 ISBN: 978-99920-70-10-9

Casi le dimos la vuelta a la enseñanza del desarrollo del software

Josep Maria Marco-Simó Maria Jesús Marco-Galindo Daniel Riera Terrén
Estudis d'Informàtica, Multimedia i Telecomunicació
Universitat Oberta de Catalunya
Barcelona
{jmarco, mmarcog, drierat}@uoc.edu

Resumen

En las XIII JENUI de 2007 presentamos la ponencia "¿Podemos darle la vuelta a la enseñanza del desarrollo del software?" en la que proponíamos una reorganización curricular descendente del ámbito del Desarrollo del Software, a partir del trabajo colectivo de los profesores del área y una vez analizados los motivos por los que el planteamiento habitual era el ascendente (de la programación al proceso de la ingeniería del software). Desde entonces, muchos de los profesores implicados en ese trabajo hemos contribuido al diseño e implantación de nuestro nuevo Grado en Informática, lo que nos ha permitido ver si aquella propuesta (planteada libre e independientemente de un contexto) se ha podido adaptar al (finalmente rígido) marco de este Grado en España. Las conclusiones que presentamos en cuanto al diseño demuestran que, como era de esperar, nos hemos dejado cosas en el camino de la adaptación pero que el resultado final encaja suficientemente con nuestra propuesta de 2007. Por otro lado, dado que la implantación de la propuesta se inició en 2010, hoy ya tenemos suficiente información histórica que nos permite también, en esta contribución, un primer análisis de su implementación real y de su impacto en el aprendizaje de los estudiantes y en la docencia del profesorado.

Abstract

In JENUI XIII (2007) we presented the paper "¿Podemos darle la vuelta a la enseñanza del desarrollo del software?)" proposing a top-down curricular reorganization of the software development area. The proposal was the result of the collective work of the faculty of the area, and the analysis of the reasons why the traditional approach was bottom-up (from programming to software engineering). Since then, the faculty involved has contributed to the design of the new Bachelor in Computer Science. This allowed us to see if the proposal (developed freely and independently of the context) could be adapted to the (finally rigid) legal framework for this Bachelor in Spain. The conclusions presented show that, as expected, we have lost some things on the way, but that the final result fits enough with our 2007 proposal. Furthermore, given that the implementation of the proposal began in 2010, today we have enough historical information to also present in this paper an initial analysis of its current implantation and impact on student learning and faculty teaching.

Palabras clave

Organización curricular, Desarrollo del software, Ingeniería del software, Programación.

1. Motivación

En el contexto de análisis de los planes de estudio de Informática previos a la implementación de los nuevos Grados adaptados al EEES, y conocedores que en otras áreas de conocimiento [1] ya se había implementado una reorganización similar, el grupo de profesores implicados en los ámbitos del Desarrollo del Software (DS) (básicamente las áreas de Programación –P–, e Ingeniería del Software –IS– que incluye Bases de Datos –BD– y Gestión del Proceso de Desarrollo del Software –G–) de nuestra universidad, iniciamos un ejercicio de reflexión sobre si era posible organizar la docencia de DS desde una perspectiva descendente, en lugar de la clásica ascendente.

Nuestra propuesta tenía puntos de conexión con las aportaciones de Meyer, quien ya en 1993 [2] asumía para la enseñanza del DS la filosofía del *inverted curricula* o *progressive opening of black boxes* propuesto por Cohen para el ámbito de la ingeniería eléctrica. En su caso no se trataba tanto de una visión *top-down* como una visión *outside-in*, esto es, basada en introducir al estudiante primero como consumidor-usuario de software para ir avanzando hacia un rol de productor-desarrollador. Sus propuestas se basaban en universalizar el uso exclusivo y sin ambigüedades de la orientación a objetos, con el diseño por contrato como estrategia y en todo caso apoyado en un potente entorno de librerías de trabajo. Sin embargo parece

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

que dichas propuestas han quedado principalmente limitadas a los cursos iniciales de programación [3] a pesar que posteriormente el mismo autor ya ha expuesto unas interesantes líneas maestras para un diseño curricular completo [4]. Independientemente de su alcance final real, las razones que alega Meyer sobre la necesidad de replantear profundamente la organización curricular de la enseñanza del DS nos resultan, como hemos dicho, también muy cercanas.

El resultado de nuestro trabajo de diseño del Grado en Ingeniería Informática en nuestra universidad, fue publicado en las XIII JENUI de 2007 [5] y fue posteriormente ampliado y revisado para Novática [6]. En él sintetizamos un conjunto de críticas a la organización curricular tradicional (en nuestra universidad, pero también en nuestro entorno cercano) y planteamos nuestras principales conclusiones, que resumimos a continuación:

C1. Ofrecer al inicio de los estudios la visión completa del DS. Tradicionalmente los estudiantes no obtienen una visión completa del DS hasta avanzados los estudios. Luego, sería necesario dar una visión inicial global y descendente del proceso del DS. Esto se podría cubrir mediante la creación de un contenido inicial marco que sirviera de referencia al estudiante durante el resto de sus estudios.

C2. Introducir en paralelo la IS y la P. Diferentes motivos desaconsejaban la inversión drástica del orden de docencia (empezar por IS y seguir por P). Entre estos motivos estaban la dependencia curricular con otras áreas de conocimiento, las expectativas de formar rápidamente a programadores, las percepciones sobre las capacidades de los estudiantes, y las propias dudas de los equipos docentes sobre la viabilidad del replanteamiento. Sin embargo, sí que sería posible realizar en paralelo la introducción tanto de IS como de P. Esto permitiría reforzar e insistir en los aspectos que son comunes entre IS y P, así como minimizar las repeticiones de contenido.

C3. Optimizar los contenidos ofrecidos. Era necesario reducir el tiempo dedicado a los temas no demasiado relevantes para el profesional actual y racionalizar el conjunto total de las explicaciones (evitando redundancias e introduciendo los aspectos en un orden más descendente).

C4. Contribución de otras asignaturas al aprendizaje del DS. Otras asignaturas, fuera del ámbito de DS, podrían contribuir a aportar las bases para la enseñanza del DS. Esta discusión, implicaría reconsiderar contenidos de otras áreas de conocimiento y requeriría una estrategia compleja para ser operativa.

Adicionalmente, en aquel trabajo reflexionamos sobre las razones (o excusas) usadas para justificar el inmovilismo en la innovación curricular en este ámbito y para criticar la viabilidad de la propuesta:

E1. El Grado debe contentar a otros ámbitos de conocimiento, no sólo al DS. Cierto, pero ¿cuántas de las limitaciones curriculares autoimpuestas (por dependencia con otros ámbitos docentes) quedarían superadas en un Grado muy centrado en IS, como el propuesto, por ejemplo, por Martín y Ruiz [7]?

E2. La falta de interés y madurez de los estudiantes. ¿Cuántas de las ideas sobre lo que quiere el estudiante –aprender a programar– o es capaz de hacer –aplicar abstracciones y capacidad de análisis– no son más que tópicos? Si en otros ámbitos más clásicos se ha logrado que los estudiantes aprendan conceptos mucho más complejos o abstractos y los apliquen, ¿por qué no es posible hacerlo en el normalmente estimulante (para un estudiante de nueva incorporación) ámbito del DS?

E3. La poca predisposición al cambio del profesorado. ¿Qué esfuerzo de coordinación y sacrificio puede (o quiere o está autorizado a) realizar el profesorado universitario para asegurar la coherencia docente de este extenso ámbito?

Tras ese trabajo, realizado como un ejercicio intelectual todavía libre de las restricciones finales fijadas por los marcos de referencia españoles [8], llegó el momento de diseñar nuestro nuevo Grado de Ingeniería Informática. Prácticamente todos los profesores implicados en nuestra reflexión de 2007 participamos en el diseño de las nuevas asignaturas formando parte, bien de la comisión de Grado encargada de liderar la propuesta, bien de las diferentes subcomisiones por ámbito de conocimiento. Y en ese diseño del Grado no olvidamos la reflexión de 2007.

En esta contribución presentamos (en la sección 2) el resultado de este diseño e intentamos detallar cómo ha respondido a las conclusiones y excusas anteriores. También presentamos (en la sección 3) un primer análisis de la forma final que ha tomado la implementación, así como de los resultados académicos obtenidos en los primeros años de implantación, tanto entre los estudiantes como entre el profesorado.

2. El diseño resultante versus la propuesta

En la figura 1 presentamos la propuesta de 2007 (el mapa de lo que llamábamos unidades mínimas de contenido y que corresponden a los rectángulos blancos) contrastada con el diseño de 2009 del Grado en Informática (las asignaturas resultantes del diseño y que corresponden a las cajas sombreadas). Las asignaturas resultantes se presentan como contenedoras de las unidades mínimas de contenido. Así la intención es representar cómo el diseño del Grado cubre la propuesta de 2007 en cuanto a contenidos.

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

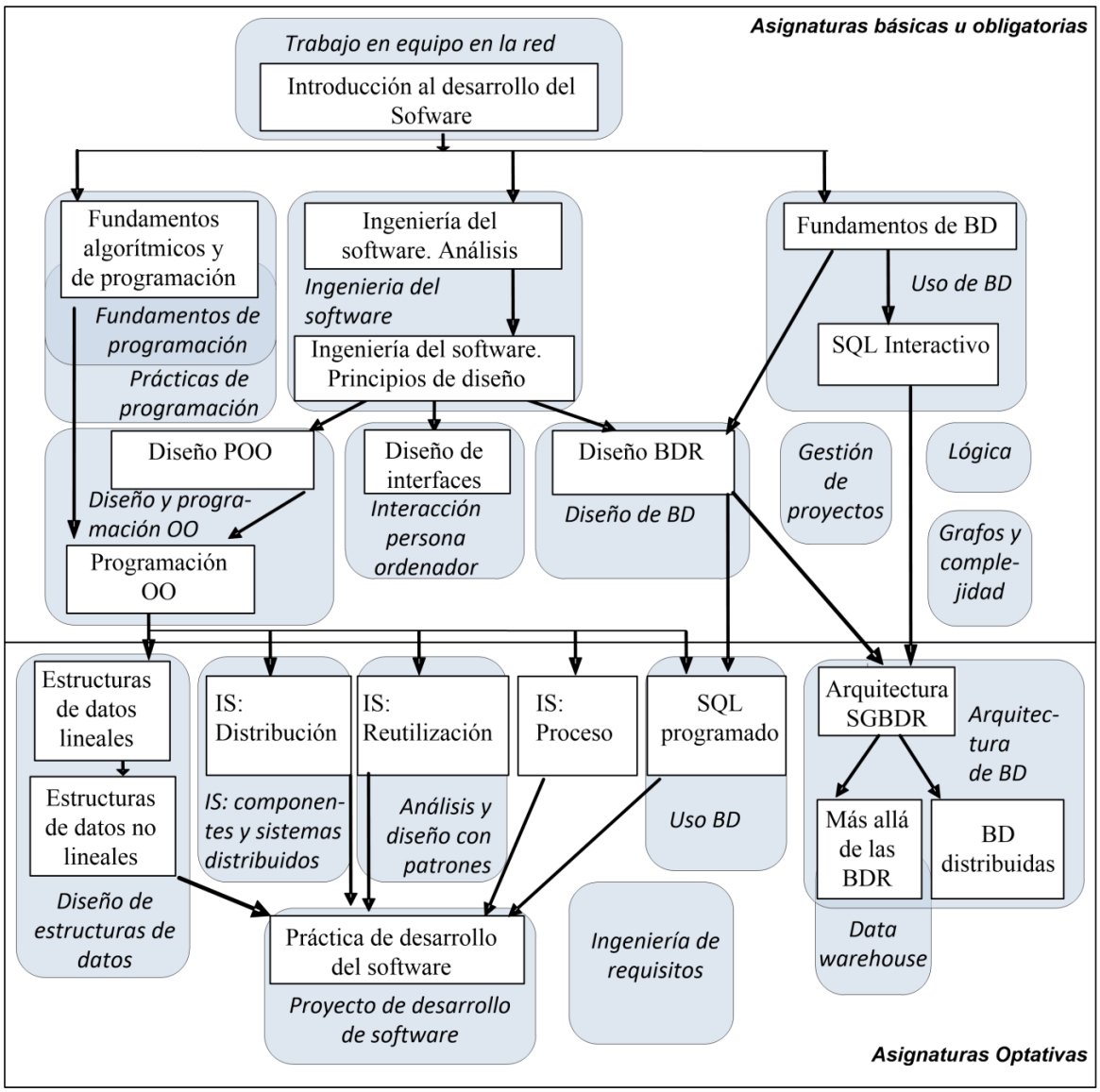

Figura 1: Nuestra propuesta de 2007 (en blanco) versus nuestro diseño de 2009 (en sombreado).

2.1. Respecto a las conclusiones

¿Cómo responde el diseño a las diferentes conclusiones del trabajo de 2007? Veámoslo:

Al respecto de C1 (Ofrecer al inicio de los estudios la visión completa del DS), la unidad mínima de contenido *Introducción al desarrollo del software* era la que pretendía ser el marco de referencia y el contenido introductorio donde se presentaba la visión global del DS. En el diseño final, este contenido queda incluido en la asignatura *Trabajo en equipo en la red*, que es obligatoria de primer semestre. Esta asignatura se basa en un libro [9] que presenta los diferentes ámbitos de la informática y que contiene un capítulo dedicado a la IS, pero también otros dedicados a P, BD y a Sistemas de Información (SI, que, entre otras cosas, recoge la parte de G). El hecho que se trate de un libro convencional, entendemos que le confiere a este contenido el rango de marco de referencia durante toda la carrera (no son unos apuntes que pueden percibirse como temporales, sino un libro que puede merecerse un hueco más o menos permanente en la biblioteca). Con todo esto, pues, intentamos asegurar que el estudiante tuviera una visión global del DS desde el inicio de la carrera.

Al respecto de **C2 (Introducir en paralelo la IS y la P)**, la unidad mínima de contenido *Fundamentos algorítmicos y de programación* que introduce P, se separa en el diseño en dos asignaturas: *Fundamentos de programación*, asignatura que es una introducción a la algorítmica y que dedica unos ejercicios mínimos a un primer contacto con un lenguaje de programación; y *Prácticas de progra-*

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

mación, que aborda las competencias sobre las técnicas de programación (el uso de un entorno de programación y la puesta en práctica del ciclo codificar-compilar-probar en un caso elemental). Esta organización en dos asignaturas (12 ECTS en total) permite afrontar una de las críticas de la organización curricular tradicional: la falta de tiempo para abordar el aprendizaje de P y para interiorizar el mecanismo algorítmico y el de la programación. Adicionalmente a todo esto, el mapa resultante permite comprobar cómo quedan ubicadas en paralelo las primeras asignaturas de P y de IS. Esta paralelización ha de facilitar también la comprensión de la conexión natural entre ambos ámbitos, otra de las críticas recogidas de las organizaciones curriculares tradicionales.

Al respecto de **C3 (Optimización de los contenidos ofrecidos)**, el diseño reduce a la mínima expresión temas menos actuales en la praxis profesional, desde nuestro punto de vista, como son a) la perspectiva estructurada de la IS en beneficio de la OO y b) los aspectos de BD pre-relacionales. El diseño también recoge la eliminación de las redundancias indicadas en la propuesta de 2007: a) el diseño de datos en IS y en BD; y b) los principios de la orientación a objetos y de la notación UML en P y en IS. Ahora estos dos aspectos se explicarán una única vez y desde una perspectiva general, para aplicarse luego donde corresponda (P o IS).

Aun al respecto de C3, el diseño resultante presenta una ordenación más similar a la lógica real del desarrollo del software. En ese sentido, en paralelo a la introducción de IS y de P, también se introducen las asignaturas de contenido transversal –no sólo para DS- *Ingeniería de Requisitos* y *Gestión de Proyectos* (dos contenidos no considerados directamente en nuestra propuesta de 2007). Además, una vez presentada la IS se abordan ya los aspectos de Diseño de interfaces en la asignatura *Interacción persona ordenador* (IPO), que queda de esta manera más integrada en el camino, cosa que permite responder a otra de las críticas de la organización curricular tradicional. Los aspectos de IPO, sin embargo, se abordan transversalmente –el ámbito de la IPO no es sólo el del DS sino también, por ejemplo, el del diseño de dispositivos–, cosa que ha dificultado su integración plena en el conjunto.

Finalmente, al respecto de **C4 (Contribución de otras asignaturas al aprendizaje del DS)**, debemos admitir que no hemos avanzado mucho. Para el diseño del Grado los profesores de los diferentes ámbitos de conocimiento se organizaron en subcomisiones para realizar sus contribuciones. Esto dificultó la interacción entre nuestro ámbito y el que incluye los contenidos de *Lógica*, a pesar de la existencia de la comisión general de Grado. Sea como sea, la organización curricular resultante, ha situado, por inercia con la organización tradicional, los contenidos de lógica en el mismo semestre que *Practicas de Programación* e *Ingeniería del software*, cuando hubiera sido preferible que las precediera. Posteriormente a *Lógica* aparece otra de las asignaturas conceptuales de soporte a la IS como es *Grafos y complejidad* que puede coincidir en el tiempo con *Diseño y programación orientada a objetos*.

2.2. Respecto a las excusas

¿Y cómo responde el diseño a las diferentes razones o excusas que detectamos en nuestro trabajo de 2007? Veámoslo:

Al respecto de **E1 (El Grado debe contentar a otros ámbitos de conocimiento, no sólo al DS)**, es cierto que el Grado es generalista y no está orientado sólo a la IS. En ese sentido las dos asignaturas iniciales de P (*Fundamentos de Programación* y *Prácticas de Programación*) se han diseñado con el objetivo, difícil y no siempre alcanzado como se verá más adelante, de sentar las bases mínimas necesarias para aquellos estudiantes que se orientan a otros ámbitos donde P es también básica (sistemas operativos, sistemas distribuidos o redes de computadores). Pero por otro lado, también es cierto que el Grado presenta la posibilidad de cursar un itinerario de IS. Esto ha facilitado mucho el encaje de nuestra propuesta. Las asignaturas que presentamos no son todas obligatorias, evidentemente, pero sí que pueden ser abordadas completamente por los estudiantes interesados en este ámbito. En definitiva, el diseño permite, sobre el papel, dar un nivel inicial a estudiantes de otros ámbitos, y suficiente a los estudiantes del ámbito de DS.

Al respecto de **E2 (La falta de interés y madurez de los estudiantes)**, la discusión entre posponer los aspectos de prácticas de programación y entrar más directamente en IS, siempre ha estado latente. ¿Están los estudiantes preparados para no programar (entendido como codificar, compilar, ejecutar-probar) desde el primer día? ¿Son maduros para introducir rápidamente conceptos de análisis de la IS? La necesidad de otros ámbitos de conocimiento, como ya dijimos en la propuesta de 2007, también ha empujado no tanto a invertir drásticamente el orden entre P e IS sino a su paralelización, más todavía en nuestro caso en el que las asignaturas iniciales de P y de IS no sólo son obligatorias en el Grado de Informática sino también en el Grado en Multimedia y el Grado de Tecnologías de la Telecomunicación. Si no hubiera existido este condicionante, que forzaba a introducir en los cursos iniciales estas asignaturas, la discusión sólo pensando en responder las necesidades del Grado de Informática, hubiera sido larga.

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

Finalmente, al respecto de **E3 (La poca predisposición al cambio del profesorado)**, en nuestro caso también han emergido algunas tensiones poco relacionadas con temas docentes: a) la necesidad de asegurar docencia de cierto tipo (obligatoriedad versus optatividad) en cada ámbito de conocimiento; y b) la resistencia a aplicar cambios de enfoque o a ceder contenidos a otras asignaturas. Sea como sea, en todos los casos, como en el momento de la propuesta de 2007, estas discusiones se han podido acabar cerrando con la decisión de la comisión de Grado en 2009.

Para finalizar, indicar que, como se refleja en el bloque "*IS: Proceso*" de la figura, la unidad mínima de contenido sobre proceso de la IS, no ha quedado cubierta en el Grado.

3. Implementación y resultados

Escapa de las posibilidades y del interés de esta contribución presentar y analizar el detalle de la implementación final de cada asignatura, es decir, de su forma real y de si se ha adaptado en cada caso a lo previsto. En lugar de eso, en esta sección, a partir de los resultados obtenidos en los primeros cursos de implantación del nuevo Grado, sí que iremos introduciendo detalles de implementación real de las asignaturas en aquellos casos en que se hayan desviado significativamente de lo previsto en el diseño. En la sección final, estas diferencias significativas serán debidamente recogidas.

3.1. Rendimiento y satisfacción

En el cuadro 1 presentamos los resultados promedio en cuanto a rendimiento y satisfacción de las diferentes asignaturas, a partir de los datos disponibles desde febrero de 2010 (fecha de la puesta en marcha del Grado) hasta julio de 2014. Entendemos como rendimiento el porcentaje de aprobados sobre matriculados; y como satisfacción el porcentaje de respuestas "muy satisfecho" y "satisfecho" (en una escala de cinco categorías que también incluye "indiferente", "poco satisfecho" e "insatisfecho") sobre el total de respuestas obtenidas en las encuestas de valoración del curso que se pasan a los estudiantes al final del mismo. A continuación, destacamos los datos que consideramos más relevantes de estos resultados:

Las asignaturas introductorias a P (*Fundamentos de Programación* y *Prácticas de Programación*) tienen el rendimiento más bajo en términos relativos, aunque están en la línea de lo que venía pasando habitualmente en el entorno previo al EEES: se trata de las primeras asignaturas que matriculan los estudiantes y reciben el impacto de los abandonos de los estudiantes de nueva incorporación. La satisfacción está cerca del 75% en la primera asignatura pero cae hasta el 53.6% en la segunda. Es este último indicador el que no es satisfactorio y responde a las dificultades expresadas por el profesorado para diseñar ejercicios que aborden los temas más complejos de la asignatura (recursividad, complejidad y eficiencia algorítmica) al nivel de madurez de los estudiantes. De hecho estos contenidos, no estaban previstos en la propuesta de 2007, ni en el diseño de 2009, pero debieron incorporarse al ser estas dos las únicas asignaturas obligatorias del ámbito de P.

Por otro lado la siguiente asignatura del ámbito de P, *Diseño y Programación Orientada a Objetos*, tiene unos valores muy aceptables de rendimiento y satisfacción (67.6% y 80.9%). El rendimiento se puede justificar por tratarse de estudiantes que están ya en semestres más avanzados de su carrera; pero la satisfacción puede interpretarse como que la exigencia de la asignatura es compatible con la formación recibida en las asignaturas previas de P y de IS, cosa que no se había logrado en los planes de estudio anteriores.

Sin embargo, la siguiente asignatura del ámbito de P, *Diseño de Estructuras de Datos*, al quedarse en un 60.2 de rendimiento, apunta (con todas las reservas de tener sólo tres semestres de historia) que la base previa de los estudiantes no es suficiente para afrontar los problemas que implican el diseño de estructuras de datos y el desarrollo de la algorítmica y programación asociada a los mismos. Esta asignatura tiene un salto muy importante de nivel y de enfoque y no acaba de encajar con las asignaturas previas (*Diseño y Programación Orientada a Objetos* y *Prácticas de Programación*).

En cuanto al ámbito de IS, la asignatura *Ingeniería del Software* que, como decíamos, se propone como asignatura que puede cursarse en un primer semestre en paralelo a las de P, los resultados tanto de satisfacción como de rendimiento son claramente aceptables (61.3% y 78.9%, respectivamente). Así pues, adelantarla en la estructura del Grado no ha resultado contraproducente.

En el mismo ámbito de IS, las asignaturas *Ingeniería del Software de componentes y sistemas distribuidos* y *Análisis y diseño con patrones*, apuntan, en sus primeros tres y cuatro semestres respectivamente, buenos resultados de rendimiento y sobre todo de satisfacción, pudiéndose entender que los estudiantes se sienten suficientemente cómodos y competentes con sus conocimientos preliminares sobre IS al llegar a estas asignaturas.

Al igual que el ámbito de IS, en el correspondiente a BD, la asignatura *Uso de Bases de Datos*, que introduce las bases de datos también en paralelo con P y con IS, obtiene a su vez buenos valores. Esta asignatura consigue un rendimiento de cerca

ISBN: 978-99920-70-10-9

Asignatura	Núm. Semestres	Rendimiento promedio %	Satisfacción promedia %
Trabajo en equipo en la red	9	73.4	75.9
Fundamentos de programación	9	36.7	74.4
Prácticas de programación	8	48.4	53.6
Ingeniería del software	7	63.1	75.4
Diseño y programación OO	8	67.6	80.9
Interacción persona ordenador	8	74.3	70.0
Uso de BD	5	59.4	86.0
Diseño de BD	4	64.4	81.4
Arquitectura de BD	2	83.7	85.7
Data warehouse	0	No aplica	No aplica
Diseño de estructuras de datos	3	60.2	74.0
IS de componentes y sistemas distribuidos	3	67.1	86.3
Análisis y diseño con patrones	4	74.6	78.7
Proyecto de desarrollo de software	3	86.1	No disponible
Ingeniería de requisitos	3	88.7	89.4
Gestión de proyectos	8	69.0	67.2
Lógica	8	48.4	76.0
Grafos y complejidad	6	59.9	67.5

Cuadro 1: Resumen de los resultados promedio de rendimiento y satisfacción.

del 60% y una satisfacción del 86%. Si añadimos a esto que las siguientes asignaturas del ámbito apuntan tendencias similares (*Diseño de Bases de Datos* con 64.4% y 81.4%; *Arquitecturas de Bases de Datos* con 83.7% y 85.7%) podemos concluir que el cambio de orientación en este ámbito se ha realizado con éxito.

3.2. Percepción del profesorado

En el análisis de los primeros resultados, no podíamos renunciar a las percepciones expresadas por el conjunto del profesorado. Estas percepciones se han recogido a partir de los informes escritos de evaluación de las asignaturas que dichos profesores realizan y entregan al final de cada curso a la dirección del Grado, así como a partir de conversaciones directas con ellos. Las resumimos a continuación relacionándolas con las categorías de conclusiones y excusas expuestas en la sección 2:

Al respecto de **C1 (Ofrecer al inicio de los estudios la visión completa del DS)**, a pesar que este contenido quedó incluido en la asignatura *Trabajo en equipo en la red*, sólo los primeros semestres en que se impartió, fue obligatoria su lectura y estudio. La dinámica de la asignatura, centrada en la adquisición de competencias transversales, llevó, poco a poco, a ir *olvidando*, por parte del profesorado responsable de dicha asignatura, la obligación de trabajar estos contenidos. En nuestra universidad, la dirección del Grado puede recordar y forzar esta obligatoriedad, pero parece más productivo buscar la forma en que la incorporación de estos contenidos no sea forzada sino natural por parte del profesorado al frente de dicha asignatura, profesorado de perfil no técnico, sino de formación en pedagogía y especializado en los procesos de *e-learning*.

Como hemos apuntado al hablar de los resultados de satisfacción y rendimiento, **C2 (Introducir en paralelo la IS y la P)** se ha implementado sin demasiados problemas curriculares. En IS el nivel que se va alcanzando con la sucesión de las asignaturas se percibe como bueno y suficiente. Sin embargo, en P se perciben problemas de nivel tanto en la segunda *Prácticas de programación*, como en la finalista *Diseño de Estructuras de Datos*. La explicación que ofrecen estos profesores se basa en un hecho ajeno a nuestra propuesta curricular: muchos estudiantes tienen convalidada la primera asignatura (*Fundamentos de Programación*) al tener estudios previos (Ciclos Formativos de Grado Superior o carreras empezadas/terminadas tiempo atrás) y su nivel real es demasiado bajo. Y las dos asignaturas requieren claros conocimientos de algorítmica y habilidades en las técnicas y lenguajes de programación. Por otro lado, los contenidos de referencia utilizados han resultado tener un enfoque ciertamente formal, con lo que se dificulta a los estudiantes ponerlos en práctica.

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

En general la percepción sobre **C3 (Optimización de los contenidos ofrecidos)** y sobre **C4 (Contribución de otras asignaturas al aprendizaje del DS)** coincide con lo asumido en el diseño del plan de estudios y que ya hemos indicado en 1.1.: se han evitado duplicidades y se han reordenado satisfactoriamente los contenidos, pero no se ha logrado una interacción fluida con asignaturas de otros ámbitos (como *Lógica*, *Grafos y Complejidad*, *Ingeniería de Requisitos* y *Gestión de Proyectos*) con las que existen puntos de conexión. Esto último se ha debido, principalmente, a la complejidad de organizar la comunicación y colaboración entre los profesores de cada ámbito y a los tempos y urgencias de implantación de las diferentes asignaturas que en ocasiones no facilitaban la coordinación entre el profesorado responsable de las mismas.

Donde no parece que se haya tenido un éxito claro es en la idea de **E1 (El Grado debe contentar a otros ámbitos de conocimiento, no sólo al DS)**. Los profesores de ámbitos como sistemas operativos, sistemas distribuidos o redes de computadores siguen percibiendo como insuficiente el nivel de habilidades de programación y algorítmica de los estudiantes que llegan a sus asignaturas, lo que da una idea del salto de nivel de exigencia que se produce en estas asignaturas sobre las necesidades de P. Por supuesto, podemos volver a argumentar, desde el ámbito de DS, que es debido a la abundancia de convalidaciones. Pero el hecho es que parece clara la necesidad de más refuerzo en P que el que se ofrece en las dos asignaturas obligatorias. Y eso a pesar que esas asignaturas, como hemos dicho, ya se ampliaron con contenidos no previstos inicialmente (como complejidad y recursividad) y que siguen recibiendo la demanda de incorporar también contenidos sobre diseño y programación de estructuras de datos, fundamentales en los mencionados ámbitos. Parece claro, pues, que sólo 12 créditos obligatorios de P no son suficientes para las asignaturas de los otros ámbitos.

La discusión al respecto de **E2 (La falta de interés y madurez de los estudiantes)**, no se ha vuelto a producir, una vez asumido que P es necesaria desde el primer momento, en tanto es requerida en otros ámbitos. Lo interesante en este aspecto es que tampoco se habla ya que los estudiantes no sean suficientemente maduros para afrontar en sus primeros cursos los principios de IS: en cierta forma se ha demostrado que ésa era una prevención superable con una buena organización curricular.

Finalmente en algunas de las conversaciones siguen apareciendo evidencias de **E3 (La poca predisposición al cambio del profesorado)**, pero no tanto orientadas al mantenimiento de la comodidad del *statu quo* actual, sino desde la convicción de la bondad de los planteamientos originales.

4. Conclusiones

Realizada una propuesta inicial (libre e ideal) de reordenación curricular (de *darle la vuelta*, en definitiva) del ámbito del DS; llevada al diseño en asignaturas concretas de un Grado en Ingeniería Informática; e implantada realmente y experimentada tras unos semestres de docencia, hemos intentado en este artículo cerrar la reflexión sobre este proceso, sus resultados y el aprendizaje que, como profesores, hemos tenido del mismo.

Para finalizar cabe intentar ahora dar una respuesta global a nuestra pregunta original de 2007: "¿Podemos darle la vuelta a la enseñanza del desarrollo del software?" Como siempre la respuesta no es ni taxativamente positiva ni taxativamente negativa. Hemos de admitir en negativo que:

- El nivel de P conseguido no es suficiente para iniciarse en otros ámbitos. Redes de computadores, sistemas operativos o sistemas distribuidos requieren más habilidades de las que podemos ofrecer en 12 créditos obligatorios en el primer curso, máxime cuando muchos estudiantes convalidan parte de ellos.
- No hemos encontrado el mecanismo para implicar al profesorado de otros ámbitos colindantes con DS.

Por el lado positivo, podemos defender que:

- Hemos introducido en el primer curso la descripción del ámbito del DS. Luego los estudiantes tienen herramientas para disponer de la visión global del DS al inicio de su formación.
- Al paralelizar P e IS, hemos neutralizado los tópicos sobre la madurez y los intereses de los estudiantes. Estos se encaran con éxito a la IS en los primeros cursos, y siguen introduciéndose, como hasta ahora, en P, al ser esta habilidad un instrumento básico para otros ámbitos.
- Hemos evitado duplicidades y uniformizado los temas que hablan del DS.
- Y hemos obtenido buenos (o tan buenos como los de las ingenierías anteriores al Grado) resultados de satisfacción y rendimiento de los estudiantes en las nuevas asignaturas.

En definitiva, siguiendo la idea del "casi" del título de esta contribución, debemos admitir que no hemos llegado a darle la vuelta completamente, pero que, con todo, hemos *removido* el diseño curricular, es decir, que hemos hecho un ejercicio de innovación para salir de nuestra zona de confort y replantearnos y afrontar los tópicos adquiridos y asumidos, con los años, como ciertos al respecto del aprendizaje del DS. Y que, en cualquier caso,

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

nos hemos quedado razonablemente cerca de conseguir implementar en la realidad el giro que imaginamos sobre el papel en 2007.

Agradecimientos

Este trabajo ha sido parcialmente apoyado por la *Secretaria d'Universitats i Recerca de la Generalitat de Catalunya* (2014 SGR 1534)

Referencias

[1] Kurose, J.F & Ross, K.W. *Computer Networking: A Top-Down Approach Featuring the Internet*, 2/e Addison Wesley, 2005

[2] Meyer, B. Towards an object-oriented curriculum. *Journal of Object-Oriented Programming*, 6(2):76–81, May 1993.

[3] Pedroni, M. & Meyer, B. The Inverted Curriculum in Practice, *Proceedings of SIGCSE 2006*, *ACM*, Houston, Texas, 1-5 March 2006.

[4] Meyer, B. Software engineering in the academy. *Computer*, 34 (5):28-35, May 2001.

[5] Marco-Simó, J. M.; Guitart, I.; Marco, M. J.; Rius, M.A.; Rodríguez, M. E (2007). ¿Podemos darle la vuelta a la enseñanza del desarrollo del software? En *Actas de las XIII Jornadas de Enseñanza Universitaria de la Informática*, *JENUI 2007*.

[6] Marco-Simó, J. M.; Guitart, I.; Marco, M. J.; Rius, M.A.; Rodríguez, M. E.; Arnedo, J.; Cabot, J.; Caballé, S.; Riera, D. ¿Podemos darle la vuelta a la enseñanza del desarrollo del software? *Novática*, 193: 59- 62, Mayo 2008

[7] Martín, G. & Ruiz, E.; Ingeniería Informática: ¿más que solo un título, menos que sólo una profesión? *Novática*, 188: 51-63, Julio 2007.

[8] BOE, núm. 187 de 4/08/2009

[9] Marco, M.J; Marco-Simó, J.M.; Prieto, J.; Segret, R. (eds.) *Escaneando la informática*. Editorial UOC, 2010. ISBN: 978-84-9788-107-4.