

LOS 330 CRÉDITOS Y LA INGENIERÍA DEL SOFTWARE

José Javier Dolado¹

¹*Facultad de Informática de San Sebastián (UPV-EHU)*
e-mail: dolado@computer.org

RESUMEN: La ingeniería del software está recibiendo una atención especial dentro de los grupos relacionados con la enseñanza y los diseños de curricula informáticos. En este trabajo vamos a exponer algunas ideas relacionadas con el diseño general de un curriculum orientado hacia la ingeniería del software. Las cuestiones se pueden abordar desde el punto de vista particular de la educación en nuestro país y, también, desde el punto de vista más general de la enseñanza de la informática a nivel mundial.

1. LA EDUCACIÓN EN INGENIERÍA DEL SOFTWARE

En estos momentos, la educación superior informática está influenciada principalmente por los contenidos que se imparten bajo el epígrafe "ciencias de la computación". Pero en los últimos años ha aparecido una tendencia que trata de dotar del carácter de profesión y de entidad propia a la "ingeniería del software". Aunque el contexto educativo es el de EEUU, bastantes de las cuestiones que se abordan son aplicables a la enseñanza en nuestro país. Considerar la "ingeniería del software como una profesión" supone tener una nueva perspectiva sobre qué constituye la esencia de la actividad profesional informática. Una manera de ilustrar la diferencia entre algunas disciplinas informáticas se muestra en la figura 1, propuesta por R.L. Glass, que se ha utilizado posteriormente en las propuestas SWEBOK y SE-BOK (que se describen más adelante).

Este artículo fue premonitorio de la polémica que estamos viviendo hoy en día alrededor de la disciplina de la ingeniería del software [Glass, 1992]. En él, este conocido autor ya apuntó las posibles relaciones entre las disciplinas de "Computer Science", "Software Engineering", "Information Systems" y los dominios de aplicación.

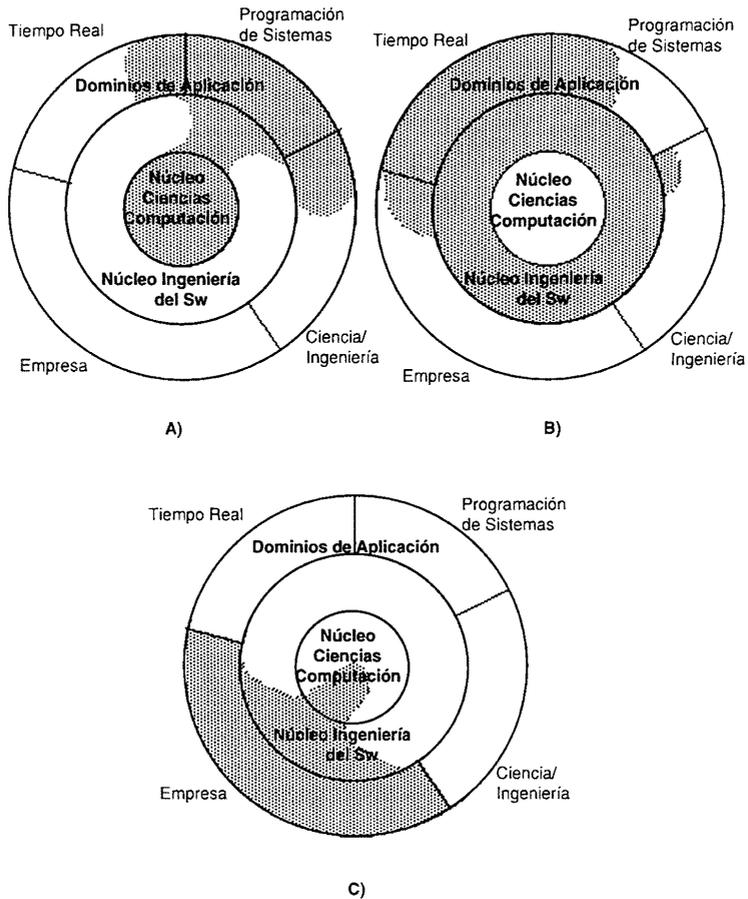


Figura 1. Núcleos y aplicaciones de las disciplinas: a) Ciencias de la Computación; b) Ingeniería del Software; c) Sistemas de Información

En la figura 1 se muestran los núcleos de la ingeniería del software, de las ciencias de la computación y los dominios de aplicación. Una clasificación general de los éstos últimos es: a) programación de sistemas; b) aplicaciones de empresa; c) aplicaciones científicas y de ingeniería y d) tiempo real. Las tres figuras nos describen qué es lo que abarcan cada una de las disciplinas. Es curioso observar cómo Glass separó los núcleos de ingeniería del software de los de las ciencias de la computación, afirmando que la primera posee su propio núcleo. Hecho éste que las publicaciones actuales están refrendando.

2. FUENTES CURRICULARES

Existen varias fuentes en las que nos podemos basar para definir la estructura de un currículum, que a continuación vamos a enunciar. De la lectura de estos textos podemos concluir que la propuesta más ambiciosa, referida a la ingeniería del software, es la del SWEBOK, aunque no sea una propuesta curricular.

a) Propuesta SWEBOK

El *SoftWare Engineering Body of Knowledge* (ver <http://www.swebok.org>, SWEBOK) es un subproducto de un comité conjunto (SWECC) de la ACM y del IEEE, cuyo objetivo es "establecer los conjuntos de criterios apropiados y las normas para la práctica profesional de la ingeniería del software, sobre las que se puedan basar las decisiones industriales, la certificación profesional y los currícula".

La principal actividad de este grupo de trabajo es la de definir el núcleo del cuerpo de conocimiento de la ingeniería del software, puesto que a partir de ahí se puede definir la profesión y los diferentes currícula universitarios. Este cuerpo de conocimiento se construye, primero, a partir del "conocimiento generalmente aceptado" (CGA). Este CGA es un subconjunto de lo que sería el cuerpo total de conocimiento de la disciplina de la ingeniería del software. El objetivo del SWECC es crear una guía que permita identificar y describir el CGA. También se indica que los ingenieros del software no sólo deben conocer el CGA, sino que también deberán tener nociones de otras disciplinas. Por otra parte, identificar el CGA no equivale a establecer los currícula para la ingeniería del software. El SWECC lo deja bien claro: "Software engineering body of knowledge and curriculum are not the same".

En estos momentos se han identificado diez áreas de conocimiento de ingeniería del software, que se están detallando, de tal manera que para finales de 2001 se disponga de la descripción del "núcleo de conocimiento de la ingeniería del software:

1. Requisitos del Software
2. Diseño del Software
3. Construcción del Software
4. Pruebas del Software
5. Mantenimiento del Software
6. Gestión de Configuración del Software
7. Calidad del Software
8. Gestión de la Ingeniería del Software
9. Infraestructura de Ingeniería del Software
10. Proceso de Ingeniería del Software

Las "disciplinas relacionadas" con la ingeniería del software que propone el SWEBOK son:

- o Ciencias de la Computación (“Computer Science”)
- o Matemáticas (“Mathematics”)
- o Gestión de Proyectos (“Project Management”)
- o Ingeniería de Computadores (“Computer Engineering”)
- o Ingeniería de Sistemas (“Systems Engineering”)
- o Gestión y Ciencias Empresariales (“Management and Management Science”)
- o Ciencias Cognitivas y Factores Humanos (“Cognitive Sciences and Human Factors”)

b) Propuesta del Software Engineering Institute SE-BOK

El proyecto SE-BOK y las guías del Software Engineering Institute de Carnegie Mellon (<http://www.sei.cmu.edu>), al contrario que el SWEBOK, sí que pretenden desarrollar currícula basados en el cuerpo de conocimiento que ellos definen. Esta propuesta se ha realizado casi simultáneamente al trabajo del SWEBOK, [Hilburn, 1999] para el cuerpo de conocimiento de la ingeniería del software. De la lectura de tal informe observamos que bastantes áreas de conocimiento SE-BOK se solapan con las del SWEBOK. Básicamente SE-BOK divide los contenidos en las categorías de Ingeniería del Producto e Ingeniería del Proceso. Además añade una categoría de Dominios del Software y otra de Fundamentos de la Computación. Cada una de ellas las divide a su vez en Áreas de Conocimiento y éstas a su vez en Unidades de Conocimiento

c) Computing Curricula 2001

El CC-2001 es la otra iniciativa conjunta ACM/IEEE “Year 2001 Model Curricula for Computing: CC-2001”, que pretende definir los contenidos del área *Computer Science*.

d) Otras referencias

Complementariamente se pueden utilizar las referencias de [Parnas, 1998] [Greening, 2000], aunque este último compendio de trabajos es excesivamente general. Han aparecido números especiales dedicados a la profesión y educación en la ingeniería del software en: IEEE Software, Nov/Dec, 1999; Information and Software Technology, vol. 40, n4, 1998; Journal of Systems and Software, vol. 49, n2, 1999 e IEEE Computer, Mayo, 2000, además de [ASE, 1998].

4.- ESTRUCTURA CURRICULAR GENERAL

Puesto que nuestro problema es de índole práctica, nos preguntamos ¿Cuántos créditos se pueden impartir teniendo en cuenta principalmente la propuesta del SWEBOK? La respuesta no

es fácil, puesto que por encima de conocimiento fundamental se sitúa el avanzado y el especializado, y esto permite tener currícula con una extensión flexible. Vamos a describir la estructura general que puede proporcionar, al menos, el núcleo del cuerpo de conocimiento del SWEBOK, incluyendo también algunas recomendaciones del SE-BOK. Es difícil proponer un plan de estudios simplemente mimetizando las guías del SE-BOK, puesto que la formación universitaria es diferente en cada país.

Lo que sí podemos hacer es proponer una distribución general de áreas de conocimiento, en créditos y porcentajes, que incluyan las ideas del SWEBOK, SE-BOK y CC-2001. La figura 2 representa (aproximadamente) los 330 créditos (también aproximadamente) de una ingeniería en informática orientada hacia la ingeniería del software, repartidos de la siguiente manera:

- A. Cuerpo de conocimiento de la ingeniería del software, incluyendo el núcleo del cuerpo de conocimiento, conocimiento avanzado y determinados aspectos de conocimiento especializado: 25 % de los créditos o unos 80 créditos.
- B. Ciencias de la computación, con los contenidos que se definen en el CC-2001: 36% de créditos (120).
- C. Ingeniería de computadores: 30 créditos o un 9% aproximadamente
- D. Ciencias matemáticas y físicas: 18% o unos 60 créditos
- E. Ingeniería de sistemas: un 2,5% (8 créditos)
- F. Gestión de proyectos: según se define por el Project Management Institute (un 5%, 16 créditos). Se debe tener en cuenta que parte de estos contenidos ya se incluyen en el área de "Gestión de la ingeniería del software" del SWEBOK.
- G. Ingeniería de calidad: que va englobada dentro de la parte de calidad del SWEBOK
- H. Gestión y ciencias empresariales: 2%, aproximadamente.
- I. Otros: un 2,5 % aproximadamente, para cursar cualquier otro contenido.

4. CONCLUSIÓN

Ya hemos indicado que la gran cuestión reside en si se define o no (o cómo se define) la disciplina de "ingeniería del software". La emergencia de la ingeniería del software puede proporcionar grandes beneficios a la comunidad de profesionales informáticos. Y la orientación de la educación hacia la profesionalización es otro hecho que va a marcar las pautas educativas de los próximos 20 años. No obstante, existen algunas posturas divergentes como la que mantiene P.J. Denning, que cree que existe "the Profession of Computing", en la que se englobarían las diferentes especialidades informáticas. Sin embargo, no es ésta la tendencia general. Actuar de esta última manera obligaría a un cambio radical en las actuaciones que siguen tanto el SE-BOK, como el SWEBOK, que son las que van a marcar las pautas en los próximos años.

Agradecimientos

Recibido apoyo de UPV-FHU 141.226 EA083-98 y CICYT TIC99-0351.

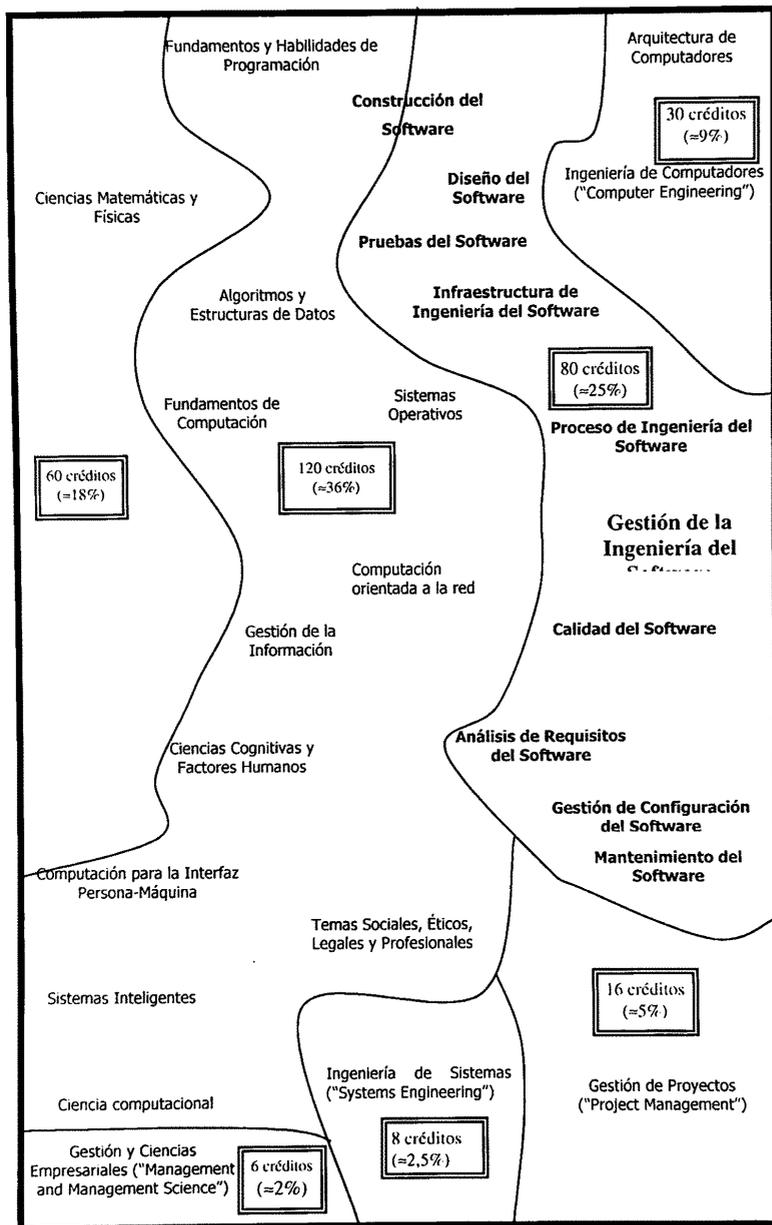


Figura 2. Posible estructura curricular orientada a la ingeniería del software.

5.- REFERENCIAS

- [ACM, 1999] ACM, ACM Panel on Professional Licensing in Software Engineering Report to Council (8 de julio,1999), <http://www.acm.org>
- [Andrews, 1999] D. Andrews, Software engineering education in the 21st century, Information and Software Technology 41 (1999) pp. 933-936
- [ASE, 1998] Annals of Software Engineering, Special issue on software engineering education, vol. 6, 1998.
- [Bagert, 1999] D.J. Bagert, T.B. Hilburn, G Hislop, M. Lutz, M. McCracken y S. Mengel.
- Guidelines for Software Engineering Education, Version 1.0. Technical Report, CMU/SEI-99-TR032, Carnegie Mellon University.
- [Bourque, 1999] Pierre Bourque, Robert Dupuis, Alain Abran, James W. Moore, and Leonard Tripp. The Guide To The Software Engineering Body Of Knowledge, IEEE Software, November/December 1999
- [Glass, 1992] R.L. Glass, "A Comparative Analysis of the Topic Areas of Computer Science, Software Engineering, and Information Systems", J. Systems Software, 1992, 19:277-289
- [Greening, 2000] T. Greening (ed.) Computer Science Education in the 21st Century, Springer-Verlag, 2000
- [Hilburn, 1999] Thomas B. Hilburn, Iraj Hirmanpour, Soheil Khajenoori, Richard Turner and Abir Qasem, A Software Engineering Body of Knowledge Version 1.0, April 1999, Technical Report CMU/SEI-99-TR-004 ESC-TR-99-004.
- [Hilburn, 1999] T. Hilburn et al., Guidance for the development of software engineering education programs, Journal of Systems and Software, vol. 49 (1999) pp. 163-169
- [Parnas, 1998] D.L. Parnas, "Software Engineering Programmes are not Computer Science Programmes", IEEE Software, November/December, 1999.