

Comandos GIT

----- COMANDOS USADOS EN EL PC PARA CREAR Y MODIFICAR ARCHIVOS -----

pwd (para saber la ruta en la que estoy)

ls (muestra todas las carpetas y archivos donde estoy)

ls -l (lista todos los archivos y los permisos, el usuario que lo creo, fecha y hora)

clear (para limpiar pantalla)

ls -a (muestra carpetas normalitas y muestra carpetas y archivos ocultas)

ls -al

mkdir (crea carpeta)

cd .. (para bajar un nivel)

cd nombre_carpeta (Para ingresar a una carpeta)

touch (para crear un archivo)

nano miArchivo.txt (para modificar el archivo [ctrl+x Y enter])

cat mi Archivo.txt (para ver lo que contiene el archivo)

history (para ver los comandos ejecutados)

rm miArchivo.txt (elimina el archivo)

rm -r (elimina la carpeta)

----- COMANDOS PARA ASIGNAR O VER QUIEN ESTA CREANDO CADA COMMIT -----

git config --list (Para saber quien hace los commit)

git config --global user.name "nombre" (para cambiar un nombre)

git config --global user.email "correo" (para cambiar el email)

----- COMANDOS PARA REALIZAR CAMBIOS EN LOS ARCHIVOS USANDO GIT-----

git init (inicializar)

git add nombreArchivo (se sube solo ese archivo a un almacenamiento temporalmente - staging)

git add . (se suben todos los archivos a un almacenamiento temporalmente - staging)

git rm --cached nombreArchivo (elimina un archivo del almacenamiento temporal, pero lo conserva en tu directorio de trabajo-)

git status (hace un seguimiento de los archivos - se ven los que están rastreados y los que no)

git commit -m "se agrega comentario relacionado específicamente a los cambios que realiza" (aquí lo sube a repositorio)

git log (rastreo de los commit -lo que muestra en color amarillo es el tag el cual es el identificador del commit, se copian los 5 primeros dígitos del tag cuando se necesiten y donde dice [HEAD - > master] es donde estamos ubicados en ese momento)

ESC : q (comando que se escribe Para salir del log)

----- COMANDOS PARA MOVERSE ENTRE COMMIT -----

git show (solo se ve los cambios que se han hecho del ultimo commit)

git show tag (muestra información relacionado de algún commit específico)

git diff tag1 tag2 (para ver los cambios que se realizaron entre un commit y otro, lo que muestra en rojo son los cambios que se hicieron, por lo general siempre se coloca de primero el ultimo tag que se desea comparar)

git reset --hard tag (se ubica en el tag seleccionado y esconde los cambios realizados- Esto permite moverse entre commit)

git reflog (historial de los commit que se han eliminado, puede servir para buscar el tag de cada commit realizado y moverme hacia dicho tag usando el comando anterior)

----- CREAR y UNIR INFORMACIÓN DE LAS RAMAS -----

Tipos de ramas:

Maxter: Es la principal

Hotfix: Reparar errores en caliente

Release: Cuando se van a unir las ramas a la Maxter, (testeo)

Develop: Es la rama de desarrollo

Feature: Significa características se refiere a la usabilidad e interfaces

`git branch` (muestra las ramas del repositorio)

`git branch nombreRama` (para crear ramas)

`git checkout nombreRama` (para cambiar de rama)

`git branch -d nombredelarama` (para eliminar una rama, pero no puede estar parado en la misma rama)

OJO cuando se cree un archivo en cualquier rama es necesario realiza el commit a ese archivo

`git merge nombredelarama` (se une uno por uno las ramas que se desean a una misma rama, recordar que la rama principal siempre será la rama master)

----- PARA SUBIR REPOSITORIO EN GITHUG CUANDO YA SE TIENE CREADO EL REPOSITORIO EN EL PC -----

`git branch -M main` (Cambia de rama master a main- para que se pueda subir)

`git branch -M cambiarnombrealarama` (permite cambiar nombre a la rama la cual debe ser main)

`git remote add origin https://github.com/xxxxxxx/xxxxxxx.git` (direccion a donde se van a subir)

`git push origin main` (se sube al github los archivos)

`git push origin nombredeotrarama` (para subir otra rama al repositorio de github)

`git pull origin nombredelarama` (Descarga una rama)