

Evaluación de Interfaces Gráficas de Usuario Usando LSP

M. Claudia Albornoz, Enrique Miranda, Mario Berón
Universidad Nacional de San Luis, San Luis, Argentina
{albornoz,eamiranda,mberon}@unsl.edu.ar

Abstract

La Interfaz Gráfica de Usuario (GUI por su nombre en inglés, Graphical User Interface) es la parte fundamental de cualquier aplicación, ya que es lo primero que ve el usuario; es donde comienza la interacción entre el hombre y la computadora. El diseño de la GUI no es una tarea fácil, se deben tener en cuenta, no sólo los requerimientos del usuario sino también una serie de características fundamentales para un buen desarrollo de la misma. En muchas ocasiones, la Interfaz Gráfica de una aplicación es la que determina si la misma será o no utilizada para resolver los problemas para los cuales dicha aplicación fue diseñada. La observación previamente mencionada permite vislumbrar la importancia de disponer de métodos, técnicas y herramientas que posibiliten llevar a cabo una evaluación eficiente, fidedigna y automatizada de GUIs. En este artículo se describe: i) Un método de evaluación multi-criterio muy útil para evaluar Interfaces Gráficas: LSP [4], ii) Una herramienta que implementa el método previamente mencionado y iii) La forma de llevar a cabo el proceso de evaluación. Todos los elementos mencionados previamente proporcionan un marco de trabajo propicio que ayuda al Ingeniero de Software a: Determinar la calidad de las GUIs y seleccionar la aplicación adecuada para la resolución de problemas específicos. Para finalizar este resumen, es importante mencionar que la aproximación descrita en este artículo no es estática sino que la misma permite la personalización del proceso de evaluación dependiendo del nivel de precisión que se necesite.

Palabras Claves

Interfaz Gráfica de Usuario (GUI), LSP (Logic Scoring of Preference), NESSy, Características deseables de GUI, Evaluación de GUI.

Introducción

La Interfaz Gráfica de Usuario (GUI por su nombre en inglés Graphical User Interface) ha sido ampliamente adoptada en los sistemas de software hoy en día. Las GUIs han evolucionado a pasos agigantados desde las primeras Interfaces de Usuario basadas en comandos,

conocidas como modo texto o consola, las que aún hoy se pueden usar en las terminales Unix o al abrir la consola del MSDOS. En este caso, la Interfaz espera que el usuario sepa introducir los comandos a ser procesados y así devolver resultados.

La Interfaz de Usuario es un componente crítico de la aplicación, ya que a través de ella, el usuario interactúa con el sistema. Una Interfaz de Usuario debe transmitir o hacerle saber al usuario todo lo que es capaz de hacer el sistema; o dicho de otra manera, es la carta de presentación del producto. El usuario está interesado en la realización de una serie de tareas, para lo cual utiliza la aplicación en cuestión. Es la Interfaz de Usuario la que debe facilitar la concreción de dichas tareas de manera rápida, intuitiva y eficaz.

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, fuentes, etc. los cuales representan funciones, acciones e información’ [1].

El objetivo de la GUI es ofrecer un entorno visual sencillo y amigable, para interactuar con las correspondientes aplicaciones. Las primeras interfaces trabajaban ingresando sentencias (algunas complejas, otras no tanto) en la línea de comandos del Sistema Operativo, lo que requería un usuario experto, con amplios conocimientos informáticos. En la actualidad las GUI han evolucionado de manera muy notoria, de hecho, por lo general, la Interfaz de Usuario comprende el 60% de las líneas de código del total de un sistema de información interactivo. Es por este motivo que con el paso de los años

se ha desarrollado toda una *Ingeniería de Interfaces de Usuario*.

Cualquier usuario inexperto, sin conocimientos previos, puede operar un sistema si la Interfaz está bien diseñada y construida. Una Interfaz mal diseñada obstaculiza la 'usabilidad', es decir aquella característica 'que hace que la aplicación sea fácil de utilizar y fácil de aprender'. La primera característica se refiere a que la aplicación realiza lo que se supone debe hacer, de manera eficiente e intuitiva. La segunda 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 [2].

El diseño y construcción de la Interfaz es tan importante como la construcción de la aplicación en sí. Dado el gran avance tecnológico, las computadoras forman parte de la mayoría de los hogares, es decir, casi todas las personas tienen acceso a una computadora. Por lo tanto, para que todos puedan operar con ellas, se debe contar con interfaces que posean las siguientes características:

- *Amigable*: fácil de entender; que el usuario pueda comenzar a trabajar en un entorno en el que se sienta cómodo.
- *Efectiva*: hace lo que se espera que haga.
- *Eficiente*: realiza las tareas en el tiempo necesario y sin errores.

Estas son, entre muchas otras, características deseables de las GUIs. Las cuales, algunas de ellas se pueden cuantificar y medir [3].

Como fue mencionado en el resumen de este artículo, la evaluación de GUIs es una tarea relevante porque permite mejorar su calidad; identificar problemas de diseño, conceptuales, de programación, etc.; seleccionar la herramienta que posee la mejor GUI para abordar problemas específicos; entre otras tantas posibilidades. LSP (Logic Scoring of Preference) es un método de evaluación multi-criterio que puede ser adaptado a diferentes contextos de evaluación y permite la realización de evaluaciones precisas. LSP es un método automatizable que ha sido ampliamente

usado por la comunidad de Ingeniería de Software, para evaluar diferentes elementos comúnmente utilizados en la vida cotidiana del ingeniero de software. Por las razones antes mencionadas LSP será utilizado para realizar las evaluaciones de GUIs mencionadas con anterioridad.

El artículo está organizado de la siguiente manera. En la segunda sección se describe el método multi-criterio LSP. En la tercera y cuarta sección se instancian las componentes de LSP con el dominio de estudio, es decir las GUIs. En la quinta sección se presenta NESSy (New Evaluation Software System), una herramienta que implementa el método LSP. En la sexta sección se presenta un caso de estudio que permite ver la utilidad de la aproximación desarrollada en este artículo. Finalmente, en la última sección se exponen las conclusiones y trabajos futuros del artículo.

Método LSP (*Logic Scoring of Preference*)

LSP (*Lógica Scoring of Preference*) es un método de evaluación multi-criterio que se basa en la definición de: i) un Árbol de Criterios, ii) Funciones de Criterios Elementales (o simplemente Criterios Elementales) y iii) una Estructura de Agregación [4]. LSP es útil para analizar, comparar y seleccionar la mejor opción de un conjunto finito de alternativas (en el caso de esta investigación, las GUIs). LSP consta de una serie de pasos que deben ejecutarse adecuadamente para llevar a cabo la evaluación [5,6,7]. Dichos pasos se explican en los siguientes apartados.

1. Árbol de Criterios

El Árbol de Criterios (AC) contiene las características deseables que los elementos a evaluar deben poseer. Con el objetivo de desarrollar una lista completa de criterios, se aplica un proceso de descomposición jerárquica. Al final de este proceso se obtiene una lista de atributos medibles. En el primer caso, se definen las características de alto nivel. A

continuación, se descomponen en sub-características y así sucesivamente. Este proceso se repite hasta obtener atributos atómicos directamente cuantificables (hojas del árbol). El resultado de esta tarea es un árbol que contiene las principales características que los elementos, bajo evaluación, deben poseer.

2. Criterios Elementales

En esta etapa se debe definir, por cada atributo (hoja) del AC, una función denominada Criterio Elemental (CE). Cada CE mapea el valor correspondiente al atributo en otro contenido en el intervalo [0,1] o [0,100]. Este valor se denomina Preferencia Elemental y representa el nivel de satisfacción correspondiente al atributo en cuestión, para el elemento evaluado. De esta manera, 0 representa una situación en que el atributo no cumple los requisitos en absoluto, y 1 (o 100) significa que el requisito está totalmente satisfecho. Los CE se pueden clasificar como: *Absolutos* o *Relativos*.

Un CE Absoluto se utiliza para determinar la preferencia absoluta de algún atributo. Un CE Relativo se emplea para establecer los indicadores relativos de los elementos que se quieren comparar. Los CE's relativos no se utilizan con frecuencia para este tipo de evaluaciones [11].

Existen diferentes tipos de CE's Absolutos:

- Variable Continua:
 - * **Multivariable:** La Preferencia Elemental se calcula utilizando una función específica. Esta función recibe parámetros de entrada y devuelve la Preferencia Elemental correspondiente al atributo bajo evaluación. Es decir, se calcula a través de una fórmula.
 - * **Directo:** La Preferencia Elemental toma un valor ingresado directamente por el evaluador.
- Variable Discreta:
 - * **Multinivel:** La Preferencia Elemental puede tomar un valor de un conjunto de valores discretos

previamente definidos. Dicho conjunto puede estar constituido por dos o más valores.

3. Estructura de Agregación

Las Preferencias Elementales que resultan de aplicar los CE's a los valores correspondientes a los atributos mensurables, deben agregarse a fin de obtener la Preferencia Global. Esta representa la satisfacción total del elemento evaluado, con respecto a todos los atributos definidos.

Para poder calcular la Preferencia Global, LSP cuenta con un conjunto de operadores lógicos (o funciones) propios del método. Cada operador recibe un conjunto de Preferencias Elementales y sus correspondientes pesos como entrada. Los pesos representan la importancia relativa de cada Preferencia. Cada operador devuelve una nueva Preferencia resultante de la agregación. Todas las Preferencias se vuelven a agregar en el siguiente nivel de la Estructura de Agregación (EA). Este proceso se repite hasta que se alcanza un operador final que calculará la Preferencia Global. El cálculo que realiza cada operador está determinado por la siguiente ecuación:

$$E = (\omega_1 e_1^r + \omega_2 e_2^r + \dots + \omega_k e_k^r)^{1/r}$$

$$-\infty \leq r \leq +\infty$$

$$0 \leq \omega_i \leq 1 \text{ y } i = 1 \dots k$$

Figura 1: Como se puede notar, E es un esquema general de instanciación que da lugar a nuevos operadores, dependiendo del valor elegido para r .

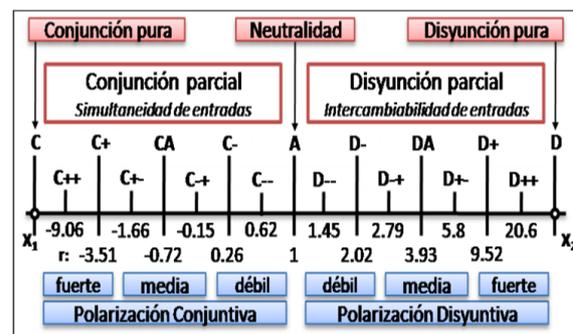


Figura 2: Funciones de Conjunción-Disyunción Generalizadas: Grados de Conjunción-Disyunción para dos operandos (x_1 y x_2).

La Tabla 1 (Tabla de valores) muestra los valores más relevantes para r , teniendo en cuenta los n valores de entrada de la función [15]. Por ejemplo, si el operador en cuestión es D- y recibe tres valores de entrada, entonces el valor de r en la fórmula precedente es 2,19. Para ser más claro, r representa el grado conjunción-disyunción de cada operador (ver Figura 2). Se dice que r genera varias funciones conocidas como funciones generalizadas disyuntivas conjuntivas (CDG) [8]. Estas funciones son los operadores utilizados para agregar las preferencias elementales.

Nombre Operación	Símbolo	r			
		n=2	n=3	n=4	n=5
Disyunción	D	$+\infty$	$+\infty$	$+\infty$	$+\infty$
Cuasi Disyunción Fuerte	D+	9.52	11.09	12.28	13.16
Cuasi Disyunción	DA	3.83	4.45	4.82	5.09
Cuasi Disyunción Débil	D-	2.02	2.19	2.30	2.38
Media Aritmética	A	1.00	1.00	1.00	1.00
Cuasi Conjunción Débil	C-	0.26	0.20	0.17	0.16
Cuasi Conjunción	CA	-0.72	-0.73	-0.71	-0.67
Cuasi Conjunción Fuerte	C+	-3.51	-3.51	-2.18	-2.61
Conjunción	C	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Tabla 1: Valores de Funciones de Conjunción-Disyunción Generalizada

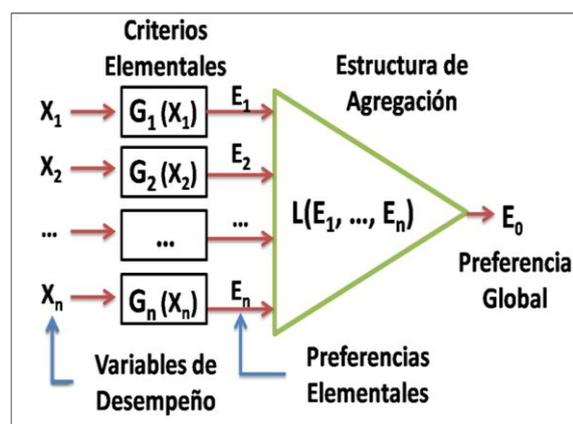


Figura 3: Vista Genérica del Método LSP

4. Proceso de Evaluación

El proceso de evaluación se lleva a cabo asignando valores para cada atributo y cada elemento sometido a evaluación.

La Figura 3 muestra una representación del método de evaluación LSP. A través de la misma se puede notar que todas las *Preferencias Elementales*, resultantes de aplicar los *Criterios Elementales* a las Variables de Desempeño (X_1, X_2, \dots, X_n), también llamados *Atributos*, serán computadas por los operadores definidos en la *Estructura de Agregación*. El resultante de dicho cómputo (representado por $L(E_1.. E_n)$), será la *Preferencia Global* del elemento evaluado.

Definición del Árbol de Criterios para GUI

En la sección anterior se explicaron cada una de las etapas o fases del método LSP. En ésta sección se define el Árbol de Criterios que se utilizará para la evaluación de Interfaces Gráficas de Usuario.

Para poder diseñar y desarrollar una GUI, se deben tener en cuenta una serie de características, cualidades, particularidades, las cuales no sólo están relacionadas con el buen funcionamiento de la Interfaz sino también con la estética y el despliegue de sus elementos [9]. Dichas características se describen a continuación:

➤ **Características de Ayuda:** toda GUI debe brindar ayuda y asistencia al usuario. Esta característica se puede medir a través de los siguientes atributos:

- Claridad en los mensajes de ayuda.
- Existencia de ayuda (Help).
- Existencia de ayuda On Line.
- Existencia de un Asistente si el usuario no sabe que responder.

➤ **Robustez:** significa que la GUI debe saber responder ante diversas situaciones; ya sean inesperadas o destructivas (eliminar). Ésta característica se puede descomponer en sub-características:

* **Manejo Errores:** hace referencia al comportamiento de la GUI ante el ingreso de información errónea. Los atributos correspondientes a esta sub-característica son los siguientes:

- Recuperación de estados anteriores.

- Prevención de errores.
 - Corrección de errores.
- * **Alertas:** la interfaz debe avisar y alertar al usuario ante situaciones destructivas. El atributo utilizado para medir ésta sub-característica es el siguiente:
- Aviso ante situaciones destructivas.
- **Funcional:** la interfaz debe permitir realizar más de una tarea al mismo tiempo (por ejemplo, imprimir y buscar un archivo); ofrecer atajos (se refiere al uso de combinación de teclas, por ejemplo *Ctrl+c* para copiar), dar respuesta inmediata al accionar del usuario (mover el cursor, resaltar la opción elegida, comunicar éxito o fracaso de la tarea; en especial ante sobrecarga de tareas), ofrecer actualizaciones, entre otras tantas funcionalidades importantes. Los atributos definidos para evaluar esta característica son:
- Multitarea (posibilidad de realizar más de una tarea a la vez).
 - Atajos (posibilidad de usar atajos).
 - Respuesta ante sobrecarga.
 - Actualizaciones.
- **Efectividad:** la Interfaz es efectiva si el usuario encuentra entre las opciones las tareas que necesita realizar, sintiendo conformidad al utilizar la misma. Para evaluar esta característica se utilizará el siguiente atributo:
- Facilidad para encontrar tareas.
- **Consistencia:** implica que todo elemento (imagen, ícono, tipo de letra, sonido, etc.) siempre represente lo mismo. Además debe existir coherencia del significado de los objetos, formas y métodos dentro de la GUI, con el mundo real. Esto quiere decir que las imágenes se deben asociar a palabras claves del proceso, los mensajes deben aparecer siempre en la misma posición y con el mismo formato (sean de error, alerta, éxito, etc.), entre otras tantas posibilidades. Los atributos que se utilizarán para medir esta característica son los siguientes:
- Cada elemento siempre representa lo mismo.
 - Cada elemento es representativo del dominio que representa.
 - Cada imagen se asocia a palabras claves del proceso.
 - Los mensajes siempre se muestran iguales.
- **Flexibilidad:** una interfaz es flexible si permite al usuario personalizar sus componentes. Los atributos que se utilizarán para medir esta característica son:
- Elección de idioma con cual trabajar.
 - Configuración personalizada de componentes.
- **Portabilidad (Gráfica):** la interfaz siempre debe verse igual en cualquier Sistema Operativo. Esta característica se medirá con el siguiente atributo:
- Igual apariencia en los distintos Sistemas Operativos.
- **Estética:** se refiere a que la GUI posea un diseño visualmente agradable. Debe ser sencilla, es decir, facilitar el trabajo sin producir distracciones. En los siguientes ítems se exponen las sub-características correspondientes:
- * **Fuente:** el tipo y tamaño de letra es de gran importancia. Es decir, la forma y el tamaño de la letra debe ser legible. Nombre de los atributos con que se medirá:
 - Legibilidad en tipo y tamaño de letras utilizadas en la interfaz.
 - * **Colores:** se refiere a resaltar lo importante sin sobrecargar la visión. Se aconseja usar entre 4 y 7 colores para evitar la fatiga visual [2,10], en especial tonos claros reservando los fuertes para información importante. La combinación de colores debe ser distinguible si se desea separar información. Los atributos utilizados para medir esta característica son:
 - Colores utilizados.
 - Combinación armoniosa.
 - Predominio de tonos claros.

* **Otros elementos:** hace referencia a ciertos tipos de elementos extras, presentes en la GUI como imágenes, sonidos, recuadro, tablas, etc. En caso en que la GUI cuente con una gran variedad de estos elementos, sería conveniente una interacción armoniosa entre los mismos, sin producir sobrecargas ni distracciones. A continuación se mencionan los atributos que describen esta característica:

- Información sobrecargada.
- Uso de recuadros, tablas, paneles para mostrar información.

➤ **Organización:** una interfaz debe estar bien organizada, las opciones del menú principal deben tener un orden y las opciones dentro de ellas deben estar relacionadas (Ejemplo: Archivo: nuevo, abrir, guardar, guardar como; Edición: copiar, cortar, pegar, etc).

Los atributos utilizados para medir esta característica son:

- Cantidad de opciones del menú principal.
- Promedio de cantidad de opciones por cada opción del menú principal.

➤ **Vocabulario (claridad):** se refiere a que el usuario pueda comprender los términos que se emplean en la GUI. Los atributos que se utilizarán para medir esta característica son los siguientes:

- Términos técnicos.
- Términos vulgares.
- Términos sofisticados.
- Términos desconocidos.
- Existencia de jergas.

La Figura 4 muestra el Árbol de Criterios obtenido a partir de las características y sub-características analizadas.

Definición de Funciones de Criterios Elementales para GUI

En esta sección se muestran algunos ejemplos de definición de Funciones de Criterios Elementales correspondientes a algunos atributos del Árbol de Criterios definido en la sección anterior (la

explicación de cada una de las funciones de Criterio Elemental excede la longitud de este artículo, el lector interesado en conocer más de esta temática puede leer [11]).

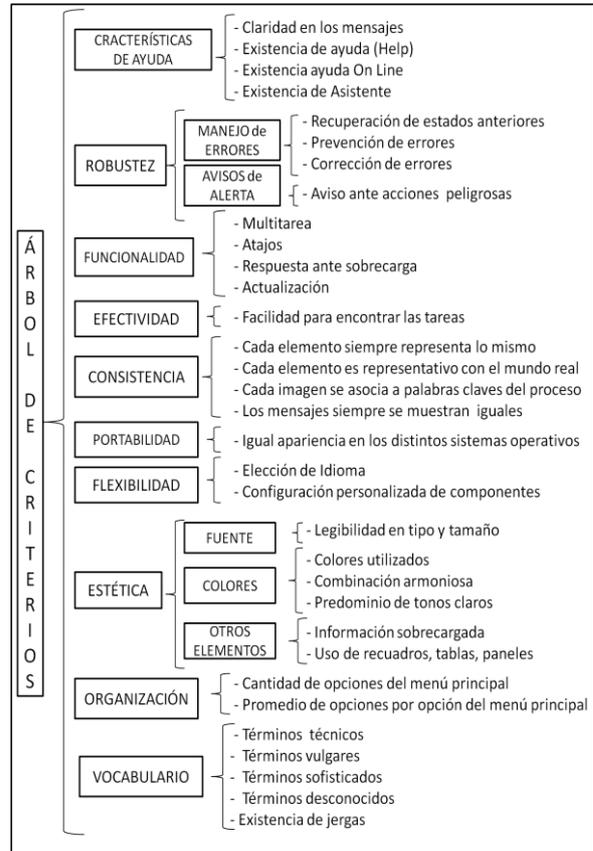


Figura 4: Árbol de Criterios

Por cada criterio se detallará: *Nombre del criterio*, *Tipo de Criterio Elemental*, *Interpretación* (Descripción del criterio), *Descripción de la Escala de Preferencia*, con su explicación correspondiente en caso de ser necesario (por ejemplo, si se trata de una fórmula, se explica las partes de la misma):

➤ **Elección de Idioma (Id):**

* *Tipo de Criterio Elemental:* Criterio de Variable Continua, Multivariable.

* *Interpretación:* Se deben tener en cuenta los diferentes idiomas que el usuario puede elegir para trabajar.

* *Descripción de la escala de Preferencia:* Se define la siguiente fórmula para medir este criterio:

$$Id = TOP3 * (50) + TOP10 * (30) + TOP20 * (20)$$

Donde *Id* es la Variable asociada a este atributo. Las variables se miden teniendo

en cuenta la información provista por EpistemoWikia¹. Este sitio brinda información estadística con respecto a la “popularidad” de los idiomas más hablados en el mundo. La misma representa las siguientes características:

~TOP3: Indica si al menos dos idiomas se pueden elegir de entre los tres idiomas mas hablados en el mundo. Puntaje: 50.

~ TOP10: Indica si al menos cinco idiomas se pueden elegir dentro de los diez primeros puestos del ranking. Puntaje: 30.

~ TOP20: Indica si al menos diez idiomas se pueden elegir dentro de los veinte primeros puestos del ranking. Puntaje: 20

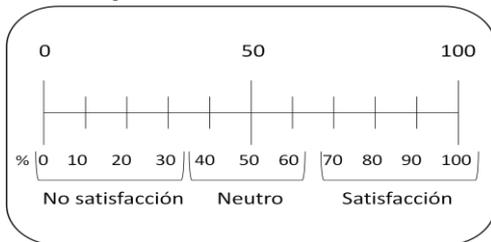


Figura 5: Escala de Preferencia Multivariable

* *Escala de Preferencia:*

La Figura 5 muestra la escala de preferencia asociada a este atributo. Los valores entre 0 y 30 indican la no satisfacción del atributo. Los valores entre 40 y 60 indican un puntaje neutro. Los valores entre 70 y 100 indican la satisfacción del atributo.

➤ Facilidad para encontrar tareas:

* *Tipo de Criterio Elemental:* Criterio de Variable Continua, Directo.

* *Interpretación:* Evalúa si se pueden encontrar fácilmente las distintas tareas.

* *Breve descripción de la escala de Preferencia:* El evaluador debe ingresar una preferencia de manera directa. Es decir, un valor entre [0,100] que indique

la satisfacción de este requerimiento por parte de la GUI evaluada.

* *Escala de preferencia:* Igual que la escala de la Figura 5.

➤ Alertas:

* *Tipo de Criterio Elemental:* Criterio de Variable Discreta, Multinivel.

* *Interpretación:* Esta característica es muy importante debido a que la GUI debe avisar/alertar al usuario si está seguro que quiere ‘eliminar’ y además esperar la confirmación del usuario para proceder.

* *Descripción de la escala de Preferencia:* Se define la siguiente escala: **Alertas = {(No,0);(Si,100)}** (Los valores mostrados significan No=Insatisfacción Si=Satisfacción)

* *Escala de preferencia:* ver Figura 6.



Figura 6: escala de Preferencia Multinivel

➤ Claridad en los mensajes:

* *Tipo de Criterio Elemental:* Criterio de Variable Discreta, Multinivel.

* *Interpretación:* Esta característica es muy importante debido a que la GUI debe ser clara, precisa en los mensajes de ayuda al usuario.

* *Descripción de la escala de Preferencia:* Se define el siguiente conjunto de valores de medición:

Claridad = {(Pobre,0); (Regular,30); (Media,50); (Buena,80); (Muy Buena,100)}

* *Escala de preferencia:* ver Figura 7.

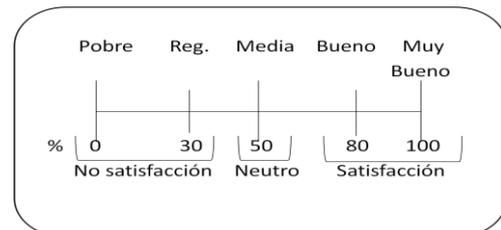


Figura 7: Escala de Preferencia Discreta

NESSy: New Evaluation Software System

En ésta sección se presenta NESSy, un sistema que implementa el método LSP. NESSy permite someter a evaluación un conjunto de elementos (en este caso GUIs.), con el fin de obtener un ranking que permita seleccionar la mejor opción entre los mismos [11,12]. Es importante destacar que dicha herramienta ha sido desarrollada en el marco de esta investigación.

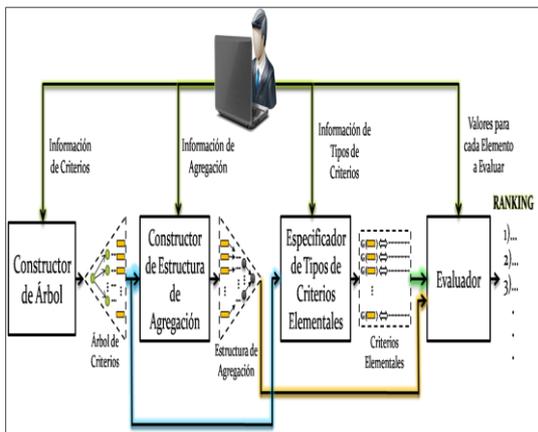


Figura 8: Arquitectura de NESSy

La Figura 8 muestra la arquitectura de NESSy. Como es posible observar, la herramienta está compuesta de cuatro componentes: *Constructor de Árbol o Constructor de Árbol de Criterios (CAC)*, *Constructor de Estructura de Agregación (CEA)*, *Especificador de Tipos de Criterios Elementales (ETCE)* y el *Evaluador (E)*.

El CAC recibe como entrada la Información de Criterios (IC) por parte del experto y produce como salida el Árbol de Criterios (AC).

CAC permite definir criterios para la caracterización de los elementos a ser evaluados, en este caso se lo utilizará para plasmar las características de las GUIs. Es evidente que esta componente tiene funcionalidades como: *Añadir Criterio*, *Borrar criterio*, *Modificar Criterio*, etc. En este contexto IC representa el conocimiento del experto. Es fundamental tener presente que este paso debe ser llevado a cabo de la mejor manera posible

porque el proceso de evaluación depende del AC. Si el AC no está bien construido los resultados obtenidos no serán correctos. La estructura del AC refleja la descomposición sucesiva de las características en sub-características y así sucesivamente hasta la obtención de los atributos.

CEA añade la lógica necesaria para llevar a cabo el proceso de evaluación. Es importante mencionar que la Estructura de Agregación se construye de manera *bottom-up*, desde las hojas (atributos) hasta que se obtiene el último operador, que computa la Preferencia Global del elemento evaluado. Este componente tiene funcionalidades como: *Agregar Operador Lógico*, *Eliminar Operador Lógico*, *Añadir Peso*, entre otras operaciones.

El ETCE recibe como su entrada el AC. De la misma forma que CEA, este componente tiene en cuenta las hojas del AC, es decir, los atributos. Para cada uno de ellos, este componente selecciona su tipo y, de acuerdo con el tipo, se definirá su función de evaluación. Los tipos de Criterios Elementales (CE) se explicaron en la sección donde se presenta el método LSP.

Por último, el *Evaluador* recibe como entrada la Estructura de Agregación y los Criterios Elementales. Una vez disponible esa información, esta componente recorre la estructura y, a partir de la información proporcionada por el experto, produce una clasificación de los elementos en evaluación.

En la siguiente sección se presenta un caso de estudio en donde se lleva a cabo una evaluación de dos GUIs utilizando NESSy. El lector interesado en conocer más detalles de las funcionalidades proporcionadas por NESSy, puede consultar los siguientes trabajos [11,12]. Muchas de estas funcionalidades serán mostradas en el caso de estudio.

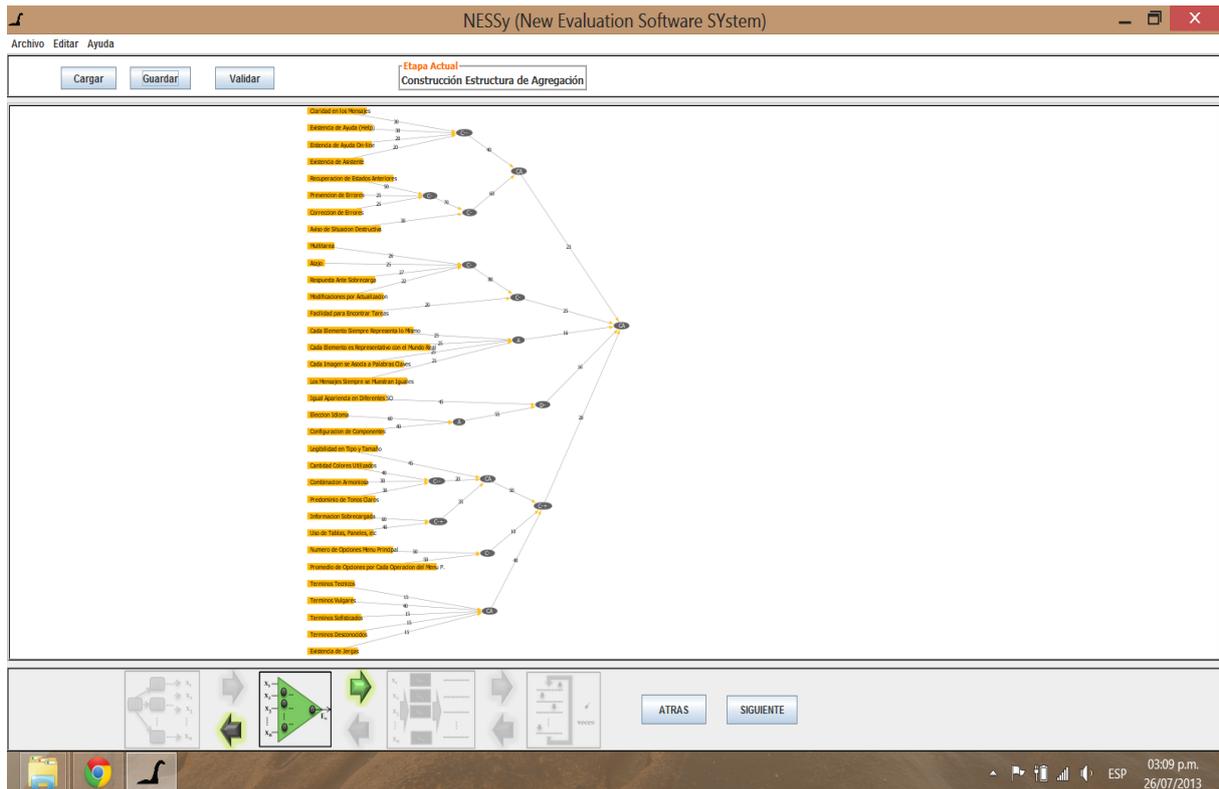


Figura 9: Estructura de Agregación construida por NESSy

Caso de Estudio: Evaluación de las GUIs de Eclipse y NetBeans

Con el objetivo de mostrar la aplicabilidad de lo propuesto en los apartados precedentes, en esta sección se presenta la evaluación de las GUI de dos IDEs muy utilizadas en el entorno de desarrollo de software: Eclipse 4.3 [13] y NetBeans 7.3.1 [14]. Como se mencionó en el apartado precedente, la evaluación fue llevada a cabo con la herramienta NESSy.

De la misma manera que LSP, NESSy facilita la evaluación a través de la definición de las siguientes componentes:

➤ **Arbol de Criterios (AC):** en esta etapa, la herramienta permite construir un árbol compuesto de características, a través de las cuales se puede evaluar las GUIs. Dicho árbol fue descrito en la tercera sección.

➤ **Estructura de Agregación (EA):** NESSy permite la construcción de una EA para evaluar GUIs, como se puede observar en la Figura 9. La misma toma como base los atributos definidos en el AC construido en la etapa anterior.

Por motivos de extensión, sólo se explicará la construcción de una rama de la EA (ver Figura 10). Más precisamente, aquella que agrega los atributos de la característica de alto nivel *Estética*. Como se mencionó en secciones precedentes, LSP utiliza operadores lógicos que, dado un determinado grado de conjunción-disyunción, permiten computar las preferencias de entrada a los mismos.

En el caso de la EA parcial para *Estética*, inicialmente se debe analizar el primer nivel de operadores correspondientes a los atributos de las sub-características *Fuente*, *Colores*, *Otros Elementos*. Como se puede observar, *Fuente* solo posee un atributo denominado *Legibilidad en Tipo y Tamaño*, por lo tanto, en esta instancia de la EA, la Preferencia correspondiente a dicho atributo se agrega en el siguiente nivel de operadores. La sub-característica *Colores* posee tres atributos: *Colores Utilizados*, *Combinación Armoniosa* y *Predominio de Tonos Claros*. En este caso, la distribución de los pesos es equitativa, salvo por el

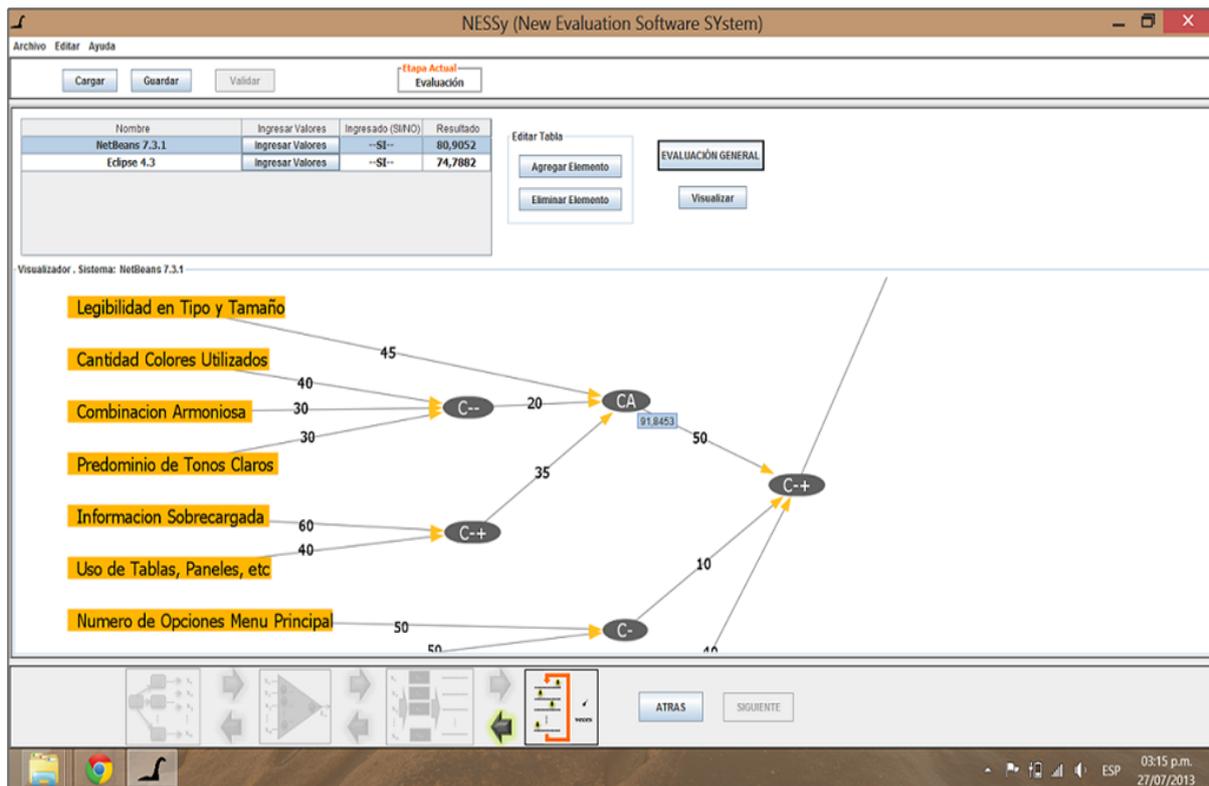


Figura 10: Etapa de Evaluación con NESSy. El resultado final se puede observar en la tabla. En el panel “Visualizador” se puede consultar los valores calculados por cada nodo operador de la Estructura de Agregación para alguna de las GUI evaluadas. De esta manera se puede conocer el puntaje obtenido por la GUI en cada característica o sub-característica.

primer atributo que tiene un peso de 40 (los otros dos 30). Esto se debe a que se considera un poco más importante la variedad en el uso de los colores, en contraste con un empleo monocromático de los mismos.

Para agregar estos atributos se ha utilizado un operador de conjunción débil (C--). Es decir, la operación de los atributos se acerca a la media pesada (A), pero se le ha establecido un cierto grado de “penalización”, en caso en que alguno, o varios, de los atributos obtengan preferencias bajas. De manera similar, los dos atributos correspondientes a la sub-característica *Otros Elementos* se han agregado mediante el operador de conjunción media C-+. En este caso, a diferencia de C--, C-+ posee un grado mayor de conjunción, por lo tanto el “castigo” a los puntajes bajos es mayor. Además se le ha asignado más importancia al atributo *Información Sobrecargada* (peso 60) por sobre *Uso de Tablas, Paneles, etc.* (peso 40). Por último, las Preferencias

correspondientes a las sub-características *Fuente*, *Colores* y *Otros Elemento*, se computan por medio del operador CA. Se utiliza un operador de conjunción media debido a que en esta instancia, no puede ser aceptable que cierta GUI obtenga puntajes muy bajos en alguna de las sub-características previamente mencionadas. En cuanto a los pesos, se puede notar que *Fuente* (45) es considerada la sub-característica más importante del conjunto, seguida de *Otros Elementos* (35) y *Colores* (20).

En el contexto de toda EA, es importante destacar una regla que se cumple en la mayoría de los casos: a medida que se van agregando las preferencias y que el proceso de agregación se va acercando al nodo final, los operadores tienden a ser más conjuntivos. Esto es así debido a que a medida que se construye la EA desde los atributos hasta el nodo final, cada vez son menos tolerables los valores bajos.

➤ **Criterios Elementales (CEs):** En esta etapa, NESSy permite la especificación de los CEs para cada uno de los atributos del AC. En el apartado anterior se exhiben ejemplos de CEs, definidos para los atributos: *Elección de Idioma, Facilidad para encontrar las tareas, Aviso ante acciones peligrosas, Claridad en los mensajes* del AC.

➤ **Evaluación:** En la última etapa del proceso de evaluación, NESSy permite ingresar las herramientas que serán sometidas a evaluación. Además, por cada una de las herramientas se debe ingresar toda la información requerida por los Criterios Elementales.

Este proceso utiliza la EA que combina tanto los criterios como los operadores de LSP. La EA es recorrida y se calcula el valor para cada criterio, siguiendo la semántica de LSP. El proceso descrito anteriormente se aplica a cada una de las GUIs sometidas a evaluación (para esta evaluación en particular, para la GUI de NetBeans y de Eclipse). Finalmente, el ranking se establece teniendo en cuenta la Preferencia Global obtenida por cada GUI.

Características de Alto Nivel	NetBeans	Eclipse
Características de Ayuda	92,1584	83,3718
Robustez	74,7828	82,4476
Funcionalidad	86,9895	71,8587
Efectividad	75	50
Consistencia	75	75
Portabilidad	100	100
Flexibilidad	88	88
Estética	91,8453	83,1933
Organización	44,4273	43,6676
Vocabulario	63,5128	54,6337

Tabla 2: tabla de preferencias obtenidas por cada GUI en cada característica de más alto nivel.

El resultado de la evaluación que muestra la herramienta es: 1° NetBeans (con un puntaje de 80,9052) y 2° Eclipse (con un puntaje de 74,7882). Es importante destacar que ambas IDEs poseen GUIs muy

similares, es por este motivo que sus puntajes son próximos (ver Tabla 2). Sin embargo existe una diferencia favorable obtenida por NetBeans en las siguientes características de alto nivel: *Características de Ayuda* (92,1584 sobre 83,3718), *Funcionalidad* (86,9895 sobre 71,8587) y *Estética* (91, 8453 sobre 83,1933). Es importante notar que además de la diferencia de puntajes, dichas características han sido ponderadas con importancia en la Estructura de Agregación, ya sea a través de los pesos relativos o mediante el grado de conjunción de los operadores. Es por este motivo que la GUI de NetBeans ha obtenido un puntaje levemente superior que la de Eclipse.

Conclusiones y Trabajos Futuros

Tal como se explicó en el inicio, el objetivo del presente trabajo es la evaluación de GUIs. Para alcanzar este objetivo, se seleccionó como método de evaluación LSP (Logic Scoring Preference). Luego de explicar cada uno de los pasos del mismo, se presentó una herramienta desarrollada en el marco de esta investigación, que implementa dicho método. Posteriormente se hace efectiva la evaluación de las GUIs. En primera instancia se define el Árbol de Criterios, es decir se establecen los criterios y atributos a evaluar de las GUI. Luego se especifican las Funciones de Criterios Elementales, es decir, cómo se van a medir los criterios antes definidos. A continuación se construye la Estructura de Agregación, para computar las Preferencias Elementales resultantes de aplicar las funciones de Criterios Elementales a los valores de cada atributo. Esta Estructura de Agregación permite computar todas las Preferencias a fin de obtener la Preferencia Global. Por último se realiza el Proceso de Evaluación.

El caso de estudio presentado es la evaluación de dos IDEs (Integrated Development Environment) muy utilizadas en el ambiente de desarrollo a gran escala: NetBeans 7.3.1 y Eclipse 4.3. La

evaluación se realizó utilizando la herramienta NESSy.

En el caso de estudio, la herramienta arrojó el siguiente resultado: 1° NetBeans (80,9052) y 2° Eclipse (74,7882). La GUI de NetBeans obtuvo el mejor puntaje debido a la diferencia favorable obtenida en las siguientes características de alto nivel: *Características de Ayuda* (92,1584 sobre 83,3718), *Funcionalidad* (86,9895 sobre 71,8587) y *Estética* (91,8453 sobre 83,1933).

La tarea de evaluación es un proceso difícil, ya sea empleando NESSy o adoptando el enfoque manual tradicional. Sin embargo, la ventaja de utilizar dicha herramienta es que una cantidad considerable de trabajo (Árbol de Criterios, Estructura de Agregación y Funciones de Criterios Elementales) queda disponible para futuras evaluaciones.

La evaluación de las GUIs no solo ayuda a seleccionar la mejor opción, o la más adecuada en cierta circunstancia cuando se debe utilizar una herramienta; sino también a mejorar el diseño al momento de desarrollar un sistema. Muchas veces se cree que la tarea del desarrollador sólo se reduce a cumplir con los requerimientos del usuario, sin considerar una gran variedad de atributos y características que debe poseer una GUI.

Por otro lado, la tarea de evaluación utilizando este método, requiere de un constante proceso de reestructuración y análisis de las componentes utilizadas en dicho proceso. Por este motivo, como trabajo futuro se pretende extender y perfeccionar el Árbol de Criterios, con el objetivo de destacar ciertos aspectos no tenidos en cuenta hasta el momento. También se pretende llevar dicho conjunto de criterios a un estándar internacional. Independientemente a la tarea anterior, se cree necesario someter la Estructura de Agregación y los Criterios Elementales a un proceso de mejora continua, donde se deben realizar distintas evaluaciones, teniendo en cuenta otros tipos de GUIs, para lograr un mejor perfeccionamiento de las mismas.

Referencias

- [1] <http://www.alegsa.com.ar/Dic/gui.php>
- [2] Galitz, Wilbert O. *The essential guide to user interface design: an introduction to GUI design principles and techniques*. Wiley.com, 2007.
- [3] Stone, Debbie, et al. *User interface design and evaluation*. Morgan Kaufmann, 2005.
- [4] J.J. Dujmovic. *A Method for Evaluation and Selection of Complex Hardware and Software Systems*. The 22nd Int'l Conference for the Resource Management and Performance Evaluation of Enterprise CS. CMG 96 Proceedings, 1:368–378, 1996.
- [5] Dasso, Aristides, et al. *Una herramienta para la evaluación de sistemas*. III Workshop de Investigadores en Ciencias de la Computación. 2001.
- [6] Funes, Ana, et al. *Creación y evaluación de modelos LSP en un contexto MDA*. XIV Workshop de Investigadores en Ciencias de la Computación. 2012.
- [7] Dasso, Aristides, and Ana Funes. *Desarrollo de modelos de evaluación usando operadores de una lógica continua*. XV Workshop de Investigadores en Ciencias de la Computación. 2013.
- [8] Dujmovic, Jozo J. "Continuous preference logic for system evaluation." *Fuzzy Systems, IEEE Transactions on* 15, no. 6 (2007): 1082-1099.
- [9] Sommerville, Ian. *Ingeniería del software 7/e*. Pearson Educación, 2005.
- [10] Nielsen, Jakob, and Hoa Loranger. *Usabilidad, prioridad en el diseño web*, Madrid, Ediciones Anaya Multimedia. Una sociedad de la información en igualdad de condiciones. Evaluación al grado de inclusión social-digital que ofrecen las TIC 267 (2006).
- [11] Miranda, Enrique. *Evaluación de funcionalidades de visualización de software provistas por librerías gráficas*. Tesis de Licenciatura. Universidad Nacional de San Luis. 2013.
- [12] Miranda, Enrique, et al. *NESSy: a New Evaluator for Software Development Tools*. In *Proceedings SLATE* (2013).
- [13] IDE Eclipse <http://www.eclipse.org/>
- [14] IDE NetBeans <https://netbeans.org/>
- [15] Dujmović, Jozo J., and Hajime Nagashima. "LSP method and its use for evaluation of Java IDEs". *International journal of approximate reasoning* 41.1 (2006): 3-22.