

INTRODUCCIÓN A GIT

Mentoría 5: Max Rodríguez

Es una navaja suiza con muchas herramientas con comandos como Linux.



INTRODUCCIÓN

Es un sistema de control de versiones que fue creado por Linux. **Sistema de control de versiones:** es un sistema que registra los cambios de unos o varios archivos a través del tiempo permitiéndote recuperar versiones específicas de esta más adelante.

- Registra cada cambio en el proyecto o repositorio quien, y cuando lo hace, en una base de datos.
- Permite volver a estados previos del desarrollo.
- Permite gestionar diferentes versiones del proyecto(ramas) para trabajar en paralelo y luego fundirlas.
- Permite colaborar entre diferentes usuarios en un mismo repositorio facilitando la resolución de conflictos.
- Se utiliza principalmente en proyectos de desarrollo de software, pero sirve para cualquier otro tipo de proyecto.

Es una aplicación que permite gestionar los cambios que se realizan sobre los elementos de un proyecto o repositorio, guardando así versiones del mismo en todas sus fases de desarrollo.

Commit: es una versión de una fotografía. Contiene información sobre el autor, el mantenimiento y el mensaje de los cambios.

FLUJO DE TRABAJO HABITUAL

- Creamos un archivo
- Lo editamos
- Lo guardamos
- Lo volvemos a editar
- Y lo volvemos a guardar 'n' número de veces.



Diseñado por Linus Torvals (creador de kernel Linux). Surge en el 2005 a partir de la necesidad de remplazar el sistema de control de versiones que se utiliza en ese entonces (perforce) para el proyecto de kernel de Linux.

Es un sistema de control de versiones de código abierto y actualmente es el más



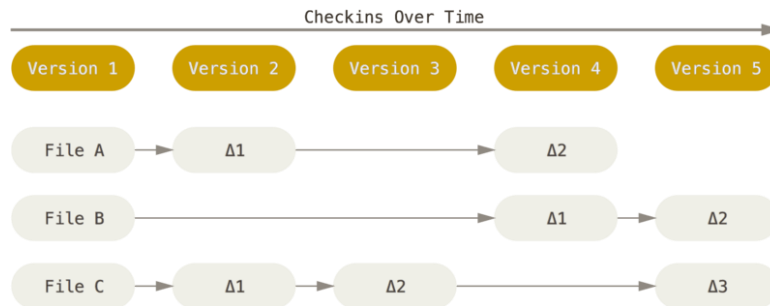
extendido. Git tiene una arquitectura distribuida, lo que significa que, en lugar de guardar todos los cambios de un proyecto en un único sitio, cada usuario contiene una copia del repositorio con el historial único sitio, cada usuario contiene una copia del repositorio con el historial de cambios completo del proyecto. Esto aumenta significativamente su rendimiento.

¿PARA QUE SIRVE?

Hacer esto te ayudará a evitar confusiones sutiles a la hora de utilizar la herramienta. Git almacena y maneja la información de forma muy diferente a esos otros sistemas, a pesar de que su interfaz de usuario es bastante similar. Comprender esas diferencias evitará que te confundas a la hora de usarlo.

Copias INSTANTANEAS, NO DIFERENCIAS

La principal diferencia entre Git y cualquier otro VCS (incluyendo Subversion y sus amigos) es la forma en la que manejan sus datos. Conceptualmente, la mayoría de los otros sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) manejan la información que almacenan como un conjunto de archivos y las modificaciones hechas a cada uno de ellos a través del tiempo.



CARACTERÍSTICAS

Git es distribuido: todos los colaboradores tienen una copia con todo el historial del repositorio (forma local).

Git tienen la habilidad combinar los cambios

Central= se necesitan un servidor central

Clon= copia de repositorio

Git seguro: Utiliza **hashes SHA1** para garantizar que todos tienen el mismo archivo. Además, al utilizar **el protocolo SSH** asegura que los datos viajan seguros por la red.

Hashes: algoritmo descifrado genera una firma de la versión

Puerto (22) protocolo SSH: (conexiones con el servidor) puede ser con un proxy o a través de un VPN. Puede servir para acceder a máquinas remotas a través de la red y manejar por completo el sistema mediante un intérprete de comandos. También podemos copiar datos de forma segura y cifrada.

Puerto (80) HTTP: (Hypertext Transfer Protocol) protocolo de comunicación que permite la transferencia de información en la Word wide web.

Puerto (443) HTTPS: (Hypertext Transfer Protocol) es un protocolo de aplicación basado en el protocolo http destinado para la transferencia segura de datos.

VPN: Una red privada virtual o VPN es una conexión cifrada a Internet desde un dispositivo a una red. La conexión cifrada ayuda a garantizar la transmisión segura de datos confidenciales. Evita que las personas no autorizadas espíen el tráfico y permite que el usuario trabaje de manera remota. La tecnología de VPN se usa ampliamente en los entornos corporativos.

Sin dependencia: Git al ser un sistema de control de versiones distribuido nos permite trabajar sin una conexión directa al servidor.

Git es rápido: la mayoría de las operaciones en Git son locales, esto disminuye bastante el tiempo la respuesta p/e al buscar en el historial. Cada repositorio de un usuario es un clon completo suponiendo que nuestro servidor central falla, no perdemos el historial de versiones.

Si algo pasa con nuestro equipo, siempre y cuando se haya subido cambios al repositorio, nuestro código estará seguro.

Opciones

Git Kraken



Es una potente y elegante interfaz gráfica multiplataforma para Git desarrollada con Electrón. De forma muy sencilla podemos llevar el completo seguimiento de nuestros repositorios, ver ramas, tags, crear nuevos, todo el historial de nuestro trabajo, commits etcétera. Con GitKraken puedes tener

controlados todos tus proyectos con una interfaz muy elegante y sencilla.

Git LAB

GitLab nació como un sistema de alojamiento de repositorios Git, es decir, un hosting para proyectos gestionados por el sistema de versiones Git. Sin embargo, alrededor de esta herramienta han surgido muchas otras herramientas muy interesantes para programadores y equipos de desarrollo, que envuelven todo el flujo del desarrollo y el despliegue de aplicaciones, test, etc. GitLab se ofrece como un software libre que puedes descargar e instalar en cualquier servidor. Esta posibilidad permite usar GitLab para una empresa, profesional u organización en sus propios servidores, sin ningún coste adicional.



BitBUCKET

Bitbucket es la herramienta ideal para compartir datos, implementar y crear código, así como automatizar pruebas. Se puede acceder a Bitbucket Cloud mediante una URL o también permite el alojamiento local, utilizando infraestructuras on-



premisses. Bitbucket Server, por otro lado, es compatible con Bamboo, automatizando procesos sin límite de licencia. Es una herramienta para colaboración de coding y con alojamiento en Git, Bitbucket es una excelente opción para equipos profesionales de gestión de proyectos e ingeniería de software. Atlassian compró la marca en 2010 y esto permitió su integración con otras herramientas de la

empresa—herramientas también orientadas para la gestión de proyectos con metodologías ágiles— tales como: Trello y Jira.

FLUJOS DE TRABAJO CON GIT

Es versátil y flexible que podemos adaptarlo a nuestras formas de trabajo, pero hay algunos flujos que trabajo “predefinidos” que podemos usar e igualmente adaptarlos a nuestras necesidades.

- Flujo centralizado (centralized Word Flow)
- Flujo de características o ramas (future Flow o Bach)
- Gitflow

INSTALACION EN CUALQUIER EQUIPO

INSTALACION MAC

- Entrar a la página oficial
- Servidor externo
- Se empieza a descargar
- Ir a preferencias de sistema y seguridad
- Contraseña y ya se instala

INSTALACION WINDOWS

- ir al buscador (Git aparece en el primer link)
- La última versión es la recomendada
- Se descarga y ejecuta
- Instalarlo y esperar.

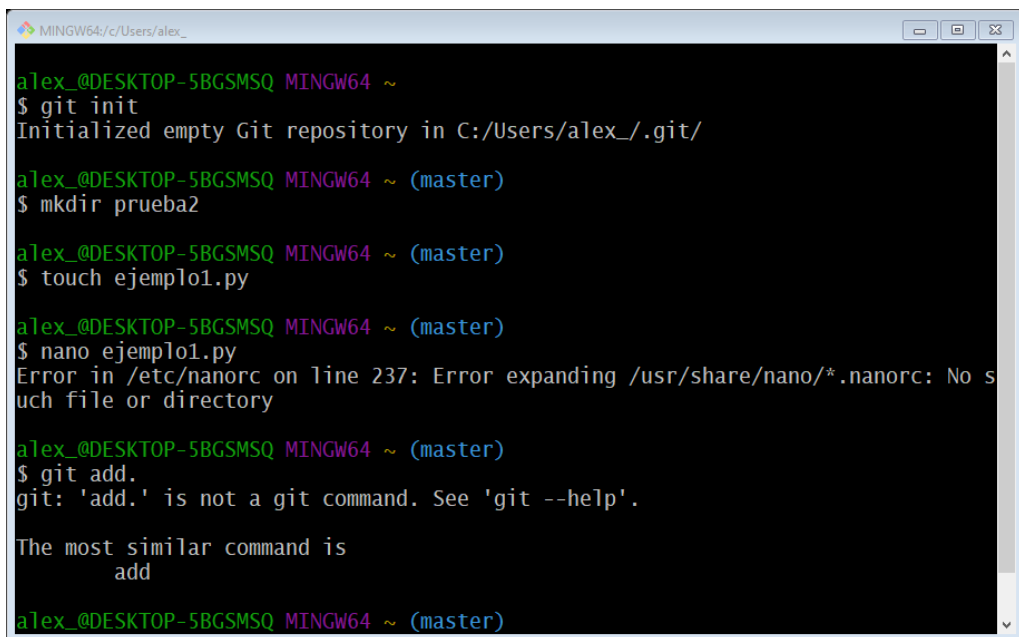


INSTALACION GNU/LINUX

Es porque seguramente nos llevamos bien con los comandos

- Abrir la terminal y actualizar la lista de paquetes.
- Sudo apt-get update
- Sudo apt-get upgrade
- apt -get install git (comando para instalar Git)

PRACTICA DE GIT



```
alex_@DESKTOP-5BGSMSQ MINGW64 ~  
$ git init  
Initialized empty Git repository in C:/Users/alex/.git/  
  
alex_@DESKTOP-5BGSMSQ MINGW64 ~ (master)  
$ mkdir prueba2  
  
alex_@DESKTOP-5BGSMSQ MINGW64 ~ (master)  
$ touch ejemplo1.py  
  
alex_@DESKTOP-5BGSMSQ MINGW64 ~ (master)  
$ nano ejemplo1.py  
Error in /etc/nanorc on line 237: Error expanding /usr/share/nano/*.nanorc: No such file or directory  
  
alex_@DESKTOP-5BGSMSQ MINGW64 ~ (master)  
$ git add.  
git: 'add.' is not a git command. See 'git --help'.  
  
The most similar command is  
    add  
  
alex_@DESKTOP-5BGSMSQ MINGW64 ~ (master)
```

COMANDOS BASICOS

git init crear repositorio

mkdir crear carpeta

touch crear archivo

nano editar

git add . añade todos los directorios

git status ver donde esta

git commit recibe -m mensaje que quiero dar

`git log` recibe diferentes parámetros

`git commit -a-m` nuevo commit

RAMAS

`git Branch` meter una rama

`git log` visualizar