

# Comandos Git

## Cheatsheet

**git init** - Crea un nuevo repositorio.

**git config user.name <nombre de usuario>** - Agrega nuestra identidad.

**git config user.name** - Muestra el nombre del usuario.

**git config user.email <[email@email.com](mailto:email@email.com)>** - Agrega nuestro email.

**git config user.email** - Muestra el email del usuario.

**git remote add origin http://...** - Apunta al repositorio remoto.

**git remote rm <nombre\_git>** - Desvincula un repositorio remoto.

**git remote -v** Muestra si estoy conectada a github.

**git add nombre\_archivo\_directorio** - Añade un archivo específico

**git add . / git add --all** - Añade todos los archivos

**git commit -m "mensaje del commit"** - Comitea los cambios hechos.

**git commit --amend -m "Mi nuevo mensaje"** - Cambia los mensajes del commit.

**git push origin main** - Envía los cambios al repositorio remoto.

**git push origin <branch>** - Envía los cambios a una bran específica.

**git push** - Los otros push no necesitan otras informaciones.

**git pull** - Descarga los archivos con cambios existentes en el repositorio.

**git fetch** - Obtener los cambios, pero no aplicarlos a la branch actual.

**git status** - Realiza un seguimiento de los estados de los archivos.

**git clone http://...** - Clonamos el repositorio en nuestro compu.

**git log** - Muestra el historial

**git log --<ruta del archivo>** - Muestra el historial de un archivo específico.

**git log --<author=usuario>** - Muestra el historial de un usuario en particular.

**git rm <archivo>** - Elimina un archivo.

**git rm -r <directorio>** - Remueve el directorio y los archivos que contiene.

**git reset HEAD <archivo>** Deshace el cambio local en el área de trabajo temporal, debe usarse cuando el archivo ya ha sido añadido en el área temporal.

**git checkout <nombre\_archivo>** Permite realizar el cambio de directorio.

**git checkout -- <archivo>** Solo debe usarse mientras el archivo no se haya añadido todavía al área de trabajo temporal.

**git branch** - Muestra todas las ramas del repositorio.

**git branch -M main** - Cambia el nombre de la main.

**git branch <Branch>** - Crea una nueva branch.

**git branch -d <Branch>** - Elimina la rama del <branch>. Git evita que se elimine si no se han fusionado los cambios con la ppal.

**git branch -D <Branch>** - Fuerza la eliminación de la rama aún si no se han fusionado los cambios.

**git branch -v** Lista branches con información de los últimos commits.

**git checkout <Branch>** - Cambia a una branch existente.

**git checkout -b <Branch>** - Crea una nueva branch y apunta a ella.

**git merge <Branch>** - Fusiona las branches - Para fusionar, debe estar en la branch que debe recibir los cambios.

**git branch --merged** lista branches que ya se han fusionado con la main.

**git branch --no-merged** listar branches que no se han fusionado con la main.

**git merge --abort o git reset --merge** Cuando tenemos problemas con la fusion y queremos deshacerla.

**git reset HEAD** Cuando queremos volver a un commit anterior, si queremos volver a más de un commit, debemos poner el número de commits después de HEAD. Ejemplo: HEAD~2.

**git config --global --unset user.name "nombre de usuario"** -

Eliminar todos los registros que se refieren al usuario.

**git config --global --unset user.email [email@email.com](mailto:email@email.com)** - Eliminar todos los registros que se refieren al e-mail

**git help** - Muestra los comandos y sus funciones.

**git help <comando>** - Muestra las funciones de un comando en

específico.