
git

Introducción a Git

La vida sin control de versiones

Cómo funciona el control de versiones

Por qué Git

Introducción a Git

VCS

Un Version Control System permite a todxs los desarrolladores y diseñadores trabajar en el mismo código base. No teniendo que estar atentos a todo lo que cambian lxs demás.

Su nombre significa que puedes controlar todas las diferentes versiones sobre las que estás trabajando; también es llamado Source Control System.

La vida sin control de versiones

Programadores cowboys

Existen programadores cowboys que eligen trabajar en solitario.

Imaginemos uno de estos programadores que está desarrollando un app para un cliente

Empezaría programando la estructura básica del proyecto para servir de plantilla antes de saber que quiere el cliente

Nuestro cowboy trabajaría varios días creando nuevos ficheros y haciendo modificaciones en otros para una nueva funcionalidad. Siempre en solitario y siempre ante el peligro.

La vida sin control de versiones

Programadores cowboys

Al enviar esta funcionalidad el cliente la rechaza y nuestro cowboy borra el código y continúa trabajando

Unos días después el cliente ha cambiado de opinión y acepta esta funcionalidad

Nuestro cowboy tendrá que reescribir todo el código o buscar entre sus correos y backups la funcionalidad

La vida sin control de versiones

Programadores cowboys

Además un proyecto por ejemplo Android tiene multitud de ficheros xml, java, imágenes … es muy complicado restaurar un proyecto antiguo

Otra opción sería tener copias de seguridad de todo el proyecto, pero necesitaría muchísimo espacio

En caso de trabajar en un equipo, tendrían que comparar línea a línea los cambios

Esto solo funcionará mientras nuestro cowboy trabaje sólo

La vida sin control de versiones

Programadores cowboys

Si nuestro programador cowboy hubiera utilizado CVS podría guardar una copia de cada fichero guardado

De esta manera tendrá una serie de versiones de su app, 0.1, 0.2, 0.3 etc.

Cada una de estas tendrá una descripción de los cambios y funcionalidades añadidas

Solo se guardarán los archivos modificados

Es muy fácil compartir código y probarlo en diferentes ramas sin interferir a los demas

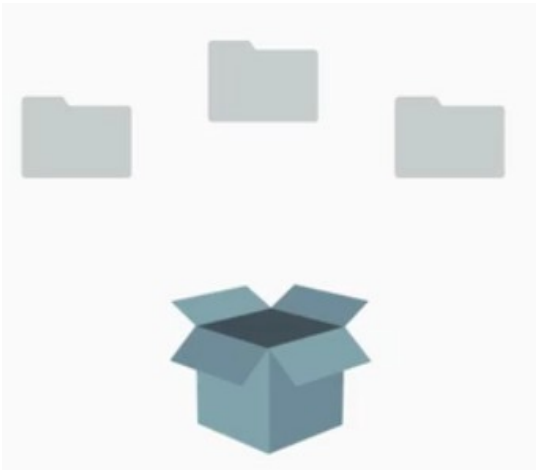
Cómo funciona el control de versiones

Un repositorio es una colección de todas las versiones de un proyecto junto a más información
Un repositorio nos da información sobre el orden en el que ocurrieron los cambios, una descripción de cada cambio y quién lo realizó

Cada proyecto debería tener su propio repositorio

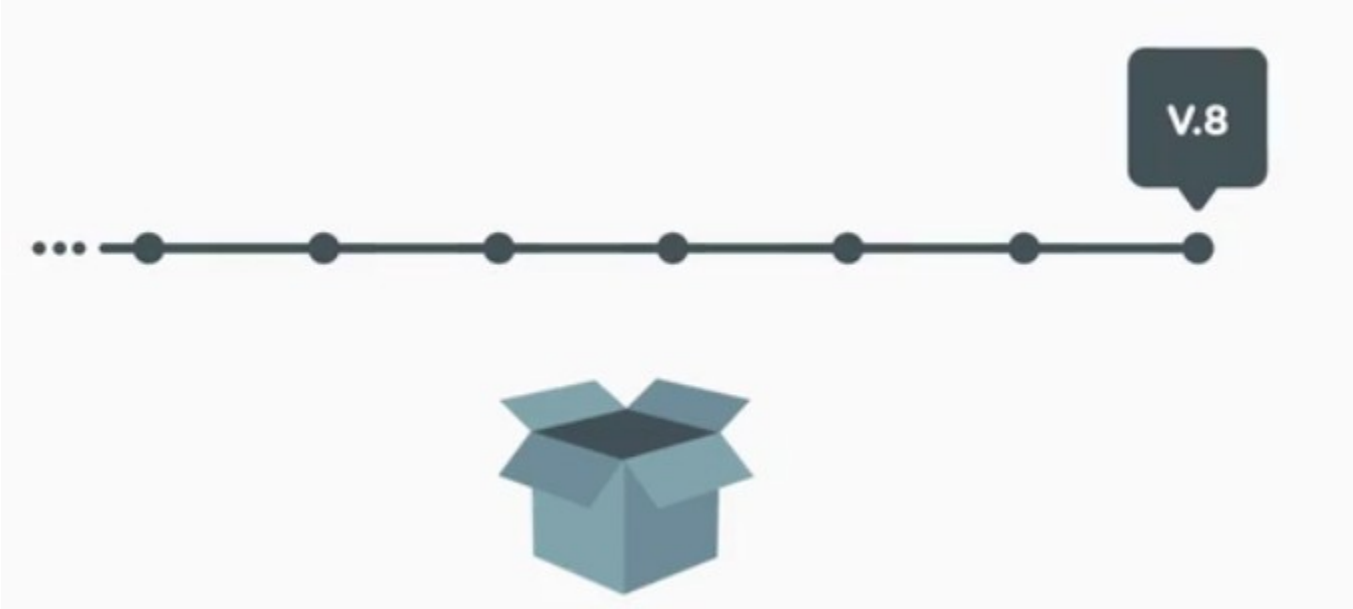
Cómo funciona el control de versiones

- Existen tareas que pueden demorarse varios días o semanas, por lo que hasta que no se complete no se publicará nueva versión
- Debemos indicar al CVS cuando una versión está terminada manualmente, esto se llama **commiting**
- Las **versiones** se llaman **commits** (del inglés enviar y comprometerse por el envío)
- Toda información es guardada en carpetas especiales ocultas para no añadir complejidad



Cómo funciona el control de versiones

- Los CVS tienen la funcionalidad de mostrar una lista de commits o incluso cambiar la versión de tu proyecto a alguna de estas versiones



Cómo funciona el control de versiones

- Los CVS también permiten compartir los proyectos con otros usuarios para colaborar y no perder de vista los cambios de los demás
- También añaden herramientas especializadas para ayudarnos a trabajar con proyectos grandes y complicados
- Existen muchos controles de versiones: CVS, SVN, Mercury, Git… depende de nuestras preferencias cual es mejor utilizar

Por qué Git

Git

- Git es un SCV desarrollado por Linus Torvalds para ayudarle a administrar todo el trabajo del mantenimiento del núcleo de Linux
- El núcleo de linux tiene más de 15 millones de líneas de código
- Cada día se añaden 3500 nuevas líneas de código
- Cada nueva versión del kernel involucra a más de 1000 desarrolladores diferentes
- Git se desarrolló para hacer la colaboración lo más rápida e indolora posible

Por qué Git

Control de Versiones Centralizados (SVN)

Existen otros sistemas de control de versiones **centralizados** donde solo existe un repositorio guardado en un servidor en red

Estos tienen una serie de desventajas frente a los distribuidos

No puedes realizar ningún trabajo en tu proyecto sin conectarte a la red

Si ocurre algo a ese repositorio podrías perder todo el historial de tu proyecto

La necesidad de requerir que todo pase por el servidor central puede dificultar la colaboración

Por qué Git

Control de Versiones Distribuidos (GIT)

Git es un sistema de control de versiones **distribuido** donde no existe un repositorio central

Git tienen una serie de ventajas frente a los centralizados
Tienes tu propio repositorio, por lo que no necesitas estar conectado a la red para trabajar en el repositorio
Las interacciones con el repositorio son más rápidas
La colaboración es más fácil, solo hay que coger tu propia copia del repositorio y empezar

Por qué Git

arquitectura de Git

Linus Torvalds no tenía originalmente la idea de que Git fuera un CVS si no una colección de herramientas y comandos para que alguien pudiera utilizarlos para construir un CVS

Este sería el motor de un buen CVS que funcionaría rápido y fácil y la gente podría utilizar cualquier carrocería con ese motor

Con el tiempo Git tendría un motor y una carrocería propias, por lo que evolucionó a un CVS completamente funcional

La ventaja es que todavía hay acceso completo al motor, pudiendo realizar operaciones muy avanzadas con Git, haciendo tu repositorio muy robusto y seguro

Por qué Git

GitHub

Es el sitio más popular para compartir tu código y administrar tus proyectos

Es como una red social para tu repositorio

Puedes compartir tus proyectos online

Hacer publicidad de tus proyectos

Compartir proyectos de forma privada con tus amigos y colaboradores

Otras personas pueden ver el historial de tu proyecto y añadir comentarios o incluso subir sus propias commits para mejorar tu proyecto

Test

¿Qué ventajas tiene trabajar con un CVS que no tiene un repositorio central como SVN?

a. No se necesita conexión a internet para realizar commits al proyecto
b. La mayoría de las operaciones se realizan rápido ya que el repositorio con el que trabajas está guardado en tu equipo
c. No hay un solo punto de fallo que pueda causar la pérdida de tu proyecto
d. Todas son correctas.
e. Ninguna es correcta

Test

En terminología de control de versiones, ¿qué es un repositorio?

a. El repositorio es donde todo el código borrado irá en caso de ser necesitado más tarde
b. El repositorio es dónde toda la información del control de versiones y los archivos se guardan.
c. El repositorio es el código de la última versión que funciona
d. Un repositorio es la descripción de los cambios realizados en un proyecto

Test

Git fue creado en gran parte para ayudar a administrar qué proyecto

a. El kernel de Linux.
b. El sistema operativo windows
c. GitHub
d. No hay un proyecto específico por el que se creó Git

Test

Cual de los siguientes conjuntos de herramientas entre otras proporciona un CVS

a. Herramientas para analizar si hay fallos en el código
b. Herramientas para compilar tu código
c. Herramientas para explorar el historial de tu proyecto.
d. Herramientas para exportar tus proyectos a otras plataformas
e. Todas son correctas
f. Todas son falsas

Test

El repositorio de cualquier proyecto que utiliza CVS es normalmente guardado en

a. Tu carpeta de documentos
b. Un conjunto de carpetas y ficheros ocultos.
c. En una base de datos diseñada especialmente para ello
d. Un único fichero de texto con todos los cambios
e. Todas son correctas

Test

En terminología de control de versiones, ¿qué nombre tiene un cambio terminado en el proyecto?

a.repository
b.commit.
c.final version
d.git-version

Test

Existen muchos controles de versiones pero en principio los distribuidos son menos vulnerables, más robustos

a.verdadero.
b.falso

Test

GitHub es un sitio que está enfocado a enseñar a la gente cual es la mejor manera de utilizar git

a.verdadero
b.falso.

Test

Un factor importante para elegir Git+Github como gestor de control de versiones y repositorio es tanto su popularidad como los servicios gratuitos y abiertos disponibles.

a.verdadero.
b.falso

Webgrafía

Cowboy coder http://c2.com/cgi/wiki?CowboyCoder

http://git-scm.com/book/en/Getting-Started-About-Version-Control

http://en.wikipedia.org/wiki/Distributed_revision_control

http://en.wikipedia.org/wiki/Linus_Torvalds

http://stackoverflow.com/questions/740053/why-should-i-use-git-instead-of-svn

http://git-scm.com/book/ch9-1.html

http://royal.pingdom.com/2012/04/16/linux-kernel-development-numbers/

http://git-scm.com/