



Una experiencia para la mejora del aprendizaje del modelado en Ingeniería del Software

Sergio Ilarri, José Merseguer, Raquel Trillo,
Diego Pérez, Francisco José Berlanga

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
{silarri, jmerse, raqueltr, diegop, berlanga}@unizar.es

Resumen

Analizar un sistema software es una tarea compleja que requiere buenas dotes de abstracción. Por ello, nos planteamos mejorar y potenciar las capacidades de modelado del alumnado. Por un lado, nuestra propuesta es explorar la relación que existe entre las tareas de análisis en el desarrollo de software (concretamente, considerando la construcción de diagramas de clases UML) y el modelado de datos para el diseño de bases de datos (diseño conceptual utilizando el modelo E/R). Por otro lado, proponemos plantear actividades de evaluación entre pares como evaluación formativa, con objeto de desarrollar en el alumnado el espíritu crítico y la capacidad de análisis.

En relación con los planteamientos indicados, en este artículo describimos una experiencia llevada a cabo durante el curso 2011/2012 en el contexto de la asignatura Ingeniería del Software II de la titulación de Ingeniería en Informática de la Universidad de Zaragoza. Las conclusiones extraídas con la realización de esta primera experiencia nos hacen ser positivos de cara a una posible ampliación de estos planteamientos en próximos cursos.

Palabras clave: Ingeniería del Software, análisis, evaluación entre pares, Bases de Datos.

Recibido: 28 DE ABRIL DE 2012; **Aceptado:** 15 DE MAYO DE 2012.

1. Contexto de la experiencia desarrollada

1.1. Contexto en la titulación

La experiencia que se describe en este artículo se desarrolló en el marco de la asignatura Ingeniería del Software II, troncal de cuarto curso de la titulación de Ingeniería en Informática de la Universidad de Zaragoza (ver Cuadro 1). Dicha titulación está en extinción y va a ser reemplazada por el nuevo Título Oficial de Graduado o Graduada en Ingeniería Informática de la Universidad de Zaragoza. El curso 2012/2013 será el último curso en el que se impartirá la asignatura y se pretende que el trabajo desarrollado sirva como primera experiencia para su posible adaptación y extensión a otras asignaturas de Ingeniería del Software y de Bases de Datos en el nuevo grado.

La asignatura se imparte en dos cuatrimestres durante el curso 2011/2012, un grupo en el cuatrimestre de otoño y otro en el cuatrimestre de primavera. La experiencia que describe este artículo fue llevada a cabo durante el primer cuatrimestre. Tuvo carácter voluntario, pero con el incentivo de poder obtener hasta 1 punto extra en la calificación final de la asignatura.

En ella participó un 30 % de los 54 estudiantes matriculados. Aunque también se ofreció la posibilidad de participación al alumnado del grupo del cuatrimestre de primavera, el tamaño de este grupo fue demasiado reducido para obtener conclusiones docentes de interés.

Es interesante señalar que parte del profesorado de la asignatura también imparte (o ha impartido) docencia en asignaturas de Bases de Datos (Ficheros y Bases de Datos, Diseño de Bases de Datos Relacionales y Bases de Datos Avanzadas). Esto nos permitió complementar bien las dos componentes de análisis y modelado que se evalúan con esta experiencia y nos facilitará una mayor interdisciplinariedad entre asignaturas de Bases de Datos e Ingeniería del Software en futuras extensiones y adaptaciones del trabajo desarrollado.

1.2. Contexto tecnológico

Podemos decir de manera simplificada que el ciclo de vida del software se compone de diferentes etapas. Primero, una captura de requisitos con el cliente (*Fase de Requisitos*) que se convierten en un análisis (*Fase de Análisis*). El análisis se plasma posteriormente en un diseño (*Fase de Diseño*) que, finalmente, se implementa (*Fase de Implementación*) y se prue-

ASIGNATURA: Ingeniería del Software II
Titulación: Ingeniero en Informática (BOE 1-2-1995)
Universidad: Universidad de Zaragoza
Curso: 4 (Segundo Ciclo) **Cuatrimestre:** 1 y 2 **Carácter:** Troncal
Departamento: Informática e Ingeniería de Sistemas
Área: Lenguajes y Sistemas Informáticos
Créditos: 7,5 UZ (6,4 ECTS)
45 h de teoría 15 h de ejercicios 15 h de prácticas de laboratorio
Profesorado (en el curso 2011/2012):
José Javier Merseguer Hernaiz (correo electrónico: jmerse@unizar.es)
Sergio Ilarri Artigas (correo electrónico: silarri@unizar.es)
Raquel Trillo Lado (correo electrónico: raqueltl@unizar.es)
Francisco José Berlanga Rivera (correo electrónico: berlanga@unizar.es)
Diego Pérez Palacín (correo electrónico: diegop@unizar.es)
Idioma en que se imparte: Castellano
Número aproximado de alumnos: 70 (54 en el primer cuatrimestre)
Grupos de teoría: 2
Grupos prácticos: 10

Cuadro 1: Información general sobre la asignatura Ingeniería del Software II

ba (*Fase de Pruebas*). Realizar un buen análisis es fundamental para llevar a cabo con éxito el desarrollo de un sistema software.

Sin embargo, el análisis es también una etapa que plantea grandes dificultades. Así, por ejemplo, los requisitos pueden estar especificados de forma ambigua, incompleta, o imprecisa. Esta labor de transformación de requisitos en análisis entraña una gran dificultad para el alumnado, dado que requiere una capacidad de abstracción importante para identificar los conceptos y relaciones relevantes. Los requisitos pueden ser *funcionales* (cuando se refieren a la funcionalidad del software) o *no funcionales* (como la seguridad, la usabilidad o las prestaciones). En este trabajo nos centramos exclusivamente en los requisitos funcionales.

Las tareas de análisis tienen dos aspectos fundamentales: el análisis de la estructura estática del sistema software y el análisis de su comportamiento. En este trabajo nos centramos exclusivamente en el análisis de la parte estática del software, es decir, de la información subyacente que queremos modelar. El análisis de la información juega un papel fundamental para un Ingeniero en Informática, tanto para el desarrollo de sistemas software en general como para el diseño de bases de datos en particular. Para estos fines suelen utilizarse dos modelos distintos:

- Para el análisis de sistemas software en general (fase de análisis estático o de la información) se utiliza habitualmente el lenguaje UML [13] (*Unified Modeling Language*), herramienta fundamental del análisis y diseño orientado a objetos. Este lenguaje se utiliza en la asignatura objeto de estudio y será un elemento fundamental en la especialización en Ingeniería del Software

en el nuevo Título Oficial de Graduado o Graduada en Ingeniería Informática de la Universidad de Zaragoza.

- Para el diseño conceptual de bases de datos (análisis de datos) se utiliza habitualmente el modelo *Entidad/Relación* [2] (*E/R*, también conocido como *Entidad/Interrelación*, *Entidad-Relación*, o *E-R*). Aunque podría utilizarse UML para el diseño conceptual de bases de datos, se puede decir que hay un mayor soporte en favor de la utilización del modelo E/R [1, 12, 14, 16]. El modelo E/R se ha venido utilizando para el diseño conceptual en diversas asignaturas de Bases de Datos de la titulación de Ingeniería en Informática de la Universidad de Zaragoza. En concreto, *Ficheros y Bases de Datos*, *Diseño de Bases de Datos Relacionales* y (en menor medida) *Bases de Datos Avanzadas*. Además, también será la herramienta de diseño conceptual que se utilizará en las asignaturas de Bases de Datos del nuevo Título Oficial antes referido. En particular, *Bases de Datos*, que es una asignatura obligatoria, y *Bases de Datos 2*, que es una asignatura obligatoria dentro de las especializaciones de *Sistemas de Información y Tecnologías de la Información*.

Torres et al. [18] señalan la complementariedad entre ambos modelos e indican que «*UML does not provide resources to model database persistency*». La utilización de dos modelos distintos para realizar el análisis (en función de si se están analizando los requisitos del software en general o los requisitos de datos en particular) resulta pertinente, dado que es lo habitual¹. Sin embargo, en ocasiones esta duplicidad de herramientas de modelado crea confusión en el alumnado. Así,

¹No obstante, podría transformarse parte del sistema software modelado en un diagrama de clases UML (la parte a la que queremos dar persistencia) a un esquema lógico de bases de datos.

como indica Neubauer [11], «*If the presentation of multiple modeling techniques is not carefully planned, students may confuse terms and mix modeling notations*». Por esta razón, al presentar al alumnado la relación existente entre estas dos herramientas, se incidió también en las diferencias entre las notaciones de ambas. No obstante, Martínez-González y Duffing [9] enfatizan que la verdadera dificultad para los estudiantes no está en el modelo o notación utilizada, sino en la dificultad de abstraerse, es decir, de modelar o conceptualizar. Por tanto, lo más importante es ayudar a los estudiantes a que desarrollen esas destrezas de abstracción y conceptualización necesarias. Así podrán obtener diagramas de clases y diagramas E/R que serán precisos, correctos y completos.

Dada la escasa experiencia del alumnado en este tipo de tareas de análisis, resulta fundamental apoyar el aprendizaje con numerosos ejemplos y ejercicios. Finalmente, el trabajo en equipo y la aplicación de técnicas de evaluación entre pares podría facilitar al alumnado el desarrollo de su capacidad de análisis y espíritu crítico.

2. Necesidades detectadas

En esta sección describimos las necesidades que han sido detectadas por el profesorado tanto de las asignaturas de Ingeniería del Software como de las asignaturas de Bases de Datos y que han motivado el desarrollo de esta experiencia.

Las necesidades existentes se pueden resumir de la siguiente manera:

1. Dificultad del alumnado para comprender la relación que existe entre el análisis de sistemas software en general y de *bases de datos* en particular (diagramas de clases UML frente a diagramas E/R).
2. Dificultad del alumnado para realizar el análisis:
 - a) para el desarrollo de sistemas software, utilizando UML para construir diagramas de clases.
 - b) para el desarrollo de bases de datos, utilizando el modelo E/R para obtener esquemas E/R.

Esta experiencia se centra en las necesidades 1 y 2.a, aunque también incide marginalmente sobre la 2.b al analizarse la relación existente los esquemas E/R y los diagramas UML.

2.1. Descripción detallada de las necesidades

Por un lado (Necesidad 1), el alumnado tiene dificultad en ver la relación existente entre los diagramas de clases UML a nivel de análisis y los diagramas E/R de bases de datos (diseño conceptual de la base de datos). Además, les resulta difícil entender la forma en la que ambos pueden coexistir en el desarrollo de un sistema software. La relación existente se muestra en la Figura 1 de forma resumida y centrando la atención en las etapas de interés para esta experiencia. Como se observa en la figura, se considera que la etapa de *diseño conceptual* de

una base de datos está relacionada con la etapa de *análisis* de un sistema software, más que con la etapa de diseño de dicho sistema.

Por otro lado (Necesidad 2), hemos detectado la dificultad del alumnado para abstraerse a la hora de realizar el análisis de sistemas (análisis estructural con diagramas de clases UML) y de bases de datos (diseño conceptual con el modelo E/R):

- La nota promedio alcanzada por el alumnado en ejercicios de modelado con diagramas de clases desde el curso 2008/2009 estaría en torno al 6.1 y menos de la mitad del alumnado es capaz de alcanzar una nota superior al 7.0 (asignatura Ingeniería del Software II).
- La dificultad del alumnado para realizar diseños de bases de datos correctos, completos y precisos es también un hecho observable. Así, tras acabar la asignatura Ficheros y Bases de Datos, el alumnado interesado puede profundizar en el diseño con la asignatura optativa de Diseño de Bases de Datos Relacionales. En esta asignatura todavía se detectan carencias en las habilidades de diseño que muestra el alumnado. Incluso en la asignatura Bases de Datos Avanzadas, que es una asignatura optativa más especializada cursada por alumnos con gran motivación en el tema, se ha observado como una carencia la falta de dominio en el desarrollo de esquemas E/R que sean completos, precisos y correctos. Esto es debido a que es una tarea que entraña gran complejidad.

Las necesidades que se pretenden abordar afectan directamente al alumnado y se consideran, por tanto, necesidades primarias. Además, son necesidades percibidas por el profesorado mediante observación del desempeño de los estudiantes en exámenes, pruebas prácticas y tutorías. Así, creemos preciso potenciar el desarrollo de la capacidad de abstracción y de análisis del alumnado. También creemos que debemos enfatizar la importancia de seguir una metodología correcta, con el fin de resolver problemas típicos que se encuentran en el análisis: diagramas incompletos, imprecisos o con exceso de información, uso incorrecto de los elementos de modelado, errores en la comprensión de conceptos, etc. Consideramos que los problemas detectados son bastante generalizados y, por tanto, la experiencia llevada a cabo se dirigió a todo el alumnado. Estos problemas no son tan graves como para calificar estas necesidades como normativas (objetivos curriculares que no se cumplen). Sin embargo, sí consideramos interesante mejorar el grado en el que el alumnado alcanza dichos objetivos.

Por último, es importante destacar que se priorizaron las necesidades según el orden mostrado anteriormente, aunque se considera que se pueden abordar ambas simultáneamente. No obstante, es conveniente aclarar que, con respecto a la Necesidad 2, no se pretendió resolver las dificultades del alumnado en el diseño conceptual de bases de datos (Necesidad 2.b). Esto excedería el ámbito del presente trabajo, pues ese problema concierne a asignaturas de Bases de Datos y no a la asignatura Ingeniería del Software II. Así, nos hemos centrado

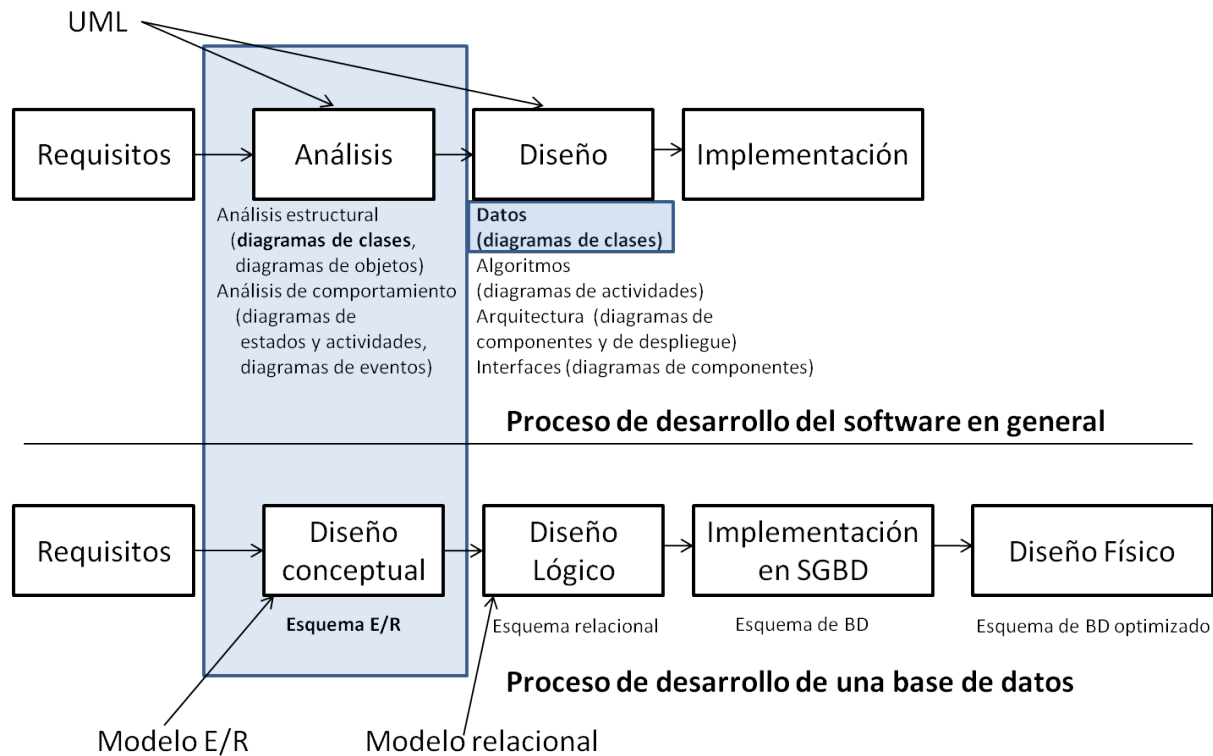


Figura 1: Proceso de desarrollo del software en general y de bases de datos

en las necesidades 1 y 2.a. Abordamos parcialmente la interdisciplinariedad entre las materias de Ingeniería del Software y Bases de Datos al incidir en la relación entre las mismas y plantear en la asignatura Ingeniería del Software II una práctica de laboratorio que exige el diseño de una base de datos. Sin embargo, con esto sólo se acomete parcialmente la Necesidad 2.b, por lo que sería preciso extender y adaptar en el futuro la experiencia a la materia de Bases de Datos para cubrir dicha necesidad de forma significativa.

Estas actividades de modelado que plantean dificultades al alumnado son complejas, dado que para su correcto desarrollo es preciso progresar hasta al menos el cuarto nivel de la taxonomía de Bloom [3, 7] (*análisis*) e incluso hasta el quinto nivel (*síntesis*), superando los niveles iniciales de *conocimiento*, *comprensión* y *aplicación*². A pesar de su dificultad, estas tareas son muy relevantes para el desarrollo profesional de un Ingeniero en Informática. Consideramos que puede mejorarse su desempeño con la propuesta planteada en este trabajo.

3. Objetivos de la experiencia

Los cinco objetivos planteados por orden de prioridad son:

1. Conseguir que en la prueba de evaluación final de la asignatura Ingeniería del Software II ninguno de los

alumnos participantes en la experiencia cometa errores graves en la utilización de los elementos de modelado de UML para la construcción de diagramas de clases.

2. Conseguir que el alumnado establezca la relación existente entre el análisis estático de un sistema software realizado con diagramas de clases UML y el diseño conceptual de bases de datos utilizando el modelo E/R. Para ello, se propone la realización de una práctica de laboratorio donde tengan que utilizarse ambos elementos para el desarrollo e implementación de un sencillo sistema software de acceso a bases de datos.
3. Conseguir que en la prueba final mencionada al menos un 65 % de los alumnos participantes obtengan una calificación igual o superior a 7.0 (sobre 10) en el diagrama de clases. En el curso 2010/2011 menos del 25 % de los alumnos consiguieron este objetivo.
4. Conseguir que en la práctica de laboratorio propuesta en el objetivo 2 al menos un 65 % del alumnado participante obtenga una calificación igual o superior a 7.0 en el diagrama de clases UML y en el diagrama E/R. En cursos anteriores no se realizó una evaluación concreta de estos diagramas sino de la práctica en su conjunto. Es por ello por lo que no podremos realizar una comparativa con cursos anteriores en este aspecto.

²Introduciendo actividades de evaluación entre pares, se pretende aquí incidir también en el sexto nivel de *evaluación*.

5. Conseguir que al menos un 50 % del alumnado y del profesorado participante evalúe de forma positiva la experiencia de evaluación entre pares.

A partir de los referentes previos mencionados en la Sección 2, consideramos que los objetivos planteados supondrán una mejora notable en la comprensión y la aplicación de los temas de análisis. Además de los objetivos indicados, también se pretende analizar la experiencia y extraer conclusiones de cara a la posible consolidación de la misma y a su posible adaptación o extensión a otras asignaturas relacionadas.

4. Actividades planteadas para el alumnado

Antes de comenzar la experiencia, se realizó una planificación de las actividades a llevar a cabo, tanto por parte del alumnado como por parte del profesorado participante. En esta sección nos centramos en las actividades realizadas por el alumnado participante en la experiencia (el profesorado, lógicamente, estuvo involucrado en diversas actividades de gestión de la experiencia, elaboración de materiales, evaluación, etc.):

1. Desarrollo de una práctica de laboratorio consistente en la realización de un sistema software que accede a una base de datos. Los alumnos elaboraron el diagrama de clases UML correspondiente al sistema software en su totalidad y el diagrama E/R correspondiente al diseño conceptual de la base de datos (Práctica 2).
2. Resolución de ejercicios de análisis elaborando diagramas de clases UML (Práctica 3 y un ejercicio adicional consistente en la transformación del diagrama E/R de la Práctica 2 en un diagrama de clases UML).
3. Desarrollo de actividades de evaluación entre pares, proporcionando correcciones detalladas sobre los diagramas elaborados por otros compañeros. Cada pareja evaluadora, utilizando una rúbrica preparada por el profesorado (donde se indicaban los aspectos a considerar para la corrección, pero no indicaciones acerca de cuál era la solución correcta, tal y como se muestra en el Anexo A), elaboró un pequeño informe y calificó el diagrama de clases UML del sistema desarrollado en la Práctica 2.
4. Recolección de las horas invertidas en actividades relacionadas con la experiencia.
5. Cumplimentación del cuestionario de satisfacción del alumnado (ver Anexo B).

En resumen, el alumnado realizó 4 ejercicios de modelado (a partir de descripciones de 2 casos de estudio diferentes). El primer ejercicio de modelado consistió en el diseño del esquema E/R de la Práctica 2 de laboratorio. El segundo ejercicio de

modelado implicó el desarrollo del diagrama de clases UML que modelaba el sistema (aplicación Java) que debía desarrollarse en la Práctica 2. El tercer ejercicio de modelado implicaba la transformación del esquema E/R de la base de datos de la Práctica 2 en un diagrama de clases UML, lo que ayudó a entender las diferencias de notación entre ambas herramientas de modelado. Finalmente, el cuarto ejercicio de modelado consistió en el desarrollo de un diagrama de clases UML para un caso de estudio considerado en la Práctica 3 de laboratorio. La realización de un diagrama de clases UML en la Práctica 3 es algo que ya venía realizándose anteriormente en la asignatura, pero como parte de la experiencia se añadieron las otras 3 actividades de modelado mencionadas, además de las actividades de evaluación entre pares.

Además, el profesorado evaluó tanto los ejercicios entregados como las correcciones realizadas sobre los mismos. El profesorado entregó a cada pareja la corrección final correcta del ejercicio que realizó y, en el caso de que ese ejercicio conllevara también una actividad de evaluación entre pares, también la evaluación de la corrección que efectuó dicha pareja. Además, comentó con los alumnos los errores comunes observados y el alumnado tuvo la oportunidad de asistir a tutorías para obtener aclaraciones adicionales sobre sus tareas (actividades de modelado o de evaluación de otros compañeros). Para el caso concreto de los ejercicios de modelado de la Práctica 2, el alumnado tuvo la oportunidad de mejorar sus modelos a partir de las evaluaciones entre pares, de modo que el profesorado corrigió una versión de los modelos que pudo ser mejor que la inicialmente propuesta.

Como puede comprobarse en la descripción de las actividades, una parte importante de la propuesta es el desarrollo de *trabajo colaborativo* y de *evaluación entre pares* para fomentar el espíritu crítico y el aprendizaje entre pares. Diversos trabajos han enfatizado el interés de utilizar actividades de evaluación entre pares en general [17, 20] y en el contexto de la Informática [6, 8, 15, 19] y de la Ingeniería del Software [6], Bases de Datos [10] y Sistemas de Información [15] en particular, por lo que se considera interesante llevar a cabo esta experiencia con el propósito de mejorar las habilidades de modelado del alumnado. Por ejemplo, Garousi [6] muestra que no hay muchos cursos en el campo de la Ingeniería que incluyan actividades de evaluación por pares, pero con la experiencia realizada ahí se extrae la conclusión de que «*Applying peer reviews greatly enhances the experience of students, leading to a more complete learning experience*». Además, enfatiza que merece la pena que los alumnos entrenen y practiquen estas habilidades, dado que un ingeniero se espera que desarrolle su carrera profesional posteriormente en un entorno de trabajo guiado por prácticas de evaluación entre pares.

Por su participación voluntaria en la experiencia, se ofreció hasta 1 punto extra al alumnado participante, valorando tanto los propios diseños realizados como las correcciones efectuadas sobre los diseños de otros compañeros. Es importante destacar que se decidió no tener en cuenta las calificaciones otorgadas por las correcciones de otros compañeros, ya que se pretendía utilizar la evaluación entre pares como forma

de *evaluación formativa* [8, 20] y no como medio para sustituir o complementar la evaluación del profesorado. Además, Sindre et al. [15] advierten que no es aconsejable usar para la evaluación final las calificaciones otorgadas entre compañeros en una evaluación por pares.

5. Evaluación de la experiencia realizada

En esta sección se presenta la evaluación de la experiencia de innovación propuesta. Separamos la evaluación en distintos aspectos: evaluación inicial del diseño, evaluación del desarrollo y evaluación de los resultados.

5.1. Evaluación inicial del diseño

Al plantear el diseño de esta experiencia ya se realizó una primera actividad de evaluación. En concreto, el diseño responde a dos preguntas que se plantearon y que constituyen una *evaluación proactiva*: ¿cuál es el problema que se quiere resolver o los aspectos que se pretenden mejorar? (ver Sección 2), ¿cuáles son los objetivos establecidos? (ver Sección 3) y ¿cuál es el programa o proyecto que se plantea? (ver Sección 4).

5.2. Evaluación del desarrollo

Un aspecto importante de la evaluación es la comprobación de si la experiencia se iba desarrollando de acuerdo con el diseño planteado. El objetivo de esta evaluación (*evaluación clarificativa*) es detectar posibles desviaciones e imprevistos que pueden ir surgiendo durante la realización de la experiencia. Esta evaluación, junto con la evaluación inicial del diseño, constituye la *evaluación formativa*.

El objetivo es evaluar la ejecución del diseño, valorando la coherencia existente entre el diseño planteado y su desarrollo real. Para ello, valoramos los siguientes aspectos:

- *La participación del alumnado.* Este es un elemento fundamental para el éxito final. Por ello, al término de cualquier actividad del alumnado relacionada con la experiencia, se recogió información relativa al grado de participación del mismo. En las actividades que eran directamente supervisadas u observadas por el profesorado (prácticas de laboratorio) se contó fundamentalmente con su opinión sobre la participación del alumnado en las mismas.
- *El desempeño del alumnado en las actividades de evaluación entre pares.* Al finalizar las actividades de evaluación entre pares el profesorado responsable valoró el desempeño mostrado por el alumnado en dichas actividades, determinando si realizó una actividad de evaluación seria o simplemente para “cubrir el expediente”. Este aspecto está muy relacionado con el anterior,

aunque se centra en el grado de implicación del alumnado concretamente en las actividades de evaluación entre pares. En concreto, se observó que no todos los participantes se tomaron las tareas de evaluación entre pares con la misma seriedad.

- *La respuesta del alumnado ante la innovación.* Era importante reflexionar también sobre las percepciones del alumnado y sobre las relaciones que se observaban entre los alumnos. Por ejemplo, ¿aparecían rencillas debido a las evaluaciones realizadas sobre otros compañeros?, ¿se observaba algún problema en el desarrollo de las actividades que podía suponer un obstáculo para la ejecución correcta del diseño de la experiencia?, ¿el alumnado tenía la sensación de que las actividades les resultan útiles o de que se estaba perdiendo el tiempo?, ¿se encontraban dificultades para poner en marcha la innovación? No se observaron problemas en este sentido, a excepción de que la participación en la experiencia no fue muy alta. No obstante, el alumnado participante en general respondió bien a la innovación.

Para realizar este seguimiento, se realizaron diversas reuniones ordinarias entre los profesores participantes en la experiencia. Aunque estaba prevista la posibilidad de celebrar reuniones extraordinarias, a petición de cualquiera de los profesores participantes, finalmente no fue necesario convocar ninguna.

5.3. Evaluación de los resultados

Al finalizar la experiencia, se realizó una evaluación de los resultados obtenidos (*evaluación sumativa*), tanto desde el punto de vista de la *eficacia* como de la *eficiencia* y la *satisfacción* de las partes implicadas. En primer lugar se determinó que la experiencia había podido desarrollarse en su totalidad, no habiendo ninguna circunstancia que impidiera el desarrollo de alguna de las actividades planteadas. Posteriormente, el impacto de la misma se evaluó mediante la comprobación de los objetivos expuestos en la Sección 3. Además, se valoró la satisfacción de las partes implicadas (tanto del alumnado como del profesorado participante) mediante el análisis de los cuestionarios de evaluación/satisfacción recogidos. Como parte de esta valoración, se consideraron también las horas invertidas por el alumnado y el profesorado en las actividades relacionadas con la experiencia, con el fin de determinar si la carga había resultado adecuada o requería algún ajuste.

La experiencia se desarrolló de forma completa, tal y como estaba previsto. En cuanto a los objetivos planteados inicialmente, se puede indicar lo siguiente:

- La participación en la experiencia no evitó completamente la aparición de errores en el examen final. En concreto, 4 de los participantes cometieron algún error que puede considerarse grave. Estos errores se deben a la inclusión en un “diagrama de clases adecuado a

la etapa de análisis” de elementos de diseño (concretamente, la utilización de atributos en lugar de asociaciones o la indicación de la navegabilidad de las asociaciones). Los profesores participantes en la experiencia asumen parte de la culpa, ya que en el mismo ejercicio se pedía también una implementación orientada a objetos y, por tanto, esto pudo despistar al alumnado incitándole a proporcionar un diagrama de diseño, a pesar de lo que se indicaba en el enunciado. También pudo contribuir a este problema el hecho de que en la actividades de evaluación entre pares se les pedía un diagrama de diseño, no de análisis.

- El alumnado participante logró establecer la relación existente entre el diseño de diagramas de clases con UML y el diseño conceptual de bases de datos utilizando el modelo E/R, ya que tuvo que realizar esta tarea como parte de una práctica de laboratorio.
- En cuanto a la nota en la prueba final, un 47 % de los alumnos participantes obtuvieron una calificación igual o superior a 7.0 en el diagrama de clases, no habiéndose alcanzado el objetivo del 65 % propuesto. No obstante, hemos analizado la situación y la razón se debe a la existencia de ciertos valores extremos. En concreto, 5 de los alumnos participantes en la experiencia suspendieron de forma global la asignatura, 3 de ellos con una nota muy baja (inferior a 3 en el examen). Si se eliminan estos casos (considerándolos atípicos), entonces un 70 % de los alumnos participantes habrían obtenido una nota igual o superior a 7.0 en el diagrama de clases. Una posible explicación para esta situación es que, al ser voluntaria la participación en la experiencia, no sólo las ganas de aprender sino también la posibilidad de obtener puntos extra de forma fácil pudo animar a involucrarse en la experiencia. En este sentido, no se observó una correlación clara entre la participación en el proyecto y el desempeño global del alumnado, a diferencia de lo que se ha informado en otros trabajos como los de Zywno [21].
- En cuanto a la práctica de laboratorio mencionada en el Objetivo 2, un 68 % de los alumnos participantes en la experiencia obtuvo una calificación igual o superior a 7.0 en el diagrama de clases UML y en el diagrama E/R entregados, superando el 65 % establecido como objetivo. Diversas notas inferiores al 7.0 se corresponden con los casos atípicos comentados en el punto anterior.
- En cuanto a la valoración de la experiencia por parte del alumnado y del profesorado participante, no se obtuvo ninguna valoración negativa y la sensación general es buena, como se comenta en detalle a continuación.

En las encuestas de satisfacción de los 16 alumnos participantes (ver Anexo B), de las que se recibieron 14, se obtuvieron los siguientes resultados: el 100 % considera que el tiempo que invirtieron en la experiencia es “normal”; un 79 %

considera que aprendieron “bastante” con el proyecto frente a un 21 % que indica que aprendieron “regular”; un 93 % califica las actividades como “ni aburridas ni divertidas” y un 7 % como “bastante divertidas”; finalmente, un 86 % valora su participación global en el proyecto como “interesante”, frente a un 14 % que la considera “de utilidad media”. Además, cada alumno participante empleó una media de 5.6 horas en la realización de ejercicios (hay que señalar un caso extraño que indicó 27 horas en este apartado), 1.2 en las correcciones, y 1.2 en otras tareas de gestión (el mismo alumno de las 27 horas de ejercicios indicó 10 horas invertidas en gestión). En cuanto a los comentarios positivos sobre la experiencia, la mayor parte de los alumnos participantes (71 %) comentó que les había ayudado a aprender más en la asignatura. Diversos alumnos también señalaron que les ayudó a llevar al día la asignatura, a evaluar mejor el trabajo propio y el de otros, a comprender la dificultad de las correcciones, e incluso a interiorizar los criterios de corrección. En cuanto a los comentarios más destacables sobre las sugerencias de mejora, un alumno propone la realización de experiencias similares en otras asignaturas y otro indica que «Creemos haber aprendido más con un proyecto sencillo como este que no con unas costosas prácticas [...]».

En las encuestas de satisfacción del profesorado (ver Anexo C), se obtuvieron los siguientes resultados: 2 de los 5 profesores participantes valoraron el tiempo invertido como “muy poco”, 2 como “poco” y 1 como “normal”; 2 de ellos considera que gracias a esta experiencia su actividad docente mejoró “poco” y 3 que mejoró “regular”; 4 valoraron las actividades de la experiencia como “ni aburridas ni divertidas” y 1 como “bastante divertidas”; 2 valoraron su participación global en el proyecto como “algo útil”, 1 como “de utilidad media”, 1 como “interesante”, y 1 como “muy interesante”. En cuanto a las horas invertidas en tareas relacionadas con la experiencia, cada profesor invirtió una media de 3.4 horas en la preparación de materiales, 3.4 horas en reuniones, y 6.2 horas en tareas de gestión; no obstante, se observa algunos desequilibrios que sugieren que hay que intentar lograr una mayor implicación del conjunto completo del profesorado en el proyecto. Llama la atención que el profesorado fue menos entusiasta que el alumnado en la valoración, pero tampoco se aportaron demasiados comentarios en las encuestas. Todos coinciden en que el proyecto permitió mejorar ciertos aspectos del aprendizaje del alumnado y algunos proponen sugerencias de mejora relacionadas con la necesidad de publicitar más la experiencia entre el alumnado para lograr una mayor participación del mismo.

5.4. Otras consideraciones

Al margen de los objetivos planteados y como consecuencia del desarrollo de la experiencia nos surgen otros ámbitos de evaluación interesantes. Por ejemplo, ¿seríamos capaces de valorar en qué medida el alumnado ha mejorado su capacidad de abstracción como resultado de llevar a cabo los ejercicios propuestos? Evidentemente a partir de los datos recopilados

no podemos obtener una respuesta. Sin embargo, nuestra experiencia docente a lo largo de los años unida a esta experiencia de innovación nos lleva a la siguiente reflexión. Creemos que a medida que los alumnos realizan más ejercicios de modelado su capacidad de abstracción se va formando y son capaces de relacionar conceptos de manera más clara. En consecuencia, sería interesante que las asignaturas de Ingeniería de Software y Bases de Datos propusieran sus ejercicios de manera coordinada, de modo que el alumno se enfrentase gradualmente a problemas cada vez más exigentes en abstracción.

6. Conclusiones

Aunque no se cumplieron todos los objetivos planteados, se considera que la experiencia fue positiva y se han determinado las posibles razones que han llevado al no cumplimiento de algunos de los objetivos. Las conclusiones extraídas con la realización de esta primera experiencia nos hacen ser optimistas de cara a una posible ampliación de la misma en próximos cursos, así como a su aplicación en otras asignaturas del nuevo Título Oficial de Graduado o Graduada en Ingeniería Informática de la Universidad de Zaragoza, tanto en las asignaturas de Bases de Datos como de Ingeniería del Software. Entre las principales lecciones aprendidas podemos señalar las siguientes:

- *Es necesario lograr una mayor participación del alumnado.* En esta primera experiencia determinamos que era mejor que la participación fuera voluntaria, dado que no estábamos seguros de qué resultados podríamos obtener ni del coste que podría suponer la realización de sus actividades. Sin embargo, esto llevó a una participación reducida (30 %) y a que no sea fácil valorar la representatividad del alumnado participante, dificultando la interpretación de los resultados obtenidos. Lograr una mayor participación e incluir más actividades de evaluación entre pares, con una planificación adecuada de los esfuerzos requeridos, podría mejorar el desarrollo de la experiencia y los resultados obtenidos.
- *Es necesario presentar en detalle y de forma clara la experiencia.* Dado que se trataba de una novedad en la asignatura, los profesores participantes actuamos probablemente con demasiada cautela, quizá dando a la innovación menos publicidad de lo que hubiera sido conveniente. Esto puede ser una de las causas de la baja participación y además llevó a algunas confusiones entre el alumnado, ya que en algunos casos entendieron que sólo con participar en el proyecto ya conseguirían automáticamente el punto extra con independencia de su desempeño.

Estamos convencidos de que estas lecciones extraídas serán muy útiles en futuras extensiones e implantaciones de esta experiencia.

Agradecimientos

La experiencia docente descrita en este artículo fue llevada a cabo en el marco de las convocatorias de innovación docente de la Universidad de Zaragoza correspondientes al curso 2011/2012; referencia del proyecto que ha financiado el trabajo: *Mejorando el Aprendizaje en el Modelado de Sistemas Informáticos: Una Primera Experiencia en Ingeniería del Software* (PIIDUZ_11_2_034). Además de este apoyo institucional, para el desarrollo del proyecto se contó con el soporte de los materiales de Escudero [4] y Rodríguez [5], así como con la ayuda (proporcionada a través de cursos) del Instituto de Ciencias de la Educación (ICE) de la Universidad de Zaragoza.

Referencias

- [1] Miloš Bogdanović, Aleksandar Stanimirović, Nikola Davidović y Leonid Stoimenov. The development and usage of a relational database design tool for educational purposes. En *Informing Science & IT Education Conference (InSITE'08)*, pp. 251–258, 2008.
- [2] Peter Pin-Shan Chen. The entity-relationship model — toward a unified view of data. *ACM Transactions on Database Systems*, Vol. 1, núm. 1, pp. 9–36, 1976.
- [3] Benjamin S. Bloom (editor). *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. Longman, 1956.
- [4] Tomás Escudero Escorza. Algunos principios básicos sobre los programas e innovaciones educativas, su diseño y su evaluación, febrero 2011. Programa de Formación del Profesorado, Instituto de Ciencias de la Educación (ICE) de la Universidad de Zaragoza.
- [5] Sebastián Rodríguez Espinar. La innovación y la mejora en la enseñanza universitaria. *Calidad de las Universidades y Orientación Universitaria*, pp. 83–102, 2002. ISBN 84-9700-077-3.
- [6] Vahid Garousi. Applying peer reviews in Software Engineering education: An experiment and lessons learned. *IEEE Transactions on Education*, Vol. 53, núm. 2, pp. 182–193, 2010.
- [7] Nurul Naslia Khairuddin y Khairuddin Hashim. Application of Bloom's taxonomy in software engineering assessments. En *Eight WSEAS International Conference on Applied Computer Science (ACS'08)*, pp. 66–69. World Scientific and Engineering Academy and Society (WSEAS), 2008.
- [8] Ilana Lavy y Aharon Yadin. Team-based peer review as a form of formative assessment – the case of a systems analysis and design workshop. *Journal of Information Systems Education*, Vol. 21, núm. 1, pp. 85–98, 2010.

- [9] María Mercedes Martínez-González y Gérald Duffing. Teaching databases in compliance with the European dimension of higher education: Best practices for better competences. *Education and Information Technologies*, Vol. 12, núm. 4, pp. 211–228, 2007.
- [10] John A. Mendonca. Implementing peer technical reviews in a large-sized database course. En *Information Systems Education Conference*, 2000.
- [11] Bruce J. Neubauer. Data modeling in the understanding database course: adding UML and XML modeling to the traditional course content. *Journal of Computing Sciences in Colleges*, Vol. 17, núm. 5, pp. 147–153, 2002.
- [12] James Rumbaugh. ER is UML. *Journal of Information Systems Education*, Vol. 17, núm. 1, pp. 21–26, 2006.
- [13] James Rumbaugh, Ivar Jacobson y Grady Booch. *El lenguaje unificado de modelado: Manual de Referencia*. Pearson Educación, 2007. ISBN 978-84-7829-087-1.
- [14] Devang Shah y Sandra Slaughter. Transforming UML class diagrams into relational data models. *UML and the unified process*, pp. 217–236, 2003. ISBN 1-931777-44-6.
- [15] Guttorm Sindre, Daniel L. Moody, Terje Brasethvik y Arne Solvberg. Introducing peer review in an IS analysis course. *Journal of Information Systems Education*, Vol. 14, núm. 1, pp. 101–119, 2003.
- [16] James Suleiman y Monica J. Garfield. Conceptual data modeling in the introductory database course: Is it time for UML? *Journal of Information Systems Education*, Vol. 17, núm. 1, pp. 93–100, 2006.
- [17] Keith Topping. Peer assessment between students in colleges and universities. *Review of Educational Research*, Vol. 68, núm. 3, pp. 249–276, 1998.
- [18] Alexandre Torres, Renata Galante y Marcelo Soares Pimenta. Towards a UML profile for model-driven object-relational mapping. En *XXIII Brazilian Symposium on Software Engineering (SBES'09)*, pp. 94–103. IEEE Computer Society, 2009.
- [19] Miguel Valero-García y Luis M. Díaz de Cerro. Autoevaluación y co-evaluación: estrategias para facilitar la evaluación continuada. En *Simpósio Nacional de Docencia en la Informática (SINDI'05)*, pp. 25–32. Thomson, 2005. Disponible en <http://bioinfo.uib.es/~joemiro/aenui/ProcSindi/P2005/1238.pdf>
- [20] Keith Willey y Mark Freeman. Completing the learning cycle: The role of formative feedback when using self and peer assessment to improve teamwork and engagement. En *17th Annual Conference of the Australasian Association for Engineering Education: Creativity, Challenge, Change; Partnerships in Engineering Education*, pp. 751–758. Australasian Association for Engineering Education, 2006.
- [21] Malgorzata S. Zywno. Optional activities as a way to improve student engagement and academic achievement in a large engineering class. En *International Conference on Engineering Education (ICEE'07)*, 2007.

A. Rúbrica para las actividades de evaluación entre pares



RÚBRICA PARA LA EVALUACIÓN DE DIAGRAMAS DE CLASES UML

Pareja evaluadora (indicar nombre, apellidos, y NIP):

Pareja evaluada (indicar nombre, apellidos, y NIP):

Categoría	Excelente	Bueno	Flojo	Malo	Nota (0-10)
Sintaxis general	Se respeta perfectamente la sintaxis de UML (considerando la sintaxis de los diagramas de clases UML equivalente a la de OMT vista en clase). Los nombres de las clases son adecuados/descriptivos, sustantivos, y empiezan con mayúsculas. Para cada clase, se indican sus atributos, métodos, visibilidad, etc. No hay incoherencias en el diagrama.	Hay algunos errores en la sintaxis de UML utilizada, pero no son muy importantes. Los nombres de las clases no son completamente adecuados/descriptivos, pero son sustantivos y empiezan con mayúsculas. Para alguna clase falta la especificación de algún elemento (atributos, métodos, etc.). Hay alguna incoherencia en el diagrama, pero poco importante.	Hay errores importantes en la sintaxis de UML utilizada. Los nombres de las clases son sustantivos. Faltan bastantes especificaciones de algunos elementos de las clases (atributos, métodos, etc.). Hay bastantes incoherencias en el diagrama.	Hay muchos errores en la sintaxis de UML utilizada. Los nombres de las clases son completamente inadecuados (por ejemplo, verbos). Faltan muchas especificaciones de los elementos de las clases, prácticamente sólo se indica el nombre. Hay muchas incoherencias en el diagrama.	
Atributos	Se han detectado todos los atributos. Los nombres de los atributos son adecuados/descriptivos, son sustantivos o adjetivos, y expresan una propiedad. Se indica la visibilidad adecuada de los atributos.	Falta algún atributo poco importante. Los nombres de los atributos no son completamente adecuados/descriptivos, pero son sustantivos o adjetivos que expresan una propiedad. Se indica la visibilidad de los atributos, pero a veces no es la adecuada.	Faltan bastantes atributos o algún atributo importante. Los nombres de los atributos son sustantivos o adjetivos que expresan una propiedad. Se indica la visibilidad de los atributos, pero muchas veces no es la adecuada.	Faltan muchos atributos importantes o no se indica ninguno. Los nombres de los atributos son completamente inadecuados (por ejemplo, verbos). No se indica la visibilidad de los atributos.	
Métodos	Se han detectado todos los métodos. Se han indicado todos los constructores. Se indican adecuadamente todos los argumentos de los métodos, así como sus tipos y el tipo del valor de retorno. Los nombres de los métodos son adecuados, son verbos, y expresan una acción/operación con sentido. Se indica la visibilidad adecuada de los métodos.	Falta algún método poco importante. Falta algún constructor. En general, se indican adecuadamente los argumentos de los métodos, así como sus tipos y el tipo del valor de retorno, aunque falta alguno. Los nombres de los métodos no son completamente adecuados, pero son verbos y expresan una acción/operación con sentido. Se indica la visibilidad de los métodos, pero a veces no es la adecuada.	Faltan bastantes métodos o algún método importante. Faltan bastantes constructores. No se especifican adecuadamente bastantes argumentos de los métodos, así como sus tipos y el tipo del valor de retorno. Los nombres de los métodos son verbos que expresan una acción/operación al menos con cierto sentido. Se indica la visibilidad de los métodos, pero muchas veces no es la adecuada.	Faltan muchos métodos importantes o no se indica ninguno. Faltan muchos constructores o no se ha indicado ninguno. No se especifican la adecuadamente la mayoría de los argumentos de los métodos, así como sus tipos y el tipo del valor de retorno, o bien faltan todos ellos. Los nombres de los métodos son completamente inadecuados (por ejemplo, sustantivos). No se indica la visibilidad de los métodos.	
Asociaciones	Se denotan correctamente todas las cardinalidades. Se especifican todos los roles necesarios. Se especifica de forma adecuada la dirección de navegación de todas las asociaciones.	Se denotan correctamente la mayoría de las cardinalidades. Se especifican la mayoría de los roles necesarios. Se especifica de forma adecuada la dirección de navegación de la mayoría de las asociaciones.	No se denotan correctamente bastantes cardinalidades. Muchos roles necesarios no se especifican. No se especifica de forma adecuada la dirección de navegación de muchas asociaciones.	No se indican las cardinalidades. No se especifica ningún rol, siendo que alguno de ellos era necesario. No se especifica de forma adecuada la dirección de navegación de la mayoría de las asociaciones o simplemente no se especifica la dirección de navegación.	
Presentación	Los distintos elementos del diagrama están bien ordenados. No hay faltas de ortografía. Las explicaciones que hay son claras, precisas, y necesarias.	Hay un cierto desorden pero se entiende adecuadamente. Hay alguna falta de ortografía. Algunas explicaciones que hay no son completamente claras, precisas, o necesarias.	Hay bastante desorden y dificulta la comprensión del diagrama. Hay un buen número de faltas de ortografía. La mayoría de las explicaciones que hay no son completamente claras, precisas, o necesarias.	Está muy desordenado, afectando seriamente a la comprensión del diagrama. Hay muchas faltas de ortografía. Muchas o la mayoría de las explicaciones que hay son poco claras, imprecisas, o innecesarias.	
Arquitectura del sistema	Se han detectado todas las clases. Se han identificado claramente las tres capas (vista, modelo, y persistencia). La interfaz entre las capas está bien definida a través de un único punto de acceso.	Falta alguna clase poco importante. Se han identificado claramente las tres capas (vista, modelo, y persistencia), pero la interfaz entre ellas no se ha definido con total precisión.	Faltan bastantes clases o alguna clase importante. No se han identificado las tres capas (vista, modelo, y persistencia), pero hay una cierta estructura que puede considerarse aceptable.	Faltan muchas clases importantes. No hay capas y las funcionalidades se han distribuido entre un número muy reducido de clases o con poca coherencia en la estructura.	
Nota global (0-10):					

Comentarios o sugerencias sobre la rúbrica:



Informe con comentarios y errores detectados:

B. Cuestionario de satisfacción: alumnado

Cuestionario de Satisfacción del Alumnado - Proyecto de Innovación Docente de Ingeniería del Software II

Rodea con un círculo la respuesta correcta:

- 1) El tiempo que he invertido en el proyecto de innovación me parece:
Muy poco - Poco - Normal - Mucho - Demasiado
- 2) Creo que gracias a las actividades del proyecto de innovación he conseguido aprender:
Muy poco - Poco - Regular - Bastante - Mucho
- 3) Las actividades del proyecto de innovación me han parecido:
Muy aburridas - Algo aburridas - Ni aburridas ni divertidas - Bastante divertidas - Muy divertidas
- 4) Globalmente, valoro mi participación en el proyecto de innovación como:
Poco útil - Algo útil - De utilidad media - Interesante - Muy interesante
- 5) En las actividades del proyecto de innovación he invertido este tiempo (en horas):

Ejercicios: _____ Correcciones: _____ Otros (gestión): _____

- 6) Gracias a las actividades desarrolladas en el proyecto de innovación, creo haber conseguido los siguientes objetivos:

- 7) Tengo las siguientes sugerencias de mejora para el proyecto de innovación:

C. Cuestionario de satisfacción: profesorado

Cuestionario de Satisfacción del Profesorado - Proyecto de Innovación Docente de Ingeniería del Software II

Rodea con un círculo la respuesta correcta:

- 1) El tiempo que he invertido en el proyecto de innovación me parece:
Muy poco - Poco - Normal - Mucho - Demasiado
- 2) Creo que gracias a las actividades del proyecto de innovación he conseguido mejorar mi actividad docente:
Muy poco - Poco - Regular - Bastante - Mucho
- 3) Mis tareas en el proyecto de innovación me han parecido:
Muy aburridas - Algo aburridas - Ni aburridas ni divertidas - Bastante divertidas - Muy divertidas
- 4) Globalmente, valoro mi participación en el proyecto de innovación como:
Poco útil - Algo útil - De utilidad media - Interesante - Muy interesante
- 5) En las actividades del proyecto de innovación he invertido este tiempo (en horas):

Preparación material: _____ Reuniones: _____ Gestión: _____

- 6) Gracias a las actividades desarrolladas en el proyecto de innovación, creo haber conseguido los siguientes objetivos docentes:

- 7) Tengo las siguientes sugerencias de mejora para el proyecto de innovación:



Sergio Ilarri Artigas es Ingeniero en Informática por la Universidad de Zaragoza y doctor por el programa de doctorado en Ingeniería de Sistemas e Informática de dicha universidad. Es profesor del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Imparte docencia fundamentalmente en materias relacionadas con bases de datos e ingeniería del software.

Sus áreas de investigación incluyen la gestión de datos para computación móvil y sistemas distribuidos, las redes de vehículos, y la Web Semántica.



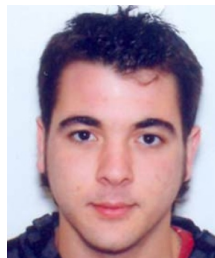
José Merseguer Hernaiz es Licenciado en Informática por la Universidad Politécnica de Valencia y doctor Ingeniero en Informática por la Universidad de Zaragoza. Es profesor del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Imparte docencia en materias relacionadas con la ingeniería del software, siendo ésta el área de su investigación.

Desde el curso 2009-2010 es Coordinador del Master Universitario en Ingeniería de Sistemas e Informática de la Universidad de Zaragoza.



Raquel Trillo Lado es Ingeniera en Informática y doctora en el programa de Computación por la Universidad de A Coruña. Desde el curso 2004/2005 es profesora del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, donde imparte docencia en materias relacionadas con la ingeniería del software, bases de datos

y sistemas de información. Además participa en proyectos de investigación en las áreas de Web Semántica, Computación Móvil y Recuperación de Información.



Diego Pérez Palacín es Ingeniero en Informática por la Universidad de Zaragoza. Desde el curso 2008/2009 realiza su doctorado en el programa de posgrado en Ingeniería Informática de la Universidad de Zaragoza. Desde el curso 2011/2012 colabora en la docencia de las asignaturas Ingeniería del Software II y Estructuras de Datos y Algoritmos.

Entre sus intereses de investigación se encuentran la calidad del software, los métodos formales y el software autoadaptativo.



Francisco José Berlanga Rivera es Ingeniero en Informática y doctor en el programa de Diseño, Análisis y Aplicaciones de Sistemas Inteligentes por la Universidad de Granada. Desde el curso 2008/2009 es profesor del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, donde imparte docencia en la asignatura Ingeniería del Software II. Algunos de sus intereses en investigación son la Extracción de Conocimiento, los Algoritmos Evolutivos y los Sistemas Difusos.

©2012 S. Ilarri, J. Merseguer, R. Trillo, D. Pérez y F.J. Berlanga
Este artículo es de acceso libre, distribuido bajo los términos de la Licencia Creative Commons de Atribución, que permite copiar, distribuir y comunicar públicamente la obra en cualquier medio, sólido o electrónico, siempre que se acrediten a los autores y fuentes originales