La Enseñanza de la Ingeniería del Software en el marco del Espacio Europeo de Educación Superior

Francisco Ruiz
Dep. de Tecnologías y Sistemas de Información
Escuela Superior de Informática
13071 Ciudad Real
francisco.ruizg@uclm.es

Resumen

En esta ponencia se presentan algunas reflexiones personales sobre el papel de la Ingeniería del Software (IS) en la Informática y las consecuencias que ello debería tener, a juicio del autor, en la reforma de los contenidos y de la manera de enseñar y aprender, de cara a la adaptación al Espacio Europeo de Educación Superior (EEES). Se argumenta que es necesario un cierto cambio de punto de vista apostando más por incorporar de forma transversal en toda la carrera de Informática una perspectiva de ingeniería. También se ofrecen datos que indican que IS debería tener un papel mas importante en los estudios de Informática. Por último se sugieren ideas sobre cómo puede reformarse la materia de IS y otras afines o de similares características.

1. Introducción

Cuando se habla de IS nos estamos refiriendo a un tipo especial de ingeniería, que se encuadra tradicionalmente dentro del campo de conocimientos llamado Informática, y que IEEE ha definido como "la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de los principios y hábitos de la ingeniería al software" [1].

En esta ponencia se presentan una serie de informaciones que pretenden provocar la reflexión en el profesor de IS. La ponencia está elaborada desde la experiencia personal de un profesor con 20 años de historia académica, que en sus orígenes tuvo una formación en ciencia pura, y que también ha conocido el mundo real de la empresa, antes y durante su vida de profesor. El auditorio al que va destinada es, fundamentalmente, el colectivo de profesores de IS y materias fines. Por ello, de manera transversal a lo largo de todos los apartados, se intentan sacar conclusiones que puedan ser útiles para los profesores interesados en esta materia.

En primer lugar, se presenta el contexto de la materia. En este apartado se analiza la IS tanto desde un punto de vista histórico como desde una perspectiva ingenieril; se da una justificación de lo que hace que sea una ingeniería distinta de cualquier otra; y se presenta el cuerpo de conocimientos que le es propio.

El siguiente apartado se dedica a analizar el papel que la IS tiene en la Informática, en sus dimensiones de como profesión y de carrera universitaria. Para ello, se presenta la opinión de los principales currículos internacionales y del sector industrial; se describe la influencia que los distintos enfoques de la profesión informática tiene en la IS; se muestran los datos del mercado de trabajo; y se presentan las causas ocultas (fuertemente relacionadas con el papel de la IS) que han conducido a una gran caída en las cifras de nuevos alumnos.

El apartado cuarto se dedica a presentar informaciones y experiencias útiles de cara a la reforma de la enseñanza de IS para su adecuación al EEES. Para ello, se presentan los aspectos del libro blanco y de la propuesta de directrices que más relación tienen con la IS; se resumen propuestas metodológicas para la reforma de un plan de estudios completo o de una asignatura suelta o en grupo; y se presentan algunas lecciones aprendidas o consejos extraidas de la experiencia personal del autor.

El documento finaliza con unas conclusiones generales.

Ponencia invitada en el II Simposio Nacional de Docencia en la Informática (SINDI'2007); integrado en el II Congreso Español de Informática (CEDI). 11 al 14 de septiembre de 2007, Zaragoza.

2. Contexto

2.1. Perspectiva Histórica

Desde los años 70 se habla de "crisis del software" con una cierta periodicidad. Parece como si, justo los que nos dedicamos a esto fuésemos los peores profesionales, los más "chapuceros", prácticamente en todos los países y en todas las empresas. En mi opinión lo que ocurre es que nos enfrentamos a un problema difícil, especial y distinto al que se enfrentaron antes otras ingenierías.

Existe una diferencia clave entre la IS y cualquier otra ingeniería actual: en IS nos dedicamos a sistemas discretos e inmateriales mientras que los demás ingenieros trabajan con sistemas continuos (casi siempre) y de naturaleza material. Nosotros no tenemos, para lo bueno y para lo malo, unas leyes de la naturaleza en las que basarnos, manifestadas en forma de conocimiento científico (física, química, etc.).

Por otro lado, existe un dicho popular sobre "morir de éxito"; y es que, a veces, los éxitos se confunden con los fracasos. Si somos malos haciendo software, ¿por qué el software es cada vez más frecuente en la vida de cualquier persona y más importante para cualquier organización?. Y lo peor es que ese "pesimismo histórico" de la IS lo transmitimos casi sin darnos cuenta a nuestros alumnos, es decir, a nuestros futuros profesionales.

En realidad, en la IS hemos seguido un proceso histórico muy interesante que nos ha permitido avanzar y superar múltiples retos y dificultades. Además, para entender donde estamos y hacia donde vamos debemos comprender de donde venimos. A lo largo del tiempo hemos sido capaces de resolver una gran cantidad de dificultades, en un camino que siempre se ha caracterizado por aprovechar el aumento de potencia y capacidad del hardware para "hacer software más cerca de las personas y más lejos de las máquinas" (ver figura 1). Así, conforme se iban superando unos retos y el software se hacía más complejo, tuvimos que enfrentarnos a otros retos nuevos. Algunos hitos que resumen esta evolución positiva del problema de hacer software, son:

- Fuimos capaces de trabajar de manera lógica y no física: Los enchufes en clavijas pasaron a ser 0's y 1's.
- Inventamos lenguajes y traductores para poder "representar" mejor los algoritmos como ideas: El código máquina dejó de usarse para programar.
- Ideamos lenguajes "cercanos" al idioma natural o a los idiomas de las ciencias (matemáticas) para mejorar nuestra capacidad de expresar: COBOL se pareció al inglés lo máximo posible; PROLOG se basaba en la lógica matemática.
- Descubrimos que teníamos que organizar bien el flujo de ejecución del código: Programación estructurada (PASCAL).
- Si tenemos mucho código mejor separarlo en varias partes: Programación modular (MODULA 3).
- Necesitábamos poder manejar informaciones complejas de distinta naturaleza: Tipos abstractos de datos; Sistemas de bases de datos.
- En un software grande es un lío "organizar" las piezas de código. Necesitamos un criterio para decidir qué piezas tener y qué datos y código poner en cada una: Orientación a objetos.

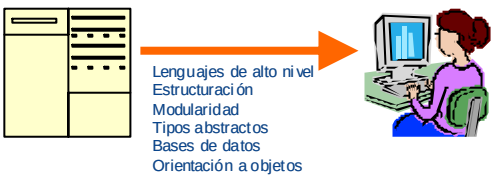

Figura 1. Evolución histórica de la creación del software.

Pero seguimos teniendo otros retos pendientes:

- Si hemos ido subiendo de nivel de abstracción en los lenguajes de programación, ¿nos permite la tecnología actual dar otro salto más?; ¿Por qué Java es código fuente, y UML no?; ¿Existe alguna manera de construir software más rápida y con menos errores?.
- La integración sigue siendo un problema difícil., tanto en cuanto a integrar sistemas como a integrar tecnologías.

- Seguimos teniendo dificultades para entender bien a los clientes/usuarios. Muchos proyectos técnicamente correctos fracasan en el sentido de que el software no sirve a los supuestos destinatarios o no lo usan.
- El software está en la red, de forma que el concepto clásico cerrado de "aplicación" software está perdiendo vigencia.

Para enfrentar estos retos surgen algunos nuevos paradigmas y tecnologías asociadas, que no son alternativos a los anteriores, sino complementarios (ver tabla 1).

Objetivo	**Paradigma**	**Tecnología**
Aumentar productividad	Desarrollo Dirigido por Modelos (DDM)	MDA – Model-driven Architecture …
Mejorar integración y tiempo de respuesta	Orientación a Servicios (SOC)	SOA – Service-oriented Architecture …
Mejor adaptación a las necesidades y cambios de las empresas	Orientación a los Procesos de Negocio (BPM)	BPMS – Business Processes Management Systems

Tabla 1. Nuevos paradigmas y tecnologías.

En la actualidad los problemas y sistemas a enfrentar en la IS pueden llegar a ser muy complejos. Por ello, un profesional puede llegar a tener que utilizar de forma integrada varios paradigmas, antiguos o novedosos. Por ejemplo, en la integración de BPM+SOC+DDM a la hora de automatizar una actividad de un proceso de negocio tenemos varias opciones: i) invocar alguna funcionalidad provista por un sistema legado existente de forma directa; ii) ídem pero siendo necesario añadir una capa externa al sistema legado para la interacción (normalmente con tecnología de SW-XML); iii) invocar algún servicio externo (servicio web) que provea dicha funcionalidad; y iv) crear un nuevo sistema que provea de dicha funcionalidad.

Una buena lectura para reflexionar sobre todo esto se encuentra en [2], donde el reconocido informático Booch expone sus reflexiones sobre el problema de hacer software a un auditorio de profesores de informática. Destacamos dos de sus opiniones, que comparte plenamente el autor de esta ponencia:

- "*Software development has been, is, and will remain fundamentally hard*".
- "*It is a tremendous privilege to be a software professional. It is also a tremendous responsibility*".

2.2. Perspectiva de Ingeniería

El DRAE [1] define Ingeniería como el "estudio y aplicación, por especialistas, de las diversas ramas de la tecnología", e Ingeniero/a como "la persona que aplica los conocimientos de una o varias ramas de la ciencia para resolver cierto tipo de necesidad de la gente, mediante el diseño, construcción u operación de algún tipo de artefacto". Ambas palabras tienen su origen en algunas acepciones del término ingenio: "industria, maña y artificio de alguien para conseguir lo que desea. Máquina o artificio mecánico (ingenio de azúcar)".

Por tanto, aplicar a la enseñanza de la Informática y de la IS una perspectiva de ingeniería significa, principalmente, inculcar en los estudiantes y futuros profesionales, que el objetivo de su profesión es diseñar, construir y hacer que funcionen cierto tipo de sistemas que sirven para resolver problemas de la gente. El foco de atención debería estar en esto último, que es la razón de ser de cualquier ingeniería: resolver problemas de la gente.

Todas las ingenierías, incluida la IS, se caracterizan porque:

- Se necesitan conocimientos avanzados para diseñar y construir el tipo de sistemas que la caracteriza.
- Existen dos "momentos": primero, conocer el problema, y sólo después, podemos diseñar y construir la solución.
- Para conseguir buenos resultados (en calidad, tiempo y costes) es necesario trabajar de forma organizada y sistemática.
- La creatividad es necesaria (diseño), pero no es suficiente. Existe una clara diferencia entre ser artista y ser ingeniero.

[1] Diccionario de la Real Academia Española, disponible en http://buscon.rae.es/draeI/.

En mi opinión, otra característica de los buenos ingenieros, sean de la rama que sea, es que saben utilizar el sentido común en su trabajo. Algunas maneras o prácticas de hacerlo, que podemos enseñar a nuestros alumnos para su futuro trabajo de ingenieros software, son:

- *Ley del Mínimo Esfuerzo*. Entre las opciones correctas elegir la más sencilla. Lo bueno si fácil dos veces bueno. Esta práctica se debe emplear no sólo para aprobar asignaturas, sino también para ser más productivos en general.
- *Reutilización*. No sólo del código, sino también del resto de artefactos software y, muy importante, del conocimiento personal y organizacional.
- *No inventar la rueda*. Emplear estándares internacionales o nacionales, y normas de la organización.
- *Zapatero a tus zapatos*. Cuidado con enfrentar un problema para el que claramente no se está capacitado. En IS, al igual que en otras ingenierías, no existe linealidad entre la escala o tamaño del sistema y la complejidad para desarrollarlo. Debemos hacer un esfuerzo para que los estudiantes entiendan esta situación. Podemos, por ejemplo, usar una metáfora sobre edificios. Para construir un muro de ladrillos de 1m de alto me basto yo solo; para construir un muro similar pero de 5 metros de alto tengo que tener cuidado porque desvíos que antes no eran significativos, ahora son claves para que el muro quede torcido y se caiga. Para construir una caseta en el campo necesito unos conocimientos que no tengo y mejor llamar a un albañil. El sabe que tiene que haber puertas, ventanas, vigas, etc. Si se trata de un edificio de varia plantas son necesarias las habilidades de un arquitecto. Incluso, si se trata de un rascacielos de 300 metros de alto, un arquitecto inexperto puede construir un rascacielos que, sorpresa, se caiga por no saber que ahora es fundamental tener en cuenta la resistencia a los vientos. En software ocurre lo mismo. Precisamente por esto es por lo que, cada vez más, para ejercer de auténtico ingeniero de software es necesaria una alta formación que solo puede conseguirse con una carrera universitaria adecuada. El resto de personas podrán trabajar en el sector informático, pero cada vez más con un papel de meros técnicos programando código o montando equipos.
- *Aprender de la experiencia*. Utilizar buenas prácticas y lecciones aprendidas, nuestras o de otros. Recordar el dicho de que "más sabe el diablo por viejo que por diablo" .

Quizás a estas alturas algún lector se haya convencido de la opinión del autor de que es necesario dar un toque más ingenieril a la carrera de Informática, pero piensa que eso no significa que sea necesaria la existencia de la IS como materia. De hecho, es sabido que una parte de la comunidad que englobamos bajo el nombre de Informática desprecia o casi a la IS. Para dichas personas vamos a dar una breve justificación de la necesidad de la existencia de la IS, otra vez comparando con las otras ingenierías.

Una ingeniería existe porque las personas usan artefactos/sistemas de un cierto tipo cada vez más complejos, de forma que se creó la necesidad de tener ingenieros capacitados para diseñarlos, construirlos y operarlos. Un indicador de la complejidad de un sistema es el número de variables independientes que afectan al comportamiento del sistema. En un sistema físico (automóvil) suelen ser decenas o cientos. En muchos sistemas software actuales llegan a ser miles o decenas de miles. El mayor nivel de complejidad que el ser humano ha enfrentado a lo largo de su historia se encuentra en algunos de los sistemas software actuales (Windows Vista, Linux, MS Office, ORACLE, SAS). Por tanto, si fueron necesarias las otras ingenierías, igual o más lo es la IS.

Si alguien necesita utilizar el método científico para convencerse, aquí tiene dos evidencias empíricas que le ayudaran:

1. Ir a una empresa, decir que no hace falta la IS y observar las caras que le ponen.
2. Ir a otra empresa y observar quien es el jefe de quien, si el que programa mejor Java o el que conoce mejor técnicas de análisis/diseño, planificación o gestión de proyectos.

2.3. Especificidad de la Ingeniería del Software

Al principio se argumentó que la IS se diferenciaba de las demás ingenierías en la propia

naturaleza de los sistemas que enfrenta, materiales y continuos unos, inmateriales y discretos otros. De cara a la enseñanza de la IS es bueno tener en cuenta esta cuestión, y por ello es bueno que, en algún momento, los estudiantes reflexionen sobre qué es el software y su naturaleza. Algunas preguntas para reflexionar sobre ello son las siguientes:

- ¿A qué se parece el software?
  - a un frigorífico (que se fabrica);
  - a un libro (que se idea y se escribe);
  - a una receta de cocina (que se inventa y se anota); o
  - a un servicio de un abogado en un juicio (que nos ayuda con su conocimiento especializado).
- ¿Qué diferencia existe entre producto y servicio software?.
  - ¿Tiene el software esta doble naturaleza?, o
  - ¿Se trata de maneras distintas de ganar dinero con él (modelo de negocio)?
- La gente que hace software, ¿qué clase de habilidades y capacidades debe tener?
  - Arquitecto,
  - Albañil,
  - Jardinero, o
  - Artista
- ¿Cuáles de dichas habilidades son más fáciles de globalizar a países menos desarrollados y cuales menos porque se necesita "estar en el sitio" para conocer la realidad y el contexto?.

Otra manera en que se manifiesta la especificidad de la IS respecto de las demás ingenierías tiene que ver con las materias científicas fundamentales que necesita. A la hora de establecer los currículos y planes de estudios, en la Academia debemos tener en cuenta esta diferencia. En otras ingenierías necesitan formar científicamente para manejar sistemas materiales continuos; por tanto, necesitan sobre todo las matemáticas de lo continuo (cálculo) y la ciencia de lo material (física). En cambio, en IS se necesita la matemática de lo discreto y la ciencia de lo inmaterial (ideas, personas, organizaciones). En Informática en general si es necesaria la física por que tenemos los sistemas hardware, pero todo lo demás sigue siendo válido, al menos, de forma global.

2.4. Cuerpo de Conocimientos

La naturaleza diferente de la IS queda reflejada, como no podía ser de otra manera, en los contenidos y competencias que los profesores de esta materia deben enseñar. Para ello, una fuente clave es el documento "Guide to the Software Engineering Body of Knowledge" o SWEBOK [3], cuya portada se muestra en la figura 2.

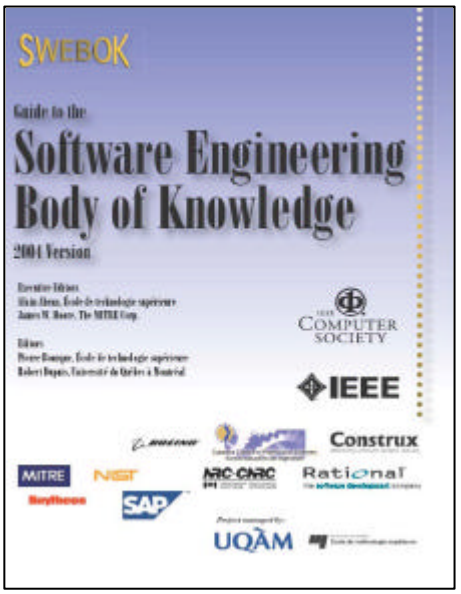

Figura 2. Portada de la última versión del SWEBOK.

SWEBOK es la principal herramienta que hoy en día tiene el autor de esta ponencia en su rol de profesor de IS. Su amplitud y estructura (ver figura 3) son muy aptas para ayudar al docente de la materia.

En SWEBOK, los conocimientos propios de la IS se clasifican en 10 áreas de conocimiento, que se muestran en la tabla 2. Se identifican las 5 áreas tradicionales típicas del desarrollo y mantenimiento del software, pero también otras 5 áreas que se dedican a lo que anteriormente hemos denominado como perspectiva de ingeniería. A título de curiosidad, en la tabla 2 se han incluido las páginas dedicadas a cada parte.

Ciclo de Vida del Software (69 pg)	Perspectiva de Ingeniería (64 pg)
Requisitos (17 pg)	Gestión de la Configuración (gestión de productos) (14 pg)
Diseño (12 pg)	Gestión de la Ingeniería (gestión de proyectos) (13 pg)
Construcción (9 pg)	Proceso de Ingeniería (orientación a procesos) (14 pg)
Pruebas (16 pg)	Herramientas y Métodos (tecnología de soporte) (9 pg)
Mantenimiento (15 pg)	Calidad (14 pg)

Tabla 2. Áreas de Conocimiento de SWEBOK.

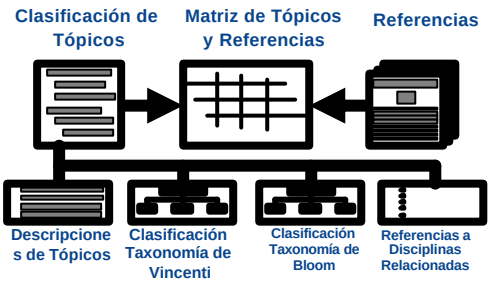

Figura 3. Estructura de la descripción de un area de conocimiento en SWEBOK.

Cada área de conocimiento se desarrolla en un capítulo de la guía, que suele continuarse con una justificación y una descripción del contexto. Adicionalmente, se realiza una detallada descripción de los temas pertenecientes a cada área de conocimiento, siendo esta parte de suma importancia para el docente en esta materia. Finalmente, en cada capítulo se incluye un amplio conjunto de referencias bibliográficas para cada área, tema y, en algunas ocasiones, epígrafe de cada tema.

Si consideramos la cifra de páginas como un indicador aproximado de la importancia que SWEBOK (es decir, la comunidad internacional de expertos) le concede a cada parte de la IS, podemos concluir que a los contenidos que aportan la perspectiva de ingeniería debería dárseles cerca del 50% de la importancia (tiempo, puntos, etc.). La experiencia de este autor es que, en prácticamente todas las universidades españolas, se está muy lejos de este casi equilibrio. De hecho, es común que la situación sea todavía más alejada; ya que dentro de las áreas del ciclo de vida del software la mayoría de la atención se presta a diseño y construcción, algo a requisitos (aunque está mejorando últimamente), y casi nada a pruebas y menos aún a mantenimiento. Es normal en las universidades españolas que más del 80% del tiempo y atención esté volcado en requisitos, diseño y construcción, que en SWEBOK suponen menos del 29% de las páginas.

De lo anterior, podemos sacar una conclusión para los profesores de IS en España: además de pedir, con razón, que la carrera conceda más importancia a la IS, internamente debe dársele más peso a las áreas de SWEBOK que aportan la perspectiva de ingeniería. En estas ocasiones algunos alegan la dificultad debida a la escasez de tiempo; pero, como casi siempre, no se trata de un problema de falta de tiempo sino de priorizar y ponderar de forma adecuada. Una manera de facilitar este cambio es la siguiente: liberar al alumno del esfuerzo/tiempo dedicado a programar/codificar dentro de las asignaturas de IS (eso es tarea de otras asignaturas) para dedicarlo a las competencias que aportan la perspectiva de ingeniería.

SWEBOK incluye también un capítulo dedicado a disciplinas relacionadas con la IS (ver tabla 3). Esto puede ser una orientación para el docente de IS a la hora de establecer correspondencias con otras asignaturas y para situar la IS dentro de un plan de estudios.

Ingeniería de Computadores
Ciencia de la Computación
Gestión
Matemáticas
Gestión de Proyectos
Gestión de la Calidad
Ergonomía del Software
Ingeniería de Sistemas (incluidos Sistemas de Información)

Tabla 3. Disciplinas relacionadas con la IS según SWEBOK.

3. Ingeniería del Software vs Informática

En este apartado se establece el papel de la IS dentro de la Informática desde diversos puntos de vista, curriculares, perfiles profesionales, demandas de las empresas, etc.

3.1. Currículos Internacionales

Existe un consenso en la comunidad académica internacional de Informática en que los currículos de ACM son la principal fuente de inspiración para la elaboración de cursos, currículos y planes de estudios completos. En este sentido, cabe destacar que en 2005 se publicó el denominado "*Computing Curricula 2005*" o CC2005 [5], que es una clara evolución del currículo de 2001, y que consiste en un informe denominado "Overview Report", que intenta resumir el contenido de los informes específicos de cada disciplina e integrarlos todos desde una perspectiva común. En este documento se intenta dar una respuesta general a la pregunta ¿qué es la Informática (Computing)? y se establece el alcance de sus diversas disciplinas. También se dan definiciones elaboradas de cada una de las disciplinas y se analizan las relaciones e interacciones entre ellas.

En el CC2005 se identifican 5 disciplinas dentro de la Informática:

- Ciencia de la Computación (*Computer Science*)
- Ingeniería de Computadores (*Computer Engineering*)
- Ingeniería del Software (*Software Engineering*)
- Sistemas de Información (*Information Systems*)
- Tecnología de la Información (*Information Technology*)

Por tanto, IS es una de las disciplinas principales de la Informática. Teniendo en cuenta que en España, al contrario que en los Estados Unidos de América, sólo se considera un único título de grado, la existencia de dicho currículo específico en IS se debe reflejar en la existencia de postgrados especializados. Además, la existencia de estos 5 currículos diferenciados indica que no esta claro que Informática sea una profesión. De hecho, algunos expertos propugnan que Informática es un sector económico en el cual se incluyen varias profesiones, siendo una de ellas la de ingeniero del software. Sobre este tema es interesante la lectura de la serie de artículos publicados por Peter Denning (expresidente de ACM) en la revista "*Communications of ACM*" bajo el título "*The profesion of IT*", en los últimos 5 años. Algunas traducciones están disponibles en castellano en la web de la revista Novática.

Para establecer la relevancia que se le ha dado a la IS en cada uno de los currículos de ACM, lo más oportuno es analizar la tabla publicada en el "Overview Report", que compara el peso relativo otorgado en cada una de las 5 disciplinas a lo que llaman "computing topics" (ver tabla 4). Se muestran rangos de valores, que varían entre 0 y 5, representando el énfasis mínimo y máximo que se le debería poner a ese tópico en el currículo de cada disciplina. Dichos valores fueron obtenidos por consenso de todas las partes que intervinieron en la elaboración del CC2005. Las disciplinas se identifican por sus siglas en inglés. La disciplina de IS (columna) se resalta en fondo amarillo y los tópicos (filas) propios de IS se resaltan en fondo verde.

Tópico	**CE**	**CS**	**IS**	**IT**	**SE**
Fundamentos de Programación	4-4	4-5	2-4	2-4	5-5
Programación Integrativa	0-2	1-3	2-4	3-5	1-3
Algoritmos y Complejidad	2-4	4-5	1-2	1-2	3-4
Arquitectura y Organización de Computadores	5-5	2-4	1-2	1-2	2-4
Principios y Diseño de Sistemas Operativos	2-4	3-5	1-1	1-2	3-4
Configuración y Uso de Sistemas Operativos	2-3	2-4	2-3	3-5	2-4
Principios y Diseño de Redes	1-3	2-4	1-3	3-4	2-4
Uso y Configuración de Redes	1-2	2-3	2-4	4-5	2-3
Plataformas Tecnológicas	0-1	0-2	1-3	2-4	0-3
Teoría de los Lenguajes de Programación	1-2	3-5	0-1	0-1	2-4
Interacción Hombre-Computador	2-5	2-4	2-5	4-5	3-5
Gráficos y Visualización	1-3	1-5	1-1	0-1	1-3
Sistemas Inteligentes	1-3	2-5	1-1	0-0	0-0
Teoría de Gestión de Información	1-3	2-5	1-3	1-1	2-5

Tópico	CE	CS	IS	IT	SE
Práctica de Gestión de la Información	1-2	1-4	4-5	3-4	1-4
Computación Científica	0-2	0-5	0-0	0-0	0-0
Aspectos Legales/Profesionales/Éticos/Sociales	2-5	2-4	2-2	2-4	2-5
Desarrollo de Sistemas de Información	0-2	0-2	5-5	1-3	2-4
Análisis de Requisitos Técnicos	2-5	2-4	2-4	3-5	3-5
Fundamentos de Ingeniería para Software	1-2	1-2	1-1	0-0	2-5
Economía de Ingeniería para Software	1-3	0-1	1-2	0-1	2-3
Modelado y Análisis de Software	1-3	2-3	3-3	1-3	4-5
Diseño de software	2-4	3-5	1-3	1-2	5-5
Verificación y Validación de Software	1-3	1-2	1-2	1-2	4-5
Mantenimiento del Software	1-3	1-1	1-2	1-2	2-4
Procesos Software	1-1	1-2	1-2	1-1	2-5
Calidad del Software	1-2	1-2	1-2	1-2	2-4
Ingeniería de Sistemas de Computador	5-5	1-2	0-0	0-0	2-3
Lógica Digital	5-5	2-3	1-1	1-1	0-3
Sistemas Distribuidos	3-5	1-3	2-4	1-3	2-4
Seguridad: Aspectos y Principios	2-3	1-4	2-3	1-3	1-3
Seguridad: Implementación y Gestión	1-2	1-3	1-3	3-5	1-3
Administración de Sistemas	1-2	1-1	1-3	3-5	1-2
Integración de Sistemas	1-4	1-2	1-4	4-5	1-4
Desarrollo Digital	0-2	0-1	1-2	3-5	0-1
Soporte Técnico	0-1	0-1	1-3	5-5	0-1

Tabla 4. Importancia de los diversos tópicos en cada disciplina.

Como se puede deducir del análisis de la tabla 4, tópicos de IS son necesarios en cada una de las 5 disciplinas. La conclusión de esto es muy importante de cara a la futura reforma de las titulaciones en España: IS debería ser materia obligatoria para cualquier estudiante de grado de España, independientemente de la orientación del currículo de cada centro hacia alguna de las 5 disciplinas de ACM. Esto significa que IS debería ser un contenido formativo común (materia troncal según la nomenclatura antigua) para todos los estudiantes del grado en Ingeniería Informática. Si un tópico tiene una valoración de 0-0 en alguna de las 5 disciplinas significa que no se considera necesario para ella. Por tanto, se deberían incluir todos los tópicos (filas) marcados en fondo verde a la hora de determinar los contenidos de la materia de IS común para todos los estudiantes de grado en Informática. La importancia final a asignar a cada tópico dependerá de la orientación hacia una o varias disciplinas que se le quiera dar al plan de estudios.

En cuanto al currículo específico de IS, "*Software Engineering 2004 – Currículo Guidelines for Undergraduate Degree Programs in Software Engineering*", conocido como SE2004 [4], su utilidad no es directa porque en España no existe ni existirá en un futuro próximo un título de grado especializado.

3.2. Opinión de la Industria

Probablemente, la opinión más significada de la industria europea sobre los currículos en Informática sea la del "*Career Space*" [2]. Este consorcio propugna que las universidades europeas sigan una estructura curricular como la mostrada en la figura 4 [6], donde la IS se encuadraría dentro de la base de aplicaciones y metodología para la solución de sistemas.

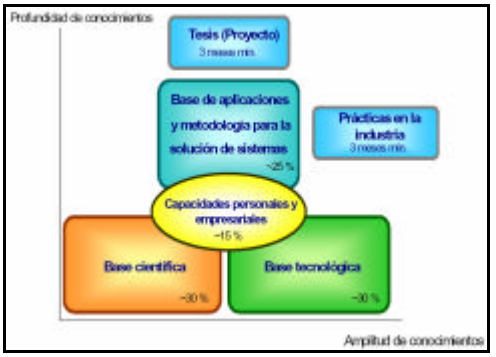

Figura 4. Ámbito de competencia del graduado TIC según *Career Space*.

[2] *Career Space* es un consorcio formado por importantes compañías del sector TIC (BT, Cisco Systems, IBM Europa, Intel, Microsoft Europa, Nokia, Nortel Networks, Philips, Siemens, Telefónica y Thales) junto con la EICTA (*European Information and Communications Technology Industry Association*) que está apoyado oficialmente por la Comisión Europea.

A nivel español también son interesantes los informes PAFET (Propuesta de Acciones para la Formación de profesionales de Electrónica, Informática y Telecomunicaciones). En el PAFET II, titulado "Alternativas y oportunidades para la implicación empresarial en las futuras estructuras curriculares relacionadas con las TIC" [8], se presenta un estudio de las necesidades de reforma de los currículos TIC y algunas propuestas al respecto. En línea con una de las propuestas del Career Space, a los conocimientos que forman parte de la cultura profesional tradicional de los ingenieros TIC, el informe PAFET propone incorporar los conocimientos relacionados con la gestión de la tecnología y de los proyectos ya que, como muestra la figura 5, "para muchos profesionales TIC actuales, la formación en gestión se ha llevado a cabo una vez culminado el proceso educativo reglado y favorecido por la promoción de estas personas en la empresa a lo largo de su actividad profesional".

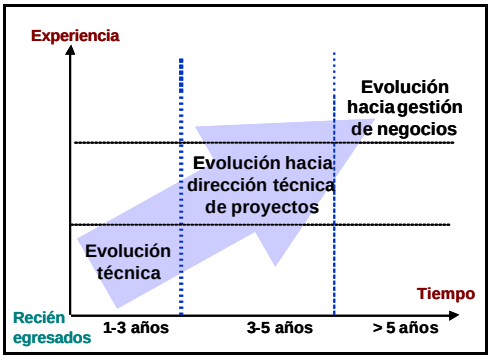

Figura 5. Evolución profesional habitual en los profesionales TIC.

Desde la perspectiva de un profesor de IS, una conclusión interesante de los informes PAFET es que debemos tener en cuenta que los estudiantes van a tener una carrera profesional a lo largo de la cual desempeñarán diversos puestos. En la actualidad, es normal estar varios años de analista-programador, para después ascender a jefe de proyecto. Después de unos 5 años más, muchos informáticos ascienden a puestos de gestión de negocio relacionados con la Informática (por ejemplo, director de sistemas de información). Una buena manera de ayudar a nuestros estudiantes es diseñar un plan de estudios en el cual las materias de IS, gestión de proyectos y gestión de negocios estén bien encajadas y relacionadas entre sí. Esto se debe concretar en cosas puntuales. Por ejemplo, que además de darles la definición de sistema de información desde un punto de vista tecnológico, también debemos dársela desde la perspectiva del negocio (¿qué papel desempeña en las empresas?; ¿a qué procesos de negocio da soporte?, etc.).

3.3. Enfoque Profesional

Otro de los aspectos que influyen a la hora de proponer un currículo para las titulaciones de Informática y, por tanto, para el planteamiento de la materia de IS, es el enfoque (científico vs. tecnológico) que se quiere dar al profesional informático. En Estados Unidos tienen currículos diferenciados para cada enfoque, pero en España no y esto origina que sea una cuestión abierta de cara a la próxima reforma de las titulaciones.

Diversos autores se han dedicado a exponer diferentes puntos de vista sobre esta dualidad ciencia vs ingeniería. Así, por ejemplo, se afirma que:

- "La Ingeniería consiste en el uso de la Ciencia y de la Tecnología para construir productos que serán utilizados por otras personas". El software es uno de esos productos.
- Los ingenieros encuentran sus problemas en la práctica. Los científicos en la literatura.
- Un 'científico' ve una máquina de estados finitos como un modelo de computación, mientras que un 'ingeniero' ve en ella una herramienta de diseño".

Respecto a la situación en España, y a pesar de la importante evolución ocurrida desde mediados de los 90, se puede afirmar que, en la mayoría de los casos, los planes de estudio de las Ingenierías Informáticas (antes Licenciaturas) se encuentran mas orientados hacia un científico (perfil de "Ciencia de la Computación" de ACM) que hacia el enfoque de ingeniería.

Por otro lado, un profesor de Informática debe ser consciente del papel social que debe jugar el ingeniero informático. Para ello, Dahlbom y Mathiassen [9] han identificado tres posturas diferentes en la orientación, actividad y papel de un informático (ver tabla 5).

	Construye cosas	Ayuda a la gente	Cambia las cosas
Orientación	Máquina Sistema	Cultura	Poder
Actividad	Construc-ción	Evolución	Interven-ción
Papel	Ingeniero	Facilitador (Técnico de soporte)	Emancipa-dor (Consultor)

Tabla 5. Enfoques de la profesión informática.

Esta visión del ingeniero informático distingue tres identidades o enfoques diferentes dentro de la informática: los ingenieros de desarrollo, que son aquellos que construyen artefactos software o hardware; el personal de soporte que maneja una forma más evolucionada de la profesión (ayudando a usar aquello que otros desarrollaron); y los consultores que centran su actividad en la generación de poder, ligado a la posibilidad de cambiar las cosas con el uso de lo que fue desarrollado antes. En general, en las Escuelas y Facultades de Informática de España se ha concedido más peso al primer rol, descuidando bastante los otros dos. En efecto, "la formación que da la universidad se orienta más al ingeniero de desarrollo (incluso algunos más críticos aluden a una orientación hacia la investigación), ni siquiera reparando en el hecho de que este mismo ingeniero deberá afrontar otras tareas menos científicas o técnicas". Esta situación se confirma al repasar los planes de estudios y observar que, en la gran mayoría, existen importantes carencias de formación en habilidades para la comunicación, trabajo en grupo, dirección de equipos o desarrollo organizacional.

Dahlbom y Mathiassen proponen incluir nuevos conceptos recurrentes tales como calidad, incertidumbre o eficiencia y efectividad, a la vez que matizan los procesos fundamentales, que ahora son teoría, diseño e interpretación. La teoría incluye en esta nueva definición conceptos de organización, de forma que permita el proceso de comprensión y evaluación de la función de los sistemas. El diseño se centra ahora en el interés por el uso de aquellos. Finalmente, el proceso de interpretación introduce las humanidades como disciplina para comprender y evaluar la problemática del uso de las máquinas y sistemas.

En esta línea, es interesante el análisis que Poore [10] hace comparando las disciplinas de ingeniería electrónica (circuitos) e ingeniería genética con la IS. Este autor eligió esas otras disciplinas porque, al igual que el software, abordan sistemas muy complejos con cantidades muy grandes de elementos diferentes. En las tres disciplinas los pasos desde la concepción hasta el producto final son similares, de forma que las especificaciones dan lugar a máquinas de estados y las tres ingenierías son actividades básicamente matemáticas. Pero Poore considera que mientras que las dos primeras han seguido un proceso responsable desde la teoría a la práctica no ha ocurrido lo mismo en el caso del software. Dice este autor que "la prueba es que todavía hoy en día, la industria del software funciona a base de 'fuerza bruta' y de cantidades masivas de recursos (dinero) antes que por medio de una ciencia e ingeniería disciplinadas". Así, se diseñan circuitos que incluyen opciones de prueba pre-construidas o genes con una manera bien conocida para probarlos. Igual se podría hacer con el software pero no se suele hacer.

En opinión del autor de esta ponencia, Poore acierta en el análisis en el sentido de que las otras dos ingenierías usadas en la comparación nos pueden ofrecer interesantes ideas sobre cómo han abordado el problema de la complejidad, pero llega a conclusiones de utilidad sólo parciales al olvidar algo muy importante: las otras ingenierías trabajan con sistemas físicos sujetos a leyes de la naturaleza (de la física electrónica y de la bioquímica) mientras que en software se trata de sistemas no físicos donde aparecen elementos que "perturban" debido a su naturaleza diferente (factores humanos, aspectos psicológicos, procesos sociales, etc.) y que, desafortunadamente, no permiten la simplificación pura de la teoría matemática. La consecuencia de esta diferencia es que la gente actúa diferente ante unas disciplinas y otras. Así, mientras que a nadie se le ocurre pedir a un genetista que diseñe y fabrique un gen para evitar una enfermedad y lo pruebe satisfactoriamente en menos de 4/5 años, es común que a un ingeniero software se le pida que haga lo equivalente con un software para que en cuestión de pocos meses esté operativo completamente. Sería curioso hacer una encuesta entre las empresas productoras y usuarios de software para saber cuántos estarían dispuestos a

asumir sistemas de pruebas tan costosos en recursos, tiempo y dinero como los que utiliza la industria farmaceútica.

3.4. Mercado de Trabajo

A pesar de la crisis económica de los años 2002 y 2003 (caída de las llamadas puntocom), la evolución del mercado de trabajo en TI (Tecnologías de la Información) en España ha sido muy positiva. Según datos oficiales de la Encuesta de Población Activa (EPA), elaborada por el Instituto Nacional de Estadística (INE), desde 1999 hasta 2006 el empleo en TI paso de 125500 personas a 214700 (ver tabla 6); lo que supone un incremento del 71% en tan solo 7 años. Este fuerte incremento ha hecho que los ocupados en el sector hayan pasado de ser el 0'85% del total de ocupados del país a ser el 1'09%.

Año	Ocupados (miles)		
	Total	En Informática	
1999	14848,8	125,5	0,85%
2000	15681,8	148,9	0,95%
2001	16294,3	167,8	1,03%
2002	16763,1	146,6	0,87%
2003	17459,4	149,9	0,86%
2004	18129,1	177,0	0,98%
2005	18973,2	198,8	1,05%
2006	19747,7	214,7	1,09%

Tabla 6. Evolución del empleo en Informática en España (INE-EPA).

En cuanto a las áreas funcionales donde los profesionales TI desempeñan su actividad laboral, en la figura 6 se muestra el desglose, en número de personas empleadas en las distintas áreas funcionales, según el último informe anual de Telefónica [11]. Como se puede apreciar en la citada figura, la "producción de software" es, con una diferencia muy significativa, el área principal de ocupación, seguida por el área "comercial y marketing". Por tanto, cabe concluir que la formación en IS es clave para la mayoría de los puestos de trabajo de TI en España.

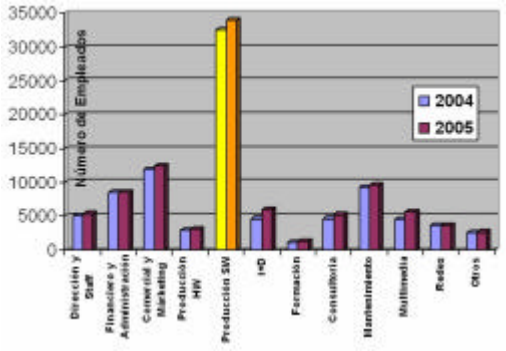

Figura 6. Desglose de personal por áreas funcionales en el sector TI en España.

No se conocen cifras a nivel nacional que desglosen con más detalle los puestos de trabajo, de forma que se pueda distinguir entre los profesionales que están meramente de programadores y los que hacen funciones de ingeniero de software. A nivel europeo sí existen datos al respecto elaborados por el ya mencionado consorcio *Career Space* en colaboración con 25 universidades de 12 países europeos, aunque son algo viejos [7]. Del total de casi 6'5 millones de empleados en TIC, las principales ocupaciones eran analistas y programadores e ingeniero de software (ver tabla 7). Entre ambas sumaban casi la mitad de todo el empleo TIC. Teniendo en cuenta que las funciones de analista son parte de la responsabilidad de un ingeniero software, se puede afirmar que ingeniero de software es, con mucho, el perfil TIC que más puestos de trabajo tiene en la Unión Europea. Además, justo ingeniero de software fue considerado por la industria como el perfil con mayores expectativas de crecimiento.

Ocupaciones (SOC90)	Ocupa-dos	Inc. 2000-04
Analistas y Programadores	1.885	+6,1
Ingenieros de Software	1.306	+10,0
Administradores de Sistemas Informáticos	1.019	+4,1
Operadores Informáticos	696	-0,5
Consultores y Gestores	437	+3,7
Ingenieros de Diseño y Desarrollo TIC	399	+0,2
Ingenieros de Computadores	348	+6,5
Ingenieros Eléctricos	203	-0,5

Ocupaciones (SOC90)	Ocupa - dos	Inc. 2000-04
Ingenieros Electrónicos	196	+3,0
Total TIC	6.489	+4,7
Total Empleo	166.696	+0,8

Tabla 7. Distribución de los ocupados en perfiles TIC en la Unión Europea (miles).

3.5. Demanda de los Estudios

Es sabido que en los últimos 5 años se ha producido una caída considerable de la demanda de acceso a los estudios de Informática en España. En 1999 era normal tener notas de corte superiores al 6 y que la demanda doblara o triplicara la oferta de plazas en las facultades y escuelas de Informática. En 8 años la situación ha cambiado completamente. Basta con comprobar que según datos oficiales del Ministerio de educación, en el curso 2006/2007 sólo se cubrieron el 68% de las plazas ofertadas en primer ciclo en los tres títulos oficiales de Informática actualmente existentes. Esta cifra es sensiblemente inferior a la media de todas las carreras técnicas (el 78%) y a la media global de todos los estudios universitarios (el 85%). Pero, ¿por qué está cayendo la demanda en las carreras de Informática?; si hemos visto que las perspectivas laborales son prometedoras y si la sociedad está cada vez más informatizada.

No es objetivo de este trabajo buscar una respuesta en profundidad a dicho interrogante, pero sí puede ser útil, desde el punto de vista del profesor de IS, conocer las causas para considerar posibles cambios en nuestra manera de actuar como individuos y colectivo.

En mi opinión, existen múltiples causas que han convergido para producir la bajada tan significativa de la demanda, si bien algunas de las cusas han ejercido de catalizadores que han acelerado el proceso.

Algunas causas, bastante conocidas, son de ámbito español:

- *Caída de la natalidad*. La gran bajada de la cifra de nacimientos hace más de 20 años se ha trasladado ya a la universidad, de forma que cada vez hay menos estudiantes universitarios en general.
- *Más trabajo no cualificado*. Los jóvenes se sienten menos proclives a estudiar en la universidad si comprueban que les es fácil trabajar sin tener dichos estudios.
- *¡Que inventen ellos!*. Una cultura subyacente todavía en una parte importante de la sociedad española considera que la ciencia y la tecnología no son parte de la cultura, con mayúsculas, y que los que se dedican a ellas son algo así como menos humanos, menos "guays".
- Imagen social devaluada del Ingeniero/a. Hace 50 años era el señor ingeniero ahora es una persona con carrera pero con un prestigio social muy devaluado debido a que no tiene en absoluto garantizada una posición social alta.

Existe otra causa que todos conocemos, y que es común para todos los países desarrollados occidentales. Se trata de que se ha impuesto socialmente una *cultura de lo fácil, del no esfuerzo*. Parece como si que los jóvenes no necesitaran tener que esforzarse para aprender y para trabajar. En España esto se ha traducido en que las carreras con fama de duras (entre las que se encuentra Informática) han tenido una evolución común negativa en sus demandas de plazas por parte de los jóvenes.

Pero todas las causas anteriores son comunes entre Informática y otras ingenierías; y en cambio, salvo Telecomunicaciones (también del sector TIC), las demás ingenierías no están teniendo una caída tan pronunciada. Por ello, surge una pregunta: ¿existe además algún factor específico de Informática?. La pista para encontrar la respuesta es que esa caída tan significativa en Informática es común a todos los países occidentales desarrollados; mientras que en los países en desarrollo (Asia, América Latina, etc.) no se ha producido el mismo fenómeno.

Denning y McGettrick publicaron un interesante artículo analizando este mismo fenómeno en los Estados Unidos [12]. Algunas cifras de lo que ha pasado en Estados Unidos pueden sorprender porque muestran un fenómeno todavía más acusado que en España:

- Los nuevos alumnos cayeron un 60% entre 2000 y 2004.

- El porcentaje del total de estudiantes que eligió estudios de Informática pasó del 3'4% en 1998 al 1'4% en 2004.
- Las perspectivas laborales son de un incremento de entre el 20% y el 50% en todos los perfiles de Informática, salvo en operadores (bajada) y programadores (estable).

Un diagnóstico habitual es que los causantes son la creencia en la pérdida del trabajo, la imagen negativa trasmitida por los medios de comunicación (el informático de parque jurásico ha hecho mucho daño), y la impresión de que Informática requiere una habilidad extraordinaria en Matemáticas (lo que asusta a muchos jóvenes). Sin embargo, Denning y McGettrick opinan que dichos factores no son convincentes porque, salvo el primero, ya existían también durante los años del boom de la demanda. Además, ya se ha comentado que el mercado de trabajo vuelve a estar boyante y no se ha notado en las preferencias de los estudiantes. Por todo ello, estos autores concluyen que existe otro factor oculto, menos evidente, que está siendo fundamental: "*The recent decreases of enrollment in computer science programs signal a chasm between our historical emphasis on programming and the contemporary concerns of those choosing careers*".

En mi opinión los citados autores aciertan con la causa profunda. Ese factor influye sólo en los países desarrollados porque durante mucho tiempo hemos "vendido" una imagen social que asocia Informática a "programar" y la gente (los jóvenes y sus padres) se hace el siguiente razonamiento: No tiene sentido estudiar una ingeniería, que encima es de las más difíciles, para luego trabajar de programador, si con eso de la globalización los hindúes y otros programan como locos por cuatro euros.

En suma, lo que está pasando en España y en los demás países occidentales desarrollados, es que el perfil de programador/codificador se está deslocalizando a países o territorios menos desarrollados, o en su defecto las empresas utilizan la globalización como excusa para pagar sueldos bajos o muy bajos para el nivel medio del país. Esto es específico de la carrera de Informática porque las demás ingenierías han transmitido una imagen de la profesión distinta, típica del ingeniero como arquitecto; mientras que nosotros hemos transmitido la imagen del albañil: informático = programador (el albañil de la informática).

En conclusión, resulta que a todos los Informáticos nos interesa cambiar la imagen tradicional de la Informática volcada en la programación y orientarla más hacia un perfil de competencias de ingeniero. El ingeniero informático no es el que programa ordenadores, sino el que diseña, construye y hace que funcionen los sistemas que resuelven los problemas de información de la gente. En esta tarea los profesores de IS deben desempeñar un papel clave; primero, convenciendo a sus demás compañeros docentes; y segundo, transmitiendo a nuestros estudiantes y a las empresas la necesidad y el interés común de dicho cambio.

Los profesores de IS tenemos un reto por adelante, hacer que nuestros egresados sean más ingenieros de software, y menos programadores de software. Con ello, ayudaremos a nuestros estudiantes a conseguir mejores puestos de trabajo, ayudaremos a las empresas de un país del nivel de España a tener los profesionales que necesita, y ayudaremos a los centros de Informática a recuperar la demanda.

Debido a la diversidad de la comunidad universitaria de Informática, es seguro que muchos académicos no estarán de acuerdo con el cambio planteado; pero no se trata de sustituir unos predominios por otros, sino de buscar un equilibrio más adecuado a los intereses de la sociedad, la tecnología y la profesión, y no sólo de la academia y la ciencia. Y es justo reconocer que hasta ahora el mundo académico no ha concedido a la IS la importancia que la sociedad necesita. Para confirmar esta afirmación basta con mirar en la tabla 8 las especialidades de los premios ACM A.M. Turing, los principales de Informática, en los últimos 40 años. La buena noticia es que aparece IS porque Frederick P. Brooks consiguió el premio en 1999 por "*landmark contributions to computer architecture, operating systems, and software engineering*".

Programación, Lenguajes	11
Informática teórica, Complejidad	10
Inteligencia artificial	6
Sistemas en Red	5
Análisis de algoritmos	4
Bases de datos	3
Sistemas operativos	3
Criptografía	3
Análisis Numérico	2
Arquitectura de computadores	1
Comunicaciones	1
Gráficos	1
Ingeniería del software	1

Tabla 8. Especialidades de los premios ACM Turing entre 1966 y 2006.

4. Adaptación al Espacio Europeo

En estos momentos estamos a la espera de un cambio incipiente para adaptar todos los títulos y planes de estudios al EEES. A continuación se presenta la situación en cuanto a lo que tiene que ver con la IS.

4.1. Libro Blanco de Informática

Por encargo de la ANECA (Agencia Nacional de Evaluación de la Calidad y Acreditación), una comisión de la Conferencia de Decanos y Directores de Informática (CODDI) elaboró un libro blanco sobre la adaptación al EEES de los estudios de Informática. Es de resaltar que en el proyecto participaron 56 universidades españolas, prácticamente todas las que tienen estudios de Informática. Entre otros análisis realizados, los autores estudiaron las diversas titulaciones existentes en diversos países en el ámbito de la Informática, para lo cual se basaron en los principales currículos internacionales (ya comentados algunos anteriormente):

- Ciencias de la Computación (Computer Science),
- Sistemas Informáticos (Computer Engineering),
- Ingeniería del Software (Software Engineering),
- Sistemas de Información (Information Systems),
- Redes de Computadores; y
- Sistemas Multimedia.

Las principales propuestas de este documento [13] son las siguientes [3]:

1. Estructura organizada en dos ciclos, Grado y Máster.
2. Una única titulación de grado, denominada Ingeniería Informática.
3. El título de grado comportará competencias profesionales plenas para el ejercicio de la profesión.
4. La titulación de Ingeniería Informática proporcionará una formación generalista, con especial cuidado en la transmisión de los fundamentos de la disciplina y en la generación de habilidades y capacidades para aprender a lo largo de toda la vida profesional. Se pretende que la especialización se realice dentro de los estudios de Master, u otros programas orientados a la formación permanente a lo largo de toda la vida.
5. Los estudios de Grado constarán de 240 créditos ECTS organizados en 4 años (8 semestres).
6. Entre los contenidos formativos fundamentales del primer ciclo o título de grado, debe incluirse la realización de un Proyecto Fin de Carrera, que integre los conocimientos adquiridos durante los estudios y aproxime al estudiante a casos reales de la profesión, así como contenidos transversales que potencien habilidades propias de los ingenieros.
7. La titulación de grado tendrá un 60% de créditos de Contenidos Formativos Comunes (CFC), incluyendo la carga asignada al Proyecto Fin de Carrera, dejando el 40% restante para materias que serán determinadas discrecionalmente por cada Universidad.
8. Entre las materias a determinar por las Universidades, se recomienda tener una oferta suficiente de optativas que procure una formación amplia al estudiante en Tecnologías Informáticas actuales así como conocimientos de dominios concretos de aplicación de la informática.
9. El Master estará destinado a la especialización profesional de los Ingenieros en Informática, o bien a su preparación para la investigación.

[3] En opinión del autor algunas de dichas propuestas han quedado desfasadas por recientes cambios normativos y legales del Ministerio de Educación.

10. El número de titulaciones de Master y su orientación queda abierto para poder adaptarse a la demanda de formaciones especializadas en cada momento.
11. Los estudios de Master constarán de entre 60 y 120 créditos ECTS, y podrán incluir la cantidad asignada a la Tesis de Master.
12. El Master deberá permitir el acceso a la realización de la tesis doctoral con el objeto de obtener el grado de Doctor.

Adicionalmente al libro blanco, la CODDI realizó varias recomendaciones complementarias en su reunión de abril de 2005. Las principales fueron:

- Todas las titulaciones de Ingeniería deberán tener una estructura uniforme que fije un referente académico y profesional claro para la sociedad.
- Los estudios universitarios en el área de las TIC deberán estructurarse en torno a dos únicos títulos, Ingeniería en Informática e Ingeniería de Telecomunicación, que cubran las distintas competencias y atribuciones profesionales de acuerdo a su especialización.

El libro blanco establece tres perfiles profesionales de grado:

- Desarrollo de Software (DS)
- Sistemas (Sis)
- Gestión y Explotación de Tecnologías de la Información (GETI)

Estos perfiles son bastante amplios y buscan satisfacer las principales necesidades del sector en España. Cabe destacar que en los tres juega un papel importante la Ingeniería del Software y otras materias muy relacionadas con ésta. Dicho papel es central en el perfil de desarrollo de software (que ya se vió que era el más demandado por el mercado de trabajo). Prueba de ello es que entre las competencias específicas (propias de la profesión) que deben tener los graduados de los tres perfiles está incluida la Ingeniería del Software y materias cercanas con una importancia destacable (asignando una importancia entre 1 - mínimo, y 4 - máximo), sobre todo en los perfiles de Desarrollo de Software y de Gestión y Explotación de las Tecnologías de la Información, como se muestra en la tabla 9.

Competencias Específicas	DS	Sis	GETI
Capacidad para entender y evaluar especificaciones internas y externas	4	3	3
Dirección, planificación y gestión de proyectos	4	4	4
Diseño y arquitectura de sistemas de información	4	1	4
Ingeniería del Software	4	1	3
Métodos y herramientas para el diseño y desarrollo de sistemas basados en computadores	4	3	3

Tabla 9. Importancia de las competencias específicas para cada perfil profesional.

De forma más precisa, se establecen las siguientes capacidades, relacionadas con la IS, en cada uno de los tres perfiles:

a) Desarrollo de Software:

- Dirigir y coordinar el proyecto de desarrollo y mantenimiento de aplicaciones, supervisando las funciones y recursos de análisis funcional, orgánico y programación, asegurando la adecuada explotación de las aplicaciones.
- Dominar todas las etapas de la vida de un proyecto (análisis de concepción, análisis técnico, programación, pruebas, documentación y formación de usuarios).
- Supervisar y coordinar el desarrollo completo de aplicaciones y administrar la introducción de los sistemas de gestión.
- Analizar y recoger nuevas técnicas y herramientas del mercado estudiando su viabilidad y necesidad.
- Interpretar las especificaciones funcionales encaminadas al desarrollo de las aplicaciones informáticas.
- Realizar el análisis y el diseño detallado de las aplicaciones informáticas.
- Realizar las pruebas que verifiquen la validez funcional, la integridad de los datos y el rendimiento de las aplicaciones informáticas.
- Estudiar el sistema actual existente y analizar e idear mejores medios para llevar a cabo los mismos objetivos u otros adicionales.

- Participar en el diseño de nuevos sistemas informáticos como consecuencia de la informatización de áreas de la empresa que utilizan para el desarrollo de sus tareas métodos y procesos manuales.
- Escuchar y asesorar a los usuarios, en la resolución de los problemas que se les plantean con el uso de los sistemas informáticos.
- Mantenerse al día en técnicas, métodos y herramientas de análisis y diseño.

b) Sistemas:
- Diseño de las soluciones informáticas relacionadas con los cambios en los sistemas existentes o con los nuevos sistemas.
- Dirección y asesoramiento a los programadores en la realización de los programas.
- Creación de los test de pruebas para verificar que los sistemas informáticos cumplen los requisitos y especificaciones de análisis y diseño.
- Estudio de métodos, técnicas y herramientas de análisis y diseño.

c) Gestión y Explotación de Tecnologías de la Información:
- Evalúa los riesgos empresariales asociados a los sistemas informáticos y establece las orientaciones y directrices para mitigarlos.
- Establece las directrices sobre las métricas e indicadores que serán utilizados para permitir a la dirección de la empresa la evaluación y el seguimiento de los sistemas informáticos.
- Realizar estudios funcionales y proyectos específicos.
- Concreción de objetivos de cualquier sistema informático.
- Estudio de los riesgos de los sistemas informáticos.
- Analizar los proyectos y las necesidades y proponer soluciones en el plano técnico, humano y financiero.
- Definir con mayor precisión las necesidades técnicas del cliente.
- Análisis de modelos de negocio asociados a la definición de nuevos productos o servicios.
- Definir normas de desarrollo en colaboración con la dirección de informática.

4.2. Propuesta de Directrices

Más recientemente, el Ministerio de Educación y Ciencia, tomando como base el libro blanco de Ingeniería Informática ha presentado la "Ficha Técnica de Propuesta de Título Universitario de Grado en Ingeniería Informática" [14]. La versión definitiva de este documento debería definir los puntos cardinales de la Ingeniería Informática. Las principales directrices son:
- El número total de créditos de formación académica básica que debe superar el estudiante es de 180 ECTS, distribuidos en 3 cursos académicos. De estos créditos, 140 serán de Contenidos Formativos Comunes, dejando los 40 restantes para materias optativas (Contenidos Formativos Específicos).
- El número de créditos de formación adicional de orientación académica o profesional que debe superar el estudiante es de 60 ECTS, de los que al menos 30 corresponderán al proyecto fin de carrera.
- Dentro de las materias de la titulación, se distinguen dos categorías (ver tabla 10): i) Materias Instrumentales, que incluye Fundamentos Matemáticos de la Informática, Fundamentos Físicos de la Informática y Gestión de las organizaciones; y ii) Materias Propias, que incluye Programación, Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes, Ingeniería de Computadores, Sistemas Operativos, Sistemas Distribuidos y Redes, y finalmente, Aspectos Profesionales de la Ingeniería Informática.
- Sin imponer a las Universidades la definición de perfiles profesionales o menciones específicas para los títulos de grado en Ingeniería Informática, se recomienda (al igual que el Libro Blanco) la implantación de las menciones de "Desarrollo de Software", de "Sistemas y Redes" y de "Gestión y Explotación de la Información". Estas menciones tendrán asignados un mínimo de 30 ECTS específicos adicionales a los indicados en las Directrices.

Tipo de Contenidos	ECTS	
TOTAL TÍTULO DE GRADO	240	
CONTENIDOS FORMATIVOS COMUNES	140	min
Materias Instrumentales	38	min
Fundamentos Matemáticos de la Informática	24	min
Fundamentos Físicos de la Informática	8	min
Gestión de las Organizaciones	6	min
Materias Propias	102	min
Programación	27	min
Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes	30	min
Ingeniería de Computadores	18	min
Sistemas Operativos, Sistemas Distribuidos y Redes	21	min
Aspectos Profesionales de la Ingeniería Informática	6	min
CONTENIDOS FORMATIVOS ESPECÍFICOS	40	max
Menciones (perfiles profesionales)	30	min
FORMACIÓN ADICIONAL DE ORIENTACIÓN ACADÉMICA O PROFESIONAL	60	
Proyecto Fin de Carrera	30	min
Formación Adicional	30	max

Tabla 10. Distribución de créditos ECTS de las directivas de grado de Informática.

Entre las capacidades, competencias y destrezas generales del título, aparecen las siguientes de interés para la IS:

- Concebir, desarrollar y mantener sistemas y aplicaciones software empleando diversos métodos de ingeniería del software y lenguajes de programación adecuados al tipo de aplicación a desarrollar manteniendo los niveles de calidad exigidos.
- Concebir, desarrollar, mantener y utilizar aplicaciones informáticas de cualquier índole …
- Dirigir y coordinar grupos de trabajo en el ámbito …, proponiendo métodos de trabajo estándar y herramientas a utilizar.

En mi opinión, el desglose de contenidos mostrado en la tabla 10, que es heredado del que se hizo en el libro blanco, peca de demasiado general. Debería haberse intentado una división algo mayor porque algunos contenidos formativos parecen un cajón de sastre con materias demasiado amplias y diversas dentro. No parece razonable que dos de los cinco currículos internacionales de ACM (sistemas de información e IS) se incluyan en el CFC de "Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes". El resultado es una lista de conocimientos, capacidades y destrezas a adquirir demasiado amplia:

- Conocer y aplicar los métodos de desarrollo de software así como las técnicas de calidad del software.
- Planificar y gestionar el desarrollo de proyectos informáticos.
- Conocer las bases para el diseño y evaluación de interfaces de usuario y saber establecer la interacción persona-computadora más adecuada.
- Definir bases de datos relacionales y orientadas a objetos.
- Emplear sistemas de gestión de bases de datos en entornos centralizados y distribuidos.
- Utilizar lenguajes de consulta.
- Sistemas Integrados.
- Conocer técnicas de organización y recuperación de información.
- Utilizar técnicas de inteligencia artificial para diversos problemas.
- Emplear técnicas de minería de datos.

Tanta amplitud difumina el objetivo básico de diferenciar los contenidos que todos los informáticos deben recibir (antiguas materias troncales) de los que sólo interesan en algún perfil concreto. Por ello, creo que se debería precisar, de forma diferenciada, la cantidad de IS que todos los estudiantes de grado de Informática deben cursar.

4.3. Reforma de Planes de Estudio y Asignaturas

No es cometido individual de un profesor de IS reformar un plan de estudio, pero si es su responsabilidad intentar que los planes de estudios reformados incorporen de forma adecuada la

perspectiva de ingeniería en general y la formación en IS en particular. A la hora de abordar esta tarea existe la dificultad de que no está claro qué y cómo se debe realizar. Para ayudar y orientar en este sentido existe el informe "Adaptación de los Planes de Estudio al Proceso de Convergencia Europea" de la Dirección General de Universidades [15]. En este informe se explica un método, con un enfoque top-down de lo general a lo particular y guiado por competencias, para diseñar planes de estudio adaptados al EEES. Establece las siguientes etapas [4]:

1. Delimitación de los objetivos y el perfil académico y profesional de la titulación.
2. Establecer la estructura organizativa del plan así como las competencias y los contenidos propios del mismo.
3. Concretar las modalidades del proceso de enseñanza-aprendizaje para el desarrollo metodológico del programa formativo.
4. Efectuar una previsión sobre los recursos humanos y materiales necesarios.
5. Especificar los requisitos administrativos que regulan la gestión de todos los aspectos implicados en las enseñanzas del plan.

No existen experiencias del caso anterior con planes de estudios oficiales porque todavía estamos a la espera de las normas definitivas. Pero lo que es común en casi todas las universidades son experiencias parciales de reforma de una asignatura individual o un grupo de asignaturas afines. Para estos casos, basándonos en el informe anterior, recientemente hemos publicado un método para el diseño de asignaturas guiadas por competencias [16], que consta de las 6 etapas siguientes:

1. Identificar competencias, (en base al libro blanco y otras propuestas curriculares).
2. Establecer la estructura general de los contenidos, (establecer unidades docentes compuestas de 1-n temas)
3. Definir los tipos de actividades de enseñanza-aprendizaje, obteniendo una lista de los tipos de actividad adecuados y su importancia (peso). El peso de cada tipo se determina en función de las competencias y los contenidos.
4. Estimar el esfuerzo del alumno, es decir, horas necesarias para cada tipo de actividad en cada contenido (tema o práctica).
5. Elaborar el calendario, indicando cada día o semana los tipos de actividad y contenidos que se trabajarán.
6. Definir un método de evaluación continua, basado en un sistema de puntos acumulativos; de forma que el alumno conozca desde el principio la lista de indicadores (maneras de conseguir puntos) y a lo largo del curso vaya conociendo de forma continua su evolución (puntos acumulados hasta la fecha).

En [16] se exponen también los resultados (positivos) de su aplicación a una asignatura que combina contenidos de gestión de negocios, gestión de proyectos e ingeniería del software.

Los tipos de actividades de enseñanza-aprendizaje vienen determinados en gran parte por la naturaleza de cada materia. En el caso de la IS, una lista inicial de partida donde elegir puede ser la siguiente:

- Clases magistrales tradicionales.
- Clases de debate y/o dudas.
- Tutorías en grupo (normalmente para seguimiento de proyectos o trabajos).
- Proyectos desarrollados en equipo.
- Tutorías individuales.
- Estudio (del alumno).
- Trabajos adicionales.
- Ejercicios y problemas.
- Exámenes.
- Seminarios y conferencias.
- Visitas a/de empresas.

4.4. Lecciones Aprendidas

Con nuestra experiencia personal hemos aprendido que, de cara a la reforma de la enseñanza de IS, dan buenos resultados las siguientes prácticas:

- Aplicar a la enseñanza/aprendizaje de IS los principios y buenas prácticas de la ingeniería presentados al principio de esta ponencia.
- Utilizar la técnica de aprendizaje basado en problemas (ABP), donde el problema es un proyecto de desarrollo de software.

[4] El autor tiene una presentación detallada de este método de adaptación de planes de estudios en http://alarcos.inf-cr.uclm.es/per/fruiz/conf/eees/eees.htm.

- Que los alumnos trabajen en equipo, organizando grupos de 4-6 alumnos y asignando roles similares a los reales (jefe de proyecto, etc.).
- Hacer un seguimiento y control del trabajo del alumno mediante la evaluación continua con entregables pequeños en vez de uno grande al final. Es mejor para todos (alumnos y profesores) tener que entregar/evaluar un entregable pequeño cada cierto tiempo que uno grande, formado por los pequeños, al final del curso.
- Utilizar fuentes internacionales. En especial, emplear SWEBOK como referencia de contenidos, y algunos cursos seleccionados de los currículos de ACM.

Otro aspecto clave es que con la adaptación al EEES los profesores tenemos que cambiar de chip, dejando de pensar sólo en términos de "nuestra asignatura" a tener una perspectiva más holística. Deberíamos comenzar a pensar en un grupo de asignaturas relacionadas ideando prácticas conjuntas, utilizando las mismas herramientas, o estableciendo pruebas de evaluación horizontales conjuntas. Las materias más claramente candidatas a forma grupo con la IS son Programación, Gestión de Información (sistemas de información y bases de datos) y Gestión de Proyectos (ver figura 7).

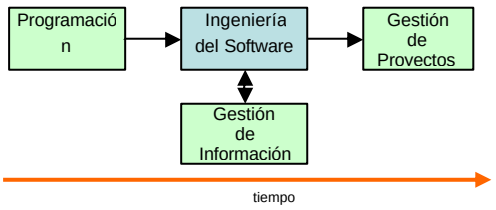

Figura 7. Grupo de materias candidatas para actividades de enseñanza/aprendizaje conjuntas.

Algunas ideas a la hora de establecer dichas actividades conjuntas con IS son:

- Con Gestión de Información: integrar análisis/diseño de datos con el análisis/diseño general del sistema, teniendo que desarrollar un proyecto conjunto y pruebas comunes. Para poder hacer lo anterior es necesario que ambas materias se impartan de forma paralela.
- Con Gestión de Proyectos: después de que en IS se aprendan los métodos y técnicas específicos de proyectos software, encuadrar dichas técnicas en un marco general de gestión de proyectos. Por ejemplo, utilizar PMBOK (*Project Management Body of Knowledge*) [17] para enseñar qué es la gestión de proyectos; y en cada parte (alcance, tiempos, costes, riesgos, etc.) referenciar a las técnicas correspondientes ya aprendidas en IS.
- Con Programación: evitar que los estudiantes dediquen parte del esfuerzo (tiempo) destinado a adquirir competencias de IS a programar. Esto no significa que no sea posible organizar actividades conjuntas para ambas materias. Por ejemplo, se pueden idear proyectos en los cuales los alumnos de cursos iniciales hacen de programadores y los alumnos de cursos superiores hacen de ingenieros de software.

5. Conclusiones

En esta ponencia se ha reflexionado sobre la situación cambiante de la IS dentro del marco general de la Informática. En opinión del autor, los datos e informaciones expuestos aquí deberían conducir a que la IS tenga un papel más relevante en los estudios de Informática. Además, es importante que, de forma transversal, sepamos transmitir a nuestros estudiantes una perspectiva de ingeniería.

Este cambio de planteamiento y de peso de la IS en las carreras de Informática puede ser llevado a cabo con motivo de la próxima reforma de los títulos y planes de estudios, para su adaptación al EEES. En esta línea, se aportan varios métodos y prácticas que puedan ser de ayuda para dicho cambio.

Agradecimientos

Parte de las reflexiones e informaciones de este trabajo fueron realizadas gracias a la reforma legal del Ministerio de Educación del año 2001. Sin ella el autor no hubiera tenido que superar un reciente proceso de habilitación nacional, tan competitivo,

que le motivo a preparar un proyecto docente a conciencia. Gracias a los compañeros Félix García y Eduardo-Fernández Medina, por facilitarme datos e informaciones actualizados de sus proyectos docentes, realizados por los mismos motivos.

Referencias

[1] IEEE (1990): *Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology*.

[2] Booch, G. (2007): *The Promise, The Limits, The Beauty of Software.* Computer Science Teachers Association, ACM. Disponible en http://csta.acm.org/Resources/sub/Turing_Lecture.ppt.

[3] IEEE-CS (2004): Guide to the Software Engineering Body of Knowledge, 2004 version. IEEE Computer Society. Disponible en http://www.swebok.org/.

[4] ACM (2004): *Software Engineering 2004 – Currículo Guidelines for Undergraduate Degree Programs in Software Engineering.* Disponible en http://www.acm.org/education/curricula.html.

[5] ACM (2005): *Computing Curricula 2005: The Overview Report.* Disponible en http://www.acm.org/education/curricula.html.

[6] Career Space (2001a): *Nuevos currículos de TIC para el siglo XXI: el diseño de la educación del mañana*. Oficina de Publicaciones las Comunidades Europeas, Luxemburgo. Disponible en http://www.career-space.com/downloads/index.htm.

[7] Career Space (2001b): *Determining the future demand for ICT skills in Europe. International Cooperation Europe Ltd.* Disponible en http://www.career-space.com/downloads/index.htm.

[8] León, G., Dueñas, J.C., Bernardos, A. et al. (2002): *Alternativas y oportunidades para la implicación empresarial en las futuras estructuras curriculares relacionadas con las TIC*. ANIEL. Informe PAFET II.

[9] Dahlbom, B. & Mathiassen, L. (1997): *The future of our profession*. Communications of the ACM, v.40 n.6, p.80-89.

[10] Poore, J.H. (2004): *A Tale of Three Disciplines ... and a Revolution*. IEEE Computer, 37(1), 30-36.

[11] Telefónica (2007): *La Sociedad de la Información en España 2006*. Disponible en http://www.telefonica.es/sociedaddelainformacion/.

[12] Denning, P.J. & McGettrick , A. (2005): *The profession of IT: Recentering computer science*. Communications of the ACM, 48(11), 15-19.

[13] ANECA (2005): *Libro Blanco del Título de Grado en Ingeniería Informática*. Agencia Nacional de Evaluación de la Calidad y Acreditación; Proyecto EICE. Disponible en http://www.aneca.es/modal_eval/conver_docs_titulos.html.

[14] MEC (2006): Ficha Técnica de Propuesta de Título Universitario de Grado en Ingeniería Informática. Disponible en http://www.mec.es/educa/ccuniv/html/GRADO_POSGRADO/Documentos/Ficha%20Ingenier%EDa%20Inform%E1tica.pdf.

[15] DGU (2004): *Adaptación de los Planes de Estudio al Proceso de Convergencia Europea*. Dirección General de Universidades, Proyecto EA 2004-0024.

[16] Ruiz, F. y García, F. (2007): *Diseño integral de una asignatura para una formación basada en competencias*. XIII Jornadas de Enseñanza Universitaria de la Informática (JENUI'2007). Teruel, 14-16 de Julio.

[17] PMI (2004): *A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Third Edition*. Project Management Institute.