



Docencia 2.0

Juan Julián Merelo, Fernando Tricas

Control de versiones y el ecosistema a su alrededor

Los proyectos en ingeniería, en general, se desarrollan por equipos. Y los equipos necesitan herramientas para controlar quién hace qué, incorporar cambios a través de un proceso establecido y poder, en caso necesario, ver la evolución del diseño a través de las versiones del mismo. En el caso del desarrollo de sistemas informáticos también es posible, y necesario en algunos casos, volver a algún punto anterior del trabajo, corregir algún error introducido en una versión anterior o explorar caminos alternativos.

En ingeniería informática los sistemas que incorporan esas características se denominan sistemas de gestión de código fuente o de control de versiones. Son programas que gestionan el acceso por parte de un equipo al código fuente, en archivos de texto guardados en diferentes directorios, de un proyecto. Bien desde la línea de órdenes o desde un interfaz gráfico, permiten *comprometer* un fichero o un grupo de ficheros que han cambiado, crear diferentes *ramas* de desarrollo y fusionarlas con una serie de órdenes simples. El trabajo concurrente sobre un mismo proyecto provocará necesariamente la necesidad de coordinarse. Pero de manera casi inevitable se producirán conflictos: cuando dos miembros del equipo han modificado la misma línea, hay que determinar si se trata de un error, un desajuste respecto a las especificaciones o, simplemente, diferentes formas de abordar un problema; el sistema de control de versiones marcará sobre el texto las líneas en las que se ha producido el conflicto, mostrando las dos versiones, y nos obligará a solucionarlo eligiendo, eliminando o modificando la línea o líneas que hayan causado el conflicto.

Hoy en día cualquier empresa con un mínimo departamento TIC usa, o debería usar, un sistema de control de versiones; entre todos los existentes uno de los que más atracción ha ganado es *git*, diseñado inicialmente por Linus Torvalds con su experiencia en la gestión del desarrollo del kernel de Linux pero hoy en día convertido en el estándar *de facto*; hoy en día lo mantiene y desarrolla Junio Hamano. Permite el desarrollo con equipos distribuidos, de manera no necesariamente lineal e incluye un sistema de publicación web de los repositorios,

entre otras características interesantes.

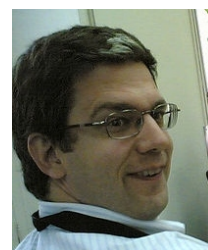
Entender el funcionamiento de *git* como principal sistema de control de versiones en la actualidad y sus patrones de trabajo es fundamental en la carrera de informática, tan fundamental como trabajar con un lenguaje de programación de *scripting* o entender el idioma inglés. Y lo es hasta tal punto que debe ser una habilidad transversal como lo es usar Internet. Esto es, no algo incluido de una asignatura como Ingeniería del Software, sino una parte integral del currículum de todas y cada una de las asignaturas de una carrera de Informática. Tampoco será raro que el informático termine encontrándose con estas herramientas cuando explore el código de otros, en cualquier caso.

Porque el aprendizaje del ingeniero debe prepararle, a través de la práctica, para su trabajo y la mejor forma de hacerlo es usando las mismas herramientas para que su uso se interiorice y se desarrollen estrategias propias para el mismo. Porque

JJ Merelo es catedrático de Universidad en el área de Arquitectura y Tecnología de Computadores, y actualmente director de la Oficina de Software Libre de la UGR. Mantiene un blog desde el año 2002, y lo ha utilizado en clase desde el año 2004; también wikis y, últimamente, agregadores y otras herramientas TIC. Últimamente le ha dado por el *flipped learning*, de lo que se informará debidamente en esta columna.



Fernando Tricas García es profesor titular de Lenguajes y Sistemas Informáticos del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Empezó a estudiar la blogosfera casi cuando aún no existía (allá por el año 2002) y a tratar de integrarla en los cursos y tareas docentes un poco después. Ha impartido numerosas charlas relacionadas con el tema de la Web 2.0. Es actualmente Director de su departamento.



no cabe duda de que usar git puede ayudar en todas y cada una de las asignaturas para desarrollar trabajos en grupo y elaborar documentos de la misma forma. Pero además, al ser por omisión una herramienta de trabajo en grupo, permite desarrollar estrategias de organización de este tipo de trabajo que son también útiles para el empleo posterior.

Por último, pero no por ello menos importante, trabajar con repositorios de git públicos como GitHub hace que el alumno desarrolle un portafolio de trabajos realizados que serán su mejor carta de presentación para obtener empleo y presentar su trabajo a la comunidad de desarrolladores. De hecho, muchas empresas de recursos humanos hacen *mining* de repositorios git para encontrar a empleados que correspondan a un perfil y nivel de habilidad requerido por una empresa.

Todo ello sin dejar de lado las características sociales de estos sitios, que nos pueden permitir conocer proyectos interesantes, desarrolladores de los que aprender y, en definitiva, tejer esa red de aprendizaje que nos puede dar soporte en el futuro para nuestros propios proyectos.

Aunque el uso no es mayoritario, git (y otros sistemas de control de versiones) también pueden utilizarse para todo tipo de archivos de texto, por ejemplo, documentación o escritura colaborativa de un trabajo. Incluso para gestionar

un blog (con Jekyll, por ejemplo). Los fuentes (en \LaTeX) de estas columnas están alojados en GitHub y uno de nosotros (J. J. Merelo, junto con Pablo Hinojosa) publicó en Amazon un libro sobre git que también está alojado en GitHub: <https://github.com/oslugr/curso-git>.

También hemos estado experimentando (tanto en repositorios públicos como privados) con la publicación de datos de experimentos, por la simplicidad a la hora de gestionar grandes cantidades de información, tenerlas disponibles en la web, compartirlas...

En resumen, git es el verdadero sistema operativo de los equipos de trabajo en informática. Y como tal debe ocupar una posición privilegiada dentro de la enseñanza de la misma.

Todas las columnas de la serie Docencia 2.0 pueden descargarse en formato LaTeX desde <https://github.com/ReVision-Docencia-20/Columnas>

©2014 J.J. Merelo, F. Tricas. Este artículo es de acceso libre distribuido bajo los términos de la Licencia Creative Commons de Atribución, que permite copiar, distribuir y comunicar públicamente la obra en cualquier medio, sólido o electrónico, siempre que se acrediten a los autores y fuentes originales