es apropiado para trabajos que presentan una baja relación procesamiento/comunicación [28].

El modelo de **Desktop Grid o Network of Workstations**[21] es similar a computación voluntaria, sólo que se utiliza en intranets, por ejemplo en una empresa o institución, haciendo que los usuarios no sean anónimos y posibilitando más control [28, 1] y mejor predicción de disponibilidad de recursos [30]. Su funcionamiento en intranets complejas es muy dependiente de la configuración y administración de la red.

En el **cluster de estaciones de trabajo virtualizadas**, un software de virtualización divide los recursos de hardware a bajo nivel, asignando una cierta cantidad de CPUs, memoria, interfaces de red y otros a una máquina virtual que será nodo de un cluster, por ejemplo la utilizada en Parker Aerospace [24]. Es más barato agregar hardware en las PCs existentes, y evita compartir recursos con el usuario de la PC, pero requiere BIOS[2] y hardware especial (chipset que

[1] La capacidad de tomar una imagen instantánea de un proceso, para luego reanudar su ejecución desde ese punto.

[2] Basic Input/Output System

soporte Intel(R) VT-d, por ejemplo) y posiblemente software propietario.

Las **imágenes LiveCD de clusters HPC** proveen en general un cluster basado en MPI[3]. En algunas como PelicanHPC [11] el "nodo maestro" arranca desde el CD, y el resto por red (PXE[4]), mientras que en otras como Cluster by Night [2, 13], todos los nodos deben arrancar desde el CD.

Los **sistemas de despliegue, provisionamiento y administración de clusters** se encargan de todos los pasos mencionados, aunque la automatización requiere trabajo adicional. Si bien tienen opción para usar sistema de archivos en red, no consideran otros puntos de conflicto con administradores de laboratorio, como los servidores DHCP[5].

Ejemplos de estos sistemas son Warewulf [19], oneSIS [7] y Perceus [12]. Este último requiere adquirir licencia para activar ciertas características.

Serán evaluados en el futuro para considerar una posible integración con el presente trabajo.

3. Arquitectura general del sistema propuesto

A los fines de permitir la reproducción de esta experiencia, facilitaremos en esta sección los elementos de hardware, software, topologías, configuraciones, criterios y demás elementos que se utilizaron en la elaboración de este trabajo y conforman el producto desarrollado.

3.1. Requisitos de Hardware y Software

Es preciso contar con una computadora que tendrá a su cargo la ejecución de la máquina virtual que constituye el **servidor** del sistema de cluster (a cargo de los servicios de DHCP, TFTP[6] y NFS[7] en la red del laboratorio). Aunque puede ejecutarse con menos recursos de procesador, RAM y disco, los requisitos mínimos recomendados para dicho *host* son un procesador AMD Sempron 3200+ o equivalente, 512 MB RAM, 30 GB HDD e interfaz de red Ethernet 100 Mbps. Como software de virtualización se recomienda VMware Server (gratuito) que ejecuta en sistemas operativos Linux, sin embargo se observó indistinto desempeño usando VMware Player (igualmente gratuito).

[3]Message Passing Interface
[4]Preboot eXecution Environment
[5]Dynamic Host Configuration Protocol
[6]Trivial File Transfer Protocol
[7]Network File System

Las computadoras del laboratorio, que serán los **nodos** del cluster, deben poseer hardware compatible con el kernel de Kerrighed 3.0 (Linux 2.6.30), o del sistema operativo de cluster que se desee ejecutar (ver sección 3.8), y deben ser capaces de iniciar desde la red (a través de PXE) y "despertar" desde la red por medio de Wake-On-LAN, preferentemente desde el estado S5.
Todas ellas deberán contar con al menos una interfaz de red Ethernet de por lo menos 100 Mbps de capacidad.

Relativo a la **conectividad de red**, es necesario que exista una infraestructura de conexión Switched Fast Ethernet como mínimo entre el servidor y todos los nodos.

3.2. Configuración del laboratorio

Todos los nodos deben tener su BIOS protegido con contraseña para dificultar modificaciones, Wake-On-LAN activado, y la placa de red como principal dispositivo de arranque. Al interrumpirse el suministro de energía eléctrica, la máquina debe recordar el estado o encenderse (ver sección 3.4).

En el ámbito del laboratorio, no debe haber otro servidor ofreciendo arranque por red (PXE). De otro modo, será necesario ajustar la configuración de los servidores, por ejemplo, apagando y encendiendo los servicios en los horarios necesarios.

Cada uno de los ajustes mencionados en esta sección debe ser coordinado con los administradores del laboratorio, y es precisamente su baja cantidad lo que determina el alto grado de independencia de la administración que esta solución presenta.

3.3. Servicios necesarios en la red local

Para iniciar el cluster, los nodos deben usar como dispositivo de arranque la placa de red. La imagen de arranque es provista por un servidor TFTP. Dicha imagen de arranque será un archivo que indica arrancar desde el disco rígido local en horarios de uso "normal" (llamemos así al horario en que el laboratorio es tradicionalmente utilizado para su propósito original); y en los horarios de inactividad, será la imagen de arranque del cluster.

Un servidor DHCP debe ser configurado para indicar la ubicación de la imagen de arranque a los nodos. Si es el único servidor DHCP sirviendo pedidos de arranque via PXE, no es necesario que sea autoritativo, y puede coexistir con otros servidores DHCP en la misma red sin interferir. Esto es debido a que el servidor DHCP del cluster sólo es necesario para el arranque por medio de PXE, dado que los nodos pueden recibir IPs estáticos a través de la línea de comando del kernel pasada por PXELINUX al

momento del arranque. Dado que las máquinas deben tener configurada la red como primera opción de dispositivo de arranque, es conveniente que estos dos servicios (TFTP y DHCP) estén permanentemente activos para evitar la espera del *timeout* del intento de arranque a través de PXE por parte del nodo.

El sistema de archivos del cluster también está en la red, en un servidor NFS, el cual, por razones de seguridad, conviene que no esté activo hasta el momento de operar el cluster. En el diseño de esta solución se decidió emplear arranque desde la red y sistema de archivos de red para no utilizar el disco rígido de las máquinas (nodos), y de este modo lograr que el cluster represente la menor invasión posible en el ambiente del laboratorio y la mayor independencia posible de sus administradores. Esta técnica además evita fallas por problemas con los discos y permite apagarlos por software para ahorrar energía. Como efectos adversos hay consideraciones de seguridad (NFS no es un servicio seguro por sí mismo), la red ya no es utilizada exclusivamente para conexión entre nodos sino también para las operaciones del sistema de archivos y por lo tanto sus desempeños (tanto el de la comunicación entre nodos como el del sistema de archivos) pueden verse afectados, en una medida dependiente del grado de utilización de cada uno de estos recursos por parte de la aplicación; por otro lado, la red incrementa la latencia y puede resultar en cuello de botella para las operaciones del sistema de archivos; finalmente, el servidor NFS también puede ser cuello de botella. No hubo posibilidad de evaluar hasta el momento la magnitud de este efecto comparando el sistema propuesto con uno que utilice el disco rígido local de cada nodo, pero las pruebas realizadas hasta el momento muestran un rendimiento aceptable, considerando las circunstancias y limitaciones de este enfoque. En caso de ser necesario un mayor rendimiento de sistema de archivos, por ejemplo, debería realizarse un acuerdo con el/los administradores del laboratorio para ocupar una porción del disco rígido local de cada computadora, o utilizar redes separadas para el sistema de archivos y la comunicación entre nodos.

Todos estos servicios están configurados y en ejecución en la máquina virtual mencionada en la sección 3.1, que puede ser vista como un "nodo maestro" (*head node*) del cluster, y será llamada así, aunque estrictamente no lo sea.

La estructura descrita hasta acá puede ser visualizada gráficamente en el esquema de la figura 1.

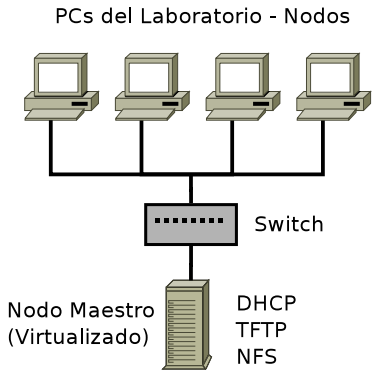


Figura 1. Estructura del cluster en un laboratorio de computadoras.

3.4. Operación

Al finalizar el horario de uso normal del laboratorio, las máquinas deben ser apagadas. Luego de cambiar la imagen de arranque del servidor TFTP a la del cluster, deben ser encendidas nuevamente. De esta forma, el sistema operativo del cluster cargará en todas ellas.

Todo este proceso puede estar automatizado por *scripts*, para evitar la necesidad de la intervención de un ser humano físicamente presente en el laboratorio.

El cluster debe ser configurado para apagarse momentos antes de finalizado el horario designado (idealmente realizando checkpoint de los procesos en ejecución), y la imagen de arranque del servidor TFTP debe ser cambiada a la de operación normal al mismo tiempo.

Los cortes de suministro de energía eléctrica pueden presentar un desafío particular: los nodos pueden no ser capaces de procesar la orden de arranque por red (Wake-On-LAN) en el estado en que se encuentran al momento de regresar la alimentación (S5). En este caso, habrá que considerar configurar el BIOS para que recuerde el estado de la máquina, y así encienda el equipo nuevamente cuando corresponda.

3.5. Acceso al cluster

Para acceso de administración y mantenimiento, puede ser necesario configurar rutas en la red para lograr el acceso desde fuera del laboratorio (ya sea dentro o fuera de la red local, dependiendo de las necesidades y posibilidades) a la máquina que ejecuta los servicios necesarios y al cluster.

Para el acceso de los usuarios del cluster, se propone, considerando seguridad y usabilidad, la implementación de

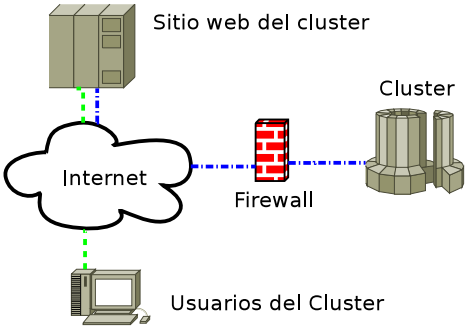


Figura 2. Acceso al cluster desde el exterior para usuarios, a través de Internet.

un sitio web con mecanismos de autenticación y autorización apropiados, que permita "subir" los trabajos a ejecutar en el cluster y "bajar" los resultados una vez disponibles.

Asimismo es recomendable que este sitio web se encuentre alojado en una máquina distinta al nodo maestro, y de ser posible, ni siquiera conectada directamente con la red del cluster. De este modo, este servidor bien podría estar en una DMZ o contratado en algún proveedor de *hosting*, y sería el intermediario entre los usuarios y el cluster, evitando toda interacción directa entre ellos (ver figura 2).

3.6. Presentación de trabajos

Los trabajos a ejecutar (programas y datos) deberán ser subidos en una forma normalizada y claramente documentada. Por ejemplo, empacada y comprimida en un único archivo de formato tar.bz2 o tar.gz, y con un *script* o archivo ejecutable principal con un nombre definido específicamente.

De igual modo debe establecerse el mecanismo para devolver los resultados. Una alternativa es establecer que el sistema comprima y suba al servidor web todo el contenido de un directorio de nombre convenido.
Una imagen de máquina virtual del sistema será provista para que los usuarios puedan realizar pruebas de ejecución y depuración antes de enviar el trabajo al cluster.

3.7. Ejecución de trabajos y entrega de resultados

Los trabajos serán planificados con un orden según criterios a determinar por los administradores, y llamados a ejecución por el host que provee los servicios internos del cluster (TFTP, DHCP, NFS, etc.). Este proceso involucra iniciar el cluster, planificar un trabajo, descargarlo desde el servidor externo, disponerlo (ubicarlo en el directorio correspondiente, descomprimirlo, etc.) y dar la orden para su ejecución llamando al *script* principal provisto por el usuario.

Al finalizar el trabajo, el servidor interno comprimirá el directorio de resultados, para luego subirlo al servidor web, relacionándolo con la cuenta de usuario correspondiente. Un email será enviado al propietario del trabajo para informar del evento.

Como último paso, por seguridad, puede disponerse el reinicio del cluster desde una imagen de respaldo de su sistema de archivos, para evitar que la ejecución de un trabajo de usuario afecte la de los siguientes.

3.8. Cualidades destacadas

A continuación se enuncian las ventajas que ofrece el enfoque elegido en este trabajo.

Flexibilidad

El método propuesto es flexible en el sentido en que permite la ejecución de diversos softwares de base y de clustering. En este caso fue evaluado conveniente utilizar Kerrighed 3.0, pero puede también servir para versiones anteriores de Kerrighed, y podría servir para cualquier otro sistema de clustering existente o futuro siempre que sea compatible con el hardware de las máquinas.

Incluso las restricciones de sistema de archivos y arranque en red podrían ser superadas con el apoyo de los administradores de los laboratorios. Esto podría habilitar para ejecutar sistemas que deben estar instalados localmente y no soportan arranque por red, usando (paradójicamente) una imagen de arranque por red que indique la partición del disco a iniciar.

Desde otro aspecto de la flexibilidad, se destaca que los usuarios del cluster pueden disponer *scripts* para instalar el software que sea necesario para ejecutar sus programas, subir el software en formato binario al servidor web, o incluso puede evaluarse la implementación de un mecanismo para que usuarios avanzados puedan subir el sistema de archivos completo a ser usado por el cluster.

Escalabilidad

El sistema es adaptable a las capacidades de los laboratorios, ya sea que se agreguen o eliminen nodos o bien que se cambie su configuración interna (siempre y cuando el hardware sea soportado por el kernel).

Portabilidad

Tener el nodo maestro o servidor del cluster en una máquina virtual "empaqueta" la solución, facilita su distribución y trae consigo la ventaja de poder ser ejecutado en una gran variedad de entornos.

Sencillez

Bajo ciertas condiciones, poner en marcha un sistema como el propuesto puede ser tan simple como encender una máquina virtual. Otras tareas administrativas tales como respaldo y restauración también se benefician de la virtualización.

Para el usuario, utilizar el cluster es "subir" su trabajo a un sitio web, y esperar el aviso de resultados via email, junto con instrucciones para descargarlos. Se dispone además un mecanismo muy cómodo y conveniente para pruebas y depuración (ver sección 3.6), donde nuevamente la virtualización hace su aporte.

Seguridad

Si bien se hicieron consideraciones de seguridad en el diseño de la solución, por ejemplo en las secciones 3.5 y 3.7, un análisis en detalle queda como trabajo futuro, para diseñar e implementar mecanismos que eviten usos inadecuados tales como que el cluster provea acceso desde el exterior por medio de túneles o que sea utilizado para enviar spam, entre otros.

4. Plataforma de evaluación

A modo de ejemplo y de guía, se presenta a continuación el listado de las características más relevantes del cluster HPC formado con las máquinas de uno de los laboratorios de la UTN-FRT[8].
Sin embargo, debe quedar claro que el objetivo del trabajo no es detallar instrucciones sobre la configuración ni operación de un cluster HPC, sino de proponer un método razonable y conveniente para convertir un laboratorio de computadoras en un cluster HPC durante las horas libres de uso.

El cluster posee una arquitectura híbrida (cluster de SMPs[9]) de nodos homogéneos, y pudo ser iniciado con un máximo de 20 nodos, debido a que el laboratorio donde las pruebas para este trabajo fueron desarrolladas posee esa cantidad de computadoras.
Cada nodo está equipado con un procesador Intel Core 2 Duo E7400 con 2 núcleos de 2,8 GHz (arq. 64 bits) y 2 GB de RAM en configuración dual-channel. El chipset del motherboard es Intel G31. La interconexión es en topología estrella a través de una red Switched Fast Ethernet provista por un switch 3Com de 24 bocas de la línea Baseline Switch 2024 (3C16471B). La placa madre posee una interfaz de red PCI Express Gigabit Ethernet RTL8111/8168B y BIOS que permite arranque desde la red (via PXE) y Wake-On-LAN.
El servidor se encuentra en una máquina virtual ejecutada por VMware Server 2.0.1 con recursos de 256 MB de RAM, 10 GB de disco rígido y una interfaz Fast Ethernet. La máquina física es una estación de trabajo Sun Microsystems Ultra 20, que cuenta con un procesador Dual-Core AMD Opteron(tm) Processor 1210 de 1,8 GHz, con 2 GB de RAM, 600 GB de disco rígido y dos interfaces de red Gigabit Ethernet (aunque sólo una está conectada).

5. Resultados

Se desarrolló una imagen de máquina virtual que permite iniciar el cluster presentado como ejemplo, en un laboratorio de computadoras que cumpla con los requisitos especificados, sin interferir con las funciones normales del laboratorio en horarios tradicionales.

El arranque y apagado del cluster pudo ser automatizado exitosamente, usando los enfoques descritos en la sección 3.

El sistema operativo elegido, Kerrighed, fue capaz de operar con el hardware del laboratorio y el sistema de archivos en red, presentando un desempeño aceptable, indistinguible desde el punto de vista de experiencia de usuario de un Debian 6.0 (también conocido como *Squeeze*, que es la versión actualmente estable) sin terminal gráfica instalado en disco local. Su estabilidad se vio comprometida al intentar utilizar memoria remota, y con la desconexión de uno o más nodos del cluster. La migración de procesos funciona correctamente, y el *process checkpointing* no pudo ser evaluado en profundidad hasta el momento.

Para hacer pruebas de rendimiento del cluster sobre MPI (OpenMPI versión 1.5), se utilizó LINPACK[6], el *benchmark* con el cual se miden los desempeños de las supercomputadoras para clasificar en top500.org[17]. Los parámetros de esta prueba todavía no han sido ajustados para obtener máximo rendimiento. El mejor valor obtenido hasta el momento es de 21,29 Gigaflops[10] (de aquí en más Gflops) en un cluster de 8 PCs (16 núcleos en total), que tomó 2003,7 segundos en ejecutar, y que en relación con el resultado obtenido para una prueba similar en sólo 1 nodo de 2 núcleos (5,09 Gflops) nos da un rendimiento del 52,3 %.

El grado de independencia de la administración del laboratorio fue el esperado. El funcionamiento del cluster no fue afectado por rupturas de discos rígidos, ni instalaciones de diversos sistemas operativos en ellos.

Un esquema de pruebas más intensivo no ha sido posible por no contar hasta el momento con la autorización pertinente para reconfigurar el BIOS de las máquinas del laboratorio según lo pautado en la sección 3.2.

[8] Universidad Tecnológica Nacional - Facultad Regional Tucumán
[9] Symmetric MultiProcessing
[10] flops: floating point operations per second

6. Conclusiones

Las pruebas preliminares realizadas hasta el momento demuestran que es posible la reutilización de recursos de un laboratorio de computadoras para la implementación de un cluster HPC. El rendimiento cercano al 50 % obtenido en el LINPACK no debe interpretarse desde un punto de vista cuantitativo, debido a que es susceptible de ser mejorado, y por otro lado, es esencialmente el resultado de un *benchmark* y el desempeño del cluster será distinto según la aplicación a ejecutar en cada caso. Desde el punto de vista cualitativo, en cambio, nos dice que el arreglo de computadoras está en condiciones de resolver problemas usando MPI, lo que se traduce en utilidad para la investigación y la docencia, para hacer algunos cálculos de baja complejidad pero que excedan las capacidades de los equipos singulares disponibles, para hacer pruebas preliminares, para ensayar soluciones novedosas, para justificar y dimensionar solicitudes de mayor poder de procesamiento (clusters dedicados), para capacitar personal de administración y mantenimiento, para desarrollar software y soluciones para HPC, entre otros usos posibles.

Es importante notar que la infraestructura de hardware empleada en este desarrollo es básica y elemental a cualquier laboratorio de computadoras de la actualidad, y que sus componentes de software son "libres". Esto significa que la inversión ya se encuentra realizada donde exista un laboratorio, por lo cual podríamos decir que el rendimiento costo-beneficio es óptimo, incluso cuando las prestaciones del sistema no lo sean. Hay que tener en cuenta que el eje del presente trabajo no es la optimización de rendimientos, sino la reutilización de recursos existentes.

La continuación de este trabajo de aquí en adelante se enfocará en evaluar el aporte a los objetivos de este trabajo que pueden realizar los sistemas mencionados en la sección 2.3 y buscar otros sistemas de administración para clusters, planificadores de procesos y gestores de colas que sean factibles de ser integrados en el esquema de funcionamiento propuesto. Habrá que considerar el desarrollo propio de aplicaciones de ser necesario.

Por otro lado, se buscará un modo conveniente de integrar diversos sistemas como el propuesto (ejecutando en distintos laboratorios) junto con esquemas de Desktop Grid y computación voluntaria (también mencionados en la sección 2.3) para configurar con ellos un grid de alcance inicialmente local a la facultad.

Finalmente, habrá que estudiar en mayor profundidad los aspectos de seguridad del cluster, teniendo en cuenta que puede ser tanto objetivo como herramienta de ataques, dado que es, a fin de cuentas, un sistema destinado a ejecutar código ajeno.

Por tratarse el presente de un trabajo en progreso, aún restan desafíos y problemas por resolver; no obstante los avances parciales aquí expuestos significan un paso en la dirección correcta hacia el objetivo principal, que es reducir el impacto de la escasez de recursos en las instituciones educativas por medio de su mejor utilización, para ayudar a investigadores, docentes y alumnos a permanecer en la carrera del conocimiento.

7. Agradecimientos.

Departamento Sistemas - Universidad Tecnológica Nacional - Facultad Regional Tucumán: P. Nazar, A. J. Nasrallah; TICs: J. Arias, G. Correa; Cátedra de Sistemas Operativos - UTN - FRT: L. R. de la Zerda, E. Loandos; Otros: F. Villacis Postigo.

Referencias

[1] Berkeley Open Infrastructure for Network Computing (BOINC). http://boinc.berkeley.edu. [Online; Agosto de 2014].

[2] Cluster by Night. http://www.dirigibleflightcraft.com/CbN/. [Online; Abril de 2012].

[3] High End Computing Terascale Resources (HECToR) Scientific Case. http://www.epsrc.ac.uk/SiteCollectionDocuments/other/HECToRScientificCase.pdf. [26 de Abril de 2004].

[4] International Review of Research Using HPC in the UK. http://www.ucl.ac.uk/research-computing/news/epsrc-uk-hpc-review-dec05.pdf. [Diciembre de 2005, Engineering and Physical Sciences Research Council (Polaris House, Northstar Avenue, Swindon, SN2 1ET, Wiltshire, UK (ISBN 1-904425-54-2)].

[5] Kerrighed. http://www.kerrighed.org/. [Online; Agosto de 2014].

[6] LINPACK. http://www.netlib.org/linpack/. [Online; Agosto de 2014].

[7] oneSIS. http://onesis.org. [Online; Agosto de 2014].

[8] OpenMosix. http://www.openmosix.org/. [Online; Agosto de 2014].

[9] OpenSSI. http://openssi.org/. [Online; Agosto de 2014].

[10] Our Beowulf Cluster. http://maven.smith.edu/~vmorales/parallel.html. [Online; Abril de 2012].

[11] Pelican HPC. http://pareto.uab.es/mcreel/PelicanHPC/. [Online; Agosto de 2014].

[12] Perceus. http://www.perceus.org. [Online; Agosto de 2014].

[13] Ron Zacharski Cluster by Night. http://zacharski.org/cluster-by-night/. [Online; Agosto de 2014].

[14] SETI@home. http://setiathome.berkeley.edu. [Online; Agosto de 2014].

[15] Status-Linux PMI. http://linuxpmi.org/trac/wiki/Status/. [Online; Agosto de 2014].

[16] The Beowulf Howto. http://tldp.org/HOWTO/html_single/Beowulf-HOWTO. [Online; Agosto de 2014].

[17] The Linpack Benchmark — TOP500 Supercomputing Sites. http://www.top500.org/project/linpack/. [Online; Agosto de 2014].

[18] Top 500. http://www.top500.org/. [Online; Agosto de 2014].

[19] Warewulf. http://warewulf.lbl.gov/trac. [Online; Agosto de 2014].

[20] WordSpy: Beowulf cluster. http://wordspy.com/words/Beowulfcluster.asp. [Online; Agosto de 2014].

[21] J. Adams and D. Vos. Small-college supercomputing: building a beowulf cluster at a comprehensive college. In *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, SIGCSE '02, pages 411–415, New York, NY, USA, 2002. ACM.

[22] R. Bose, A. Crosswell, V. Hamilton, and N. Mesa. Piloting sustainable hpc for research at columbia. May 2010. A position paper for the NSF Workshop on High Performance Computing Center Sustainability.

[23] R. G. Brown. What's a Beowulf? In Engineering a Beowulf-style Compute Cluster. http://www.phy.duke.edu/~rgb/brahma//beowulf_book/node9.html, 2003. [Online; Agosto de 2014].

[24] D. Engineering, editor. *Speed Product Development via Virtual Workstation Clustering*. Hewlett Packard Development Company, 2011.

[25] L. Ferreira, G. Kettmann, A. Thomasch, E. Silcocks, J. Chen, J.-C. Daunois, J. Ihamo, M. Harada, S. Hill, W. Bernocchi, and E. Ford. *Linux HPC Cluster Installation*. IBM Redbooks, first edition, 2001.

[26] W. Gropp, E. Lusk, and T. Sterling. *Beowulf Cluster Computing with Linux*. The MIT Press, second edition, 2003.

[27] R. Lottiaux, B. Boissinot, P. Gallard, G. Vallée, and C. Morin. OpenMosix, OpenSSI and Kerrighed: A Comparative Study. Research Report RR-5399, INRIA, 2004.

[28] C. N. Monica Vlădoiu, Zoran Constantinescu. Availability of computational resources for desktop grid computing. In *BULETINUL UniversităŃii Petrol - Gaze din Ploieşti*, volume LXI, pages 71–76, 2009.

[29] D. Ridge, D. Becker, P. Merkey, and T. Sterling. Beowulf: Harnessing the power of parallelism in a pile-of-pcs. In *Proceedings, IEEE Aerospace*, pages 79–91, 1997.

[30] S. A. Salinas, C. G. Garino, and A. Zunino. Sistema de predicción y evaluación de disponibilidad operativa de recursos en desktop grids. In *40JAIIO - HPC 2011*, pages 105–116, 2011.

[31] T. M. Taha, P. Yalamanchili, M. Bhuiyan, R. Jalasutram, C. Chen, and R. Linderman. Neuromorphic algorithms on clusters of playstation 3s. In *WCCI 2010 IEEE World Congress on Computational Intelligence*, pages 3040–3049, 2010.

Técnicas de Metaprogramación Para el Diseño de EDSLs en C++0x11

Marcelo Arroyo
Departamento de Computación
Universidad Nacional de Río Cuarto

Resumen

Un lenguaje de dominio específico (DSL) es un lenguaje computacional especializado a un dominio de aplicación. El desarrollo de estos lenguajes es una práctica cada vez mayor como técnica de ingeniería de software para resolver problemas en dominios de aplicación particulares. Las técnicas comúnmente utilizadas para su desarrollo son el desarrollo de parsers e intérpretes o generadores de código muchas veces utilizando herramientas externas (ej: lex/yacc, JavaCC, etc).

Algunos de los lenguajes de programación modernos incluyen mecanismos que permiten el desarrollo de lenguajes de dominio específico en el mismo lenguaje anfitrión. En éste caso se los conoce como ***embedded domain specific languages*** *(EDSLs).*

En este trabajo se presentan los mecanismos y técnicas de programación modernas comúnmente utilizadas en C++ que permiten definir EDSLs y se desarrollan algunos ejemplos. En particular se describen las técnicas de meta-programación basadas en plantillas y otras técnicas complementarias.

1. Introducción

El desarrollo de lenguajes de dominio específico (DSL) es una práctica bastante común en ingeniería de software. Un lenguaje de dominio específico es un lenguaje computacional especializado a un dominio de aplicación. Los DSLs pueden ser utilizados como lenguajes de marcado especializados (ej: HTML), lenguajes de descripción de modelos, de datos (ej: JSON), lenguajes de especificación o de programación específicos.

Existen numerosos ejemplos de DSLs tales como lenguajes de configuración de aplicaciones, de consulta (ej: SQL), de presentación (ej: PDF), etc.

El enfoque tradicional para la implementación de DSLs se basa en técnicas de diseño de compiladores e intérpretes para lo cual se requiere el desarrollo de reconocedores de lenguajes (parsers), intérpretes o traductores, incluyendo las etapas específicas como análisis sintáctico, semántico y generación de código. Estas técnicas han sido ampliamente estudiadas en el campo de compiladores por lo que existen herramientas especializadas basadas en especificaciones que generan el código requerido, principalmente para las etapas de análisis exicográfico y sintactico. Entre algunas de estas herramientas ampliamente utilizadas podemos mencionar desde las más simples como lex-yacc y javacc hasta lenguajes y formalismos específicos como las basadas en gramáticas de atributos[1] como Eli[9], Silver[10] o lenguajes de especificación y/o programación basados en sistemas de reescritura como *Rascal*[11], el cual ha derivado del desarrollo del framework ADF-SDF[12]. Estos ambientes contienen lenguajes de especificación específicos para cada etapa.

El uso de herramientas externas presenta algunas desventajas tales como la mayor complejidad del sistema de generación de código (build-system), mayores dependencias de bibliotecas y/o herramientas externas, control sobre el código generado y requiere contar con desarrolladores familiarizados en tales

herramientas, entre otros.

Algunos lenguajes de programación modernos incluyen mecanismos que permiten el desarrollo de mini-lenguajes como extensiones del mismo lenguaje anfitrión. Entre los mecanismos que proveen se pueden mencionar desde los más simples como introspección, reflexión, sobrecarga de operadores, lenguajes de macros y plantillas (c++, Lisp, Scala, Haskell y otros); hasta los más sofisticados como la extensibilidad de la gramática inicial y definición de la semántica de las nuevas construcciones como en Seed7[4], Stratego/XT y otros.

En este trabajo se presentan los mecanismos que dispone C++ (en su nuevo estándar C++0x11[2]) y algunas de las técnicas modernas de programación utilizadas, principalmente las de meta-programación con plantillas (template meta-programming).

La principal contribución de éste trabajo es la compilación de técnicas y mecanismos utilizables en el nuevo estándar de C++ para la definición de EDSLs. Las técnicas y mecanismos se describen en forma simple y concisa para su mejor lectura y comprensión. Es difícil encontrar artículos que describan un buen conjunto de técnicas, sus posibles combinaciones conjuntamente con sugerencias de buenas prácticas de programación en C++. Además se intenta impulsar a los desarrolladores de software que utilicen los nuevos paradigmas de programación en los lenguajes modernos. Las técnicas descriptas en este artículo son comparables conceptualmente a las técnicas de meta-programación en lenguajes modernos como Haskell y Scala.

En la siguiente sección se describen algunos conceptos sobre meta-programación en C++ con algunos ejemplos. En la sección 3 se describen técnicas para representar expresiones y su evaluación. En la sección 4 se describen algunas aplicaciones y algunas bibliotecas comúnmente usadas que utilizan técnicas de meta-programación. Finalmente se describen las conclusiones algunas experiencias realizadas y trabajo futuro relacionado con estas técnicas.

2. Meta-programación en C++

La meta-programación es un conjunto de técnicas y mecanismos que permiten el desarrollo de programas que manipulan programas.

Como se ha descripto en la sección anterior, en su forma mas tradicional, se pueden implementar escribiendo procesadores de lenguajes, esto es, intérpretes o compiladores como herramientas externas[1] tal como es práctica común en la construcción de procesadores de lenguajes. Estas técnicas han sido ampliamente estudiadas en la bibliografía sobre teoría de lenguajes y compiladores tales como gramáticas, autómatas, técnicas y algoritmos de análisis léxico y sintáctico (parsing), análisis semántico y síntesis de código o interpretación.

Sin embargo, el uso de estas herramientas externas en el proceso de desarrollo de software moderno tiene algunas desventajas, entre las cuales podemos mencionar:

1. Las limitaciones del lenguaje de especificación tanto para la sintaxis como para la semántica.

   Ejemplos: resolución de conflictos de gramáticas ambiguas, semántica con familias de gramáticas de atributos poco expresivas (L-atribuidas, S-atribuidas), . . .

2. La dificultad de interacción del código generado con el resto del sistema.

   A menudo la herramienta genera código poco modular o en un estilo inconsistente al resto del sistema.

3. El uso de diferentes modelos o paradigmas de programación

4. La complejidad adiconal del sistema de desarrollo (build system). El sistema de compilación requerirá pasos previos de invocación y compilación de módulos generados por estas herramientas.

5. La introducción de mayores dependencias de herramientas y bibliotecas de software.

[1] Las cuales pueden ser implementadas en un lenguaje diferente.

Un lenguaje de programación con sintaxis extensible permitiría el desarrollo de lenguajes de dominio específico dentro del mismo lenguaje anfitrión. Si bien hay algunos lenguajes de programación con sintaxis extensible como *Seed7*, esto es difícil de lograr en otros lenguajes existentes.

La tendencia general en varios lenguajes de programación modernos es la inclusión de mecanismos que permiten realizar meta-programación directamente en el lenguaje sin depender de herramientas externas.

Lisp, por ejemplo, permite la manipulación de programas ya que es posible visualizar a los programas como datos y viceversa mediante el uso de macros y evaluación. En Lisp, las macros se evalúan en tiempo de compilación por lo cual permite que en Lisp sea una práctica común realizar *extensiones*.

Como desventaja, un EDSL basado en Lisp, tendrá la limitación de la sintaxis minimalista del lenguaje anfitrión.

La meta-programación en lenguajes dinámicos como Python, Smalltalk, Ruby y otros se realiza mediante los mecanismos de introspección y reflexión en forma dinámica (en tiempo de ejecución).

La *introspección* permite extraer información de objetos su tipos, representación de datos y otra información dinámicamente (en tiempo de ejecución).

La *reflexión* va un poco más allá que la introspección, permitiendo, además, modificar la estructura y comportamiento de un programa dinámicamente.

La mayoría de los lenguajes de programación estáticamente tipados (C++, Java, etc) proveen mecanismos de reflexión e introspección limitados por lo que la meta-programación dinámica no es una alternativa efectiva. C++ permite una introspección muy limitada, básicamente basada en el uso de `sizeof()` y `typeid()`. Esta última característica requiere que se habilite o genere código para usarse en tiempo de ejecución (`Run-time Type Onformation (RTTI)`).

Para la mayoría de los lenguajes de programación se han desarrollado herramientas externas o bibliotecas de meta-programación que permiten manipular programas a nivel de código fuente, como por ejemplo `OpenC++`[7] para C++ y `OpenJava`[8] para Java.

Estas herramientas pueden verse como preprocesadores de programas fuentes C++ y Java que permiten desarrollar bibliotecas y/o aplicaciones que pueden actuar como *plug-ins* en una suite de compilación, generando código y/o realizando optimizaciones nivel de código fuente.

Por ejemplo, estas herramientas se han utilizado para generar *stubs* o *proxies* para CORBA[19] o incluir nuevos mecanismos como aspectos.

2.1. Plantillas en C++ y la STL

En C++, el uso de plantillas (templates) permite definir abstracciones como tipos genéricos (que toman tipos como argumentos) que luego serán instanciados con tipos concretos.

Más adelante se verá que las plantillas permiten extender la limitada introspección de C++ por medio de los *type traits*.

Una plantilla permite especificar fácilmente estructuras parametrizadas por tipos (ejemplo: *contenedores homogéneos*) y funciones genéricas, tal como se muestra en la figura 1

Listado 1: Plantillas C++

```
template<typename T1, typename T2>
struct pair {
  pair(const T1 & first, const T2 & second)
    : fst(first), snd(second)
    {}
  ...
  T1 fst;
  T2 snd;
};

template<typename T1, typename T2>
T1 first(const pair<T1,T2> & p)
{
  return p.fst;
}
...
pair<int,int> p(0,1);
int x = first(p);
```

En el ejemplo del listado 1 se definen dos plantillas: Una define el tipo `pair<T1,T2>` y la otra una función genérica sobre tipos `pair<T1,T2>`.

Las dos últimas líneas del listado realizan dos *instanciaciones*: La anteúltima línea instancia el tipo del objeto `p` al tipo concreto `pair<int,int>`, la siguiente línea instancia la plantilla de función `T1 first<T1,T2>(const pair<T1,T2>& p)` con la función concreta `int first(const pair<int,int>& p)`.

Esto muestra que las plantillas y el mecanismo de instanciación actúan como *esqueletos* de código a partir de los cuales el compilador genera una o más versiones en base a las instanciaciones producidas.

Inicialmente, las plantillas fueron utilizadas simplemente para la definición de abstracciones de estructuras de datos (abstract containers) y algoritmos genéricos. Estas definiciones forman parte de la biblioteca de plantillas estándar: la Standard Template Library (stl).

Listado 2: Ejemplo de uso de `transform`

```
int add1(int i)
{
  return i+1;
}
{
  std::vector<int> v = {0,1,2,3};
  std::vector<int> r;
  r.resize(v.size());
  std::transform(v.begin(),v.end(),r.begin(),
                 add1);
  // r={1,2,3,4}
}
```

La `stl` define *abstract containers* o tipos de datos parametrizados como `std::vector<T>` y `std::list<T>` con otras utilidades y abstracciones relacionadas como por ejemplo, los iteradores.

También la `stl` define algoritmos (funciones) genéricos como `find()`, `transform()`, `sort()`, etc.

Estos algoritmos operan sobre colecciones genéricas las cuales pueden ser recorridas por medio de *iteradores*: objetos que son abstracciones de punteros a los elementos contenidos en una colección y equipados con operadores de acceso relativo (siguiente, anterior, etc).

El listado 2 muestra el uso del algoritmo `std::tranform()`, el cual permite aplicar una función dada como parámetro (en este caso `add1`) a los elementos de un contenedor abstracto y dejando los resultados en otro (o el mismo) contenedor. Esta función es similar a `map` de Haskell.

Listado 3: Factorial

```
template<int N>
struct factorial {
  enum {value = factorial<N-1> * N};
};

// specialization
template <>
struct factorial<0> {
  enum {value = 1};
};

factorial<4> f4; int v = f4.value;
```

El mecanismo de instanciación de plantillas selecciona los tipos más concretos o específicos y además permite definir plantillas recursivas como se muestra en el listado 3. Estas dos características permiten que el lenguaje de plantillas sea Turing completo, definiendo un lenguaje funcional sobre tipos y valores constantes.

Note la definición de una especialización: Se omite el parámetro en `template<...>` y se incluye a continuación del identificador de la definición con un tipo o valor constante concreto.

Por supuesto que no se debe abusar de la instanciación recursiva de plantillas porque cada compilador define un número máximo de profundidad de instanciaciones.

Si se necesita definir una función con cero o mas argumentos de diferentes tipos se podrá usar el esquema mostrado en el listado 4.

Listado 4: Plantillas variádicas (esquema)

```
r f()
{
  // ...
}

template<typename T, typename... R>
r f(T arg1, R... others)
{
  g(arg1);
  f(others...);
  // ...
}
```

Una plantilla con un número variable de argumentos se conoce como *variadic template* y está soportado desde C++11. Con C++03 se debían definir un conjunto de plantillas con diferente número de argumentos hasta un máximo definido.

El operador sufijo `...` aplicado a una declaración describe una *tupla* (o paquete) de argumentos. El operador (sobrecargado) `...` cuando se aplica a una expresión realiza un *desempaquetado* de los argumentos, extrayendo el primero y reconstruyendo otro paquete con los subsiguientes.

Cabe destacar que el estilo de definición de abstracciones genéricas se basa en la definición por casos, similar a las técnicas de los lenguajes de programación funcional modernos como ML o Haskell.

De esta forma, C++ con platillas se convierte en un lenguaje de dos niveles. Un nivel que permite computar entidades estáticamente y el resto dinámicamente, lo que lo hace adecuado para aplicar técnicas de meta-programación.

Posteriormente, casi por accidente[13][14], se fue descubriendo que el lenguaje de plantillas era más poderoso de lo que se pensó inicialmente y se desarrollaron las técnicas de meta-programación, algunas de las cuales se describen en este trabajo.

2.2. Un EDSL de ejemplo

Imaginemos un EDSL dentro de C++ que represente consultas SQL de la forma:

```
db * select(f1,f2).from(t).where(f1==v1 &&
f3>v2);
```

Este fragmento[2] del EDSL describe sentencias utilizando la técnica de encadenamiento de métodos (method chaining) para sus cláusulas. Para poder representar las expresiones usadas como argumentos (por ejemplo, en la cláusula `where`) se requieren técnicas de meta-programación para la definición de los tipos de las expresiones junto con el uso de *factorías* de expresiones, que en este caso toman la forma de los operadores del lenguaje sobrecargados (`operator==()`, `operator>()`, `operator&&()`).

En la sección 3 se describen las técnicas usadas para la creación, representación y evaluación de expresiones de un EDSL.

Listado 5: select() con plantillas variádicas

```
sql & sql::select(field & f)
{
  fields.add(f);
  return *this;
}

template<typename... T>
sql & sql::select(field & f, T... others)
{
  fields.add(f);
  return select(others...);
}
```

Las cláusulas `select(f1,f2)` y `from(t)` en el EDSL dado como ejemplo requieren un número variable de argumentos, lo que en C++0x11 se pueden definir fácilmente con plantillas variádicas como se muestra en el listado 5 muestra el uso de funciones variádicas usando *variadic templates* para una posible implementación de `select()`. La cláusula `form` se definiría de manera análoga.

De esta forma, la función `select()` se invoca recursivamente con los argumentos siguientes. En el caso descripto se requiere al menos un argumento y todos son del mismo tipo (`field`).

Este mecanismo es similar a la manipulación de listas de Haskell, ML o Lisp.

C++ es un lenguaje de programación multiparadigma, es decir que incluye mecanismos que permiten la programación imperativa (heredada de C), programación orientada a objectos (clases, herencia, invocación dinámica), programación funcional (alto orden) y programación generativa (templates).

Es posible implementar diferentes tipos de polimorfismo de varias formas: basado en herencia, genericidad (plantillas con tipos paramétricos) por sobrecarga de operadores (polimrfismo ad-hoc), o utilizando trucos de bajo nivel (uniones, etc).

[2]Este fragmento pertenece al EDSL `SQL` del framework *Ultimate++*[20].

Esta variedad de mecanismos hace que C++ se haya transformado en un lenguaje muy versátil y lo hace muy adecuado para definir diferentes tipos de abstracciones[3].

Entre las técnicas utilizadas para definir EDSLs en C++, se pueden mencionar:

- Computación en tiempo de compilación (meta-funciones)
- Evaluación (instanciación) parcial.
- Computación de propiedades de tipos (*traits*)
- Representación y evaluación de estructuras computacionales (expresiones)

En las siguientes secciones se describen cada una de las técnicas mencionadas.

2.3. Meta-funciones

Una *meta-función* permite *computar* en tiempo de compilación. Contrario a su nombre, una meta-función se define mediante un tipo (*struct* o *class*).

Listado 6: Meta-función If

```
template<bool, class L, class R>
struct If {
        using type = R;
};
template<class L, class R>
struct If<true> {
        using type = L;
};
...
// i:int if ARCH64 is true
If<ARCH64,int,long>::type i;
```

En el listado 6 se muestra un ejemplo de una meta-función. En este caso la metafunción computa un tipo. Este tipo de metafunciones son las más comunes en meta-programación. Algunos autores definen a la meta-programación como *computación de tipos*.

Con la plantilla `If` es posible realizar selección de código o tipos sin necesidad de usar un pre-procesador.

[3] *C++ is a language for defining abstractions - Bjarne Stroustrup.*

El listado 3 de la meta-función *factorial* descripto anteriormente también es un ejemplo de computación (o generación) de tipos estáticamente (`factorial<0>`, ..., `factorial<4>`). Los valores generados, en realidad están encapsulados en símbolos de una enumeración.

Cabe aclarar que en ambos ejemplos de meta-funciones se computan tipos. Esto significa que no requiere memoria en tiempo de ejecución.

En ambas definiciones, éstas se realizan por casos, al estilo de los lenguajes funcionales modernos. El segundo caso em ambos ejemplos definen una *especialización* (o caso base).

2.4. Evaluación parcial

Un evaluador parcial[6] realiza evaluaciones estáticas (en tiempo de compilación) y dinámicas (en tiempo de ejecución). La evaluación estática genera un programa especializado denominado *programa residual*. Un evaluador parcial debe realizar algún tipo de *binding time analysis*, esto es, rotular construcciones del lenguaje y datos como estáticos o dinámicos. Un lenguaje de este tipo se conoce como un *lenguaje de dos niveles*.

Listado 7: Evaluación parcial

```
template<int N>
int pow(int v)
{
  int r = 1;
  for (int i=0; i<N; ++)
    r *= v;
  return r;
}

template<>
int pow<3>(int v)
{
  return v*v*v;
}
...
assert(pow<5>(2) == 32); // generic version
assert(pow<3>(3) == 27); // spec. version
```

C++ califica como un tal lenguaje: las plantillas permiten instanciar entidades (tipos y valores constantes) estáticamente. Las sentencias y expresiones basadas en variables se ligan o instancian dinámicamente.

En el listado 7 se muestra un ejemplo de evaluación parcial en C++.

Las plantillas permiten hacer evaluadores parciales mediante *instanciación parcial*, permitiendo realizar optimizaciones o generadores de código especializados de alto nivel (dependientes del dominio de aplicación).

Estas técnicas se han utilizado masivamente en bibliotecas de computación numérica como `Blitz`[18] para generar código eficiente en casos especiales.

2.5. Traits

Los *traits*[4] describen propiedades o características sobre tipos.
En el listado 8 se muestra un *trait* para indagar si un dado tipo es un puntero o no.

En C++0x11 el archivo de cabecera `type_traits` define un buen conjunto de *traits*, entre los cuales se encuentra una versión mas realista de `is_pointer`.

Listado 8: Type traits

```
template<typename T>
struct is_convertible {
  enum {value = false};
  typedef typename T base_type;
};

template<typename T>
struct is_pointer<T*> {
  enum {value = true};
  typedef typename T base_type;
};
...
int *p;
...
assert(is_pointer<p>.value,true);
assert(is_pointer<int>.value == false);
```

Utilizando `traits` es posible definir *conceptos*. Un *concepto* permite definir restricciones o requerimientos sobre los tipos que pueden ser parámetros de ciertas plantillas. En el listado 9, se define una plantilla con una restricción sobre el tipo recibido como parámetro, requiriendo que el tipo usado en la plantilla sea *comparable*[5].

Listado 9: Concepto

```
template<typename T>
struct searchable {
  static_assert(
        std::is_comparable<T>::value
  );
  ...
};
```

Con *traits* es posible emular generecidad restringida tal como se soporta en Eiffel o Java.

3. Representación de expresiones

Para definir un EDSL se requiere representar las entidades del lenguaje a implementar. Generalmente el lenguaje contiene sentencias y expresiones. Comúnmente las sentencias se representan por funciones u objetos que toman expresiones como argumentos. Existen lenguajes completamente basado en expresiones.

En el ejemplo del SQL EDSL de la sección 2, la cláusula `where` (implementado como un método) requiere tomar una expresión lógica como argumento.

Si bien el parámetro actual (`f1==v1 && f3>v2`) parece ser una expresión lógica C++, en realidad lo que `where` requiere es una *representación* de la expresión, no el resultado de su evaluación. Esto es porque la cláusula deberá generar (como string) la cláusula *where* correspondiente a SQL.

Esto requiere que la expresión `f1==v1 && f3>v2` genere una *representación* en lugar de ejecutarse. En un lenguaje sin sobrecarga de operadores, ésta expresión podría describirse como `land(eq(f1,v1),gt(f3,v2))` lo que lo hace antinatural ante un programador de aplicaciones.

En C++, la sobrecarga de operadores y la metaprogramación permiten realizar implementaciones simples y eficientes, tal como se muestra en el listado 10. Esta técnica se conoce como *expression templates*.

[4] Think of a trait as a small object whose main purpose is to carry information used by another object or algorithm to determine "policy" or "implementation details". - Bjarne Stroustrup

[5] El trait `is_comparable<T>` chequea que tenga definidos algunos operadores relacionales.

La técnica utilizada en el listado 10 utiliza el patrón *curiously recurring template pattern (CRTP)*[5][6] , el cual permite definir una interface común por medio de un wrapper (*expr*), evitando así que la sobrecarga de operadores definidos en otros componentes sobre tipos genéricos puedan generar versiones ambiguas.

La expresión `f1 == v1 && f3>v2` retorna un valor de tipo

```
land< eq<field,value>, gt<field,value> >
```

El tipo de la expresión encapsula su semántica, luego un meta-programa puede procesar expresiones basadas en pattern-matching, tal como describe en la próxima sección.

Listado 10: Expression templates

```
template<class E>
struct expr {
  operator const E&() const {
    return *static_cast<const E*>(this);
  }
};

template<class L, class R>
struct eq : expr< eq<L,R> > {
  const L& l; const R& r;
  eq(const L& l, const R& r) : l(l),r(r) {}
};

template<class L, class R>
struct l_and : expr< l_and<L,R> > {
  const L& l; const R& r;
  l_and(const L& l, const R& r)
        : l(l),r(r) {}
};

template<class L, class R>
inline eq<L,R> operator==
(const expr<L>& l,const expr<R>& r)
{ return eq(l,r); }

template<class L, class R>
inline l_and<L,R> operator&&
(const expr<L>& l,const expr<R>& r)
{ return l_and(l,r); }
...
```

Si bien en C++0x11 sería posible postergar la evaluación de expresiones usando expresiones lambda (funciones anónimas) como argumentos, éstas no dan lugar a la posibilidad de realizar un procesamiento a medida de las expresiones para poder definir acciones como evaluación no estándar, optimizaciones de alto nivel, transformaciones, generación de código, entre otras.

[6]CRTP ocurre cuando una clase hereda de una clase de plantilla consigo misma como parámetro de la plantilla de la cual hereda.

3.1. Evaluación de expresiones

Un EDSL deberá realizar algún tipo de evaluación de las expresiones del lenguaje. La evaluación podría generar valores, generar código o aún generar información extra para su posterior procesamiento.

En el caso del EDSL de sentencias SQL, un evaluador podría generar consultas a un motor DBMS como se muestra en el listado 11. El evaluador consiste en un conjunto de meta-funciones.

Listado 11: Generación de SQL

```
std::string gen_sql(const field & f)
{
  return f.name;
}

template<typename L, typename R>
std::string gen_sql(const eq<L,R>& e)
{
  return gen_sql(e.l)+'='+gen_sql(e.r);
}

template<typename L, typename R>
std::string gen_sql(const l_and<L,R>& e)
{
  return gen_sql(e.l)+"and"+gen_sql(e.r);
}
...
```

Como se puede apreciar en el listado 11, la evaluación de expresiones se define nuevamente como un conjunto de operadores sobrecargados, donde cada uno opera sobre tipos específicos.

El estilo de programación se basa en la definición por casos, es decir, programación por patrones (pattern matching), al típico estilo de la programación funcional.

También es posible utilizar un estilo más orientado a objetos. En este caso cada expresión contendrá métodos (por ejemplo `gen_sql()`) que realicen la evaluación. Sin embargo este patrón de diseño se considera

intrusivo y en C++ se prefiere un estilo mas funcional, consiguiendo componentes menos ligados. Otra desventaja en pensar a las expresiones como objetos que *saben* generar código es que se necesitará *dynamic dispatching*, es decir, resolución en tiempo de ejecución del método a invocar.

Cabe hacer notar que en ambos estilos no es necesario utilizar *dynamic dispatching*[7] ya que cada función es distinguible porque tienen diferentes signaturas. En el caso de usar el estilo orientado a objetos, cada método se corresponde con un tipo concreto diferente (polimorfismo paramétrico), por lo que tampoco hay necesidad del uso de *funciones virtuales*.

3.2. Almacenamiento de expresiones

Algunos EDSLs pueden requerir que las expresiones puedan ser almacenadas para asignarles un identificador. Uno de los problemas que tenía C++03 era que había que utilizar trucos muy ingeniosos para hacerlo ya que el tipo generado por una expresión puede ser muy compleja.

Por ejemplo, para asignar la expresión `f1==v1 && f3>v2` a una variable `e` hay que conocer el tipo de la expresión:

```
land<eq<field,value>,gt<field,value>> e =
f1==v1 && f3>v2;
```

Sería deseable poder definir una variable de algún tipo mas general y luego poder manipular los objetos.

Es posible definir un wrapper `base_expr` que sea el tipo base de (expr) tal como se muestra en el listado 12.

La técnica se basa en polimorfismo basado en herencia por lo que requiere definir al menos una función miembro virtual en cada tipo de expresión para realizar la evaluación de expresiones almacenadas en una variable.

Se requiere polimorfismo basado en herencia porque es imposible recuperar el tipo del objeto embebido sin el uso de casts por parte del usuario. Esta técnica se basa en la idea de *type erasure*.

Habiendo definido las cosas en un estilo más orientado a objetos podría almacenar expresiones:

```
base_expr e;
e = f1==v1 && f3>v2;
```

El costo de esta técnica es que para manipular las expresiones se requiere definir métodos virtuales, lo cual afecta a la performance y además se introduce el método `gen_sql()` en cada tipo de manera intrusiva.

Listado 12: Almacenamiento de expresiones

```
base_expr {
};
template<class E>
struct expr {
  virtual std::string gen_sql();
  ...
};
struct expr_container {
  ...
  template<typename T>
  T & operator=(T & expr) { e = &expr; }
  std::string gen_sql() { e->gen_sql(); }
  base_expr * e;
};
```

Si bien es posible almacenar valores de tipos genéricos sin perder el tipo y sin utilizar métodos virtuales, su implementación es bastante engorrosa. La idea es definir un método contenedor estático como:

```
struct base_expr {
...
  template<typename T>
  T & get_set(T & expr) {
    static std::map<base_expr*,T> store;
    return store[&expr] = expr;
  }
}
```

De esta forma, para cada tipo `T` se definirá un contenedor (map) homogéneo con alcance local a la versión `get_set<T>()` correspondiente.

El operador de asignación y el constructor de copia pueden utilizar el método `get_set()`.

En C++0x11, el problema de almacenar valores de tipos complejos se reduce al uso de `auto`:

```
auto e = f1==v1 && f3>v2;
```

ya que realiza *inferencia de tipos*.

De esta manera el problema de almacenar objetos de tipo muy complejo y arduo de describir se reduce significativamente.

[7] Conocido en C++ como *funciones virtuales*.

4. Ejemplos de EDSLs

El lenguaje SQL embebido en C++, como parte de la biblioteca del framework Ultimate++[20], es un ejemplo de un EDSL, el cual el autor de éste artículo ha contribuido.

La biblioteca estándar de plantillas de C++ define de alguna manera un EDSL en torno a los *abstract containers*, sugiriendo un estilo de programación mas funcional que orientado a objetos.

El nuevo estándar incluye versiones mejoradas de `tuple` y otras estructuras haciendo uso de los nuevos mecanismos como *variadic templates*. Estos permiten la definición de tuplas con un número arbitrario de componentes.

Además incluye un conjunto de *traits* muy útiles para implementar *conceptos*.

`Boost`[15] incluye bibliotecas de plantillas que extienden, generalizan y complementan la `STL`. Además incluye una biblioteca de plantillas para realizar metaprogramación, `boost::mpl` que brinda facilidades como meta-funciones, secuencias y un framework para definir EDSLs.

Listado 13: Ejemplo: Spirit

```
...
rule decl = *(ident_ >> char('=')) >>
            (double_ | int_);
phrase_parse("x=y=8", decl, space);
...
```

`Boost` incluye algunos EDSLs, los cuales están basados en `boost::mpl`. Un EDSL para destacar es `boost::spirit` el cual permite definir parsers basado en el formalismo conocido como *expression grammars*[16].

El listado 13 muestra un fragmento de un programa C++ que define un parser para una gramática simple.

donde `decl` es un parser de una sentencia de (posible) asignación a variables de un valor real o entero. `ident_`, `int_` y `double_` son parsers predefinidos por `spirit`. `char_(c)` es un parser que reconoce el caracter '='.

La regla `decl` es equivalente a la EBNF:

```
decl ::= (ident '=')* (double | int)
```

El tipo `rule` (un parser polimórfico) está definido usando las técnicas descriptas en la sección anterior.

El operador *shift derecho* denota una secuencia, el operador *bitwise or* denota alternativa el operador * (prefijo) indica cero o más ocurrencias de su operando (clausura de Kleen).

La función `phrase_parse()` toma una secuencia (generalmente un string), un parser y un *skipper* opcional (en este caso, salta los espacios entre tokens).

Spirit permite además, incluir acciones semánticas a cada parser. Una expresión $P[f]$ asocia una función f al parser P.

Listado 14: Spirit con expression templates

```
vector<int> v;
// ...
for_each( v.begin(), v.end(),
          cout << (_1 + 1) // (*)
        );
```

Spirit utiliza `boost::lambda`, otro EDSL que implementa abstracciones *lambda*(funciones anónimas) en C++.

El listado 15 muestra un ejemplo de uso de `boost::lambda`.

La expresión `(*)` construye un objeto que representa una expresión sobre un valor enviado al output stream `cout`. La expresión `_1` es un *placeholder*, el cual representa el valor pasado por `for_each` como argumento al objeto que representa la expresión.

C++0x11 incluye la creación de expresiones *lambda* en forma nativa, con la ventaja de que es posible definir las variables *capturadas* del ambiente.

En C++11 es posible reescribir el ejemplo del listado 15 como:

Listado 15: Spirit con expression templates

```
vector<int> v;
// ...
for_each( v.begin(), v.end(),
          cout << [](int x) {x+1;}
        );
```

El proyecto `Boost` ha influenciado mucho en la incorporación de nuevos mecanismos en C++, tales

como expresiones *lambda* y *traits*.

Existen varias bibliotecas de computación numérica, como `Blitz++`[18] que explotan las técnicas de meta-programación con plantillas en C++ para representación de datos, expresiones y computaciones. `Blitz++` hace énfasis en lograr máxima performance (similar a código Fortran optimizado) pero en un EDSL natural para el programador. `Blitz++` define arreglos multidimensionales, slicing y define operadores sobre matrices, vectores y tensores. Las técnicas de meta-programación utilizadas permiten seleccionar patrones de código óptimos en base a la información que arrastran las expresiones definidas en el programa.

El autor de este artículo ha desarrollado un EDSL (`parallel template library (ptl)`) para programación concurrente y paralela. `Ptl` define varios patrones de diseño paralelos aplicables tanto a modelos de memoria compartida (threads) y sistemas distribuidos (MPI).

El listado 16 muestra algunos constructores de patrones de paralelismo en `ptl`.

El algoritmo `transform` tiene la misma interface que `std::transform`, pero aplica en forma paralela una expresión `_1*2` a cada elemento del vector `v`.

El ejemplo del *pipeline* toma un generador, un grupo de tareas y un consumidor. Cada uno de estos elementos genera valores, los procesa y los comunica al siguiente elemento en forma concurrente.

Listado 16: Parallel Template Library

```
using namespace ptl;
vector<int,N> v;

// parallel map (v[i] *= 2, 0>=i<N)
transform(v.begin(), v.end(), _1*2);

// pipeline example
input >> task1 >> task2 >> sink;

// concurrent/parallel execution
task1 || task2 || task3;
// implicit barrier here
```

El operador *logical or* genera tareas que se ejecutan concurrentemente y espera a su terminación.

Estas abstracciones forman un conjunto de patrones que permiten hacer *concurrencia estructurada* y están implementadas usando los mecanismos estádares de C++0x11.

5. Conclusiones

C++ se ha convertido en un lenguaje de programación multiparadigma y que soporta técnicas de meta-programación con plantillas lo que lo hace muy adecuado para la implementación de EDSLs.

Las técnicas de programación utilizadas en el sublenguaje de plantillas son similares a las de la programación funcional moderna, principalmente utilizando pattern-matching y definiciones por casos.

El nuevo estándar C++0x11 incluye nuevos mecanismos para facilitar la definición de EDSLs e incluye algunas soluciones a los principales problemas encontrados por los programadores de EDSLs, como por ejemplo, el poder evitar la copia de valores temporarios (move-semantics) en las expresiones.

El concepto de *move semantics* permite definir constructores de copia sobre valores temporarios permitiendo que se *capture* la representación en lugar de copiarla.

Por ejemplo, es posible refinir referencias a *rvalues* (o valores temporarios inmutables), evitando así la invocación de constructores de copia.

La inclusión de la palabra reservada **`auto`** facilita la asignación de *expression templates* y otros tipos complejos generados a partir de plantillas.

La inclusión de funciones *lambda* (o clausuras) permite pasar expresiones como parámetros permitiendo implementar evaluación demorada o *lazy*.

Finalmente, la estandarización del modelo de memoria y de la biblioteca de threads junto con algunos mecanismos de lanzamiento de threads (como `async`) y sincronización de alto nivel (como `future<T>`) hacen la programación concurrente más segura y legible.

Actualmente el autor de este artículo se encuentra trabajando en la extensión de un EDSL basado en gramáticas de atributos en el cual se está trabajando en la computación estática de las dependencias entre las instancias de los atributos. La idea es utilizar el mismo

compilador de C++ para que compute una relación de dependencia que luego será usada por el evaluador de los atributos con el fin de establecer el orden de evaluación.

Esta tarea es bastante compleja porque el mecanismo de instanciación de plantillas recursivas tiene límites en las implementaciones de los compiladores, ya que la instanciación podría no terminar.

Referencias

[1] D. Knuth. *Semantics of context free languages*. Math Systems Theory2.June 2.Pag: 127-145. 1968.

[2] ISO/IEC 14822:2011. *C++0x11. Working Draft, Standard for Programming Language C++*. http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf. 2011. Revisited in 2014.

[3] Bjarne Stroustrup. *The C++ Programming Language. Fourth edition.* Addison Wesley. Pearson Education. ISBN 978-0321563842. May 2013.

[4] Thomas Mertes. *Seed7, The Extensible Programming Language*. http://seed7.sourceforge.net/.

[5] Jim Coplien. *The Column Without a Name: A Curiosly Recurring Template Pattern*. CppReport. 1995.

[6] Jones N.D. *An introduction to partial evaluation*. ACM Computing Surveys 28, 3 (480-503). Sep. 1996.

[7] OpenC++. `http://opencxx.sourceforge.net/`

[8] OpenJava. `http://opencxx.sourceforge.net/`

[9] *The Eli System*. `http://eli-project.sourceforge.net/`.

[10] Erik Van Wyk, Derek Bodin, Jimin Gao, L. Krishnan. *Silver: an Extensible Attribute Grammar System*. Science of Computer Programming, 75 (1-2):39-54. Elsevier. January 2010.

[11] *Rascal Metaprogramming language*. `http://www.rascal-mpl.org/`

[12] *The Meta-Environment*. `http:www.meta-environment.org/`

[13] Erwin Unruh. *Prime Number Computation*. ANSI X3J16-94-0075/ISO WG21-462.

[14] Todd Veldhuizen. *Using C++ Template Metaprograms*. C++ Report, SIGS Publications Inc., ISSN 1040-6042, Vol. 7, No. 4, pp. 36-43, May 1995.

[15] *Boost C++ Libraries*. `http://www.boost.org/`

[16] Bryan Ford. *Parsing Expression Grammars: A Recognition-Based Syntactic Foundation*. POPL 2004. ACM.

[17] *Ultimate++. A C++ Cross-platform Rapid Application development Framework*. `http://www.ultimatepp.org/`

[18] *The Blitz++ Library*. `http://www.blitz.sourceforge.net/`

[19] *Common Object Request Broker Architecture*. `http://www.corba.org/`

[20] *Ultimate++. C++ cross-platform rapid application development framework.* `http://www.ultimatepp.org/`

Servidor de Comunicaciones Convergentes

Antonio Castro Lechtaler [1 2], Rafael Mario Olivieri [1], Luis Fumagalli [1], Osaba Mariano [1], Francisco Echazú [1],Matías Bönke [1]

[1] Escuela Superior Técnica, Facultad de Ingeniería del Ejército, Instituto Universitario del Ejército, Ciudad Autónoma de Buenos Aires, C1426; [2] Universidad Nacional de Chilecito, Chilecito, F5260, Provincia de La Rioja.

[1] {acastro, lfumagalli } @iese.edu.ar ; [1] {rmolivieri, marianoosaba, fechazú,} ; [1]matias@bonke.com.ar

Abstract

El presente trabajo describe un proyecto cuyo objetivo es desarrollar un servidor de comunicaciones para uso móvil[1] y rápido despliegue, que permita dar soporte de comunicaciones a un sistema de comando y control montado para hacer frente a desastres naturales o antrópicos.

La particularidad de este sistema reside en cumplir con la necesidad de integrar organizaciones que cuentan con distintas doctrinas de trabajo y equipamientos heterogéneos, pero que deben actuar en forma conjunta para responder a distintas situaciones de emergencias, catástrofes o crisis, donde participan organizaciones como fuerzas armadas, bomberos, rescatistas, defensa civil, diversas ONGs[2], entre otros.

1. Introducción.

El concepto de convergencia en los sistemas de comunicaciones se aplica en el ámbito de las empresas y de las instituciones públicas y privadas. Implica el envío y recepción de mensajes de voz, video y datos, junto con las conversaciones de voz por el mismo medio. Esto redunda en beneficios para los usuarios, puesto que permite diversos servicios de comunicaciones por medio de un mismo dispositivo.

Además, permite integrar medios disponibles como la red de telefonía fija (pública y privada), la de telefonía celular, Internet, y otros servicios como correo electrónico y videoconferencia.

2. Contexto de empleo.

Del concepto de convergencia, tomaremos los conceptos necesarios que son aplicables a escenarios donde sea necesario la integración de redes heterogéneas.

Particularmente nos referiremos al despliegue de redes radioeléctricas que trabajan en las bandas de HF - High Frequency, VHF - Very High Frequency y UHF - Ultra High Frequency, que estén integradas a redes territoriales públicas o privadas de telefonía, telefonía celular e Internet.

Estos escenarios son particularmente aplicables a despliegues de equipos de trabajo para hacer frente a cualquier tipo de operación que implique un despliegue de personal y medios en una zona o área dónde es necesario hacer frente a una situación que implique cualquier tipo de acción de apoyo a la comunidad frente a desastres naturales o antrópicos. Allí es necesario asegurar el comando y control de todos los recursos que intervienen, siendo la integración de los medios de comunicaciones la llave del asunto.

Si recordamos el atentado a las Torres Gemelas de Nueva York en septiembre de 2001, las lecciones aprendidas indican que los modernos y confiables sistemas de telefonía fija y celular se vieron colapsados ante el evento, y se destacó la necesidad de integrar grupos operativos heterogéneos (bomberos, policía, rescatistas, etc.) operando en el mismo escenario.

La declaración, que se muestra en la figura 1, expresa como los medios radioeléctricos convencionales operaron eficientemente frente al colapso de los medios públicos.



Figura 1. Declaración que destaca la confiabilidad de los sistemas radioeléctricos en situaciones extremas[3]

[1] Dentro del concepto de convergencia de los sistemas comunicaciones.
[2] Organizaciones No Gubernamentales.
[3] Declaración publicada. Fue reproducida por Harris Communications.

Sin embargo, éstos medios tienen restricciones en cuanto a la cantidad de información que pueden transmitir, y por otro lado, conforman redes operadas por alguna fuerza militar, policial, agencia estatal, etc., que tienen características propias a cada empleo particular.

Estas restricciones condicionan cualquier integración que se pretenda realizar entre estas redes.

La primera restricción se contrarresta con procedimientos y la necesaria disciplina de empleo por parte de los operadores, y la segunda, es la que impone el desafío de lograr la interoperabilidad de esas redes heterogéneas. Lograr la misma es precisamente el objeto de este proyecto.

Los avances tecnológicos posibilitaron el desarrollo de integradores de comunicaciones avanzados por parte de la industria para lograr una mejor interoperabilidad.

Por otro lado, las tecnologías de software y hardware abiertos también avanzaron sostenidos por una importante comunidad de desarrollo integrada por particulares, empresas y universidades.

Por este lado es posible entonces desarrollar un sistema propio, que pueda adaptarse a nuestras necesidades, y contribuir a esa comunidad de desarrollo.

3. Objetivo del proyecto.

Implementar un sistema especialmente pensado para situaciones de emergencias o catástrofes, que integre las redes de radio propietarias de cada organización, con redes de telefonía fija (PSTN), privada (PBX), y la telefonía móvil, que permita comunicaciones de voz y datos de persona a persona, independientemente del medio disponible[4].

El sistema es de empleo dual, es decir para uso estatal militar y civil, con interfaces adaptables a sistemas de comunicaciones militares y civiles.

4. Alcance del proyecto.

Lograr un prototipo de bajo costo, basado en hardware comercial de arquitectura Intel o similar, placas analógicas y digitales para VoIP, software libre y telefonía VoIP, que integre tanto los modernos equipos radioeléctricos de tecnología militar o de uso civil que permitan asegurar el empleo dual.

Para ello se empleará una interfaz analógica de voz de la radio: PTT (push-to-talk) y COR (Carrier Operated Relay).

Este desarrollo está destinado a complementar las capacidades de distintos Organismos tanto Públicos[5] como Privados, que en casos de emergencias o desastres naturales o de otro tipo, sean estos de orden nacional o internacional, requieren una integración entre diversos actores[6]. En particular la integración entre sistemas de comunicaciones militares y civiles.

El sistema debe funcionar sobre redes públicas y privadas, integrándolas indistintamente.

5. Fundamentos metodológicos

El comando y control de operaciones de asistencia a la solución de situaciones de crisis producidas por desastres naturales o antrópicos requiere la conducción y coordinación de diferentes grupos de trabajo que actúan en un área común, dónde normalmente existen riesgos.

Desde un puesto de comando o sala de crisis, la autoridad responsable ejerce la conducción, con el asesoramiento de especialistas que deben contar con información oportuna y confiable. Desde este puesto, con las herramientas informáticas adecuadas, se puede seguir la operación en todo momento y lugar.

Particularmente, en los casos de situaciones de crisis producidas por desastres naturales o antrópicos donde operan grupos heterogéneos, los canales de voz, de persona a persona tienen un valor superlativo, porque establecen una comunicación directa entre los distintos actores, y no imponen mayores restricciones de métodos o procedimientos cuando la urgencia o la importancia de algún evento se quiere destacar.

Se privilegia así, la comunicación directa y el achatamiento de las estructuras de conducción verticales frente a situaciones que requieren la mejor resolución.

Éstas deben ser adoptadas con toda la información que se pueda reunir en el menor tiempo posible.

Nuestro sistema apunta a obtener una integración total de las diferentes organizaciones que puedan actuar mancomunadamente para lograr la mejor eficacia de los sistemas de comunicaciones para un mejor comando y control.

6. Fundamentos tecnológicos.

El servidor de comunicaciones de despliegue rápido se basa en una CPU de arquitectura Intel o similar, sobre la que pueda correr un sistema operativo Linux.

Sobre este sistema operativo se instala el software del sistema Asterisk[7], que permite implementar una central telefónica[8].

Ésta a su vez, permite implementar interfases físicas y de software que facilitan la conexión de canales analógicos

[4] Radio o teléfono.

[5] En particular, el Sistema Integrado de Comando y Control del Ejército Argentino - SITEA.

[6] La idea es posibilitar la integración de sistemas militares con otros civiles para el trabajo conjunto en estos casos especiales. Un ejemplo de ello es la acción que a través directivas del Gobierno Nacional se desarrolla en Haití con motivo del terremoto que asoló ese país.

[7] www.asterisk.org.

[8] Del tipo PBX - Private Branch Exchange

tipo FXS/FXO, E&M o E1/T1. Implementa también canales VoIP sobre cualquier red IP LAN o WAN.

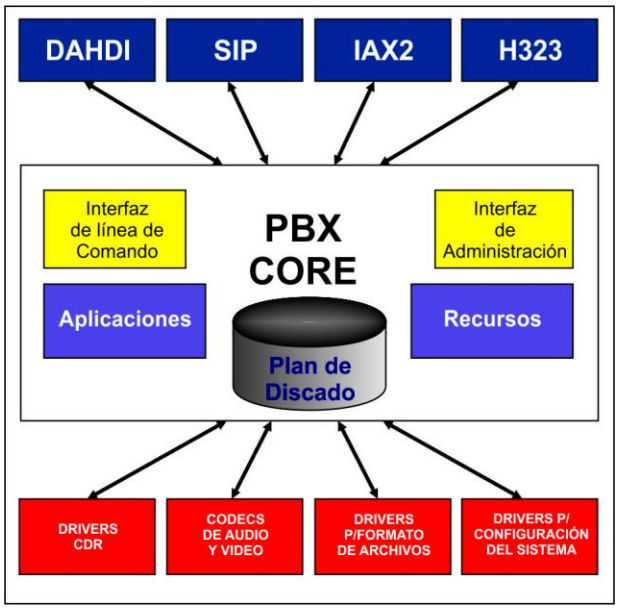


Figura 2 - Arquitectura Asterisk

Como se trata de software abierto, es posible implementar nuevas interfaces que integren equipos de radio, que requieran al menos un canal de voz y otro de control para el PTT Push to Talk.

Posee otra ventaja por cuanto cuenta con una interfaz de aplicaciones AGI - Asterisk Gateway Interface, que permite desarrollar aplicaciones propias empleando la capacidad de procesar y conmutar llamadas de Asterisk.

En breve síntesis: el uso de Asterix permite crear una matriz de conmutación de canales para integrar diferentes medios de comunicaciones, que a su vez implementan tecnologías diferentes.

7. Descripción del sistema propuesto.

Las redes ad hoc, para situaciones de emergencias o catástrofes se caracterizan por la heterogeneidad de sus enlaces, puesto que cada uno se adecua a un empleo y situación táctica particular.

A su vez, esas redes pueden estar integradas a redes territoriales públicas o privadas.

El proyecto se aplica particularmente sobre redes IP, redes de radio, y redes de telefonía convencional y celular. Se basa en la arquitectura abierta para VoIP Asterisk para proporcionar:

- ✓ Repetidor Controlador Full Dúplex.
- ✓ DTMF para control remoto de equipos de radio.
- ✓ Base de controlador remoto.
- ✓ Interconexión entre sistemas o nodos, a través de internet (VoIP) o redes privadas.
- ✓ Teléfono de acceso (saliente y entrante), incluyendo el control remoto a través de DTMF.
- ✓ Enrutamiento inteligente de datos.

Un sistema de comunicaciones ad hoc utilizando esta tecnología permitirá integrar mediante interfaces de bajo costo, sobre computadoras, notebooks, tablet, etc., comunicaciones de voz y datos entre las distintas redes y niveles de comando de diferentes organizaciones.

Para un sistema de conducción, particularmente frente a situaciones de emergencia o catástrofe, esto permitirá la integración de la red de sensores de campo y la difusión inherente de la información para toma de decisiones hacia los diferentes niveles de comando.

Además, la integración de comunicaciones de voz y de mensajes[9], tanto sean desde equipos antiguos o modernos, de dispositivos de comunicaciones de uso civil, de elementos de seguridad o de organizaciones públicas y privadas.

La adaptación de los dispositivos de comunicaciones (radios) depende de cada modelo particular puesto que se hace por medio de la interfaz de audio del equipo.

Las redes telefónicas y de celular se integran por su interfaz específica.

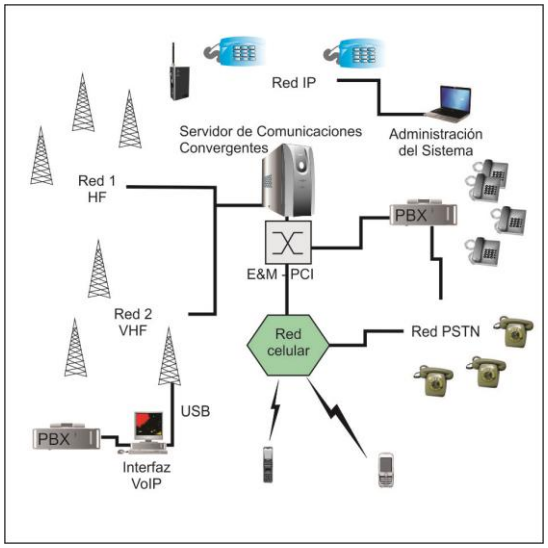


Figura 3 - Esquema de implementación

Como se dijo, el servidor integrador se basa en arquitectura PC Intel, determinando a priori dos niveles de nodos con opciones disponibles como interfaz de hardware:

- ✓ Nodo simple (pequeñas organizaciones) empleando una Interfaz USB de radio, que es un dispositivo al que se conecta un único sistema de radio a la computadora a través del puerto USB. Es un nodo simple, económico y de rápido despliegue.

[9] Sean estos tanto de voz, como MNS o Correo.

- ✓ Nodo maestro, para niveles mayores que lo requieran, como Centros de Comunicaciones Fijos, empleando algún tipo de tarjeta PCI con una interfaz telefónica de tipo E & M, adaptada por software a desarrollar por el proyecto. Esto permite integrar más de dos redes, dependiendo de la capacidad del hardware. Es un nodo integrador de redes.

La Figura 3 muestra un esquema de implementación, donde se ven diferentes redes (radio, teléfono, red de telefonía móvil, etc.) integradas en un nodo.

8. Implementación tecnológica.

8.1. Aspectos generales al desarrollo.

La solución adoptada fue emplear Asterisk PBX como nodo en una red IP y como un nodo repetidor/controlador para la red de radio.

La arquitectura de Asterisk permite hacerlo mediante una aplicación escrita en lenguaje C, "app_rtp", que se ha tomado, en principio, sin modificaciones.

Con ello, se pueden interconectar varios nodos en la red IP en modo "full dúplex", cada uno enlazado con una o más redes de radio.

Para las interfaces a radio se han empleado placas estándar de telefonía con interfases físicas "ad hoc", adecuadas a cada radio. Cabe aclarar que el modo "full dúplex" con esta configuración estará disponible solo en la red IP usando la tecnología VoIP, o en las redes telefónicas dónde naturalmente es nativo, pero no en las redes de radio.

Cada nodo puede operar solo, o interconectado con otros mediante una red IP. No existe un número máximo, ni óptimo de nodos a interconectar, dependiendo de las capacidades de la CPU, pero, se supone que agregar nodos genera una complejidad operativa respecto a la programación estandarizada que se pretende en nuestro sistema de emergencias. El número óptimo de nodos posibles surgirá de la propia experimentación práctica, y no podemos definir esto en este momento.

Asimismo, cada nodo puede establecer y controlar múltiples conexiones dependiendo también de la capacidad del CPU que en cada caso se utilice, puesto que app_rtp es multithreding, y está diseñado para el manejo de múltiples conexiones.

A cada nodo se le asignó una dirección única de 8 dígitos. Cada dirección opera con una secuencia de tonos DTMF (0-9). Sin embargo no es necesario emplear los 8 dígitos, sino que se pueden definir direcciones de 2, 4, etc., siempre y cuando no existan direcciones que comiencen con el mismo dígito. Las direcciones IP de los nodos son configurables por el administrador. Sirven para las conexiones VoIP, y se pueden asegurar por claves de tipo MD5, o de tipo clave pública y privada.

Asterisk es un software que implementa una PBX completa sobre una CPU de arquitectura Intel o similar, y sistema operativo Linux. Se distribuye bajo licencia GNU, tiene una buena comunidad de desarrollo, y también implementaciones comerciales.

El proyecto liderado por Mark Spencer está disponible en www.asterisk.org y cuenta además con numerosas interfaces PCI para telefonía: FXS/FXO, TDM, E&M, E1/T1, para celulares, entre otros. Muchas son comerciales a precios muy competitivos con respecto a las de las empresas líderes de telefonía, y en general parten de un proyecto de hardware open source denominado Zaptel - Zapata Telephony[10]. Ambos proyectos se encuentran vinculados.

El bajo costo de las interfaces es posible porque gran parte del procesamiento lo hace la CPU. Asterisk, por otra parte, cuenta con los códec de audio, generación de tonos DTMF, cancelación de eco y algoritmos DSP para procesar las señales, entre otros.

8.2. Por qué Asterisk como plataforma.

Asterisk se comporta como un "Hybrid Soft Switch", como se denomina en las aplicaciones de redes convergentes, puesto que la misma plataforma opera la tecnología legacy TDM junto con la tecnología VoIP.

En Asterisk ambas tecnologías pueden interoperar en forma transparente, porque tiene la capacidad de interconectar y transcodificar en tiempo real de una tecnología a la otra.

Asterisk emplea el concepto de "canal" para representar una conexión entre dos entidades, por ejemplo dos teléfonos IP. Cada canal se describe y caracteriza por su tecnología, por ejemplo un canal VoIP, puede ser SIP (Sesion Internet Protocol), IAX2 (Inter-Asterisk eXchange, version 2), y un canal de tipo TDM se identifica como "ZAP".

Como se puede observar, el dominio de diferentes tecnologías está privilegiado por la arquitectura de Asterisk, donde la radio será una más, con las interfaces adecuadas y el control de la radio debidamente programado.

8.3. Interfaces con equipos de radio.

Hay que tener en cuenta que se debe conectar a una radio, con la cual no solo se envía y recibe audio, sino también señales de control. Por ejemplo el "apretar para hablar" (Push To Talk - PTT), para pasar a modo transmisión y retornar a recepción cuando corresponda, teniendo en cuenta que estamos en este caso en un enlace

[10] Este Proyecto está liderado por Jim Dixon. Está disponible en www.zapatatelephony.org.

de tipo "half - duplex".

Otra señal de control es el COR (Carrier Operated Relay). También se enviarán señales DTMF para diversos fines. Asterisk, además de enrutar apropiadamente las llamadas de voz, permite procesar señales de control y tonos DTMF una vez establecido un canal a través de cualquier interface.

Para esto se tienen dos posibilidades básicas. La primera consiste en emplear puertos FXS de Asterisk (placa FXS), para un canal simple analógico contra la radio, empleando una interface más de control. La otra es emplear una placa PCI especialmente diseñada para controlar hasta cuatro radios (Quad Radio PCI).

La interfaz ARIB (Amtelco Radio Interface Board), permite interconectar los puertos analógicos de Asterisk con la radio. Por software se configura la señalización de estos puertos, que puede ser "loop start" para el canal de recepción (RX) y "ground start" para el canal de transmisión (TX).

Por ejemplo, cuando desde algún canal de Asterisk se quiere activar el transmisor repetidor, coloca la tensión en el par TX. Esto es detectado por la placa ARIB que activa el PTT de la radio. La señalización ground start desde la placa FXS controlada por Asterisk satisface esta funcionalidad. Igualmente, la señal de audio pasa por los pares de TX y RX, mientras que la placa ARIB permite adaptar niveles cuando fuera necesario.

8.4. Repetidores, sitios y repetidores múltiples en un mismo sitio.

Un sitio es un nodo, cuya lógica de funcionamiento y dirección en la red está dada por el servidor Asterisk, mientras que un repetidor está asociado a una red de radio conectada al sitio.

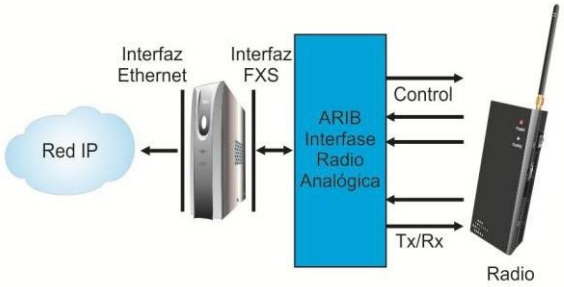


Figura 4 - Nodo múltiple

Un sitio o nodo puede tener conectados uno o varios repetidores. Con una placa ARIB, hasta cuatro repetidores. Existen otras formas de conectar repetidores, por ejemplo interfaces USB, que son prácticas y económicas cuando se trate de conectar un solo repetidor al modo.

La Figura 4 muestra un nodo múltiple con teléfonos analógicos y VoIP integrados al sitio.

8.5. Networking entre nodos.

Es posible vincular en una misma red a múltiples nodos, tanto por razones geográficas, como de balance de carga. Como se dijo, nodos y repetidores están asociados a direcciones, que se acceden a través de dígitos por tonos DTMF. Los nodos se vinculan por medio de una red IP, mediante protocolo IAX2 (Inter-Asterisk eXchange, version 2).

La siguiente figura representa las posibilidades de networking de una red con tres nodos o sitios interconectados por red IP, donde cada nodo tiene una o más redes de radio asociadas.

Los sitios se conectan empleando IAX2, que, al emplear GSM como códec de compresión de audio, ocupa canales de 35 KBPS de ancho de banda sobre la red IP. Además presenta ventajas adicionales a SIP para usar como troncal.

8.6. Redes IP estáticas o dinámicas.

El mapeo de los nodos sobre la red IP es mejor hacerlo sobre direcciones estáticas. Sin embargo, las versiones actuales de app_rpt, permiten el mapeo de nombres simbólicos sobre enlaces con direcciones IP dinámicas. Esto obviamente permite usar conexiones a Internet de tipo ADSL con IP dinámicas, que son más baratas que las de IP fija, y su calidad es aceptable para un número limitado de conexiones.

Cuando una entidad IP solicita una conexión al nodo, app_rpt sabe a partir del nombre simbólico, qué dirección debe tener la conexión, la dirección IP del nodo de origen, y el número de nodo de origen. Con esta información app_rpt puede hacer un DNS forward en la dirección simbólica y comparar las dos direcciones IP. Si las direcciones IP coinciden, será ejecutada la solicitud de conexión.

Para esta aplicación, dentro de una red gubernamental privada, se emplearán direcciones estáticas, mientras que el empleo de direcciones dinámicas están previstas para el uso en nodos conectados por la Internet pública.

Por último, estas conexiones serán seguras empleando autenticación basada en MD-5 o clave pública y privada.

8.7. Comandos DTMF.

La aplicación Asterisk app_rpt proporciona una manera flexible para definir secuencias de comandos DTMF. Estas secuencias son definidas totalmente por el usuario, con el única restricción de que todos los comandos DTMF comienzan con '*'.

La aplicación app_rpt permite definir un comando para cada nodo separado pero, sin embargo, se puede compartir un comando común entre varios nodos repetidores.

También, en un nodo se pueden definir comandos para limitar las operaciones permitidas para los nodos remotos. Las tablas de comandos DTMF se definen en rpt.conf.

8.8. Nodo base y nodo repetidos.

Un nodo base es un nodo más simple, que solo cuenta un puerto FXS (TX/RX), o sea una conexión de dos vías, mientras que un nodo repetidor cuenta con al menos una conexión de cuatro vías (dos FXS).

En un mismo sitio pueden convivir en forma autónoma nodos repetidores y nodos base. Se pueden conectar entre ellos, y los canales físicos no son los mismos. Las bases pueden conectarse a equipos de frecuencia fija o no.

8.9. Telecontrol.

La funcionalidad de telecontrol sirve para verificar el estado de los nodos remotos.

Para este fin se emplean secuencias de tonos DTMF que identifican a cada estación, junto con mensajes de estado predefinidos. Alternativamente pueden emplearse cadenas de caracteres en telegrafía Morse.

Se pueden predefinir tonos para funciones particulares que se ejecutan en el nodo. Cada nodo se asocia a un conjunto de tonos predefinidos.

8.10. Consola especial de control.

Puede emplearse una consola corriendo en cualquier PC (Con placa de sonido) para comunicarse desde ella a cualquier radio o dispositivo de la red.

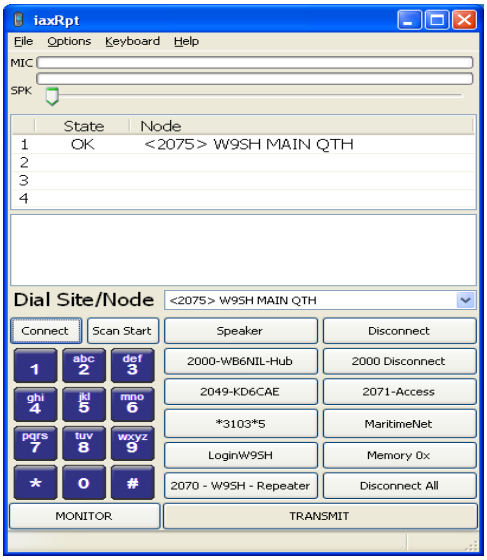


Figura 5 - Apariencia de la consola de control

Existe un proyecto (iaxRTP) de tipo open source, que presenta la consola que se muestra en la figura. Esta corre en Linux o Windows y se conecta con Asterisk por medio de IAX2 configurado adecuadamente. Cuenta con PTT y COR, permitiendo controlar varias radios desde una PC.

La Figura 5, muestra la apariencia de la consola iaxRTP.

9. Conclusiones.

Por el análisis realizado, y sin tener aún operativo el prototipo básico, en primer término podemos concluir que el proyecto es viable, por cuanto la tecnología existente permite conectarlo.

Por otro lado existen suficientes antecedentes de integraciones radio - telefónicas realizadas con la plataforma Asterisk.

Las comunicaciones orales son el medio principal para ejercer el comando y control en situaciones de emergencia o catástrofe. Los mensajes, cortos, de tipo SMS, servirán para transmitir información de sensores o elementos destinados a capturar datos de la propia fuente. Estas comunicaciones son posibles con el sistema propuesto.

El material no es crítico. Se puede emplear una computadora industrial montada dentro de una caja de transporte robustecida con rack normalizado de 19".

La Figura 6 muestra un ejemplo como el descripto.



Figura 6 - Ejemplo de un computador montado sobre una caja de transporte

Sobre la misma caja, se puede incluir un tablero de conexiones de audio para las radios, y un tablero de conexiones para líneas telefónicas.

Las líneas de telefonía celular podrán integrarse mediante dispositivos especiales o placas integradas a Asterisk.

Las radios, se integran por medio de las interfaces de audio, COR y PTT. Como los conectores de las diferentes marcas de equipos de radio son diferentes, será necesario hacer un relevamiento previo para contar con los conectores apropiados para cada tipo diferente.

Esto es lo único que no puede estandarizarse del todo, pero los equipos comerciales tampoco pueden hacerlos, y

los conectores deben preparase ad-hoc para cada equipo particular.

El despliegue del sistema será rápido, por cuanto cada nodo puede estar pre configurado previamente, y solo bastará agregar los usuarios.

Una breve instrucción de uso será necesaria previo a su empleo.

Por último, el sistema es flexible y escalable, puesto que se basa en nodos interconectados, cada uno de los cuales puede tener una capacidad variable según el hardware que se disponga. El software es siempre flexible y se adaptará a cada caso particular.

10. Agradecimientos.

Destacamos el apoyo brindado por el Laboratorio de Informática de la Escuela Superior Técnica "Gral. de División Manuel Nicolás Savio", Facultad de Ingeniería del Ejército, y por parte del Centro de Investigación y Desarrollo de Software (CIDESO), laboratorio asociado a la mencionada Facultad.

11. Referencias.

[1] Asterisk. The Future of Telephony. Jim Van Meggelen, Leif Madsen, and Jared Smith - Second Edition. Ed O' Reilly Network.

[2] www.dmkeng.com; www.asterisk.org; www.voip.info.org; www.qsl.net; www.allstarlink.org; www.zapatatelephony.org.

Evaluación de la Interferencia para Esquemas de Radio Cognitiva mediante Plataformas de Radio Definida por Software

Enrique Mariano Lizárraga*†, Gastón Peretti‡, Víctor Hugo Sauchelli*, Walter Edgardo Herrera†
*Facultad de Ciencias Exactas, Físicas y Naturales. Universidad Nacional de Córdoba.
†Facultad de Tecnología y Ciencias Aplicadas. Universidad Nacional de Catamarca.
‡Facultad Regional San Francisco. Universidad Tecnológica Nacional.
{*lizarem@efn.uncor.edu, gaston.peretti@frsfco.utn.edu.ar,*
vsauch@com.uncor.edu, wherrera@tecno.unca.edu.ar}

***Abstract*—En este trabajo se analiza un esquema de adquisición de señales para comunicaciones inalámbricas, el cual se utiliza para evaluar la interferencia observada en un receptor ante la eventual introducción de nuevas señales en el espectro electromagnético. El enfoque tomado consiste en modelar un sistema global que responda al concepto de *radio definida por software*, (SDR, *software-defined radio*). Este esquema es puesto en práctica por medio de un receptor comercial de la norma DVB-T (*Digital Video Broadcasting - Terrestrial*) el cual se maneja mediante un software controlador modificado. Por medio de un apropiado procedimiento de control, es posible manejar el oscilador local interno del dispositivo junto con otros bloques elementales, y finalmente conseguir un conversor de frecuencia sintonizable, apto para implementar la cadena de radio frecuencia (RF) de un receptor típico. Una característica explotada en este esquema es que el sistema conseguido muestra inherentemente el desempeño real que se encuentra en dispositivos de comunicaciones comerciales. Opuestamente, instrumentos de laboratorio frecuentemente presentan etapas de procesamiento que, aunque producen señales de salida con mayor calidad, no son representativas del desempeño observable en dispositivos de la electrónica de consumo. Las muestras de tiempo discreto de la señal recibida se adquieren finalmente en un software especialmente diseñado. Luego, un procesamiento adecuado se implementa para evaluar la interferencia experimentada en una transmisión de *frecuencia modulada* (FM), tomada como un caso de ejemplificación. Se obtienen algunos parámetros que resultan útiles para la evaluación de la interferencia y se propone que estos podrían ser aplicados al estudio de ambientes con *radio cognitiva* (CR, *cognitive radio*).**

I. Introducción

La asignación estática de porciones del espectro electromagnético para el uso de sistemas de comunicaciones ha resultado en un uso ineficiente de este recurso [1]. Numerosos estudios de campo han determinado la importancia de la afirmación anterior. A este hecho se suma la observación de que ciertas bandas de frecuencias, que en muchos casos se encuentran en desuso real, presentan propiedades que son altamente beneficiosas para sistemas de comunicaciones inalámbricos. Por ejemplo, se pueden citar las atractivas características de penetración de muros que se advierten en ciertas bandas de UHF. Esta cualidad resulta en un valioso beneficio para servir áreas urbanas con servicios inalámbricos para comunicación de datos. Resulta, sin embargo, que estas bandas están en total desuso en vastas áreas geográficas, mientras que en otras regiones al menos existen franjas horarias o breves periodos para las cuales el espectro permanece ocioso. Este concepto se encuentra intensamente estudiado en este momento, tanto por la comunidad académica como por sectores de la industria, con el propósito de alcanzar adecuadas regulaciones de *acceso dinámico al espectro* (DSA, *dynamic spectrum access*). Está claro que este enfoque provée desafíos significativos y así el principio de *radio cognitiva* (CR, *cognitive radio*) [2] ha ganado gran atención [3],[1].

En el esquema CR se asume la coexistencia de usuarios *primarios*, representados por los usuarios tradicionales, y usuarios *secundarios* [1]. Para los últimos se establece que contarán con una etapa de procesamiento capaz de detectar la ausencia de señales primarias en una banda determinada, con lo cual estarán habilitados para utilizar el espectro de acuerdo con políticas que aún están siendo estudiadas. Algunos temas abiertos abarcan la administración de la potencia y la cooperación entre ambos tipos de dispositivos. Entonces la cuestión del *sensado del canal* ha atraído gran atención dado que ha demostrado una estrecha relación con el desempeño global que presenta el sistema. Técnicas de *sensado compresivo* (o *muestreo compresivo*) han sido propuestas para abordar este temática [4].

En este trabajo, nos concentramos en un problema relacionado con el sensado del canal, aunque independiente en esencia. Entonces este análisis se enfoca en la proposición de un sistema práctico que permite obtener métricas apropiadas acerca del efecto de la inteferencia sobre ciertas señales de comunicaciones que eventualmente serán agrupadas dentro de la categoría de señales primarias.

La motivación para el abordaje presentado surge de considerar las características no ideales de las transmisiones que se preven. Por ejemplo podemos referirnos a la existencia de señales limitadas en el tiempo, que inherentemente generan emisión de potencia fuera de la banda espectral deseada. En particular, se resalta que las transmisiones OFDM (*orthogonal frequency-division multiplexing*) [5] hacen uso de

señales limitadas en tiempo. Asimismo, OFDM es una técnica ampliamente utilizada en numerosos estándares que se espera que implementen el concepto CR, por ejemplo LTE-A (*Long Term Evolution - Advanced*), WiMAX (IEEE 802.16), etc. Esto implica que aunque una banda determinada se detecte como ociosa, el uso de esta banda por un transmisor secundario puede generar alguna interferencia apreciable en un canal secundario adyacente que sí se encuentre activo. Entonces para maximizar el uso de la banda ociosa, se necesita cuantificar la máxima interferencia permitida. Por esto se considera valioso desplegar un esquema de medición apropiado para establecer este análisis. En nuestro trabajo se utiliza un sistema que incluye un conversor de frecuencia programable para obtener muestras en banda base (en la literatura es común referirse a un *down-converter*). Este dispositivo puede ser sintonizado sobre un amplio rango de frecuencias que abarca desde VHF hasta la zona espectral de microondas. La implementación responde a ensayos divulgados acerca de la modificación de dispositivos receptores de televisión digital. En particular nos referimos a ciertos receptores del estándar DVB-T (*Digital Video Broadcasting - Terrestrial*) con conectividad USB (*Universal Serial Bus*). De este modo es posible aprovechar ciertos componentes físicamente disponibles dentro del dispositivo original. Un aspecto que se desea resaltar de este enfoque es que las señales adquiridas son representativas de las que se adquieren mediante dispositivos electrónicos de producción masiva. Contrariamente, la circuitería disponible en instrumentos de medición para laboratorio es debidamente optimizada pero esto deriva en niveles de calidad para la adquisición que no están presentes en la denominada electrónica de consumo, es decir, en la variedad de dispositivos electrónicos que alcanzan una distribución masiva. A continuación, se utiliza un software controlador modificado, el cual depende del sistema operativo utilizado en el computador de gestión (*host*). Mediante este procedimiento, se obtienen las señales complejas de banda base (con sus componentes en fase y cuadratura, frecuentemente denominadas *señales I-Q*) que provienen de la señal presente en la porción espectral de radio frecuencia (RF) analizada. Esto es correspondiente con el concepto de que para lograr una transmisión electromagnética de información, la banda de paso de la señal se encuentra en RF para un sistema de comunicaciones típico.

En nuestro análisis, el sistema de adquisición se utiliza para capturar señales analógicas de *frecuencia modulada* (FM) provenientes de estaciones de radiodifusión comercial. La importancia de este tipo de señales se funda en el amplio despliegue de este tipo de estaciones alrededor del mundo. Si entonces se analizan dispositivos (o redes) con CR, se espera que estas coexistan con las transmisiones FM tradicionales. Notar también que la transición desde la radiodifusión analógica hacia la de tipo digital se espera que ocurra en el futuro. Entonces, este estudio presenta resultados que pueden ser extendidos para la evaluación de otros métodos de modulación.

La secuencia de secciones en este trabajo es como sigue. En la Sección II, el sistema implementado se describe. La Sección III detalla algunas características específicas de las señales adquiridas. En la Sección IV se presenta un criterio de cuantificación de la interferencia de acuerdo con los resultados obtenidos en campañas de medida. Una breve discusión se presenta en la Sección V, mientras que las conclusiones se detallan en la Sección VI.

II. Implementación del Sistema de Adquisición

Una breve descripción acerca del uso del espectro electromagnético nos lleva a decir que las transmisiones de FM comercial se encuentran ubicadas entre los 88 y los 108 MHz. Además, los canales de televisión comerical se encuentran en las siguientes bandas VHF: 54 a 72 MHz, y 174 a 216 MHz. También se encuentran canales de televisión en bandas UHF: 470 a 608 MHz, y 614 a 746 MHz. En nuestro país, las transmisiones de televisión digital se implementan siguiendo el estándar ISDB-T (*Integrated Services Digital Broadcasting - Terrestrial*) sobre las bandas de frecuencia UHF que fueron mencionadas[1]. Este resumen nos permite inferir ciertas bandas espectrales donde los esquemas CR podrían tomar lugar.

Asimismo, otra situación que es tenida en cuenta en este trabajo es la existencia de receptores de televisión digital disponibles bajo el formato de pequeños dispositivos periféricos diseñados para computadores con conectividad USB, según se mencionó antes. De acuerdo a la variedad comercial de estos dispositivos, se encuentra que frecuentemente soportan múltiples estándares para permitir múltiples funcionalidades. Por ejemplo, algunos receptores son capaces de recibir señales FM, DAB (*Digital Audio Broadcasting*), DVB-T. A partir de estas características, se puede predecir el amplio rango de sintonización que es permitido en estos dispositivos. Para el caso del dispositivo particular utilizado en nuestra campaña de medición, la banda de operación fue establecida entre 40 y 888 MHz.

Resumiendo, un receptor de televisión digital se utiliza conjuntamente con un software controlador especialmente modificado. De esta manera, es posible gobernar el funcionamiento del oscilador local del dispositivo para utilizar la circuitería disponible en este receptor como un conversor de frecuencia ajustable. Por medio de un control adecuado, las señales I-Q de tiempo discreto se obtienen en tiempo real. En nuestro caso, el establecimiento de este modo de operación deshabilita la operación de demodulación que fue originalmente introducida en el disposiivo (por ejemplo FM, DAB, etc.). Como consecuencia, el comportamiento logrado es el de un conversor de frecuencia totalmente transparente que además cuenta con una salida digital.

III. Descripción del Escenario de Medición

Los resultados principales para este trabajo fueron obtenidos en la ciudad de Córdoba, Argentina. El sistema de adquisición fue configurado con un oscilador local de 105.5 MHz y un ancho de banda de 2,8 MHz. En el área geográfica de análisis

[1]El estándar ISDB-T es funcionalmente equivalente al estándar DVB-T referenciado antes. No obstante, no se utiliza en Argentina aunque ha sido desplegado en gran parte de Europa.

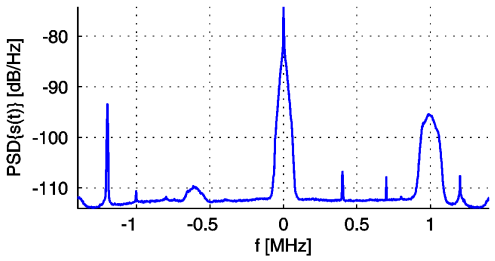


Fig. 1. Estimación de la PSD de la señal adquirida mediante el método de los periodigramas promediados de Welch.

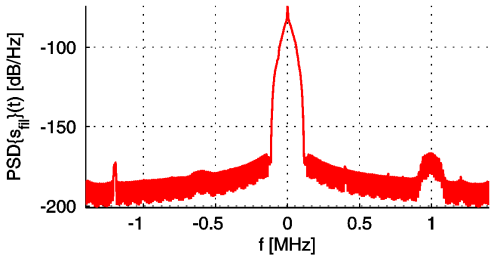


Fig. 2. Estimación de la PSD de la señal filtrada mediante el método de los periodigramas promediados de Welch.

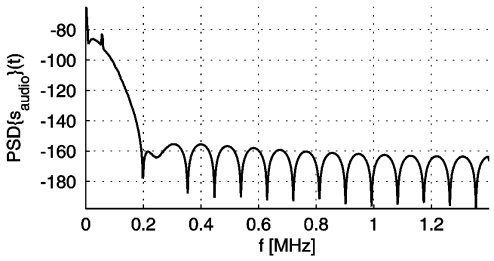


Fig. 3. Estimación de la PSD de la señal de audio mediante el método de los periodigramas promediados de Welch.

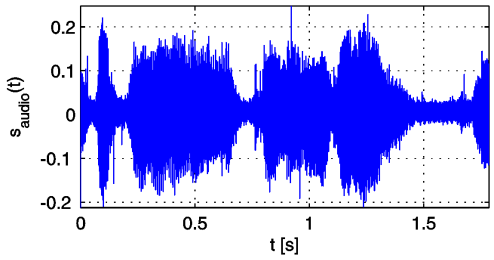


Fig. 4. Muestras de tiempo discreto de la señal de audio obtenida de la etapa de demodulación FM.

se encontró una potente señal de FM que puede ser adecuadamente demodulada mediante receptores convencionales, tales como los portables, los de automóvil, etc. Uno de los registros de adquisición obtenidos mediante la operación del sistema de adquisición se presenta gráficamente en la Fig 1. En ella se muestra la *densidad espectral de potencia* (PSD, *power spectral density*) estimada utilizando el método de los periodigramas promediados de Welch [6]. Se utiliza una ventana de 2^{11} muestras de tipo Hanning, y se especifica 8% de solapamiento. Notar que esta parametrización para la estimación de la PSD se mantiene a lo largo de este trabajo. Luego, la señal recibida se define como $s(t)$.

Se debe notar que en la Fig. 1 se muestra que existen otras transmisiones en curso ocupando canales adyacentes al de la transmisión FM. Sin embargo, la señal más cercana con potencia apreciable se encuentra 400 KHz por encima. Esta banda podría ser potencialmente de interes para un equipamiento secundario.

En una segunda etapa de este estudio, la meta es obtener un receptor basado en software para emular la operación de un dispositivo receptor FM de los de tipo comercial. Entonces la demodulación se lleva a cabo utilizando un lenguaje de programación de alto nivel y software de programación convencional. El procesamiendo digital de señal que es aplicado incluye, en una primera instancia, filtrado de banda, el cual se implementa por medio de un filtro pasabajos de orden 200 con una frecuencia de corte de 90 KHz. La PSD a la salida de este filtro se presenta en la Fig. 2. Un procesamiento subsiguiente consiste en la conversión de frecuencia a amplitud. Esta operación es principalmente comprendida por un discriminador polar y ciertas operaciones dedicadas a limitar la amplitud de la señal obtenida. Finalmente, se aplica un nuevo proceso de filtrado por medio de un esquema FIR (*finite impulse-response*) de orden 30 con frecuencia de corte establecida en 15 KHz. De este modo es posible identificar que una señal de audio $s_{\text{audio}}(t)$ se obtiene de la señal original $s(t)$, por medio de la emulación de un receptor de FM convencional. Este procesamiento se simboliza con

$$s(t) \xrightarrow{\text{demod. FM}} s_{\text{audio}}(t). \tag{1}$$

Luego, la Fig. 3 muestra la PSD estimada para la señal de audio. A su vez, la Fig. 4 muestra la forma de onda obtenida en el dominio del tiempo. Esta última representación se realiza con la intención de verificar la efectividad del esquema de procesamiento implementado. En efecto, se observa que la forma de onda encontrada es consistente con aquellas que típicamente se observan para señales de voz. Adicionalmente, el audio de esta señal es reproducido para confirmar su integridad y validez.

Hasta aquí se consigue una puesta en funcionamiento de un receptor basado en el concepto de SDR. A continuación se propone un uso de este esquema práctico para analizar la interferencia en ciertas situaciones de interés.

IV. Cuantificación de los Efectos de la Interferencia

El método seleccionado para la evaluación de la interferencia en una señal analógica, que eventualmente será tomada como una señal primaria, está basado en la siguiente observación. Dado que se intenta evaluar la interferencia en un ambiente realista, la señal original que se utiliza proviene de una estación de radiodifusión comercial. En este contexto,

no se dispone de la señal de excitación del transmisor. Esto es, no se conoce la señal moduladora (o *mensaje*). En otras palabras, en este escenario no se puede obtener una copia de la señal original que la estación de radiodifusión intenta transportar hacia los receptores primarios. Sin embargo, dadas las condiciones del canal que se observan al momento de poner en práctica la campaña de medida, se asume que se cuenta con una adecuada recepción de la señal RF. Esta afirmación es, en parte, fundamentada en el hecho de que ambos equipos, es decir, el transmisor y el receptor, se encuentran inmóviles, y fuentes significativas de inteferencia no parecen estar presentes (al menos en la banda espectral indicada), de acuerdo con la PSD observada en la Fig. 1. Se considera que en el caso de no encontrar esta situación al momento de poner en funcionamiento este esquema de medida, una alternativa válida sería modificar las condiciones de recepción para mejorar las propiedades de recepción. Una opción razonable es, sencillamente, cambiar la ubicación del sistema de medición para incrementar la potencia de la señal FM recibida.

Luego, se propone que manteniendo los parámetros de demodulación FM tal como fue detallado en la Sección III, es posible comparar la señal demodulada lograda antes, es decir $s_{\text{audio}}(t)$, con respecto a la que se obtiene en un posible escenario de interferencia que será especificado a continuación.

Mientras tanto, se resalta que la métrica escogida para cuantificar la similaridad entre las dos señales analizadas es el *error cuadrático medio* (MSE, *mean square error*).

Entonces resulta que el foco de este análisis está dado sobre el aprovechamiento de la naturaleza SDR en el esquema implementado, para incorporar la componente de interferencia de una manera numérica.

A. Tono Único

El primer caso considerado en este estudio evalúa el desempeño del receptor FM ante la presencia de una señal agregada de banda angosta. Este efecto puede ser causado por la presencia de un tono interferente, frecuentemente mencionado en la literatura como *jamming*. Para establecer esta situación, se utiliza un tono sinusoidal dado por la expresión

$$s_{\text{int}}(t) = A.cos(2\pi f_{c,\text{int}} t). \tag{2}$$

Esta señal se agrega a la señal proveniente del receptor físico mientras que distintos valores pueden ser especificados para la frecuencia interferente $f_{c,int}$. Como consecuencia de este procedimiento, la señal efectivamente procesada por el demodulador (FM) en software consiste en la suma de la señal obtenida utilizando el sistema de adquisición $s(t)$, más aquella que es numéricamente generada para modelar la señal de interferencia, i.e.,

$$r(t) = s(t) + s_{\text{int}}(t). \tag{3}$$

Entonces el procesamiento establecido genera la señal $s_{\text{audio}}^{(\text{tone})}(t)$. Esta transformación no lineal se indica mediante

$$r(t) \xrightarrow{\text{demod. FM}} s_{\text{audio}}^{(\text{tone})}(t). \tag{4}$$

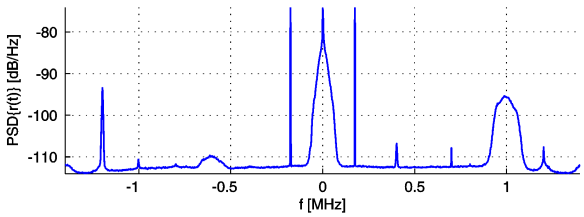


Fig. 5. Estimación de la PSD de la señal interferida mediante el método de los periodigramas promediados de Welch. La señal de interferencia se representa mediante un tono con frecuencia portadora $f_{c,int} = 175$ KHz.

Tomando en cuenta los parámetros $f_{c,int} = 175$ KHz, $A = 10^{(-35/20)}$, se obtiene la PSD estimada que se visualiza en la Fig 5. En este caso se encuentra que la métrica de desempeño alcanza el valor

$$\text{MSE}\{s_{\text{audio}}(t) - s_{\text{audio}}^{(\text{tone})}\} = 2.26\,.\,10^{-9}. \tag{5}$$

B. OFDM

El segundo caso que se analiza en este trabajo toma en cuenta una señal OFDM [7], [5]. Esta selección se fundamenta en la amplia inclusión de este esquema de modulación y multiplexación, en sistemas que en el futuro podrían implementar capacidades CR.

Para la parametrización de la señal OFDM que es incorporada a la señal original, se considera especialmente la señal ubicada 400 KHz encima de la frecuencia central. Entonces siguiendo el objetivo de insertar una señal (secundaria) que actúe de la forma más transparente posible para el sistema global, el ancho de banda de la señal OFDM se debe especificar correctamente. Con este criterio, se fija un ancho de banda de 100 KHz para la señal OFDM. Para lograr consistencia con esta definición, se establece que la frecuencia portadora del sistema secundario se ubica 225 KHz por arriba de la frecuencia central de adquisición.

Este criterio se explica como sigue. Para el escenario con interferencia que se establece en la Sección IV-A, la componente de interferencia más cercana afectando la recepción FM se halla 175 kHz por encima de la frecuencia central de este receptor. Dado que en la especificación de la señal OFDM se desea alcanzar la misma separación espectral entre la señal FM y la componente de interferencia más cercana de este nuevo caso, la frecuencia central para la transmisión OFDM toma el valor indicado arriba. Mientras tanto, se indica que el sistema OFDM que es emulado emplea 2048 subportadoras, un intervalo de guarda (fraccional) de $1/4$ y modulación 16-QAM. La señal OFDM numéricamente generada reemplaza $s_{\text{int}}(t)$ en (3). Consecuentemente, la transformación establecida es la siguiente,

$$r(t) \xrightarrow{\text{demod. FM}} s_{\text{audio}}^{(\text{OFDM})}(t). \tag{6}$$

Con este procesamiento es posible observar la siguiente métrica de interferencia

$$\text{MSE}\{s_{\text{audio}}(t) - s_{\text{audio}}^{(\text{OFDM})}\} = 2.66\,.\,10^{-6}. \tag{7}$$

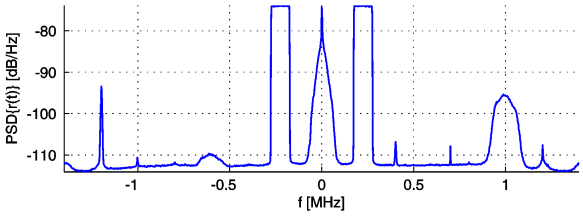


Fig. 6. Estimación de la PSD de la señal interferida mediante el método de los periodigramas promediados de Welch. La señal de interferencia se representa mediante una señal OFDM con frecuencia central $f_{c,int} = 225$ KHz y ancho de banda de 100 KHz.

Por su parte, la PSD de la señal procesada por el receptor FM definido en software es la que se presenta en la Fig. 6.

C. Señales OFDM con Precodificación Espectral

Se encuentra en la literatura que por su característica natural de estar formada por señales limitadas en el tiempo, las señales OFDM[2] son responsables de la ocurrencia de fugas espectrales de potencia. Consecuentemente, se observan emisiones espectrales fuera de la banda de frecuencias idealmente asignada para la señal OFDM. Para mitigar este defecto, el filtrado tradicional es una alternativa válida [8]. Sin embargo, existen perspectivas que consideran las ventajas de los sistemas con precodificación espectral (SP, *spectral precoding*), dado que de esta manera es posible aplicar un ajuste dinámico para seguir las variaciones de canales variantes en el tiempo (tales como los esperados para entornos CR). La introducción de la técnica SP reduce distorsión fuera de la banda deseada, en comparación con la del sistema OFDM crudo planteado en la Sección IV-B. Entonces se espera observar una menor distorsión de la señal FM, en la presencia de un sistema SP-OFDM. La señal que se inserta en este caso es determinada mediante la misma parametrización OFDM que fue descrita antes. Además un precodificador definido como en [9] se implementa con orden[3] 2. El sistema analizado se simboliza mediante la expresión

$$r(t) \xrightarrow{\text{demod. FM}} s_{\text{audio}}^{(\text{SP-OFDM})}(t), \tag{8}$$

donde la señal SP-OFDM es utilizada en lugar de $s_{\text{int}}(t)$ para obtener $r(t)$ como en (3). La estimación de la PSD del sistema analizado se omite en esta sección debido a su natural similitud con aquella presentada en el caso de análisis anterior (ver Fig. 6). Estos ajustes producen la siguiente métrica de interferencia,

$$\text{MSE}\{s_{\text{audio}}(t) - s_{\text{audio}}^{(\text{SP-OFDM})}\} = 1.14\,.\,10^{-9}. \tag{9}$$

Notar que este resultado confirma la desventaja que se prevé para una señal OFDM convencional, al tiempo que verifica la efectividad de la técnica de conformación espectral tomada de [9].

[2]Formadas por la concatenación de símbolos OFDM de duración finita.
[3]Orden $N = 2$.

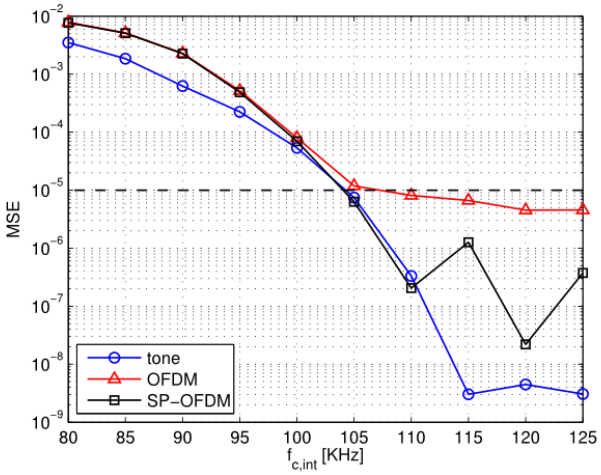


Fig. 7. Efecto de la separación espectral con respecto a la métrica de interferencia analizada.

D. Análisis de la Separación Espectral de la Fuente de Interferencia

Los escenarios de análisis propuestos previamente permiten por sí mismos la declaración de algunas importantes conclusiones. Sin embargo, se debe notar que la separación establecida entra la señal FM y la que se interpreta como interferencia para la primera, fue mantenida invariante en todos estos casos. Más allá de la búsqueda de condiciones justas para la comparación de desempeño en las Secciones IV-A, IV-B y IV-C; es interesante plantear un desplazamiento en frecuencia de las fuentes de interferencia, evaluando el comportamiento global ante esta condición. Esta estrategia se practica evaluando la métrica de interferencia con respecto a diferentes calibraciones para la componente de frecuencia más cercana que proviene de cada fuente de interferencia. El análisis se realiza para cada una de las tres señales interferentes, o secundarias, que se describieron antes. Los resultados se presentan en la Fig. 7. Para evaluar el comportamiento de estos escenarios, una línea entrecortada indica el umbral 10^{-5} fijado empíricamente para la métrica MSE. Se puede ver entonces que ambas señales, la del tono único y la del sistema SP-OFDM se desempeñan de una forma equivalente. Sin embargo, las emisiones de potencia fuera de banda en una tranmisión OFDM cruda fuerzan una separación adicional de 5 KHz entre la señal FM y la señal interferente (o secundaria). De esta forma se puede suponer un deterioro de la eficiencia espectral del 5%, lo cual en muchos casos es significativo. Por su parte, el piso observado para la métrica MSE se empobrece en varios órdenes de magnitud para este caso, cuando se compara con sus alternativas. Mediante este análisis se verifica la importancia del efecto de las emisiones espectrales no deseadas, que eventualmente produciría la introducción de comunicaciones secundarias en un eventual despliegue CR.

V. Discusión

Algunas palabras se pueden decir acerca de esta implementación para evaluación, con la intención de destacar ciertas características. Por un lado, se debe tener en cuenta el ancho de banda utilizado en este esquema. En sistemas de transmisión digital moderna es sumamente común la consideración de anchos de banda cercanos a los 5 MHz y superiores a este valor. Estas especificaciones son por tanto esperables para los futuros sistemas CR. Luego, se observa que el ancho de banda analizado en este trabajo podría resultar insuficiente. Entonces se propone que un sistema de adquisición paralelizado, basado en el uso de dos o más receptores, representa un trabajo más allá del presentado aquí, el cual sería capaz de superar esta limitación. Notar que el nivel de paralelismo necesario, en principio estaría determinado por la relación observada entre el ancho de banda requerido y el que está disponible en el receptor SDR actual (ver Sección III).

Por otro lado, es importante resaltar que el umbral especificado para la métrica MSE, que fue mencionado en la Sección IV-C, fue determinado de una forma arbitraria. Esta métrica consiste en valores que son difíciles de analizar objetivamente debido a la naturaleza analógica de las señales involucradas. En otras palabras, es difícil establecer si un cierto valor de MSE respresenta uno aceptable o no. Sin embargo, se resalta que esta métrica puede ser satisfactoriamente utilizada para establecer comparaciones entre diferentes sistemas, tal como se hace en este trabajo.

VI. Conclusiones

Este trabajo describe la implementación de un sistema de medida dedicado al análisis de la interacción entre señales empleadas por estaciones de difusión comercial y otras provenientes de fuentes que son insertadas intencionalmente en el espacio de la señal original. El concepto escogido es el de implementar un receptor con *radio definida por software* (SDR, *software-defined radio*). Por medio de este esquema, se muestra la capacidad de utilizar este marco de análisis para considerar la eventual existencia de señales secundarias bajo el paradigma de comunicaciones con *radio cognitiva* (CR, *cognitive radio*). Pruebas específicas se llevaron a cabo considerando transmisiones de FM comercial, tonos interferentes (*jamming*), y comunicaciones OFDM. Estas mostraron la importancia del control de la potencia emitida fuera de banda, y la efectividad de la precodificación espectral para señales OFDM. Estas conclusiones se elaboraron a partir de una métrica de cuantificación de la interferencia que se propone en este artículo, que serviría para prever el futuro comportamiento de los sistemas actuales ante la aparición de señales de un eventual despliegue CR.

References

[1] M. Nekovee, "Cognitive Radio Access to TV White Spaces: Spectrum Opportunities, Commercial Applications and Remaining Technology Challenges," in *IEEE Symposium on New Frontiers in Dynamic Spectrum*, Apr. 2010, pp. 1–10.

[2] P. Pawelczak, K. Nolan, L. Doyle, S. W. Oh, and D. Cabric, "Cognitive Radio: Ten Years of Experimentation and Development," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 90–100, Mar. 2011.

[3] R. Andrew, H. Horikoshi, and A. Terzoli, "Keynotes Summaries," in *Kaleidoscope 2011: The Fully Networked Human? - Innovations for Future Networks and Services (K-2011), Proceedings of ITU*, Dec. 2011, pp. 1–4.

[4] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*.

[5] R. van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. London, UK: Artech House, 2000.

[6] J. Proakis and D. Manolakis, *Digital Signal Processing, Third Edition*. Pearson Prentice Hall, 2007.

[7] J. Bingham, "Multicarrier Modulation for Data Transmission: An Idea whose Time has Come," *"IEEE Commun. Mag."*, vol. 28, no. 5, pp. 5 –14, may 1990.

[8] M. Faulkner, "The Effect of Filtering on the Performance of OFDM Systems," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1877 –1884, Sep. 2000.

[9] J. van de Beek and F. Berggren, "EVM-Constrained OFDM Precoding for Reduction of Out-of-Band Emission," in *IEEE Veh. Tech. Conf. Fall*, Sep. 2009, pp. 1–5.

Implementación de un servicio Peer to Peer Nativo en IPv6

Isabel Marko, Víctor Fernández, Federico Valles, Pablo Vera, Rocío Rodríguez,
Daniel Giulianelli, Graciela Cruzado
GIDFIS – Grupo de Investigación, Desarrollo y Formación en Innovación de Software
Departamento de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de La Matanza
Florencio Varela 1903, San Justo, Buenos Aires, Argentina
gidfis@ing.unlam.edu.ar

Resumen

Las redes Peer to Peer (P2P) son el basamento para algunas de las aplicaciones más populares en Internet, como el acceso a archivos en forma distribuida, los streamings de video, el acceso a redes de sensores, actuadores, de forma distribuida, confiable y muchas veces en tiempo real. Sin embargo el protocolo IPv4 presenta grandes inconvenientes en la construcción y funcionamiento de las redes P2P. Inconvenientes que el IPv6 permite resolver. En este artículo se analiza la utilización del protocolo de red IPv6 a nivel aplicación, viendo como las aplicaciones nativas en el desarrollo de redes de sensores y actuadores, pueden verse beneficiadas en la implementación de redes P2P.

Palabras Clave: IPv6, Peer to Peer, PNRP, Sensores

1. Introducción

Para explicar la topología P2P, se comenzará mencionando la topología Cliente-Servidor que es más clásica y actualmente también la más utilizada.

Las redes cliente servidor, son las predominantes en el caso de Internet, donde hay un conjunto muy minoritario de servidores, que tienen los contenidos que serán requeridos por uno o más clientes. En este tipo de redes el cliente envía una petición a un servidor, que luego devuelve la información requerida, por ejemplo un archivo al cliente [1].La arquitectura funciona muy bien salvo por el tema de la escalabilidad, ya que al aumentar la cantidad de clientes los servidores requieren ser más complejos. Por otra parte, cuando los requisitos de descarga de información de los clientes es grande, inevitablemente los servidores colapsaran en algún momento.

Las redes P2P están compuestas por esquemas donde cada nodo es equivalente a los otros, o sea todos los nodos pueden funcionar como clientes y como servidores [2], la información está distribuida y replicada en una gran cantidad de nodos (ver Figura 1). [3]

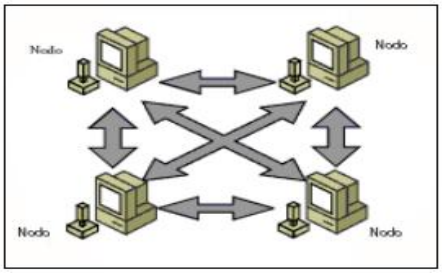


Figura 1 – Red P2P

Redes de este tipo son las que utilizan por ejemplo la tecnología BitTorrent [4]. En teoría cada nodo puede ser un cliente y un servidor dentro de la red. La escalabilidad de sistema es potencialmente muy flexible. P2P requiere de ciertas características que el protocolo IPv4 en su forma nativa estaban levemente o para nada soportadas, y cuya implementación posterior era complicada debido fundamentalmente a la limitada capacidad de direccionamiento relativa del IPv4, que limita la unicidad de direcciones IP en todo el dominio de Internet.

Por ejemplo en una red P2P, cada nodo debe ser capaz de: Descubrir otros clientes; Conectarse con otros clientes; Comunicarse con otros clientes

IPv6 surge para solucionar el problema de agotamiento de direcciones IPv4. La entidad encargada de asignar los bloques de direcciones IP es IANNA (Internet Assigned Numbers Authority), la asignación la realiza a cinco RIR (Regional Internet Registry): AFRINIC (African Network Information Center); APNIC (Asia-Pacific Network Information Center); ARIN (American Registry for Internet Numbers); LACNIC (Latin American and Caribbean Internet; Addresses Registry); RIPE (Regional Internet Registry).

Sucede que los bloques de IPv4 se han agotado y IANNA ya no cuenta con direcciones de IPv4 para repartir, de hecho la mayor parte de los RIR ya no tienen en su haber direcciones para distribuir a los nuevos usuarios de internet. Este agotamiento estaba previsto y en función de ello nace IPv6. “El protocolo IPv4, dispone solo de 32 bits de direcciones proporcionando un espacio teórico de 2^{32} (aproximadamente cuatro mil millones) interfaces de red únicas globalmente direccionables.

IPv6 en cambio tiene direcciones de 128 bits y por tanto puede direccionar 2^{128} interfaces de red (340.282.366.920.938.463.463.374.607.431.768.211.456)[5]". "Se estima que si se repartiesen en toda la superficie de la Tierra habría 6,67x1023 de direcciones IP por m²" [6].

La cantidad de direcciones disponibles en IPv6 permite pensar que no sólo podrán tener direcciones IP asignadas las computadoras y los dispositivos (teléfonos celulares, tablet) sino también a un extenso conjunto de objetos con los que se puede contar los cuales podrán conectarse en forma autónoma a la red, tal como se mencionó anteriormente. Debido a esto diversos autores la han denominado "La internet de las cosas" (IoT Internet of Things) [7] [8] [9] y también fue creado un consorcio con esa denominación (Internet of Things Consortium [10]). "La era de la información ahora toma otras dimensiones al guardar bases de datos, no solo de personas y de finanzas, sino de todo tipo de cosas animadas, información tal como de los signos vitales de personas, mascotas, ganado, fauna, ecosistemas; o también datos de objetos inanimados tales como sillas, mesas, camas, lámparas, electrodomésticos, automóviles, masetas, electrónica de entretenimiento y juegos interactivos. Información de parajes urbanos como tráfico en tramos de calle y carretera, iluminación emergente y de emergencia, detección de fallas en puentes y estructuras civiles, detección de derrumbes e inundaciones en vías de transporte, vigilancia automatizada y alertamiento para mantenimiento preventivo" [11].

Surgen infinidad de aplicaciones desde el monitoreo

de la ubicación del ganado y la salud de los animales; hasta objetos de la casa que transmiten datos para facilitar las tareas diarias. La idea es que diversos objetos puedan mejorarse para la comodidad del usuario y que incluso puedan ser monitoreados en forma remota por ejemplo desde su teléfono celular. Es decir se incrementa la idea que ya se comenzaba a aplicar en IPv4 de poder dotar a determinados objetos de la posibilidad de registrar datos y enviar información por la web. Acentuada con las direcciones públicas disponibles y además con las características que incorpora el protocolo que serán de importancia para un diverso número de aplicaciones.

Entre las ventajas de IPv6 se pueden destacar:

(1) Mayor espacio de direccionamiento: Sus 128 bits permiten tener una gran cantidad de direcciones IP disponibles, contra 32 bits con los que se contaba con IPv4. Se estima que la capacidad de direccionamiento de IPv6, cubrirá los próximos 20 años.

(2) Extensibilidad: IPv6 ha sido diseñado para ser extensible y ofrece soporte optimizado para nuevas opciones y agregados, permitiendo introducir mejoras en el futuro.

(3) Movilidad: IPv6 incluye mecanismos de movilidad más eficientes y robustos lo cual beneficiará no sólo a los usuarios de telefonía y dispositivos móviles, sino que también permitirá (por ejemplo) tener buenas conexiones a internet durante los vuelos de avión. Inicialmente al tener un dispositivo una dirección IP fija sin importar en que parte del planeta se encuentre, facilitara en mucho la tarea de las operaciones telefónicas.

(4) Multicast: IPv6 no emplea la técnica de difusión (broadcast) para comunicar todos los nodos en un segmento de red. Esa funcionalidad se realiza empleando multidifusión (multicast) [12]. La multidifusión IP, consiste en enviar datagramas IP a un grupo de receptores interesados, previa suscripción a un grupo multicast, empleando una única trasmisión. Puede funcionar en comunicaciones de tipos uno-varios ó varios-varios.

(5) VOIP: Dos de los problemas actuales de los servicios de Voz sobre IP (VoIP) son QoS y NAT. Las comunicaciones pueden resultar en baja calidad de voz (QoS), y presentar dificultad para atravesar firewalls (NAT). Al incorporar IPv6 una gran cantidad de direcciones, no será necesario utilizar NAT, y sus nuevas capacidades de Plug and Play, seguridad, y QoS implicarán mejores conexiones de voz. La eliminación de las NAT beneficiara de sobremanera la creación de redes P2P.

(6) Seguridad Incorporada: IPv6 incluye de forma nativa IPSEC, que permite la autenticación, encriptación y compresión del tráfico IP. Esto permite utilizar seguridad sin tener que configurar algo particular en cada aplicación [12]. IPv4 había surgido con la finalidad de simplemente establecer una comunicación entre computadoras y en una primera etapa la seguridad fue un tema secundario. La misma se fue incorporando al protocolo a través de parches de seguridad.

(7) Calidad de Servicio: IPv6 incluye en su encabezado un campo TOS (type of service) que proporciona información de parámetros de prioridad, retardo, rendimiento y fiabilidad [13]. Esto permite poder dar un tratamiento distinto a los paquetes.

(8) Autoconfiguración: En IPv4, la configuración de la dirección IP es un tema difícil de manejar, hay 2 formas de dar una dirección a un usuario. utilizando un servidor - 1) usando un DHCP que adiciona también un costo extra. 2) dar a cada usuario una única dirección IP y esperar que cada uno ingrese estas direcciones físicas en sus computadoras, lo que sería demasiado oneroso para la mayoría de los usuarios. En Ipv6, existe una auto-configuración de la dirección, que es muy sencilla. Un router envía un prefijo del link local en sus anuncios, entonces el host al recibirlos puede generar la dirección IP agregando la dirección MAC de su link-layer.

El presente artículo se encuentra organizado de la siguiente manera: En la sección 2 se detalla la

implementación de una red P2P en IPv6, en la sección 3 las aplicaciones que pueden verse beneficiadas, en la sección 4 se trata la posibilidad de gestionar a través de P2P bajo IPv6 una red de sensores y sus potenciales aplicaciones y además se expondrán campos en los cuales se aplican soluciones nativas de IPv6 utilizando sensores. Finalmente se expondrá como se ve beneficiada una red de sensores con las características particulares del protocolo y en la sección 5 se presentan las conclusiones.

2. Implementación de P2P en IPv6

Los framework de desarrollo cuentan con funciones ya implementadas para IPv6. En esta sección se listan las herramientas o tecnologías para implementar una red P2P.

2.1. Tecnologías utilizadas

- **Protocolo PNRP** (Peer Name Resolution Protocol - Protocolo de resolución de nombres de pares). Este protocolo permite resolver la dirección de red (dirección, protocolos y puertos). Las propiedades más destacadas del protocolo son: (A) No se necesita servidor para generar el nombre del peer; (B) Nombra no solo el ordenador (dirección IP), puede nombrarse una dirección y un puerto, de esta forma se puede nombrar un servicio o una aplicación. El framework ofrecido por .net tiene la posibilidad de generar el nombre del peer de manera segura e insegura.
- **Clouds:** A través de esta tecnología los nodos se pueden comunicar a través de clouds, el cual es un grupo de computadoras que se pueden comunicar entre sí. Pueden clasificarse en: (1) Global Cloud: El cloud global corresponde a direcciones en IPV6 en Internet. (2) Link Local Cloud: El cloud local corresponde a direcciones IPV6 locales por ejemplo en un a LAN.
- **PNRP Name:** El nombre de un peer es un nodo de comunicación que puede ser una computadora, un usuario o un grupo, un servicio, o lo que sea necesario resolver a través de una dirección IPV6. Los nombres pueden ser: (1) Inseguros pueden utilizarse en una red privada o en algún tipo de red que ya se encuentre protegida. (2) Seguros son protegidos con un certificado y una firma digital, el cual maneja el generador del nombre.
  Un PNRP ID está compuesto por 256 bits. Los 128 bits altos que son conocidos como (P2P ID), los cuales son un hash con la clave pública que se utiliza para generar los nombres de los nodos. Los 128 bits bajos son usados para determinar la ubicación del servicio el cual representa las diferentes instancias del mismo P2P ID en el mismo cloud.

2.2. Utilización en .NET

Se ha escogido el framework de .NET para implementar un servicio Peer to Peer para IPv6. La mayoría de las clases para manejar los Peers se encuentra en el Namespace System.Net.PeerToPeer. Se deben realizar tres pasos:

1. Registración de un Peer [15] [16] - La codificación correspondiente se muestra en la figura 2.

```
public  void RegisterLocaPeer(PeerInfo info)
{
    PeerName peerName = new PeerName(info.Classifier,PeerNameType.Unsecured);
    PeerNameRegistration peerNameRegistration = new PeerNameRegistration();
    peerNameRegistration.PeerName = peerName;

    peerNameRegistration.Port = info.PortNumber;
    peerNameRegistration.Comment = info.Comment;
    peerNameRegistration.Cloud = Cloud.AllLinkLocal; // cloud local
    peerNameRegistration.Start();
    info.PeerHostName = peerNameRegistration.PeerName.PeerHostName;
}
```

Figura 2. Registración de un Peer

2. Resolución de Dirección a través del PeerHostName (Id) [15] [16] - La codificación correspondiente se muestra en la figura 3.

```
public List<PeerInfo> ResolveByPeerHostName(string peerHostName)
{
 List<PeerInfo> foundPeers = new List<PeerInfo>();

 PeerNameResolver resolver = new PeerNameResolver();
 var resolvedName = resolver.Resolve(new PeerName(peerHostName, PeerNameType.Unsecured), Cloud.AllLinkLocal);

 foreach (var foundItem in resolvedName)
 {
     foreach (var endPointInfo in foundItem.EndPointCollection)
     {
         PeerInfo info = new
         info.PeerHostName =
         info.Classifier = fo
         info.Comment = found
         info.PortNumber = en

         if (endPointInfo.Add
             info.Ipv6Adress
         foundPeers.Add(info)
         endPointInfo.Address
     }
 }

 return foundPeers;
```

```
IsIPv4MappedToIPv6
IsIPv6LinkLocal
IsIPv6Multicast
IsIPv6SiteLocal
IsIPv6Teredo
MapToIPv4
MapToIPv6
ScopeId
ToString
```

```
ing();
```

```
string IPAddress.ToString()
Converts an Internet address to its standard notation.

Exceptions:
  System.Net.Sockets.SocketException
```

Figura 3. Resolución de Direcciones

3. Llamado a Servicio de un Peer a través de WCF (Windows Communication Foundation) usando la URL(Uniform Resource Locator) indicada[17] [18]. La codificación correspondiente se muestra en la figura 4.

```
        public void CallHost(PeerInfo peerinfo)
        {
            string Address = string.Format("net.tcp://{0}:{1}/TransferEngine",
              peerinfo.PeerHostName, peerinfo.PortNumber);
            Uri uri = new Uri(Address);
        }
    }
}
```

Figura 4. Llamado a Servicio de un Peer

3. Las Aplicaciones e IPv6

Si bien IPv6 es un protocolo de red sus nuevas características permiten beneficiar a diversos tipos de aplicaciones, entre ellas las que: Manejan grandes volúmenes de datos; Requieren de streaming de audio y video; Almacenan información en la nube; Utilizan telefonía digital (VoIP); Brindan soporte a la Telemedicina; Permiten el aprendizaje a Distancia (e-learning); Utilizan redes de Sensores.

Por lo dicho previamente IPv6 tiene diversas ventajas que pueden ser aprovechadas al momento de desarrollar aplicaciones nativas para este protocolo.

Es importante pensar en un gran número de aplicaciones que podrían realizarse, entre ellas aquellas que toman información provista por sensores, tema que será tratado en la sección siguiente.

4. Red de Sensores

"Para que exista el Internet de las cosas, no sólo se requiere hardware capaz de conectarse a la red, sino también es indispensable software que haga algo con los datos recibidos de la red de sensores y actuadores… En este sentido hay mucho que hacer en la capa de aplicación…" [11]. Hay unas 25 mil millones de "cosas" conectadas a Internet y se estima que en el 2020 serán 50 mil millones los objetos conectados. Todas esas "cosas" transmitirán información [19].

4.1. Sensor o Captador

Un sensor o captador es un dispositivo que se encarga de recibir información de magnitudes del exterior (magnitudes físicas o químicas) y transformarla en otras magnitudes, generalmente eléctricas, para poder analizar, cuantificar y manipular dicha información.

Algunos tipos de sensores son: de posición, de proximidad, de luz, de gas, de sonido, de humedad, de velocidad, de campo magnético, de presión, etc. Lugares típicos donde se pueden encontrar sensores son: fábricas, empresas de seguridad, laboratorios, etc.

Un tipo de sensor muy común en el área de robótica es el sensor de ultrasonido el cual se encarga de detectar objetos cercanos por medio de la generación y posterior recepción de ondas sonoras.

En la actualidad existen módulos sensores de ultrasonido que constan de un micrófono y un parlante que se encargan de generar y recibir las ondas de sonido. (ver Figura 5).



Figura 5. Sensor de ultrasonido HC-SR04 y sensor de luz con foto resistor GL 2258

Es posible pensar en un ambiente inteligente en donde distintos sensores puedan estar integrados en él en forma ubicua (transparente para el usuario final). Dichos sensores son nodos independientes en una red, pudiendo enviar información por Internet y que una persona pueda por ejemplo ver en forma remota desde su dispositivo móvil diversas alertas sobre el ambiente inteligente.

4.2. Actuador

Son dispositivos encargados de transformar energía en una fuerza aplicada con el fin de mover o activar un dispositivo de control. Dependiendo del tipo de energía que recibe el actuador, se lo denomina de 3 maneras distintas:

- Hidráulicos: Los que pueden clasificarse según el modo de funcionamiento: (1) Cilindro hidráulico: Cilindro de presión dinámica; Cilindro de efecto simple; Cilindro de efecto doble; Cilindro telescópico; (2) Motor hidráulico: Motor de engranaje; Motor de hélice; Pistón Axial; Motor de Veleta; Motor de leva excéntrica; Motor con eje inclinado; Motor hidráulico de oscilación.
- Eléctricos: Se clasifican en tres tipos: (1) Motores paso a paso: De imanes permanentes; Híbridos; De reluctancia variable; (2) Motores de corriente continua; (3) Motores de corriente alterna.
- Neumáticos los cuales se dividen en dos tipos: (1) Cilindros neumáticos: De Efecto Simple; De Efecto Doble; (2) Motores neumáticos: Motores de aletas rotativas; Motores de pistones axiales.

Tabla 1. Ventajas y Desventajas de los Actuadores

	Actuadores Neumáticos	*Actuadores Eléctricos*	*Actuadores Hidráulicos*
Ventajas	Económicos Rápidos Sencillos Robustos	Precisos Fiables Fácil Control Sencilla	Rápidos Alta relación potencia-peso Alta capacidad

		instalación	de carga Estabilidad frente a cargas estáticas.
Desventajas	Instalación especial (compresor, filtros) Ruidosos	Potencia limitada	Difícil Mantenimiento Instalación especial (filtros, eliminación de aire) Costosos Frecuentes fugas

4.3. Sensores e IPv6

En la actualidad es grande el espacio que se están ganando los sensores en el ámbito de redes IPv6. A continuación se mencionan cuatro campos en los cuales existen diversas aplicaciones nativas en IPv6:

- Agricultura de precisión: "…Por medio de sensores estratégicamente situados, se realizará un monitoreo de la temperatura y/o la humedad relativa, con el fin de proveer de gran cantidad de datos confiables para la investigación agronómica" [20].
- Sistema de gestión de alumbrado público: Con el objetivo de una reducción del consumo y la gestión en tiempo real del alumbrado se ha considerado "un sistema basado en nodos instalados en farolas, los cuales se comunicarán entre sí para funcionar de la forma más optimizada posible, complementándose todo ello con un "nodo base", que se encargará de servir de nexo entre la red y los diferentes elementos necesarios para la configuración de los nodos y la recogida de información. Esta información servirá para que una página web pueda mostrar al usuario final toda la información necesaria para tener un control sobre el estado actual de funcionamiento de cada una de las farolas, control del consumo, así como detección de averías" [21].
- Monitoreo de Ganado: "En el mundo de las IoT hasta las vacas estarán conectadas…". Mediante "sensores en las orejas del ganado, esto permitirá a los granjeros monitorear la salud de sus vacas y seguir sus movimientos…"[19].
- Ambientes Inteligentes: IBM entre otras empresas ha presentado proyectos para la construcción de una casa inteligente llamada "Smarter Home" el cual se basa en tecnologías inalámbricas que trabajan bajo el protocolo IPV6 [4].

Tomando como referencia el último ítem, un ambiente inteligente es aquel dentro del cual es posible interactuar de forma transparente con una multitud de dispositivos intercambiando información y/o servicios.

Dentro del concepto de Ambiente Inteligente se muestra una visión de la sociedad y de la información en donde se enfatiza la facilidad de uso, el soporte eficiente de los servicios y la posibilidad de mantener interacciones naturales con el ser humano. El objeto central se materializa a grandes rasgos en una persona rodeada de interfaces inteligentes e intuitivas que se encuentran integradas en partes y objetos corrientes, todo esto en un entorno que sea capaz de reconocer y responder a la presencia y necesidades de diferentes individuos, de una forma completamente discreta e imperceptible más que a través de los resultados obtenidos. La idea es que cada uno de los sensores pueda estar integrado en forma ubicua (trasparente al usuario) y adquirir datos del entorno lo que permita actuar de una determinada manera en base a los mismos.

La conectividad y las aplicaciones móviles serán de vital importancia para obtener una comunicación integral dentro del ambiente inteligente para poder llevar el control del mismo.

Dentro de un medio ambiente inteligente se podrá mediante sensores de proximidad y conectividad interactuar con las luminarias, climatizar el ambiente, etc.

Es decir es posible pensar en una red de sensores implementada nativamente en IPv6; en donde se aprovechen principalmente las siguientes características del protocolo: **(1)** Mayor Espacio de Direccionamiento: Al tener cada dispositivo una IP accesible públicamente por internet, se simplifica considerablemente la configuración ya que incluso podrían montarse sensores y actuadores en forma gradual, sin necesidad de configurar un mediador en cada locación para acceder a ellos, evitando configuración de routers y redirecciones. Este esquema permitirá controlar y acceder a la información desde cualquier lugar incluso directamente desde un dispositivo móvil.**(2)** Calidad de Servicio y Seguridad: Es importante en este tipo de aplicaciones asegurar el tiempo de respuesta y la seguridad de la información. **(3)** Movilidad: Permitirá extender la red de sensores fuera de un ámbito determinado haciendo un monitoreo permanente cuando la persona se traslada de un lugar a otro. Cada dispositivo se conecta independientemente a internet aunque está dentro de la misma red de sensores. **(4)** Multicast: Permitirá que aquellos mensajes que tengan que enviarse a varios nodos de la red, se envíen exclusivamente a esos nodos. Reduciendo el tráfico de la red y mejorando la performance.

5. Conclusiones

IPv6 es un protocolo que surge para solucionar el problema de agotamiento de las direcciones Ipv4 y dada la gran cantidad de direcciones disponibles con sus 128 bits y a las innovaciones en cuanto a calidad de servicio,

seguridad, movilidad y asignación de las direcciones IP en forma transparente, facilita la implementación de redes P2P a tal forma que no solo podemos pensar en que se conformen con computadoras personales de cualquier tipo y servidores, sino también a un sin fin de objetos y sensores embebidos, en un ambiente que sea de forma transparente al usuario. Por ello se identifica a IPv6 como "la Internet de las cosas" en donde los distintos objetos pueden tener una dirección IP individual. Esto sumado a otras características que incorpora el protocolo, como por ejemplo: Calidad de servicio, Seguridad, Movilidad, Multicast, permite planificar y construir aplicaciones nativas de gran utilidad.

En el presente trabajo se exponen las ventajas de las redes P2P en IPv6, sobre todo cuando la cantidad de información a transferir es grande; así como la importancia de trabajar con sensores y actuadores para la construcción de aplicaciones innovadoras en forma nativa para IPv6.

Queda en evidencia que al implementar una red P2P con nodos que incluya computadoras de cualquier tipo, sensores y actuadores con conectividad mediante IPv6, se logran mejoras en cuanto a calidad de servicio, seguridad y movilidad. Por otra parte diversos tipos de aplicaciones se ven beneficiadas con las características del protocolo.

Es importante hacer hincapié en la construcción de las aplicaciones nativas. Seguramente en un futuro cercano, las redes P2P implementadas en IPv6 nos cambien la forma en que interactuamos con las "cosas".

6. Referencias

1. MSDN Library. 2014 http://msdn.microsoft.com/es-es/library/bb902818(v=SQL.105).aspx
2. Microsoft Press. Peer - to - Peer Networking. 2009 http://blogs.wrox.com/files/2013/05/502259_c45_p2_ed.pdf
3. Windows – Dev Center – Desktop. 2014 http://msdn.microsoft.com/en-us/library/windows/desktop/bb153250(v=vs.85).aspx
4. BitTorrent is a P2P file sharing client or torrent client, letting users to download and upload large files. 2014. http://www.bittorrent.com/intl/es/
5. FEYRER Hubert, O' Reilly. "The future of the Internet". 2001. http://onlamp.com/pub/a/onlamp/2001/05/24/ipv6_tutorial.html
6. GIULIANELLI D., RODRÍGUEZ R, VERA P, CORNEJO M. "Desarrollo de Aplicaciones Nativas para IPv6". 2013. http://sedici.unlp.edu.ar/bitstream/handle/10915/27093/Desarrollo+de+Aplicaciones+nativas+para+IPv6.pdf?sequence=1
7. ATZORI L., IERA A., MORABITO G., "The Internet of Things: A Survey", J. Computer Networks, Vol.54, no.15, Oct. 2010, pp 2787-2805G
8. CASTELLANI A.P., et al, Architecture and protocols for the Internet of things: A case study", Proc. 1st IEEE Int. Workshop on Web of Things (WoT 2010), IEEE PERCOM, 2010, pp 678-683
9. GUINARD D., TRIFA V., WILDE E., "Architecting a mash able open world wide web of things", TRCS-663 ETH Zurich, 2010. www.vs.inf.ethz.ch/publ/paprs/WoT.pdf
10. Internet of Things Consortium. http://iofthings.org/
11. C. Ortega Corral; L. E. Palafox Maestre. "6LowPAN, un paso inicial hacia el Internet de las cosas". 2014. http://www.conatic2014.com.mx/conatic2012/papers/CONATIC014.pdf
12. "Introducción a IPv6: Multicast en IPv6". Portal IPv6 Cuba. http://www.cu.ipv6tf.org/multicastipv6.htm
13. O. J. Salcedo Parra, D. Lopez, A.P. Ríos, "Desempeño de la calidad de servicio (QoS) sobre IPv6", vol. 15, núm. 28, enero-junio, 2011, Universidad Distrital Francisco José de Caldas, Colombia, pp32-41 http://www.redalyc.org/articulo.oa?id=257019614004
14. CISCO. The Internet of Things. How the Next Evolution of the Internet Is Changing Everything. 2011. http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
15. Windows TechNet Library. Peer Name Resolution Protocol. http://technet.microsoft.com/en-us/library/bb726971.aspx
16. MSDN Library. System.Net.PeerToPeer Namespace. 2014 http://msdn.microsoft.com/en-us/library/bb726971.aspx
17. MSDN Library. Peer-to-Peer Networking. 2014 http://msdn.microsoft.com/en-us/library/ms733761(v=vs.110).aspx
18. MSDN Library. ¿Qué es Windows Communication Foundation? 2014 http://msdn.microsoft.com/es-es/library/vstudio/ms731082(v=vs.100).aspx
19. MSDN Library. Conceptos básicos de Windows Communication Foundation. 2014 http://msdn.microsoft.com/es-es/library/ms731079(v=vs.110).aspx
20. G Mercado, R Borgo, F G Antivilo, C Taffernaberry, A Diedrichs, M Aguirre, M Robles, G Gruynwaldt, G Tabacchi, S Tromer, M Pecchia. "Aplicaciones de Internet de las Cosas - SIPIA6 - Red de Sensores Inalámbricos con IPv6", UTN Facultad Regional Mendoza. 201. http://hdl.handle.net/10915/27092
21. ALFARO J. G.. "Desarrollo de Sistema de Gestión de Alumbrado Público Mediante WSN". Universidad Politécnica de Madrid. 2013. http://oa.upm.es/22306/1/PFC_JORGE_GALVEZ_ALFARO.pdf

Specifying Security Characteristics, Attributes, and Metrics for Evaluating Web Applications

Alexander Dieser, Guillermo Covella, Luis Olsina
Universidad Nacional de La Pampa, Facultad de Ingeniería,
General Pico, La Pampa, Argentina
[alexander.dieser, guillermo.covella]@gmail.com, olsinal@ing.unlpam.edu.ar

Abstract
At a time when almost every facet of life is influenced by Information and Communications Technologies (ICT), security becomes a critical, widespread issue. Addressing ICT Security issues imply not only to solve problems after they arise, but also to foster its systematic, sustainable and integrated lifecycle management. This work is aimed at defining and specifying Security characteristics, sub-characteristic and attributes, which are supported by an integrated strategy for measurement, evaluation and recommendation. In particular, we specify a set of about twenty-five security measurable attributes and sub-characteristics. Additionally, the strategy's steps are illustrated by using excerpts of a case study performed on a well-known web application.

Keywords
Security Characteristic and Attribute. Evaluation. Metric. Indicator. Web Application.

1. Introduction

Nowadays, ICT (Information and Communications Technologies) Security is largely a matter of interest to individuals, business and state/government which requires, to succeed, a systematic, engineered, and multi-disciplinary approach. The globalized and interconnected world we live in has become a sort of battlefield, where some “cyber battles” have a direct impact on the real world and, of course, in our lives.

Not only our laptops, PC's, servers and smart devices (e.g. smartphones, watches and glasses too) which are always connected downloading and sending data, but also the vast critical state infrastructure (e.g. energy, communications, transport, etc.) are handled using common ICT resources and share, not rarely, the same network. The difference may be in the sophisticated protection mechanisms used by the latter, but protection principles are the same, for example, only authorized users should access, sensitive data cannot be tampered with, and so on.

In this regard, individuals are frequently advised about security issues on using ATM´s, Internet services, on-line credit-card operations, etc. At the same time, media reflects meaningful cybercrime facts [4], whose victims are at first, but not only, the states/governments. However, at different levels, everybody is exposed against the same risks: from the most common, like Spamming, Stealing Credit Card Information, Cyber Trespass [17] to more complex ones, such as Cyber Theft, Cyber Vandalism/Terrorism, or Denial of Service.

Thus, whose are responsible of any organization to deal with this critical issue need to develop, at a long-term, not only a particular solution to single vulnerabilities, but also a robust strategy to accurately diagnose and support the whole ICT security area. In order to achieve this goal, the understanding and characterization of the meaningful assets to be managed and protected are key factors. Moreover, security requirements of these assets -sometimes intangible ones like data or pieces of software-, need to be specified clearly at different levels of granularity, i.e., their main characteristics, their attributes and relationships among them. In this sense, the use of quality (security) models and strategies for security management can be a valuable contribution.

Considering this approach, the particular contributions of this paper are twofold: on the one hand, the method for defining and specifying security characteristics and attributes, i.e., the security non-functional requirements specification for a selected target entity at hand; and, on the other hand, the instantiation of a well-known measurement and evaluation (M&E) strategy for systematically exploiting those specifications. Both contributions are illustrated using excerpts of a security evaluation case study performed on a web application used in Argentinean universities [13].

Following this introduction, Section 2 provides an overview of state-of-the-art ICT Security works, summarizing those that characterize and model the security feature. Section 3 specifies Security characteristics and attributes considering both the ISO 25010 standard [6] and the well-known OWASP (*Open Web Application Security Project*) mechanisms [16]. Then, Section 4 illustrates the specification of Security characteristics and attributes used in the abovementioned case study, where a detailed analysis of attributes is included. Using a specific quality M&E strategy and its

process [2, 14], Section 5 shows detailed specifications of metrics and indicators as well as the analysis of results. Finally, Section 6 draws the main conclusions and outlines future work.

2. Related Work

ICT Security issues have gone mainly from protection of corporation data and information assets to embrace new ends: as a matter of government security [9], on the one hand, to personal privacy protection issues, on the other hand.

Related to the former, let's say that today not only the cyber war but also the real war is under the umbrella of ICT Security. As per [8], "*modern weapon systems are an amazing collaboration of people, processes, hardware, software, and firmware that must function correctly to enable mission-essential capabilities; they are large, complex System-of-Systems* (SoS) *implementations*". Such SoS implementations are so far clearly monopolized by government agencies.

This government's interest is expressed, for example, by publishing a SE (Software Engineering) guide for SoS [12]. However, "*even though SE and SoS SE practices have been established for a number of years and used in different fields, they have not been widely accepted or applied to the security domain*" [8]. These authors also claim that "*traditional fields of security engineering include security protocols, trust relationships, access controls, and cryptology, with each supplying a vast set of best practices and helpful resources, which should be explored for applicability early in the design phase*"

Related to the latter, i.e. personal privacy protection issues, nowadays, the expansive growing of personal computing and smart devices (e.g. 56 billion smartphone apps, and 14 billion tablet apps downloaded, indicated in [1] for 2014) has extended the interest on ICT Security to individuals too. Supporting this phenomenal growth, there exists a continuous community effort to prevent, detect and, eventually, suppress (sanitize) vulnerabilities that cause damages to software-based apps and devices. OWASP initiatives [16] are a great example of this support.

OWASP, as they state is "*an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted*" under the "*Free & Open, Governed by rough consensus & running code ..., and Risk based approach*" principles.

The OWASP contribution to small, mid and non-profit organizations and government agencies of growing countries can be considered very important in terms of data, application and other ICT assets protection at a low implementation cost. Specifically, it provides a lot of meaningful endeavors in different formats such as tutorials, guidelines, rule sets, and so forth. The insight and added value of OWASP proposals reside on the practical utility for developers, managers and, why not?, end users too.

However, to widely support this kind of efforts in a more sound and rigorous way, a richer conceptual support should help to avoid ad-hoc solutions that not lead but rather be always behind the threats. In this regard, standards are always welcomed, particularly, those coming from well-known and active organizations and agencies [6, 11], since they contribute to valuable discussions and more agreed and accurate definitions which, in turn, lead to deeper ICT Security understanding and definitions. There are abundant literature in the ICT Security area, [6, 7, 10, 16] just to quote a few, and often "Security" is referred –from the nonfunctional viewpoint- to the CIA (*Confidentiality*, *Integrity* and *Availability*) triad. For example, Integrity is defined in ISO 25010 [6] as the "*degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data*". Even though ISO standards are usually valuable sources of definitions for characteristics in the security area, they aim at defining higher level security characteristics very often neglecting lower level definitions and specifications of sub-characteristics and measurable attributes.

In this way, we have identified many security sub-characteristics for Integrity, Confidentiality, Authenticity characteristics which are not defined in [6]. For instance, we define the Cross-site Scripting Immunity (1.2.1) sub-characteristic in Table 1 as the "*degree to which the system ensures the integrity of data by providing cross-site scripting immunity capabilities*". Moreover, we have identified for 1.2.1 many attributes that can be measured and evaluated, as we discuss later on.

On the other hand, ISO Security is one out of eight characteristics that relates to system quality. We adhere to the ISO approach, in which ICT Security can be seen as one of the quality factors for evaluating software product or system quality. For example, we can say that a lower system security performance results in a lower level of system quality too [13]. The higher the application vulnerability is the lower the system security and, consequently, the lesser the quality.

Moreover, we have also strengthened this view, by supporting the ICT Security assessment area with an integrated M&E strategy, which is based on metrics (for measurement) and indicators (for evaluation and interpretation). This strategy, named GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) [14, 15] can be used to understand and improve the quality of any organizational entity (asset) such as resource, product, system, system in use, among others. Specifically, we developed a set of metric and indicator specifications for vulnerability attributes regarding the

Security characteristic for a web information system as the target entity [13].

In the nutshell, vulnerabilities can be seen as inherent weaknesses in a target entity that could be exploited by a threat source. Therefore, understanding the current quality acceptability level met for each vulnerability attribute -by performing measurement and evaluation- can help to plan actions for change and improvement. The underlying hypothesis is that each meaningful attribute associated with the target entity to be controlled, should show the highest quality satisfaction level as an elementary non-functional requirement. Thus, the higher the security quality indicator value achieved for each attribute, the lower the vulnerability indicator value and so also the potential impact.

Even though many higher level security characteristics have been defined in the quoted related work, explicit definitions and specifications of its lower level sub-characteristics and attributes have very often left apart. Thus, the present work (instead of [13]) contributes mainly in this sense.

3. Security in ISO Standard

According to the GOCAME conceptual framework, a system is a target entity category, which can include many sub-entities in addition to related context entities that model the situation of the target entity. Thus, when talking for instance about a web application, the application server as part of the target system or even the network infrastructure can be included. Additionally, in GOCAME the quality requirements are specified and instantiated in a tree-structured form, called non-functional requirements tree. The tree is a possible instantiation of a quality model. Usually, a quality model can be taken from an international standard such as ISO 25010. Quality models for entities structure quality requirements regarding main characteristics that are further subdivided into sub-characteristics and associated attributes (or properties). In sub-section 3.1 we go further into security-related ISO 25010 sub-characteristics.

On the other hand, a basic issue is how can we discover security attributes and lower-level sub-characteristics for entities? A potential source for discovering attributes and lower-level sub-characteristics can be security policies and mechanisms [3], which are part of a given target entity. Security policies are abstract declarations of what is allowed and what is not, and they are enforced by security mechanisms. In other words, a security policy leads us to security mechanisms where we can find security attributes to measure and evaluate. Moreover, a security policy could also represent an attribute itself. In sub-section 3.2 we illustrate by examples the relationship between security policy/mechanism and sub-characteristic/attribute.

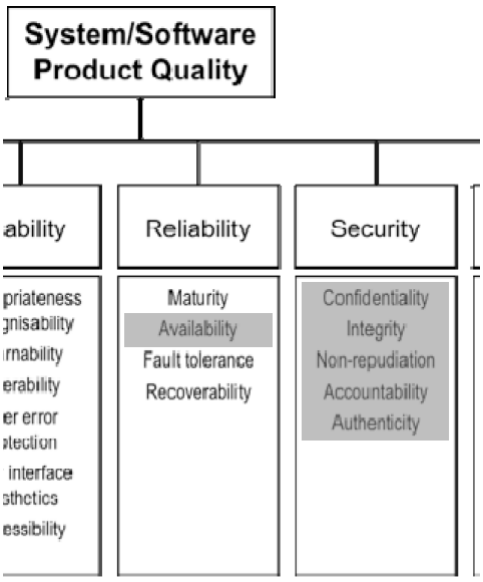


Figure 1. Excerpt of the ISO 25010 System Quality Model. For the Security characteristic the five sub-characteristics are highlighted; for Reliability, the Availability sub-characteristic is remarked.

3.1 Security as per ISO 25010

Considering the Security characteristic, ISO 25010 specifies it with five sub-characteristics, namely: Confidentiality, Integrity, Authenticity, Accountability, and Non-repudiation. Nevertheless, there is another important security characteristic that ISO doesn't include in its Security characteristic, namely: Availability. Instead, Availability is included as one out of the four sub-characteristics of the Reliability characteristic (see Figure 1).

Thus, when instantiating a security requirements tree using this standard, it might be considered both ISO characteristics viz. Security and Reliability. The former represents the "*degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization*"; while the latter, the "*degree to which a system, product or component performs specified functions under specified conditions for a specified period of time*". Following we describe some of their sub-characteristics (see Table 1), mainly those that will be used for illustration purposes in next sections.

Confidentiality is defined as the "*degree to which a product or system ensures that data are accessible only to those authorized to have Access*". In order to check if the target entity accomplishes this sub-characteristic, one or more sub-entities should be taken into account.

Table 1. Definition of Security characteristics and sub-characteristics used in [13]. (Note that characteristic codes match those codes in Table)

Characteristic/Sub-characteristic	Definition
Security (1)	Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization (Note that this definition is taken from [6]).
Confidentiality (1.1)	Degree to which a product or system ensures that data are accessible only to those authorized to have access [6].
Authentication Schema Protectability (1.1.1)	Degree to which the system ensures the confidentiality of data by providing access protection capabilities.
Session Data Protectability (1.1.2)	Degree to which the system ensures the confidentiality of data by providing session data protection capabilities.
Integrity (1.2)	Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data. ISO Note: Immunity (the degree to which a product or system is resistant to attack) is covered by integrity [6].
Cross-Site Scripting Immunity (1.2.1)	Degree to which the system ensures the integrity of data by providing cross-site scripting immunity capabilities.
Authenticity (1.3)	Degree to which the identity of a subject or resource can be proved to be the one claimed [6].
Session ID Protectability (1.3.1)	Degree to which the system prevents the session ID from being compromised by providing protection against ID-thieving and ID-tampering
Session impersonation Protectability (1.3.2)	Degree to which the system ensures session impersonation protection by providing secure session handling protocols.
Session Management Protectability (1.3.2.1)	Degree to which session management configurations satisfy security standards
Password Protectability (1.3.3)	Degree to which password management satisfies security standards (e.g., aging policy and string robustness)

Control access policies are for instance a good place to start with; e.g., many times a bad definition of roles and permissions leads to an information leakage. Also, weaknesses of cipher protocols in data transfer also attempts against confidentiality.

Integrity was cited and defined in Section 2. In order to check for integrity issues, we have to focus on data storage. On the client side, for example, cookies are a good place to look for integrity weaknesses. Many of these cookies could carry permanent information. If this data changes without the consent of an authorized user, then an integrity issue will arise. On the server side, the main source of integrity problems is the database. Lastly, another source is the network, where data flows and can be modified or tampered with if it is not correctly handled.

Another Security sub-characteristic is Authenticity, which is defined as "*degree to which the identity of a subject or resource can be proved to be the one claimed*" [6]. All aspects concerning user identity fall into the Authenticity scope. For example, every impersonation attack strictly belongs to this sub-characteristic. Weaknesses in authentication schemas often lead to authenticity problems, because an attacker could trespass the login mechanism and take control of a legitimate user account. In this case, a malicious entity gets its objectives on its own. There is another case where a malicious entity uses a valid session to make some legitimate user to perform malicious actions without being aware. So when evaluating authenticity both cases should be taken into account.

Regarding the ISO Reliability characteristic, just the Availability sub-characteristic is related to security as per the abovementioned CIA triad. It is defined in [6] as "*degree to which a system, product or component is operational and accessible when required for us*". Most of the issues related to availability can be found on the server side, such as DoS attacks. OWASP also provides a definition for this kind of attack: "*the Denial of Service (DoS) attack is focused on making unavailable a resource (site, application, server) for the purpose it was designed. There are many ways to make a service unavailable for legitimate users: by manipulating network packets, programming, logical, or resources handling vulnerabilities, among others*".

It is worthy to remark that characteristics and sub-characteristics are abstract calculable concepts which can be evaluated but not measured directly for an instantiated requirements tree. Only associated attributes can be measured -and evaluated as elementary nonfunctional requirements. As above mentioned, a potential source for discovering security attributes (and lower-level sub-characteristics) can be security policies and mechanisms.

3.2 Security Policies and Mechanisms

At the beginning of Section 3, we shortly explained security policies/mechanisms as a source to find

attributes. In this sub-section we illustrate a potential linkage between security policies/mechanisms, and sub-characteristics/attributes. Although not every security policy/mechanism maps directly to a sub-characteristic/ attribute, most of them could serve as a base for defining the latter. A clear example is the login system or access module, where the application asks for authentication data to grant the user with access. This module is a security mechanism itself, and e.g., the Authentication Schema Protectability (1.1.1) sub-characteristics deals with it. We define Authentication Schema Protectability in Table 1 as "*degree to which the system ensures the confidentiality of data by providing access protection capabilities*". In turn, we define the Authentication Schema Bypass (1.1.1.1) attribute for this sub-characteristic in Table 2 as follow: "*due to negligence, ignorance or understatement of security threats often result in authentication schemes that can be bypassed by simply skipping the login page and directly calling an internal page that is supposed to be accessed only after authentication has been performed*".

Other example is the password for user authentication. There are security policies commonly adopted for applications which require a login in which the password should be strong enough in order to avoid (or make difficult) guess attacks or brute force attacks. Therefore, the security policy states that the password string must meet specific properties such as length, type of characters (alphabetic, numeric, symbols, lowercase and uppercase), minimum occurrences of every type of character, among others. The associated security mechanism implementing this policy is a coded script which makes the required validations on the password at the setting time.

To cover the above situation, we have defined the Password Protectability (1.3.3) sub-characteristic as "*degree to which password management satisfies security standards (e.g., aging policy and string robustness)*", which includes two attributes namely: Password Aging Policy (1.3.3.1) and String Password Strength (1.3.3.2). The latter is defined in Table 2 as: "*password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. In its usual form, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. The strength of a password is a function of length, complexity, and unpredictability*".

4. Illustrating the Specification of Security Sub-characteristics and Attributes

In this Section, using excerpts of a security case study, we illustrate the first GOCAME activity. Specifically, we analyze security sub-characteristics and attributes of the instantiated requirements tree (see Table 3) that was not thoroughly discussed in [13]. The case study in this cited work was carried out to a M&E quality (security) project, where the concrete target entity was the "XYZ register system" -a student management web system widespread used in Argentinean universities.

On the other hand, the GOCAME first activity (see A1 in Figure 2) deals with the definition of non-functional requirements. The information need for the M&E security project stated that "*it is required to know the security level for XYZ web application from a student's point of view, in order to check the actual security level against an acceptable satisfaction threshold*". So, the purpose was firstly to understand the current satisfaction level achieved for the external quality of the XYZ web application, particularly for non-vulnerabilities regarding the Security quality focus, from the student user viewpoint. To accomplish this, a set of Security characteristics and attributes were selected considering the ISO 25010 quality model (as shown in Section 3) and OWASP documents. For example, an attribute (property) is specified considering its name, definition, objective and additional comments if needed.

Regarding the instantiated quality (security) model for the XYZ entity, we selected three ISO Security sub-characteristics viz. Confidentiality, Integrity and Authenticity, as we introduced in sub-section 3.1. (Note that Availability was not included in the tree since it is more related to the server application rather than to the assessed target entity). Additionally, twenty-five new concepts were specified which were not included in ISO 25010, i.e., seven sub-sub-characteristics and eighteen measurable attributes. The whole definitions of these new concepts are in Table 1 and Table 2. Also, the yielded requirements tree after performing A1 is in Table 3.

For Confidentiality, we associated two sub-characteristics with one attribute each, namely, Authentication Schema Protectability (1.1.1) and Session Data Protectability (1.1.2). For instance, 1.1.2 is defined as "*degree to which the system ensures the confidentiality of data by providing session data protection capabilities*". This sub-characteristic is based on the fact that data might be stored in a session cookie. Cookies are small pieces of information which are interchanged between client and server in order to support the concept of "session" (note this session concept will be further explained for Authenticity).

The 1.1.2 sub-characteristic includes the Session Data Theft Protectability (1.1.2.1) attribute defined as "*session-data cookies can be stolen by several means. One, for instance, is XSS. Some attempt to avoid stealing would be to disable the access via javascript. Another approach would be to restrict the cookie transmission only to SSL (Secure Socket Layer) channels*". So, the objective for measuring this property is to determine if session-data cookie has the aforementioned protection.

Table 2. Specification of attributes. (Note that attribute codes match those in Table 3)

Attribute Name	Attribute Definition/Objective/Comment
Authentication Schema Bypass (1.1.1.1)	**Definition:** Due to negligence, ignorance or understatement of security threats often result in authentication schemes that can be bypassed by simply skipping the login page and directly calling an internal page that is supposed to be accessed only after authentication has been performed. **Objective:** Find out the degree to avoid bypassing the authentication schema.
Session Data Theft Protectability (1.1.2.1)	**Definition:** Session-data cookies can be stolen by several means. One, for instance, is XSS. Some attempt to avoid stealing would be to disable the access via javascript. Another approach would be to restrict the cookie transmission only to SSL (Secure Socket Layer) channels. **Objective:** Determine if session-data cookie has the aforementioned protection
Reflected Cross-Site Scripting Immunity (1.2.1.1)	**Definition:** The non-persistent (or reflected) cross-site scripting vulnerability shows up when the data provided by a web client (most commonly in links or in HTML form submissions, by means of URL variables) is used immediately by server-side scripts to generate a page of results for that user, without properly sanitizing the request. Because HTML documents structure that mixes control statements, formatting, and the actual content, any non-validated user-supplied data included in the resulting page without proper HTML encoding, may lead to markup injection. **Objective:** Determine the degree to prevent the execution of offending code trough the URI on web browsers. **Comment:** A classic example of a potential vector is a site search engine: if one searches for a string, the search string will typically be redisplayed verbatim on the result page to indicate what was searched for. If this response does not properly escape or reject HTML control characters, a cross-site scripting flaw will be raised. A reflected attack is typically delivered via email or a neutral web site. The bait is an innocent-looking URL, pointing to a trusted site but containing the XSS vector (in an URL variable). If the trusted site is vulnerable to the vector, clicking the link can cause the victim's browser to execute the injected script.
Stored Cross-Site Scripting Immunity (1.2.1.2)	**Definition:** Stored XSS occurs when a web application gathers potentially malicious input from a user, then stores that input in a data store for later use (persistent data), so the input that is stored is not correctly filtered. As a consequence, the malicious data will appear to be part of the web application and run within the user's browser under the privileges of the application. **Objective:** Determine the degree to prevent the execution of offending code trough stored data on web browsers. **Comment:** Stored XSS is particularly dangerous in application areas where users with high privileges have access. When the administrator visits the vulnerable page, the attack is automatically executed by its browser. This might expose sensitive information such as session authorization tokens.
DOM-based Cross-Site Scripting Immunity (1.2.1.3)	**Definition:** DOM-based XSS (also called "type-0 XSS" in some texts) is an XSS attack wherein the attack payload is executed as a result of modifying the DOM "environment" in the victim's browser used by the original client side script, so that the client side code runs in an "unexpected" manner. That is, the page itself does not change, but the client side code contained in the page executes differently due to the malicious modifications that have occurred in the DOM environment. **Objective:** Determine the degree to prevent the execution of offending code trough data inputs directly executed on web browsers (without passing through the server).
Cross-Site Request Forgery Immunity (1.2.1.4)	**Definition:** Cross-Site Request Forgery (CSRF) is an attack that tricks the victim into loading a page that contains a malicious request. It is malicious in the sense that it inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something. The attacker uses the knowledge of the URL that performs some operation using a user's session, and, if the session is still active, it can access the site and perform harmful operations without the user's knowledge. **Objective:** Determine the degree to prevent a user from performing actions out of the application interface scope. **Comment:** A typical CSRF scenario is embedding known secure URL to HTML image tag, for example <img src="http://www.mybank/com/myaccount/sendmoney.html?toaccount=122121">. If the unsuspecting user loads a malicious web page with this URL, and at the same time user has the active session to the www.mybank.com web application, this image tag will be able to transfer money from the user's account without his/her knowledge. Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser.
Session Data Tampering Protectability (1.2.2)	**Definition:** Session data is stored in session-data cookies within the browser. Most web applications do not want their users (or other parties, i.e. attackers) to be able to modify session data. In fact, the ability to modify session data can lead to security vulnerabilities. The server can sign the transmitted session data in order to prevent any tampering with on the client side and verify the session data when received from the client within requests. **Objective:** Determine if session data can be manipulated (tampered with).
Session ID Theft Protectability (1.3.1.1)	**Definition:** Session-id cookies can be stolen by several means. One, for instance, is XSS. Some attempts to avoid stealing would be to disable the access via javascript, restrict the cookie transmission only to SSL channels, and, avoid passing ID through URL. **Objective:** Determine if session-id cookie has the aforementioned protection.
Session ID Tampering	**Definition:** Session ID is stored in session-ID cookie within the browser. It's almost mandatory that users (or other parties, i.e. attackers) don't be able to modify session ID. In fact, the ability to modify session ID can lead

Attribute Name	Attribute Definition/Objective/Comment
Protectability (1.3.1.2)	to security vulnerabilities. The server can sign the transmitted session ID in order to prevent any tampering on the client side and verify the session ID when received from the client within requests. **Objective:** Determine if session ID can be manipulated (i.e., tampered with).
Session ID Expiration (1.3.2.1.1)	**Definition:** A user's session cookie (session-ID cookie) for a web application exists in temporary memory only while the user is reading and navigating the website. When an expiration date or validity interval is not set at cookie creation time, a session cookie is created. Web browsers normally delete session cookies when the user closes the browser. **Objective:** Determine if session cookie is set as a session cookie, or if it expires at a short-secured time.
Session Expiration Due to Idle Timeout (1.3.2.1.2)	**Definition:** The user remains idle for a certain amount of time and the application automatically logs him/her out. The most appropriate logout time should be a right balance between security (shorter logout time) and usability (longer logout time) and heavily depends on the criticality of the data handled by the application. A 60 minutes logout time for a public forum may be acceptable, but such a long time would be too much in a home banking application. In any case, any application that does not enforce a timeout-based logout should be considered not secure, unless such a behavior is addressing a specific functional requirement. **Objective:** Determine if logout time (session expiration time) is secure.
Re-authentication Mechanism Availability (1.3.2.1.3)	**Definition:** When there exists sensitive data in an application, it's almost mandatory to implement a re-authentication mechanism prior to execute actions that affect the data. This capability ensures security for sensitive data even when user authenticity was compromised. **Objective:** Determine if the application ask for user credentials prior to execute an action over sensitive data.
Session ID Regeneration Availability (1.3.2.1.4)	**Definition:** Regenerating the session token after each change in user privileges helps the application to protect against any session fixation attack. **Objective:** Determine if session ID is regenerated after log in and log out.
Keep-me-logged-in Mechanism Availability (1.3.2.1.5)	**Definition:** The "keep-me-logged-in" capability allows a user not to log in after he/she leaves the application without log out (as if session never expires). This mechanism is only available for the computer from which the user chooses "keep me logged in". Hence, he/she can quit the application (without logging out), even can shut down the PC, and still being able to access the application without the necessity of a new log in. **Objective:** Determine the existence of a "keep-me-logged-in" mechanism.
Session Non-Replay Protectability (1.3.2.2)	**Definition:** The attacker obtains a valid session ID after it is associated with a legitimate user session. Using the stolen-session token, the attacker can gain the permissions of the user and access to the web service. **Objective:** Determine the degree of protection against hijacking attacks, assuming that an attacker has obtained a valid session ID.
Session Fixation Protectability (1.3.2.3)	**Definition:** Session fixation arises in systems where the attacker is able to specify the session identifier for the session of a legitimate user. In this attack, the attacker first sets the application user's session ID to an explicit value. The victim then authenticates him/herself to the application server using the session ID fixed by the attacker. Once the user is authenticated, the attacker can use the predefined session ID to access the application server and impersonate the victim user. **Objective:** Determine if session is vulnerable to a fixation attack.
Password Aging Policy (1.3.3.1)	**Definition:** As the password gets old, the likelihood of being guessed/discovered grows. Because of this, is easier for attackers to find out the password due to, for instance, they can perform long-time term attacks. **Objective:** Find out the degree to cover the password aging issue.
String Password Strength (1.3.3.2)	**Definition:** Password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. Usually, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. **Objective:** Determine the password strength. **Comment:** The strength of a password is a function of length, complexity, and unpredictability.

Under the Integrity's scope, we have specified a sub-characteristic and five attributes as shown in Table 3. The Cross-Site Scripting Immunity (1.2.1) sub-characteristic is defined as "*degree to which the system ensures the integrity of data by providing cross-site scripting immunity capabilities*". It is worth mentioning that Cross-Site Scripting attack (XSS) is widely known among ICT Security issues. For example, in OWASP [16], XSS attacks are a type of injection problem, in which malicious scripts are injected into the otherwise benign and trusted web applications. These attacks occur when an attacker uses the application to send malicious code, generally, in the form of a browser side script to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user in the output it generates without validating or encoding it. The malicious script can access any cookies, session tokens, or other sensitive information retained by your browser and used with that site.

There exist a few different types of XSS attacks. Each attribute defined for the 1.2.1 sub-characteristic covers one XSS attack type. For instance, the Stored Cross-Site Scripting Immunity (1.2.1.2) attribute is defined as "*stored XSS occurs when a web application gathers input from a user which might be malicious, and then stores that input in a data store for later use (persistent data). The input that is stored is not correctly filtered. As a consequence, the malicious data will appear to be part of the web site and run within the user's browser under the*

privileges of the web application". So, the objective for measuring this attribute is to determine the degree to prevent the execution of offending code trough stored data on web browsers. As additional comment, stored XSS is particularly dangerous in application areas where users with high privileges have access. When the administrator visits the vulnerable page, the attack is automatically executed by its browser. This might expose sensitive information such as session authorization tokens.

Regarding the Authenticity sub-characteristic (see Table 3), this is strongly related to the web application session concept. OWASP states that a session is "*a sequence of network HTTP request and response transactions associated to the same user. Modern and complex web applications require the retaining of information or status about each user for the duration of multiple requests. Therefore, sessions provide the ability to establish variables –such as access rights and localization settings – which will apply to each and every interaction a user has with the web application for the duration of the session. Web applications can create sessions to keep track of anonymous users after the very first user request. An example would be maintaining the user language preference. Additionally, web applications will make use of sessions once the user has authenticated. This ensures the ability to identify the user on any subsequent requests as well as being able to apply security access controls, authorized access to the user private data, and to increase the usability of the application. Therefore, current web applications can provide session capabilities both pre and post authentication. Once an authenticated session has been established, the session ID (or token) is temporarily equivalent to the strongest authentication method used by the application, such as username and password, passphrases, one-time passwords (OTP), client-based digital certificates, smartcards, or biometrics ...*"

Moreover, session protocols could have a lot of vulnerabilities, and that is why several security mechanisms are implemented for web applications. The objective is to grant security for sessions, so that nowadays crimes as impersonation attacks can be avoided. For instance, banking web applications often implement sessions which expire after a few idle minutes, trying to prevent active accounts from being manipulated by attackers. In many cases, the user is given with the possibility to set the idle timeout or even to disable this feature. Another usual mechanism is to ask the user for his/her credentials before a sensitive operation (e.g., a financial transaction) is performed.

For the target "XYZ register system" most of the data and information related to the session handle is therefore covered by Authenticity. Hence, four new sub-characteristics were specified such as Session ID Protectability (1.3.1), Session Impersonation Protectability (1.3.2), Session Management Protectability (1.3.2.1), and Password Protectability (1.3.3). Eleven attributes associated to them were specified as shown in Table 3.

For example, Session Impersonation Protectability (1.3.2) is defined as "*degree to which the system ensures session impersonation protection by providing secure session handling protocols*". One of the most relevant attribute, for our case study, associated to 1.3.2 was Session Non-Replay Protectability (1.3.2.2). It is defined as "*the attacker obtains a valid session ID after it is associated with a legitimate user session. Using the stolen session token, the attacker can gain the permissions of the user and access to the web service*". Assuming that an attacker has obtained a valid session ID, the degree of protection against hijacking attacks can be measured and evaluated.

Next Section summarizes the remainder of GOCAME activities.

5. Designing and Implementing the M&E to Assess Security using the GOCAME Strategy

In the previous Section, we illustrated the specification of security characteristics and attributes instantiated in a requirements tree, following the first GOCAME (A1) activity. This Section summarizes the activities that are performed from designing the measurement and evaluation through implementing the measurement and evaluation in addition to the analysis of results, i.e., following the A2 to A6 activities as depicted in Figure 2.

The requirements tree produced in the A1 activity serves as input for M&E activities. That is to say, the requirements tree represents the backbone for the remainder activities.

Measurement is specified and implemented by using metrics, which define how to represent and collect attributes' values; while evaluation is specified and implemented by using indicators, which define how to interpret attributes' values yielded by metrics, and how to calculate higher-level calculable concepts (sub-characteristics and characteristics) of the requirements tree.

In [13] we discussed why metrics and indicators are key, reusable organizational assets for providing suitable data and information for analyzing, recommending, and ultimately decision-making processes. So the reader can refer to it for a more thorough comprehension of the added value of metrics and indicators for M&E and change projects.

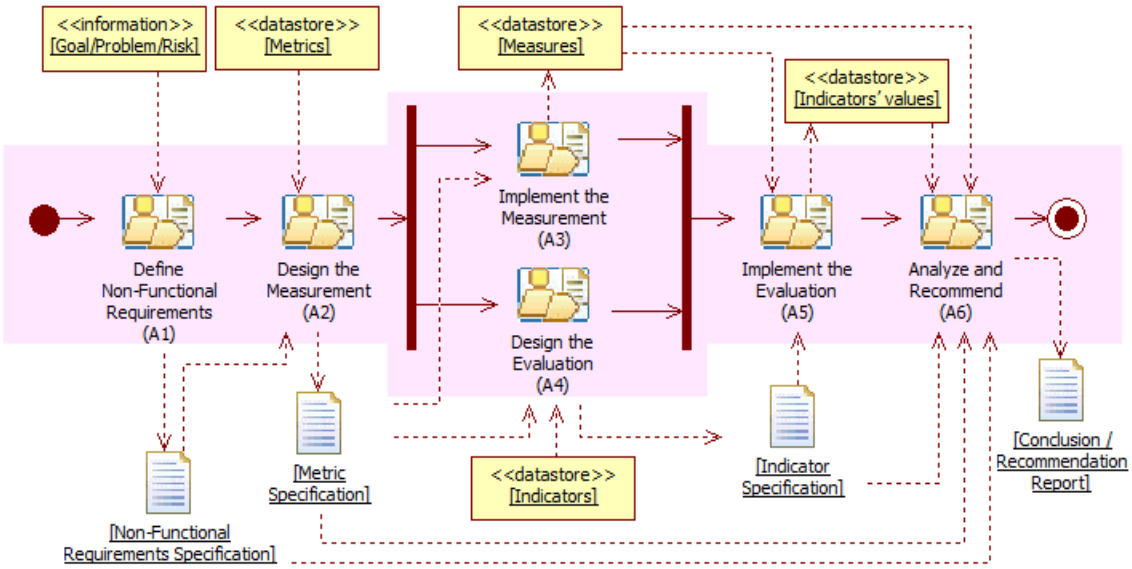


Figure 2. High-level activities for the GOCAME M&E process. Legend A means Activity

Table 3. Instantiated Security characteristics and attributes. (Note that attributes are highlighted in *italic*)

1.Security
1.1. Confidentiality
1.1.1. Access Schema Protectability
1.1.1.1. Authentication Schema Bypass
1.1.2. Session Data Protectability
1.1.2.1. Session Data Theft Protectability
1.2. Integrity
1.2.1. Cross-Site Scripting Immunity
1.2.1.1. Reflected Cross-Site Scripting Immunity
1.2.1.2. Stored Cross-Site Scripting Immunity
1.2.1.3. DOM-based Cross-Site Scripting Immunity
1.2.1.4. Cross-site request forgery Immunity
1.2.2. Session Data Tampering Protectability
1.3. Authenticity
1.3.1. Session ID Protectability
1.3.1.1. Session ID Theft Protectability
1.3.1.2. Session ID Tampering Protectability
1.3.2. Session Impersonation Protectability
1.3.2.1. Session Management Protectability
1.3.2.1.1. Session ID Expiration
1.3.2.1.2. Session Expiration Due to Idle Timeout
1.3.2.1.3. Re-authentication Mechanism Availability
1.3.2.1.4. Session ID Regeneration Availability
1.3.2.1.5.Keep-me-logged-in Mechanism Availability
1.3.2.2. Session Non-Replay Protectability
1.3.2.3. Session Fixation Protectability
1.3.3. Password Protectability
1.3.3.1. Password Aging Policy
1.3.3.2. String Password Strength

In a given M&E project, for each attribute of the requirements tree a metric should be selected in the A2 activity. Table 4 shows the direct metric specification for the String Password Strength (coded 1.3.3.2 in Table) attribute. The objective of the PS direct metric is to "*determine the strength requirements that the string password is forced to meet*". For repetitiveness, consistency and comparability reasons, a direct metric specification includes not only the measurement procedure (i.e. the counting rules) but also the scale, among other metadata.

From the evaluation design standpoint (A4 activity), for the above 1.3.3.2 attribute, the elementary indicator named Performance Level of the String Password Strength interprets the metric's value of it (see specification in Table 5). Note that each elementary indicator interprets the level of satisfaction achieved for each attribute regarding an attribute as an elementary non-functional requirement.

Furthermore, a new scale transformation and decision criteria (in terms of acceptability ranges) should be defined. In our case study, we have used three agreed acceptability levels in percentage unit. A value between zero and sixty ([0-60)) represents an "Unsatisfactory" level and indicates that "*change actions must be taken with high priority*"; a value between sixty and ninety-eight ([60-98)) represents a "Marginal" level and indicates "*a need for improvement actions*"; finally, a value between ninety-eight and a hundred ([98-100]) represents a "Satisfactory" level and indicates "*no need for further actions*". Additionally, in the A4 activity an aggregation model to calculate elementary, partial and global indicators –which include characteristics and sub-characteristics- should be selected, such as the LSP (*Logic Scoring of Preference*) [5] scoring model.

Table 4. Metric specification for the *String Password Strength* (1.3.3.2) attribute.

Attribute Name: (1.3.3.2) *String Password Strength*
Direct Metric Name: Level of Password Strength **(PS)**
Objective: Determine the strength requirements that the string password is forced to meet.
Author: Dieser A. Version: 1.0
Reference:http://technet.microsoft.com/en-us/library/cc756109(v=ws.10).aspx
Measurement procedure
Specification:
Change password of an existing and valid account.
PS = 0;
if (it is forced to write a password at least seven characters long**)**
then PS = 1;
if (it is forced to write a password that doesn't contain your user name, real name, or company name**)**
then PS = PS + 1;
if (it is forced to write a password that doesn't contain a complete dictionary word**)**
then PS = PS + 1;
if (it is forced to write a password that is significantly different from previous passwords**)**
then PS = PS + 1; /*Note that passwords that increment (*Password1*, *Password2*, *Password3* ...) are not strong*/
if (it is forced to write a password that contains characters from at least two of the groups below**)**
then PS = PS + 1;

Groups	**Examples**
Uppercase letters	*A, B, C …*
Lowercase letters	*a, b, c …*
Numerals	*0, 1,2, 3, 4, 5, 6, 7, 8, 9*
Symbols	*` ~ ! @ # $ % ^ & * ( ) _ + - = { } | \ : " ; ' <> ? , . /*

Type: Objective.
Numerical Scale
Representation: Discrete Value Type: Integer
Scale Type: Interval
Unit
Name: Requirements Acronym: R
Definition: Strength requirements

Table 5. Specification of the elementary indicator named *Performance Level of the String Password Strength.*

Attribute Name: (1.3.3.2) *String Password Strength*
Elementary Indicator:
Name: Performance Level of the String Password Strength **(P_PS)**
Author: Dieser A.
Version: 1.0
Elementary Model:
Specification: let PS be the direct metric then the mapping is:
P_PS = 100 iff PS = 5;
P_PS = 90 iff PS = 4;
P_PS = 60 iff PS = 3;
P_PS = 40 iff PS = 2;
P_PS = 20 iff PS = 1;
P_PS = 0 iff PS = 0;
Decision Criterion:
[Acceptability Levels]
Name 1**:** Unsatisfactory
Description**:** indicates change actions must be taken with high priority
Range**:** if 0 ≤ P_PS ≤ 60
Name 2**:** Marginal
Description: indicates a need for improvement actions
Range: if 60 < P_PS ≤ 98
Name 3: Satisfactory
Description: indicates no need for further actions
Range: if 98 < P_PS ≤ 100
Numerical Scale:
Representation: Continuous Value Type: Real
Scale Type: Proportion
Unit:
Name: Percentage Acronym: %

Implement the measurement (A3) activity consists of getting the measured value, either numeric or categorical, for each attribute, accordingly using as input the set of metrics selected in the A2 activity. Also, the set of tools (if any) selected for data collection and calculation are employed. For the PS direct metric, the measurement produced a value of 0, i.e., PS=0.

With regard to the evaluation activity, its implementation (A5) produces, first, elementary indicator values. So, by their interpretation, evaluators can gauge what level the attributes satisfy the established requirements by the stated information need. For example, the quality (security) satisfaction level that attribute 1.3.3.2 will reach, can be interpreted accordingly by the implementation of the indicator specified in Table 5.

Enacting the P_PS elementary model, the metric value of zero maps to the 0% indicator value, as shown in the fourth column in Table 6. This outcome falls at the bottom of the "Unsatisfactory" acceptability level, meaning that a change action must be performed with high priority, since certainly it represents an actual vulnerability in the target system security.

Once all the measurements and elementary evaluations -of the 18 attributes shown in Table 3- were performed, the selected aggregation (scoring) model should be executed. This A5 sub-activity yields all additional partial/global indicator values, taking as input all the elemental indicator values, in addition to weights and operators for aggregation and calculation. Partial/global indicator values are shown in the last two columns of Table 6.

Lastly, the A6 activity of the GOCAME process takes place: Analyze and Recommend. In A6, every work product (i.e. artifacts, outcomes) previously yielded is collectively and systematically analyzed in order to get recommendations that may lead to plan change actions for security improvement based on the exposed application vulnerabilities.

Table 6. Security indicator values in [%]. Acronym W stands for Weight; Op. stands for LSP Operator; EI stands for Elementary Indicator; PI/GI stands for Partial/Global Indicator.

Characteristic / Attribute	W	Op.	EI	PI	GI
1.Security		CA			17.67
1.1. Confidentiality	0.33	C--		100	
1.1.1.	0.6				
1.1.1.1.	1		100		
1.1.2.	0.4				
1.1.2.1.	1		100		
1.2. Integrity	0.33	C--		10.54	
1.2.1.	0.8	C-		1.08	
1.2.1.1.	0.25		100		
1.2.1.2.	0.25		0		
1.2.1.3.	0.25		100		
1.2.1.4.	0.25		0		
1.2.2.	0.2		100		
1.3. Authenticity	0.33	C--		12.47	
1.3.1.	0.45	C-		3.51	
1.3.1.1.	0.5		50		
1.3.1.2.	0.5		0		
1.3.2.	0.45	C--		32.75	
1.3.2.1.	0.3	A		40	
1.3.2.1.1.	0.3		100		
1.3.2.1.2.	0.2		0		
1.3.2.1.3.	0.2		0		
1.3.2.1.4.	0.2		0		
1.3.2.1.5.	0.1		100		
1.3.2.2.	0.35		0		
1.3.2.3.	0.35		100		
1.3.3.	0.1	A		0	
1.3.3.1.	0.4		0		
1.3.3.2.	0.6		0		

Consequently, the output of the A6 activity is the Conclusion/Recommendation report, as shown in Figure 2. We are not going further in the analysis of this report, so the reader can refer to [13]. However, from the last column of Table 6, we observe at first glance that the Security score for the XYZ application falls at the bottom of the "Unsatisfactory" acceptability level (17.67%). It means that change actions to the XYZ application version must be planned urgently, i.e. with high priority.

From the partial indicators (PI) column, we also see for instance that Authenticity needs changes as well. In turn, if we go deeper to the EI column, evaluators can realize that not only the illustrated 1.3.3.2 attribute have to be fixed, but also many other attributes that contribute to better performance for the Authenticity indicator.

Ultimately, by using GOCAME which is based on target entities, characteristics/attributes, metrics and indicators, it was possible to understand at the moment we carried out the case study (i.e. by end of 2012) the situation on the "XYZ" vulnerabilities. Then, for reducing the vulnerabilities' impact on the "XYZ" target entity, the planning of improvement actions on weakly benchmarked attributes/capabilities should be envisioned, considering the Conclusion and Recommendation report.

As the reader can surmise, in order to increase the 1.3.3.2 indicator value to the 100%, the value for the PS direct metric must be 5 as shown in Table 5 (recall that the measurement produced a value of PS = 0, which mapped to P_PS = 0%). If we look at the measurement procedure specification for the PS metric in Table 4, clearly we have the clue to fix this problem for guaranteeing the password string be the strongest. As a consequence, metric specifications may also help to design and implement the change actions for the improvement plan.

6. Conclusion and Future Work

To conclude this paper, we would like to point out the two listed contributions in Section 1. The first contribution viz. "*the method for defining and specifying security characteristics and attributes, i.e., the security non-functional requirements specification for a selected target entity at hand*" has been one of the main drivers of the paper.

We have argued that there is abundant literature in the ICT Security area (e.g., both *de facto* and official standards) that deals with Security from the nonfunctional requirements viewpoint. However, we have discussed that ISO documents aim at defining higher level security characteristics neglecting very often lower level definitions and specifications of sub-characteristics and measurable attributes. Conversely, OWASP is a very valuable source for documenting security attacks and mechanisms, but very often sound and explicit definitions and specifications of sub-characteristics and attributes for concrete entities are missing or blurred. So our approach contributes in establishing explicitly relationships and specifications for entities, characteristics, sub-characteristics and measurable attributes by means of quality (security) models and views. Particularly, we have illustrated security attributes for a specific case study, where attributes have explicitly a name, definition, objective, among other metadata.

Regarding the second stated contribution, we have used a well-known measurement and evaluation strategy and its process -i.e. GOCAME- for systematically exploiting those non-functional specifications by means of metrics and indicators. Basically, we have argued that metrics and indicators are useful organizational assets for providing suitable and consistent data and information for analyzing, recommending, improving, and decision-making activities. This contribution was also illustrated using excerpts of the abovementioned evaluation case study.

Regarding future work, we are planning to model more rigorously the relationships between security policies/mechanisms and sub-characteristics/attributes, which aim at favoring the identification and specification of sub-characteristics and attributes to be used in generic quality (security) models -in addition to the identification and specification of measurement procedures for metrics. Also, these specifications can be reused from catalogues, therefore fostering for instance the edition of security models at the moment of defining nonfunctional requirements, among other benefits.

7. Acknowledgment

This work is partially supported by the POIRE-10 project at UNLPam, and the PROMINF/SPU 2013-2016 plan, Argentina.

8. References

[1] ABI Research, "Android Will Account for 58% of Smartphone App Downloads in 2013, with iOS Commanding a Market Share of 75% in Tablet Apps" Available at https://www.abiresearch.com/press/android-will-account-for-58-of-smartphone-app-down, Accessed Jul., 2014.

[2] Becker, P., Papa, F., Olsina, L. Process Conceptual Base for Enriching a Measurement and Evaluation Ontology, In proceedings of XVII Conferencia Iberoamericana en Software Engineering (CIbSE'14), Pucón, Chile, pp. 53-66. ISBN: 978-956-236-247-4, 2014.

[3] Bishop M. *Computer Security: Art and Science*, Addison Wesley, 2002.

[4] Cornwell R., "US declares cyber war on China", Available at *www.independent.co.uk*, Accessed Jul., 2014.

[5] Dujmovic, J.J., A Method for Evaluation and Selection of Complex Hardware and Software Systems, *22nd Int'l Conf. for the Resource Management and Performance Evaluation of Enterprise CS. CMG 96 Proceedings*, pp. 368-378. 1996.

[6] ISO/IEC 25010. "Systems and software engineering - systems and software product quality requirements and evaluation (SQuaRE)."System and software quality models"; 2011.

[7] ISO/IEC 27004. *Information technology — Security techniques — Information security management — Measurement,* 2009.

[8] Mailloux L., Grimaila M., Colombi J., Hodson D. and Baumgartner G., "System Security Engineering for Information Systems", *Chapter 1, Emerging Trends in ICT Security, Elsevier (Morgan Kaufmann)*, 1st Edition, Akhgar & Arabnia (Eds.), Print Book ISBN: 9780124114746, eBook ISBN: 9780124104877, 2013, pp. 5-23

[9] Ministry of Foreign Affairs of the People's Republic of China, "International code of conduct for information security", *NETmundial, – Global Multistakeholder Meeting on the Future of Internet Governance*, Available at *content.netmundial.br/contribution/international-code-of-conduct-for-information-security/67),* Accessed Jul., 2014.

[10] Nelson C. Security Metrics: An Overview, In: ISSA Journal, pp. 12-18, August 2010.

[11] NIST SP 800-55. *Performance Measurement Guide for Information Security.* Available at http://csrc.nist.gov/publications/PubsSPs.html, July 2008, Accessed Jul., 2014.

[12] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, "Systems and Software Engineering. Systems Engineering Guide for Systems of Systems", Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008. Available at www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf, Accessed Jul., 2014.

[13] Olsina L., Covella G., Dieser A., "Metrics and Indicators as Key Organizational Assets for ICT Security Assessment", Chapter 2 in *Emerging Trends in ICT Security*, Elsevier (Morgan Kaufmann), 1st Edition, Akhgar & Arabnia (Eds.), Print Book ISBN: 9780124114746, 2013, pp. 25-44.

[14] Olsina L, Lew P, Dieser A, Rivera B. "Updating quality models for evaluating new generation web applications." *In: Journal of Web Engineering, Special issue: Quality in new generation web applications*, Rinton Press: New Jersey. 2012;11(3), pp. 209-246.

[15] Olsina L, Papa F, Molina H. "How to measure and evaluate web applications in a consistent way." In: *Springer, HCIS Series; Web engineering: Modeling and implementing web applications*. Rossi G, Pastor O, Schwabe D, Olsina L, editors. 2008, pp. 385-420

[16] OWASP: The Open Web Application Security Project. Available at: https://www.owasp.org, Accessed Jul., 2014.

[17] Razzaq A., Hur A., Farooq Ahmad H., Masood M. "Cyber Security: Threats, Reasons, Challenges, Methodologies and State ofthe Art Solutions for Industrial Applications", *ISADS, IEEE*. Print ISBN: 978-1-4673-5069-3, 2013, pp.1-6.

Benchmarking y optimización de algoritmos criptográficos

Miguel Montes
Facultad de Ingeniería
Instituto Universitario Aeronáutico
mmontes@iua.edu.ar

Daniel Penazzi
Famaf
Universidad Nacional de Córdoba
penazzi@famaf.unc.edu.ar

Resumen

El presente trabajo presenta los resultados de la optimización de algoritmos criptográficos en procesadores que disponen de las nuevas instrucciones AES, desarrolladas por Intel. En particular se trabajó sobre AES-CPFB y Silver, dos algoritmos de cifrado autenticado presentados en la competencia CAESAR.

1. Introducción

AES (Advanced Encryption Standard) es un algoritmo de cifrado de bloques estandarizado por NIST (National Institute of Standards and Technology) en 2001 [1] . Desarrollado por dos criptógrafos belgas, Vincent Rijmen y Joan Daemen, utiliza bloques de 128 bits, y admite claves de 128, 192 y 256 bits [2, 3].

Se trata del algoritmo de cifrado simétrico más utilizado, y es probablemente el algoritmo de cifrado mejor analizado de la historia, por lo que es frecuente su uso como componente de otros algoritmos y protocolos. Tal es el caso de Silver [4] y AES-CPFB [5], que constituyen el objeto del presente trabajo. Ambos son algoritmos de cifrado autenticado, presentados en el marco de la competencia CAESAR: Competition for Authenticated Encryption: Security,Applicability, and Robustness, y usan AES como componente esencial. AES-CPFB es un modo de operación, que utiliza AES en modo "caja negra", mientras que Silver utiliza una versión modificada de AES.

Si bien la seguridad es la característica primordial de un algoritmo criptográfico, también es extremadamente importante su velocidad. Los grandes anchos de banda en uso en la actualidad requieren de cifradores que sean capaces de procesar varios gigabits de datos por segundo. Eso ha motivado, por un lado, el desarrollo de algoritmos de cifrado de flujo cada vez más rápidos, y por el otro, la aceleración por hardware de algoritmos estándar como AES o SHA.

En este trabajo nos centraremos en la utilización de una de estas tecnologías de aceleración por hardware: las nuevas instrucciones AES desarrolladas por Intel, y disponibles en las últimas generaciones de procesadores x86.

2. Benchmarking

La medición de velocidad de algoritmos criptográficos presenta tres problemas básicos: qué medir, en qué unidades hacerlo, y cómo hacerlo. Se busca encontrar una métrica que permita hacer comparaciones significativas entre algoritmos. Al comparar algoritmos se suele recurrir a evaluar su comportamiento asintótico, pero ese recurso no es útil en este caso: todos los algoritmos de cifrado de bloques tienen un costo lineal con respecto al tamaño de la entrada expresado en bloques.

Sin embargo, suelen tener un elevado costo de inicialización, habitualmente asociado con la expansión de la clave, el cual se vuelve relevante para pequeños tamaños de entrada. Si bien este costo de inicialización se amortiza fácilmente en el cifrado de un archivo de varios megabytes, es significativo en el cifrado de paquetes pequeños a ser enviados por la red. Este problema es especialmente severo en el caso de los algoritmos de cifrado autenticado, en los cuales se suele utilizar un *nonce* asociado con cada mensaje, lo cual implica costos de inicialización adicionales para cada *nonce*. Por ello se suele medir el comportamiento de estos algoritmos para distintos tamaños de entrada, y a veces se usa un promedio ponderado de ciertos tamaños típicos de paquete. En nuestro caso hemos utilizado un promedio ponderado de paquetes de 44, 552, 576 y 1500 bytes, tal como se muestra en el cuadro 1. Esta proporción se corresponde con el *Internet Performance Index* (IPI) propuesto por David McGrew y John Viega [**?**], y utilizada por Rogaway en la implementación de referencia de OCB [6]. El IPI es el número esperado de bits procesados por ciclo de reloj cuando la distribución de paquetes

se corresponde con la del cuadro 1, y se considera un indicador útil del rendimiento de un módulo criptográfico que protege tráfico IP.

Además, hemos medido el rendimiento en paquetes de 65536 bytes, como muestra del comportamiento en mensajes largos.

Cuadro 1. Distribución típica de tamaños de paquete

Bytes	Peso
44	0.05
552	0.15
576	0.20
1500	0.60

Una vez definido qué vamos a medir, es necesario determinar en qué unidades mediremos. Parecería razonable medir en bytes por unidad de tiempo (por ejemplo megabytes por segundo), pero no es una solución adecuada, ya que es una métrica muy dependiente de la velocidad del procesador utilizado, y no permite comparar algoritmos entre sí. Es necesario encontrar una forma de medir que sea relativamente independiente de la velocidad del procesador, y que permita extrapolar el rendimiento de un algoritmo en distintos procesadores (al menos dentro de una misma arquitectura). Una solución es utilizar bits por ciclo, tal como en el IPI mencionado más arriba. Otra solución, muy difundida en el ambiente criptográfico y que es la que hemos adoptado, consiste en medir ciclos por byte (cpb). Es decir, cuantos ciclos de procesador se consumen para procesar un byte de entrada.

Nuestro promedio ponderado no es, pues, el IPI, ya que éste mide bits por ciclo y nosotros medimos ciclos por byte. Pero como utilizamos la distribución de tamaños de paquetes del IPI, y Silver y AES-CPFB operan sobre bytes completos, el IPI puede calcularse fácilmente a partir de nuestro promedio ponderado.

El siguiente problema es cómo medir esta velocidad. No existe un mecanismo uniforme de medición de los ciclos de procesador utilizados. Algunos mecanismos de benchmarking, como por ejemplo el de la biblioteca CryptoCPP, miden bytes por segundo y convierten a ciclos por byte conociendo la velocidad del procesador. En otros casos, se utilizan instrucciones del procesador que permiten obtener la cantidad de ciclos en forma directa. En nuestro caso, hemos utilizado este último sistema.

Como el trabajo de optimización se realizó en equipos con arquitectura x86, utilizamos la instrucción `rdtsc` (Read Timestamp Counter)[7]. Esta instrucción puede ser utilizada desde un programa en C utilizando *intrinsics*, en el caso de los compiladores de Intel y de Microsoft, y usando inline assembler en el caso de GCC.

El uso de `rdtsc` presenta varias sutilezas.

- Distintas familias de procesadores incrementan el contador de forma diferente. En procesadores como el Pentium M, P6, o los primeros Pentium 4, el contador se incrementa con cada ciclo del reloj del procesador. En procesadores posteriores, el contador se incrementa a una tasa constante, que depende de la máxima frecuencia nominal del procesador configurada en el momento del arranque. Esto significa que el contador sólo equivale a ciclos cuando el procesador funciona a esa frecuencia.
- Los procesadores actuales no trabajan a una frecuencia fija. Los mecanismos de ahorro de energía hacen que el procesador pueda trabajar en una gama de velocidades que pueden ser controladas por el sistema operativo. Por ejemplo, un procesador Core i7-4771 tiene una frecuencia mínima de 800 MHz y una frecuencia máxima de 3.5 GHz. El sistema operativo puede optar por correr siempre a 800 MHz (modo ahorro de energía), siempre a 3.5 GHz (modo rendimiento), o trabajar a demanda (800 MHz cuando no hay carga, y aumentar a 3.5 GHz cuando la hay). Esto altera las mediciones, por lo que es necesario forzar a que el equipo trabaje a su máxima frecuencia nominal (en este caso, 3.5 GHz).
- Los últimos procesadores Intel cuentan con una tecnología denominada Turbo Boost, que incrementa de forma automática la velocidad de procesamiento de los núcleos por encima de la frecuencia operativa nominal si no se han alcanzado los límites especificados de energía, corriente y temperatura. Esto dificulta aún más la medición, ya que depende de factores externos como la temperatura. En el caso del procesador Core i7-4771 mencionado anteriormente, la frecuencia máxima con Turbo Boost es 3.9 GHz, lo cual representa un incremento de velocidad de un 11.43 %. En procesadores de bajo consumo la diferencia es aún mayor. Por ejemplo, en un procesador Core i5-4200U, la máxima frecuencia operativa nominal es 1.6 GHz, mientras que la frecuencia con Turbo Boost es 2.6 GHz, es decir, un incremento del 62.5 %. Esto implica que para medir adecuadamente es necesario deshabilitar esta tecnología. En una máquina con Linux esto puede hacerse en tiempo de ejecución, modificando el archivo `/sys/devices/system/cpu/cpufreq/boost`. En

una máquina con Windows esto no es posible, por lo que es necesario inhabilitarla previo al arranque, accediendo al setup de la máquina.

- La instrucción `rdtsc` no se serializa u ordena con otras instrucciones. No necesariamente espera a que hayan finalizado otras instrucciones para leer el contador.
- No existe garantía de que todos los procesadores de un sistema multiprocesador o multicore tengan sus contadores sincronizados, por lo que es conveniente ligar cada proceso a un procesador específico. En Linux esto puede realizarse en tiempo de ejecución mediante el comando `taskset`.
- Existe sobrecarga asociada con el proceso de medición, por lo que el algoritmo de medición debe encargarse de calcular y sustraer esta sobrecarga.

El procedimiento de *benchmark* implica por lo tanto:

1. Deshabilitar Turbo Boost
2. Configurar el sistema operativo para trabajar en modo "rendimiento", es decir, que el procesador trabaje siempre a su máxima frecuencia operativa nominal.
3. Ejecutar el programa de medición ligándolo a un procesador usando los mecanismos de afinidad de procesador disponibles en cada sistema operativo.

El pseudocódigo de la figura 1 muestra la estructura del programa de medición.

```
BENCHMARK(M,N,P)
 1  for i = 1 to M
 2      c1 = RDTSC()
 3      for j = 0 to N
 4          P()
 5      c1 = RDTSC() − c1
 6      c2 = RDTSC()
 7      P()
 8      c2 = RDTSC() − c2
 9      A[i] = (c1 − c2)/N
10  return MEDIAN(A)
```

Figura 1. El procedimiento BENCHMARK

M y N son dos constantes utilizadas para parametrizar la medición. P es el procedimiento cuya velocidad queremos medir. Como puede verse en el lazo interno, ejecutamos $N+1$ veces el procedimiento P y medimos el tiempo utilizado. Luego medimos una vez más, y sustraemos ambas medidas. El resultado es el tiempo insumido en ejecutar N veces el procedimiento P, y con la sobrecarga asociada con la medición ya sustraída (asumimos que la sobrecarga de ejecutar una vez es la misma que al ejecutar $N+1$ veces). Repetimos el proceso M veces, almacenando los resultados parciales en un vector A de tamaño M (nótese que en este pseudocódigo, el primer elemento del vector es el 1). Finalmente, devolvemos la mediana de los valores medidos.

3. Las instrucciones AES-NI

Las nuevas instrucciones AES (cuadro 2) fueron introducidas por Intel en el año 2008 y constituyen una extensión del conjunto de instrucciones x86 que tienen el propósito de acelerar las operaciones de cifrado y descifrado con AES.

Cuadro 2. Instrucciones AES-NI

`AESENC`	Ejecuta una ronda de cifrado AES
`AESDEC`	Ejecuta una ronda de descifrado AES
`AESENCLAST`	Ejecuta la última ronda de cifrado AES
`AESDECLAST`	Ejecuta la última ronda de descifrado AES
`AESKEYGENASSIST`	Asiste en la generación de claves de ronda
`AESIMC`	Asiste en la operación Inverse Mix Columns

Estas instrucciones son accesibles mediante lenguaje assembler, o pueden ser utilizadas en C/C++ mediante extensiones provistas por el compilador (*intrinsics*). Los principales compiladores disponibles en la plataforma x86 (GCC, Intel y Microsoft) cuentan con estas extensiones. Por ejemplo, la instrucción `AESENC` puede ser utilizada desde C mediante la siguiente función *intrinsic*:

```
__m128i _mm_aesenc_si128(__m128i a,
                         __m128i RoundKey)
```

cuyo prototipo está declarado en el archivo `wmmintrin.h`. Los argumentos son registros de 128 bits, de tipo `__m128i`. La función ejecuta una ronda de cifrado AES en los datos en `a` usando la clave de ronda en `RoundKey`. Los argumentos deben estar alineados a 128 bits (es decir, en una dirección de memoria múltiplo de 16), por lo que para transferir los datos desde y hacia posiciones no alineadas deben usarse operaciones tales como `_mm_loadu_si128` y `_mm_storeu_si128`. La figura 2 muestra el código en C de una implementación simple de la función de cifrado de un bloque.

Ahora bien, si bien el código de la figura 2 es correcto, no aprovecha plenamente las capacidades de las nuevas instrucciones. Como puede verse en el cuadro 3 la instrucción `AESENC` tiene una latencia de 6 y un throughput de 2 en procesadores Westmere, mientras que

```
void AES_128_encrypt(
  const unsigned char* in,
  unsigned char* out,
  const unsigned char* subkeys){

  int i;
  __m128i subkey = _mm_loadu_si128(
                    (__m128i*) subkeys);
  __m128i output = _mm_xor_si128(
        _mm_loadu_si128((__m128i*)in),subkey);

  for (i = 1; i < ROUNDS; i++){
    subkey = _mm_loadu_si128(
         (__m128i*)(subkeys+(i*BLOCK_SIZE)));
    output = _mm_aesenc_si128(output,subkey);
  }

  subkey = _mm_loadu_si128(
              (__m128i*)(subkeys+
              (ROUNDS*BLOCK_SIZE)));
  output = _mm_aesenclast_si128(output, subkey);
  _mm_storeu_si128((__m128i*) out, output);
}
```

Figura 2. Cifrado de un bloque

esos valores son 7 y 1, respectivamente, en procesadores posteriores. Esto significa que en un procesador Haswell, como por ejemplo el Core i7-4771 antes mencionado, la ejecución de una ronda de AES toma 7 ciclos de reloj, pero es posible iniciar la ejecución de una segunda instrucción AES un ciclo después de comenzada la primera. Si el programa es capaz de introducir instrucciones AES independientes, es decir, en las cuales la entrada de una no dependa de la salida de la anterior, se podrá obtener el throughput prometido, es decir, se ejecutará una instrucción AES por ciclo.

Cuadro 3. Performance de `AESENC`

Arquitectura	Latencia	Throughput
Haswell	7	1
Ivy Bridge	7	1
Sandy Bridge	7	1
Westmere	6	2

Obviamente el código mostrado no cumple con este requerimiento, ya que las rondas deben ejecutarse en forma secuencial. No es posible comenzar la segunda ronda hasta que no haya finalizado la primera. Aún suponiendo que la planificación de instrucciones es perfecta, la ejecución de las 10 rondas de AES-128 tendrá una cota inferior de $10 \times 7 = 70$ ciclos.

La solución es cifrar varios bloques en paralelo. Supongamos que se decide cifrar 10 bloques en paralelo. Comienza la ejecución de la primera ronda del primer bloque. Un ciclo más tarde comienza la primera ronda del segundo bloque. Un ciclo más tarde comienza la primera ronda del bloque siguiente, y así sucesivamente. De esta forma, la ejecución de las 10 rondas independientes tomará 16 ciclos de reloj, lo cual representa una mejora significativa sobre los 70 ciclos del caso anterior.

Esta optimización requiere que el modo de operación sea paralelizable. De los modos convencionales [8] ECB y CTR son completamente paralelizables, CBC y CFB son paralelizables en el descifrado, pero no en el cifrado, y OFB no es paralelizable. En el caso de los algoritmos que nos ocupan, ambos fueron diseñados teniendo en cuenta esta propiedad. Silver es completamente paralelizable, mientras que AES-CPFB es paralelizable en el cifrado, pero no en el descifrado.

4. Benchmarking de la implementación sin instrucciones AES-NI

Los dos algoritmos evaluados poseen una implementación de referencia desarrollada en lenguaje C de acuerdo a los requisitos de la competencia CAESAR y cuya API se muestra en la figura 3. En la implementación de referencia el énfasis está puesto en la claridad y portabilidad del código, y no en su eficiencia, por lo que no es útil para medir la velocidad de ambos algoritmos. Por lo tanto, y para poder comparar adecuadamente los beneficios obtenidos por la utilización de las instrucciones AES-NI usamos otras versiones más optimizadas.

Como AES-CPFB usa AES como caja negra, su velocidad depende de la versión de AES utilizada. Existen múltiples implementaciones disponibles, entre las que se destacan dos:

- La implementación optimizada de Brian Gladman.
- La implementación optimizada en ANSI C desarrollada por Vincent Rijmen, Antoon Bosselaers y Paulo Barreto.

AES-CPFB posee implementaciones basadas en ambas. Las velocidades son muy similares, y el rendimiento relativo depende de la arquitectura. En algunas computadoras es más rápida la versión de Gladman, y en otras ocurre lo contrario. A los efectos de este trabajo, usamos la versión de Rijmen, Bosselaers y Barreto.

En el caso de Silver, es necesario modificar AES, por lo que usamos una versión desarrollada ad-hoc, basada en la implementación de Rijmen, Bosselaers y Barreto.

En los cuadros 4 y 5 puede verse el resultado de correr el programa de benchmarking en las versiones mencionadas de ambos algoritmos. Todos los resultados fueron obtenidos con el procedimiento de benchmarking descripto en la sección 2, y corresponden, salvo

```
#include "crypto_aead.h"

int crypto_aead_encrypt(
  unsigned char *c,unsigned long long *clen,
  const unsigned char *m,unsigned long long mlen,
  const unsigned char *ad,unsigned long long adlen,
  const unsigned char *nsec,
  const unsigned char *npub,
  const unsigned char *k
)
{
 /*
 * the code for the cipher implementation goes here,
 * generating a ciphertext c[0],c[1],...,c[*clen-1]
 * from a plaintext m[0],m[1],...,m[mlen-1]
 * and associated data ad[0],ad[1],...,ad[adlen-1]
 * and secret message number nsec[0],nsec[1],...
 * and public message number npub[0],npub[1],...
 * and secret key k[0],k[1],...
 */
  return 0;
}

int crypto_aead_decrypt(
  unsigned char *m,unsigned long long *mlen,
  unsigned char *nsec,
  const unsigned char *c,unsigned long long clen,
  const unsigned char *ad,unsigned long long adlen,
  const unsigned char *npub,
  const unsigned char *k
)
{
 /*
 * the code for the cipher implementation goes here,
 * generating a plaintext m[0],m[1],...,m[*mlen-1]
 * and secret message number nsec[0],nsec[1],...
 * from a ciphertext c[0],c[1],...,c[clen-1]
 * and associated data ad[0],ad[1],...,ad[adlen-1]
 * and public message number npub[0],npub[1],...
 * and secret key k[0],k[1],...
 */
  return 0;
}
```

Figura 3. API de CAESAR

que se indique lo contrario, a una computadora de escritorio con procesador Intel® Core i7-4771 con Ubuntu 14.04. El código fue compilado con gcc 4.8.2.

Cuadro 4. AES-CPFB sin instrucciones AES-NI

Op.	44	552	576	1500	65536	PP
Enc	51.30	20.14	20.03	18.55	17.64	20.72
Dec	51.44	20.19	20.11	18.54	17.62	20.75

En ciclos por byte

Los cuadros muestran el resultado de cifrar y descifrar paquetes de 44, 552, 576, 1500 y 65536 bytes. La última columna presenta el promedio ponderado de los cuatro primeros tamaños, según los pesos establecidos en el cuadro 1. Los valores indican ciclos por byte, por lo que menores números indican mejor rendimiento.

Puede apreciarse que:

- El costo de AES-CPFB es aproximadamente un 50 % mayor que el de Silver.

Cuadro 5. Silver sin instrucciones AES-NI

Op.	44	552	576	1500	65536	PP
Enc	34.04	13.23	12.56	11.93	11.33	13.35
Dec	42.58	16.47	15.88	14.91	14.10	16.72

En ciclos por byte

- En AES-CPFB no existe diferencia significativa entre las operaciones de cifrado y descifrado.
- En Silver el cifrado es ligeramente más eficiente que el descifrado.
- En paquetes pequeños, la eficiencia merma considerablemente, debido a los costos fijos de expansión de clave.

5. Benchmarking de la implementación con instrucciones AES-NI

Una implementación trivial de AES con las instrucciones AES-NI utiliza un código similar al de la figura 2. Sin embargo, este código no aprovecha las oportunidades de paralelización que brindan las instrucciones AES-NI. Tal como se describe en la sección 3 es posible cifrar varios bloques en paralelo, con un código similar al que se muestra en la figura 4, el cual cifra 8 bloques en paralelo.

```
void AES_128_encrypt(const unsigned char* in,
                     unsigned char* output,
                     const unsigned char* subkeys){
  int i,j;
  __m128i subkey = _mm_loadu_si128(
        (__m128i*) subkeys);
  __m128i out[8];
  for (j = 0; j < 8; j++)
    out[j] = _mm_xor_si128(_mm_loadu_si128(
        (__m128i*)(in + (j*BLOCK_SIZE))),subkey);
  for (i = 1; i < ROUNDS; i++){
    subkey = _mm_loadu_si128(
        (__m128i*)(subkeys+(i*BLOCK_SIZE)));
    for (j = 0; j < 8; j++)
      out[j] = _mm_aesenc_si128(out[j],subkey);
  }
  subkey = _mm_loadu_si128(
        (__m128i*)(subkeys+(ROUNDS*BLOCK_SIZE)));
  for (j = 0; j < 8; j++)
    out[j] = _mm_aesenclast_si128(out[j],subkey);
  for (j = 0; j < 8; j++)
    _mm_storeu_si128(
        (__m128i*) (output+(j*BLOCK_SIZE)), out[j]);
}
```

Figura 4. Cifrado de 8 bloques en paralelo

Si bien las técnicas de optimización que pueden aplicarse en los dos algoritmos son similares, nos concentraremos en los resultados obtenidos con AES-CPFB. En este algoritmo el cifrado es paralelizable, pe-

ro el descifrado no, por lo que nos permite comparar los resultados de distintas optimizaciones en ambos casos. Silver, por el contrario, es completamente paralelizable, por lo que solo mostraremos los resultados finales de la optimización.

En el cuadro 6 pueden verse los resultados obtenidos con una versión relativamente temprana de la implementación de AES-CPFB con instrucciones AES-NI.

Cuadro 6. AES-CPFB con AES-NI

Op.	44	552	576	1500	65536	PP
Enc	19.99	2.79	2.75	1.78	1.30	3.03
Dec	24.84	8.06	7.98	7.16	6.68	8.34

En ciclos por byte

Puede apreciarse que las mejoras de rendimiento son notables. En mensajes largos, el costo del cifrado se ha reducido de 17.6 cpb a 1.3 cpb, es decir, la velocidad se ha multiplicado por un factor superior a 13. En mensajes cortos la mejora es menor, pero también es importante. En mensajes de 44 bytes se ha reducido de más de 50 cpb a menos de 20 cpb.

También es posible apreciar como el cifrado es apreciablemente más rápido que el descifrado. Esto se debe a que por las características del modo de operación, el descifrado no es paralelizable, ya que para descifrar un bloque es necesario haber descifrado el bloque anterior. Como en la implementación sin instrucciones AES-NI los costos de cifrado y descifrado son equivalentes, es claro que las diferencias que se perciben en este caso se deben a la capacidad de paralelización de las nuevas instrucciones.

Esta versión incluye algunas optimizaciones adicionales, tales como una reducción binomial del grado de paralelización para mensajes cortos. Se definieron funciones que cifran en paralelo 8, 4 y 2 bloques. De esta forma, para cifrar un mensaje de 7 bloques, por ejemplo, primero se cifran 4 bloques en paralelo, luego 2 bloques, y finalmente 1 bloque. Esto permite reducir en forma apreciable el tiempo de procesamiento de mensajes cortos, con una pequeña reducción de la eficiencia en mensajes largos.

Sin embargo aún es posible realizar optimizaciones adicionales:

- El algoritmo requiere la generación de al menos dos claves auxiliares mediante el cifrado con la clave maestra de dos *nonces* derivados del número de mensaje público. Es posible paralelizar estas operaciones con la expansión de la clave maestra, en un procedimiento similar al descripto por Shay Gueron en [10] (AES encryption with On-The-Fly Key Expansion). En este caso, a diferencia del planteado por Gueron, se trata de una expansión de clave ejecutada en paralelo con el cifrado de dos bloques.

- Las dos claves auxiliares pueden expandirse en paralelo.[1]

- Las operaciones `load` y `store` de 128 bits, necesarias para transferir datos entre memoria y registros, son más eficientes que las transferencias de bytes, inclusive cuando se trata de bloques incompletos. Es decir, es más rápido transferir 96 bits con una operación de 128 bits, por más que se desperdicien 32 bits, que hacer la transferencia en unidades más pequeñas. Sin embargo, es necesario tener en cuenta que las operaciones mencionadas no salgan de los buffers proporcionados, tanto para lectura como para escritura. Inclusive dentro del mismo buffer, hay que tener en cuenta que los buffers de entrada y salida podrían apuntar a la misma área de memoria, y debe evitarse sobrescribir las partes del mensaje aún no cifradas.

La aplicación de las optimizaciones mencionadas produce los resultados que se muestran en el cuadro 7.

Cuadro 7. AES-CPFB con AES-NI (final)

Op.	44	552	576	1500	65536	PP
Enc	13.09	2.01	1.92	1.44	1.11	2.21
Dec	18.07	7.04	7.01	6.47	6.16	7.24

En ciclos por byte

El cuadro 8 muestra los resultados correspondientes a Silver.

Cuadro 8. Silver con AES-NI (final)

Op.	44	552	576	1500	65536	PP
Enc	11.85	1.64	1.41	1.10	0.76	1.78
Dec	15.08	2.07	1.80	1.37	0.95	2.25

En ciclos por byte

6. Otras plataformas

A continuación se presentan los resultados de correr el mismo código en distintas plataformas.

[1] En rigor no siempre es necesaria la generación y expansión de las dos claves. Sin embargo, esto sólo ocurre en el infrecuente caso de mensaje nulo, y por otro lado la paralelización de las dos operaciones hace que su costo no sea sustancialmente superior al de generar y expandir una sola clave.

Intel® Core i7-3770k

El procesador Intel® Core i7-3770k (Ivy Bridge) es una generación anterior al Core i7-4771 (Haswell) de las pruebas precedentes. La frecuencia nominal, 3.5 GHz, es la misma, por lo que las diferencias de rendimiento son atribuibles a la diferencia en la arquitectura.

Cuadro 9. AES-CPFB sin AES-NI, Ivy Bridge

Op.	44	552	576	1500	65536	PP
Enc	57.37	22.68	22.57	20.90	19.86	23.33
Dec	57.45	22.65	22.52	20.84	19.82	23.28

En ciclos por byte

Cuadro 10. AES-CPFB con AES-NI, Ivy Bridge

Op.	44	552	576	1500	65536	PP
Enc	12.78	2.19	2.01	1.66	1.32	2.37
Dec	18.12	7.78	7.74	7.25	6.96	7.97

En ciclos por byte

Cuadro 11. Silver sin AES-NI, Ivy Bridge

Op.	44	552	576	1500	65536	PP
Enc	37.06	14.48	13.81	13.16	12.51	14.68
Dec	46.39	17.97	17.31	16.29	15.39	18.25

En ciclos por byte

Cuadro 12. Silver con AES-NI, Ivy Bridge

Op.	44	552	576	1500	65536	PP
Enc	13.25	1.89	1.63	1.31	0.91	2.06
Dec	15.65	2.47	2.21	1.77	1.33	2.66

En ciclos por byte

Como puede verse en los cuadros 9 a 12, el rendimiento en Ivy Bridge es consistentemente inferior al rendimiento en Haswell.

Intel® Core i5-4200U

El procesador Intel® Core i5-4200U es de la misma generación que el Core i7-4771 (Haswell). Es un procesador de bajo consumo, utilizado habitualmente en ultrabooks, y tiene una frecuencia nominal de 1.6 GHz.

Los benchmarks en este procesador pueden verse en los cuadros 13 a 16.

Cuadro 13. AES-CPFB sin AES-NI, i5-4200U

Op.	44	552	576	1500	65536	PP
Enc	74.05	29.33	29.17	26.89	25.44	30.07
Dec	74.66	29.22	29.06	26.75	25.42	29.98

En ciclos por byte

Cuadro 14. AES-CPFB con AES-NI, i5-4200U

Op.	44	552	576	1500	65536	PP
Enc	18.82	2.90	2.76	2.09	1.60	3.18
Dec	25.98	10.15	10.07	9.31	8.86	10.42

En ciclos por byte

Cuadro 15. Silver sin AES-NI, i5-4200U

Op.	44	552	576	1500	65536	PP
Enc	48.39	18.61	17.73	16.89	16.10	18.89
Dec	61.86	23.60	22.78	21.44	20.20	24.05

En ciclos por byte

Cuadro 16. Silver con AES-NI, i5-4200U

Op.	44	552	576	1500	65536	PP
Enc	16.98	2.36	2.01	1.58	1.09	2.56
Dec	21.67	2.99	2.59	1.97	1.37	3.23

En ciclos por byte

7. Conclusiones

La experiencia obtenida en este trabajo de medición y optimización nos permite concluir que:

- Medir adecuadamente el rendimiento de algoritmos criptográficos es una tarea no trivial, ya que existen numerosas oportunidades de realizar mediciones erróneas. No sólo es necesario encontrar la métrica adecuada, sino que las características de los procesadores modernos, tales como ejecución fuera de orden, multiplicidad de núcleos y frecuencia dinámica afectan el proceso de medición.
- Los algoritmos criptográficos que puedan utilizar las nuevas instrucciones AES-NI se verán altamente beneficiados, aún en el caso en que no sean paralelizables. Estos beneficios no se reducen a incrementos de velocidad. La seguridad de los algoritmos se ve mejorada a través de la protección contra *side channels attacks* (como por ejemplo los *cache timing attacks* a los que ha sido sometido AES).
- Si además de poder usar estas instrucciones el algoritmo puede paralelizarse, las mejoras de rendimiento son todavía mayores. Los números presentados en este trabajo hablan por sí solos.

- Distintas optimizaciones funcionan de manera distinta en diferentes plataformas. Las estrategias de optimización de los distintos compiladores hacen que la optimización que mejora el rendimiento en uno de ellos puede empeorarlo con otro.

8. Bibliografía

Referencias

[1] "Specification for the Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197, 2001

[2] Joan Daemen y Vincent Rijmen. "AES Proposal: Rijndael". National Institute of Standards and Technology, 1999.

[3] Joan Daemen and Vincent Rijmen, "The Design of Rijndael, AES - The Advanced Encryption Standard", Springer-Verlag 2002 (238 pp.)

[4] Daniel Penazzi y Miguel Montes, "Silver v1", CAESAR website, 2014

[5] Miguel Montes y Daniel Penazzi, "AES-CPFB v1", CAESAR website, 2014

[6] David McGrew y John Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation", In INDOCRYPT, volume 3348 of LNCS, 2004

[7] P. Rogaway, M. Bellare, J. Black, y T. Krovitz, "OCB: a block-cipher mode of operation for efficient authenticated encryption", ACM CCS, 2001

[8] "Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A, 3B & 3C): System Programming Guide"

[9] Morris Dworking, "NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation", 2001 Edition

[10] David McGrew y John Viega, "The Galois/Counter Mode of Operation (GCM)", NIST website,

[11] Shay Gueron, "Intel® Advanced Encryption Standard (AES) New Instructions Set", Intel Corporation, Mayo 2010.

On the Interleaving Process Applied to the Trivium Algorithm

Antonio Castro Lechtaler[1, 2], Marcelo Cipriano[1], Edith García[1], Julio Liporace[1], Ariel Maiorano[1], Eduardo Malvacio[1],

[1] Laboratorio de Criptografía y Seguridad Teleinformática – Escuela Superior Técnica – IUE, Ciudad Autónoma de Buenos Aires, C1426; [2] Universidad Nacional de Chilecito, Chilecito, F5360, Provincia de La Rioja. República Argentina.

[1]{acastro, marcelocipriano}@iese.edu.ar
[2]{edithxgarcia, jcliporace, maiorano, edumalvacio}@gmail.com

Abstract

Interleaving processes with different orders are applied to the Trivium linear subgenerator. Generated sequences are analyzed. The procedure is shown to originate weaknesses in the generator, increasing the occurrence of short cycles. The application of interleaving is found to be inadequate to increase the subgenerator linear complexity and the key space.

Keywords: Trivium, linear subgenerator, interleaving, cycles.

1. Introduction

Pseudorandom sequences with high linear complexity and large periods are widely used in cryptology and communications. The development of algorithms to generate such bit sequences is a significantly complex endeavor. Each part of the sequence needs to be thoroughly analyzed to meet the required cryptologic properties of the algorithm as a whole.

Trivium is a stream cipher designed to generate up to 2^{64} bits of key stream [2]. The cipher uses an 80 bit secret key and an initialization vector of also 80 bits (160 bits total). Both, key and vector comprise the initial state of the generator. The fundamental principles of the design aim at the creation of a linear subgenerator, determining the algorithm strength. The structure is completed with an interleaving process of order 3 with the addition of AND operations. An open problem in the analysis consists of the period determination. A constraint is established with the linear subgenerator. However, it cannot be determined if the interleaving process as well as the AND operations are introduced.

The following section defines the interleaving process. Then, it is applied to a linear subgenerator, using different orders: 1, 2, 3, and 4. The short-cycle sequences obtained are presented for each case.

The authors of Trivium used an interleaving process of order 3. Consequently, they increase the linear complexity and the key space. However, their work does not explain the rationale behind its application.

The present work analyzes each of these processes and calculates the number of short-cycle sequences obtained.

2. Overview

2.1. FSR

Let $f(x) = x^n + c^{n-1}x^{n-1} + ... + c^1x+1$ be a polynomial over GF(2). A sequence $s=\{s_i\}$ is called *FSR sequence*

generated for f(x) if it satisfies the following recurrence linear relation:

$$s_{n+k} = \sum_{i=0}^{n-1} c_i\, s_{k+i} \qquad k = 0; 1; 2; \ldots \qquad (1)$$

f(x) is called the *characteristic polynomial* of the sequence *s*. The array $(s_0,s_1,\ldots, s_{n-1})$ is the initial state of the FSR generating *s*. If *f(x)* is a primitive polynomial, the series is defined as an *m-sequence* [6].

The *minimal polynomial* of *s* is the least degree polynomial generated by *s*. If *m(x)* is the minimal polynomial of *s*, then *m(x)* divides *f(x)*.

The linear complexity of *s* (*LC(s)*) is the degree of the *m(x)*. In general, *m(x)* is found with the Belekamp-Massey algorithm, using *2LC(s)* consecutive bits of *s*.

For any *f(x)* over *GF(2)[x]* with $f(0)\neq 0$, the *order of f(x)* is defined as the least integer $e\geq 1$ such that $f(x) \mid x^e + 1$.

The period of a sequence *s* is equal to the degree of its minimal polynomial. It is the least integer *p* such that $s_n=s_{n+p}$ for every $n\geq 0$. The array $(s_0;s_1;\ldots;s_{p-1})$ is called *cycle of the sequence s* and its length is equal to *p*.

2.2 Interleaved Sequences

Let *f(x)* be a polynomial over *GF(2)* and *m* a positive integer. For every sequence $\boldsymbol{s}=s(k)$ over *GF(2),* we write $k=im+j$ (i=0,1,2,…) (j=0, 1,…,m-1). If for every *j*, $s_j=\{s(im+j)\}_{i\geq 0}$ is generated for *f(x)*, then *s* is called the *interleaved sequence* over *GF(2)* of order *m* associated to *f(x)*.

We can write $s=(s_0, s_1,\ldots, s_{m-1})$ where each $\boldsymbol{s}_j$ *(j=0,1,...,m-1)* is a subsequence of *s*. Each s_j is a decimation of order *m* obtained from *s*, sampling every *m* bits.

For example: The interleaved sequence s of order *m=4* associated to the polynomial $f(x)=x^3+x+1$

$\boldsymbol{s}_0$=1,1,1,0,0,1,0,…
$\boldsymbol{s}_1$=1,0,0,1,0,1,1,…
$\boldsymbol{s}_2$=0,1,0,1,1,1,0,…
$\boldsymbol{s}_3$=0,1,1,1,0,0,1,…

$\boldsymbol{s}$ =1,1,0,0,1,0,1,1,1,0,0,1,0,1,1,1,...

2.3 Interleaved FSR

Let *FSR* be a linear generator defined by a polynomial *f(x)* over *GF(2)* and *m* be a positive integer. Varying *m* initial possible states of the FSR yields the sequences $\boldsymbol{s}_j$ *(j=0,…,m-1)*. The FSR interleaving process of order *m* of *f(x)* is defined to be the sequence $s=(s_0,s_1,\ldots,s_{m-1})$

Example: Interleaving process of order *m=4*

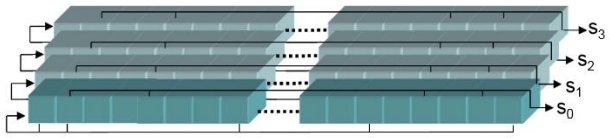


Figure 1

	$\boldsymbol{s}_0$	$\boldsymbol{s}_1$	$\boldsymbol{s}_1$	$\boldsymbol{s}_3$
0	1	1	0	0
1	1	0	1	1
2	1	0	0	1
3	0	1	1	1
4	0	0	1	0
5	0	1	1	0
6	1	1	0	1

Table 1: Example of an interleaving process of order 4

$s= s_0(0),s_1(0),s_2(0),s_3(0),s_0(1),s_1(1),s_2(1),s_3(1),\ldots$

$s= 1,1,0,0,1,0,1,1,1,0,0,1,0,1,1,\ldots.$

The following lemmas derive from the interleaving process:

Lemma 1: if s_j is generated by *f(x)*, then the interleaved sequence *s* is generated by $f(x^m)$.

Lemma 2: If the linear complexity of each sequence s_j is *L*, then the linear complexity of the interleaved sequence is *mL*.

Lemma 3: If the period of s_j is *p*, then the period of the interleaved sequence s is mp. [13]

2.4 Finite Fields

The following theorems apply to the different characteristic polynomials resulting from each interleaving process. In this manner, the sequences with minimal cycles are obtained.

Theorem 1: Let $f(x) = \prod_i f_i^{b_i}$ where $f_i(x)$ are distinct irreducible polynomials over *GF(2)* and b_i are positive integers. Then

$$S\big(f(x)\big) = S(f_1(x)^{b_1}) + \cdots + S(f_n(x)^{b_n}) \qquad (2)$$

where $S(f_1(x)^{b_1}) + S(f_2(x)^{b_2}) + \cdots + S(f_n(x)^{b_n})$ is defined to be the set of all sequences $s_1+ s_2+\ldots+ s_n$ with $s_i \in S(f_i(x)^{b(i)})$.

Theorem 2: For each i = 1, 2,…, n let s_i be a linear recurring sequence in *GF(2)* with a minimal polynomial $f_i(x) \in GF(2)[x]$ and a least period p_i. If the polynomials $f_1(x), f_2(x), ..., f_n(x)$ are pairwise relatively prime, then the least period of $s_1 + s_2 + \ldots\ldots + s_n$ is equal to the least common multiple of $p_1, p_2, ..., p_n$.

Theorem 3: Let $f(x) = (g(x))^b$ with $g(x) \in GF(2)[x]$ irreducible over *GF(2)* $g(0) \neq 0$, degree $(g(x)) = k$, *order* $(g(x)) = e$ and b a positive integer. Let t be the smallest integer with $2^t \geq b$. Then *S(f(x))* contains the following number of sequences and least periods: one sequence with least period *1*, 2^k-1 sequences with least period e; and for $b \geq 2$, $2^{2^{j}k} - 2^{2^{j-1}k}$ sequences with least period $e*2^j$ *(j=1;,.;t-1),* and $2^{kb} - 2^{2^{t-1}k}$ sequences with least period $e*2^t$.

3. Trivium

The stream algorithm TRIVIUM was designed by *Christophe De Cannière* and *Bart Preneel* to generate at least 2^{64} bits, using a *80-bit* secret key and an initialization vector (IV) of also 80 bits which determine part of its initial state. [5] [2]

It consists of three combined NLFSRs. [3][4] The first register controls the second, the second controls the third, and the last controls the first.

The core idea behind the design focuses on using the principles of block cipher construction to create equivalent components in stream ciphers.

Three parts are easily identified in the generator design:

1. A linear part resulting from a linear subgenerator of size 96 and consisting of three feedforward and feedback linear registers. [14]

The resulting characteristic polynomial is given by:

$$p(x) = x^{96} + x^{73} + x^{70} + x^{67} + x^{47} + x^{44} + x^{41} + x^{29} + x^{24} + x^{20} + x^{18} + x^{15} + x^{14} + x^{9} + x^{5} + 1 \quad (3)$$

2. An interleaving of order m=3 of the linear subgenerator (every third element). [13]

3. A nonlinear part resulting from the inclusion of AND operations to the linear Trivium. [1]

The output consists of three combined non-linear shift registers of lengths 93, 84, and 111 in which specific positions are selected to obtain a key bit stream. Whereas no efficient attack has broken the generator so far, its period remains undetermined, remaining as an unsolved problem open for research. [8, 9, 10]

A complete description is given by the following simple pseudo-code:

```
INPUT: s0, s1,..,s287 initial state,
integer n., si {0,1}.
OUTPUT: binary sequence {kt}
    1.Initialization.
    1.1 t1 ← s65 ⊕ s92
    1.2 t2 ← s161 ⊕ s176
    1.3 t3 ← s242 ⊕ s287
  2. While ( t<n ) do the following:
        2.1 kt ←t1⊕ t2 ⊕ t3
        2.2 t1 ←t1 ⊕ s90 ⊗s91 ⊕ s170
        2.3 t2 ←t2 ⊕ s174 ⊗ s175 ⊕ s263
        2.4   t3 ←t3 ⊕ s285 ⊗ s286 ⊕
    s68
        2.5
    (s0;s1;...;s92)←(t3;s0;..;s91)
        2.6
    (s93;s94;..;s176)←(t1;s93;..;s175)
        2.7
    (s177;s178;..;s287)←(t2;s177;..;s285)
    3. Return {kt}
```

Note that ⊕ is the XOR operation and ⊗ the AND operation.

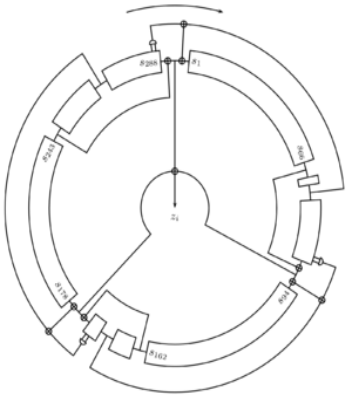


Figure 3: Trivium algorithm

The linear Trivium algorithm follows the same procedure with the exception of the omitted AND operations. The terms $s_{90} \otimes s_{91}$, $s_{174} \otimes s_{175}$ and $s_{285} \otimes s_{286}$ are eliminated.

4. Interleaving process of order m

In this section, an interleaving process applied to the linear Trivium is presented for the orders m = 1, 2, 3, and 4. For every case, the characteristic polynomial is shown

and factorized. Finally, the minimum cycles of the generated sequences are determined.

4.1 For *m = 1*

This case corresponds to the analysis of the characteristic polynomial defined by the subgenerator of expression (3):

$$p(x) = q(x) * r(x) \quad (4)$$

$$q(x) = (x+1)^3 \quad (5)$$

$$\begin{aligned} r(x) = x^{93} + x^{92} + x^{89} + x^{88} + x^{85} + x^{84} \\ + x^{81} + x^{80} + x^{77} + x^{76} \\ + x^{73} + x^{72} + x^{70} + x^{68} \\ + x^{67} + x^{44} + x^{43} + x^{41} \\ + x^{39} + x^{38} + x^{35} + x^{34} \\ + x^{31} + x^{30} + x^{27} + x^{25} \\ + x^{23} + x^{20} + x^{19} + x^{17} \\ + x^{14} + x^{13} + x^{12} + x^9 + x^8 \\ + x^6 + x^4 + x + 1 \end{aligned} \quad (6)$$

The polynomial *p(x)* is not irreducible. Hence, particular sequences are obtained from the different initial states, with distinct periods or cycles as well. Theorems 1 to 3 are applied to each factor to calculate them.

The polynomial $q(x) = (x+1)^3$ – a factor of equation (4) - yields k = 1 (the degree of x+1), e = 1 (the order of x+1), b = 3, t = 2. Applying theorems 1 to 3, the results shown in the following table are obtained:

Number of Sequences	Least Period
2	1
2	2
4	4

Table 2: Number of sequences and least periods for *q(x)*

For the primitive polynomial *r(x)* of equation (7), the trivial null sequence is obtained and the rest of all possible sequences of maximum length are shown in table 3.

Number of Sequences	Least Period
1	1
2^{93}-1	2^{93}-1

Table 3: Number of sequences and least periods for *r(x)*

Hence, *p(x)* yields:

Number of Sequences	Least Period
2	1
2	2
4	4
$2*(2^{93}-1)$	$2^{93}-1$
$2*(2^{93}-1)$	$2*(2^{93}-1)$
$4*(2^{93}-1)$	$4*(2^{93}-1)$

Table 4: Number of sequences and least periods for *p(x)*

Note that eight short period sequences are obtained – of lengths 1, 2, and 4 bits. Thus, the linear subgenerator may prove to produce **few** sequences with short cycles of lengths 1, 2, and 4.

4.2 For *m = 2*

An interleaving process of order *m* = 2 is applied to the characteristic polynomial of the linear subgenerator in (3):

$$\begin{aligned} p(x^2) = x^{192} + x^{146} + x^{140} + x^{134} + x^{94} \\ + x^{88} + x^{82} + x^{58} + x^{48} \\ + x^{40} + x^{36} + x^{30} + x^{28} \\ + x^{18} + x^{10} + 1 \end{aligned} \quad (7)$$

$$p(x^2) = q(x) * r(x) \quad (8)$$

$$q(x) = (x+1)^6 \quad (9)$$

$$\begin{aligned} r(x) = (x^{93} + x^{92} + x^{89} + x^{88} + x^{85} + x^{84} \\ + x^{81} + x^{80} + x^{77} + x^{76} \\ + x^{73} + x^{72} + x^{70} + x^{68} \\ + x^{67} + x^{44} + x^{43} + x^{41} \\ + x^{39} + x^{38} + x^{35} + x^{34} \\ + x^{31} + x^{30} + x^{27} + x^{25} \\ + x^{23} + x^{20} + x^{19} + x^{17} \\ + x^{14} + x^{13} + x^{12} + x^9 + x^8 \\ + x^6 + x^4 + x + 1)^2 \end{aligned} \quad (10)$$

Formula (8) shows that the characteristic polynomial *p(x)* of the interleaved linear subgenerator of order m = 2 is not irreducible. Thus, different initial states yield distinct least periods or cycles. Theorems 1 to 3 are applied to obtain the following values:

Given it is not primitive, q(x) yields from (9): k = 1, e = 1 (the order of x+1 is 1), b = 6, t = 3. The following table details the obtained results:

Number of Sequences	Least Period
2	1
2	2
2^4-2^2	4
2^6-2^4	8

Table 5: Number of sequences and least periods for *q(x)*

r(x) from (10) yields: k = 93, (the order of $x^{93}+x^{92}+x^{89}+x^{88}+x^{85}+x^{84}+x^{81}+x^{80}+x^{77}+x^{76}+x^{73}+x^{72}+x^{70}+x^{68}+x^{67}+x^{44}+x^{43}+x^{41}+x^{39}+x^{38}+x^{35}+x^{34}+x^{31}+x^{30}+x^{27}+x^{25}+x^{23}+x^{20}+x^{19}+x^{17}+x^{14}+x^{13}+x^{12}+x^9+x^8+x^6+x^4+x+1$ is 2^{93} -1 because it is a primitive polynomial), b = 2, t = 1. The results obtained are detailed in the following table:

Number of Sequences	Least Period
1	1
2^{93}-1	(2^{93}-1)
2^{186}-2^{93}	(2^{93}-1)*2

Table 6: Number of sequences and least periods for r*(x)*

Finally, for $p(x^2)$*:*

Number of Sequences	Least Period
2	1
2	2
12	4
48	8
2*(2^{93}-1)	2^{93}-1
2 *(2^{93}-1) * (2^{94}+1)	2*(2^{93}-1)
12 *(2^{93}-1) * (2^{93}+1)	4*(2^{93}-1)
48 *(2^{93}-1) * (2^{93}+1)	8* (2^{93}-1)

Table 7: Number of sequences and least periods for $p(x^2)$

In this case, the number of sequences with minimal short periods has increased – lengths *1, 2, 4,* and *8.*

4.3. For *m = 3*

An interleaving process of order *m = 3* is applied to the characteristic polynomial of the linear subgenerator in (3):

$$p(x^3) = x^{288}+x^{219}+x^{210}+x^{201}+x^{141}+x^{132}+x^{123}+x^{87}+x^{72}+x^{60}+x^{54}+x^{45}+x^{42}+x^{27}+x^{15}+1 \quad (11)$$

$$p(x^3) = q(x) * s(x) * t(x) * u(x) \quad (12)$$

$$q(x)=(x+1)^3 \quad (13)$$

$$s(x)=(x^2+x+1)^3 \quad (14)$$

$$\begin{aligned} t(x) = {} & x^{93}+x^{90}+x^{87}+x^{86}+x^{84}+x^{83} \\ & +x^{82}+x^{81}+x^{80}+x^{79} \\ & +x^{78}+x^{77}+x^{74}+x^{72} \\ & +x^{71}+x^{70}+x^{67}+x^{65} \\ & +x^{63}+x^{62}+x^{51}+x^{44} \\ & +x^{41}+x^{38}+x^{35}+x^{34} \\ & +x^{31}+x^{29}+x^{27}+x^{25} \\ & +x^{24}+x^{21}+x^{19}+x^{17} \\ & +x^{16}+x^{15}+x^{11}+x^9 \\ & ++x^8+x^6+x^5+x+1 \end{aligned} \quad (15)$$

$$\begin{aligned} u(x) = {} & x^{186}+x^{180}+x^{179}+x^{175}+x^{174} \\ & +x^{173}+x^{172}+x^{167}+x^{166} \\ & +x^{164}+x^{163}+x^{162}+x^{161} \\ & +x^{160}+x^{158}+x^{157}+x^{152} \\ & +x^{151}+x^{150}+x^{149}+x^{148} \\ & +x^{147}+x^{144}+x^{142}+x^{141} \\ & +x^{140}+x^{139}+x^{138}+x^{135} \\ & +x^{130}+x^{129}+x^{127}+x^{124} \\ & +x^{121}+x^{120}+x^{116}+x^{115} \\ & +x^{113}+x^{110}+x^{109}+x^{107} \\ & +x^{101}+x^{100}+x^{98}+x^{96} \\ & +x^{95}+x^{91}+x^{90}+x^{88} \\ & +x^{86}+x^{80}+x^{78}+x^{75}+x^{74} \\ & +x^{71}+x^{70}+x^{69}+x^{66} \\ & +x^{64}+x^{61}+x^{58}+x^{53} \\ & +x^{52}+x^{50}+x^{48}+x^{46} \\ & +x^{45}+x^{44}+x^{43}+x^{42} \\ & +x^{41}+x^{40}+x^{39}+x^{38} \\ & +x^{32}+x^{31}+x^{29}+x^{26} \\ & +x^{22}+x^{21}+x^{19}+x^{17} \\ & +x^{16}+x^9+x^8+x^7+x^6 \\ & +x^5+x^2+x+1 \end{aligned} \quad (16)$$

For *q(x)* in (13), the same values of table 2 are obtained. For the polynomial $s(x) = (x^2 + x + 1)^3$ from (14), the following values yield:

Number of Sequences	Least Period
1	1
3	3
12	6
48	12

Table 8: Number of sequences and periods for s*(x)*

For the primitive polynomial *t(x)* of equation (15), the following table is obtained:

Number of Sequences	Least Period
1	1
2^{93}-1	2^{93}-1

Table 9: Number of sequences and periods for *t(x)*

And, for u*(x)* from (16) - irreducible but not primitive:

Number of Sequences	Least Period
1	1
2^{186} -1	3*(2^{93}-1)

Table 10: Number of sequences and periods for u*(x)*

Finally, the characteristic polynomial $p(x^3)$ of the linear Trivium obtained yields the following values:

Number of Sequences	Least Period
2	1
2	2
6	3
4	4
54	6
444	12
2*(2^{93}-1)	(2^{93}-1)
2*(2^{93}-1)	2*(2^{93}-1)
8a+6b+8ab	3*(2^{93}-1)
4*(2^{93}-1)	4*(2^{93}-1)
56a+54b+56ab	6*(2^{93}-1)
448a+636b+256ab	12*(2^{93}-1)

Table 11: Number of sequences and periods for $p(x^3)$

Note: for clarity, values have been replaced by a = $(2^{186}-1)$ and *b* = $(2^{93}-1)$

In the case of the linear Trivium, note the existence of 512 short cycle sequences and 444 sequences originating cycles of size 12. Note that from 8 short period sequences, interleaving by an order $m = 3$ (linear Trivium) yields 512 short cycle sequences, showing an increase of weak initial states in the linear Trivium.

4.4. For $m = 4$

An interleaving process of order $m = 4$ is applied to the characteristic polynomial *p(x)* of the linear subgenerator from (3):

$$\begin{aligned} p(x^4) = x^{384} + x^{292} + x^{280} + x^{268} \\ + x^{188} + x^{176} + x^{164} \\ + x^{116} + x^{96} + x^{80} + x^{72} \\ + x^{60} + x^{56} + x^{36} + x^{20} + 1 \end{aligned} \quad (17)$$

Factors:

$$p(x^4) = q(x) * r(x) \quad (18)$$

$$q(x) = (x+1)^{12} \quad (19)$$

$$\begin{aligned} r(x) = (x^{93} + x^{92} + x^{89} + x^{88} + x^{85} + x^{84} \\ + x^{81} + x^{80} + x^{77} + x^{76} \\ + x^{73} + x^{72} + x^{70} + x^{68} \\ + x^{67} + x^{44} + x^{43} + x^{41} \\ + x^{39} + x^{38} + x^{35} + x^{34} \\ + x^{31} + x^{30} + x^{27} + x^{25} \\ + x^{23} + x^{20} + x^{19} + x^{17} \\ + x^{14} + x^{13} + x^{12} + x^9 + x^8 \\ + x^6 + x^4 + x + 1)^4 \end{aligned} \quad (20)$$

q(x) from (19), given that it is not primitive, yields: k = 1, e = 1 (the order of x+1 is 1), b = 12, t = 4. The following table details the obtained results:

Number of Sequences	Least Period
2	1
2	2
2^4-2^2	4
2^8-2^4	8
2^{12}-2^8	16

Table 12: Number of sequences and periods for *q(x)*

r(x) yields: k = 93, (the order of $x^{93} + x^{92} + x^{89} + x^{88} + x^{85} + x^{84} + x^{81} + x^{80} + x^{77} + x^{76} + x^{73} + x^{72} + x^{70} + x^{68} + x^{67} + x^{44} + x^{43} + x^{41} + x^{39} + x^{38} + x^{35} + x^{34} + x^{31} + x^{30} + x^{27} + x^{25} + x^{23} + x^{20} + x^{19} + x^{17} + x^{14} + x^{13} + x^{12} + x^9 + x^8 + x^6 + x^4 + x + 1$ is 2^{93}-1), b = 4, t = 2. Thus, the following table shows the obtained results:

Number of Sequences	Least Period
1	1
2^{93}-1	2^{93}-1
2^{186}-2^{93}	2*(2^{93}-1)
2^{372}-2^{186}	4*(2^{93}-1)

Table 13: Number of sequences and periods for *r(x)*

The resulting characteristic polynomial $p(x^4)$ yields the following values:

Number of Sequences	Least Period
2	1
2	2
12	4
240	8
3840	16
$(2^{93}-1) * 2$	$2^{93}-1$
$2* (2^{93}-1) *(2^{94}+1)$	$2*(2^{93}-1)$
$(2^{93}-1) *(2^{93}+1)*(2^{190}+12)$	$4*(2^{93}-1)$
$240*(2^{93}-1) *(2^{93}+1)*(2^{186}+1)$	$8*(2^{93}-1)$
$3840*(2^{93}-1) *(2^{93}+1)*(2^{186}+1)$	$16*(2^{93}-1)$

Table 14: Number of sequences and short periods for $p(x^4)$

In this case, the number of sequences with short minimal periods has increased (1, 2, 4, 8, and 16).

6. Conclusions

Interleaving processes of orders 1, 2, 3 (linear Trivium), and 4 were applied to a linear subgenerator of the Trivium algorithm. In each case, a significant increase in the number of sequences with short cycles was observed.
Thus, it can be concluded that the interleaving process for the linear Trivium increases the number of initial states producing sequences with short cycles.

The interleaving process of order *m* increases the key space from 2^{96} to $2^{96.m}$, the linear complexity from 96 to 96m, and the period from $(2^{93}–1)$ to $m(2^{93} – 1)$. The significant increase in the initial states producing short cycles affects the cryptologic strength of the generator.

An improved alternative might include an additional register and a different linear filter, observing the correlation properties and branch number established in the original design.

7. Further Research

Further work shall explore AND operations in the generators, analyzing them as NLFSR or as the combination of linear filters (feedforward and feedback) with non-linear inputs. The authors of the stream cipher Trivium restricted their scope to linear expressions. Advancing their analysis to more complex forms may prove a reasonable direction to pursue.

8. References

[1] De Canniére, C. and Preneel, B. “TRIVIUM A Stream Cipher Construction Inspired by Block Cipher Design Principles”. In Workshop on Stream Ciphers Revisited (SASC2006), 2006.

[2] De Canniére, C. and Preneel, B. “TRIVIUM Specifications”. eSTREAM, ECRYPT Stream Cipher Project, Report. 2008.

[3] Dubrova, E. “A List of Maximum-Period NLFSRs”**,** Cryptology ePrint Archive, Report 2012/166, March 2012, http://eprint.iacr.org/2012/166

[4] Dubrova, E. “A Scalable Method for Constructing Galois NLFSRs with Period $2^n -1$ Using Cross-joint Pairs”. Technical Report 2011/632, Cryptology ePrint Archive, November 2011. http://eprint.iacr.org/2011/632.

[5] eSTREAM: eSTREAM – The ECRYPT Stream Cipher Project: http://www.ecrypt.eu.org/stream/

[6] Golomb. “Shift Register Sequences”. Aegean Park Press, 1982.

[7] Massey, J.L. “Shift-register Synthesis and BCH Decoding”. IEEE Transactions on Information Theory 15, 1969.

[8] Maximov, A. and Biryukov, A. “Two Trivial Attacks on Trivium”, Selected Areas in Cryptography, Lecture Notes in Computer Science, Vol.4876, Springer, 2007.

[9] McDonald, C. and Pieprzyk, C. “Attacking Bivium with MiniSat”, Cryptology ePrint Archive,Report 2007/040, 2007.

[10] Raddum, H.“Cryptanalytic Results on Trivium”, eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006.

[11] Yun Tian, Gongliang Chen, Jianhua Li: “On the Design of Trivium”. IACR Cryptology ePrint Archive 2009.

[12] Lidl R., Niederreiter H. “Introduction to Finite Fields and their Applications”. Cambridge University Press, 1986.

[13] Gong, Guang. “Theory and Applications of q-ary Interleaved Sequences”. IEEE Transactions on Information Theory. Vol. 41 No. 2. March 1995.

[14] Castro Lechtaler, A.; Cipriano, M.; García, E.; Liporace, J.; Maiorano, A.; Malvacio, E.“Model Design for a Reduced Variant of a Trivium Type Stream Cipher” In XVIII Congreso Argentino de Ciencias de la Computación, ISBN. 978-987-23963-1-2. Pg. 1483-1491. Mar del Plata, Argentina, 2013.

Implementation and Analysis of Opensource Honeypots Solutions

Sebastián Norberto Mussetta
Dpto Ing. en Sistemas de Información
UTN – Facultad Regional Villa María
smussetta@frvm.utn.edu.ar

Norberto Gaspar Cena
Dpto Ing. en Sistemas de Información
UTN – Facultad Regional Villa María
ngcena@frvm.utn.edu.ar

Ignacio Daniel Favro
Dpto Ing. en Sistemas de Información
UTN - Facultad Regional Villa María
idfavro@frvm.utn.edu.ar

Abstract

The work was based on the implementation of a Honeypot[1] at a Regional Faculty belonging to Universidad Tecnológica Nacional in a controlled environment to carry out an analysis of the different computer security incidents generated. The use of Honeypots in the network is based on the idea of simulating applications with problems and record all events produced by the attacker, in this way, the network administrator can learn about the different types of attacks to protect the systems of the organization that are in production. The work was carried out with free software tools. The results were very significant in terms of the quantity and types of security incidents reported by the Honeypot. In addition, the development of this project allowed a study of attacks based on the protocols used, addresses, ports, time of events, source addresses of the incidents, among others, resulting this information of vital importance for the implementation of computer security policies on the local network.

Keywords

Security, computer science, incident, honeypot, networks, services, and to detection, vulnerability, attack, intrusion, Dionaea.

Introduction

This work is the continuation of a project whose main focus is on networking administration and the development of strategies to improve IT security.
ICTs, information technology and communications comprise the elements and techniques used in the treatment and transmission of information, mainly digital, Internet and telecommunications; due to its fast development, the value of information has become increasingly important, a valuable asset for organizations of any kind. Using different technologies of information and communications, people around the world have more and more access to information of the organizations often automatically from inside and outside organizations.[1] Users use different devices such as Smartphones, Notebooks, or computer hardware desktops to access information via computer networks. This information is hosted on servers that are usually found within organizations. Services such as Web,[2] Ftp[3], Email[4], e.g. collaborative work spaces, Moodle[5], among others, are the different network services running on servers and are accessed by users from software clients locally or remotely, with the implementation of different network protocols such as http, ftp, smtp, pop3, imap, among others. Network administrators should set computer security as a priority in the developments of its activities since information systems often fail due to the exploitation of vulnerabilities in the information technology infrastructure of organizations and these compromise the security of the information thereof. Network administrators are people in the area of networking responsible for ensuring the security of information in organizations. Failures in the security of the information are often produced by people or systems that detect a vulnerability in information systems, cause a security incident,[6] and may have access to confidential information and deny service, copy, modify and even delete such information. Honeypots are computer tools that simulate application and vulnerable production services enticing others to attack trying to exploit some vulnerability of the systems. The use of Honeypots in the network is based on the idea of simulating applications with problems and record all events produced by the attacker, in this way, the network administrator can learn about the different types of attacks to protect the systems of the organization that are in production. This information is of vital importance for the generation of a suitable information security policy[7] in the organization. A security policy is a formal expression of rules to be met by those who have access to information and technologies of an organization. The use of honeypots contributes to the definition of a security policy.

[1] Honeypots are computer tools that simulate vulnerable application and services in production enticing others to attack trying to exploit some vulnerability of the systems.

Working Elements and Technology

The description of the architecture of hardware and software used and also the topology and network address will now be presented. At this stage the research group opted for the use of available hardware for the development of the work without the need to purchase new equipment since the requirements were satisfied. On the other hand, the use of free software tools was prioritized.

Support Architecture for Operating Systems and Services

Hardware: Intel Core 2 Duo E7500 Processor, Memory: 2 GB Ram, 20 GB Disk with rackeable cabinet.
Virtualization Plataform: Proxmox[8] Virtual Enviroment Version 3.1.21, which uses Openvz[9] as
Virtualization technology based on Linux kernel.

Definition of Operating Systems and Services

Operating system: Debian[10] Gnu/Linux 7.0 installed on the virtualization platform.
Honeypot: Dionaea Cache bugs[11]
Framework: DionaeaFR[12]
Database: sqlite3
During this stage we opted for the installation of the Honeypot Dioanea after the analysis of different solutions within the Honeynet Project[13]. Honey Project is a leading global nonprofit organization dedicated to the investigation of recent security attacks as well as the development of free software applications aimed at contributing to the security of Internet. Even though there are different alternatives for honeypots software, Dionaea was chosen since it is not discontinued and because it allows the simulation of different Internet services. Most of the developments of other honeypots are discontinued and they are only intended for the publication of a service. The operating system used for the support of the Honeypot is Debian GNU/Linux 7.0.

Operating Systems Configuration and Network Services Strategies

As previously mentioned, the Honeypot was implemented in a controlled environment. For this purpose, a DMZ[14] (demilitarized zone) was created with a unique vlan[15] (150) behind the Internet firewall of the organization. The Debian Gnu/Linux operating system is configured with a private addressing. Besides, all the traffic coming from Internet was redirected to address Ip 190.105.208.94 from the firewall to the Honeypot server. No domain name linked to the ip was set. On the other hand, the firewall was configured to deny all traffic originated in the Honeypot to the local network. This configuration was carried out to prevent any attack that may arise after exploiting the vulnerabilities of the Honeypots.

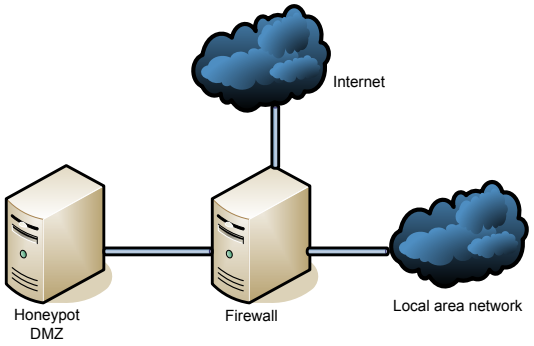


Figure 1. Network topology

Dionaea Installation

Once the Debian Gnu/linux operating system was installed on the virtual platform, the following libraries were installed to satisfy Dinoaea software dependencies.

libev >=4.04, libglib >=2.20, libssl, liblcfg, libemu, python >=3.2 , sqlite >=3.3.6 , readline >=3 , cython libudns, libcurl >=7.18, libpcap >=1.1.1 ,libnl , libgc >=6.8

Dionaea was then installed and configured to store records of incidents in a sqlite3 database. The services that are published by the Honeypot can be displayed in the figure 2. The installation of Dionaea was carried out with a default configuration. The parameterization of the configuration of the services was not necessary since the default configuration runs efficiently. Published services have vulnerabilities that are trying to be exploited, attackers cannot distinguish if the published service is in production or if it's a honeypot.

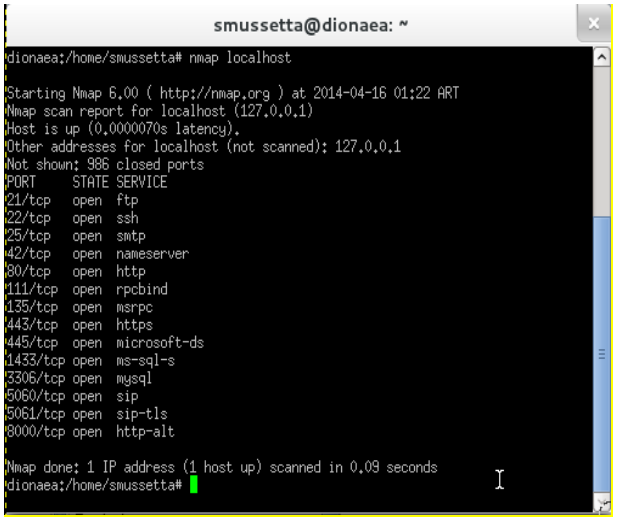


Figure 2. Published Services

Installation of DionaeaFR

For the analysis of the results, queries to the database Sqlite3 were made and the Framework DionaeaFR was used, which is developed in Python and uses the following technologies: Python 2.7.3, Django 1.4, Jquery 1.7.2 Bootstrap Framework 2.1.1, jVectorMap 1.0, Kendo-UI v2011.3.1129, SQLite3. DionaeaFR provides a web interface with preset queries to the Sqlite3 database with reported incidents, thus the administrator can get the details of the incidents in a fast and simple way.

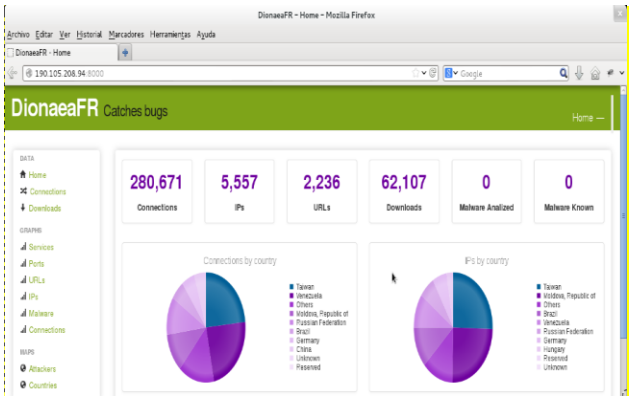


Figure 3. DionaeaFR

Results

The Honeypot remained online for 1 month and recorded the following:

Ip: 5557
Urls: 2236
Downloads: 62107

One of the attempts to exploit vulnerabilities was the smb MS08-067[16] Protocol.
The number of login attempts to the MSSQL Server with different users was also observed.

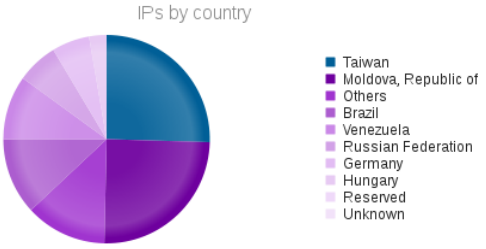


Figure 4. Ips by Country

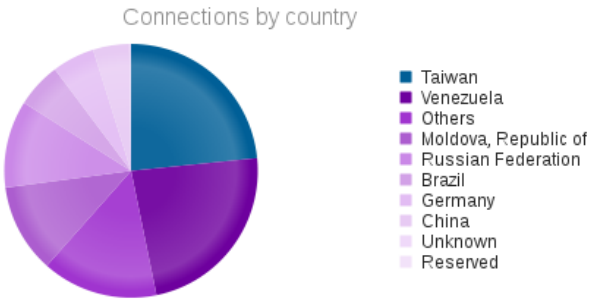


Figure 5. Connections by country

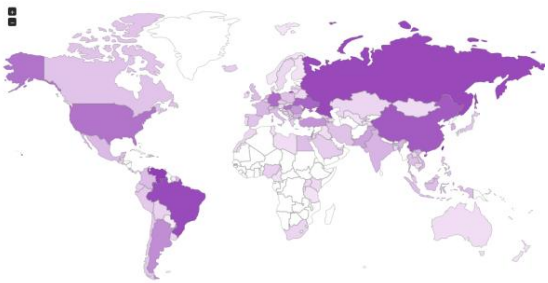


Figure 6. Countries which originated incidents

Table 1 Top 10 Connections and Ips

Connections	IP
32139	124.8.135.111
22838	124.8.131.120
21140	95.65.56.246
9909	177.44.3.189
8291	190.38.20.203
6437	190.38.26.95
6322	186.93.30.169
5288	89.151.148.149
4945	92.228.119.218
4782	89.151.145.31

Table 2. Top 6 Ports and Connections

Port	Connections
445	114025
139	28671
3306	6215
1433	2558
80	426
5060	231

Table 3. Download URL

Connections	Download URL
3943	http://190.38.20.203:7953/vuemt
3084	http://190.38.26.95:7953/afdlvdvq
3060	http://186.93.30.169:7953/jiza
2537	http://89.151.148.149:3835/stzt
2371	http://92.228.119.218:8715/bpmhn
1929	http://95.65.56.246:4148/lgrchx
1849	http://95.65.56.246:4148/ykhovj
1652	http://89.151.145.31:3835/srfcd
1601	http://95.65.56.246:4148/syttjv
1485	http://78.50.34.12:8715/rkrfnvxq

Tabla 4. MSSQL Usernames

Connections	User names
10269	Root
5597	Sa
1031	admin
759	mysql
80	Admi
32	admin12
32	administrator
13	administrato
6	Roo

Discussion

The results obtained through this research allowed to discuss them, highlighting that the systems exposed to the Internet suffered attacks almost instantly, and these sought to undermine the installed services. The service of active Directory (tcp/445) was the service that received the highest number of remote connections, well above other published services. Then Netbios (tcp/139) service also had a lot of connections. Both services are implemented in MS Windows operating systems. If we compare the two published database services, we can affirm that, mysql (tcp/3306) got more than double the connections which the service of Mssql (tcp/1433) database received. Http (tcp/80), sip (udp/5060), telnet (tcp/23), and remote desktop (tcp/3389) services have also received a large number of connections but to a lesser extent with respect to the ones mentioned above.

Moreover, it is crucial to highlight the importance of the customization of the settings of the services that an organization exposes on the Internet, as the default configuration is often not strong enough. The due updates for each service and the regular backups are also recommended. The fact that the vulnerabilities of a system are usually attacked in an intangible way for the user makes it very difficult for the latter to realize when their system is under attack, thus the relevance of applying these security policies.

Conclusion

The construction of the honeypot in itself is an exercise that involves having knowledge about the architecture and the services and applications of the organization, since we cannot build one without the knowledge about how they operate and what the architecture is and what applications and services are provided.

When implementing a honeypot on a local network of an institution which offers a wide variety of services in production, controlling the impact of the honeypot in this network is of vital importance. Different design issues

about the implementation of the honeypot should be carefully studied, as for example, routing, firewalls, virtualization, or the isolation of the honeypot, among others. All these matters should be taken into account so that the interaction between the attacker and the honeypot does not affect the organization's computing resources or generate serious incidents in the production systems.

This work could discloses not only what the most attacked services are but also the risks computer services may suffer if strict measures of parameterization, configuration, upgrade, among others are not taken. A large number of connections were carried out automatically from different countries around the world trying to violate the published services. Note that the services that are published on the honeypot did not have any associated via dns domain, and the public ip used on the honeypot had not been used by the organization before.

The installation of the honeypot required carrying out additional security measures to isolate it from the network environment that is currently in production in the organization.

The fall of a service in production has a high cost, not only in economic terms but also in terms of users' trust, the reputation of the organization, etc.

The current degree of technological progress is so vertiginous that it is very important for honeypots that monitor the evolution of attack techniques to be implemented in the network of the organization in order to upgrade WIPO computer security measures.

Finally, the relevance of this study regarding the application of honeypots in networks whose data are of a public nature should be mentioned, since unlike a company's information assets, which entail private goals, the data to be protected in the present project belong to a public University. On the other hand, it stresses the importance of this research in pursuit of an increasing awareness about the great utility of this powerful security tool.

References

[1] Andrew S. Tanemabun, "Computer Networks", Pearson Education, 2003
[2] www.apache.org, May 2014
[3] http://www.proftpd.org/, June 2014
[4] www.postfix.org, June 2014
[5] www.moodle.org June 2014
[6] An information security incident is a violation or imminent threat to the security policy of the information implicitly or explicitly www.cert.uy, June 2014
[7] Security Policy: RFC2196
[8] http://www.proxmox.com, June 2014
[9] http://openvz.org/Main_Page, June 2014
[10] www.debian.org, June 2014
[11] http://dionaea.carnivore.it, May 2014
[12] https://github.com/rubenespadas/DionaeaFR, June 2014
[13] https://www.honeynet.org/, June 2014
[14] William Stallings, "Cryptography and Network Security, Principles and Practice", Prentice Hall, 1998
[15] http://www.ieee802.org/1/pages/802.1Q.html, June 2014
[16] https://technet.microsoft.com/library/security/ms08-067 June 2014

Aspectos Optimizables en un Protocolo Non - Interactive Dining Cryptographers

Pablo García
Universidad Nacional de La Pampa (UNLPam) - Argentina

Germán Montejano
Universidad Nacional de San Luis (UNSL) – Argentina

Silvia Bast
Universidad Nacional de La Pampa (UNLPam) - Argentina

Abstract

El concepto de seguridad incondicional implica que la información se encuentra segura independientemente del tiempo y los recursos informáticos con los que pueda contar un criptoanalista. En particular, ese aspecto se torna de muy alto valor cuando lo que debe protegerse es la privacidad. Por ejemplo, resulta un elemento fundamental si se trata de voto electrónico.

Dining Chryptographers es un protocolo diseñado para E-Voting cuya principal característica es promover el anonimato eterno del emisor de una información. Una derivación posterior, Non Interactive Dining Cryptographers (NIDC), relaja la condición de concurrencia online exigida en la versión original, incrementando de manera significativa el campo de aplicación del protocolo.

El presente trabajo analiza dos aspectos puntuales de NIDC (almacenamiento de la información y funcionamiento del protocolo antifraudes) y presenta técnicas concretas para optimizar los mismos.

Palabras Clave:

Seguridad Incondicional, Anonimato, Birthday Paradox, Voto Electrónico, Non Interactive Dining Cryptographers, Fraude, Logaritmos discretos, Bit Commitment, Prover, Verifier.

Introducción

Tradicionalmente, los esquemas de voto electrónico se han propuesto como objetivo central la provisión de seguridad incondicional para el proceso específico de votación. Al mismo tiempo, han otorgado un nivel computacional para el mantenimiento del anonimato de los votantes. Esto es lo que puede observarse en todos los esquemas basados en Mix Net.

Tal situación es un claro error conceptual: el anonimato debe garantizarse indefinidamente, mientras que el proceso eleccionario sólo debe ser protegido durante un período limitado de tiempo, dado que, una vez finalizado el mismo, los resultados serán públicos. Tal observación incrementa significativamente el interés por aquellos protocolos que provean niveles de seguridad acorde a lo expresado.

Dining Cryptographers, presentado en [1], es un esquema que se ajusta perfectamente a esas exigencias. Su esquema original puede describirse en los siguientes términos:

"Tres criptógrafos se reúnen a cenar en un restaurant. Al llegar el momento de pagar, el mozo les indica que la adición ya ha sido abonada y, que quién lo hizo, no desea que se conozca su identidad. Los criptógrafos desean saber si el pago fue realizado por alguno de los comensales fue o por alguien externo al grupo."

Los comensales quieren únicamente averiguar si el pagador pertenece al grupo, sin que la identidad del mismo resulte revelada. Obviamente, en caso de un pagador externo, la privacidad está garantizada. La solución que encuentran es la siguiente:

"Cada uno de los comensales lanza una moneda al aire y observa el resultado obtenido. A continuación, comparte esa información con su vecino de la izquierda. Por lo tanto, cada uno de ellos ve exactamente dos monedas, la propia y la del vecino que comparte con él. Finalmente, cada uno debe indicar si las dos monedas que pudo observar son"idénticas" o"diferentes", con la condición de que si alguno de ellos abonó la adición, debe mentir en su afirmación."

La mecánica del modelo asegura que, un número impar de criptógrafos que proclama diferentes indica que el pagador pertenece al grupo. Por el contrario, un número par indica que la cuenta fue abonada por alguien externo al grupo. La seguridad del esquema se demuestra en [2].

Dining Cryptographers exige la concurrencia en el tiempo de todos los participantes. Esto reduce notablemente el campo de aplicación del mismo, dado que existen numerosas aplicaciones que requieren anonimato incondicional, pero que presentan características asincrónicas. Entre ellas, el voto electrónico.

En [3] se propone una metodología que otorga seguridad incondicional al anonimato sin exigir la concurrencia temporal de todos los participantes, que combina el concepto de Firmas Ciegas ([4]) con una derivación del protocolo original a la que se denominó Non - Interactive Dining Cryptographers (NIDC), que puede describirse de la siguiente manera:

1. Cada par de participantes intercambia bits aleatorios, que permitirán la comunicación bilateral segura.
2. En base a los bits aleatorios y la entrada de los participantes, se publica un mensaje.
3. Finalmente todos los mensajes se combinan mediante operaciones que garantizan la cancelación de todos los bits aleatorios, garantizando la permanencia de las entradas de todos los participantes.

La característica asincrónica del modelo aumenta de manera importante el nivel de redundancia, porque es imprescindible implementar un número significativo de verificaciones que permitan asegurar que ningún participante deshonesto pueda arruinar voluntariamente el proceso.

Es importante remarcar que la información circulante debe ser protegida por un período finito de tiempo, por ejemplo, si se trata de un proceso de E-Voting, todas las firmas se publicarán una vez cerrada la elección, haciendo pública esa información en pos de aumentar la transparencia del procedimiento.

El esquema presentado en [3] proporciona seguridad incondicional para el anonimato, resultando de gran valor. Sin embargo, presenta dos elementos que resultan claramente ineficientes:

1. La forma de almacenar los sufragios exige enormes cantidades de memoria.
2. El esquema para prevenir fraudes incluye demasiadas operaciones de máquina.

En consecuencia, se exponen en las próximas dos secciones, metodologías alternativas que optimizan la utilización de los recursos.

Optimización del almacenamiento

El modelo original de NIDC propone un almacenamiento basado en un vector, en el cual cada sufragio se colocará en una posición absolutamente aleatoria, condición imprescindible para garantizar el anonimato.

Aparece entonces la idea de utilizar canales paralelos, replicando cada sufragio en posiciones aleatorias potencialmente diferentes, de manera que resulte suficiente con que una réplica de un sufragio determinado no colisione, para asegurar que ese voto no se perderá.

Concretamente, se implementó un simulador de actos eleccionarios para obtener conclusiones sobre el comportamiento de tres modelos:

1. CUMD (Canal Único y Múltiples Datos): consiste en implementar un único vector de slots y replicar cada sufragio en varias posiciones aleatorias. Los resultados obtenidos no resultaron satisfactorias, fundamentalmente porque se genera un aumento significativo de las colisiones al aumentar la carga, efecto que no resulta mejorado por el hecho de implementar varias réplicas. El esquema de esta variante puede observarse en la Figura 1.

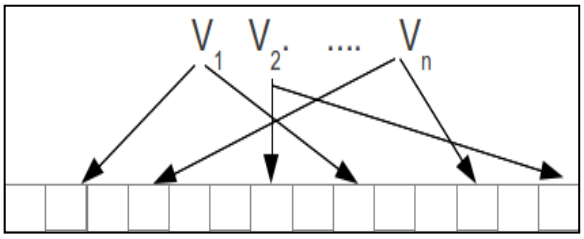


Figura 1

2. MCMD (Múltiple Canales y Múltiples Datos): Manteniendo constante el número total de slots, los mismos se dividen en Q>1 canales. Cada voto es replicado R>1 veces en cada canal. La Figura 2 muestra la implementación de esta propuesta. Al igual que en el esquema anterior los resultados no fueron favorables, por razones similares.

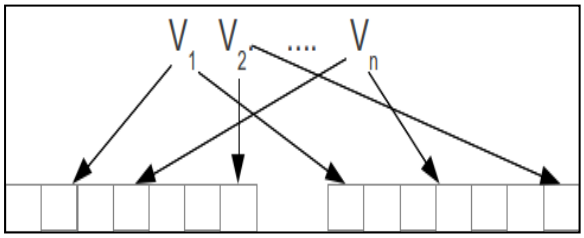


Figura 2

3. MCDU (Múltiple Canales y Dato Único): Similar al anterior, pero cada dato se replica una sola vez en cada canal. El esquema se exhibe en la Figura 3. Esta propuesta generó resultados muy importantes en términos de optimización.

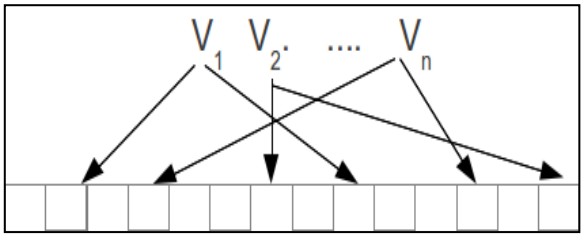


Figura 3

MCDU exhibió resultados claramente mejores, con una evidente optimización en la utilización de los recursos. Sin embargo, para que tal afirmación resulte cierta es imprescindible que el número de slots en cada canal paralelo se mantenga razonablemente por encima del número de votos que se debe administrar. En caso contrario, el número de colisiones locales crece excesivamente y los resultados decaen.

Por ejemplo, la Figura 4 muestra el comportamiento del modelo si se implementan 4800 slots para 480 votantes. El número de canales implementado tomo los valores 1, 2, 3, 4, 5, 6, 8, 10 y 12. La observación de esa figura indica la posible existencia de un Q óptimo, hecho que fue demostrado posteriormente.

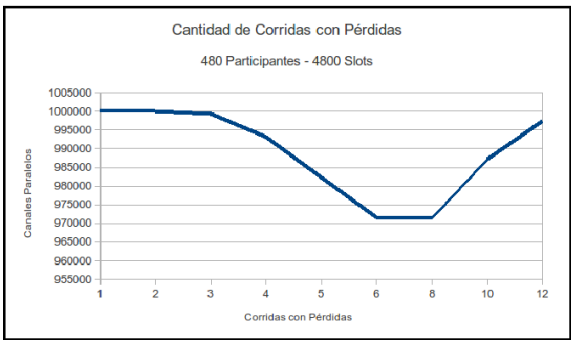


Figura 4

En [5] se presentan una serie de resultados, obtenidos por simulación, que indican una severa optimización en la eficiencia del uso del espacio mediante la nueva propuesta. En [6] se enuncian las primeras fórmulas matemáticas que describen el comportamiento del modelo. En [7] se formaliza un procedimiento sistematizado que permite llevar la seguridad hasta un nivel deseado indicando con precisión los valores que deben otorgarse a los parámetros involucrados para obtener la máxima eficiencia. Tales parámetros son:

- ***N*:** # total de votantes.
- ***T*:** # total de slots.
- ***S*:** # slots en cada canal.
- ***Q*:** # canales paralelos que se van a implementar.
- Q_{ot}**:** # canales paralelos. (Valor óptimo teórico).
- Q_{oa}**:** # canales paralelos. (Valor óptimo aplicable.
- ***PVP*:** porcentaje de votos perdidos en un acto eleccionario.

Los principales aportes realizados durante el desarrollo de [2], [3], [5] y [7] son los siguientes:

- El valor esperado de la variable *PVP* (porcentaje de votos perdidos), puede ser estimado aplicando la fórmula:

$$\mathbf{E}(PVP) = (1 - e^{-NS})^{Q} \quad (1)$$

- El valor óptimo de la variable *S* (cantidad de slots para cada canal paralelo) está dado por:

$$S = | N \ln 2 | + 1 \quad (2)$$

- Para valores dados de T y N, existe un número óptimo de canales paralelos. Tal valor se expresa por la ecuación:

$$Q_{ot} = (\ln 2)\, N \quad (3)$$

Obviamente, el resultado anterior no da como resultado un entero, razón por la cual en la práctica se utiliza el entero siguiente. Tal magnitud es lo que llamamos Q_{oa}.

Otro resultado trascendente obtenido en [7] y refinado en [6] consiste en la obtención de una cota superior para la probabilidad de que no se pierda ningún voto en una votación determinada. Sea:

$$X = \text{"No se pierde ningún voto"}.$$

La cota obtenida indica que:

$$\Pr(X) > (1 - ((1/S))\,(N-1))^{Q})^{N} \quad (4)$$

La demostración de la validez de la ecuación se puede ver en [7]. Es importante remarcar que la cota es razonablemente aproximada porque el error cometido es proporcional a las colisiones múltiples que se produzcan, la cual es poco significativa.

Para mostrar el comportamiento de la ecuación, la Figura 5, compara la curva correspondiente a la fórmula que describe la Paradoja del Cumpleaños, con la cota propuesta, para un único canal y desde 1 a 170 slots.

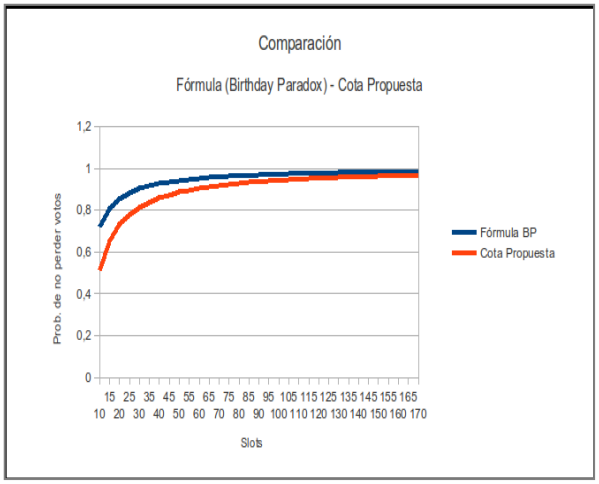


Figura 5

En base a las fórmulas enunciadas previamente se presenta una propuesta concreta para obtener valores de S y Q_{oa} (y en consecuencia, $T = Q_{oa}\, S$) para que el evento X tenga una probabilidad mayor que un valor dado. Se deben llevar a cabo los siguientes pasos:

1. Se elige un valor de S, dado por la fórmula (2).
2. Se calcula el valor de la fórmula (4) para Q=1.
3. Si el valor obtenido se encuentra en el rango exigido, se finaliza el proceso.
4. En caso contrario, Se incrementa el valor de Q en 1 y se vuelve al paso 3.

El algoritmo descripto permite garantizar que la probabilidad de que no se pierda ningún voto es mayor que un valor dado.

Optimización del Protocolo antifraudes

El esquema original de NIDC permite llegar a cualquier nivel de seguridad que se exija. Sin embargo, la utilización de BCX genera un número elevado de operaciones para llegar a valores razonables. Esto ocurre porque cada BCX se refiere a un bit; por lo tanto, cada commitment tiene probabilidad ½ de ser engañado. En consecuencia, para llegar a niveles del orden de, por ejemplo, 10^{-10}, es necesario implementar un número significativo de compromisos.

En [9] se propone entonces un modelo que utiliza otros recursos:

- Logaritmos discretos, que resultan atractivos porque a la fecha no existen algoritmos conocidos que puedan calcularlos en tiempo aceptable. Esto implica que se dispondrá de una situación sólida frente a cualquier ataque de un criptoanalista.
- Commitments de Pedersen. Constituyen una herramienta de enorme utilidad en criptografía. Un esquema basado en commitments permite a alguien comprometer un valor pero sin develarlo, de manera tal que el mismo pueda ser dado a conocer posteriormente. El enfoque garantiza una doble seguridad:
  - El emisor puede mantener el secreto sobre el valor en cuestión hasta que decida revelarlo.
  - El receptor tiene la garantía de que quién publicó la información no podrá modificarla a posteriori.

  El esquema es ampliamente utilizado en protocolos criptográficos, como por ejemplo, Zero Knowledge Proofs (ZKP, [8]). Los commitments de Pedersen, además, son homomórficos, lo cual les otorga un valor adicional desde el punto de vista criptográfico.

Varias de las propiedades de los Commitments de Pedersen que resultan de interés para el presente trabajo son analizadas en detalle en [10]. Algunas de las mismas se mencionan a continuación:

- Otorgan, independientemente del valor a comprometer, una medida constante para el commitment.
- Proveen seguridad incondicional para la privacidad.
- Permiten llevar el nivel de seguridad hasta cualquier valor requerido.
- Garantizan a quien deba utilizarlos que es posible demostrar a cualquier participante de que dos instancias de compromiso corresponden a un mismo valor con pruebas sólidas, que además podrán repetirse un número arbitrario de veces.
- Permiten abrir un subconjunto arbitrario de commitments, sin revelar ninguna información relacionada con aquellos que no han sido abiertos.

El modelo propuesto puede describirse de la siguiente manera:

1. Cada par de participantes (p_i y p_j) intercambia bits aleatorios, formando r_{ij}. Valor se compromete ante el resto de los participantes.
2. Cada participante p_i, selecciona un slot al azar. Es importante destacar que la aleatoriedad de tal elección es condición necesaria para que el anonimato que asegurado.
3. Cada p_i construye un mensaje m_i, el cual contiene valores significativos en el slot elegido y ceros en los demás. El funcionamiento del esquema se detalla en [3]. Al finalizar el proceso, los slots elegidos por cada usuario contendrán los respectivos datos y los restantes solamente ceros. Las colisiones son posibles porque la selección del slot debe ser totalmente aleatoria. Si bien existe la posibilidad de implementar herramientas de recuperación, en una etapa inicial una colisión genera la pérdida de todos los sufragios coincidentes. En [2] y [6] se proponen técnicas para mantener la probabilidad de colisión tan baja como se desee.
4. Se divide m_i en bloques x_i. Cada bloque será un slot del mensaje construido en el paso 3. Es necesario que la medida de los slots sea razonablemente mayor que el mensaje que se desea enviar para poder trabajar con un nivel de redundancia que permita implementar herramientas de verificación de autenticidad. También es aconsejable que dicho tamaño sea potencia de dos, para aprovechar de manera óptima el espacio.
5. Se publica un conjunto de generadores $g_0 \ldots g_l$. El valor l representa la cantidad de slots que se implementan.
6. Cada p_i construye:

$$h = (g_0)^k (g_1)^{x1} \ldots (gl)^{xl} \quad (5)$$

Este punto representa un commitment del mensaje creado, en el cuál k es la clave. Tal commitment debe verificar dos condiciones:

- Es necesario demostrar que *h* es un commitment que corresponde a un mensaje bien formado de acuerdo a las reglas del protocolo, esto es que contiene ceros en la totalidad de los slots, excepto en el elegido. Esto, debe llevarse a cabo sin revelar la ubicación final seleccionada para el sufragio.
- Debe demostrarse que existe relación lineal entre lo comprometido y la contribución final que se va a realizar. Es imprescindible que, luego de demostrar que h corresponde a un mensaje bien formado, se demuestre también de manera fehaciente que la contribución final se corresponde con lo comprometido.

El protocolo final que cumple con ambas condiciones se muestra en la Tabla 1. En efecto, el mismo permite asegurar la primera condición y, simultáneamente garantiza una relación lineal entre el commitment y el mensaje.

La primera condición puede asegurarse mediante la aplicación de los siguientes pasos:

- construir una permutación *h'* del commitment *h*, en la que el orden de los slots resulte modificado.
- Comprometer *h'*.
- Se permitirán dos tipos de desafíos que serán elegidos aleatoriamente por *verifier*:
  - Si elige el primer tipo, *prover* abrirá todos los commitments, menos aquellos que incluyen el mensaje. Obviamente, el resultado de los mismos debe ser 0 en todos los casos.
  - Si *verifier* elige el segundo tipo de reto, lo que se abre es la permutación completa.

Lo expuesto es totalmente equivalente al esquema presentado en [3]. En todos los casos, queda claro que lo comprometido corresponde a un mensaje bien formado.

Con respecto a la segunda condición en cambio, se aplican los siguientes pasos del proceso.

7. Cada pi computa su contribución final:
$$ci = mi + \sum r_{ij} \quad (6)$$
8. Se divide la contribución en bloques w_i, equivalentes a los x_i, en el sentido de que

Tabla 1: Descripción del Protocolo de Comunicación

Votante		**Autoridad**
Bits aleatorios r_i	→	
	←	Bits aleatorios r_j
Construye m_i con el mensaje en la posición elegida y cero en el resto. Lo divide en bloques del tamaño de un slot.		
Construye $h = (g_0)^k (g_1)^{x1} ... (g_l)^{xl}$		
Construye h', permutación (por bloques) de h		
h'	→	
		Elige aleatoriamente $x \in \{0, 1\}$
	←	x
Genera y: Si (x=0) ⇒ y = Abrir compromisos Si (x=1) ⇒ y = Exhibir permutación		
y	→	
		Verifición correcta ⇒ o = OK Verificación incorrecta ⇒ o = CANCEL
	←	o
Calcula la contribución definitiva $c_i = mi + \sum rij$		
Calcula $a = g_0^s g_1^{w1} ... g^{wl}$ (compromiso de la contribución definitiva)		
a	→	
		Elige aleatoriamente $v_i \in Z^+$
	←	v_i
Calcula $b = g_0^{vi\, r + s} g_0^{vi\, x1 + w1} ... g_0^{vi\, xl + wl}$		
b	→	
		$(b = h^c\, a) \Rightarrow y$ = OK $(b \neq h^c\, a) \Rightarrow y$ = CANCEL
	←	y

cada uno tiene el tamaño de un slot. La diferencia es que estos nuevos bloques wi corresponden a un slot de la contribución definitiva c_i.

9. Se construye un nuevo commitment a, que hace referencia a la contribución final:

$$a = g_0^{s}\, g_1^{w1} \ldots\ g_l^{wl} \quad (7)$$

Cada participante podrá comprobar que este nuevo commitment se corresponde exactamente con lo comprometido en *h*. Para ello podrá desafiar con tantos valores v_i como se desee. En todos los casos, *prover* responderá con:

$$b = g_0^{vi\, r + s}\, g_0^{vi\, x1 + w1} \ldots\ g_0^{vi\, xl + wl} \quad (8)$$

Verifier podrá controlar que:

$$b = h^{c}\, a \quad (9)$$

Si la probabilidad de fraude en el i-ésimo caso es p_i, la probabilidad de engañar *n* desafíos será $p_i n$, en base a propiedades de los sucesos independientes, las cuales se desarrollan en profundidad en [11].

Entonces, la probabilidad de fraude se puede hacer tan pequeña como se desee. Cabe mencionar que la probabilidad p_i va a ser más pequeña que el ½ propuesto en el modelo original de [3], el cuál es un valor que se origina en la propia operación de XOR. Por lo tanto, para alcanzar niveles satisfactorios de seguridad, el número de desafíos necesario será significativamente menor.

Por lo tanto, la probabilidad del evento:

X="falla un commitment"

es:

$$P(X) = 1/q \quad (10)$$

Donde *q* es el mayor valor que pueda tomar el commitment en cuestión, es decir el orden del grupo asociado. En consecuencia, es un número primo grande. Por lo tanto, para muchas de las aplicaciones habituales, el uso de un único reto dará niveles aceptables de seguridad. Sin embargo, si en alguna ocasión fuera necesario un nivel de seguridad extrema, el producto de dos o más valores tan cercanos a cero puede proporcionar cualquier nivel de seguridad exigido.

10. El resultado de la elección se calcula como el producto de todas las contribuciones.
11. Al resultado final se le aplica XOR de todos los r_{ij}.

La seguridad del esquema ha sido demostrada en [7], mediante un análisis de las vistas (*views*) que cada participante dispone al finalizar el proceso. Todos los participantes dispondrán de información que resulta insuficiente para reconstruir el resto de los datos. Por lo tanto, aún cuando pudieran existir conspiraciones, el esquema resulta seguro.

Se deben distinguir dos situaciones diferentes:

- El proceso se lleva a cabo con la participación de un grupo reducido de autoridades. Es el modo más aplicable a voto electrónico. En tal caso, cada votante sólo debe intercambiar mensajes con dichas autoridades.
- No se implementa la figura de las autoridades. Es una generalización del caso anterior. En este caso, cada participante cumple la función de autoridad de todos los votos, menos del propio.

Se pone énfasis en el modelo que incluya autoridades, dado que es el modelo más habitual para voto electrónico. Sin embargo el esquema es fácilmente generalizable a un modelo que no las implemente.

Definimos:

- P_i: Participante i-ésimo. Para el caso de un esquema de voto electrónico, es un votante.
- A_i: Autoridad i-ésima. Para el caso de un esquema de voto electrónico, es aquella persona cuyo rol consiste en auditar la correctitud del proceso.

La tabla 1 muestra, de manera esquemática, el intercambio de información entre un participante y una autoridad. Obviamente, se generaliza con facilidad, resaltando que, en este caso, cada P_i tiene comunicación con todas las autoridades A_i, pero no con los demás participantes.

La seguridad del esquema puede confirmarse observando las vistas de los involucrados en el proceso:

Cada votante tendrá a la vista los siguientes datos de las autoridades:

1. Grupo de bits aleatorios r_j.
2. Reto *x*.
3. Retos v_i. Podrían ser *n*, con $n \in Z+$. Cuanto mayor sea el valor de *n*, mayor es el nivel de seguridad obtenido.
4. Respuesta *b*.

Los datos que observa un votante son exclusivamente desafíos más el intercambio inicial de bits aleatorios necesario para comenzar el proceso de intercambio.

Cada autoridad tendrá una vista consistente en los siguientes datos provenientes del votante:

1. Grupo de bits aleatorio r_{ij}.

2. Commitment *h*.
3. Respuesta y al desafío *x*.
4. Commitment *a*.
5. Respuesta *b* a los desafíos *vi*.

La información que un votante presenta a una autoridad está conformada por commitments y respuestas a desafíos, más el necesario intercambio inicial de bits aleatorios. La seguridad relacionada con tal información puede llevarse al nivel que se desee. La publicación final de los resultados no agrega información que pueda comprometer el anonimato.

El protocolo se basa en lo expuesto en la sección 2.4.3 de [1], cuya seguridad es demostrada en la sección 2.4.8 del mismo documento.

Conclusiones

Con respecto a la optimización del uso del espacio, se pueden enunciar las siguientes conclusiones:

- La utilización de canales paralelos resulta favorable si cada voto es replicado una sola vez en cada uno de ellos.
- Los parámetros involucrados deben ser elegidos apropiadamente. Para ello [6] describe una técnica específica.
- La esperanza matemática de la variable PVP (porcentaje de votos perdidos), puede ser estimada aplicando la fórmula (1).
- El número de slots de cada canal individual debe mantenerse en un nivel razonablemente superior a la cantidad de votos. Si no se verifica esa condición, la cantidad de colisiones aumenta de manera considerable y la calidad de los resultados decae.
- Para garantizar que tal situación no se produzca, se puede utilizar la ecuación (2), que define, para un número fijo de votantes N, el valor óptimo de la variable S (cantidad de slots para cada canal paralelo.
- Para valores dados de T y N, existe un número óptimo de canales paralelos. Tal valor se expresa por la ecuación (3).
- Es muy importante la obtención de una cota superior apropiada para la probabilidad de que no se pierdan votos. La misma se obtiene mediante la aplicación de la ecuación (4).

En base a las fórmulas enunciadas, en [6] se presenta una técnica concreta que sistematiza la búsqueda de los valores óptimos para S y Q de manera que la probabilidad de que no se pierda ningún voto resulte mayor o igual que un valor dado.

En lo referido al protocolo antifraudes, es posible afirmar que la propuesta basada en logaritmos discretos y commitments de Pedersen resulta significativamente más eficiente que el modelo basado en BCX, reduciendo drásticamente el número de operaciones realizadas. Tal mejora se deriva fundamentalmente de la la semántica de cada commitment individual.

Concretamente, en el esquema original, cada BCX tiene una probabilidad ½ de ser engañado. Para obtener un nivel deseado de seguridad *ns*, deben implementarse *d* BCX para cada bit de información, de manera que se verifique:

$$ns < ½\ d$$

En el nuevo modelo, cada compromiso tiene asociada una probabilidad de *1/n* de fraude, con $n \in Z^+$, un primo grande. En consecuencia, cada commitment tiene una probabilidad pequeña de ser engañado. Adicionalmente, el agregado de un commitment aumenta significativamente el nivel de seguridad, porque se trata de sucesos independientes cuyas probabilidades son muy bajas, por lo tanto el nuevo nivel de seguridad debe verificar:

$$ns < d\ (1/n)$$

Finalmente, una diferencia fundamental entre ambos enfoques es que en la nueva propuesta los commitments no se aplican a nivel de bit, como en [3]. Por el contrario cada uno de ellos trabaja sobre la totalidad del mensaje, lo cual deriva en una significativa optimización, en términos de las operaciones que deben llevarse a cabo. Un análisis detallado de los niveles de optimización obtenidos se expone en [6].

Agradecimientos

El agradecimiento permanente de los autores para los docentes de la Universidade Federal de Minas Gerais que con suma generosidad dieron un apoyo logístico fundamental durante la estadía de Pablo García en Belo Horizonte, en el marco del convenio UNSL – UFMG para el desarrollo de una tesis de Maestría en Ingeniería de Software [7] que fue defendida con calificación de Sobresaliente en noviembre de 2013:

- Jeroen van de Graaf, PhD.
- Dr. José Monteiro da Mata.
- Dr. José Marcos Silva Nogueira.
- Dr. Carlos Camarao.

- Dr. Luiz Chaimowicz.
- Dr. Wagner Meira Jr.
- Dra. Giselle Lobo Papa.
- Dra. Jussara Almeida-

Referencias

1. Chaum D.: ”The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability”. Journal of Criptology. 1988.

2. van de Graaf J., Montejano G., García P.: “Manejo de Colisiones en un Protocolo Dining Cryptographers”. Anales del Workshop de Seguridad Informática (WSegI, ISSN: 2313-9110) de las 42 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO 2013, ISSN: 1850-2776). Ps. 35 a 50.

3. van de Graaf J.: ”Anonymous One Time Broadcast Using Non Interactive Dining Cryptographer Nets with Applications to Voting”. Publicado en: ”Towards Trust-worthy Elections”. Ps. 231241. Springer - Verlag Berlin, Heidelberg. ISBN: 978-3-642-12979-7. 2010.

4. Fujioka A., Okamoto T., Ohta K.: ”A Practical Secret Voting Scheme for Large Scale Elections”. AUSCRYPT 1992. LNCS, Vol. 718. Páginas 244 a 251. Springer Heidelberg. 1993.

5. van de Graaf J., Montejano G., García P.: “Optimización de un esquema “Ocuppancy Problem” orientado a E – Voting”. Memorias del XV Workshop de Investigadores en Ciencias de la Computación 2013 (WICC 2013). ISBN: 9789872817961. Ps. 749 – 753.

6. García P., van de Graaf, J., Montejano G., Bast S., Testa O.: “Implementación de Canales Paralelos en un Protocolo Non Interactive Dining Cryptographers ”. Workshop de Seguridad Informática (WSegI) de las 43 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO 2014).

7. García P.: “Optimización de un Esquema Non–Interactive Dining Cryptographers”. Tesis de la Maestría en Ingeniería de Software (UNSL). Directores: Jeroen van de Graaf, PhD. (UFMG) y Dr. Germán Montejano (UNSL) .

8. Feige U., Fiat A., Shamir A: ”Zero Knowledge Proofs of Identity”. Journal of Cryptology. 1988.

9. van de Graaf, J., Montejano G., García P.: “Optimización de un Protocolo Non-Interactive Dining Cryptographers”. Congreso Nacional de Ingeniería Informática / Sistemas de Información. CoNaIISI 2013. 21 y 22 de noviembre de 2013. Córdoba, Argentina.

10. van de Graaf J.: ”Voting with Unconditionally Privacy: CFSY for Booth Vo-ting”. IACR Cryptology ePrint Archive. Ps. 574-579. 2009.

11. Feller W.: ”An Introduction to Probability Theory and its Applications”. Volúmen I. Tercera Edición. John Wiley and Sons. New York, 1957.

Datos de Contacto

Pablo García. Facultad de Ciencias Exactas y Naturales – Universidad Nacional de La Pampa. Uruguay 151 – (6300) Santa Rosa - La Pampa – Argentina.
E-mail: pablogarcia@exactas.unlpam.edu.ar

Germán Montejano. Facultad de Ciencias Físico Matemáticas y Naturales - Universidad Nacional de San Luis. Avenida Ejército de Los Andes 950 - (5700) San Luis – San Luis – Argentina.
E-mail: gmonte@unsl.edu.ar

Silvia Bast. Facultad de Ciencias Exactas y Naturales – Universidad Nacional de La Pampa. Uruguay 151 – (6300) Santa Rosa - La Pampa – Argentina.
E-mail: silviabast@exactas.unlpam.edu.ar

Redes anónimas como instrumento para la conformación de un ciberataque sobre una infraestructura crítica

Eduardo Casanovas
Instituto Universitario Aeronáutico
Av Fuerza Aérea 6500 - Córdoba
ecasanovas@iua.edu.ar

Maximiliano Delfino
Instituto Universitario Aeronáutico
Av Fuerza Aérea 6500 - Córdoba
delfino.maxi@gmail.com

Abstracto

Los ciberdelitos se incrementan día a día, como es sabido distintos son los objetivos que mueven a un atacante para cometer un ilícito. Cuando esos ataques se producen contra estructuras cuyo normal funcionamiento se ve afectado y se ponen en riesgo o comprometen la gobernabilidad de una Nación, entonces estamos frente a un evento de seguridad realizado a una Infraestructura Crítica. Hace ya varios años que distintos países en el mundo han comenzado a definir cuáles son sus Infraestructuras Críticas y han comenzado a tomar acciones a los efectos de montar y generar un esquema defensivo que les permita hacer frente a esta nueva amenaza. Uno de los objetivos que tiene un delincuente cuando realiza un ataque, es borrar sus huellas, es decir, que no se pueda determinar desde donde fue lanzado el ataque. Para ello, la utilización de redes anónimas le proporcionan un medio más que interesante sobre el cual puede acercarse hasta su víctima, acceder al sistema y salir sin dejar rastros de trazabilidad.El presente trabajo busca de hacer un aporte a la preparación de las barreras defensivas que es necesario construir y que será necesario articular con otras medidas para completar un bloque homogéneo e inexpugnable. Asimismo para el caso que el ataque haya sido exitoso, disponer de los mecanismos de recolección de evidencia digital.

1. Introducción

Si realicemos un análisis –Vulnerabilidad – Impacto – Mitigación, estamos dando los primeros pasos en la conformación de las barreras defensivas de la organización. Ahora bien, si esa organización se trata de una Infraestructura Crítica, el atacante está pensando en algo que tenga un gran impacto. Es por eso que la preparación que hay que tener es mucho mayor. A nivel nacional se ha tomado la experiencia de otros países y se ha plantado un programa que tiene como finalidad impulsar la creación y adopción de un marco regulatorio específico que propicie la identificación y protección de las infraestructuras estratégicas y críticas del Sector Público Nacional, los organismos interjurisdiccionales y las organizaciones civiles y del sector privado que así lo requieran, y la colaboración de los mencionados sectores con miras al desarrollo de estrategias y estructuras adecuadas para un accionar coordinado hacia la implementación de las pertinentes tecnologías. Estados Unidos manifiesta en su portal de la Casa Blanca: "La amenaza cibernética es uno de los desafíos económicos y de seguridad nacional más serios que enfrentamos como nación". España define por ley que es una Infraestructura Crítica: "Las infraestructuras estratégicas (es decir, aquellas que proporcionan servicios esenciales) cuyo funcionamiento es indispensable y no permite soluciones alternativas, por lo que su perturbación o destrucción tendría un grave impacto sobre los servicios esenciales". En Argentina se crea el ***"Programa Nacional de Infraestructuras Críticas de Información y Ciberseguridad" (ICIC)***, creado mediante la Resolución JGM Nº 580/2011. Es decir, que ya se ha dado un gran paso, que es el reconocimiento de un enemigo cibernético y la necesidad de prepararnos para poder construir una defensa y reaccionar frente a incidentes.

Este programa funciona en el ámbito de la Oficina Nacional de Tecnologías de Información (ONTI) y este es dependiente de la Jefatura de Gabinete de Ministros. Cuatro grupos diferentes integran el programa y tienen tareas específicas que desarrollar.

El Grupo de Infraestructura Crítica de Información es el encargado del relevamiento, identificación y clasificación de las infraestructuras estratégicas y criticas

de Información, así como el monitoreo de los servicios que el Sector Público Nacional brinda a través de la red de Internet y aquellos que se identifiquen como infraestructura crítica, para la prevención de posibles fallas de seguridad.

Al igual que otros países Argentina define: "Las infraestructuras críticas de información (ICI) son las instalaciones, redes, servicios y equipos físicos y de TI cuyo funcionamiento es indispensable para brindar servicios a los ciudadanos y las instituciones. En ese sentido su interrupción, perturbación o destrucción podría tener un grave impacto y repercusión importante en la continuidad de las operatorias de los Organismos del Sector Público Nacional (SPN), como por ejemplo interrumpiendo servicios básicos como salud, seguridad, energía eléctrica, comunicaciones, agua, hasta inclusive el bienestar económico de los ciudadanos".

Otro de los grupos creados es el Grupo de Acción Preventiva, El GAP es el responsable de monitorear los sistemas y servicios informáticos expuestos en las redes. Realiza relevamientos de estos en diferentes grados de intensidad detectando posibles riesgos a la seguridad de los mismos. Este grupo es el encargado de proponer los Planes Estratégico que deben ser implementados para lograr una adecuada protección de las Infraestructuras Críticas.

El CERT es el Grupo de Respuesta a Emergencias Teleinformáticas, se basan en la coordinación de emergencias de incidentes informáticos.

Entre sus acciones cuenta con:

• Actuar como repositorio de toda la información sobre incidentes de seguridad, herramientas, técnicas de protección y defensa.

• Promover la coordinación entre las unidades de administración de redes informáticas del Sector Público Nacional, para la prevención, detección, manejo y recopilación de información sobre incidentes de seguridad.

• Centralizar los reportes sobre incidentes de seguridad ocurridos en redes teleinformáticas del Sector Público Nacional que hubieren adherido al Programa y facilitar el intercambio de información para afrontarlos

• Interactuar con equipos de similar naturaleza.

Ya hemos identificado y entendido que es una infraestructura crítica y además comprendimos la importancia de poner esfuerzo en el armado de una defensa sólida. Para ello, el planeamiento estratégico, que es responsabilidad del GAP, será de fundamental importancia.

El grupo de investigación en Seguridad Informática del Instituto Universitario Aeronáutico está trabajando en la preparación de vectores de ataque que puedan ser utilizados contra las Infraestructuras Críticas. Esto permitirá identificar potenciales vías de acceso al sistema. Es por ello que hemos comenzado a experimentar la ejecución de ataque por medio de redes anónimas.

Históricamente, el objetivo principal de redes anónimas era evitar la censura "política" de algunos países y permitir la libertad de expresión en Internet a través del anonimato de los usuarios. Sin embargo, a través del tiempo se le ha dado otro tipo de uso en actividades ilegales de venta de armas y drogas, pornografía infantil, documentos falsos, publicación de documentos confidenciales de diferentes gobiernos, etc.

Existen diferentes herramientas para preservar el anonimato y lograr la privacidad de los usuarios. Describimos brevemente tres herramientas de las más importantes:

I2P (Invisible Internet Project): es una red de nodos que crean canales de comunicación unidireccionales seguros (túneles). Este tipo de red adopta un modelo InProxy, es decir, es un modelo VPN (Virtual Private Network) "puro" donde ninguna entidad externa puede participar en la red sin antes unirse a ella. Los principios contenidos en este modelo normalmente son conocidos como "darknet".

Freenet: Es una red de distribución de información descentralizada y resistente a la censura. El objetivo de Freenet es almacenar documentos y permitir su acceso posterior por medio de una clave asociada, impidiendo que sea posible la censura de los documentos y ofreciendo anonimato tanto al usuario que publica el documento como al que lo descarga. Esta red se diseñó como una red P2P no estructurada de nodos no jerarquizados que se transmiten mensajes y documentos entre ellos. Los nodos pueden funcionar como nodos finales, desde donde empiezan las búsquedas de documentos y se presentan al usuario, o como nodos intermedios de enrutamiento.

TOR (TheOnionRouter): Es un proyecto cuyo objetivo principal es el desarrollo de una red de comunicaciones distribuida de baja latencia y superpuesta sobre internet en la que elencaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad, es decir, su dirección IP (anonimato a nivel de red) y que, además, mantiene la integridad y el secreto de la información que viaja por ella.

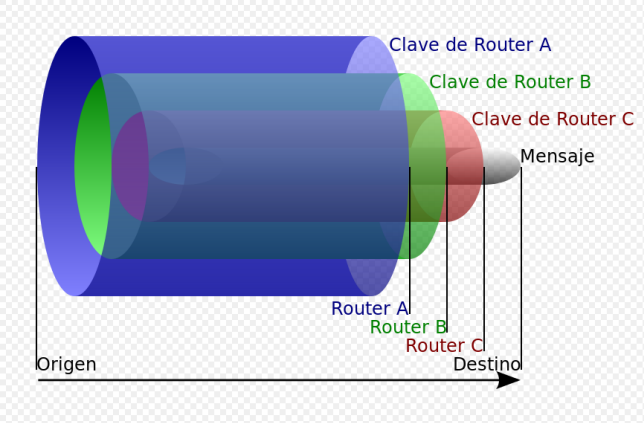
Ya que TOR está montada sobre Internet será la herramienta en la que concentraremos la investigación.

2. Descripción y característica de la red TOR

Como se describió anteriormente TOR es una red anónima montada sobre Internet. La red funciona a partir de un conjunto de organizaciones e individuos que donan su ancho de banda y poder de procesamiento.

TOR propone el uso de encaminamiento de cebolla de forma que los mensajes viajen desde el origen

al destino a través de una serie de routers especiales llamados 'routers de cebolla' (en inglés onionrouters). A continuación una ilustración del encaminamiento de cebolla para un mensaje determinado:

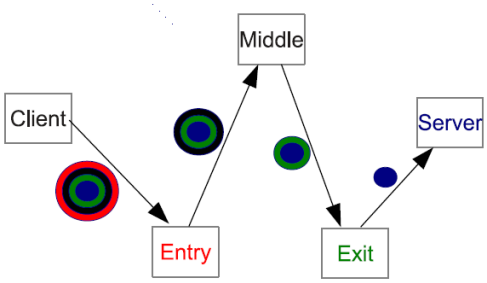


1. Encaminamiento de cebolla

Fte: http://es.wikipedia.org/wiki/Encaminamiento_de_cebolla

No es una red entre iguales (peer-to-peer) ya que por un lado están los usuarios de la red y por otro lado los encaminadores del tráfico, algunos de los cuales hacen una función de servicio de directorio.

Tor sólo permite hacer anónimo el tráfico TCP y nunca transmite tráfico de usuarios a través de más de dos nodos. Entonces nuestro camino está dado por un nodo de entrada, un nodo intermedio y un nodo de salida:



2. Camino típico de TOR

Fte: Eric Chan-Tin, Jiyoung Shin and Jiangmin Yu: "Revisiting Circuit Clogging Attacks on Tor", 2013 International Conference on Availability, Reliability and Security.

El primer nodo (entrada) conoce la fuente, pero no el destino, el último nodo (salida) conoce el destino, pero no la fuente y cualquier nodo intermedio desconoce el origen yel de destino. Podemos decir que, el objetivo principal de TOR es conseguir que internet pueda usarse de forma que el encaminamiento de los mensajes proteja la identidad de los usuarios, es decir, persigue que no se pueda rastrear la información que envía un usuario para llegar hasta él. Esta característica es la que hace que este tipo de redes sean buscadas por aquellos que intentan realizar acciones fuera de la ley.

Para aquellos que van a utilizar esta plataforma como medio para el montaje de un ataque saben que el enrutado anónimo no asegura el que la entidad origen sea desconocida para la entidad destino. Esto es debido a que los protocolos de nivel superiorpueden transmitir información sobre la identidad. Por ejemplo un servicio web puede usar cookies o simplemente pedir que nos identifiquemos. Cuando el atacante busca un anonimato a nivel de aplicación debe configurar el cliente adecuadamente y usar proxys que modifican los contenidos en este sentido. Por lo tanto para conseguir más enmascaramiento del ataque configuran el navegador adecuadamente (Ej. deshabilitando cookies, no permitiendoplugins Java, Flash o ActiveX, o deshabilitando el historial) y redirigir el tráfico hacia un proxy web intermedio que filtre el contenido que puede ser aprovechado para identificarlo. De esta forma el atacante se prepara para borrar todo dato que lo pueda vincular con su víctima.

3. Montaje de unataque

El montaje de un ataque se realiza en distintas etapas, las que el intruso va desarrollando hasta que consigue su objetivo. La primera de ellas consiste en el Reconocimiento que es el proceso de investigación de la organización objetivo, reunir información sobre el mismo a partir de fuentes de acceso público, como los servicios de registro de dominios, páginas web, análisis de metadata de documentos públicos etc. Algunos intrusos han incorporado técnicas como la ingeniería social en la fase de reconocimiento. La ingeniería social consiste en engañar a los usuarios, posiblemente a través del teléfono, mails, etc. donde intentan que empleados distraídos o poco conscientes del problema de seguridad, divulguen información sensible.

Ya con información base disponible comienza con la otra fase, Scanning es el proceso de búsqueda de las aberturas de la organización objetivo, tales como gateways de Internet, puntos de acceso inalámbricos, sistemas disponibles, puertos de escucha, sistemas Operativos etc. Con esta información adicional disponible, el atacante la vincula con las vulnerabilidades que tiene cada sistema en uso.

Finalmente se llega a la fase de explotación, los atacantes explotan los sistemas objetivo logrando comprometerlos, posiblemente, conseguir el control de ellos, causar una denegación de servicio, robar información, generar fallas en los sistemas, etc.

4. Preparar las barreras defensivas en la infraestructura

4.1. Aplicar técnicas de Forensia

Es importante entender que frente a un ataque una organización debe: rápidamentedetectarlo, reaccionar en consecuencia, es decir, poner en marcha, si fuera necesario, el plan de contingencia e intentar rastrear quien fue el responsable de ese evento de seguridad. Para ello y como parte de una estrategia defensiva se deben aplicar técnicas forenses que permitan preparar a la organización tener las evidencias del ataque y ponerlas a disposición de la justicia para su posterior aplicación en estrados judiciales.

La informática forense tiene por objeto:

- La compensación de los daños causados por los criminales o intrusos.
- La persecución y procesamiento judicial de los criminales.
- La creación y aplicación de medidas para prevenir casos similares.

En particular queremos quedarnos con el tercer concepto que marca la aplicación directa en medidas defensivas que surgen a partir de la detección y el entendimiento de ataques ya producidos.

La forensia informática es la aplicación de técnicas, científicas y analíticas especializadas, a infraestructuras tecnológicas que permitan identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal. Dichas técnicas incluyen reconstruir el bien informático, examinar datos residuales, autenticar datos y explicar las características técnicas del uso aplicado a los datos y bienes informáticos.

El cómputo forense se encarga de analizar diferentes sistemas informáticos en la búsqueda de datos, información y evidencia, potenciales o relevantes, que colaboren con una causa judicial o una negociación extrajudicial.Es de carácter científico ya que utiliza el método científico, el que supone la adquisición de nuevos conocimientos, mediante el estudio de la evidencia observable y medible, aplicando un razonamiento lógico, elaborando modelos e hipótesis y corrigiendo o mejorando estas últimas según se obtiene más evidencia.

Por otra parte, la República Argentina no está completamente adherida a convenios internacionales sobre cibercriminalidad que contribuyan a la rápida y eficaz cooperación entre naciones. En particular, se encuentra en proceso de adecuarse al Convenio de Budapest (sancionado en el año 2001 por el Comité de Ministros del Consejo de Europa junto a Estados Unidos, Canadá, Japón, Costa Rica, México y Sudáfrica, y en vigencia desde el año 2004), el único que se encarga de la seguridad informática y trata los delitos contra la Confidencialidad, Integridad y Disponibilidad de los datos y los sistemas informáticos. La Convención de Budapest es prioritaria debido a que provee un marco veloz y seguro, de cooperación y colaboración internacional para garantizar la persecución de delitos transnacionales, por lo que la participar de este acuerdo de forma plena le permitiría a la Argentina tanto ofrecer como recibir asistencia técnico legal en materia de cibercrimen.

Desde la sanción de este acuerdo, nuestro país enfrentó y enfrenta un largo proceso tendiente a minimizar las lagunas jurídicas que existen en el ordenamiento nacional respecto a los delitos producidos a través de las nuevas tecnologías de la información. Si bien se han alcanzado hitos importantes (como la Ley 25.326 de Habeas Data en el año 2000 y la Ley 26.388 de Delitos Informáticos en el año 2008) es bastante todavía el trabajo que debe realizarse en materia de reglamentación, siempre con la intención de unificar y estandarizar criterios.

Cuando se sufre un delito tecnológico es necesario neutralizarlo, saber cómo se ha perpetrado la vulnerabilidad, el alcance del delito realizado y fundamentalmente realizar las acciones necesarias para prevenir futuros ataques mediante el uso de técnicas, programas y herramientas forenses que determinen de manera infalible la evidencia legal.

Después de que ha ocurrido un crimen o incidente que implique una computadora, un especialista adiestrado en informática forense puede examinar la misma para encontrar pistas de lo que ha pasado. Este es el papel del examinador forense de computadoras. Este especialista podría trabajar para el estado como agente de la ley, o para una empresa privada en algunos casos, como los incidentes de seguridad en un sistema. Aunque en cada uno de los dos casos la ley es diferente, la estrategia de investigación para el especialista es más o menos la misma.

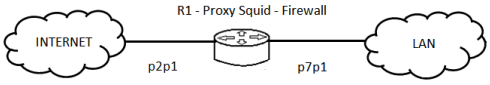
Un principio muy utilizado y fundamental en la ciencia forense es el Principio de Intercambio o transferencia de Locard, (EdmondLocard, francés fundador del instituto de criminalística de la universidad de Lion). Este principio dice que cualquiera o cualquier objeto que entra en la escena del crimen deja un rastro en la escena o en la víctima y vice-versa. Es decir que se lleva consigo algo de la escena, porque cada contacto deja un rastro. Si entramos en contacto con la escena del crimen con toda seguridad dejaremos algo ahí, pelo, sudor, huellas, etc., y también nos llevaremos algo cuando abandone la escena, ya sea barro, olor, una fibra, etc. Con algunas de estas evidencias, los forenses podrán demostrar que hay una posibilidad muy alta de que el criminal estuviera en la escena del crimen. Podemos hacer una analogía total si lo pensamos en términos

informático, pero no siempre el sistema está preparado para poder analizar con posterioridad si algún intruso ingreso al sistema, cuánto tiempo estuvo en el mismo, en que partes del sistema estuvo etc. Es por ello que si somos responsables de la seguridad de una infraestructura crítica, se plantea como mandatorio, la necesidad de contar con especialistas en forensia que asesoren adecuadamente dando los lineamientos indispensables que serán puestos en marcha al momento de la presencia de un incidente.

4.2. Configuración dinámica de un Firewall como forma para mitigar ataques realizados desde redes anónimas

Estudios recientes intentan demostrar que el anonimato de la red TOR no está garantizado. Con lo cual aquellos que utilizan esta red como forma para garantizar su privacidad está comenzando a ser cuestionada. Es decir que esta red no escapa a las generales de la ley y también es atacada, buscando romper el postulado del anonimato garantizado.Uno de los ataques más conocidos para romper el anonimato sobre TOR es infectar una buena cantidad de nodos y esperar que se dé el camino necesario para obtener toda la información de un usuario. Otra forma es analizar el tráfico que pasa por un nodo infectado y de esta forma obtener información. Estos son ataques no dirigidos, no escalables por el tamaño actual de la red TOR (6000 nodos aproximadamente) y según otros estudios para romper el anonimato es necesario analizar tráfico durante 6 meses aproximadamente. Por estas razones no son ataques que nos interesen por el momento, porque si nuestra organización es atacada desde ese tipo de redes, todavía no está garantizada que podamos saber la identidad de nuestro atacante.

Entonces debemos asumir que todavía la utilización de redes anónimas es un acceso que al atacante le sirve para ocultar la entrada y salida del sistema. Adicionalmente debemos entender que este tipo de redes no le aportan nada a la organización por lo que tanto el trafico entrante como el saliente se supone de alto riesgo y por lo tanto se debe eliminar. A continuación describimos una forma de defendernos de ataques provenientes de la red TOR:Se planteó una topología simple de red en la que conectamos nuestra red LAN, que queremos proteger, a la red de Internet. Se utilizó el sistema operativo Fedora 18, el firewall Iptables y un proxy Squid.

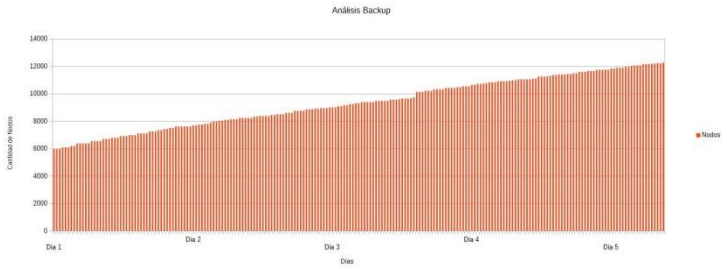


Configuramos nuestro proxy como transparente para poder conectarnos a internet pero que desde internet no se pueda acceder a nuestra LAN. Esto se logra redirigiendo todo el tráfico del puerto 80 y si es necesario el puerto 443 para los protocolos HTTP y HTTPS respectivamente.

Hemos desarrollamos un script con reglas de Iptables en las cuales bloqueamos tanto los nodos activos como los nodos que alguna vez registramos activos en la red TOR. Este es un primer paso para la defensa de nuestra red ya que no permitimos el tráfico de las IP de los nodos TOR que pasa por nuestro router.También, si un usuario se quiere conectar a la red TOR para navegar y poner en peligro nuestra LAN no lo podrá hacer ya que no se le podrá asignar una dirección IP para navegar sobre TOR.

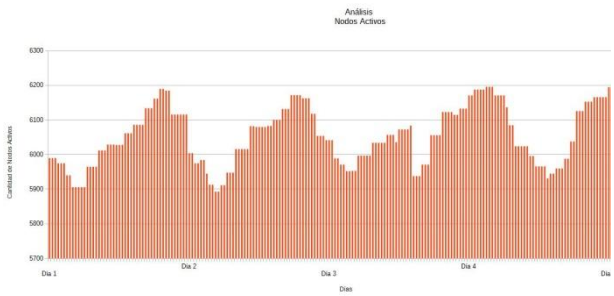
Por otro lado, se ha automatizado el script para tomar datos de los nodos activos cada 30 minutos y se ejecutó durante 5 días. Esto nos permitió recolectar información sobre los nodos activos, los nodos que se han “bajado” de la red y la cantidad total de nodos almacenados en nuestra base de datos (esto quiere decir, los nodos activos o que alguna vez estuvieron activos en la red TOR).

A continuación, podemos ver que, en el inicio de la recolección, en nuestra base de datos tenemos aproximadamente 6000 nodos que coinciden con la cantidad de nodos activos ya que la base de datos está vacía. En el día 5, podemos apreciar que hemos acumulado aproximadamente 12000 nodos diferentes, lo que significa que tenemos 6000 nodos nuevos y diferentes a los del primer día:



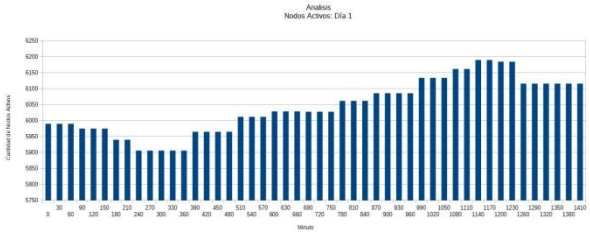
3. Nodos almacenados

Para observar el comportamiento de los nodos, graficamos la cantidad de nodos activos en el momento de ejecutar el script y vemos que en promedio siempre hay activos 6000 nodos.



4. Nodosactivos

Por último, observamos el comportamiento de los nodos activos cada 30 minutos del Día 1.



5. Nodos activos: Día 1

A través de este análisis de comportamiento de los nodos de la red TOR, resulta obvio que se necesita actualizar la lista de nodos permanentemente, así como llevar un registro de los nodos que estuvieron activos en la red.

5. Conclusiones

5.1. Plantear un esquema defensivo para ser aplicado a una Infraestructura Crítica, el mismo cobra una relevancia superior y esto está directamente asociado al impacto que puede tener un evento de seguridad en este tipo de estructuras.

5.2. El presente trabajo ha considerado los ataques que pueden suceder tanto desde afuera como desde dentro de la organización.

5.3. Pensar en técnicas forenses como esquema defensivo genera un nuevo espacio y nos amplía la suite de herramientas disponibles para hacer frente a ataques. Y adicionalmente abre un camino que no es común de ser utilizado y que permite aprender de eventos que su mitigación incrementará nuestro arsenal de defensa.

5.4. Aplicado en nuestro país podemos asumir que el uso de las redes anónimas será un vehículo hacia la ilegalidad y por tanto asumimos que será una plataforma especialmente elegida por el atacante el cual pretende mantener su anonimato.

5.5. El dinamismo que tiene este tipo de redes presenta un desafío adicional y pone al especialista en seguridad frente a la necesidad de aplicar su máxima creatividad para enfrentar los ataques que vienen desde estas redes.

5.6. Es esencial que todo el tráfico, tanto saliente como entrante, a nuestra red se encuentre filtrado y analizado, en particular no permitimos el tráfico de las IP de los nodos TOR que pasa por nuestro router.

6. Bibliografía

[1] Xiao Wang, Jinqiao Shi, Binxing Fang and Li Guo: “An Empirical Analysis of Family in the Tor Network”, IEEE ICC 2013 - Communication and Information Systems Security Symposium.

[2] Muhammad AliyuSulaiman and Sami Zhioua: “Attacking Tor through Unpopular Ports”, 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops.

[3] AbdelberiChaabane, Pere Manils, Mohamed Ali Kaafar: “Digging into Anonymous Traffic: a deep analysis of the Tor anonymizing network”, 2010 Fourth International Conference on Network and System Security.

[4] Yang ZHANG: “Effective Attacks in the Tor Authentication Protocol”, 2009 Third International Conference on Network and System Security. *Conference on Software Engineering (ICSE'03)*, Portland, OR, May 3-10 2003, pp. 125-137.

[5] Douglas Kelly, Richard Raines, Rusty Baldwin, Michael Grimaila, and Barry Mullins: “Exploring Extant and Emerging Issues in Anonymous Networks: A Taxonomy and Survey of Protocols and Metrics”, IEEE Communications Surveys & Tutorials, VOL. 14, NO. 2, Second Quarter 2012.

[6] Peipeng Liu, Jinqiao Shi, Lihong Wang, Xiao Wang, Qingfeng Tan: “IX-Level Adversaries on Entry- and Exit-Transmission Paths in Tor Network”, 2013 IEEE Eighth International Conference on Networking, Architecture and Storage.

[7] StjepanGroš, Marko Salki´c, Ivan Šipka: “Protecting TOR exit nodes from abuse”, MIPRO 2010, May 24-28, 2010, Opatija, Croatia.

[8] Eric Chan-Tin, Jiyoung Shin and Jiangmin Yu: “Revisiting Circuit Clogging Attacks on Tor”, 2013 International Conference on Availability, Reliability and Security.

[9] Florian Tschorschand BjörnScheuermann: “Tor is Unfair – and What to Do About It”, 36th Annual IEEE Conference on Local Computer Networks.

[10] Nicholas A. Fraser, Douglas J. Kelly, Richard A. Raines, Rusty O. Baldwin, and Barry E. Mullins: “Using Client Puzzles to Mitigate Distributed Denial of Service Attacks in the Tor Anonymous Routing Environment”, IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

[11] The Tor Project, Inc, “Tor project”: ”http://www.torproject.org

[12] Redesanónimas, I2P, FreeNet, TOR: “www.wikipedia.org".

[13] Comparación redes anónimas: “http://thehackerway.com/2012/02/08/preservando-el-anonimato-y-extendiendo-su-uso-comparacion-de-redes-anonimas-y-conclusiones-finales-parte-xxxxii/”

Desarrollo de una guía de auditoria para la verificación de la calidad del software crítico en sistemas ferroviarios

Eterovic, Jorge Esteban
Donadello, Domingo
Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas

Abstract

En la industria ferroviaria hay una gran cantidad de sistemas críticos que tienen productos de software. Éste software debe cumplir con los criterios de Confiabilidad, Disponibilidad, Mantenibilidad y Seguridad (RAMS, por sus siglas en inglés) establecidos en las normas internacionales, en particular en la norma EN 50126, utilizada en la Unión Europea.
Para ello, es necesario desarrollar un proceso de evaluación de la conformidad del software adquirido y/o desarrollado para el control ferroviario. La conformidad será de acuerdo con los requisitos establecidos en la norma EN 50126.
Considerando que ésta norma describe las fases del desarrollo del software, y que las organizaciones involucradas en el desarrollo de software deben implementar y usar un Sistema de Garantía de Calidad conforme con la Norma ISO 9000, es *altamente recomendable la certificación de conformidad con ésta norma.*
El objetivo de este trabajo es el desarrollo de una guía de auditoria para la verificación de la calidad del software crítico en sistemas ferroviarios, basada en la aplicación de la norma ISO 90003, que da las directrices para la aplicación de norma ISO 9001.
De esta manera, una empresa que desarrolle software para sistemas ferroviarios, al certificar la calidad del proceso con las normas ISO 9001 y 90003, estaría en condiciones de certificar su producto con la norma EN 50126.

Palabras Clave

Software crítico. RAMS. Calidad del software. Proceso de evaluación de calidad. Verificación de la calidad del software.

Introducción

La gran evolución del Transporte Ferroviario a nivel mundial en las últimas décadas hizo que la demanda de prestaciones y servicios sea cada vez mayor. En este sentido, los requisitos asociados a la Calidad y Seguridad Ferroviaria cada vez son más exigentes.
Calidad y Seguridad están directamente relacionados y marcan el nivel de confianza que ofrece un sistema. Los objetivos de Seguridad y Disponibilidad sólo pueden alcanzarse cumpliendo los requisitos de Confiabilidad y Mantenibilidad.
Como en la industria ferroviaria hay una gran cantidad de sistemas críticos con un alto contenido de software, es necesario desarrollar un proceso de verificación de la calidad de dicho software crítico a efectos de asegurar la Confiabilidad, la Disponibilidad, la Mantenibilidad y la Seguridad, representadas por las siglas RAMS [1], acrónimo de Reliability, Availability, Maintainability and Safety.
RAMS representa un indicador, tanto cualitativo como cuantitativo, del grado de confianza que ofrece un sistema para comportarse de acuerdo a la funcionalidad especificada, de forma segura y con una alta disponibilidad.
En la Unión Europea se han adoptado los requisitos establecidos en las normas CENELEC (Comité Europeo de Normalización Electrotécnica) [2] en materia de RAMS ferroviaria, y la necesidad de mejora de procesos exigida por dichas normas, en especial en el proceso de desarrollo general del producto.
La normativa CENELEC está compuesta por tres normas de la familia EN, y son las EN 50126 [3], EN 50128 [4] y EN 50129 [5].
El grado de integridad de las funciones de seguridad se mide y tabula mediante el SIL, Nivel de Integridad de la Seguridad (Safety Integrity Level) [6]. El SIL mide y tabula la confianza que nos merece que una función

de seguridad se vaya a ejecutar adecuadamente. Es una unidad de medida para cuantificar la reducción del riesgo.
Por ello, las organizaciones involucradas en el desarrollo del software crítico, deben implementar un Sistema de Garantía de Calidad. El concepto de “Calidad” es muy ambiguo, y también lo es el de “Calidad de Producto de Software” [7], [8], [9], [10]. Una de las definiciones más aceptada es [9]: "Calidad es la totalidad de las características del producto que influyen en la capacidad del producto para satisfacer las necesidades explicitas o implícitas".
En el marco del sistema que lo contiene, el software es una herramienta, y las herramientas tienen que ser seleccionadas por su calidad y pertinencia.
El software determina el rendimiento de los procesos a los que brinda apoyo, impactando en el desempeño del sistema global, por lo tanto es importante para la calidad de este sistema. Por lo tanto, evaluar con máxima objetividad las características de calidad deseadas, no es una tarea menor, y debe dedicarse mucho esfuerzo.
Con la creciente sofisticación de los productos de software y su uso en áreas críticas como en medicina, cirugía, aeronavegación, militar, ferroviaria etc., se han intensificado las actividades de evaluación de la calidad de los productos y artefactos de software [11].
El objetivo de este trabajo es el desarrollo de un guía de auditoria para la verificación de la calidad del software crítico en sistemas ferroviarios basado en la aplicación de la norma IRAM-ISO 90003 [12], que da las directrices para la aplicación de norma IRAM-ISO 9001 [13].

Elementos del Trabajo y metodología

Las normas proporcionan una serie de requisitos que se deben cumplir en las fases de desarrollo, implantación y mantenimiento del software crítico destinado a aplicaciones de control y protección de ferrocarriles. Se definen los requisitos relativos a la estructura organizativa, a la relación entre organizaciones y a la división de responsabilidades relativas a las actividades de desarrollo, implantación y mantenimiento. Se proporcionan además los criterios relativos a la calificación, experiencia y competencia del personal.
El concepto clave es el de los niveles de integridad de seguridad del software. Se identifican cinco niveles de integridad de seguridad del software, siendo 0 el nivel mínimo y 4 el máximo. Cuanto más peligrosas sean las consecuencias de un fallo del software, mayor será el nivel requerido de integridad de seguridad del mismo.
Se deben identificar técnicas y medidas para los cinco niveles de integridad de seguridad del software.
Como resultado de este trabajo se muestran las técnicas y medidas requeridas para los niveles 0 a 4 de integridad de seguridad del software. Las técnicas requeridas para el nivel 1 son las mismas que para el nivel 2 y las técnicas requeridas para el nivel 3 son las mismas que para el nivel 4. Lo qué no se puede indicar es qué nivel de integridad de seguridad del software es apropiado para un riesgo determinado. Esta decisión dependerá de muchos factores, incluyendo la naturaleza de la aplicación, del grado en que otros sistemas llevan a cabo funciones de seguridad y de factores sociales y económicos.
A medida que se descompone la especificación en un diseño que incluye sistemas y componentes relacionados con la seguridad, se produce una nueva asignación de niveles de integridad de seguridad. Finalmente, se llega a los niveles de integridad de seguridad requeridos para el software.
De todos modos ni la aplicación de métodos para garantizar la calidad (como las medidas para evitar y detectar errores) ni la aplicación de soluciones de software tolerante a errores, pueden garantizar la seguridad absoluta del sistema. No hay manera conocida para demostrar la ausencia

de errores en un software relacionado con la seguridad razonablemente complejo, especialmente la ausencia de errores de especificación y diseño.
La Especificación de Requisitos de Seguridad del Sistema identifica todas las funciones de seguridad asignadas al software y determina el nivel de integridad de seguridad del sistema para dichas funciones. En la figura 1 se muestran las etapas funcionales:

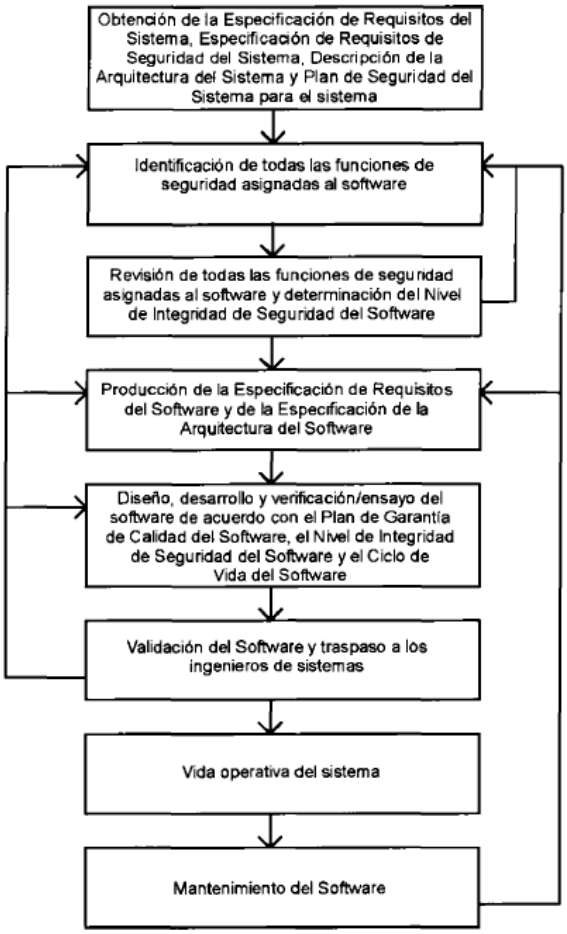


Figura 1- Etapas funcionales

Se debe seleccionar un modelo de ciclo de vida para el desarrollo del software y se debe detallar en el Plan de Garantía de Calidad del Software, cuyo objetivo es identificar, supervisar y controlar toda actividad, tanto técnica como de gestión, necesaria para garantizar que el software alcanza la calidad requerida.
Es necesario para proporcionar la defensa cualitativa necesaria contra errores sistemáticos y para garantizar que se puede establecer una pista de auditoría que permita realizar las actividades de verificación y validación de forma efectiva.
Las organizaciones involucradas en el desarrollo del software deben implementar y usar un Sistema de Garantía de Calidad conforme con la Norma IRAM-ISO 9000 [14], para satisfacer los requisitos de esta norma europea. Es altamente recomendable la certificación de conformidad con la Norma IRAM-ISO 9001.
Se debe redactar un Plan de Garantía de Calidad del Software, donde se deben especificar los siguientes elementos:

a) Definición del modelo del ciclo de vida:
   1) actividades y tareas básicas compatibles con los planes, por ejemplo, el Plan de Seguridad que se ha establecido a nivel del sistema;
   2) criterios de entrada y salida de cada actividad;
   3) entradas y salidas de cada actividad;
   4) principales actividades de calidad;
   5) entidad responsable de cada actividad.
b) Estructura de la documentación.
c) Control de la documentación:
   1) roles de aquellos implicados en su redacción, control y aprobación;
   2) campo de aplicación de la distribución;
   3) archivo.
d) Seguimiento y trazabilidad de las desviaciones.
e) Métodos, medidas y herramientas para la garantía de calidad en función de los niveles de integridad de seguridad asignados.
f) Justificaciones de que cada combinación de técnicas o medidas seleccionada es apropiada para cada nivel definido de integridad de seguridad del software.

Cierta información requerida en el Plan de Garantía de Calidad del Software puede aparecer en otros documentos, como en un Plan de Gestión de la Configuración del Software, un Plan de Mantenimiento, un Plan de Verificación del Software y un Plan

de Validación del Software separados. Los apartados del Plan de Garantía de Calidad del Software deben proporcionar la referencia de los documentos en los que aparece la información. En cualquier caso, se debe especificar el contenido de cada apartado del Plan de Garantía de Calidad del Software, ya sea directamente o mediante referencia a otro documento.
Finalmente se debe redactar un Informe de Verificación de la Garantía de Calidad del Software, pudiéndose usar como base la guía de auditoria que se desarrolla en este trabajo.

Ciclo de vida del sistema

El ciclo de vida del sistema es una secuencia de fases, cada una de las cuales contiene tareas que abarcan la vida completa de un sistema desde su concepto inicial hasta la retirada del servicio y la eliminación. El ciclo de vida proporciona una estructura para la planificación, la gestión, el control y la supervisión de todos los aspectos de un sistema, incluida la RAMS, a medida que el sistema avanza a través de sus fases, con el fin de entregar el producto adecuado al precio correcto dentro del plazo acordado.
Un ciclo de vida de un sistema, adecuado en el contexto de una aplicación ferroviaria, se muestra en la figura 2.
La figura 2 representa el ciclo de vida del sistema en el modelo en "V", donde la rama descendente (lado izquierdo) se llama generalmente Desarrollo y consiste en un proceso de perfeccionamiento que finaliza con la fabricación de componentes del sistema. La rama ascendente (lado derecho) está relacionada con el montaje, la instalación, la recepción y el funcionamiento de todo el sistema.
La representación en "V" supone que las actividades de aceptación están intrínsecamente vinculadas a las actividades de desarrollo, dado que lo que es realmente diseñado tiene que ser finalmente comprobado en relación con los requisitos.

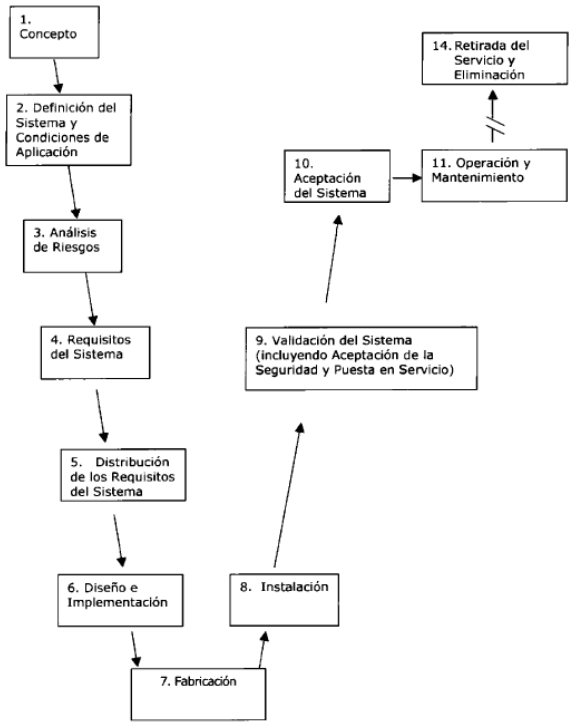


Figura 2 – Ciclo de vida (representación en V)

Las actividades de validación correspondientes a la aceptación en varias etapas de un sistema, se basan en la especificación del sistema y deben ser planificadas en las primeras etapas; es decir, empezando en las fases correspondientes de desarrollo del ciclo de vida, como se muestra en la figura 3.

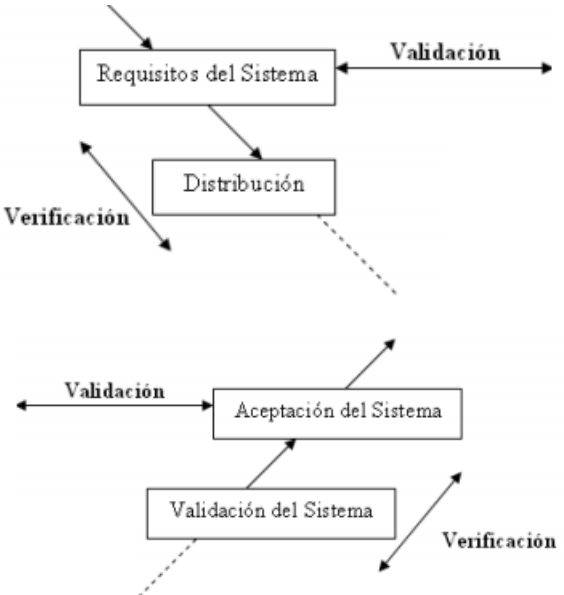


Figura 3 - Validaciones

Se muestran por separado las tareas de verificación y validación dentro del ciclo de

vida. El objetivo de la verificación consiste en demostrar que, para las entradas de información específicas, las entregas de cada fase cumplen, en todos los aspectos, los requisitos de dicha fase. El objetivo de la validación consiste en demostrar que el sistema de que se trate, en cualquier momento de su desarrollo y después de su instalación, cumple sus requisitos en todos los aspectos.

Las tareas de verificación están incluidas dentro de cada fase del ciclo de vida. Si buscamos el aseguramiento del sistema en el contexto RAMS, las tareas de verificación y validación forman parte integral de la demostración global de aseguramiento de los sistemas.

Para la realización de éstas tareas se debe definir el Rol "Validador", cuyas responsabilidades serán:

1. debe desarrollar una comprensión del sistema de software dentro del entorno previsto de aplicación;
2. debe desarrollar un plan de validación y especificar las tareas y actividades esenciales para la validación del software y ponerse de acuerdo sobre este plan con el evaluador;
3. debe revisar los requisitos del software en relación a su uso/entorno previsto;
4. debe revisar el software en relación a los requisitos del software de forma que se garantice que se cumplen todos ellos;
5. debe evaluar la conformidad del proceso del software y del software desarrollado en relación a los requisitos de la normativa incluyendo el SIL asignado;
6. debe revisar la corrección, coherencia y adecuación de la verificación y de los ensayos;
7. debe comprobar la corrección, coherencia y adecuación de los casos de ensayo y de los ensayos realizados;
8. debe garantizar que se realizan todas las actividades del plan de validación;
9. debe revisar y clasificar todas las desviaciones en términos de riesgo (impacto), registrarlas y comunicarlas al organismo competente de la gestión de las modificaciones para su evaluación y toma de decisiones;
10. debe proporcionar una recomendación sobre la idoneidad del software para su uso previsto e indicar cualquier restricción de la aplicación según sea apropiado;
11. debe registrar las desviaciones a partir del plan de validación;
12. debe realizar auditorías, inspecciones o revisiones del proyecto global (como instancias del proceso de desarrollo genérico) según sea apropiado, en varias fases del desarrollo;
13. debe revisar y analizar los informes de validación relativos a aplicaciones previas según sea apropiado;
14. debe revisar si las soluciones desarrolladas son trazables hasta los requisitos del software;
15. debe garantizar que se revisan los registros de situaciones peligrosas asociadas y los casos de no conformidad y que se resuelven todas las situaciones peligrosas de manera adecuada, ya sea mediante medidas que las eliminen o con medidas de control/transferencia de los riesgos;
16. debe desarrollar un informe de validación y
17. debe expresar su acuerdo/desacuerdo sobre la versión del software publicada

Resultados

Para cada fase de este ciclo de vida, se han definido las principales tareas a ser auditadas. Se resumen en la siguiente tabla (figura 4):

Fase del ciclo de vida	**Tareas a ser auditadas**
1.Concepto	Ámbito y propósito del proyecto Ferroviario. Definición del concepto del proyecto ferroviario. Análisis Financiero y estudios de viabilidad. Existencia del equipo de gestión. Las implicaciones de seguridad del proyecto. La política y objetivos de la seguridad.

Fase ciclo del de vida	Tareas a ser auditadas
2.Definición del sistema y condiciones de aplicación	El perfil de la misión del sistema. La descripción del sistema. La estrategia de Operación y Mantenimiento. Las condiciones de operación y mantenimiento. La influencia de las restricciones de la infraestructura existente. El análisis preliminar de las amenazas. El plan de seguridad. Definición de las condiciones de operación y mantenimiento a largo plazo. Identificar la influencia en RAM de las restricciones en la infraestructura existente.
3.Análisis de riesgos	El Análisis de riesgos relacionado con el proyecto. El análisis amenazas y riesgos de la seguridad del sistema. El registro de las amenazas. La evaluación de riesgos.
4.Requisitos del Sistema	El análisis de requisitos. Especificaciones del sistema. Especificación el entorno. Los criterios de Demostración y Aceptación del sistema. El plan de Validación. Los requisitos de Gestión, Calidad y Organización. El procedimiento de control de cambios. Los requisitos de Seguridad del Sistema. Los criterios de aceptación de la Seguridad. Los requisitos relacionados con la segundad Funcional. La Gestión de Seguridad.
5.Distribución de los Requisitos del Sistema	Especificación de los requisitos de los subsistemas y componentes. Especificación de los criterios de aceptación de subsistemas y componentes. Los requisitos de seguridad de los subsistemas y componentes. Los criterios de aceptación de seguridad de los subsistemas y componentes. El Plan de Seguridad del Sistema.
6.Diseño e implementación	La planificación. El Diseño y desarrollo. El análisis del diseño y pruebas. La Verificación del diseño- La implementación y validación. El diseño de los recursos de apoyo logísticos. El Registro de amenazas. El Análisis de amenazas y evaluación de riesgos. La Gestión de la Seguridad. El Control de subcontratos y proveedores. Un Caso de Seguridad.
7.Producción	El plan de producción. La fabricación de código. La fabricación y prueba del montaje de componentes. La documentación. La capacitación. La implementación del plan de seguridad. El uso del registro de amenazas.
8.Instalación	El montaje del sistema La instalación del sistema. El programa de instalación. La implementación del programa de instalación.
9.Validación del sistema	La puesta en servicio. El período de pruebas de operación. La capacitación. El programa de puesta en servicio. La implementación del programa de puesta en servicio. El Caso de Seguridad específico de la aplicación.
10.Aceptación del sistema	Los procedimientos de aceptación, basados en criterios de aceptación. La recopilación de las pruebas para la aceptación. La entrada en servicio. El periodo de pruebas de operación. El Caso de Seguridad específico de la aplicación.
11.Operación y mantenimiento	La operación del sistema a largo plazo. El mantenimiento. La capacitación en el mantenimiento centrado en segundad. El control de la ejecución de segundad y mantenimiento del registro de las amenazas.
12.Supervisión de la ejecución	La recopilación estadística de la ejecución operacional. La adquisición, el análisis y la evaluación de los datos. La recopilación, el análisis, la evaluación el uso de las estadísticas de Seguridad y ejecución.
13.Modificación y realimentación	Los procedimientos de cambio de requisitos. Los procedimientos de modificación y realimentación. Las implicaciones de Seguridad para la modificación y realimentación.
14.Retirada de servicio y eliminación	El plan de retirada de servicio y eliminación. La retirada de servicio. La eliminación. El plan de Seguridad. El análisis de amenazas y la evaluación de riesgos. La implementación del plan de Seguridad.

Figura 4 – Tareas a auditar

Además de lo expresado, en todas las fases se deberían auditar las siguientes tareas:

- Control de Cambios
- Gestión de la Configuración
- Verificación y Validación
- Análisis de riesgos

Los procesos del sistema de gestión se deben evaluar de acuerdo a la norma IRAM-ISO 9000, teniendo que considerar los siguientes temas:

1. Identificación y comunicación de los requisitos del cliente (5.4, 7.2);
2. Identificación de la vinculación (secuencia e interrelación) con otros procesos (4.1);
3. Identificación de los objetivos del proceso (7.1);
4. Definición de responsabilidad y autoridad (5.5.1);
5. Competencia del personal (6.2);
6. Adecuación de recursos y ambiente de trabajo (6.3, 6.4, 7.1);
7. Adecuación de la documentación que describe las prácticas de operación (Cap. 7);
8. Seguimiento del desempeño del proceso y control de no conformidades (8.3, 8.4);
9. Aplicación de acciones correctivas y preventivas (8.5.2, 8.5.3);
10. Evidencia de mejora continua (8.5.1);
11. Disponibilidad de registros (4.2.4, 7.1d);
12. Divulgación de la certificación. Uso de logos. (Aplica sólo en auditorías de seguimiento o recertificación) y
13. Gestión del cumplimiento de requisitos legales del producto.

Discusión

Si bien las normas CENELEC nos permiten identificar los requerimientos para la verificación de la calidad del software crítico en sistemas ferroviarios de manera que en la norma EN 50126 se define el ciclo de vida, en la norma EN 50128 las técnicas de software y en la EN 50129 las técnicas de hardware, podríamos plantear y un requisito para cada nivel de integridad de seguridad del software (SIL) para cada técnica o medida de la Garantía de la Calidad del Software en función del Nivel de Integridad de la Seguridad (SIL) de la siguiente manera:

Técnica/Medida	**SIL O**	**SIL 1**	**SIL 2**	**SIL 3**	**SIL 4**
Acreditada según la Norma ISO 9001	R	HR	HR	HR	HR
Conforme con la Norma ISO 9001	M	M	M	M	M
Conforme con la Norma ISO/IEC 90003	R	R	R	R	R
Sistema de Calidad de la Compañía	M	M	M	M	M
Gestión de la Configuración del Software	M	M	M	M	M
Listas de Comprobación	R	HR	HR	HR	HR
Trazabilidad	R	HR	HR	M	M
Registro y Análisis de Datos	HR	HR	HR	M	M

Donde los requisitos para los niveles de integridad de seguridad del software 1 y 2 son los mismos para cada técnica. Del mismo modo, cada técnica tiene los mismos requisitos en los niveles de integridad de seguridad del software 3 y 4. Estos requisitos pueden ser:

- M: mandatorio;
- HR: altamente recomendable:
- R: recomendable

La combinación de técnicas o medidas se deberán incluir en el Plan de Garantía de Calidad del Software.

Conclusión

Una conclusión importante a la que se arribó, luego del análisis de la guía de auditoria para la verificación de la calidad del software crítico en sistemas ferroviarios, es que el Nivel de Integridad de la Seguridad (SIL) depende de la integridad ante Fallos Sistemáticos y ante Fallos Aleatorios.

A los Fallos Aleatorios, inherentes a la fiabilidad de los equipos, fallos debidos a la fatiga, deterioro por el tiempo de vida, etc. no los hemos considerado, ya que la

investigación se basó en la verificación de la calidad del software.

Los Fallos Sistemáticos son causados por errores humanos durante el diseño, fabricación, verificación, validación o mantenimiento del software.

Una forma de minimizar fallos sistemáticos es utilizar un adecuado ciclo de vida y técnicas de software y hardware adecuadas para el diseño y desarrollo del producto.

Para cada nivel SIL, las normas EN 5012X son más o menos exigentes, determinando la forma de minimizar los fallos sistemáticos. En los Fallos sistemáticos las características del ciclo de vida, de las tareas a realizar, de las técnicas software y hardware a aplicar etc. deberán ser más exigente para sistemas con funciones de un SIL elevado.

Referencias

1. Zárate Fraga, Marta. Análisis RAMS. Proyecto Fin de Carrera. Universidad Carlos III de Madrid; Febrero de 2012.
2. CENELC: Comité Européen de Normalisation Electrotechnique-. http://www.cenelec.eu.
3. Norma EN 50126. Aplicaciones Ferroviarias. Especificación y demostración de la fiabilidad, la disponibilidad, la mantenibilidad y la seguridad (RAMS). 2005.
4. Norma EN 50128. Aplicaciones Ferroviarias. Sistemas de comunicaciones, señalización y procesamiento. Software para sistemas y protección del ferrocarril. 2012.
5. Norma EN 50129. Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Sistemas electrónicos relacionados con la seguridad para la señalización. 2005.
6. Mark Charlwood, Shane Turner and Nicola Worsell. A methodology for the assignment of safety integrity levels (SILs) to safety-related control functions implemented by safety-related electrical, electronic and programmable electronic control systems of machines. Health and Safety Executive. 2004
7. Brosseau, Jim. Software Quality Attributes: Following All the Steps, Clarrus Consulting Group Inc. Noviembre de 2010.
8. Kitchenham, B., Lawrence, Pfleeger, S.L.: Software Quality: The Elusive Target, Software, IEEE (Volume:13, Issue: 1, pp. 12-21), Enero de 1996.
9. Dromey, R.G.: Cornering the Chimera, Software, IEEE (Volume: 13, Issue: 1, pp. 33-43), Enero de 1996.
10. Wallace, D. and Reeker L., Software Quality,"in Guide to the Software Engineering Body of Knowledge SWEBOK, A. Abram and P. Bourque, Eds.: IEEE, pp. 165 - 184, 2001.
11. Thomas E. Murphy, Nathan Wilson. Gartner: Magic Quadrant for Integrated Software Quality Suites. Julio de 2013.
12. Norma IRAM-ISO 9001. Sistemas de gestión de la calidad. Requisitos. 2008.
13. Norma IRAM-ISO 90003. Ingeniería del software. Guía de aplicación de la ISO 9001:2000 al software. 2004.
14. Norma IRAM-ISO 9000. Sistemas de gestión de la calidad. Fundamentos y vocabulario. 2000.

Datos de Contacto:

Mag. Jorge Esteban Eterovic.
Universidad Nacional de La Matanza.
Departamento de Ingeniería e Investigaciones Tecnológicas.
Florencio Varela 1903, (B1754JEC) San Justo, Prov. de Buenos Aires, Argentina.
Tel: (54 11) 4480-8900
E-mail. jeterovic@hotmail.com
jeterovic@ing.unlam.edu.ar

Mag. Domingo Donadello.
Universidad Nacional de La Matanza.
Departamento de Ingeniería e Investigaciones Tecnológicas.
Florencio Varela 1903, (B1754JEC) San Justo, Prov. de Buenos Aires, Argentina.
Tel: (54 11) 4480-8900
E-mail.ddonadel@ing.unlam.edu.ar