

# Revisión de software entre pares y reutilización de código en la enseñanza de desarrollo de software web

Rosa Arruabarrena, José Ángel Vadillo, José Miguel Blanco  
Facultad de Informática. Departamento de Lenguajes y Sistemas Informáticos  
UPV/EHU

rosa.arruabarrena@ehu.eus      ja.vadillo@ehu.eus      josemiguel.blanco@ehu.eus

## Resumen

El Aprendizaje Basado en Proyectos (ABP) es una metodología ampliamente empleada cuya valía está reconocida en la docencia de Ingeniería del Software. La combinación de ABP y un enfoque incremental, con entregas semanales que extienden las funcionalidades de un sitio web, ha venido sustentando la propuesta docente empleada en la asignatura Sistemas Web de un Grado en Ingeniería Informática. Los laboratorios se desarrollan, entregan y se califican en plazos de una semana (excepcionalmente, dos). Los resultados obtenidos en un laboratorio sirven como base para el siguiente. En cursos precedentes hemos constatado que aproximadamente un tercio de los grupos tiene dificultades para seguir el ritmo de la asignatura, siendo uno de los motivos principales que acumulan problemas derivados de sus decisiones de diseño y programación en laboratorios previos. En este artículo presentamos una propuesta que permite realinear los grupos con dificultades, para que puedan finalizar su proyecto en evaluación continua con éxito y con un desarrollo de mayor calidad. La revisión de código por pares y la posibilidad de intercambio de código entre compañeros son la base de la experiencia realizada. Completamos este trabajo describiendo la experiencia de implantación de la propuesta, el proceso de evaluación, los resultados obtenidos, así como las reflexiones y conclusiones derivadas.

## Abstract

Project Based Learning (PBL) is a widely used methodology whose value is recognized in Software Engineering teaching. The combination of PBL and an incremental approach, with weekly deliveries that extend the functionalities of a website, has been supporting the teaching proposal used in the subject Web Systems of a Degree in Computer Engineering. The laboratories are developed, delivered and assessed every week (exceptionally, within two weeks) and the output obtained in a laboratory is the

input for the next one. In previous courses, we observed that approximately one third of groups had difficulties in following the rhythm of the subject, being the main weakness the accumulation of problems derived from their design decisions and programming in previous laboratories. In this article, we present a proposal that allows realigning groups with difficulties, so that they can finish their project in continuous evaluation successfully and with a development of higher quality. The *peer review* and the possibility of code exchange between peers are the basis of the experience. We complete this work describing the experience of implementing the proposal, the evaluation process, the results obtained, as well as the reflections and conclusions derived.

## Palabras clave

Aprendizaje Colaborativo, Aprendizaje Basado en Proyectos, Evaluación entre pares e Ingeniería del Software.

## 1. Introducción

En los últimos años se busca conseguir que los estudiantes asuman un rol más dinámico en todo el proceso educativo, que sean agentes activos a la hora de configurar su educación [2, 11]. Para ello se han ido introduciendo diferentes metodologías activas de enseñanza, como el aprendizaje basado en proyectos, aprendizaje basado en problemas, aula invertida, valoración por pares, etc. [5, 7, 9, 13].

Si bien los autores ya han venido empleando diversas estrategias de aprendizaje con éxito (en términos de tasas de rendimiento y satisfacción del alumnado) en la asignatura Sistemas Web del Grado en Ingeniería Informática de la UPV/EHU [14], también hemos constatado las dificultades de cierto número de alumnos para amoldarse a nuestro modelo pedagógico.

En este trabajo presentamos una propuesta que integra el uso activo de revisiones por pares dentro de un contexto ABP de desarrollo de software para la

web con el objetivo principal de facilitar con éxito la continuidad de nuestros estudiantes en evaluación continua. Presentamos el contexto académico de la experiencia y su motivación, para pasar a describir con detalle la innovación docente introducida por primera vez en la asignatura. El objetivo principal es evitar abandonos por parte de los estudiantes, a la par que reforzar otras competencias propias de la disciplina de la Ingeniería del Software. Seguidamente exponemos los resultados obtenidos, las reflexiones y las conclusiones extraídas de la experiencia implantada.

## 2. Trabajo relacionado

El aprendizaje basado en proyectos (ABP) (*Project Based Learning*) es una estrategia de aprendizaje centrada en el estudiante en el que el eje de la propuesta docente es la planificación, desarrollo y evaluación de proyectos. Su uso se ha ido extendiendo en la última década en la enseñanza de la Ingeniería Informática [1, 4, 6, 7, 8, 13, 15]. Como toda estrategia de aprendizaje, el ABP tiene sus ventajas y riesgos. La literatura recoge beneficios, como son la posibilidad de conectar el aprendizaje con la realidad, el incremento de la motivación, la potenciación del trabajo colaborativo, y la construcción y el control del conocimiento personal. También se han identificado objeciones entre las que cabe resaltar un aumento importante de la carga de trabajo para todos los implicados [15].

La valoración por pares o entre iguales (*Peer Assessment o Peer Review*) es el proceso en el cual los estudiantes son evaluados o evalúan el desarrollo del trabajo o los logros de sus *peers* o iguales (habitualmente compañeros de asignatura). La valoración puede tener en cuenta el nivel, el mérito, la calidad o el éxito de los productos o los resultados del aprendizaje de compañeros en situaciones análogas [12]. El objetivo puede ser formativo o no. Puede hacerse de forma cuantitativa o cualitativa, con o sin *feedback*. Todos los estudiantes pueden ser evaluadores y evaluados, o bien solo algunos evaluar a todos o parte de sus compañeros, y puede realizarse de forma anónima o no. El interés por el *peer-* y el *self-review* ha crecido en los últimos años como método para fomentar la colaboración entre estudiantes y para promover una mayor implicación en el proceso de aprendizaje-enseñanza [3, 8, 10, 16].

## 3. Contexto académico de la experiencia

Entre las líneas de especialidad del Grado de Ingeniería Informática de la UPV/EHU está la de Ingeniería del Software. La asignatura Sistemas Web (SW) es una de las asignaturas obligatorias (de seis créditos)

de esta rama y, también, puede seguirse como optativa de las otras líneas curriculares. La matrícula en SW varía dependiendo de la distribución en especialidades de los alumnos y del número de estudiantes migrante de otros centros y universidades. En los últimos cursos el número de estudiantes matriculados se ha situado en la horquilla 20 a 36 por cada grupo (se imparte uno en euskera y otro en castellano). En todo caso, es una asignatura con una demanda estable, y altamente valorada por los estudiantes, tanto por sus contenidos como por su enfoque didáctico. Aunque forma parte de la especialidad Ingeniería del Software, es suficientemente autocontenida como para que la pueda cursar cualquier estudiante interesado.

La asignatura está diseñada en base a la metodología ABP con el objetivo de que el estudiante adquiera las competencias construyendo, desde una maqueta inicial, un sitio web que tendrá algunas de las características relevantes de los sistemas web actuales. Este proceso se canaliza en el proyecto de la asignatura, que tiene un peso del 70% en la calificación final. Los estudiantes desarrollan un sitio web, cuyo dominio/temática es común para todos ellos, de forma incremental y con entregas semanales [14].

El trabajo práctico se realiza en grupos de dos estudiantes, con dominio/temática común, y se desarrolla de forma incremental. Se parte de un proyecto inicial sin apenas funcionalidad, salvo algún enlace y el estilo implementado con CSS. Semanalmente los estudiantes dotan de una funcionalidad adicional específica al proyecto en desarrollo, para lo cual descubren los conceptos y la tecnología web, y los conocimientos que sean pertinentes en las dos sesiones previas a la sesión de laboratorio. Las tareas semanales a acometer se descomponen en varias etapas/fases más simples, normalmente dependientes entre sí, y mientras que las primeras fases son obligatorias para todos los alumnos, las últimas, que son optativas y más avanzadas, sirven para obtener mejores calificaciones. Estos dos últimos cursos, la descripción de los requisitos de las tareas se ha completado con vídeos que ilustran los casos de uso más significativos. El *feedback* de los desarrollos recogidos se devuelve en 2-3 días, de manera que cada grupo está al corriente de los aspectos que debe mejorar antes de la próxima entrega. El trabajo práctico se extiende a lo largo de 13 semanas. Las tareas semanales se describen en once enunciados distribuidos semanalmente. Estos enunciados se plasman en nueve entregas parciales por parte de los estudiantes. Hay una correspondencia directa entre la secuenciación de las funcionalidades con que se va dotando al sitio web con el proceso de desarrollo centrado en una arquitectura de tres capas (nivel de presentación, lógica de negocio y acceso a datos). Esto permite estructurar el código del proyecto con unas directrices

comunes a todos los estudiantes e independizar los aspectos de presentación de la lógica de negocio y del acceso a datos.

El último enunciado tiene asociado un mayor alcance y duración temporal (tres semanas), ya que en él los alumnos pueden efectuar otras caracterizaciones más avanzadas de su sitio web a fin de optar a una calificación más elevada. El trabajo práctico tiene dos hitos intermedios: uno finalizado el laboratorio L4 (final de la semana 6) y el otro tras el laboratorio L7 (final de la semana 10). En esos momentos, es condición *sine qua non* que los requisitos de todas las tareas obligatorias de cada uno de los enunciados previos se satisfagan para poder progresar en la ampliación de las subsiguientes funcionalidades.

## 4. Motivación

A pesar de que la asignatura tiene una alta tasa de éxito (rondando el 95% en las tres últimas promociones) y logra resultados muy positivos en las encuestas de satisfacción realizadas por la universidad, aproximadamente un tercio de los estudiantes que cursa SW tiene problemas para realizar el proyecto en evaluación continua. Además, se presentan desviaciones muy notables respecto al nivel conseguido entre los estudiantes que alcanzan los mínimos exigidos y aquellos que realizan trabajos excelentes.

Una de las causas principales de la diversidad de logros y dificultades de los estudiantes es la heterogeneidad de sus perfiles de entrada, la cual se multiplica al trabajar en grupos. Esta diversidad va en aumento debido a la tendencia creciente en el número alumnos que han entrado a tercero mediante traslado de expediente. Así, nos encontramos en el aula con diferentes perfiles curriculares: estudiantes repetidores de segundo que la cursan para adelantar asignaturas de tercero, estudiantes de tercero nuevos en el centro que provienen de otros Grados, estudiantes de tercero (o cuarto) que hacen la especialidad de Ingeniería del Software y estudiantes de cuarto que cursan otras especialidades. Esta situación hace que, desde el principio, los grupos de estudiantes trabajen a diferentes velocidades el proyecto central de la asignatura, pudiéndose clasificar en cuatro grupos de estudiantes:

1. Los que llegan a completar con dificultad y esfuerzo las tareas obligatorias mínimas y, a menudo, tienen problemas para poder alcanzar los siguientes hitos.
2. Los que completan las tareas obligatorias, pero no pueden realizar las opcionales.
3. Los que realizan todas las tareas obligatorias y parte, o la mayoría, de tareas opcionales complementarias.

4. Los que rebasan el marco propuesto con carácter general y definen objetivos propios de excelencia más allá de la definición común.

En consecuencia, no todos los que empiezan en evaluación continua son capaces de superar la asignatura por esta vía, ya que les resulta complicado mantener el ritmo de entregas y compatibilizarla con las otras asignaturas. Además, un número significativo de estudiantes no pueden plantearse retos por encima de los mínimos, entre otras razones por arrastrar problemas en la solución que han ido progresivamente desarrollando.

Con todo ello, nuestro objetivo al introducir una innovación metodológica es que los niveles de satisfacción se mantengan y la tasa de éxito en ningún caso empeore. Además, que los estudiantes puedan alcanzar mayores niveles de competencia e integración de los conocimientos dentro del ámbito de la Ingeniería del Software, concretamente en lo relativo a calidad del software, modularización y reusabilidad.

Frente al enfoque seguido anteriormente, donde los diversos grupos de estudiantes desarrollaban en paralelo su código y no compartían (al menos de manera regulada por el profesorado) entre ellos sus propuestas de solución, hemos decidido apostar porque las soluciones desarrolladas por los diversos grupos de estudiantes sean compartidas, analizadas, valoradas y, llegado el caso, integradas en su desarrollo. Eso sí, de forma coordinada y enmarcada en unos hitos de seguimiento y control del proyecto de la asignatura.

## 5. Descripción de la innovación implantada

La idea en que se basa la propuesta es que los estudiantes accedan, revisen, valoren y reutilicen, si lo desean, el código que plasma las soluciones desarrolladas por sus compañeros a través de un proceso transparente, incluido en la planificación del proyecto y coordinado por el profesorado de la asignatura. Este proceso es parte explícita del modelo de desarrollo de software del proyecto, e incluye aspectos de ingeniería del software: validación funcional (en este caso, entre iguales), construcción por capas y compartición/reutilización de código.

### 5.1. Validación entre iguales

En la asignatura perteneciente al área de Ingeniería del Software, concretamente al desarrollo web, las valoraciones entre iguales se centran en la validación de software y se realizan sólo con fin formativo. Hemos optado por valoraciones no anónimas, proporcionando un *feedback* público de todos los resultados obtenidos en el proceso. Como se describirá más adelante se plantean dos revisiones con objetivos

diferentes. En la primera revisión todos los estudiantes valoran una selección previa de aplicaciones web con el objetivo de fijar un desarrollo completo y correcto que será candidato a ser reutilizado por otros estudiantes. En la segunda cada grupo valora los trabajos de tres grupos diferentes. El objetivo de esta valoración es identificar errores y mejoras para que cada grupo pueda finalizar sus proyectos de forma correcta, y por tanto mejorar su entrega final. Los profesores juegan un rol de supervisión de la coherencia de las validaciones y publicación de los resultados finales. Paralelamente, es el equipo docente el responsable de la evaluación y calificación de cada una de las entregas [14].

## 5.2. Herramientas y arquitectura para reutilización y desarrollo web

Con nuestra propuesta pretendemos impulsar una reutilización de código revisado y validado previamente como soporte al desarrollo de nuevas funcionalidades. Hemos decidido abrir la posibilidad de ver y emplear soluciones alternativas de las mismas funcionalidades, con más o menos acierto, gusto, coste, usabilidad, que las desarrolladas por el propio estudiante. También como fuente de inspiración para posteriores funcionalidades. Además, intentamos minimizar el esfuerzo de realizar dicha integración.

Al utilizar una arquitectura común basada en capas, los ficheros que se van produciendo se estructuran siguiendo unas directrices compartidas por todos los grupos. Es clave también el uso de herramientas de desarrollo de software online (GitHub par mantenimiento de código y un servidor web [14]), facilitando no solo la gestión de entregas, su evaluación y despliegue, sino también la propia migración. Así mismo, el uso de nomenclatura común para ficheros y funcionalidades posibilita su intercambio de una forma más sencilla, es decir, facilita la exportación/importación de módulos de código, favoreciendo así su reutilización.

## 5.3. Planificación y desarrollo

Tal y como se ha mencionado anteriormente, se parte de un proyecto inicial sin apenas funcionalidad. Semanalmente (ver cuadro 1), los estudiantes incorporan funcionalidad adicional específica al proyecto en desarrollo, para lo cual se introducen los conceptos y la tecnología web, y los conocimientos que sean pertinentes en las dos sesiones de aula previas a la sesión de laboratorio. Las tareas semanales a acometer se descomponen en varias etapas/fases más simples, normalmente dependientes entre sí, siendo estas primeras fases obligatorias para todos los alumnos. Realizadas estas, y con carácter opcional, los alumnos pueden efectuar otras caracterizaciones más avanzadas de su sitio web a fin de optar a una calificación más elevada.

| Semana                           | Actividad                            | Tema          |
|----------------------------------|--------------------------------------|---------------|
| 1                                | Lab0                                 | Hosting       |
| 2                                | Lab1                                 | HTML y CSS    |
| 3                                |                                      | JavaScript    |
| 4                                |                                      | JQuery        |
| 5                                | Lab2                                 | Formularios   |
| 6                                |                                      | PHP y MySQL   |
| 7                                | Lab3<br><i>Primera Revisión</i>      | Seguridad     |
| <i>Posibilidad de reenganche</i> |                                      |               |
| 8                                | Lab4                                 | XML           |
| 9                                | Lab5                                 | Ajax          |
| 10                               | Lab6                                 | Servicios Web |
| 11                               |                                      |               |
| 12                               | Lab7<br><i>Segunda Revisión</i>      | Sesiones      |
| 13,14,15                         | <i>Ampliación final del proyecto</i> |               |

Cuadro 1: Cronograma de la asignatura.

### Primera revisión

El objetivo de la primera revisión entre iguales es valorar un conjunto común de tres proyectos web. La selección inicial de los proyectos la realiza el equipo docente, de forma que cumplan unos requisitos previamente establecidos de completitud, calidad y corrección. Además, la capa de presentación debe mantenerse fiel al proyecto web de partida. Concretamente, se considera el correcto funcionamiento de las partes obligatorias y ciertos aspectos relevantes dentro del área de desarrollo software y, en particular, de desarrollo web, como son aspectos relativos a los formularios e interfaces, navegación, *feedback*, gestión de imágenes, adecuada organización e identificación de los ficheros, gestión de roles, etc. Estos criterios de evaluación, empleados por el equipo docente en la selección de los sitios, son a su vez los que usan los alumnos validadores. La valoración se realiza mediante un cuestionario *online*. El proyecto web que obtenga mayor valoración se ofrecerá como código de reenganche a aquellos grupos que en las cuatro evaluaciones de las correspondientes entregas previas no hayan mostrado una calidad suficiente. Esta revisión se realiza por grupos y todos ellos evalúan los tres proyectos propuestos para el reenganche, más el suyo propio. Los estudiantes conocen de antemano el objetivo de la actividad, y reciben los

resultados en muy poco tiempo (en horas). Asimismo, a los grupos con desarrollos insatisfactorios se les invita a proseguir el trabajo práctico a partir del código con mejor calificación, al que tienen la opción de acogerse para la siguiente entrega del proyecto.

### Segunda revisión

El objetivo de la segunda revisión es validar las funcionalidades de los desarrollos de los compañeros, para identificar errores y mejoras, con vistas a mejorar la entrega final propia. Cada grupo revisa los proyectos de tres grupos que le son asignados específicamente por los profesores. De esta forma, cada grupo recibe valoraciones de tres grupos distintos. La valoración se realiza nuevamente a través de un cuestionario *online*<sup>1</sup>. En esta ocasión, las cuestiones se organizan en varios bloques: cuestiones relativas a la interfaz (navegación, facilidad de uso y validación de errores en los datos de entrada), a la lógica de negocio (corrección y completitud de las funcionalidades principalmente) y a la capa de datos (encapsulamiento de las funciones de acceso a datos y el modelo de datos principalmente).

En este caso el equipo docente juega un rol de supervisión de la coherencia de las validaciones y publicación de los resultados finales.

## 6. Resultados

En este apartado se mostrarán los resultados obtenidos en las actividades propuestas anteriormente y descritas dentro de la metodología docente. Además, se presentarán las valoraciones generales obtenidas de una encuesta anónima completada por los estudiantes al final del cuatrimestre. Estos resultados se han obtenido a partir de 30 respuestas obtenidas de los 42 alumnos en evaluación continua encuestados. Para finalizar este capítulo, se incluirán los resultados académicos en términos de calificaciones finales.

La primera actividad que se realiza es la selección de un proyecto candidato a ser reutilizado por otros grupos. Para esta selección se parte de un conjunto de proyectos previamente seleccionados que son valoradas por todos los estudiantes. También se pide a cada grupo que evalúe su propio proyecto con los mismos indicadores y escala (rango de 1 a 5).

Esta revisión permite comparar los resultados obtenidos por los proyectos de los demás frente a la valoración que cada grupo hace de su propio proyecto. Las figuras 1 y 2 recogen estos resultados. Nótese en la figura 2 que no se muestran las valoraciones de los grupos T14 y T21, esto es debido a que no realizaron la actividad. A partir de estas valoraciones pudimos identificar ocho grupos que potencialmente eran candidatos a adaptarse al proyecto seleccionado (tres

en el grupo de castellano: G11, G16 y G17 y cinco en el grupo de euskera: T13, T14, T18, T20 y T21). Como puede comprobarse, en estos grupos hay una mayor desviación positiva entre las valoraciones de los proyectos propuestos (barras a la izquierda) frente al suyo propio (barras a la derecha), siendo siempre sus autovaloraciones inferiores a las de los proyectos candidatos.

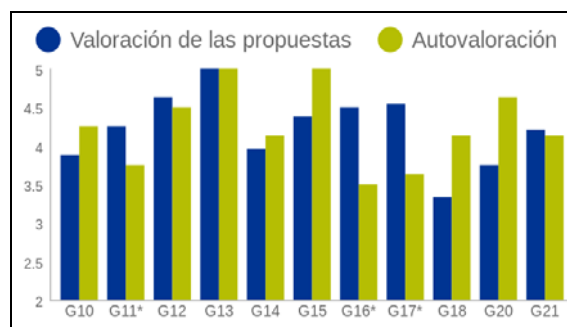


Figura 1: Valoraciones de los grupos de castellano.

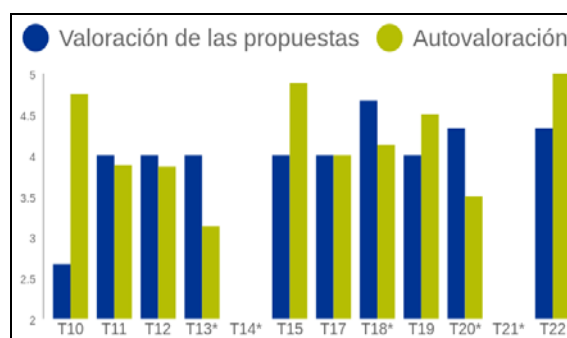


Figura 2: Valoraciones de los grupos de euskera.

Aunque no se disponía de las valoraciones de los grupos T14 y T21, fueron identificados también como candidatos a adaptarse, a partir de los resultados obtenidos hasta ese momento en su evaluación. Basándonos en los resultados de la figura 3, seleccionamos los proyectos candidatos (G20 en el grupo castellano y T11 en el grupo de euskera), ambos con valoraciones medias significativamente superiores a 4. La realidad es que una vez propuesta la migración al proyecto seleccionado a los ocho grupos identificados, y al ser esta opcional, sólo dos grupos de estudiantes optaron por realizarla: los grupos T13 y T14.

El resto de los grupos prefirió continuar con su proyecto, aun reconociendo que los proyectos para compartir eran mejores que el suyo. Preguntados sobre las razones para continuar con su proyecto, manifestaron que preferían seguir desarrollando un proyecto propio, aunque fuera inferior al propuesto para la adaptación.

<sup>1</sup> Accesible por medio del enlace: <http://tiny.cc/SWCuestionario>

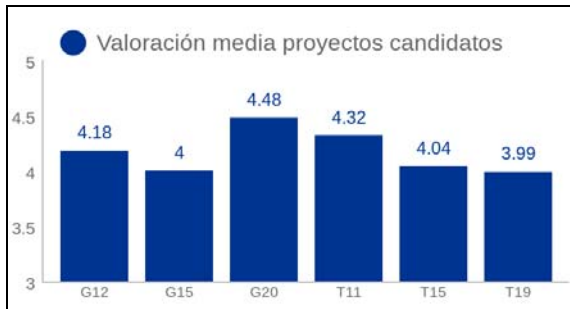


Figura 3: Valoraciones medias obtenidas por los proyectos candidatos.



Figura 4: Valoraciones de la revisión por pares.

La segunda valoración por pares se realizó en la semana 12 del curso (ver cuadro 1). Posteriormente se recogió la opinión de los estudiantes en relación a las valoraciones emitidas y recibidas. En la figura 4 se sintetizan los resultados más relevantes.

Como puede comprobarse, manifiestan un nivel de exigencia de 3,7 a la hora de realizar la validación y un 3,8 de satisfacción con las validaciones recibidas. La puntuación más baja, 3,6 se obtiene a la hora de valorar el nivel de exigencia que han tenido sus compañeros a la hora de valorarlos. Los comentarios recibidos van en la línea de extender esta actividad con más validaciones, incluso realizarlas desde el principio del curso. Otras opiniones plantean incluso aumentar el número de valoraciones de proyecto a realizar (actualmente cada grupo evalúa a otros tres).

### 6.1. Resultados globales de las actividades

Como muestra la figura 5, los tres indicadores utilizados para medir de forma global la satisfacción de los estudiantes han obtenido valoraciones entre 3,6 y 4,5. La pregunta sobre el aprendizaje de competencias profesionales se incluyó para valorar si el enfoque dado a la asignatura acercándola más al área de Ingeniería del Software, y el hecho de que las actividades propuestas son también utilizadas en las empresas, había sido bien recibido por los estudiantes. Por el resultado obtenido, consideramos que han valorado muy positivamente este esfuerzo.



Figura 5: Valoraciones generales de la experiencia.

### 6.2. Rendimiento académico

Todos los estudiantes que comenzaron la asignatura en evaluación continua la han superado en la convocatoria ordinaria, excepto uno que no se ha presentado. En la figura 6 puede verse la distribución de las calificaciones en porcentajes de los 42 estudiantes. Las calificaciones más bajas han correspondido a estudiantes identificados como idóneos para realizar la adaptación de su proyecto. Así, el 40% aproximadamente de aprobados, corresponde con seis de los ocho grupos identificados para realizar la migración.

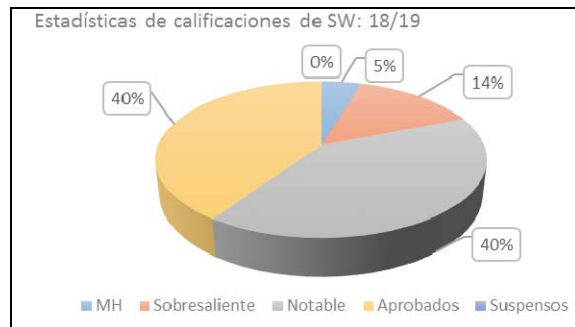


Figura 6: Distribución de las calificaciones obtenidas en el curso 2018-19.

En el grupo de euskera, todos los presentados han aprobado, los que migraron han superado la asignatura con una media cercana al 6, uno de los grupos que optó por no migrar, aun conociendo las deficiencias de sus desarrollos y las limitaciones de sus destrezas, han conseguido llegar al 5.

En otros dos grupos que declinaron migrar, concedores de su situación, la validación les sirvió de acicate para superar las debilidades de su proyecto (en media han llegado al 7,1). Estos resultados se han obtenido con una distribución grupal poco habitual respecto a cursos anteriores, ya que un 40% del alumnado ha provenido de otras titulaciones y un 25% han cursado SW como asignatura optativa, que no de línea curricular.

## 7. Discusión

Aunque sólo dos grupos de los ocho identificados como candidatos haya migrado su proyecto en la sexta semana, hemos comprobado que la metodología aplicada ha servido para que los grupos candidatos hayan podido validar y, por lo tanto, visualizar los requisitos mínimos necesarios para continuar con el proyecto. Todos ellos fueron capaces de mejorar sus soluciones y completar el proyecto de la asignatura de forma satisfactoria. Ningún grupo manifestó arrepentimiento por no haberse acogido a la adaptación, aunque un grupo manifestó tener dudas. Los alumnos que migraron consideraron acertada la decisión tomada y, además, opinan que les ayudó en posteriores desarrollos y a organizar mejor el código.

Hemos podido comprobar la resistencia de los estudiantes a utilizar soluciones desarrolladas por sus compañeros, prefiriendo continuar con sus desarrollos aun reconociendo que eran peores. Es posible que las normas sociales establecidas en el entorno académico conducen a que piensen que el uso de software no propio es una forma de copia, aunque en este caso sea “benedicida” por el profesorado.

Los resultados académicos podemos decir que son muy buenos, ideales desde el punto de vista de éxito (aprobados sobre presentados). La distribución de notas obtenida, con sólo un 40% de aprobados, también es muy satisfactoria. Además, hay que remarcar que los grupos candidatos a migrar su proyecto, excepto dos, obtuvieron las calificaciones más bajas de la asignatura.

Sobre la valoración por pares hay que insistir en la necesidad de motivar para que se realice con seriedad y rigor e incluyendo críticas constructivas. No es desdeñable el riesgo de que algunos estudiantes completen sus valoraciones de forma mecánica y puramente burocrática por lo que relacionar la inspección del código con el desarrollo de competencias en Ingeniería del Software parece un enfoque adecuado.

En cuanto al trabajo adicional que la implantación de esta iniciativa ha supuesto para el profesorado cabe señalar que ha sido asumible, en la medida que no se ha profundizado en la valoración de la calidad de las evaluaciones de los estudiantes. Al haber optado por fomentar el compromiso responsable de los estudiantes con sus valoraciones frente a la calificación de éstas, corremos el riesgo de que algunos afronten estas actividades de manera superficial. Entendemos que, en todo caso, es una minoría cuya detección no merece el esfuerzo de revisión en detalle de las valoraciones. Un trabajo futuro interesante consistiría en extraer patrones o heurísticos que permitan identificar el eventual problema de realización de evaluaciones mecánicas.

## 8. Conclusiones

La experiencia de innovación docente presentada, integra un método de aprendizaje activo (el ABP) con una técnica que fomenta la capacidad de análisis comparativo y crítico (la validación por pares) en el marco del desarrollo incremental de software. Además, enlaza el análisis y la valoración de software con la posibilidad de reutilización del mismo en un marco pautado a lo largo del curso, lo que aporta un sentido adicional a las validaciones realizadas por los estudiantes. Todo ello en el ámbito del aprendizaje de técnicas de desarrollo de software para la web que, por sus características, facilita todos los procesos de acceso, compartición y reutilización de código.

La experiencia se ha mostrado muy prometedora y abre un camino de consolidación y mejora en base a los resultados obtenidos en el primer año de su implantación. Las reticencias de algunos estudiantes a utilizar un código que ellos mismos consideran mejor que el que han realizado demanda realizar un diagnóstico más fundamentado. Es posible que los sistemas de calificación y enseñanza de la programación que, a menudo, ponen énfasis en la exigencia de que el estudiante desarrolle su propio código, condicionen su perspectiva y limiten la capacidad, cada vez más necesaria, de construir software aprovechando al máximo soluciones y librerías ya disponibles.

Por otra parte, se ha intentado vincular una metodología de enseñanza de propósito general (el ABP) con un ciclo de vida de desarrollo de software. Creemos que, en la enseñanza de la ingeniería del software, es necesario armonizar el ciclo de vida asociado a los proyectos ABP con el ciclo de vida de desarrollo de software, para ello es fundamental tener en cuenta no sólo la planificación y los hitos intermedios de desarrollo, sino que la arquitectura de la solución software a desarrollar, y la tecnología y herramientas utilizadas, constituyan un conjunto integrado. Creemos que la propuesta presentada puede ser una inspiración interesante en contextos similares y un paso en el sentido de plantear un ajuste específico de la metodología ABP al problema del desarrollo de software complejo.

## Agradecimientos

Deseamos hacer constar nuestra gratitud a los estudiantes de la asignatura SW, para los que trabajamos, y con los que hemos ido refinando y mejorando la propuesta aquí presentada. También queremos reconocer las aportaciones de los profesores César Domínguez, Arturo Jaime, Ana Sánchez e Imanol Usandizaga, así como las sugerencias y comentarios de los revisores, han sido de gran ayuda en el proceso de elaboración de este trabajo.

## Referencias

- [1] José Miguel Blanco, Imanol Usandizaga y Arturo Jaime, (2014). Gestión de Proyectos en el Grado en Ingeniería Informática: del PBL a la espiral de proyectos. En *ReVisión*, vol. 7 N° 3.
- [2] C. Bovill, A. Cook-Sather, P. Felten, L. Millard y N. Moore-Cherry, (2016). Addressing potential challenges in co-creating learning and teaching: overcoming resistance, navigating institutional norms and ensuring inclusivity in student-staff partnerships. *Higher Education*, 71(2), pp. 195-208.
- [3] César Domínguez, Arturo Jaime, Ana Sánchez y Jónathan Heras, (2016). A comparative analysis of the consistency and difference among online self-, peer-, external-and instructor-assessments: The competitive effect. *Computers in Human Behavior*, 60, pp. 112-120.
- [4] Simone C. dos Santos, y Felipe S. F. Soares, (2013, May). Authentic assessment in software engineering education based on PBL principles: a case study in the telecom market. En actas del 2013 *International Conference on Software Engineering*, pp. 1055-1062. IEEE Press.
- [5] G. Fuertes, M. Vargas, I. Soto, K. Witker, M. Peralta y J. Sabattin, (2015). Project-based learning versus cooperative learning courses in engineering students. *IEEE Latin America Transactions*, 13(9), pp. 3113-3119.
- [6] Alfredo Goñi, Jesús Ibáñez, Jon Iturrioz y José Ángel Vadillo, (2014). Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software. En actas de las *XX JENUI*, Oviedo, pp. 133-140.
- [7] Arturo Jaime, José Miguel Blanco, César Domínguez y Ana Sánchez, (2013). Interuniversity telecollaboration to improve academic results and identify preferred communication tools. *Computers & Education*. Vol. 64, pp. 63-69.
- [8] Arturo Jaime, José Miguel Blanco, César Domínguez, Ana Sánchez, Jónathan Heras e Imanol Usandizaga, (2016). Spiral and project-based learning with peer assessment in a computer science project management course. *Journal of Science Education and Technology*, 25(3), pp. 439-449.
- [9] Jacqueline O'Flaherty, Craig Phillips, Sophia Karanicolas, Catherine Snelling, Tracey Winning, (2015). The use of flipped classrooms in higher education: A scoping review. *The Internet and Higher Education*, 25, pp. 85-95.
- [10] Ernesto Panadero y Gavin T. L. Brown, (2017). Teachers' reasons for using peer assessment: positive experience predicts use. *European journal of psychology of education*, 32(1), pp. 133-156.
- [11] Jane Seale, (2009). Doing student voice work in higher education: an exploration of the value of participatory methods. *British Educational Research Journal*, 36(6), pp. 995-1015.
- [12] Keith Topping, (2003). Self and peer assessment in school and university: Reliability, validity and utility. En *Optimising new modes of assessment: In search of qualities and standards*, pp. 55-87. Springer, Dordrecht.
- [13] Miguel Valero-García y Javier García-Zubia, (2011). Cómo empezar fácil con PBL. En actas de las *XVII JENUI*, Sevilla, pp. 109-116.
- [14] José Ángel Vadillo, Rosa Arruabarrena y José Miguel Blanco, (2017). Uso de herramientas web en la asignatura Sistemas Web: facilitando el aprendizaje del alumnado y el proceso de evaluación. En actas de las *XXIII JENUI*, Cáceres, pp. 183-189.
- [15] Miguel Valero-García, (2012). PBL (Piénsatelo Bien antes de Liarte). En *ReVisión*, vol. 5 N° 2.
- [16] Thomas Wanner y Edward Palmer, (2018). Formative self-and peer assessment for improved student learning: the crucial factors of design, teacher participation and feedback. *Assessment & Evaluation in Higher Education*, pp. 1-16.