

Git y GitHub aplicados en la docencia

Francisco J Lopez-Pellicer, Rubén Béjar, Miguel A. Latre, Javier Noguera-Iso, Fco Javier Zarazaga-Soria
Depto. de Informática e Ing. de Sistemas
Universidad de Zaragoza
{fjlopez, rbejar, latre, jnog, javy}@unizar.es

Motivación Los **sistemas de control de versiones** (SCV) son una herramienta esencial para manejar proyectos de software desde hace décadas pero, desde hace algunos años, también se han introducido en la enseñanza TIC y no TIC como herramienta docente. Proporcionan una serie de funcionalidades claves para el desarrollo de proyectos como es el control de cambios en el código, la reversibilidad de dichos cambios, y la posibilidad de colaborar en el desarrollo del código. La incorporación a la enseñanza se justifica para ayudar a desarrollar la competencia de trabajo en equipo [7], facilitar la retroalimentación de los alumnos [5], establecer escenarios de desarrollo realistas [4] e, incluso, poder utilizarse como herramienta de monitorización [6]. Su uso no está exento de retos. Los alumnos pueden, por ejemplo, utilizarlo como un sistema de entrega más sin aprovechar su funcionalidad [2]. Desde una perspectiva académica existen riesgos como que la curva de aprendizaje sea tan elevada que afecte al normal desarrollo del curso [8]. Esta ponencia presenta la experiencia docente desarrollada en la asignatura *Ingeniería Web*¹ del itinerario de *Ingeniería de Software* del grado de *Ingeniería Informática* de la Universidad de Zaragoza durante el año 2014. En esta asignatura y para el desarrollo de la parte práctica, se ha utilizado Git², un SCV distribuido, y, para afrontar los retos y riesgos anteriormente mencionados, la plataforma GitHub³, un servicio comercial de alojamiento en la Web de repositorios Git.

Git y GitHub Git fue diseñado y desarrollado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux [1]. La licencia de Git es libre y hay distribuciones oficiales para los sistemas operativos OS X, Windows, Linux y Solaris. Según [10], si en 2011 más de la mitad de los programadores encuestados utilizaban como herramienta principal el SCV centralizado Subversion (51,3 %) y solo un 12,8 % utilizaba Git, hoy casi la mitad de los programadores usan Git o GitHub (41,9 %) seguido de Subversion (30,7 %). GitHub es un servicio comercial de alojamiento de repositorios Git remotos creado en el año 2008. GitHub proporciona una interfaz Web que permite al usuario registrado crear y gestionar repositorios Git remotos. Además del alojamiento, GitHub proporciona a cada repositorio una wiki, un gestor de tareas, un sistema de gestión de comentarios, un cuadro de control e, incluso, una página web propia. Hay que hacer notar que GitHub no es un proyecto de código abierto. Sin embargo, existen actualmente soluciones de software abierto como Gitorious y GitLab que permiten instalar en servidores propios un servicio de alojamiento de repositorios remotos con herramientas similares a las que ofrece GitHub. GitHub está considerado como la plataforma de alojamiento de repositorios remotos más popular. En abril de 2015, el número de usuarios es más de 9,4 millones y el número de repositorios alcanza los 22,4 millones [3]. Además, se estima que en 2014 había más 1200 cursos y 70.000 estudiantes utilizando este tipo de cuentas [9].

Git y GitHub aplicados a la asignatura Ingeniería Web Ingeniería Web es una asignatura de evaluación continua en la que el trabajo práctico pondera un 80 % en la nota final. A continuación se enumeran herramientas y funcionalidades de GitHub utilizadas en este curso distinguiendo entre las derivadas de Git y las exclusivas de GitHub. Las herramientas y funcionalidades se presentan agrupadas bajo tres criterios. El primero atiende a su uso por parte del alumno en su aprendizaje. Dentro de esta categoría incluimos herramientas y funcionalidades relacionadas con actividades de comunicación, productividad y participación de los alumnos. Las derivadas de Git son el diario de actividades, el trabajo fuera de línea y el trabajo en grupo, y las exclusivas de GitHub son los micro foros de discusión y el motor de búsqueda facetado. El segundo se fija en su papel como soporte del curso. Dentro de esta categoría incluimos herramientas y funcionalidades relacionadas con actividades el soporte del curso, incluyendo aspectos administrativos, de desarrollo del contenido y de realización del curso. Las derivadas de Git son

¹<http://titulaciones.unizar.es/asignaturas/30246/index14.html>

²<https://git-scm.com/>

³<https://github.com/>

integridad y compartir material de prácticas, y las exclusivas que ofrece GitHub son transparencia, publicación de contenido, edición en línea de contenido, seguimiento de los alumnos y plataforma de entrega con realimentación. Finalmente, el tercero denominado facilitadores describe aspectos relacionados con tecnologías, licencias y precios que han facilitado el curso. Tres están directamente derivados de Git: multiplataforma, clientes de Git pesados y código abierto. Otros tres son exclusivos de GitHub: clientes de GitHub pesados, interfaz Web y coste del servicio.

Git y GitHub, tal como se han utilizado en esta asignatura, presentan beneficios tanto para el aprendizaje de los alumnos como para la gestión de la asignatura. El alumno puede adquirir una serie de competencias profesionales transversales relacionadas con la capacidad para usar herramientas de la Ingeniería Informática y con el trabajo en grupo. Un beneficio indirecto y a más largo plazo es el uso de una herramienta que servirá al alumno para comunicar a terceros sus habilidades y destrezas profesionales ya que la cuenta de GitHub puede convertirse en su portafolio profesional. Desde el punto de vista del soporte del curso, GitHub ha permitido distribuir el contenido de prácticas con los alumnos de una forma más cercana a los usos y costumbres asociados al tópic de la asignatura. El flujo de trabajo y entrega así como los diferentes informes que proporciona la plataforma han permitido hacer un seguimiento de las actividades de los alumnos. Además, las herramientas de comunicación disponibles se han utilizado para señalar y corregir problemas potenciales relacionados con el desempeño de los alumnos. No obstante, las costumbres adquiridas por los alumnos son difíciles de cambiar. Por ejemplo, algunos alumnos han seguido utilizando servicios como Dropbox para coordinar su trabajo y, una vez finalizado, han utilizado GitHub solo como plataforma de entrega. También puede percibirse como un riesgo a la sostenibilidad la dependencia de una plataforma comercial como GitHub. Este riesgo se mitiga por la disponibilidad de plataformas de código abierto como Gitorious y GitLab que podrían ser instaladas en equipos docentes.

Conclusiones La plataforma GitHub aplicada a la docencia no debe ser considerada como un mero servicio de alojamiento de repositorios de Git en la Web con funcionalidades extra. La experiencia docente descrita da una idea de su potencial papel como herramienta de aprendizaje y de gestión de la enseñanza pero es necesario profundizar en el análisis de esta plataforma como herramienta docente para dimensionar adecuadamente dicho papel. Futuras investigaciones deben abordar aspectos cuantitativos y cualitativos relacionados, por ejemplo, con la satisfacción de alumnos y profesores, los resultados académicos, y la evolución de la carga docente.

Agradecimientos Este trabajo ha sido financiado a través del proyecto PIIDUZ_2014_127 correspondiente a las ayudas del Programa de Incentivación de la Innovación Docente en la Universidad de Zaragoza.

- [1] Scott Chacon y Ben Straub. *Pro Git*. Apress, 2nd edición, Noviembre 2014, ISBN 9781484200773.
- [2] Michael Cochez, Ville Isomöttönen, Ville Tirronen y Jonne Itkonen. *How Do Computer Science Students Use Distributed Version Control Systems?* En Vadim Ermolayev, HeinrichC Mayr, Mykola Nikitchenko, Aleksander Spivakovsky y Grygoriy Zholtkevych (editores): *Metadata and Semantic Research*, páginas 210–228. Springer International Publishing, 2013, ISBN 978-3-319-03997-8.
- [3] GitHub. *Press*, 2015. <https://github.com/about/press>.
- [4] Ken T N Hartness. *Eclipse and CVS for group projects*. *Journal of Computing Sciences in Colleges*, 21(4):217–222, Abril 2006.
- [5] Oren Laadan, Jason Nieh y Nicolas Viennot. *Teaching operating systems using virtual appliances and distributed version control*. En *the 41st ACM technical symposium*, página 480, New York, New York, USA, 2010. ACM Press, ISBN 9781450300063.
- [6] Ying Liu. *CVS historical information to understand how students develop software*. *International Workshop on Mining Software Repositories (MSR 2004)"W17S Workshop - 26th International Conference on Software Engineering*, 2004:32–36, Enero 2004.
- [7] Andrew Meneely y Laurie Williams. *On preparing students for distributed software development with a synchronous, collaborative development platform*. En *40th ACM technical symposium on Computer science education*, página 529, New York, New York, USA, Marzo 2009. ACM Request Permissions, ISBN 9781605581835.
- [8] Ivan Milentijevic, Vladimir Ciric y Oliver Vojinovic. *Version control in project-based learning*. *Computers & Education*, 50(4):1331–1338, Mayo 2008.
- [9] Paul Sawers. *GitHub Wants Schools to Collaborate Around code*, Febrero 2014. <http://thenextweb.com/insider/2014/02/11/github-wants-schools-collaborate-code/>.
- [10] Ian Skerrett. *Eclipse Community Survey 2014 Results*, Junio 2014. <https://ianskerrett.wordpress.com/2014/06/23/eclipse-community-survey-2014-results/>.