

# Autenticidad y aprendizaje basado en proyectos en una asignatura sobre diseño de software

Rubén Béjar, Francisco J. Lopez-Pellicer, Javier Noguerras-Iso, F. Javier Zarazaga-Soria  
Departamento de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza  
rbejar@unizar.es, fjlopez@unizar.es, jnog@unizar.es, javy@unizar.es

## Resumen

Este artículo presenta un estudio sobre los aspectos relacionados con la autenticidad en el aprendizaje basado en proyectos aplicado al diseño de software, especialmente en los aspectos de dominio de problema de este diseño. Para proporcionar un problema relevante para los estudiantes, el dominio del proyecto es sobre sistemas de información geográfica aplicados a un campus universitario. Los estudiantes llevan a cabo sus proyectos siguiendo la metodología ágil Scrum. Los estudiantes aprenden sobre diseño dirigido por el dominio, que es un conjunto de principios y prácticas para diseñar software con modelos de dominio ricos. Para valorar el grado de éxito sobre la autenticidad proporcionada y otros resultados positivos esperados en entornos de aprendizaje basado en proyectos, se analizan encuestas rellenas por los estudiantes tres años consecutivos. Este análisis ha permitido aprender algunas lecciones que pueden ser útiles para diseñar entornos de aprendizaje similares. Estas lecciones pueden resumirse en la necesidad de equilibrar autenticidad y trabajo de los estudiantes, la importancia de facilitar la competición entre equipos, el difícil equilibrio entre aprender sobre un dominio y aprender sobre ingeniería del software, y en el papel que las prácticas de laboratorio tienen en el aprendizaje basado en proyectos.

## Abstract

This paper presents a study on the authenticity-related aspects of project-based learning applied to software design, especially on the problem domain aspects of this design. To provide a relevant problem for the students, the project domain is about geographic information systems applied to a University campus. The student teams carry out their projects following the agile methodology Scrum. The students learn Domain-Driven Design, which is a set of principles and practices to design software with rich domain models. To assess the degree of success in providing authenticity and

other positive outcomes that are expected in project-based learning environments, the answers to a questionnaire filled in by the students in three consecutive years are analyzed. This has allowed to extract several lessons that may be helpful to design similar courses. These lessons can be summarized in the necessity to balance the authenticity with the expected student work, the importance of facilitating the competition among the teams, the difficult balance between learning about a domain versus learning new software engineering techniques and the role that laboratory assignments have in project-based learning.

## Palabras clave

Diseño de software, Ingeniería del Software, Proyectos de Software, Aprendizaje Basado en Proyectos.

## 1. Introducción

Los aprendizajes basados en problemas y en proyectos se han aplicado con éxito en entornos de aprendizaje de ingeniería del software durante muchos años. El basado en proyectos busca una relación cercana a la realidad profesional, algo que la literatura sobre educación superior ha denominado “autenticidad”.

Este trabajo presenta un estudio basado en una asignatura que aplica aprendizaje basado en proyectos al diseño de software, y discute los resultados de una encuesta completada por los estudiantes en tres cursos consecutivos. La asignatura, llamada Laboratorio de Ingeniería del Software, está focalizada en los aspectos relacionados con el dominio de problema en el diseño del software, como los modelos de objetos de dominio [8, p. 116-124] y el diseño dirigido por el dominio (DDD, *Domain-Driven Design*) que pone el foco del diseño sobre el dominio de problema [7].

La encuesta se basa en un cuestionario diseñado para que los estudiantes valoren el grado de éxito o fracaso alcanzado en proporcionar autenticidad y otros resultados positivos que se esperan de los entornos de apren-

dizaje basados en proyectos. El análisis de sus resultados proporciona algunas lecciones que otros pueden encontrar útiles al diseñar entornos para el aprendizaje del diseño de software.

El resto del artículo se estructura como sigue: en la sección 2 se revisa literatura relevante relacionada con los aprendizajes basados en problemas y proyectos, con la autenticidad en la educación sobre diseño en ingeniería y sobre el diseño dirigido por el dominio; en la sección 3 se describen los elementos principales de nuestro entorno de aprendizaje y de los proyectos. La sección 4 explica el diseño de la encuesta y la sección 5 analiza las respuestas dadas por los estudiantes. La sección 6 discute algunas lecciones aprendidas. El artículo termina extrayendo algunas conclusiones y con algunas ideas para el futuro.

## 2. Trabajo relacionado

### 2.1. Aprendizajes basados en problemas y proyectos para el diseño de software

El aprendizaje basado en problemas es una aproximación educativa centrada en los estudiantes, bajo la guía de un tutor, que ocurre en grupos pequeños con los docentes actuando como facilitadores o guías [2].

En ingeniería del software, el aprendizaje basado en problemas se ha usado ampliamente. Por ejemplo, para enseñar los fundamentos del diseño y la programación orientada a objetos en el desarrollo de videojuegos [16] e incluso para dar forma a grados completos sobre ingeniería del software [6].

El aprendizaje basado en proyectos tiene mucho en común con el aprendizaje basado en problemas, pero también algunas diferencias: los proyectos suelen estar más cerca de la realidad profesional que los problemas, están orientados hacia la aplicación del conocimiento más que a la adquisición del mismo, y la gestión de recursos y la diferenciación de roles entre los estudiantes es muy importante [14].

El *ACM/IEEE Computer Science Curricula* señala que la mejor aproximación para aprender sobre el diseño de software es hacer que los estudiantes participen en proyectos para desarrollar sistemas de software, y que eso es más efectivo si usan aproximaciones iterativas o ágiles [1, p. 174]. Y efectivamente, el aprendizaje basado en proyectos se ha usado con éxito en ingeniería del software [13, 5].

### 2.2. Autenticidad en la educación sobre diseño en ingeniería

Los profesionales afrontan muchos problemas mal definidos durante su trabajo por la naturaleza compleja

de los entornos del mundo real [3] y por la naturaleza poco estructura de su trabajo [4]. Estos problemas son comunes en la práctica profesional y requieren competencias relacionadas con la resolución de problemas [11]. Sin embargo, muchos de los problemas que se usan en la educación son descripciones parciales de situaciones reales que pueden no proporcionar las habilidades de resolución de problemas que van a ser necesarias en el mundo real, así que es importante identificar aquellos problemas que pueden ser más efectivos para construir estas habilidades [4]. Una revisión de trabajos previos concluye que para ser efectivos, los problemas deben ser realistas, relevantes y actuales, abordar temas emergentes, hacer que los estudiantes se sientan identificados con, e interesados en, el problema y proporcionarles una oportunidad de explorar y aplicar conocimiento profesional [4].

Los problemas de diseño (de productos, procesos, sistemas y métodos) son los más frecuentes en la práctica profesional de la ingeniería [10] porque el diseño es esencial en la ingeniería [12, p. 28] [15, p. 66]. Estos problemas están entre los más complejos y peor estructurados [9], además de ser personales, creativos y permitir muchas soluciones distintas [18].

La búsqueda de la autenticidad en la educación, en el sentido de realizar actividades cercanas a situaciones del mundo real y evitar simplificaciones extremas, aparece en la literatura hace tiempo, aunque el término se ha usado sin una definición clara y para cosas diferentes. Algunos de los factores clave de la autenticidad en la educación que aparecen en la literatura son: situaciones profesionales del mundo real, toma de decisiones en contextos prácticos y uso de datos reales [17]. Además, los problemas mal estructurados, tan característicos de la ingeniería, son un estímulo para las actividades auténticas [4].

### 2.3. Diseño dirigido por el dominio

El diseño dirigido por el dominio es un conjunto de principios y prácticas para diseñar software con modelos de dominio ricos [7]. Tiene algunas características que lo hacen interesante en un entorno de enseñanza-aprendizaje:

- La estrecha relación entre modelos expresados en código y en diagramas, normalmente UML, mediante el denominado lenguaje ubicuo<sup>1</sup> (*ubiquitous language*), ayuda a que los estudiantes vean más claramente el significado de los distintos elementos de UML traducidos a código.

<sup>1</sup>Un lenguaje en constante evolución compartido entre expertos del dominio y desarrolladores, que conecta el código, los diagramas y las discusiones sobre el dominio mediante el uso de los mismos términos para los mismos conceptos.

- El fuerte énfasis que se hace en la colaboración con los expertos del dominio y en la comunicación entre los desarrolladores lo hace muy adecuado para un proyecto en el que se sigue una metodología ágil.
- Usa algunos patrones de diseño de software comunes como las *Factorías*, la *Arquitectura por Capas*, los *Servicios* o los *Objetos Valor*. Aunque algunos de esos patrones son conocidos por los estudiantes, su aplicación en un modelo de dominio rico guiado por el DDD proporciona excelentes ejemplos de su uso en contexto.

### 3. Entorno de aprendizaje

La asignatura en la que se basa este estudio se llama Laboratorio de Ingeniería del Software, y se imparte en el grado en Ingeniería Informática de la Universidad de Zaragoza, especialidad Ingeniería del Software, desde el curso 2013-14 en el cuatrimestre de primavera del cuarto curso. Es una asignatura de especialidad y el número de estudiantes matriculados es bajo (9 en 2013, 6 en 2014, 13 en 2015 y 6 en 2016).

La descripción de la asignatura enfatiza la importancia de afrontar distintos dominios de problema al diseñar sistemas de software. El dominio escogido para la asignatura es el de los sistemas de información geográfica en entornos inteligentes aplicado a un campus universitario. Ese dominio permite que los estudiantes trabajen en problemas relevantes y conocidos por ellos.

#### 3.1. Clases de teoría y prácticas

Las clases de teoría están centradas en el diseño dirigido por el dominio, con 10 horas para presentar y discutir los principales contenidos teóricos, y otras 10 horas para trabajar en el diseño de los modelos necesarios para los proyectos de los estudiantes. También se dan unos fundamentos sobre geodesia, cartografía y servicios web geográficos.

Hay cinco prácticas de laboratorio dedicadas a los sistemas de información geográfica. Los estudiantes deben resolver un conjunto de tareas mientras aprenden los fundamentos de algunas herramientas de este tipo de sistemas. Algunas de esas herramientas y algunos de los procesos aprendidos durante las prácticas son necesarios para procesar los datos y configurar algunos servicios web necesarios para el proyecto de la asignatura.

#### 3.2. Resultados de aprendizaje y evaluación

Además de cumplir con los requisitos acordados al principio del curso para cada proyecto, los resultados

de aprendizaje esperados se agrupan bajo tres categorías:

- Aplicar buenas prácticas de ingeniería del software. Por ejemplo, los estudiantes deben usar un sistema de control de versiones distribuido (Git en Github), pruebas automáticas y generar documentación de diseño apropiada.
- Aplicar patrones y conceptos del diseño dirigido por el dominio.
- Manipular datos y configurar herramientas de sistemas de información geográfica. Por ejemplo, los estudiantes tienen que configurar un servicio web de mapas con los edificios del campus.

La asignatura también permite que los estudiantes trabajen diversas competencias transversales, especialmente la capacidad para concebir, diseñar y desarrollar proyectos de ingeniería, la capacidad para planificar, organizar y controlar tareas, personas y recursos y la capacidad para trabajar en equipo la discusión y toma razonada de decisiones.

La evaluación de los estudiantes se basa principalmente en los proyectos (75 % de la nota final), con un examen pequeño, opcional, que proporciona el 25 % restante. Completar y entregar a tiempo lo que se pide en las prácticas de laboratorio proporciona hasta 0.5 puntos adicionales (sobre una nota máxima de 10). Hay una evaluación a mitad del cuatrimestre, de carácter formativo, que ofrece la oportunidad de dar realimentación y hacer sugerencias a los equipos para que mejoren sus proyectos hasta el final del curso.

#### 3.3. Proyectos

Los estudiantes tienen que desarrollar una aplicación móvil de tipo “campus inteligente” como proyecto en equipo siguiendo la metodología ágil Scrum. Tienen que procesar los datos de los edificios del campus, configurar una base de datos espacial y servicios web geográficos, y desarrollar un sistema cliente-servidor con un cliente que pueda funcionar en un teléfono inteligente que use estos datos y proporcione funcionalidad de su elección, como cálculo de rutas, localización de lugares, acceso a los menús de la cafetería, o control de distintos sistemas en los edificios. Los equipos están formados por entre 3 y 5 personas dependiendo del número de matriculados, y se permite a los estudiantes que los formen como quieran. Se establecen para todos unos objetivos de carácter general, que van variando ligeramente de año en año, pero cada grupo tiene que concretar y proponer unos requisitos particulares para su proyecto.

## 4. Método

Los estudiantes han contestado a un cuestionario en los cursos 2014-15, 2015-16 y 2016-17, que se diseñó para que expresaran su visión sobre el grado de éxito, o fracaso, en alcanzar autenticidad con la aproximación a la enseñanza-aprendizaje elegida, y también sobre otros aspectos de la asignatura. El cuestionario contiene preguntas abiertas y cerradas. Las cerradas buscan proporcionar una visión rápida, cuantitativa, mientras que las abiertas proporcionan la oportunidad de obtener una comprensión más profunda.

Las preguntas cerradas (Q1 a Q10) tienen respuestas que siguen una escala de Likert con cinco opciones desde “1 - Muy en desacuerdo” hasta “5 - Muy de acuerdo”. Entre las cuestiones abiertas, dos clarifican y expanden una respuesta a una cerrada (la Q4.1 y la Q10.1), y las otras tres (Q11, Q12 y Q13) son más explorativas y permiten que los estudiantes proporcionen opiniones sobre el curso (lo mejor, lo peor, e ideas para mejorarlo). Las 15 preguntas se muestran en el cuadro 4.

La literatura ha propuesto distintas características de la autenticidad y los entornos de aprendizaje basados en proyectos (ver la sección 2). Hay diferencias, y muchos de los términos no son universales, pero las preguntas Q1 a la Q9 intentan cubrirlos: los problemas no están bien especificados —algunos incluso son desconocidos— (Q3); las soluciones no son predecibles y son originales (Q3); puede haber varias soluciones (Q3) y requerir la integración de distintos dominios (Q4), además de que requieren conocimiento de dominio y estratégico (Q4); los problemas son interesantes para los estudiantes (Q1) y abordan temas actuales (Q2); el aprendizaje ocurre en grupos pequeños bajo la guía de un tutor (esto es por diseño de la asignatura); los proyectos son cercanos a la realidad profesional (Q5, Q6, Q7), están orientados hacia la aplicación de conocimiento (Q4), requieren la gestión de recursos y la diferenciación de roles entre los estudiantes (Q8), la toma de decisiones en contextos prácticos (Q9) y el uso de datos reales (Q6, Q7).

La encuesta se lleva a cabo después de la presentación final de los proyectos de los estudiantes, y por tanto después de que la asignatura se ha completado aunque antes de la evaluación, que es cuando es más probable tener a todos los estudiantes en el aula. Los cuestionarios se distribuyen impresos y se rellenan a mano.

Se han considerado algunas amenazas a la validez de los resultados. Lo primero es que no hay muchos estudiantes. Como esto se sabe por adelantado, se incluyeron preguntas abiertas que permitirían un análisis cualitativo para completar al cuantitativo. Además se han tomado encuestas de tres años

|      |   |
|------|---|
| 1    | El problema abordado en el proyecto de la asignatura me resulta de interés.   |
| 2    | El problema abordado en el proyecto de la asignatura me parece actual.  |
| 3    | Hay distintas formas de cumplir los requisitos del proyecto (lo que por ejemplo me permite diferenciar mi solución de las de otros equipos, tanto de este curso como de cursos anteriores). |
| 4    | El proyecto requiere que aplique e integre conocimientos y capacidades obtenidos en varias asignaturas de la titulación.  |
| 4.1  | Si estás de acuerdo en algún grado (has respondido 4 o 5), ¿puedes indicar de qué asignaturas?  |
| 5    | Lo aprendido en esta asignatura será útil en mi vida profesional.   |
| 6    | El proyecto es realista desde un punto de vista profesional .   |
| 7    | El trabajo en esta asignatura se parece más al tipo de trabajo que tendré que desempeñar como ingeniero en informática que la asignatura típica de la titulación.                           |
| 8    | La asignatura requiere trabajar en equipo, gestionar tiempo y otros recursos y repartir roles entre los miembros de este equipo .   |
| 9    | El proyecto requiere tomar decisiones en el marco de un contexto práctico.  |
| 10   | Recomendaría a futuros estudiantes que cursaran esta asignatura.  |
| 10.1 | ¿Por qué?   |
| 11   | ¿Cómo crees que se podría mejorar la asignatura?  |
| 12   | ¿Qué ha sido para ti lo peor de la asignatura?  |
| 13   | ¿Qué ha sido para ti lo mejor de la asignatura?   |

Cuadro 1: El cuestionario

consecutivos. Para evitar sesgos, los autores de este artículo no involucrados directamente en la impartición de la asignatura han revisado independientemente los resultados y las conclusiones extraídas. Finalmente, las respuestas han sido digitalizadas y están disponibles en <https://doi.org/10.6084/m9.figshare.5769726> para que cualquiera pueda examinarlas.

## 5. Resultados

En esta sección se analizan los resultados recopilados en los años 2014-15, 2015-16 y 2016-17 —ver las

| Número de respuestas |       |       | Asignaturas          |
|----------------------|-------|-------|----------------------|
| 14-15                | 15-16 | 16-17 |                      |
| 4                    | 5     | 5     | Gestión de proyectos |
| 5                    | 3     | 6     | Diseño de software   |
| 5                    | 7     | 4     | Tecnologías web      |
| 1                    | 3     | 0     | Bases de datos       |

Cuadro 2: Asignaturas previas relacionadas

figuras 1, 2 y 3—.

Q1 y Q2 están relacionadas con el interés y actualidad del proyecto elegido. La mediana de ambas en los tres años es “4 - De acuerdo” y solo un estudiante, el segundo año, y dos, el tercero, muestran algún desacuerdo.

Q3 y Q4 se relacionan con la naturaleza poco estructurada de los problemas abordados en el proyecto. La mediana para Q3 es “4 - De acuerdo” los primeros dos años y “3 - Ni de acuerdo ni en desacuerdo” el tercero. La mediana para Q4 es “5 - Muy de acuerdo” el primer año y “4 - De acuerdo” el segundo y el tercero. Solo un estudiante muestra desacuerdo con Q3 el segundo año y otro el tercero. No hay estudiantes en desacuerdo con Q4.

En 2014-15 todos los estudiantes estuvieron de acuerdo con Q4, en 2015-16 fueron siete y en 2016-17 los seis, así que se les pidió que respondieran a Q4.1 (nombrar asignaturas previas que habían necesitado). Las respuestas se han agrupado en cuatro categorías<sup>2</sup> que se muestran en el cuadro 5. La variedad de las respuestas confirma que esta es una asignatura que requiere integrar conocimientos de otras asignaturas anteriores.

Las preguntas de la Q5 a la Q9 están relacionada con la naturaleza basada en proyectos del entorno de enseñanza-aprendizaje. La mayoría de los estudiantes están de acuerdo con esas preguntas, sobre la utilidad profesional de lo aprendido, el realismo del trabajo y el tipo de aproximación basada en equipo requerida.

La mediana de Q10 es “5 - Muy de acuerdo” el primer año, “4 - De acuerdo” el segundo y “3 - Ni de acuerdo ni en desacuerdo” el tercero. Esto muestra que la mayoría de los estudiantes recomendarían el curso a otros, y solo uno en los tres años indica que definitivamente no lo haría. Las respuestas a Q10.1 nos dan una idea de por qué la recomendarían:

- El realismo y la naturaleza práctica de la asignatura se señalan por tres estudiantes el primer año (“una aplicación completa”, “cercano a un pro-

yecto real”, “lejos del típico marco teórico”, “permite probar conceptos aprendidos en otras asignaturas”) y por dos en el segundo (“proyecto realista, cercano al mundo profesional”, “dominio y características reales y actuales”).

- El trabajo en equipo se menciona por dos estudiantes el primer año (“gestión del trabajo en equipo”, “el éxito depende de la organización”) y por tres el segundo (“aprendes otros métodos para afrontar un proyecto”, “gestionar un proyecto entre varios”, “un proyecto más o menos grande [...] y la distribución del trabajo”). Hay una respuesta interesante el tercer año donde un estudiante indica que la recomendaría solo si se va a tener un buen equipo.
- El resto de las respuestas se relaciona con lo nuevo que aprenden. El primer año se mencionan tanto los sistemas de información geográfica como el diseño dirigido por el dominio, y también la posibilidad que tienen de elegir tecnologías. En el segundo y tercer año las respuestas son más diversas, pero de nuevo relacionada con eso y con los métodos de diseño de software que aplican (p.ej. mencionan explícitamente la profundización en el diseño orientado a objetos o la arquitectura hexagonal).

Solo uno de los estudiantes que no la recomendaría (respuesta de 3 o menos) señala una razón, que es que algunos estudiantes pueden que no hayan cursado algunas asignaturas previas que pueden serles necesarias para seguir adecuadamente esta.

La Q11 pide a los estudiantes sugerencias para mejorar la asignatura. En 2014-15 las cinco respuestas eran sobre dos temas: más práctica y menos teoría, y más estudiantes. En 2015-16 y 2016-17 se quitaron algunas partes de la teoría y se hicieron más problemas. Las respuestas estos años son más variadas, pero con dos temas comunes: una mayor integración de las prácticas de laboratorio, las clases de teoría-problemas y el proyecto, y más libertad para elegir tecnologías. El resto de las respuestas son diversas: dos estudiantes sugieren seguir aún más de cerca el trabajo de los grupos, otro pide traer a expertos externos y otro incluir algunos casos de estudio reales de diseño dirigido por el dominio. También se piden más ejercicios, y un estudiante quiere un trabajo que permite aún una mayor variedad de soluciones.

La Q12 pide a los estudiantes que señalen lo peor de la asignatura. En 2014-15 dos estudiantes contestan que lo peor está relacionado con la teoría (“demasiada teoría”, “algunas de las lecciones de teoría”), uno cree que el “comienzo es lento”, otro contesta que su propia gestión del tiempo y el otro la deja en blanco. En 2015-16 y 2016-17 el tema del comienzo lento aparece en tres respuestas y las clases de teoría en una, pero pa-

<sup>2</sup>Hay más de cuatro asignaturas en las respuestas, pero las diferencias son sutiles en algunos casos (p.ej. “Ingeniería Web” o “Sistemas y Tecnologías Web”) o son versiones especializadas unas de otras en otros (p.ej. “Ingeniería de Requisitos” o “Arquitectura Software” frente a la más general “Ingeniería del Software”).

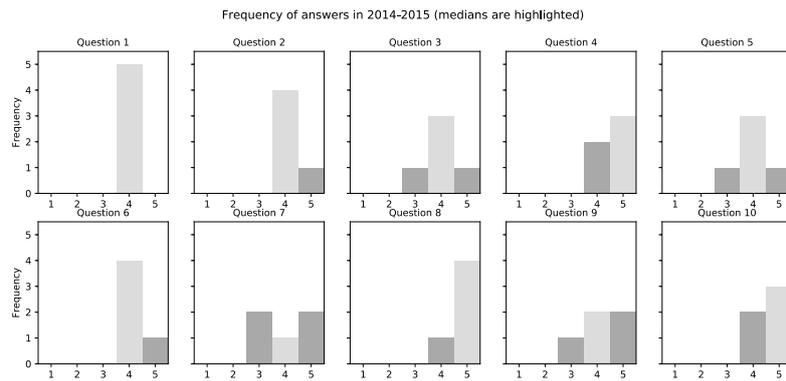


Figura 1: Resultados de la encuesta 2014-15 para las preguntas 1-10

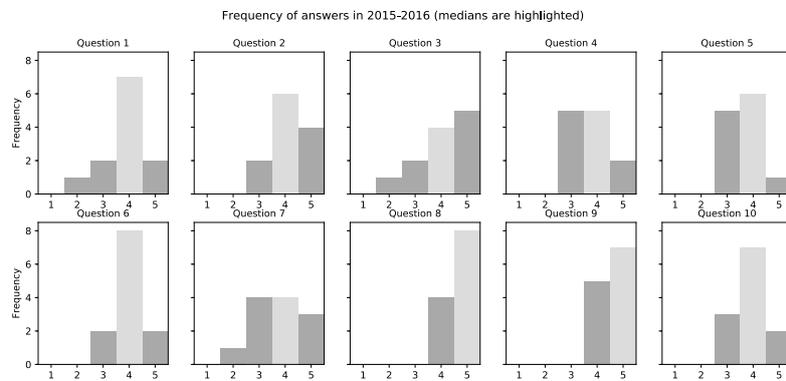


Figura 2: Resultados de la encuesta 2015-16 para las preguntas 1-10

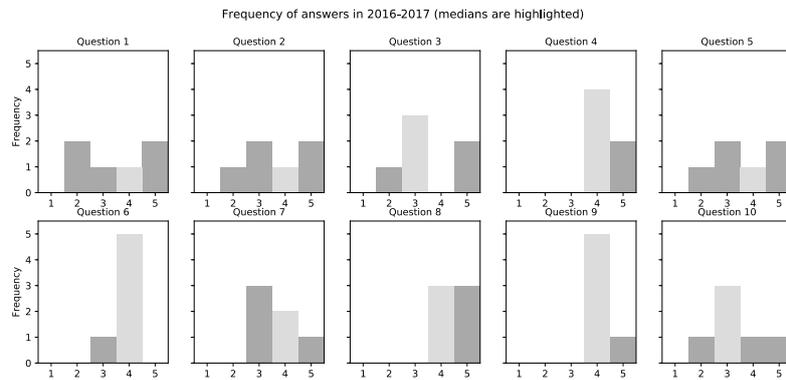


Figura 3: Resultados de la encuesta 2016-17 para las preguntas 1-10

rece que lo que menos gusta es lo relacionado con los sistemas de información geográfica, con hasta seis estudiantes señalando eso en 2015-16 y tres en 2016-17. También hay dos respuestas diciendo que los proyectos son demasiado parecidos entre sí.

En Q13 los estudiantes señalan lo mejor de la asignatura. En 2015-15 tres estudiantes mencionan las

prácticas de laboratorio, cuatro el trabajo en equipo y uno las clases de diseño dirigido por el dominio. En 2015-16 las respuestas son variadas; aprender más sobre modelizado de dominio y diseño dirigido por el dominio aparece cuatro veces, el proyecto y el trabajo en equipo también varias veces pero con distintos matices — el proyecto y el trabajo en equipo como tales, que

el proyecto sea realista, que se puede completar en el tiempo disponible, la libertad para llevarlo a cabo, que el profesor sigue de cerca el desarrollo del mismo y que las clases de problemas también lo siguen de cerca. Las prácticas de laboratorio son lo mejor para dos estudiantes y otro menciona lo relacionado con la información geográfica. Finalmente, cuatro estudiantes señalan que aprenden más sobre diseño de software. En 2016-17 la oportunidad de aprender más sobre arquitectura del software y diseño dirigido por el dominio aparecen en cinco de las respuestas, y un estudiante menciona explícitamente “un proyecto que podría ser un caso real”.

## 6. Lecciones aprendidas

Analizar las respuestas de los estudiantes ha permitido extraer algunas lecciones que pueden ser de interés para quienes preparen asignaturas sobre desarrollo de software en equipo<sup>3</sup>.

Los estudiantes perciben el proyecto como realista (Q6), cercano a la práctica profesional (Q7) y útil para su vida profesional (Q5). Cinco estudiantes recomendarían el curso por el realismo del proyecto (Q10.1) y “un proyecto real” es lo mejor del curso para tres (Q13). Sin embargo, el realismo tiene un coste. Por ejemplo, los datos geográficos reales son problemáticos (nulos inesperados, errores de geoposicionamiento, inconsistencias geométricas) y prepararlos cuesta tiempo. Varios estudiantes han señalado eso como lo peor del curso (Q12). Esto sugiere que mucha autenticidad puede ser demasiada y que o se proporciona ayuda extra con las tareas que cuestan más tiempo (p.ej. dándoles datos preprocesados), o que estas actividades deberían valorarse más en las notas.

La competición también es algo a considerar. Varios estudiantes han sugerido que quieren una oportunidad para diseñar mejores aplicaciones que los otros equipos. En general aceptan que hay distintas formas de cumplir los requisitos (Q3), pero sugieren que más estudiantes harían el trabajo más desafiante, y que les gustaría un trabajo que permitiera más variedad de soluciones (Q11). De hecho, dos han señalado entre lo peor que los proyectos son demasiado similares (Q13). Esto es importante en una asignatura donde el dominio de problema es el mismo para todos. Proporcionar pues opciones suficientes para que los estudiantes puedan hacer sus proyectos únicos valdría el esfuerzo.

Otro tema relacionado con la naturaleza del curso, diseño de software en un dominio específico, es el contraste entre las competencias relacionadas con el domi-

nio y las de ingeniería del software. Lo primero es señalar que disfrutan aprendiendo nuevas cosas: seis estudiantes recomendarían el curso por lo que se aprende sobre información geográfica, mientras que hasta ocho mencionan habilidades relacionadas con la ingeniería del software (Q10.1). Sin embargo, diez incluyen el dominio geográfico entre lo peor del curso, mientras que ninguno se queja de aprender más sobre diseño de software o tecnología (Q12). Este patrón vuelve a emerger en Q13, sobre lo mejor del curso, donde muchos estudiantes mencionan el modelizado del dominio, el diseño dirigido por el dominio y el diseño de software, mientras que solo uno señala el dominio geográfico.

Si el patrón sugerido en el párrafo anterior es específico del dominio elegido o no, no está claro. Las respuestas a Q1 y Q2 muestran que el proyecto lo encuentran interesante y actual, y en general recomendarían el curso a otros (Q10). Hay también unas cuantas respuestas en la que las partes relacionadas con la información geográfica se resaltan como positivas. Considerando todo esto, este contraste parecería probablemente con otros dominios de problema. Esto debería considerarse al preparar un curso de este tipo: a igualdad de otros aspectos, un dominio que los estudiantes ya conozcan podría ser una mejor alternativa que uno del que desconocen lo más básico.

Otro punto interesante es que, en general, les gustan las prácticas de laboratorio. Cinco estudiantes las mencionan entre lo mejor del curso (Q13) y ninguno entre lo peor (Q12), aunque cuatro sugieren integrarlas mejor con el proyecto (Q11). Esto sugiere que en una asignatura de aprendizaje basado en proyectos sigue habiendo sitio para prácticas “tradicionales”, y que esto funcionará mejor si el trabajo en esas prácticas ayuda a los estudiantes con los proyectos.

## 7. Conclusiones

Este artículo ha presentado un estudio sobre los aspectos relacionadas con la autenticidad en una asignatura de ingeniería del software que sigue una metodología de aprendizaje basado en proyectos y está especialmente focalizada en los aspectos relacionados con el dominio de problema en el diseño de software. El estudio analiza los resultados de una encuesta realizada por los estudiantes en tres años consecutivos, y concluye que en general los estudiantes han encontrado “autenticidad” en la asignatura, que era un objetivo de la misma. Las respuestas nos han proporcionado mayor comprensión sobre la visión de los estudiantes, lo que hemos plasmado como lecciones aprendidas en sección 6.

El proyecto es sobre sistemas de información geográfica aplicados a un campos universitario. Esto ha permitido que los estudiantes trabajen en algo relacio-

<sup>3</sup>Aparte de lo que se discute en esta sección, los estudiantes comentan algunos problemas que tienen mala solución, como el hecho de que haya pocos alumnos, y señalan algunas cosas que en general eran de esperar, como que en general disfrutaban con el trabajo en equipo, o que les gusta la libertad para elegir objetivos o tecnologías.

nado con sus problemas e intereses, algo que ha sido relacionado con la autenticidad en la literatura. Como ejemplos de esto, algunos equipos han desarrollado funcionalidad de cálculo de rutas en interiores para guiar a alguien hasta un laboratorio o aula, otros han incluido los aparcamientos del campus y un sistema, simulado, para saber dónde hay plazas libres, y otros han desarrollado funcionalidad para determinar el grado de ocupación de los laboratorios y las cafeterías.

Nuestra experiencia al combinar el diseño dirigido por el dominio y el aprendizaje basado en proyectos ha sido positiva y los estudiantes han respondido bien. No sabemos si el dominio de problema elegido ha influido, para bien o para mal, en estos resultados. La experiencia de otros cursos similares sería necesaria para proporcionar una visión más clara de este tema.

En un futuro, continuaremos trabajando en incorporar las sugerencias más comunes, como la de una mayor integración entre las prácticas de laboratorio y el proyecto. También continuaremos buscando nuevas ideas para que los estudiantes tengan más donde elegir y así poder llevar a cabo proyectos más variados, con mayor competición y diferenciación entre ellos.

## Referencias

- [1] ACM/IEEE-CS Joint Task Force on Computing Curricula. Computer science curricula 2013. Technical report, ACM Press and IEEE Computer Society Press, December 2013.
- [2] Howard S. Barrows. Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, 1996(68):3–12, 1996.
- [3] Ching-Huei Chen. Reframing narrative cases for ill-structured contexts: The design with learners in mind. *Journal of Learning Design*, 3(1), 2009.
- [4] Nada Dabbagh and Susan Dass. Case problems for problem-based pedagogical approaches: A comparative analysis. *Computers & Education*, 64:161–174, 2013.
- [5] Marian Daun, Andrea Salmon, Thorsten Weyer, and Bastian Tenbergen. Project-based learning with examples from industry in university courses: An experience report from an undergraduate requirements engineering course. In *Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, pages 184–193, 2016.
- [6] Peter Dolog, Lone Leth Thomsen, and Bent Thomsen. Assessing problem-based learning in a software engineering curriculum using bloom's taxonomy and the IEEE Software Engineering Body of Knowledge. *ACM Transactions on Computing Education*, 16(3):9:1–9:41, 2016.
- [7] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, 2003.
- [8] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.
- [9] Woei Hung, David H. Jonassen, and Rude Liu. Problem-based learning. In J. Michael Spector, M. David Merrill, Jeroen van Merriënboer, and Marcy P. Driscoll, editors, *Handbook of research on educational communications and technology*, pages 485–506. Routledge, third edition, 2008.
- [10] David Jonassen, Johannes Strobel, and Chwee Beng Lee. Everyday problem solving in engineering: Lessons for engineering educators. *Journal of Engineering Education*, 95(2):139–151, 2006.
- [11] David H. Jonassen. Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology Research and Development*, 45(1):65–94, 1997.
- [12] Billy Vaughn Koen. *Discussion of the Method. Conducting the Engineer's Approach to Problem Solving*. Oxford University Press, 2003.
- [13] José A. Macías. Enhancing project-based learning in software engineering lab teaching through an e-portfolio approach. *IEEE Transactions on Education*, 55(4):502–507, 2012.
- [14] J. C. Perrenet, P. A. J. Bouhuijs, and J. G. M. M. Smits. The suitability of problem-based learning for engineering education: Theory and practice. *Teaching in Higher Education*, 5(3):345–358, 2000.
- [15] Henry Petroski. *An Engineer's Alphabet. Gleanings from the Softer Side of a Profession*. Cambridge University Press, 2011.
- [16] Jungwoo Ryoo, Frederico Fonseca, and David S. Janzen. Teaching object-oriented software engineering through problem-based learning in the context of game design. In *Proceedings of the IEEE 21st Conference on Software Engineering Education and Training*, pages 137–144, 2008.
- [17] J. Strobel, J. Wang, N.R. Weber, and M. Dyehouse. The role of authenticity in design-based learning environments: The case of engineering education. *Computers & Education*, 64:143–152, 2013.
- [18] Elise van Dooren, Els Boshuizen, Jeroen van Merriënboer, Thijs Asselbergs, and Machiel van Dorst. Making explicit in design education: generic elements in the design process. *International Journal of Technology in Design Education*, 24:53–71, 2014.