**Aprendizaje de Git**Fuente:<https://github.com/elliotgaramendi/learning-git>

Aprendizaje de Git desde 0. En este repositorio muestro como voy aprendiendo poco a poco los comandos de git.

Espero aprender mucho.

[](https://camo.githubusercontent.com/79fb609c802ed989ed0978690e279c8ccd05f87fa16ee2007df50ae5add900d2/68747470733a2f2f6d69726f2e6d656469756d2e636f6d2f6d61782f3335322f312a71523678703639545a53533944765f5a4278547731772e6a706567)

**Comandos empleados en el aprendizaje**

**Muestra la versión de git instalada**

git --version

**Brinda ayuda sobre los comandos**

git help <comandos>\*

**Repositorio local**

**Configura el usuario**

git config --global user.name <usuario>

**Configura el email**

git config --global user.email <correo>

**Lista la configuración de git**

git config --list

**Inicia un nuevo repositorio**

git init

**Lista el estado de los archivos**

git status

**Agrega todos los archivos pendientes de cambios**

git add --all

**Agrega todos los archivos pendientes de cambios desde la carpeta actual**

git add .

**Guarda en el repositorio local todos los archivos agregados**

git commit -m "<mensaje>"

**Abre un editor y permite modificar el último commit (Reemplaza el último commit)**

git commit --amend

**Crea un tag**

git tag <tag> -m "<mensaje>"

**Lista tags**

git tag

**Borra un tag en específico**

git tag -d <tag>

**Hace un tag en un commit anterior**

git tag -a <tag> <commit> -m "<mensaje>"

**Mostrar información del tag**

git show <tag>

**Deshace el último commit, eliminando los cambios del área de preparación.**

git reset

**Deshace el último commit, conservando cambios en el área de preparación**

git reset --soft

**Cambiamos a un commit en específico perdiendo todos los cambios posteriores a este.**

git reset --hard <commit>

**Muestra los commit realizados hasta el momento**

git log --oneline

**Lista todos los commits de todas las ramas de forma gráfica tomando como base la rama actual**

git log --oneline --graph --all

**Cambia a un commit en específico con todos sus cambios**

git checkout <commit>

**Muestra los cambios dos commits**

git diff <commit> <commit>

**Crea una nueva rama**

git branch <rama>

**Nos muestra en que rama estamos y lista las demás**

git branch

**Nos movemos de la rama actual a una específica**

git checkout <rama>

**Crea y nos movemos a la nueva rama**

git checkout -b <rama>

**Renombra la rama actual**

git branch -m <rama>

**Eliminar una rama**

git branch -d <rama>

**Permite juntar dos ramas. Trae los cambios de la rama especificada a la rama actual**

git merge <rama>

**Permite juntar dos ramas, pero las mantiene. Genera un commit del merge en la rama actual**

git merge --no-ff <rama>

**Permite traer los commits de otra rama a la rama actual reorganizan los commits**

git rebase <rama>

**Repositorio remoto**

**Para subir nuestro proyecto debemos crear un repositorio remoto. Al crearlo nos mostrará una serie de comandos para subir el proyecto. Te pedirá el usuario y contraseña de tu cuenta de git si aún no lo registras.**

**Vincular repositorio remoto con repositorio local**

git remote add origin <url>

**Cambiar URL del repositorio remoto**

git remote set-url origin <url>

**Nos muestra en que repositorio estamos enlazados remotamente.**

git remote -v

**Sube los cambios del repositorio local al remoto y especifica la rama principal**

git push -u origin <rama>

**Sube los cambios del repositorio local al remoto de la rama principal**

git push

**Sube los cambios del repositorio local al remoto de una rama específica**

git push origin <rama>

**Eliminar una rama remota**

git push origin --delete <rama>

**Sube todos los tags locales al remoto**

git push --tags

**Eliminar un tag remoto**

git tag -d <tag>

git push origin :refs/tags/<tag>

**Descarga los cambios del repositorio remoto y actualiza el local en la rama por defecto**

git pull

**Descarga los cambios del repositorio remoto y actualiza el local en una rama específica**

git pull origin <rama>

**Clona un repositorio remoto en la rama por defecto**

git clone <url>

**Clona un repositorio remoto en una rama específica**

git clone --branch <rama> <url>