# git Git: Fundamentos, líneas de código y funcionamiento

Configuración:

* $ git config --global user.name "nombre": sirve para colocar nuestro nombre de usuario para identificarnos
* $ git config --global user.email "correo": sirve para colocar nuestro correo de usuario para identificarnos, es importante que sea el mismo correo con el perfil de nuestro git/github para el repositorio.

Comandos básicos:

* Ls: recorre las carpetas o archivos existentes en una rama ->> pc/pipe/escritorio (mostrara archivos/carpetas o cualquier cosa de la ruta anterior)
* Cd (nombre) : permite ingresar a una carpeta
* Pwd: muestra la ruta actual
* Mkdir (nombreCarpeta): crea una carpeta

Comandos básicos Git:

* Git init: inicia un repositorio de git
* Git status: revisa el estado de los archivos dentro del repositorio
* Git add (nombreArchivo): agrega un archivo al repositorio con el fin de mandarlo a almacenar en el repositorio
* Git restore: reestablece un archivo cambiado al estado original al momento de agregar
* Git commit -m “mensaje”: se almacena los cambios y se queda listo para subir al repositorio en la nube
* Git diff: muestra los cambios dentro de los archivos siempre y cuando se hayan almacenado en un add antes, es parecido al status
* Git log:listado de los commits
* Git log –oneline: lo mismo pero mas ordenado
* Git log –oneline –all :lo mismo pero trae contenido tanto local como remoto (commits)
* Git tag “nombreTag”: coloca un nombre clave en un commit
* Git switch nombreRama: cambia de rama, ojo, SOLO CAMBIA ENTRE ELLAS, NADA MAS
* Git stash: te permite guardar temporalmente cambios no confirmados
* Git stash pop: restaura el último stash aplicado y lo elimina de la pila
* Git fetch: descarga los cambios más recientes del repositorio remoto sin fusionarlos con tu rama actual
* git fetch –prune: lo mismo que fetch solo pero lo fuerza
* Git checkout (nombreArchivo): reestablece el archivo a su ultima versión, ósea, elimina modificaciones sin registrar
* Git reset: saca del estado “add” los archivos que estaban en ese estado, no borra los últimos cambios
* Git reset –hard: los saca y elimina el ultimo contenido
* Git reset –hard (numero de commit): nos mueve de commit (posiciona el head en otro lado) y los demás que estén por encima del head nuevo, se eliminan o dejan de existir

Comandos de ramas Git:

* Git Branch: nos dice en que rama estamos (si existen otras, saldrá el nombre, casi siempre será master en tu caso)
* Git Branch -a : lista de las ramas
* Git checkout nombreRama: Genera una nueva rama
* Git checkout -b nombreRama: se genera y mueve a la rama nueva
* Git checkout nombreRamaExistente: te mueves a la rama mencionada
* Git push origin (nombreRama): envia los archivos de x rama
* Git checkout (serial de un commit o al tag) te mueve al commit en cuestión junto con los archivos de ese commit
* Git merge (nombreRama) -m “mensaje”:combina los contenidos de las ramas, ósea, en este caso,nombreRama compartira sus archivos con master/ existirán momentos en que hayan conflictos, debes decidir con que cambios te quedas tu.
* Git pull (nombreRama): trae los nuevos cambios a x rama

Enlazar con repositorio remoto:

1. Crear repositorio en github
2. Git remote add… : enlazamos el repositorio remoto con nuestro proyecto
3. Git push origin master: mandamos todo nuestro avances a github desde la rama master
4. Revisar el repositorio remoto a ver si esta todo ok

Comandos frágiles:

* Git Branch -d (nombreRama): elimina una rama de forma local
* Git push origin –delete (nombreRama): elimina una rama de forma permanente (repositorio)
* Git clone (url en https): clona el repositorio en local

Documentacion:

* https://stackedit.io/app#

# Orden esperado

# Configuramos nuestros datos de usuario en Git

git config --global user.name "TuNombre" && git config --global user.email "tuemail@ejemplo.com"

# Creamos una nueva carpeta para el proyecto y entramos en ella

mkdir nombreProyecto && cd nombreProyecto

# Inicializamos el repositorio Git local

git init

# Creamos o agregamos archivos al proyecto (esto puede hacerse manualmente o ya pueden existir)

# Verificamos el estado actual del repositorio

git status

# Añadimos todos los archivos al área de preparación (staging)

git add .

# Hacemos el primer commit con un mensaje

git commit -m "Primer commit"

# Creamos un repositorio vacío en GitHub y lo enlazamos con el proyecto local

git remote add origin https://github.com/tuUsuario/tuRepositorio.git

# Subimos el contenido al repositorio remoto (rama master o main, según corresponda)

git push origin master

# (A futuro) Para traer cambios desde GitHub usamos:

git pull origin master

# (A futuro) Para agregar nuevos cambios:

git add . && git commit -m "mensaje del cambio" && git push origin master

# (Opcional) Para trabajar en ramas: creamos y nos movemos a una nueva rama

git checkout -b nombreRama && git push origin nombreRama

# (Consulta de ramas y commits)

git branch -a # Lista todas las ramas

git log --oneline # Muestra el historial de commits resumido

# (Eliminación de ramas)

git branch -d nombreRama # Elimina rama local

git push origin --delete nombreRama # Elimina rama en GitHub