

SiNDI

Simposio Nacional de Docencia
en la Informática

La Enseñanza de la Ingeniería del Software en el marco del EEES

Francisco Ruiz

Universidad de Castilla-La Mancha

II Congreso Español de Informática

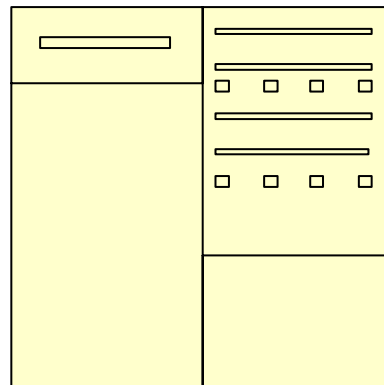
II Simposio Nacional de Docencia en la Informática

Zaragoza, septiembre 2007



- **Contexto de la Ingeniería del Software**
 - Evolución Histórica
 - ¿Por qué Ingeniería del Software?
 - Perspectiva de Ingeniería
 - Cuerpo de Conocimientos - SWEBOK
- **Ingeniería del Software vs Informática**
 - Currículos Internacionales
 - Mercado Profesional
 - Demanda de Estudios
 - Academia
- **Adaptación al EEES**
 - Libro Blanco
 - Propuesta de Directrices
 - Reforma de los Planes de Estudios
 - Algunos Consejos
- **Conclusiones**

- 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*”.



Lenguajes de alto nivel

Estructuración

Modularidad

Tipos abstractos

Bases de datos

Orientación a objetos



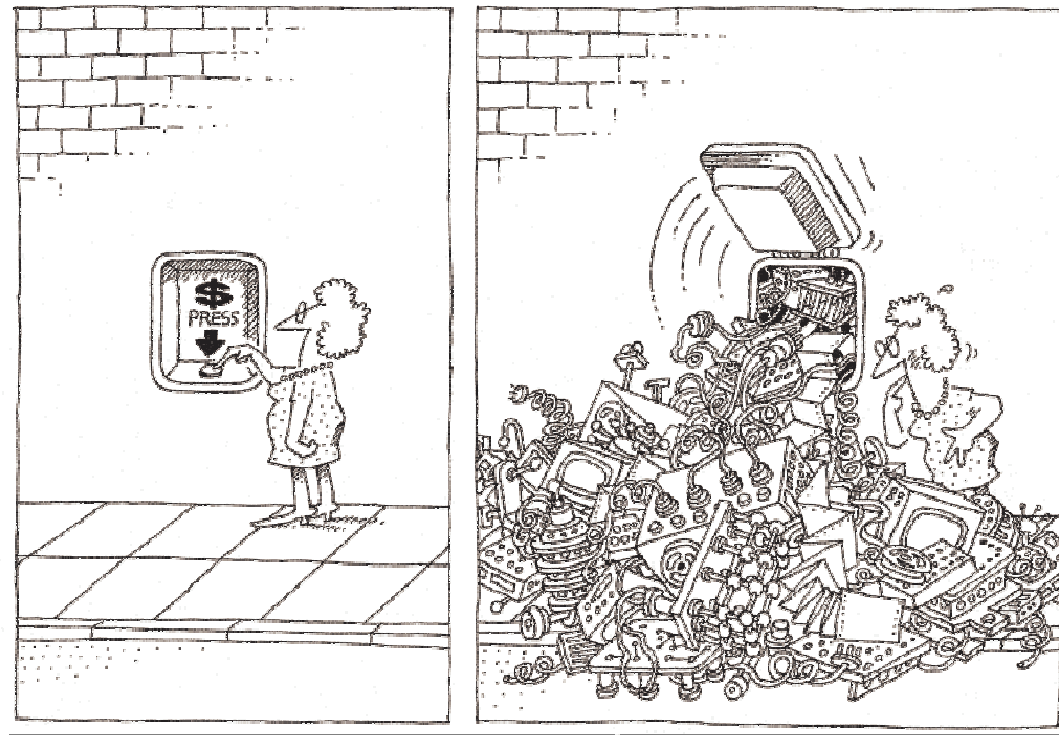


- Pero seguimos teniendo otros retos pendientes:
 - Si hemos ido subiendo de **nivel de abstracción** en los lenguajes de programación, ¿nos permite la tecnología actual dar otro salto más?
 - Java es código fuente, y ¿UML no?.
 - ¿Existe alguna manera de construir software más rápida y con menos errores?.
 - La **integración** sigue siendo un problema difícil.
 - Integrar sistemas
 - Integrar tecnologías
 - Seguimos teniendo dificultades para **entender** bien a los clientes/usuarios.
 - Muchos proyectos técnicamente correctos fracasan (el software no sirve a los supuestos destinatarios o no lo usan).
 - El software es la **red**.
 - El concepto clásico cerrado de “aplicación” software está desapareciendo.



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

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





- La ingeniería existe porque las personas diseñan y construyen artefactos/sistemas cada vez más complejos.
- Probablemente, 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), ..



- ¿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).
 - A un servicio de un abogado en un juicio (que nos ayuda con su conocimiento especializado).
- ¿Producto o Servicio?.
- Entonces, ¿la gente que hace software qué clase de habilidades y capacidades debe tener?
 - Arquitecto
 - Albañil
 - Jardinero
 - Artista

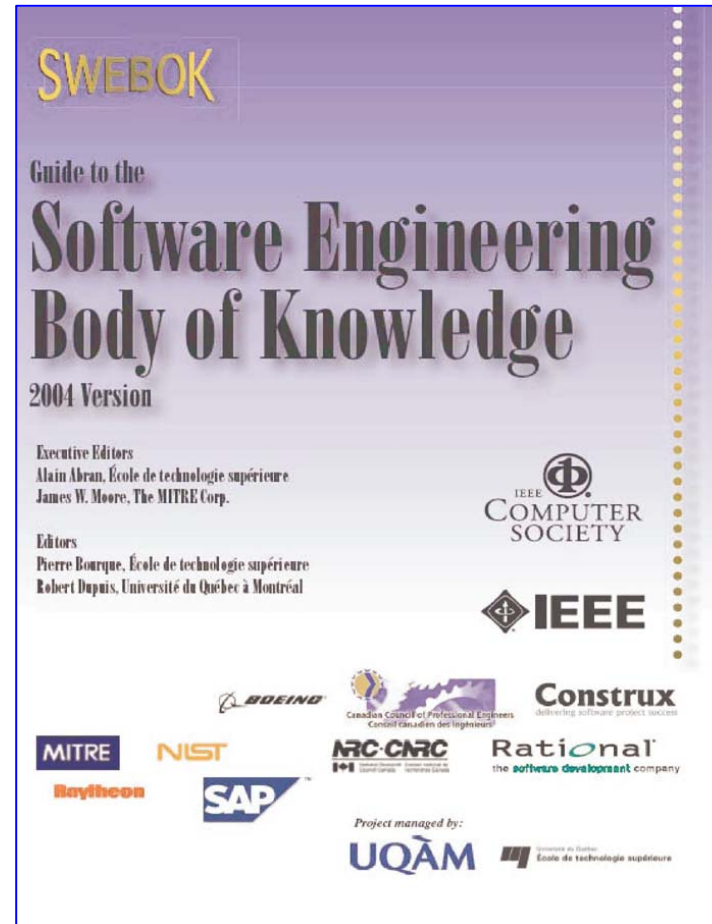


El **sentido común** es muy importante.

- Ley del Mínimo Esfuerzo
 - Entre las opciones correctas elegir la más sencilla.
 - Reutilización
 - Del código, del resto de artefactos software y del conocimiento.
- No inventar la rueda
 - Emplear estándares.
- Zapatero a tus zapatos
 - No linealidad entre escala vs complejidad.
 - No es lo mismo construir un pequeño muro de ladrillos que una casa, un edificio de 5 plantas o un rascacielos.
- Aprender de la experiencia (nuestra o de otros).
 - Utilizar “buenas prácticas”, “lecciones aprendidas”, “patrones”.

Software Engineering Body of Knowledge

<http://www.swebok.org/>





RESUMEN DE ÁREAS: (133 pgs)

- Fases del Proceso de Desarrollo (69 pgs)
 - Requisitos (17 pgs)
 - Diseño (12 pgs)
 - Construcción (9 pgs)
 - Pruebas (16 pgs)
 - Mantenimiento (15 pgs)
- Perspectiva de Ingeniería (64 pgs)
 - Gestión de la Configuración (gestión de productos) (14 pgs)
 - Gestión de la Ingeniería (gestión de proyectos) (13 pgs)
 - Proceso de Ingeniería (orientación a procesos) (14 pgs)
 - Herramientas y Métodos (tecnología de soporte) (9 pgs)
 - Calidad (14 pgs)
- Disciplinas Relacionadas



ACM distingue 5 currículos diferentes

- 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 qué los distingue?

¿Es que Informática no es una profesión, y sí lo es Ingeniería del Software?



ACM Computing Curricula 2005. Overview Report

- Pesos asignados a los tópicos de Ingeniería del Software

Área de Conocimiento	CE		CS		IS		IT		SE	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Fundamentos de Programación	4	4	4	5	2	4	2	4	5	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
.....										
Desarrollo Digital	0	2	0	1	1	2	3	5	0	1
.....										



Papel Social del Informático (Dahlbom & Mathiassen, 1997):

	Construyo cosas	Ayudo a la gente	Cambio las cosas
Orientación	Máquina	Cultura	Poder
Actividad	Construcción	Evolución	Intervención
Papel	Ingeniero	Facilitador (téc. soporte)	Emancipador (consultor)

Esto implica tres tipos de roles diferentes dentro de la informática:

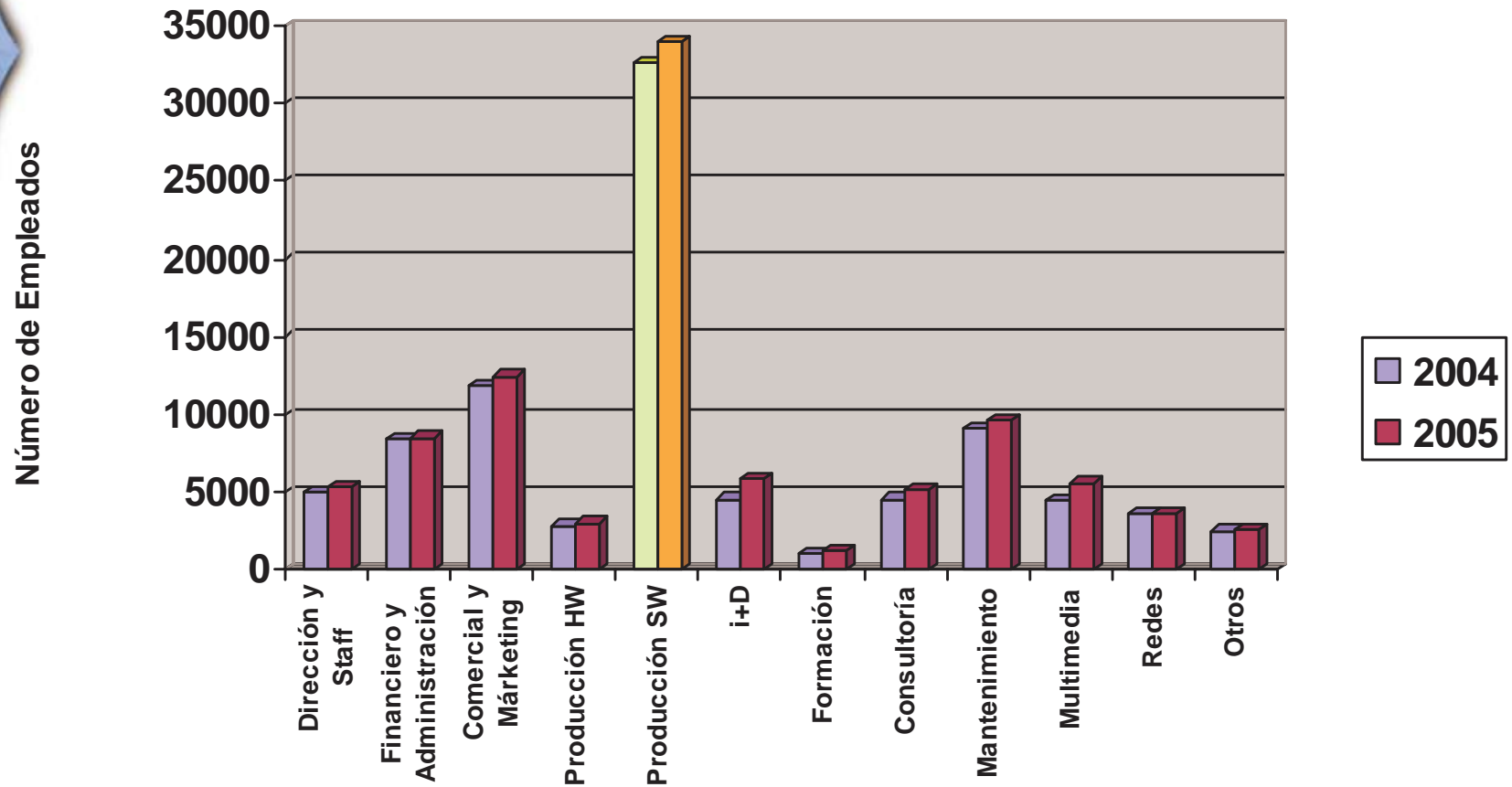
- los ingenieros de **desarrollo**: construyen artefactos software o hardware;
- el personal de **soporte**: ayuda a usar aquello que otros desarrollaron; y
- los **consultores**.



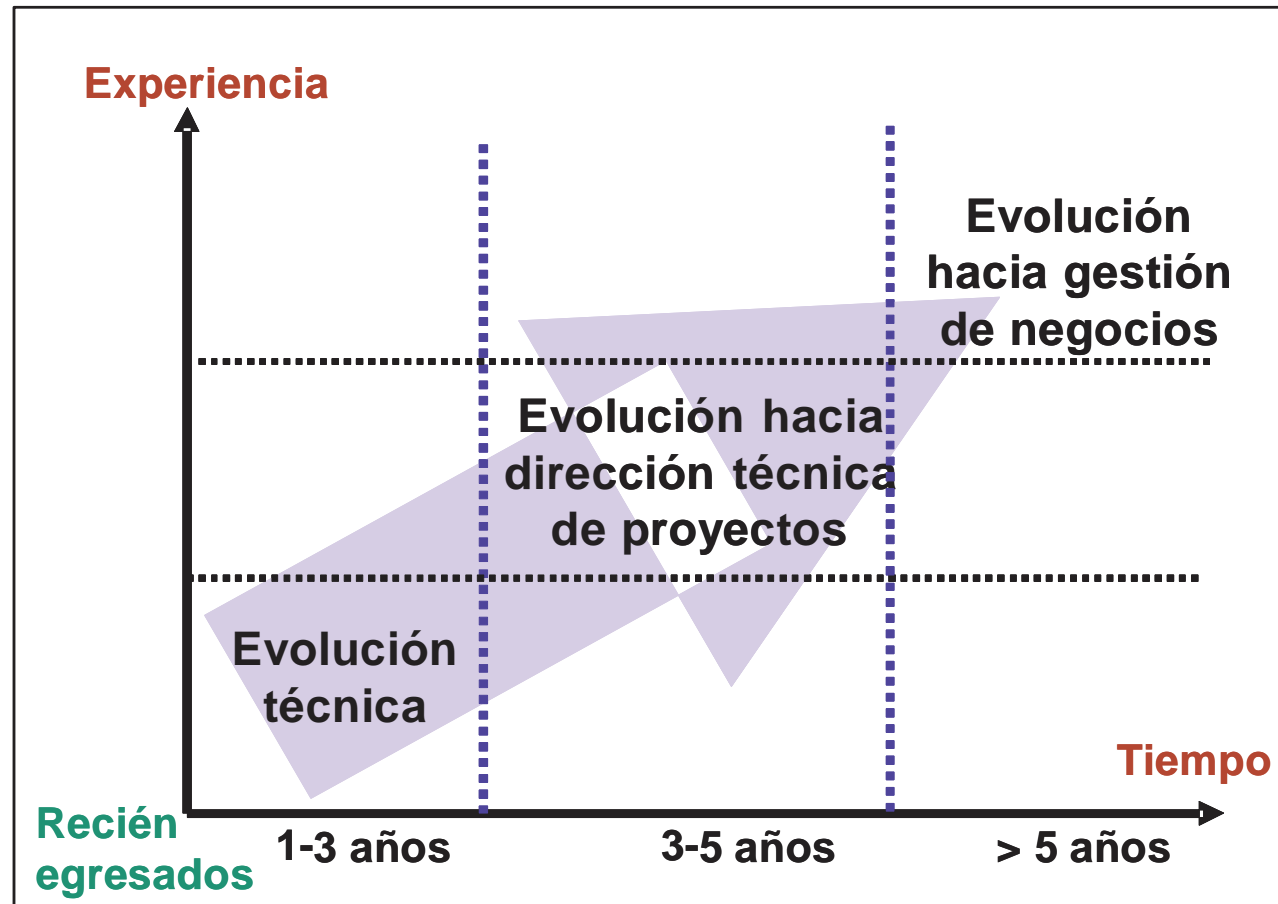
Distribución de los ocupados en perfiles TIC en la Unión Europea 15 (miles), *Career-Space*.

Ocupaciones (SOC90)	Total Puestos TIC	% Incr. 2000-2004
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
Ingenieros Electrónicos	196	+3,0
Total TIC	6.489	+4,7
Total Empleo	166.696	+0,8

Telefonica (2007): Personal por áreas funcionales en el sector de TI en España).



Informe PAFET (2002): Evolución profesional habitual de los profesionales TIC.





¿Por qué está cayendo la demanda en las carreras de Informática?

- Si hemos visto que las perspectivas laborales son prometedoras.
 - Si la sociedad está cada vez más informatizada.
-
- ¿Sólo ocurre con Informática o es un fenómeno más general?.
 - ¿Sólo ocurre en España o es más global?.
 - ¿existe algún factor específico sólo de Informática?



- Peter J. Denning and Andrew McGettrick. CACM 48(11), nov-2005.
 - **Recentering Computer Science**
 - *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.*
- Algunas cifras de lo que ocurre en EE.UU.
 - Los nuevos alumnos cayeron un 60% entre 2000 y 2004.
 - Las perspectivas laborales son de un incremento de entre el 20% y el 50% en todas las especialidades, salvo en operadores (bajada) y programadores (estable).



- 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.

ES EL SENTIDO COMÚN DE NUESTRA GENTE

! HAGAMOS QUE NUESTROS TITULADOS SEAN MAS INGENIEROS DE SOFTWARE Y MENOS PROGRAMADORES DE SOFTWARE !



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

Especialidad de los ganadores de los premios Turing de ACM entre 1966-2006



- El libro blanco de la titulación de Ingeniería en Informática (ANECA, 2005) establece tres **perfiles profesionales de grado**:
 - Desarrollo de Software
 - Sistemas
 - Gestión y Explotación de TI
- Estos perfiles son bastante amplios y buscan satisfacer las principales necesidades del sector en España.

- En los tres perfiles juega un papel importante la Ingeniería del Software:

Competencias Específicas	Desarrollo de Software	Sistemas	Gestión y Explotación de las TI
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

- Ingeniería del Software aparece como eje de uno de los contenidos formativos comunes

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



- Entre las capacidades, competencias y destrezas generales aparecen:
 - **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.



- El CFC de “**Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes**” es el que más carga tiene en créditos.
- Incluye los siguientes conocimientos, capacidades y destrezas:
 - **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.**
 - 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.



Fuente de Información General


- Dirección General de Universidades. Adaptación de los Planes de Estudio al Proceso de Convergencia Europea. Proyecto EA 2004-0024. Octubre, 2004.

Experiencia real con una Asignatura

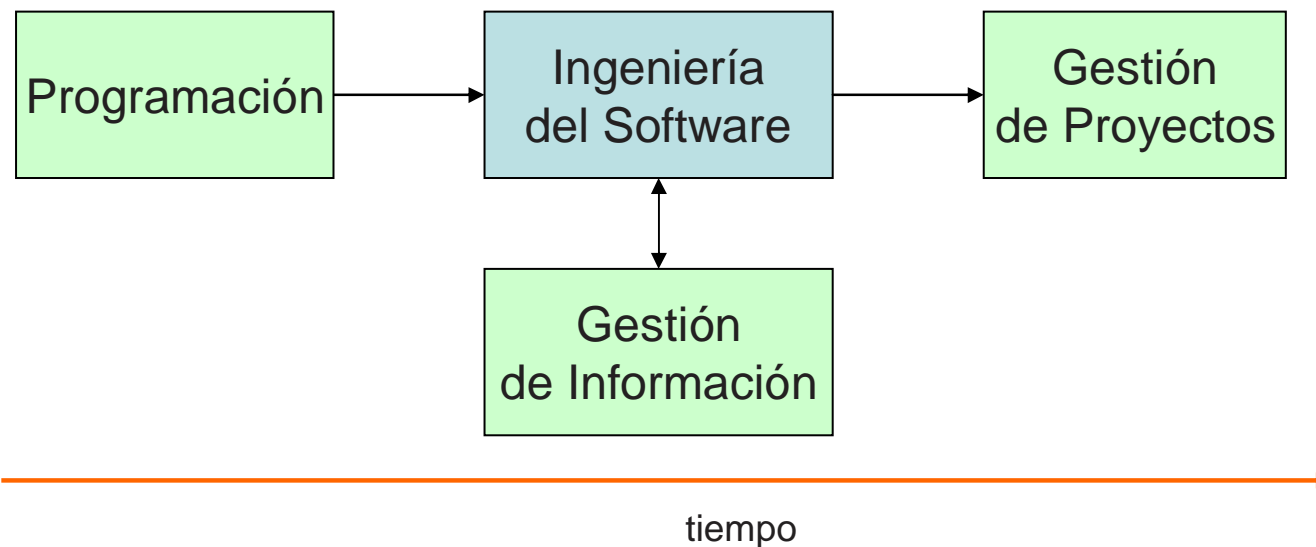
- Francisco Ruiz, Félix García (2007): Diseño integral de una asignatura para una formación basada en competencias. XIII Jornadas de Enseñanza Universitaria de la Informática (JENUI'2007). Teruel, 14-16 de Julio.
 - Planificación y Gestión de Sistemas de Información
 - <http://alarcos.inf-cr.uclm.es/per/fruiz/conf/eees/eees.htm>
 - <http://alarcos.inf-cr.uclm.es/doc/pgsi>



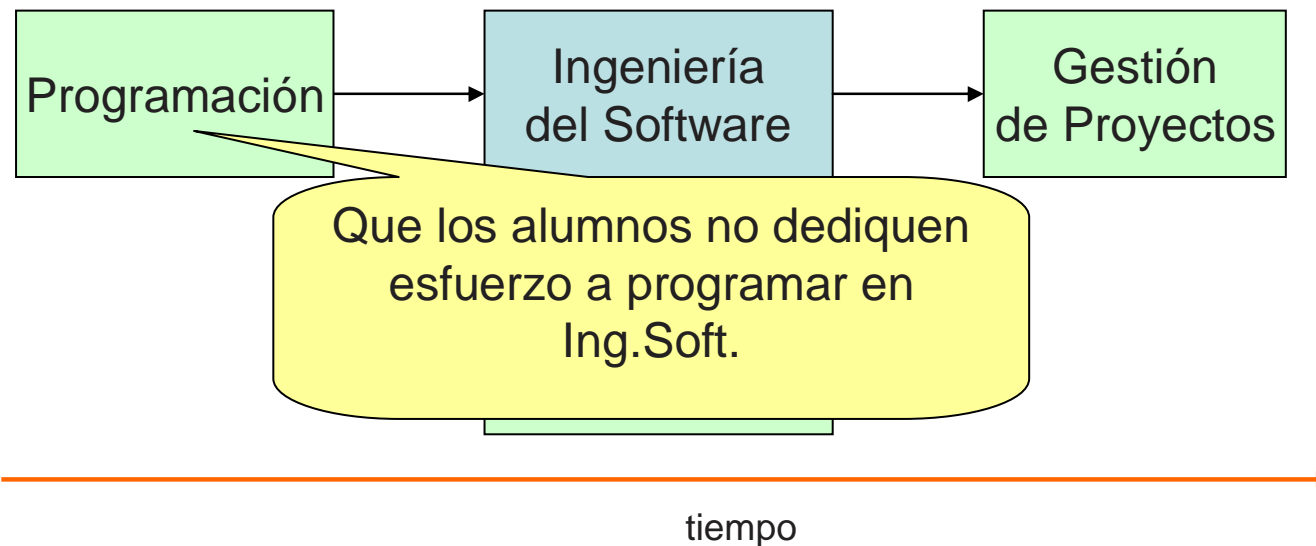
1. **Identificar competencias**
A partir del libro blanco y propuestas internacionales
2. **Estructura general de los contenidos**
Unidades docentes (1-n temas)
3. **Tipos de actividades de enseñanza-aprendizaje**
Lista y peso de cada una (según su papel respecto de las competencias y los contenidos)
4. **Esfuerzo del alumno.**
Estimación de las **horas** dedicadas a cada tipo de actividad para cada contenido
5. **Calendario**
Actividades y esfuerzo previstos en cada momento
6. **Sistema de evaluación continua**
Basado en **puntos acumulativos.**
Lista de **indicadores** (maneras de conseguir puntos)

- 
- Aplicar a la enseñanza/aprendizaje de Ingeniería del Software algunos principios y buenas prácticas de la ingeniería:
 - Aprendizaje basado en problemas (ABP).
 - Problema => proyecto de desarrollo de software
 - Trabajo en equipo.
 - Organizar equipos de 4-6 alumnos y asignar roles similares a los reales (jefe de proyecto, etc.).
 - Seguimiento y control del trabajo.
 - Evaluación continua con entregables pequeños en vez de uno grande al final.
 - Utilizar fuentes internacionales.
 - SWEBOK como referencia de contenidos.
 - Cursos seleccionados de los currículos de ACM.

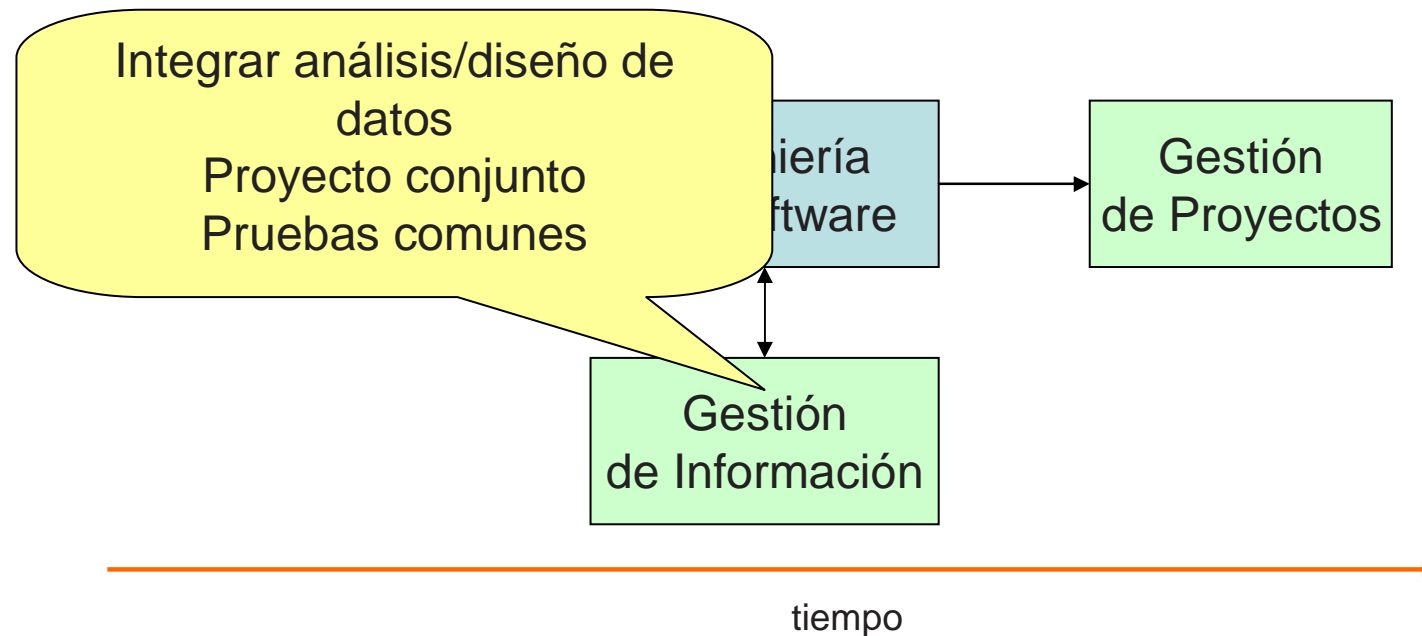
- Pensar en términos de grupos de asignaturas
 - Ideando prácticas conjuntas.
 - Utilizando mismas herramientas.
 - Estableciendo pruebas horizontales conjuntas.
- Asignaturas con las que se debe realizar un análisis conjunto:



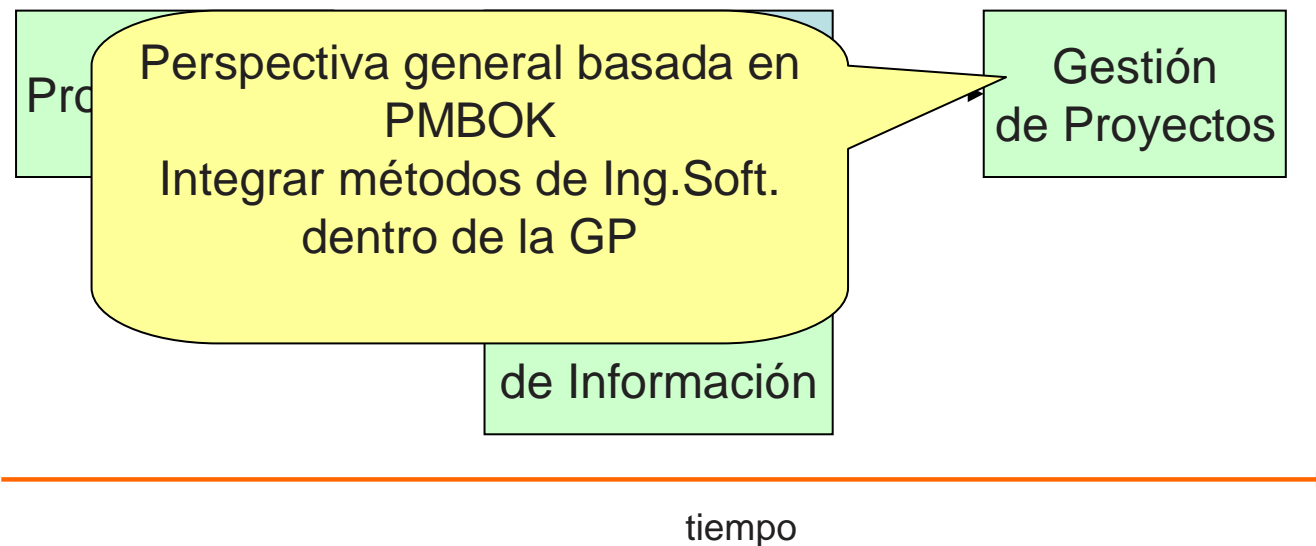
- Pensar en términos de grupos de asignaturas
 - Ideando prácticas conjuntas.
 - Utilizando mismas herramientas.
 - Estableciendo pruebas horizontales conjuntas.
- Asignaturas con las que se debe realizar un análisis conjunto:



- Pensar en términos de grupos de asignaturas
 - Ideando prácticas conjuntas.
 - Utilizando mismas herramientas.
 - Estableciendo pruebas horizontales conjuntas.
- Asignaturas con las que se debe realizar un análisis conjunto:

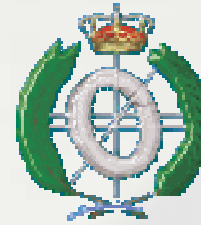


- Pensar en términos de grupos de asignaturas
 - Ideando prácticas conjuntas.
 - Utilizando mismas herramientas.
 - Estableciendo pruebas horizontales conjuntas.
- Asignaturas con las que se debe realizar un análisis conjunto:





- La Ingeniería del Software es una **parte clave** de lo que llamamos Informática.
- Es vital para el futuro (profesional, laboral y académico) de la Informática **incidir más en la perspectiva de ingeniería**.
 - Más arquitecto, menos albañil.
- En la reforma para implantar el EEES:
 - Debe ser una **materia independiente pero coordinada con otras**: Gestión de Información, Gestión de Proyectos, Programación.
 - Es bueno emplear **nuevos métodos** (trabajo en equipo, aprendizaje por proyectos, ..).



Francisco Ruiz

<http://alarcos.inf-cr.uclm.es/per/fruiz/>
francisco.ruizg@uclm.es

Zaragoza, septiembre 2007