

# Uso de nuevas métricas orientadas a las competencias en la gestión curricular

Zanga, Mabel Dejean, Gustavo Aubin, Verónica Blautzik, Leonardo  
*Universidad Nacional de la Matanza*

## Abstract

*El trabajo presenta nuevas pautas para la estandarización de la Gestión curricular y medición de la calidad en los procesos de enseñanza-aprendizaje en carreras de ingeniería y licenciatura en informática. En particular, profundiza la posibilidad de usar métricas orientadas a las competencias que mejoren sustancialmente el estado actual y aporten múltiples ventajas adicionales. Se muestra que las métricas utilizadas en la Ingeniería de Software son apropiadas para medir el resultado final del proceso de enseñanza-aprendizaje, fáciles de medir y útiles para la gestión curricular. Se obtienen las equivalencias entre estas nuevas métricas y las tradicionales horas-cátedras o el sistema de créditos europeo (créditos ECTS; European Credit Transfer System). Se destaca la superioridad y ventajas de usar las nuevas métricas versus las métricas tradicionales.*

## Palabras Clave

créditos, currículo, gestión curricular, movilidad académica, competencias, métricas.

## Introducción

Actualmente, en la República Argentina, los contenidos mínimos de un plan de carrera universitaria, deben cumplir con las normas enunciadas en el boletín oficial del Ministerio de Educación Nro. 31.667 del año 2009 [1]. Las mismas están basadas en las recomendaciones de la RedUNCI [2] (Red de Universidades Nacionales con Carreras en Informática) formuladas en su Documento de Acreditación de las Carreras de Informática - RedUNCI Diciembre 2005, y en su Propuesta Curricular [3].

Por ejemplo, la recomendación respecto al número mínimo de horas por Núcleo Curricular Básico, para la carrera de Licenciado En Ciencias de la Computación se muestra en la tabla 1.

Núcleo Curricular	Carga horaria
Ciencias Básicas	400 hs.
Teoría de la Computación	500 hs.
Algoritmos y Lenguajes	550 hs.
Arquitectura, Sistemas Operativos y Redes	250 hs.
Ingeniería de Software, Bases de Datos y Sistemas de Información	450 hs.
Aspectos Profesionales y Sociales	50 hs.
<b>Total</b>	<b>2.200 hs.</b>

**Tabla 1 Núcleos curriculares para las Licenciaturas**

A estas 2200 hs, se le adicionan 1000 hs. mínimas para llegar a las 3.200 hs. y permitir que cada Institución Académica establezca las orientaciones y contenidos específicos que considere más adecuados. Las carreras de Licenciado En Sistemas / Sistemas de Información y Análisis de Sistemas se las agrupa y se les da una distribución horaria ligeramente distinta. Análogamente, a la Carrera de Licenciatura en Informática se le da otra distribución también con ligeras modificaciones. Por último, a las carreras de Ingeniería en Computación e Ingeniería en Sistemas de Información / Informática se les estableció los núcleos curriculares y distribución de carga horaria mostrados en la tabla 2.

Núcleo Curricular	Carga horaria
Ciencias básicas	750hs
Tecnologías Básicas	575hs
Tecnologías aplicadas	575hs
Complementarias	175hs
<b>Total</b>	<b>2.075 hs.</b>

**Tabla 2. Núcleos curriculares para las carreras de Ingeniería**

A estas 2075 hs, se le adicionan 1675 hs. mínimas para llegar a las 3.750 hs. Estas recomendaciones, en algunos casos, son el fruto de la experiencia empírica de cada uno de sus participantes como así también por una razonable asociación con las prácticas usadas en carreras de otras áreas de la ingeniería. No deja de ser una estandarización de facto según lo que se venía haciendo históricamente en las principales universidades del país.

A nivel internacional, no se difiere mucho de esta propuesta, el proyecto Tuning Europeo plantea un conjunto de competencias genéricas y un sistema de créditos análogo (créditos ECTS; European Credit Transfer System) [4], la diferencia principal es que se computan todos los tiempos del estudiante y no solo las horas de clase, por lo que podría decirse que queda igual al multiplicarlo por una constante. El proyecto Tuning Latinoamericano [5] no se expidió en las carreras del área de informática, sin embargo, comprendió 8 áreas del conocimiento; las competencias genéricas y el sistema de créditos son análogos al europeo.

La característica principal de este enfoque, es el uso del tiempo como principal métrica física. El problema, es la independencia existente entre dicha métrica y los contenidos que se puedan impartir, la calidad del conocimiento adquirido por los alumnos y su aptitud práctica para resolver problemas. La hora cátedra es una métrica no significativa o es mal usada y con significados difusos que dependen de factores psicológicos: ¿qué significa una hora cátedra en una clase de programación? o ¿qué significa 600 hs en un Núcleo

Curricular que contiene 20 ítems? difícilmente sean 30 hs por ítem, algunos temas requerirán más tiempo que otros dependiendo no solo del tema sino del docente que lo interprete. Dicha métrica, posiblemente y en este contexto, sea útil a fines de contaduría para obtener costos de carreras, pero nunca se llegará a la calidad por esa vía o medición de conocimientos adquiridos. Dicha métrica no ayuda a la Gestión de la Calidad en el proceso educativo. En la práctica, se delega únicamente en el docente a cargo de la cátedra, la responsabilidad de definir la “profundidad” de cada ítem y la administración de los créditos, siendo esto, una desnaturalización de los currículos. Otro problema adicional se da en los planes de carreras a distancia, dónde tomar las horas-cátedra como métrica, carece de sentido.

En [6] se propone usar a las líneas de Código, conocidas por su siglas en inglés: LOC, como nuevas métricas junto a las LOC/hs como métrica personal de software, ambas aplicadas al Núcleo curricular: Algoritmos y Lenguajes de las Licenciaturas o a su núcleo análogo Tecnologías Básicas de las carreras de las Ingenierías. También se propone usar una segmentación de tres niveles para el universo de problemas, cada nivel habilita para determinadas competencias. En el presente trabajo, se le agrega un cuarto nivel para captar una nueva competencia: la programación Cliente-Servidor. Se aplican las propuestas hechas en [6] al Núcleo Curricular de Tecnologías Básicas.

En [7] y [8], se destaca la formación de las carreras de Ingeniería por Competencias, en [8] se afirma que: “En la actualidad es una tendencia internacional en el diseño de los planes de estudio de ingeniería el uso de las competencias como horizonte formativo.” También se destaca: “el diseño por competencias o su integración en el Plan de Estudios ayudaría a vigorizar el saber hacer requerido a los ingenieros recién recibidos.”

Esta integración entre competencias y el Plan de estudio es lo que mostraremos a continuación.

### Elementos del Trabajo y metodología

Los datos usados para el presente análisis, se obtuvieron de las mediciones y registros que se vienen realizando desde el año 2009 al presente por el equipo docente de la cátedra de programación avanzada correspondiente al tercer año de la carrera de Ingeniería Informática de la Universidad Nacional de La Matanza. Los datos son obtenidos por sucesivos, refinados y constantes análisis de las guías de trabajos prácticos, de los trabajos prácticos especiales (entregables), del trabajo del Taller, encuestas tomadas a los alumnos, el registro de las planillas simplificadas según el *Proceso Personal Software* PSP [9] obligatorias para los alumnos, registro de evaluaciones, o evidencias obtenidas a partir de: evaluaciones parciales, finales, proyectos grupales de aula y por el conocimiento experto que tienen sus integrantes sobre el universo de problemas ofrecido a sus estudiantes. Así mismo, han sido considerados los datos relevados en la materia “Programación”, correspondiente al segundo año de la misma carrera y Universidad.

El procedimiento para el análisis de los datos obtenidos se centrara en mediciones objetivas aplicando las Líneas de Código y líneas de código /hora, ambas segmentadas en competencias específicas como principal métricas de software propuesta en [6] para la gestión de curricular.

La técnica PSP simplificada para su uso en el aula, ayuda en mucho para formar en los alumnos y docentes la cultura de “medir todo lo que se hizo” y “estimar y planificar todo lo que se piense hacer”. En [10] se detalla el uso de PSP en el Aula.

La metodología XP [11], adaptada a la cátedra, ayuda para mejorar la calidad del trabajo del alumno, aumenta su desempeño y resulta motivadora para el alumnado.

### Resultados

Se estudió parcialmente el núcleo curricular de Tecnologías Básicas, y dentro de él, a las cátedras de Programación y Programación Avanzada. Del análisis detallado, se obtuvo las LOC discriminadas por cada objeto utilizado en el aprendizaje. En la tabla 3, a modo de ejemplo, se muestra el resultado para la cátedra de Programación avanzada.

	Cantidad	Líneas de código
Guías	9	2470
Evaluaciones	2	160
TP del Taller	1	1400
por clase del taller	16	1440
TTPP especiales	4	900
<b>Total</b>		<b>6370</b>

Tabla 3 Resumen de líneas de código

### Comparación de los resultados obtenidos versus horas cátedra y ECTS

Comparando las métricas usadas y las mediciones realizadas versus la métrica horas cátedras, usada según el boletín oficial del Ministerio de Educación Nro. 31.667 del año 2009 y los ECTS, se construye la tabla 4 comparativa.

cátedra	LOC totales	horas cátedra	ECTS
Programación	7855	160	15
Programación Avanzada	6370	160	15

Tabla 4 Resumen de las distintas métricas

La diferencia aparente de LOC entre ambas cátedras (aproximadamente un 20 %) se explica fundamentalmente, que en una de ellas solo se alcanzan problemas de Nivel 1 y en la otra se ven los cuatro niveles, también se observa que al no seguir las mismas metodologías que en programación avanzada, los alumnos no emplean tiempo en preparar la prueba del software ni en una documentación mínima, por lo cual dedican más tiempo a la codificación en desmedro de las otras actividades igualmente importantes del desarrollo del software.

Si discernimos las LOC por nivel de dificultad obtenemos los resultados mostrados en la tabla 5.

Métricas	Cátedras	
	Programación	Programación Avanzada
LOC nivel 1	7855	1900
LOC nivel 2		1800
LOC nivel 3		1000
LOC nivel 4		1670
LOC totales	7855	6100
horas de clase	160	160
ECTS	15	15

Tabla 5 resumen de métricas discriminado

Al discernir las LOC por Nivel de dificultad se observa que la métrica utilizada es mas descriptiva, tanto del trabajo del alumno aula adentro como fuera del horario de clase, y por sobre todas las cosas, da una medida acorde a las Competencias alcanzadas por el alumnado. Tan descriptiva resulta esta tabla, que incluso hace irrelevante el nombre de la cátedra y se podría simplemente expresado por las competencias logradas por el alumno. Este enfoque, netamente orientado a las competencias, disminuye varios riesgos numerados por Rue en [12]; algunos de ellos son: *“la creencia de que toda competencia enunciada – como todo contenido de programa - será lograda por los estudiantes sólo por haber sido prevista. Finalmente, es común no considerar el grado de profundidad de este logro al que se aspira para dichos estudiantes, en cada materia y o curso académico. El “olvido” de su forma de desarrollo vuelve irrelevante cualquier formulación que se haga de las mismas.”*

Los créditos ECTS; European Credit Transfer System [4], son análogos a las horas cátedras, la diferencia principal, es que computan todos los tiempos del estudiante, como ser: resolución de ejercicios, trabajos prácticos especiales, preparación de evaluaciones, etc. y no solo las horas de asistencia a clase, por lo tanto, podría decirse que los ECTS queda igual a las horas cátedras si se lo multiplica por una constante. Se define que una carrera de grado de ingeniería de cinco años de

duración, debe tener 300 ECTS, por lo cual, el alumno debe hacer 60 ECTS por año (en el supuesto caso que termine su estudio en cinco años). Aproximadamente 1 ECTS corresponde a unas 25 o 30 hs invertidas por el alumno en el aprendizaje y de éstas solo una fracción serán de asistencia a clase. Establecido esto, el alumno deberá invertir aproximadamente 1650hs al año en tareas para el aprendizaje (ver fórmula 1).

$$1650hs = 60ECTS * 27.5 \quad (1)$$

Se toma 27.5 como un promedio entre 25 y 30 hs.

La equivalencia de 160 horas cátedras a 15 ECTS se obtiene, de manera levemente conservadora, de la siguiente manera: suponiendo que por cada hora de asistencia a clase, el alumno debe invertir un factor de 1.5 horas adicionales, resulta que 160 hs cátedra son equivalentes a 400 hs dedicadas

$$400hsDedicadas \cong 160hC + 160hC * 1.5 \quad (2)$$

Luego, dividiendo este resultado por 27,5 horas para obtener los ECTS queda:

$$14.55ECTS = \frac{160hC+160hC*1.5}{27.5} \quad (3)$$

Redondeando: 15 ECTS

En general tendremos:

$$X hCátedra = \left(X * \frac{2.5}{k}\right) ECTS \quad (4)$$

Dónde  $25 \leq k \leq 30$ ;

En nuestro caso, las 160 horas de cátedra, da que son equivalentes a entre 13.3 y 16 ECTS.

Como era de esperar, los valores de LOC expresados en la tabla 5, son equivalentes y esto se demuestra de la siguiente manera: tomado como ejemplo a programación avanzada que totalizan 6370 LOC, observando nuestros propios registros, en promedio, los alumnos codifican a una velocidad media de 60 LOC por hora,

obtenemos aproximadamente 100 horas de codificación. Por otro lado, sabiendo que la tarea de codificación lleva aproximadamente un 25 % del total (tiempo de análisis, planificación, diseño, preparación de la prueba, la prueba, documentación mínima y corrección de errores cubren el 75 % restante) tenemos que el alumno ocupa 400 horas en el trabajo total y esto último es equivalente a los 15 créditos ECTS ( $400 / 27$ ). Todo esto aproximadamente. Con esto, se demuestra que la métrica usada en estos casos, no solo son equivalentes a las hs de clase y ECTS, sino que llevan más información ligada a las competencias logradas por el alumnado.

### Discusión

Es claro que la métrica *Líneas de código* solo puede utilizarse en las cátedras de programación y parcialmente en otras como Base de Datos. Este conjunto de materias puede representar generalmente el 25 % del total de las carreras de Informática (tanto de Ingeniería como las Licenciaturas). Esta utilidad parcial sigue siendo una mejora, más si se transita el camino de direccionar los contenidos curriculares actuales hacia las Competencias. Por otra parte, las mismas ideas pueden extenderse fácilmente a otros Núcleos curriculares, en efecto, por ejemplo, en las Cátedras de Matemáticas se puede cuantificar el esfuerzo de los alumnos en ejercicios resueltos y a esos dividirlos igualmente como hicimos con los problemas de programación en varios niveles. En otros Núcleos curriculares podrán usarse otras métricas análogas a las LOC como ser: modelos, diseños, ejercicios, casos, papers, capítulos de libro. Siempre, en lo posible, segmentados en no más de 3 o 4 niveles.

Es claro su aplicación en los planes de carreras con la modalidad no presencial.

La aplicación de estas nuevas métricas están más ligadas a las metodologías activas de enseñanza-aprendizaje y no tanto a las metodologías más tradicionales donde el profesor imparte su clase y los alumnos resuelven los problemas independiente-

mente. En [6] se aconseja, de usar estas métricas, incluir específicamente las técnicas PSP y XP dentro del currículo, por ser muy útiles para alcanzar los objetivos. El presente trabajo no distingue a las LOC reutilizadas, este un tema pendiente para futuros refinamientos. En [6] se discute la posibilidad de usar LOC/hs distintas en cada uno de los Niveles, en el presente trabajo, se pudo utilizar el promedio porque se resuelven problemas de todos los Niveles en cantidades semejantes y es parte de lo registrado en clase.

### Conclusión

Pudimos medir el esfuerzo de los alumnos a partir de métricas nuevas en este campo, pero que son ampliamente usadas en la Ingeniería de software. Esto se hizo solamente para un Núcleo curricular, pero es posible extenderlo a todos los demás Núcleos curriculares.

La utilización de las líneas de código como nueva métrica para medir tanto el esfuerzo del alumno es superior en varios aspectos a las métricas tradicionalmente usadas (horas cátedra y ECTS).

Las ventajas principales son:

- a) pueden ser usadas en carreras con modalidad semi-presencial o no presencial.
- b) indican claramente las competencias alcanzadas por el alumno. En cambio, las métricas tradicionales no indican nada al respecto.
- c) indican claramente cuál es la profundidad de los temas alcanzados, las tradicionales no aclaran nada al respecto.
- d) indican el “qué sabe hacer” y no solo lo que debería saber. En este sentido diferenciamos bien entre “saber” y “saber hacer”.
- e) Amplían claramente el significado del plan de contenidos mínimos o más precisamente, podía reconstruirse dicho plan a partir de estas métricas y no viceversa.
- f) ídem a e) pero con respecto al plan de contenidos detallado.

g) facilita la movilidad estudiantil al poder comparar por competencias y no por simple numeración de contenidos.

h) Las métricas propuestas no excluye el uso de las tradicionales, se las puede usar como un complemento.

### **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 Lic. Zulema Nisi y Julio Crispino por sus meticulosas tomas de métricas.

### **Referencias**

[1] (2009). Boletín oficial número 31.667 del Ministerio de Educación; Resolución 786/2009

[2] RedUNCI. (2005) diciembre. Documento de acreditación de las Carreras de informática

[3] RedUNCI. (2006). Carreras de Grado en Informática, Propuesta de Currícula RedUNCI – Mar del Plata, 6 de junio.

[4] CONFEDI. (2007). (Consejo Federal de Decanos de Ingeniería). XXXVII Reunión Plenaria de Santa Fe “Proyecto Estratégico de Reforma Curricular de las Ingenierías 2005 – 2007”

[5] BENEITONE, P. OTROS (2007). Reflexiones y perspectivas de la Educación Superior en América Latina. Informe Final del Proyecto Tuning América Latina 2004-2007. Bilbao: Publicaciones de la Universidad de Deusto. Publicaciones de la Universidad de Deusto.

[6] Dejean G. (2010). Uso de nuevas métricas en la Gestión de carreras de ingeniería y licenciaturas en Informática; Congreso Mundial INGENIERIA 2010 - Capítulo 5.1.1;

[7] Villa, A., Poblete, M. (2007). Aprendizaje basado en Competencias. Una propuesta para la evaluación de las Competencias Genéricas. Bilbao

[8] La Formación del Ingeniero para el Desarrollo Sostenible, Aportes del CONFEDI. Congreso Mundial Ingeniería 2010 – Buenos Aires – Octubre 2010

[9] Humphrey Watts S. (2001). Introducción al Proceso Software Personal. Addison-Wesley Iberoamericana

[10] Dejean G., Aubin V., Blautzik L. (2011) “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).

[11] Beck. K. (2000). Extreme Programming Explained. Embrace Change. Reading, Massachusetts: Addison-Wesley. 2000.

[12] Rué, J. (2008, Abril). Formar en competencias en la universidad: entre la relevancia y la banalidad. Red U. Revista de Docencia Universitaria, numero monográfico 1 “Formación centrada en competencias”.

### **Datos de Contacto:**

*Universidad Nacional de La Matanza. Florencio Varela 1903 (B1754JEC) -- San Justo, Buenos Aires, Argentina .  
dejean2010@gmail.com;  
leoblau@gmail.com; vaubin@unlam.edu.ar*

