

# Uso de herramientas web en la asignatura Sistemas Web: facilitando el aprendizaje del alumnado y el proceso de evaluación

José A. Vadillo, Rosa Arruabarrena, José M. Blanco  
Facultad de Informática. Departamento de Lenguajes y Sistemas Informáticos, UPV/EHU  
Donostia/San Sebastián  
{ja.vadillo, rosa.arruabarrena, josemiguel.blanco} @ehu.eus

## Resumen

En las asignaturas con mucha actividad práctica, donde una gran parte de la evaluación se basa en el análisis del desarrollo de un proyecto de software, resulta muy pesada la evaluación del código y de la funcionalidad implementada por los estudiantes.

Este artículo aborda el problema y plantea alguna solución basada en un procedimiento y un conjunto de herramientas acordado previamente con los estudiantes. El objetivo de facilitar y mejorar la evaluación de la asignatura ha supuesto establecer un protocolo de entrega de resultados que también mejora el proceso desde el punto de vista de los alumnos, y además hace que mejoren sus competencias profesionales en desarrollo. La descripción del proceso, las herramientas y los resultados constituyen el núcleo principal de este trabajo.

## Abstract

In subjects with significant practical activity, where a great part of the assessment is based on the analysis of the development of a software project, it is particularly tiresome the code evaluation, as well as the evaluation of the implemented functionality by students.

This article discusses the problem and provides a solution based on a procedure and a set of tools previously agreed with students. The protocol to deliver student's developments improves not only their professional skills in web development but also the goal of making easier and better than before the assessment of the subject. Process description, tools and results are the core of this work.

## Palabras clave

Sistemas Web, Procedimientos docentes, Herramientas para el desarrollo web, Sistemas PaaS.

## 1. Introducción

En asignaturas con una actividad práctica de desarrollo importante, como es el caso de Sistemas Web, se presentan tres cuestiones a definir: la metodología y las herramientas a utilizar para el desarrollo de las soluciones, la entrega de los resultados obtenidos por los estudiantes y la evaluación de los mismos por parte del profesorado. Para abordar estas cuestiones hay una gran variedad de herramientas en la nube que pueden resultar útiles. En [13], por ejemplo, recurren a herramientas de *Google* y *Amazon* para automatizar la gestión de las entregas prácticas. En nuestro caso, también utilizamos herramientas de Google y otras de hospedaje en la nube, y que se detallarán más adelante. La idea es combinar herramientas locales junto con otras en la nube, siguiendo un procedimiento previamente establecido. Hay tres elementos clave: un entorno de desarrollo local, una aplicación de *hosting* tipo PaaS (*Platform as a Service*) [1] y software de control de versiones (concretamente, *Git* y *GitHub*).

El objetivo principal de este trabajo es mostrar cómo el uso adecuado del procedimiento propuesto, junto a la elección de herramientas realizada, facilita tanto el proceso de entrega del código de la aplicación al alumnado, como aligera la labor de corrección al profesorado. Más aún, desde el plano formativo, el uso de dichas herramientas, usadas en entornos profesionales, aporta un acercamiento al mundo laboral a nuestros estudiantes.

El artículo está organizado de la siguiente forma: Se comienza contextualizando la asignatura, para seguidamente pasar a describir el método de trabajo adoptado, muy relacionado este con las herramientas propuestas. En el siguiente apartado se representan las herramientas a emplear, así como su aportación al proceso de entrega de resultados y de evaluación. En el quinto apartado se detalla la secuenciación de los procesos de trabajos desde los puntos de vista de alumnos y profesores. A continuación, en el sexto apartado, se sintetizan las valoraciones aportadas por

los alumnos del curso 16/17 con respecto a la metodología y las herramientas empleadas. Además, se recogen las principales mejoras detectadas en el desarrollo de la asignatura durante los cursos 15/16 y 16/17, y se discuten posibles proyecciones a otras asignaturas. El artículo finaliza con las conclusiones.

## 2. Contexto

La asignatura Sistemas Web (SW) se centra principalmente en las tecnologías web, en el desarrollo de proyectos web y en sistemas web de apoyo al desarrollo e implantación de los mismos. El curso comienza con una somera presentación de la evolución histórica de los SW, junto con sus fundamentos, para ir presentando una a una las tecnologías básicas, tanto en el lado cliente como en el lado servidor: *HTML*, *JavaScript*, *JQuery*, *Aplicaciones de Servidor con PHP*, *XML*, *JSON*, *Ajax*, *Sesiones y Seguridad*, *Servicios Web SOAP* y *Arquitecturas REST* [6,15].

La asignatura está diseñada con el objetivo de que el estudiante adquiera las competencias mediante “*learn by doing*” y con metodología “*bottom-up*”, construyendo desde una maqueta inicial un sitio web que tendrá algunas de las características más relevantes de los sitios web actuales. Los estudiantes, en grupos de dos personas, desarrollan un sitio web, cuyo dominio/temática es común para todos ellos, de forma incremental. Las habilidades adquiridas les serán de utilidad para, posteriormente, entender mejor, por ejemplo, los Sistemas de Gestión de Contenidos (tipo WordPress, Joomla, Drupal) y poder adaptarlos a diferentes necesidades.

Se parte de un proyecto inicial sin apenas funcionalidad, salvo algún enlace, y el estilo implementado con páginas CSS. Semanalmente los estudiantes dotan de una 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 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 opcionalmente, los alumnos pueden efectuar otras caracterizaciones más avanzadas de su sitio web a fin de optar a una calificación más elevada.

El sistema de evaluación consta de nueve entregas prácticas a lo largo de doce semanas, con un peso del 70%, y tres pruebas escritas individuales, con un peso del 30%.

## 3. Método de trabajo

Como hemos descrito anteriormente, el principal eje de organización de la asignatura consiste en el desarrollo de una aplicación web de forma incremen-

tal. Prácticamente todas las semanas se diseñan e implementan nuevas funcionalidades a partir de un núcleo inicial. Esto genera una importante cantidad de entregas por parte del alumnado, mientras que sucesiva y rápidamente el profesorado debe revisarlas, dando retroalimentación continua sobre las mismas.

### 3.1. La gestión de las entregas

Actualmente muchos docentes trabajamos con aulas virtuales tipo Moodle, o bien versiones similares desarrolladas por las propias universidades. En el caso de nuestra universidad, el aula virtual se denomina *eGela*, y es una plataforma *Moodle* con una configuración específica a nivel de institución. Los estudiantes suben semanalmente cientos de ficheros en formatos comprimidos; y el profesor los descarga a su máquina, los descomprime, revisa y, si es necesario para la evaluación, los prueba. En esta última situación, el profesor tiene que cargarlos con la aplicación correspondiente e intentar hacer una o varias ejecuciones. Con suerte obtendrá los resultados esperados, si bien con demasiada frecuencia se dan casos de error de distinta índole como, por ejemplo, errores de *paths*, de sentencias SQL y de formato. En el supuesto de poder subsanar los errores, tendrá una visión aproximada de la implementación realizada por un grupo determinado de una asignatura concreta. En caso contrario, tendrá que concertar una cita con el grupo para que le haga una demo de la aplicación y poder tener los argumentos necesarios para la evaluación continua del grupo.

En nuestro caso, hemos simplificado al máximo el proceso y, sólo les requerimos que nos entreguen en cada laboratorio dos enlaces vía un cuestionario *Google Form* y la estimación de las horas invertidas. Las respuestas del cuestionario semanal se pueden descargar en formato hoja de cálculo, de manera que los enlaces del cuestionario se salvan automáticamente. Así, serán suficientes dos clics para acceder directamente al código, por un lado, y ver los resultados de la ejecución de las aplicaciones desarrolladas, por otro.

Para llegar a esta situación hemos tenido que cambiar tanto el método de trabajo común como, por supuesto, las herramientas a utilizar.

En concreto, para cada entrega, los estudiantes implementan las funcionalidades requeridas y efectúan las pruebas en sus equipos locales (*http://localhost*). Una vez asegurado su correcto funcionamiento, deben probar la aplicación en un servidor en la nube. Para ello deben hacer el despliegue de su aplicación y las pruebas oportunas en el *hosting* elegido. Una vez depurados los errores, sólo tienen que hacernos llegar la *url* a la página de inicio de su aplicación.

A su vez, la última versión del código generado y probado debe ser subida a un gestor *de desarrollo*

*colaborativo y de versiones* en la nube, proporcionándonos el enlace al mismo.

En algunas entregas se pide también que realicen una descripción por escrito de la estrategia de solución empleada, o bien algún diagrama de entidades o de casos de uso. En estos casos la documentación la generan en un documento que pueden almacenarlo en *Google Drive* (o similar) y les pedimos un tercer enlace, con permisos de lectura, al documento en cuestión. Alternativamente, pueden guardarlo en el propio gestor de desarrollo colaborativo en la nube.

### 3.2. Revisión de las entregas

Desde el formulario semanal, y por cada grupo, el profesor dispone simultáneamente de los enlaces a la aplicación en ejecución y al código. Desde un navegador, y sin ninguna instalación de software adicional, podrá ejecutar los casos de prueba pertinentes para comprobar el correcto funcionamiento del software entregado. Ante soluciones curiosas, imaginativas o dudosas, por ejemplo, y también en un solo clic, el profesor podrá acceder al código y ver la implementación exacta desarrollada por el grupo; es decir, qué y cómo lo han resuelto [5]. Así pues, esta combinación de herramientas y método de trabajo alivia en gran medida la recepción y revisión de las entregas al profesor, de manera que se puede dar la correspondiente retroalimentación con mayor celeridad también.

Además, esta estrategia permite gestionar fácilmente la revisión entre pares [2, 11, 12]. Una vez que el sitio web tiene cierta entidad, se pueden hacer una sesión de valoraciones cruzadas entre alumnos y grupos que sirvan de inspiración y/o, en ocasiones, de acicate, e incorporar algunas mejoras propuestas por compañeros en posteriores desarrollos.

## 4. Las herramientas

En el primer laboratorio los estudiantes instalan, configuran y prueban el correcto funcionamiento del entorno básico para desarrollar el proyecto de la asignatura. Este entorno, compuesto por herramientas locales y web, se describe a continuación.

### 4.1. Herramientas locales

Para realizar el desarrollo de proyecto en un ámbito local hemos seleccionado un conjunto de herramientas que cumplen dos características: son gratuitas y de libre distribución, y están disponibles para diferentes plataformas. Esto facilita el desarrollo en equipos personales de los estudiantes y les permite trabajar en diferentes lugares (laboratorios, clases, fuera de la universidad). Las herramientas son tres: el entorno de desarrollo PHP, un editor de texto plano avanzado orientado al mismo y una herramienta para control de versiones local.

**Entorno de desarrollo PHP**<sup>1</sup>. En nuestro caso se recomienda el uso de XAMPP<sup>2</sup>, o bien WAMPSEVER. Ambas son distribuciones de Apache, PHP y MySQL. La primera tiene versiones para Windows, Linux y Mac. La segunda es específica para Windows. Son entornos de desarrollo y depuración muy adecuados para el enfoque de nuestra asignatura.

**Editor de texto plano avanzado orientado a PHP**. Este aspecto lo dejamos más abierto, ya que cualquier editor avanzado actual soporta sintaxis PHP. Aunque en los laboratorios trabajamos con NotePad++, existen alternativas muy interesantes, como Atom y Sublime Text.

**Herramienta local para control de versiones**. Para esta tarea usamos la herramienta Git<sup>3</sup>. Es un sistema de control de versiones multiplataforma para la gestión y mantenimiento eficiente y confiable de proyectos con gran número de archivos. Los cambios se guardan mediante *instantáneas* (o “*snapshots*”) etiquetadas de la información almacenada. Consideramos que es una de las mejores herramientas para este propósito [7, 14, 16]. Además tiene una perfecta integración con GitHub, que usaremos como repositorio remoto con valor añadido.

### 4.2. Herramientas en la web

Para realizar el despliegue del proyecto en la nube el criterio de selección de herramientas es principalmente la gratuidad de las mismas. Otro aspecto importante es que no requieran un complicado proceso de registro/ingreso. De hecho, las herramientas seleccionadas pueden ser usadas por medio de la identificación proporcionada por una cuenta de Google, lo que agiliza el registro para trabajar con ellas. Las herramientas son un *hosting* de sitios web, GitHub y Google Drive.

**Hosting de sitios web**. Esta herramienta, dentro de la categoría de PaaS [1, 3, 4, 17], es absolutamente fundamental para probar el desarrollo sin necesitar el equipo local y, en consecuencia, sin necesidad de instalaciones ni configuraciones adicionales. Más aún, es una competencia a desarrollar en la asignatura el despliegue de aplicaciones web en servidores externos. En nuestro caso usamos *Hostinger* por su sencillez, su acceso gratuito (con o sin cuenta de Google+) y el número de incidencias relativamente bajo respecto al uso. Además, permite de forma gratuita el uso de un servidor de MySQL de forma similar al usado en local con XAMPP.

**Herramienta web para control de versiones: GitHub**<sup>4</sup>. Es una plataforma de desarrollo colaborativo para albergar proyectos empleando el sistema de control de versiones Git [9, 10]. Los archivos se

1 <https://secure.php.net/>

2 <https://www.apachefriends.org/es/index.html>

3 <https://git-scm.com/>

4 <https://github.com/>

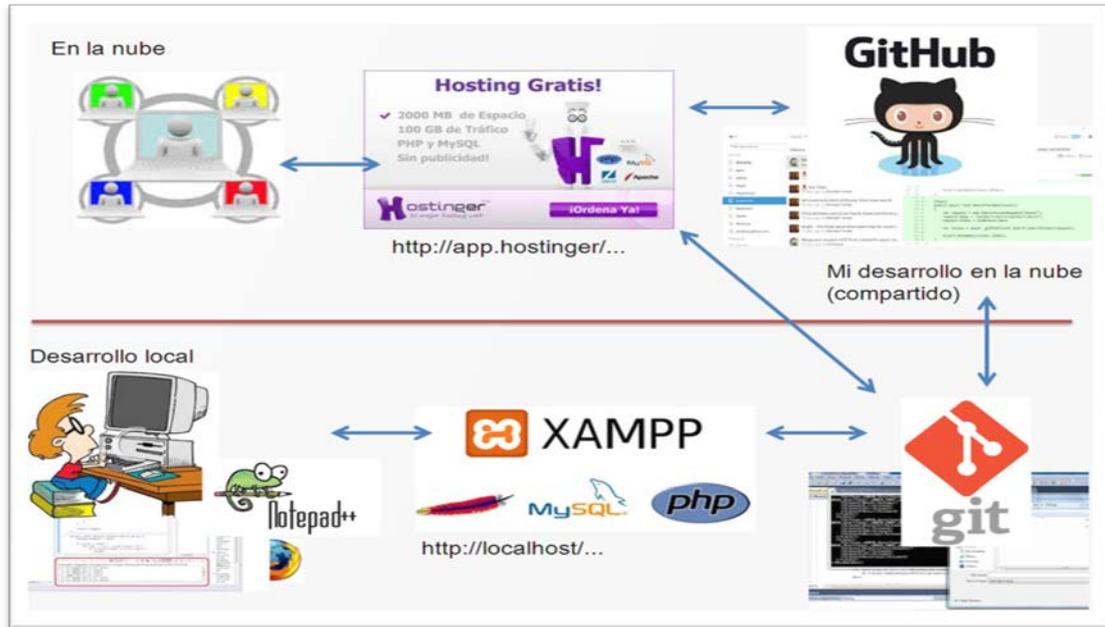


Figura 1: El proceso de trabajo y las herramientas utilizadas.

almacenan de forma pública (la privada es sólo mediante una cuenta de pago). Esta herramienta resulta de gran utilidad, ya que la usamos para distribuir a los estudiantes la versión 0, maqueta inicial del proyecto, consistente básicamente en una conjunto de menús con hojas de estilo adaptativas (“responsive”) [8].

**Google Drive.** Conjunto de herramientas para edición de textos, hojas de cálculo, formularios, etc. En nuestro caso, usamos los *Google Forms* para recoger los envíos de los estudiantes. En concreto, los enlaces al código y a la aplicación en la nube, y los enlaces a los documentos complementarios solicitados (dedicación, modelos y respuestas a preguntas abiertas). Los envíos de *Google Forms* se almacenan en una hoja de cálculo y es posible obtener rápidamente, en caso necesario, resúmenes agrupados de todas las respuestas recibidas.

## 5. El proceso de trabajo

Cada estudiante instala un paquete de distribución XAMPP (alternativamente, Wampp, Lampp o Mampp), un navegador con varios plugins específicos (el curso 16/17, Firefox) y un editor de texto plano con varios *plugins* es suficiente para la edición del código (el curso 16/17, NotePad++). Otras IDEs alternativas más potentes, si bien pesadas, podrían ser: *Aptana Studio*, *NetBeans*, *SublimeText*, etc.

Cada grupo/estudiante instala en su equipo de trabajo *Git* (software ligero para el control de versiones) y configura el directorio dónde se albergarán los sucesivos desarrollos semanales (versiones locales de las tareas realizadas). La presentación de los concep-

tos básicos de esta herramienta se realizan en el primer laboratorio de la asignatura.

A continuación los estudiantes crean dos cuentas por grupo ya en la nube: una en *GitHub* y otra en *Hostinger*. La primera es para publicar las versiones de los códigos fuentes (carpetas, subcarpetas, ficheros de textos, imágenes, manuales...) y la segunda para desplegar y verificar/ratificar la efectividad de su aplicación en un servidor web remoto, quedando disponible su uso para la comunidad.

En cada una de las semanas posteriores se pondrán a los estudiantes funcionalidades nuevas a incorporar en su sitio web, funcionalidades que deberán desarrollar, probar en local, subir a *Hostinger*, e instalar y validar en el servidor remoto.

Los alumnos efectuarán los desarrollos que consideren oportunos, junto con sus pertinentes verificaciones, usando las herramientas instaladas en sus equipos locales. Una vez alcanzados los objetivos, cada grupo actualizará los correspondientes repositorios remotos (el de *GitHub* y el de *Hostinger*) y remitirá al profesorado sus respectivos enlaces. Seguidamente, éste procederá a su validación por medio de un navegador web cualquiera.

El despliegue en *Hostinger* permite al profesorado evaluar el funcionamiento de forma directa; y a otros usuarios, conocedores de la *url*, validarlo como si estuviera en explotación. Si, además, el profesor desea consultar el correspondiente código fuente, bastará con visitar el repositorio remoto en *GitHub*.

El uso de estas herramientas posibilita la realización de revisiones por pares entre los propios estudiantes. En nuestro caso, y ya próximos a la última

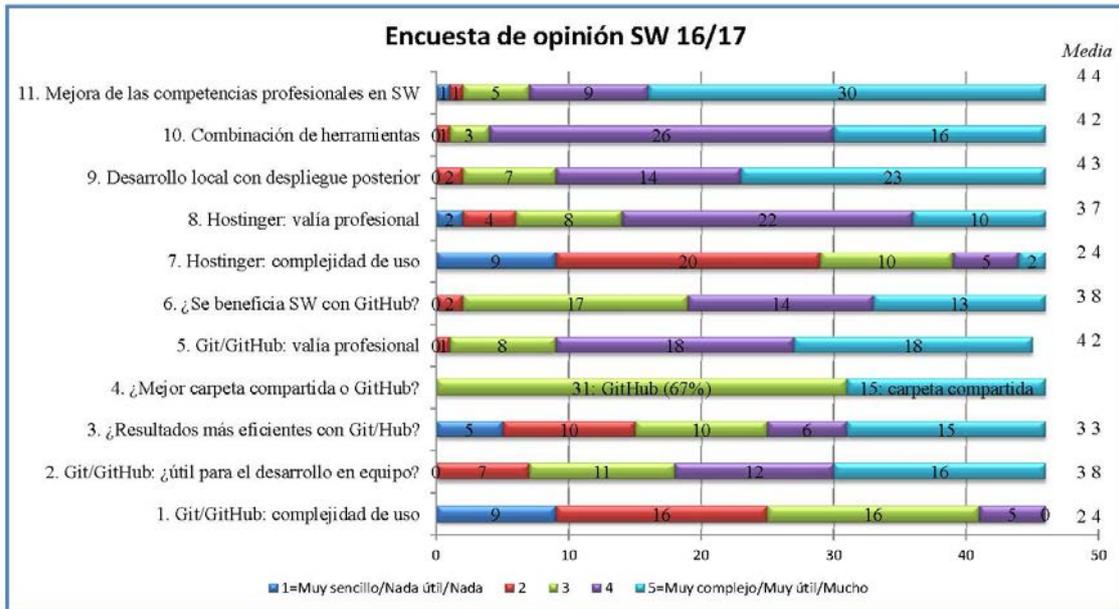


Figura 2: Resultados destacados de la encuesta de opinión del alumnado en SW.

fase del proyecto, los resultados obtenidos por cada grupo hasta ese punto son analizados por otros dos grupos. Para ello, los profesores facilitamos un conjunto de casos de prueba a valorar y los estudiantes realizan un informe de evaluación. Dicho informe se envía a cada grupo para que, de haberlos, subsane los errores antes de la entrega final.

## 6. Resultados, valoración de los estudiantes y extensiones

En las encuestas institucionales de opinión de cursos previos, la satisfacción general con respecto a SW ha sido de 4,3 sobre 5, y el 94,7% considera que la dificultad de la asignatura es media, (escala cualitativa [muy difícil...muy fácil]). El curso 16/17 Sistemas Web ha tenido 60 alumnos matriculados en los grupos de castellano y euskera, manteniéndose el nivel de satisfacción en valores muy similares a los de los cursos previos. A través de los formularios Google, administrados prácticamente de forma semanal, también les hemos solicitado una estimación de horas invertidas fuera del horario lectivo para rematar las tareas de caracterización de sus sitios web, y han manifestado haber dedicado 43,3 horas/estudiante, en media. Cabe, por último, indicar que el 98% ha superado la asignatura en la primera convocatoria.

Aun con una buena tasa de éxito y satisfacción, hemos querido dar un paso más y hemos administrado un cuestionario *ad hoc*. Esta vez se han realizado justo al finalizar el cuatrimestre, para recabar de primera mano la percepción del alumno con relación,

principalmente, a las herramientas, tecnologías y método de trabajo en SW, y a las competencias profesionales en el área de la Ingeniería del Software. De los 46 cuestionarios recibidos, seguidamente extraemos y comentamos los resultados más destacados (siempre con escala de tipo *Likert* [1-5]).

Para el objetivo de la asignatura, se les ha preguntado sobre la *complejidad del uso* tanto del binomio Git/GitHub como de Hostinger, habiendo sido el valor medio 2,4, si bien con gran dispersión en la distribución de valores (véase en la Figura 2, ítems 1 y 7). Esto es, no les han parecido herramientas ni especialmente sencillas ni especialmente complicadas, pero sí que han apreciado un valor añadido en su uso. Específicamente con respecto a Git/GitHub, consideran que les ha sido de utilidad para el trabajo en equipo (con 3,8, ítem 2), que les ha ayudado a realizar desarrollos más eficientes (con 3,3, ítem 3), y no solo ha sido beneficioso para SW sino también para la especialidad de Ingeniería del Software (con 3,7, ítem 6). Además, el 67% piensa que su uso es preferible a limitarse a una carpeta compartida para el equipo (ítem 4). Finalmente, y con vistas más allá de la propia asignatura, cabe destacar que los estudiantes opinan que el empleo del gestor *de desarrollo colaborativo y de versiones* les podría ayudar en el ejercicio de su profesión con una valoración media de 4,2, valor superior al 3,7 otorgado al conocimiento y uso de la herramienta de *hosting de sitios web* (ítems 5 y 8).

Con relación al método de desarrollo basado inicialmente en una implementación en local y su poste-

rior despliegue en la nube, lo valoran altamente con un 4,3, (ítem 9), la combinación de las herramientas implicadas - trinomio XAMP, Git/GitHub y Hostinger - con un 4,2 (ítem 10). Como conclusión de las cuestiones cuantitativas, el resultado más destacado es su apreciación sobre la mejora sus competencias profesionales, que la valoran en un 4,4 (sobre 5, ítem 11), y que redondea los buenos valores de las tasas de éxito y satisfacción, que ya conocíamos.

Acorde con las valoraciones numéricas, las respuestas a preguntas abiertas han sido mayoritariamente positivas y con gran participación (ya que no eran campos de respuesta obligada). Se han recabado 62 contribuciones sobre 78 posibles cuadros a completar; es decir, un 79%. Con respecto a Git/GitHub, los términos más frecuentes han sido “herramienta interesante”, “fácil uso”, “permite clonar”, “facilita el trabajo colaborativo/equipo”; si bien también hay alguna manifestación menos positiva como “farragosa la subida de ficheros” o bien propuestas de IDEs alternativos como “Cloud9” o “Codeanywhere”. Con respecto a Hostinger, las valoraciones otorgadas han sido más heterogéneas. Así, si bien les resulta la herramienta “útil”, “intuitiva”, “sencilla”, también ha habido comentarios menos favorables, como: “había momentos en los que fallaba”, “pueden bloquear una cuenta si haces demasiadas pruebas”, “la cuenta al ser gratuita su uso es limitado” y “existen alternativas más completas, de pago”<sup>5</sup>.

En el apartado *Aporta comentarios positivos/negativos sobre SW*, no ha habido ningún comentario negativo. Opiniones del estilo “más competente ahora”, “completa, exigente y muy interesante”, “practicar mediante trabajos, me ha resultado menos pesado y más fácil, llevando la teoría a la práctica”, “asignatura bien planteada”, “útil para el futuro y llevadera de trabajar”, entre otras, animan al equipo docente a seguir en la misma línea.

Para finalizar este apartado, queremos comentar las consecuencias que ha tenido esta forma de enfocar la parte práctica de la asignatura. En primer lugar, ha tenido una proyección hacia cursos anteriores (Sistemas Web es de tercer curso). En la actualidad, la asignatura Ingeniería del Software de segundo ya imparte un laboratorio de Git/GitHub para que los estudiantes suban sus proyectos a dicho repositorio. Otra proyección se ha dado en la asignatura Herramientas Avanzadas de Software del segundo cuatrimestre de tercer curso. En esta asignatura se ha pasado a usar el mismo método, un entorno local de desarrollo (*Visual Studio.NET* y *Git*) y un entorno en la nube (en este caso *Azure* y *GitHub*). Excepto las

herramientas, el resto de los procesos de entrega y evaluación se han adaptado de forma similar.

## 7. Conclusiones

El método de trabajo presentado, junto con la combinación de herramientas descritas, es común a los diferentes grupos de SW, y tiene una trayectoria de dos cursos académicos, si bien en el curso 16/17 se ha impartido la quinta edición de la asignatura en el Grado de Ingeniería Informática. En comparación con nuestra experiencia previa en la asignatura, así como experiencias en otras asignaturas afines, hemos constatado mejoras en tres ámbitos: Gestión del proyecto de la asignatura, gestión de los procedimientos de recepción y evaluación de los entregables, y resultados asociados al desarrollo de la asignatura.

En cuanto a la *Gestión del proyecto de la asignatura* hemos logrado una mejora de la distribución inicial del núcleo del proyecto, lo que facilita la sincronización de todos los grupos al comenzar las actividades prácticas de la asignatura. El acceso agrupado y directo a los desarrollos permite mostrar de forma sencilla en clase aspectos problemáticos o bien soluciones brillantes relativas a tareas concretas planteadas a los estudiantes. Además, se facilita la posibilidad de hacer distribuciones de código en puntos de control intermedio. El uso de herramientas de control de versiones en la nube permite en un momento dado compartir con los grupos soluciones estándar necesarias para conseguir nuevos objetivos logrados a partir de la consecución de otros previos. Si un grupo no llegase, por diferentes motivos, a un nivel estándar, no podría seguir las tareas derivadas. Ahora, de forma sencilla, se puede ofrecer la posibilidad de reengancharse a la asignatura. Por último, se simplifica la posibilidad de reutilizar los resultados como ejemplos o demos en años posteriores.

En lo que se refiere al *Procedimiento de recepción de resultados* destacamos aspectos positivos como la sencillez o la ausencia de limitaciones de tamaño de las entregas. También cabe resaltar que al tratarse de herramientas gratuitas multiplataforma, los estudiantes pueden instalárselas en sus equipos personales y trabajar fuera de los laboratorios del centro sin problemas. Además, para el profesorado, el que las entregas estén recogidas en hojas de cálculo, habilita las ordenaciones por fecha y hora de entrega, por entrega de actividades optativas frente a obligatorias, por horas de dedicación, etc.

Con respecto al *proceso de evaluación* de los trabajos cabe resaltar la eficiencia y simplicidad en el acceso al código y a ejecuciones de pruebas en tareas de evaluación. Como aspecto adicional, remarcar que facilita la posibilidad de realizar revisiones por pares de las entregas finales o en puntos intermedios del desarrollo.

<sup>5</sup> Para el futuro, hemos adquirido una licencia institucional de Hostinger, en la cual hemos desplegado algunos de los proyectos más relevantes, que permite aprovechar algunas ventajas (por ejemplo en el manejo de bases de datos) excluidas de la versión gratuita.

Por último, pensamos que el uso de las herramientas planteadas mejora las competencias técnicas y profesionales en el ámbito del desarrollo software de los estudiantes, así como su satisfacción con el aprendizaje. Por esta razón, la extensión de su uso a otras asignaturas del área, alineada con los usos actuales en el desarrollo de software, se está produciendo de una manera natural.

## Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Vicerrectorado de Innovación, Compromiso social y Acción cultural de la Universidad del País Vasco/Euskal Herriko Unibertsitatea, a través del proyecto DIMAROVE dentro del Programa de Innovación Educativa bienio 2017-18.

Nuestro agradecimiento al resto del profesorado del proyecto DIMAROVE por su colaboración en el desarrollo de este trabajo, y a los estudiantes de la asignatura de Sistemas Web por su participación en el mismo.

## Referencias

- [1] Philipp Brune, Michael Leiser, y Erica Janke, E. (2014). Towards an Easy-to-Use Web Application Server and Cloud PaaS for Web Development Education. En *2014 IEEE Intl Conf on High Performance Computing and Communications; 2014 IEEE 6th Intl Symp on Cyber-space Safety and Security; y 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS)*. IEEE (pp. 1113-1116).
- [2] César Domínguez, Arturo Jaime, Ana Sánchez, José Miguel Blanco, y Jónathan Heras, J. (2016). A comparative analysis of the consistency and difference among online self-, peer-, external-and instructor-assessments: The competitive effect. En *Computers in Human Behavior*, 60, 112-120.
- [3] G. Fylaktopoulos, G. Goumas, M. Skolarikis, A. Sotiropoulos, y I. Maglogiannis. (2016). An overview of platforms for cloud based development. En *SpringerPlus*, 5(1), 38.
- [4] Joel Gibson, Robin Rondeau, Darren Eveleigh, y Qing Tan. (2012). Benefits and challenges of three cloud computing service models. En *Fourth International Conference on Computational Aspects of Social Networks (CASoN)*. IEEE (pp. 198-205).
- [5] A. Gonzalez Álvarez, B. Aparicio Cotarelo, A. Lossent, T. Andersen, A. Trzcinska, D. Asbury, N. Høimyr y H. Meinhard. (2015). Extending software repository hosting to code review and testing. En *Journal of Physics: Conference Series*, vol. 664 núm.6, 062018. IOP Publishing. doi:10.1088/1742-6596/664/6/062018.
- [6] Hong He (2010). Applications deployment on the SaaS platform. En *5th International Conference on Pervasive Computing and Applications (ICPCA)*. IEEE (pp. 232-237).
- [7] Joseph Lawrance, Seikyung Jung y Charles Wiseman (2013). Git on the cloud in the classroom. En *44th ACM technical symposium on Computer science education*. ACM (pp. 639-644).
- [8] Pete LePage. Conceptos básicos de diseño web adaptable. En <https://developers.google.com/web/fundamentals/design-and-ui/responsive/?hl=es>
- [9] Justin Longo y Tanya M. Kelley (2015). Use of GitHub as a platform for open collaboration on text documents. En *11th International Symposium on Open Collaboration*. ACM (pp. 22).
- [10] Francisco J. Lopez-Pellicer, Rubén Béjar, Miguel A. Latre, Javier Nogueras-Iso y Francisco Javier Zarazaga-Soria (2015). GitHub como herramienta docente. En *XXI Jornadas de la Enseñanza Universitaria de la Informática*. Universitat Oberta La Salle (pp. 66-73).
- [11] David López y David Carrera (2016). Si elimino el examen ¿cómo evaluó? Una discusión sobre las actividades sustitutivas del examen y su escalabilidad. En *ReVisión*, 9(3), 65-73.
- [12] Afef Jmal Maâlej, Moez Krichen y Mohamed Jmaïel (2015). A comparative evaluation of state-of-the-art load and stress testing approaches. En *International Journal of Computer Applications in Technology*, 51(4), 283-293.
- [13] Germán Moltó y J. Damian Segrelles (2016). Panel web de gestión automatizada para actividades educativas no presenciales. En *XXII Jornadas de la Enseñanza Universitaria de la Informática*. Universidad de Almería. (pp. 311-318).
- [14] Karthik Ram (2013). Git can facilitate greater reproducibility and increased transparency in science. En *Source code for biology and medicine*, 8(1), 7.
- [15] Ionuț Cristian Resceanu, Cristina Floriana Reșceanu y Sabin Mihai Simionescu (2014). SaaS solutions for small-medium businesses: Developer's perspective on creating new SaaS products. En *18th International Conference System Theory, Control and Computing (ICSTCC)*. IEEE (pp. 140-144).
- [16] Diomidis Spinellis (2012). Git. En *IEEE Software*, 29(3), 100-101.
- [17] Nabil Sultan (2010). Cloud computing for education: A new dawn?. En *International Journal of Information Management*, 30(2), 109-116.