

¿Podemos darle la vuelta a la enseñanza del desarrollo del software?

Josep M. Marco-Simó, Isabel Guitart Hormigo, M.
Jesús Marco-Galindo, Àngels Rius Gavidia, M.Elena
Rodríguez González

Estudios de Informática, Multimedia y Telecomunicación
Universitat Oberta de Catalunya

{jmarco, iguitarth, mmarcog, mriusg, mrodriguezgo}@uoc.edu



JENUI 2007

Contenido

- La motivación
- El punto de partida
- El proceso de reflexión
- Los resultados obtenidos
- Las conclusiones

Motivación

- ¿Por qué enseñamos el DS de lo particular (P) a lo general (IS) si...
- ... el proceso de DS es precisamente el contrario...
 - ... y los ciclos de vida comúnmente aceptados también lo son?

Posiblemente:

- Por motivos históricos
- Por falta de valentía respecto a reconsiderarlo (a diferencia de áreas como las de redes de computadores), no fuera caso que nos diéramos cuenta que:
 - ... sobra algo de lo que enseñamos
 - ... repetimos contenidos
 - ... no insistimos en lo que realmente es importante hoy
 - ... los itinerarios que planteamos son obsoletos

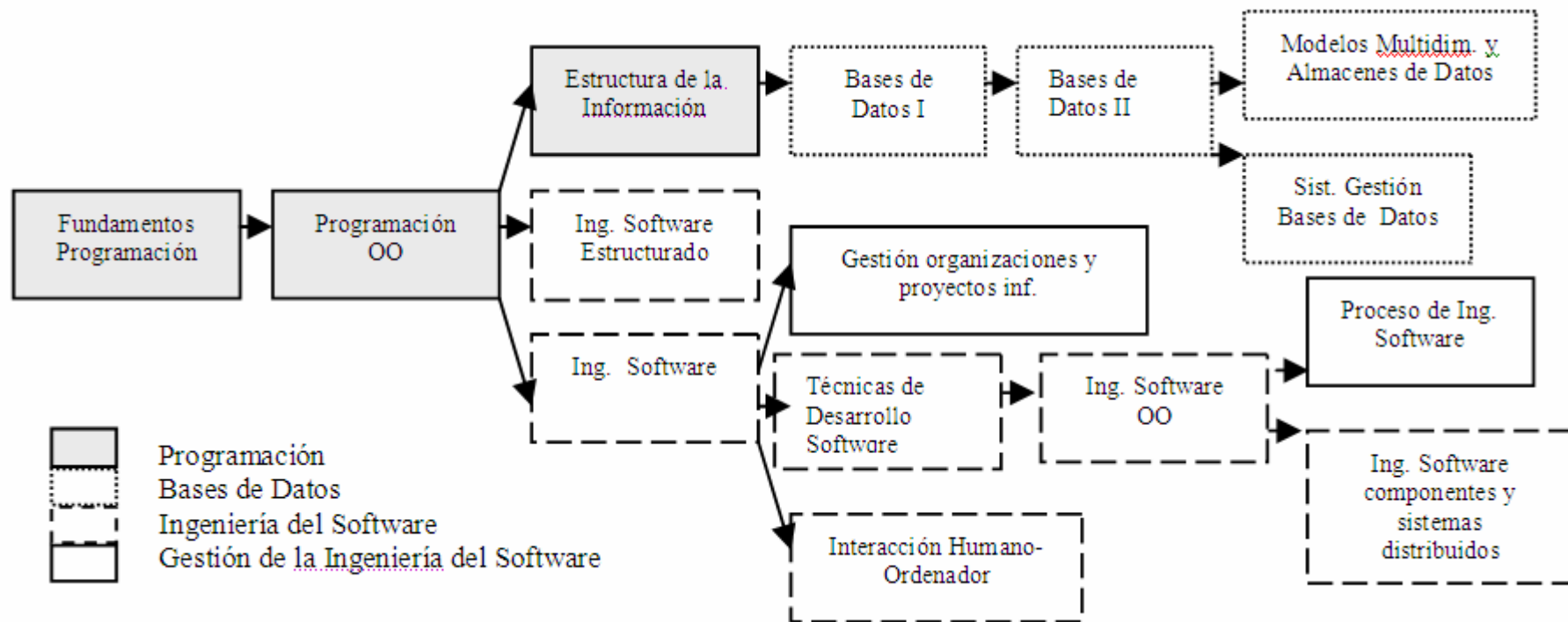
Motivación

Nuestro objetivo genérico final: Intentar plantear la docencia de lo general a lo particular

¿**Cómo?** Con un equipo interdisciplinar del área genérica del DS:

Ámbitos del Desarrollo del Software		Profesores
P	Programación	2
BD	Bases de Datos	2
IS	Ingeniería del software	3
G	Gestión de IS	2

Punto de partida



Punto de partida

Tras años de aplicar esta organización hemos aprendido que...:

- ... el estudiante de DS necesita **habilidades genéricas** (aplicadas a problemas de diferente magnitud) comunes tanto en P y IS como son: abstracción, análisis, y aplicación de métodos a casos concretos.
- ... tanto en P como en IS usamos los **mismos paradigmas**: estructurado, orientado a objetos, distribuido, aunque no siempre mediante esfuerzos alineados.
- ... al estudiante de P en el primer curso le pedimos que entienda aspectos conceptuales pero también que sepa aplicarlos prácticamente. Y no lo logramos: **los rendimientos son bajos...**

Luego: **¿intentamos darle la vuelta a la organización curricular para empezar con esta parte común, posponiendo la parte práctica, y acercándonos al modelo habitual real del DS?**

El proceso de reflexión

Fase 1: Buscando otras experiencias

- No las encontramos en nuestras universidades más cercanas
- En el contexto internacional sólo el currículo de la ACM sobre IS propone empezar por IS: sin embargo, estudiado en detalle, se ve que considera P como la primera etapa de la IS
- Sí que encontramos alguna experiencia expuesta en otras ediciones de las JENUÍ (aunque desde diferente perspectiva y no tan globalmente):

Cachero, C.; López, O. & Durá, M.J. *Del Diseño a la Implementación del Software: Una metodología de cohesión* (2004)

El proceso de reflexión

Fase 2: Más allá (o crítica pura) de la visión clásica

Para superar los tópicos preconcebidos y intentar un análisis crítico y sin tabúes de nuestros contenidos actuales, **nos obligamos a mantener en mente dos cuestiones básicas:**

- ¿Es necesario retrasar la **codificación**? ¿Es ventajoso o contraproducente “codificar-compile-probar” desde el primer día? ¿Entenderían los estudiantes no hacerlo así?
- ¿Es posible empezar directamente **introduciendo en genérico el análisis de problemas y el diseño de soluciones** sin conocer ninguna de las palabras nucleares clave de P e IS (como variable, parámetro, tipo, procedimiento, compilar, depurar, entorno, descomponer, llamar, abstraer...)?

Y empezamos una **revisión plenaria de contenidos**, de la que sacamos....:

El proceso de reflexión

Fase 2: Más allá (o crítica pura) de la visión clásica

- El concepto de ciclo de vida no aparece hasta la primera asignatura IS
- UML se estudia en diferentes momentos, pero sin visión unitaria
- Aparecen desconectadamente conceptos sobre eficiencia y calidad en los 4 ámbitos
- Se explica modelado de datos tanto en BD como en IS (¡pero con diferentes métodos!)
- La desconexión entre P y IS
- Poco tiempo para la asimilación y maduración de P en el primer curso.
- No se incluye diseño y programación distribuida hasta las últimas asignaturas de IS
- La interacción persona-ordenador se considera colateral (no está integrada en todo el ciclo de vida)
- Se necesitan asignaturas de síntesis donde experimentar todo el ciclo de vida del DS

El proceso de reflexión

Fase 3: Planteando los cambios

Los profesores de cada ámbito plantearon sus cambios, confrontadamente con los profesores de los otros ámbitos. Y las novedades aparecidas fueron:

Para IS:

- Un bloque introductorio de IS como un proceso centrado en la calidad y el usuario (requisitos y interacción hombre-máquina)

Para BD:

- Creación de BD a partir de UML también
- Datos semiestructurados y XML

El proceso de reflexión

Fase 3: Planteando los cambios

Para P:

- Diferenciar *tecnología de la programación* de *algorítmica*.
- La tecnología de la programación debe prestar importancia a la calidad, las pruebas, la depuración y la eficiencia.
- Programar a partir de diseños UML.

Para G:

- Plantear la calidad del proceso como un requisito de todo el ciclo de vida de la creación y la gestión del software.
- Diferenciar la *gestión del proceso de desarrollo del software* (mantenimiento, métricas, gestión de la configuración) de la *dirección de sistemas de información* (estrategia organizativa) e incluso de la *gestión de proyectos* (de TIC en general).

El proceso de reflexión

Fase 4: Dibujando un nuevo mapa

A partir del debate de F3 y del intento de organizar conjuntamente los cambios propuestos, tenemos algunas respuestas a las preguntas F2:

¿Posponer la codificación?

- Aunque parece factible, ***no puede retrasarse mucho por condicionantes curriculares***: se necesita en otros ámbitos (redes, arquitectura, sistemas operativos...) e ***implica renunciar a uno de los aspectos prácticos más estimulantes para muchos estudiantes.***

¿Empezar por el análisis de problemas y el diseño de soluciones?

- Sí que se puede y se debe dar una visión global de todo el proceso de DS, creando un ***marco de referencia*** que sirva de hoja de ruta durante la introducción de las siguientes competencias/contenidos/objetivos.
- A partir de aquí, los ***condicionantes curriculares*** vuelven a impedir ser mucho más atrevidos. ***Quizá no podemos invertir el orden, pero sí simultanear la introducción de los dos ámbitos.***

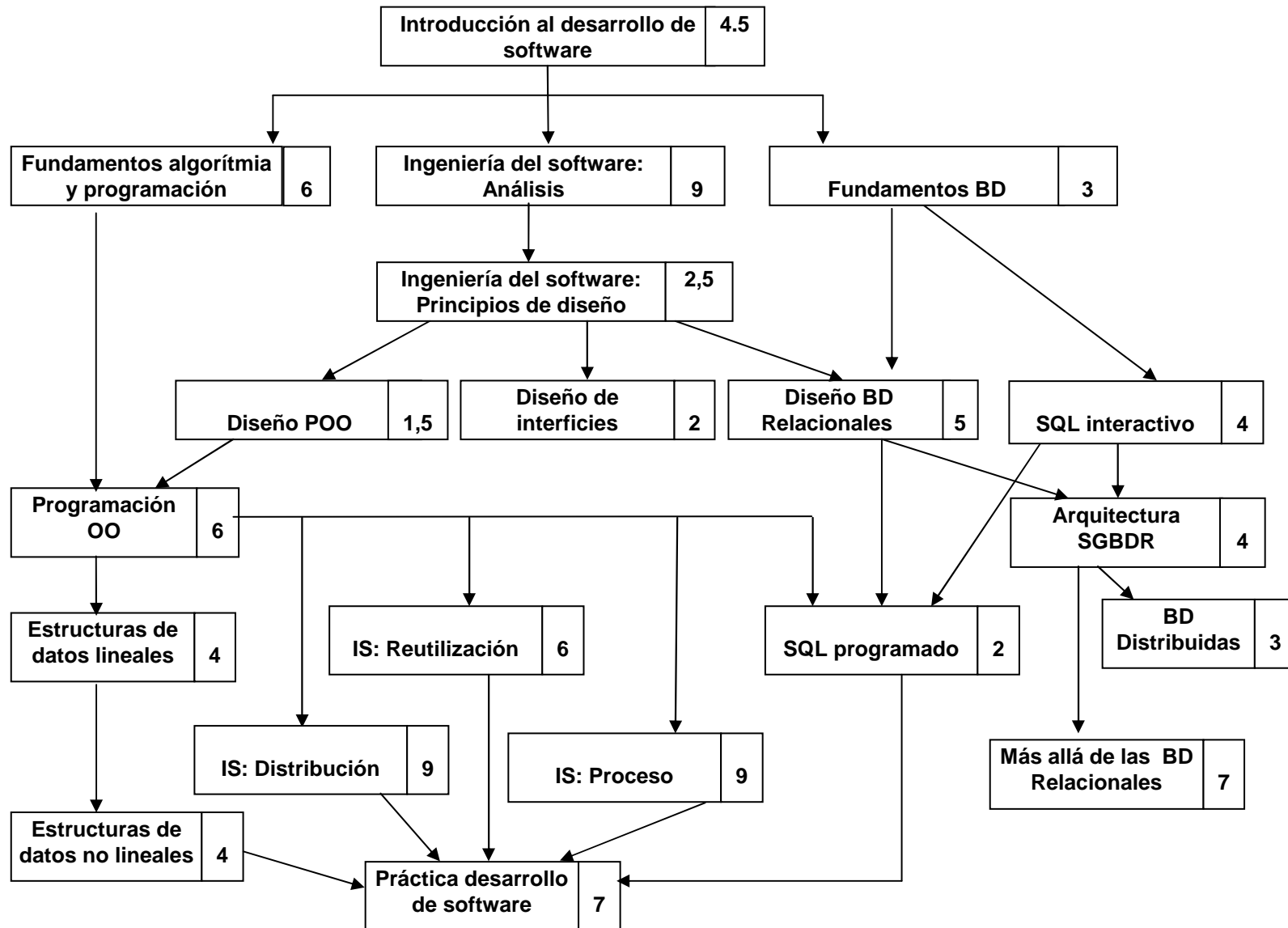
El proceso de reflexión

Fase 4: Dibujando un nuevo mapa

Para dibujar el nuevo mapa aplicamos las siguientes **simplificaciones**:

- **Centrado en el currículo de *grado de informática*** (no pudimos abstraernos de la complejidad de abordar toda el área con independencia de en qué oferta formativa estuviera incorporada)
- Todavía **centrado en contenidos y no en competencias**: en el momento de proponer el mapa los profesores implicados tenían diferente dominio de la organización por competencias.
- Por la misma razón, todavía calculado en créditos **no ECTS** (pre-EEES)
- Dibujar no “materias” ni asignaturas sino “**unidades de aprendizaje mínimas**” que facilitaran su ensamblaje posterior y definidas como “el conjunto máximo de contenidos mínimos (indivisibles) desde el punto de vista docente y para un período lectivo”,

El resultado



Las conclusiones obtenidas

En cuanto a la organización de contenidos:

Motivos curriculares ajenos a la propia materia desaconsejan la inversión drástica del orden. (Pero ¿qué pasaría en una oferta formativa donde no existieran estos condicionantes, es decir, centrada sólo en el desarrollo del software?)

La necesidad de una visión global del proceso descendente del DS puede cubrirse mediante un **contenido inicial marco** que sirva de referencia al estudiante.

Aunque no se invierta el orden sí que **puede simultanearse la introducción de P y de IS**, insistiendo en los conceptos comunes y fundamentales, minimizando las repeticiones de contenido y **manteniendo como un continuo aspectos como la calidad y el diseño centrado en el usuario.**

Hay que seguir intentando eliminar los **temas que no son imprescindibles** para la competencia profesional que se busca en cada nivel formativo.

Las conclusiones obtenidas

En cuanto al proceso:

La ausencia de experiencias similares nos ha obligado a superar las **reticencias al cambio** en la acción docente.

El conocimiento de las dificultades e intereses del estudiante no nos ayuda a ser más atrevidos.

Otras materias deben contribuir a aportar las bases para la enseñanza del DS (Lógica, Matemáticas) No es una tarea fácil.

En cuanto al futuro:

Estamos pasando ya del cuadro de contenidos al de competencias y a créditos ECTS a partir de las herramientas y directrices que nuestra universidad está desarrollando en espera de los marcos legales de referencia EEES definitivos.

¿Podemos darle la vuelta a la enseñanza del desarrollo del software?

Josep M. Marco-Simó, Isabel Guitart Hormigo, M.
Jesús Marco-Galindo, Àngels Rius Gavidia, M.Elena
Rodríguez González

Estudios de Informática, Multimedia y Telecomunicación
Universitat Oberta de Catalunya

{jmarco, iguitarth, mmarcog, mriusg, mrodriguezgo}@uoc.edu



JENUI 2007