

Git:

sistema de control de versiones



MINISTERIO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y DEPORTES

Dirección General de Evaluación y Cooperación Territorial

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF)

Recursos Educativos Digitales

Mayo 2024

NIPO (web) 164-24-014-5

ISSN (web) 2695-4176

DOI (web) 10.4438/2695-4176_OTE_2019_847-19-121-5

NIPO (formato html) 164-24-015-4

NIPO (formato pdf) 164-24-013-X

DOI (formato pdf) 10.4438/2695-4176_OTEpdf123_2020_847-19-134-3

Git: sistema de control de versiones

Por José Ramón Rodríguez Sánchez para INTEF

<https://intef.es>

Obra publicada con licencia de Creative Commons

Reconocimiento-Compartir Igual 4.0 Licencia Internacional.

<https://creativecommons.org/licenses/by-sa/4.0/>



Para cualquier asunto relacionado con esta publicación contactar con:
Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado
C/Torrelaguna, 58. 28027 Madrid.
Tfno.: 91-377 83 00. Fax: 91-368 07 09
Correo electrónico: recursos.educativos@educacion.gob.es



El autor de este artículo

José Ramón Rodríguez Sánchez es Ingeniero en Informática por la UCM y profesor de Formación Profesional de la especialidad de Informática desde el año 2000. Ha encaminado su propia formación, así como su actividad lectiva, al desarrollo web. En este campo, ha impartido varios cursos de formación para profesores y ha coescrito dos libros de texto para módulos de ciclos de grado superior de informática.

En el plano personal, es padre de dos niñas que felizmente “consumen gran parte de sus recursos”, y aficionado a caminar por la montaña, para posteriormente disfrutar de una buena comida y conversación en agradable compañía.

 joseramon.rodriguezsanchez@educa.madrid.org



Introducción

Este artículo está dedicado a la herramienta de control de versiones Git que nos permite guardar las diferentes versiones de un documento, volver a una versión anterior y fusionar versiones, de forma que resulta muy útil especialmente en ciclo formativos relacionados con la informática para guardar el progreso de los distintos archivos de programación en la nube.



Logo de Git



La Herramienta

Git es una herramienta usada masivamente en el ámbito del desarrollo/programación de aplicaciones informáticas. Esto no quiere decir que no se pueda usar en otros contextos. Por ejemplo, se ha usado para la escritura de este artículo.

En lugar de ir generando un nuevo documento cada vez que hay modificaciones que puede sobrescribir contenido antiguo, e ir llamando al documento artículo_v1.doc, articulo_v2.doc, articulo_v3.doc... pues se deja la tarea de Git de ir “tomando instantáneas” de las diferentes versiones, de manera que, a ojos del usuario, sólo hay una versión del artículo (la última) pero se le puede pedir a Git que recupere cualquiera de las versiones anteriores.

Un concepto usado masivamente en el ámbito de Git es el de **repositorio**, entendiéndose como tal una carpeta que contiene archivos relacionados entre sí y que son susceptibles de ser gestionados por un sistema de control de versiones. Volviendo al ejemplo de los libros, para un escritor de libros de texto, cada libro publicado con todo su material complementario (imágenes, tablas, ejercicios...) podría ubicarse en una carpeta, y Git la podría tratar como un repositorio.

Su uso más frecuente, sin embargo, es para el trabajo colaborativo. Siguiendo con el ejemplo de la escritura de un artículo o libro, si se estuviera llevando a cabo entre varios autores simultáneamente, podría darse el caso de que un autor sobrescribiese accidentalmente lo que otro ha escrito. Para estos casos se recomienda su uso con un repositorio centralizado, que es un ordenador en internet (también podría usarse uno en una red local) en el que se almacena una versión maestra, contra la que cada autor tiene que sincronizarse cada vez que tiene que introducir cambios.

Mientras que para el alumnado y profesorado de programación (informáticos) su uso es imprescindible (ésta u otra herramienta que realice la misma función), para el resto de los usuarios puede ser un buen complemento para tener un seguimiento de las versiones de sus documentos. Su uso no es muy complejo a un nivel básico, pero manejarla en profundidad y sacarle todo el jugo ya es harina de otro costal, y queda reservado a expertos.



Explicación del uso en el ámbito educativo

Su uso es gratuito. Además, es una aplicación de código abierto, lo que significa que su creador, el ingeniero de software finlandés Linus Torvalds, permite que se descargue el código fuente que escribió para crearla, leerlo y, el que sepa cómo, modificarlo generando su propia versión.

Aunque se desarrolló inicialmente para el sistema operativo Linux, está disponible también para los principales Sistemas Operativos, como Windows y MacOS. En Windows se instala un emulador de Linux sobre el que se ejecutan directamente comandos de Linux, como ls, cp o nano.

Para un usuario no técnico que sólo quiere tener las diferentes versiones de uno o más archivos en su propio ordenador, únicamente es preciso descargar el archivo instalable, instalarla y empezar a usarla.

En cualquiera de los ciclos formativos de la Familia Profesional de Informática y Comunicaciones (Desarrollo de aplicaciones web y Desarrollo de aplicaciones multiplataforma) su uso es imprescindible, y se recomienda usarla en primer curso, a la vez que se aprende a programar.

También se podría hacer una introducción a su uso en alguna asignatura tecnológica de Bachillerato, así como en ciclos formativos, tanto de grado medio como superior de la misma rama tecnológica (Electrónica, Diseño gráfico...).

En cuanto al tiempo de aprendizaje para un uso básico (una introducción), tal vez se podría conseguir en una formación de unas 10 horas, asumiendo que los destinatarios de la formación tienen conocimientos de informática a nivel de usuario medio (qué es un sistema de ficheros, qué es un archivo, un directorio, qué significa/representa copiar/mover/eliminar un archivo/directorio...).

Para convertirse en un usuario medio de la herramienta son necesarias bastantes más horas, del orden de 30-40, así como su uso regular: "A caminar se aprende caminando".

Su uso experto, con un manejo de su potencia/funcionalidades, se reserva a unos pocos gurús en todo el mundo (si existe alguno, además de su creador citado).

Conceptualmente, no es una herramienta compleja, pero cuando se empieza a profundizar, se descubre que está llena de matices y que, en el fondo, no está pensada para usuarios no técnicos. Es muy potente y sofisticada, pero eso le resta usabilidad.

Un itinerario simplificado sobre cómo se podría enseñar esta aplicación sería:

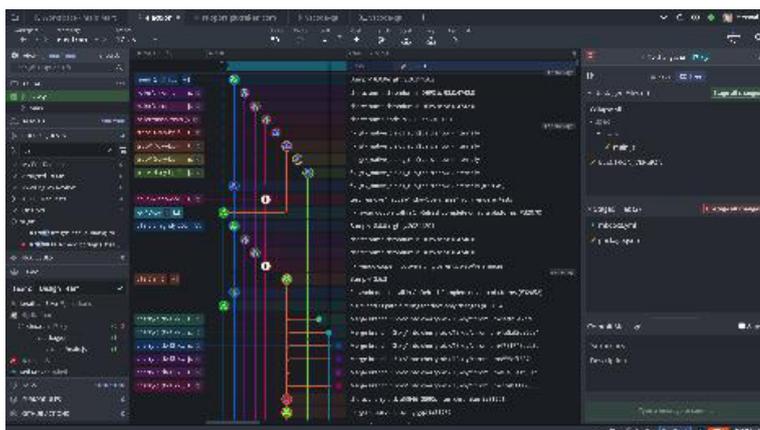
1. Introducción, conceptos generales, descarga e instalación.
2. Qué es un repositorio: creación y clonado.
3. Versiones (instantáneas) de un archivo: tomar instantáneas, consultar las instantáneas existentes, recuperar una instantánea pasada...
4. Deshacer cambios. Repercusión para Git de las operaciones sobre archivos (creación, modificación, eliminación, copia, movimiento...).
5. Trabajo con ramas, que es el mecanismo que Git nos ofrece para trabajar en paralelo con diferentes archivos y, llegado el momento y si se considera oportuno/necesario, fusionar el trabajo en una sola rama.
6. Trabajo con repositorios remotos. Sincronización de versiones.
7. Operaciones avanzadas con repositorios e instantáneas.

En cuanto a la interfaz de usuario, se puede usar de dos maneras. Para los usuarios noveles, existen aplicaciones con una interfaz gráfica (ventanas, menús, iconos, botones...) que permiten realizar la mayoría de las tareas necesarias para manejarse con Git.

Un ejemplo de estas aplicaciones con interfaz gráfica es **GitKraken**.



Logo de GitKraken



Captura de una pantalla de la aplicación

Para aquellos que se sientan duchos con las interfaces de línea de comandos (esas ventanas con fondo negro en las que para copiar un archivo hay que escribir **copy c:\archivo_origen d:\archivo_destino**), se puede sacar el máximo partido a Git mediante comandos (de hecho, es para esta forma de uso para la que inicialmente se desarrolló).

```
e739538 (HEAD -> main) Anadido un nuevo file3
e995fbb Eliminado file3
ad1d12f Anadido file3
3a2a6e4 Anadido file2
931dabe (origin/main, origin/HEAD) Intento de fusionado fallido sobre f1.html
aa19572 (origin/rama2) Modificado f1.html en rama2
24d0a59 Modificado f1.html en main
3a8d274 Creación de f1.html
f90ef31 Solución de conflicto en archivo file1
bba9ca0 (origin/rama1) Creación de archivo file1 en rama1
a08d4e6 Creación de archivo file1 en rama main
4a0835d Merge branch 'main' of
into main
a7cc0f1 Create arch4
d7a6a7e Modificación de .gitignore para excluir e1
0185e36 Creación de archivo arch5 en local
b26678e Create arch3
1fad6c7 Resuelto conflicto local vs. remoto en arch2
6d23cd0 Create arch2
```

Captura de una terminal de línea de comandos tras la ejecución del comando git log

Los comandos más comúnmente usados desde la línea de comando, por citar algunos, son: **git init**, para crear un repositorio; **git commit**, para crear una instantánea; **git log**, para listar las instantáneas tomadas; **git status**, para comprobar el estado del repositorio y sus archivos; **git push**, para subir un repositorio local de nuestro ordenador a un almacén de repositorios en la nube (como GitHub) donde se sincronizará con la versión remota o **git pull**, que realiza la tarea inversa, descargando la versión del repositorio remoto al equipo local.



Logos de GitHub

Y no se puede escribir un artículo sobre Git sin citar a GitHub. Se trata de un almacén de repositorios en la nube, para que los usuarios de Git puedan tener una copia de sus repositorios locales, compartirlos con otros usuarios, trabajar de manera colaborativa y mucho más. Existen otros, como GitLab o BitBucket, pero GitHub es el más ampliamente conocido y usado.



Metodología y Didáctica Aplicada

Git es una herramienta compleja que se aprende con el uso. La forma de aprenderla es mediante una introducción conceptual de los comandos básicos y, a partir de ahí, que el alumnado entregue todas sus prácticas de programación de aplicaciones usándola. Adicionalmente, una vez se aprenden los comandos, lo ideal es usar una aplicación gráfica como el anteriormente citado GitKraken o GitHub Desktop.

Está completamente orientada al trabajo colaborativo, en particular al desarrollo de software en equipos de trabajo.

La metodología de aprendizaje más adecuada debe ser, como ya se ha comentado, eminentemente práctica. Esto significa que, tras una explicación conceptual, se debe pasar a resolver ejercicios concretos.

Cada día, se podría indicar al grupo con qué grupo de comandos se va a trabajar al día siguiente, para que investigasen por su cuenta en qué consisten (una forma de clase invertida). El día de la explicación se empezaría con una lluvia de ideas sobre qué les han inspirado esos comandos (dificultad conceptual, de uso...), pasando posteriormente a una explicación teórico-práctica por parte del profesor. Finalmente, los alumnos deberían resolver supuestos prácticos concretos, inicialmente de forma individual, y progresivamente en grupo.



Valoración Personal

Git es una herramienta muy “poderosa” con la que se pueden hacer cosas “sorprendentes” muy rápido, pero cuyo manejo avanzado está al alcance de muy pocos. Esencial en el mundo del desarrollo de aplicaciones. Además, es una aplicación que permite que el alumnado sea activo en el aprendizaje y desarrolle la competencia digital tan fundamental en el currículo del siglo XXI.



Recomendación final

La recomendación final es usarla en ciclos tecnológicos o, en su defecto, en grupos de Bachillerato con una gran orientación al desarrollo de aplicaciones y gusto por los desafíos en general, y por los programas complejos en particular. Y tener paciencia porque los resultados vistosos tardan en llegar.



Información y materiales complementarios

Referencias:

- ▶ [Sitio web de Git](#)
- ▶ [Referencia](#)
- ▶ [Libro online](#)
- ▶ [Cheat sheet o "Chuleta"](#)
- ▶ [Vídeos](#)
- ▶ [A Visual Introduction to Git](#)
- ▶ [A Grip On Git](#)
- ▶ [GitHub](#)
- ▶ [GitKraken](#)

Licencias de las aplicaciones citadas:

- ▶ **Git:** GNU GPL v2
- ▶ **GitKraken:** Propietaria. Uso gratuito con repositorios públicos.
- ▶ **GitHub:** 3 tipos principales de licencias *Open Source*: **MIT** (Esta es la licencia más ajustada. Es un tipo de licencia del tipo "Haz lo que quieras con mis cosas, pero no me demandes"), **Apache** (lo mismo que la MIT, pero con más cláusulas; ideal para abogados y juristas) y **GPL** (la licencia más exigente).

Recursos:

- ▶ [Logo de Git](#)
- ▶ [Logo de GitKraken](#)
- ▶ [Captura de una pantalla de la aplicación GitKraken](#)
- ▶ [Logos de GitHub](#)
- ▶ [Logos de GitHub](#)
- ▶ Captura de una terminal de línea de comandos tras la ejecución del comando `git log` (imagen propia): **Git CLI.jpg**

Derechos de uso

- ▶ Todas las marcas nombradas en el artículo son nombres y/o marcas registradas por sus correspondientes propietarios.
- ▶ Las imágenes han sido proporcionadas por el autor. Algunas de ellas corresponden a capturas de pantalla de la herramienta.
- ▶ El texto ha sido elaborado por el autor expresamente para este artículo.

