

## CONFIGURACIÓN

Configuración de la información de usuario utilizada en todos los repositorios locales

- **git config --global user.name "[firstname lastname]"**  
establezca un nombre que sea identificable para crédito cuando revise el historial de versiones
- **git config --global user.email "[valid-email]"**  
establecer una dirección de correo electrónico que se asociará con cada marcador de historial
- **git config --global color.ui auto**  
configurar automáticamente el coloreado automático de la línea de comandos para Git para facilitar la revisión

## CONFIGURACIÓN E INIT

Configuración de la información del usuario, inicialización y clonación de repositorios

- **git init**  
inicializa un directorio existente como Repositorio de Git
- **git clone [url]**  
recuperar un repositorio completo desde una ubicación alojada a través de URL

## ESCENARIO E INSTANTÁNEA

Trabajar con instantáneas y el área de preparación de Git estado de git mostrar archivos modificados en el directorio de trabajo, preparados para su próxima confirmación

- **git add [archivo]**  
agregar un archivo como se ve ahora en su próxima confirmación (escenario)
- **git reset [archivo]**  
deshabilite un archivo mientras conserva los cambios en el directorio de trabajo  
git diff diff de lo que se cambió pero no se escenificó  
git diff --staged diff of w que está organizado pero aún no confirmado git commit -m "[mensaje descriptivo]" confirma tu contenido preparado como una nueva instantánea de confirmación

## BRANCH & MERGE

Aislamiento del trabajo en sucursales, cambio de contexto e integración de cambios git branch list tus sucursales. aparecerá un \* al lado de la rama actualmente activa

- **git branch [branch-name]**  
crea una nueva rama en el commit actual git checkout cambia a otra rama y búscala en tu directorio de trabajo

- **git merge [branch]**

fusiona el historial de la rama especificada en el registro actual de git muestra todas las confirmaciones en el historial de la rama actual

### **INSPECCIONAR Y COMPARAR**

Examen de registros, diferencias e información de objetosregistro de git mostrar el historial de confirmaciones para la rama actualmente activa

#### **git log branchB.branchA**

mostrar las confirmaciones en branchA que no están en branchB

#### **git log --follow [archivo]**

mostrar las confirmaciones que cambiaron el archivo, incluso a través de cambios de nombre

#### **git diff branchB...branchA**

mostrar la diferencia de lo que está en branchA que no está en branchB

#### **git show [SHA]**

mostrar cualquier objeto en Git en formato legible por humanos

## **OTROS COMANDOS**

#### **git remote add origin [url]**

conecta el repositorio local con github

#### **git remote add origin <https://token@URL>**

se usa cuando se configura el token en github

#### **git remote -v**

status del repositorio en github

#### **git push origin main**

subir archivos a github

#### **git push -u origin main**

crea el repositorio en github aunque no exista

#### **git clone -token-@-url del repositorio-**

descarga los repositorios desde github la primera vez y que no los tenga. token se usa solo si se tiene

#### **git pull origin main**

actualiza los archivos del repositorio que ya existe

#### **git log**

muestra el historial de lo commits hechos en el repositorio

#### **git branch**

Enumera todas las ramas de tu repositorio, es similar a git branch --list.

#### **git branch [nombre de la branch]**

Crea una nueva rama llamada <branch>.

**git branch -d [nombre de la branch]**

Elimina la rama llamada <branch>. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.

**git branch -D [nombre de la branch]**

Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.

**git branch -M main** \*\*\*renombra la branch principal a main

**git checkout [nombre\_rama]**

Para moverse de una rama a otra, generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si

tenemos cambios, para cambiarnos de rama, debemos:

1. Eliminarlos (deshaciendo los cambios).
2. Confirmarlos (haciendo un git commit).

**git push origin [nombre de la branch]**

subir a github en una rama

**git pull origin [nombre de la branch]**

trae los archivos guardados de una rama en github

**git merge [nombre rama que quiero traer los cambios]**

une los commits de las ramas, se posiciona en la rama que quiero hacer el cambio

**git status**

muestra status del repositorio

**git add .**

agrega todos los archivos en el repositorio local

**git commit -m [descripcion]**

crea una version, un commit nuevo