

Git

Texto del video bien separado

¿Que es este sistema?

Un control de versiones, me permite tener un historial completo de todo el código que se desarrolla sobre la aplicación

Es normal introducir errores entre versiones para ello se regresa a la anterior y se sigue trabajando lo que se denomina **Rollback**

Dos soluciones:

1. Tratar de resolver el error y tratar de seguir agregando código a la versión
2. Tratar de recordar como era la versión y tratar de devolver el código a como se encontraba

Git permite trabajar sobre nuestro código de manera descentralizada.

Cuando varios desarrolladores están trabajando en un mismo proyecto estos deben actualizar sus cambios en un servidor que se encuentra en algún lugar de la nube y si este servidor deja de funcionar por alguna razón deja de funcionar no se puede acceder al código en el servidor u actualizar el mismo

Pero afortunadamente Git trabaja de manera descentralizada

cada uno de los desarrolladores tiene una copia completa del código que se esta trabajando y la misma puede ser compartida por medios alternativos.

Usos de GIT

- Historial
- Almacenar codigo
- Trabajar en equipo
- CUando se introdujo un error

Historial

Genera los antecedentes de todo el código realizado y el historial que otros desarrolladores han realizado en particular

Trabajar en equipo

Permite trabajar en equipo de manera rápida y simple

Cuando se introdujo un error

Cuando se encuentra un error en producción, se puede empezar a volver atrás dentro del historial para ver qué commit fue que introdujo el error y así reducir la cantidad de código a analizar para dar con el mismo

Instalación

Se debe ingresar a la página de git y la plataforma ofrece en pantalla la versión más reciente

Al instalarse viene con un programa llamado Git BASH (acrónimo de: Bourne-Again Shell)

Que es una terminal que permite ejecutar los comandos de Git y emular los comandos de Linux o los de Mac

Énfasis en usar Git BASH, no la terminal de Windows

Voy a la pantalla principal estando en Windows, clic derecho abrir Git BASH

En Mac presionar la tecla CMD + espacio y escribir terminal

Seguido muestra la terminal

Comienzo a configurar Git

Primero

git --version (para saber si se instaló bien, debería aparecer una versión de Git)

git config --global use.name "Cristhian Sosa"

git config --global user.email cristhiansosazurita@gmail.com

En este momento debo tener instalado un editor de texto en mi caso tengo VS code
git config —global core.editor “code —wait” (— wait es para que la terminal se quede esperando hasta que yo cierre el editor de texto)

Como hago para ver mi archivo de configuracion global?

git config —global -e

Core.autocrlf

Tengo 2 personas que se encuentran trabajando con el mismo repositorio

Persona A en Windows

Salto de line se utilizan los caracteres especiales de CR (Carriage Return) o LF (Line Feed)

Cuando persona A desee subir codigo tendra que eliminar el caracter CR

y por el contrario si deseo descargar codigo debo agregar el caracter CR

Persona B Linux o Mac

Si persona B sube algo al repositorio Git no deberia realizar ninguna accion sobre el codigo .pero si por alguna razon debedo al editor de texto que esta utilizando o porque introdujo manualmente el caracter de CR , Git debe ser capaz de eliminarlo y no dejar subir el caracter

para esto el valor de Core.autocrlf debe tener el caracter input

Cuenta con el caracter LF

Y de esta forma puedo dejar configurado como Git va a tratar los saltos de linea ya sea en Windows en Linux o Mac

Como se lo configura en Git?

Core.autocrlf esta propiedad tiene que tener el valor de true para bajar contenido

Ultima configuracion que me falta

`git config —global core.autocrlf true` (debido a que me encuentro en windows

`git config —global core.autocrlf input` (si me encontrara en Linux o Mac)

Comando para ver los comandos

`git config -h`

Comandos básicos de Git BASH

`ls` : permite listar todos los archivos y carpetas en un directorio que se encuentra uno

`pwd` : permite saber donde se encuentra uno en la computadora

En windows se llaman carpetas

En Linux o Mac se llaman directorios

`cd nombrecarpeta` : me permite moverme

(En este momento cree una carpeta llamada “carpeta ejemplo”

Use `cd carpeta ejemplo` para moverme y `pwd` para saber en que ruta me encuentro

Y si me quiero salir de este directorio?

`cd ..` (es para salir un paso atras en la jerarquia)

Como creo un nuevo directorio para almacenar el codigo de mi aplicacion

`mkdir nombredirectorio`

Como inicializar la carpeta como si fuera un proyecto

`git init`

El mensaje que me deberia salir es

Initialized empty Git repository in

`C:/Users/crist/OneDrive/Escritorio/workspace/miweb/.git/`

El que finalice con `.git` significa que esta oculto y si quiero verlo debo con (`ls -a`) ya que no basta con la instruccion de `ls`

ls -a aparece esto ./ ../ .git/

los .. me hace dar un paso atras en la jerarquia

. me deja en el mismo directorio

Entro en con cd .git

Ingreso ls y me aparecen los detalles de implementacion

Detalle de implementacion

Aqui se van a almacenar las distintas versiones de mi codigo, las ramas, los comits absolutamente todo

La carpeta .git es ignorada por todos los repositorios no pasa a un servidor central o no se comparte

Flujo de trabajo que debe seguir

Trabajando en el computador puedo agregar o modificar todos los archivos que quiera, pero eso no significa que se vayan a agregar inmediatamente a un repositorio

git add selecciona los archivos que pasan a una etapa que se llama stage (es una etapa intermedia para que nosotros podamos indicar cuales son los cambios que se efectuaron para que pasen al repositorio)

Que pasa si no quiero pasar estos cambios al repositorio, porque se pueden sacar los elementos de esa etapa

git commit este comando permite pasar los archivos de la etapa stage a archivos de commit

git push este comando manda los cambios comprometidos a un servidor en la nube

En caso de que quiera **eliminar** un archivo tengo que utilizar el comando git add asi agrego un archivo a la etapa stage luego pasa a la etapa commit y por ultimo termina por ser eliminado del repositorio

Como abrir el editor de texto en la carpeta que estoy

code .

git status muestra el estado actual del repositorio

Untracked files (son archivos que git no esta siguiendo)

git add nombre del archivo

git add .txt (agrega todos los archivos de una característica)

git add .(agrega todos los adocumentos) usarlo siempre es una mala practica solo cuando este 100%seguro de que todo los archivos comprometidos quiera marcar

Cuales son las dos formas de comprometer

Comprometer es la traduccion de commit

1. git commit -m "Mensaje que tenga sentido referido a el accionar que se realiza"

los archivos en stage se han comiteado

2. git commit

abre el editor de texto donde me muestra el archivo que tuvo un cambio, luego en la primera linea puedo agregar el mensaje relevante para el commit

Como eliminar un archivo

rm nombre archivo

git status

y me aparece el archivo borrado en rojo debo agregarlo a la etapa de stage

git add nombre archivo

esto lo agrega al stage

luego lo commiteo

git commit -m "mensaje de referencia"

git status me aparece que se elimino el archivo

Todo este procedimiento es algo repetitivo por que existe un comando que me ahorra 1 paso

```
git rm nombreadarchivo
```

ahora lo comiteo

Si quisiera sacar algun cambio que pase a stage

```
git restore --staged nombreadarchivo
```

 que se llevo a stage y ya no se quiere ahi

ahora si quiero recuperar el archivo tengo que

```
git restore nombreadarchivo
```

 y listo

Como cambiar de nombres los archivos

```
mv nombreadarchivo nuevo nombre de archivo
```

ahora tengo que commitear o comprometer los cambios

La otra forma para que pueda renombrar y agregar directamente a stage los cambios

```
git mv nombre archivo nuevo nombre archivo
```

Como se pueden ignorar archivos

El propósito de esto es que estos archivos no sean incluidos nunca

A veces se busca tener archivos de configuración que sean específicos solamente para mi maquina

Por ejemplo: variables de entorno

Supongo trabajo con una base de datos y esa base de datos la tendré instalada en local

Usuarios contraseñas van a ser completamente diferentes a los de producción

Creo un archivo .env (nombre generico para las variables de entorno)

Creo un archivo .gitignore (aqui todas rutas o archivos que quiero ignorar y no vayan al repositorio)

agrego dentro .env

```
git status
```

me parece en rojo **.gitignore**

Lo paso a stage y luego lo commiteo

comando

`git status -s`

M .gitignore (la M en rojo debido a que no esta siendo contemplado por git)

M archivo1.txt(la M en rojo debido a que no esta siendo contemplado por git)

?? archivo2.txt(la ?? en rojo debido a que no esta siendo contemplado por git)

M significa que el archivo a sido modificado

?? significa que no ha sido agregado para que git le pueda hacer seguimiento

`git add archivo2.txt`

Lo que me aparece es esto:

M .gitignore

M archivo1.txt

A archivo2.txt (La A en verde significa que se esta agregando)

Comando

`git diff` (permite ver todos los cambios que estoy dispuesto a agregar a la etapa de stage)

`git diff —staged` (me permite ver los cambios que se encuentran en nuestra etapa de staged)

Esto es util ya que me permite ver los cambios que yo quiero comprometer y no tener que ver absolutamente todo

Viendo el historial

`git log` (me sirve para ver informacion acerca de los commit ,nombre , correo y un mensaje de commit

En la misma se puede navegar con el espacio

Otra alternativa es:

git log —online

Este comando genera un historial con un pequeño hash que sirve como identificador de ese commit

Muestra los mensajes puestos en los commits

Ramas en git o branches

Hasta el momento se trabajó de manera lineal

Cada vez que se realiza un cambio este se va a colocar por encima en el historial es decir

Siempre se colocan cambios con respecto a algo que se encuentra en el pasado por ende podemos seguir agregando cambios, pero ¿qué ocurre cuando más de un desarrollador trabaja?

Lo que se quiere es tomar una pequeña bifurcación o salir de la parte del historial de manera que se pueda seguir trabajando en nuestro propio código y una vez terminada la funcionalidad o terminado el error que tengo intentar volver a la rama principal

Luego de haber trabajado con la rama puedo solicitar un merge a la rama de main (la rama principal)

Git branch es el comando que me dice en qué rama me encuentro

Como crear una rama

git checkout -b nombre_rama (o algunas empresas prefieren usar features/nombre de la funcionalidad)

En el ejemplo se usará

git checkout -b ramab

Luego indica que se ha cambiado a la rama b

Ahora puedo realizar cambios sobre los archivos y voy al editor de texto

Comando para ver el contenido de un archivo

cat nombre_archivo

¿Qué ocurre si me cambio de rama?

`git checkout master`

`car nombreadarchivo`

En el ejemplo se observan líneas de menos que corresponden a un momento de las ramas que no se las había modificado

Como traer los cambios de una rama a otra

es importante que no este en la rama que trato de traer

`git merge nombredelarama`

De esta forma cada quien crea su rama implementa la funcionalidad y luego la mezcla o hacer un merge con la rama master

Ahora vamos a la pagina de Git

Apreto en new para realizar un nuevo repositorio

Le agrego un nombre

Y crear repositorio

Luego

Me da posible acciones realizables compio la siguiente

```
git remote add origin https://github.com/CristhianSZ/miwed.git
```

Lo que manifiesta el código anterior es

`git remote` sirve para indicar si es que vamos a tener un servidor remoto en el cual nosotros podemos subir nuestros cambios

`add origin` sirve para indicar de donde vamos a obtener nuestro código y también a donde tenemos que subir los cambios que realicemos

indicamos al url de donde se encuentra almacenado nuestro código