

Git para trabajo colaborativo

Instalacion de git

Desde la consola ejecutar

```
sudo apt update && sudo apt install git
```

Ahora debemos acceder a nuestra cuenta en github, gitlab, bitbucket o similar, crear un repositorio y clonarlo donde queramos tener nuestro proyecto.

- Ahora dependiendo de como se configuro el repositorio se deberan generar claves ssh, agregar los colaboradores al repo, etc. Antes de hacer los siguientes pasos
-

- Al ser la primera vez que utilizamos git nos pedira unos datos para las configuracion:

```
git config --global user.email "Correo"  
git config --global user.name "Nombre"
```

Clonamos el repositorio

Definicion de comandos

- clone = Copia un repositorio a el directorio seleccionado
- add = Agrega los archivos al commit
- commit = Prepara la subida de los archivos con su descripcion
- push = Sube el commit a la rama seleccionada
- pull = Descarga los commit desde la rama seleccionada
- checkout = Nos cambia de rama
- branch = Crea una rama a partir de la que esta seleccionada
- merge = Combina las ramas

```
git clone <aqui va el enlace, sin los simbolos estos>  
git clone https ://github.com/author/repositorio.git
```

Lo siguiente es crear el proyecto a trabajar y hacer el push inicial.

```
git add *
git commit -a -m "Descripcion breve"
git push origin master
```

Creamos las ramas (Branch)

- Esto se opcional pero muy recomendable Una rama es una bifurcacion de la rama seleccionada, en nuestro caso de la rama maestra (master). Nos posicionamos en la carpeta de nuestro proyecto y ejecutamos.

```
git branch -b <nombre de la rama>
git branch -b desarrollo
git add .
git commit -a -m "Mensaje del comit"
git push origin desarrollo
```

Esta rama nos permitira trabajar y hacer commits sin alterar la rama master, ya que la master es la que se utiliza para mostrar al cliente, profe, jefe y siempre debe estar funcionando. En cambio la rama de desarrollo es donde todo el equipo sube su trabajo y estos cambios pueden alterar otros y el producto puede fallar, por eso es bueno siempre tener una rama de desarrollo, que obviamente puede tener otro nombre.

- Esto tambien es opcional pero muy recomendable

Si trabajamos en equipo es bueno tener una rama en local para hacer nuestros cambios y una vez funcionen los subimos a la rama de desarrollo, para esto hacemos.

```
git branch -b <nombre de la rama>
git branch -b local_s384
```

- Al ser una rama local no es necesario subirla a git.

Cuando tengamos listo un desarrollo hacemos

```
git add *
git commit -a -m "Mensaje"
git checkout desarrollo
git merge local_s384
```

Con esto actualizamos la rama desarrollo con nuestros cambios

Cambios de desarrollo a local

Ahora si un compañero de equipo hizo un push a desarrollo, nosotros los bajaremos para tenerlos en local. Nos posicionamos en desarrollo.

```
git checkout desarrollo
git pull origin desarrollo
git checkout local_s384
git merge desarrollo
```

- Con esto actualizamos la rama local a la de desarrollo

Actualizando la rama desarrollo

Caso contrario si ya tenemos un cambio listo en la rama local la queremos mezclar con desarrollo para subir los cambios

```
git add *
git commit -a -m "Breve descripcion"
git checkout desarrollo
git merge local_s384
git push origin desarrollo
```

- No es necesario agregar los archivos nuevamente o escribir el mensaje, ya que el commit viene listo desde local_s384, solo nos falta hacer push desde la rama indicada

Instalacion de sublime merge

Sublime merge es una gui que nos permite gestionar nuestros repositorios, es muy intuitiva y util, pero nunca esta de mas la consola, yo la prefiero sobre las gui para git. dejo el enlace de referencia [Sublime Merge](#)