


Introducción a GIT


Inicializar git:

 git init


Versión de git

 git --version


Configurar correo

 git config — global user.email "aca va tu correo"


Configurar nombre

 git config --global user.name "aca nombre/usuario"


Añadir un archivo

 git add index.html


Añadir más de un archivo

 git add index.html styles.css main.js


Añadir todos los archivos

 git add .

Añadir todos los archivos según su extensión

 git add *.js


Añadir todos los archivos de una carpeta:

 git add img/


Ver los archivos modificados:

 git status

Ver los archivos modificados y en que rama estamos trabajando:

 git status -sb

Ver los archivos modificados (con menos información):

 git status -s


Crear un registro de los cambios realizados:

 git commit -m 'first commit'

Ver todos los commits realizados:

 git log


Ver todos los commits con información básica:

 git log --oneline

Ver todos los commits con las ramas:

 git log --graph

Ver todos los commits realizados de otra forma:

 git log --oneline --decorate --all --graph

Ver todas las modificaciones entre el commit anterior y donde estamos parados ahora:

 git diff

Ver todos los archivos que estén en el Staged

 git diff --staged


Recuperar archivos:

 git checkout .

Agregar al staged y hacer el commit en un solo comando:

 git commit -am 'README actualizado'


Modificar el mensaje del commit:

 `git commit --amend -m 'Actualizamos el mensaje!'`


Lista de commits originados:

 `git reflog`


Renombrar archivos:

 `git mv styles.css estilos.css`


Eliminar archivos:

 `git rm estilos.css`

Crear una rama (branch):

 `git branch miRama`

Ver las ramas de nuestro repositorio:

 `git branch`

Pasar a trabajar en una rama en especifica:

 `git checkout miRama`

Crear y moverse a una rama en un solo comando:

 `git checkout -b miRama`

Volvemos a la rama master

 `git checkout master`


Unir rama con master:

 `git merge miRama`

Borrar rama:

 `git branch -d miRama`

Crear tag:

 `git tag -a v1.0.0 -m "Versión 1.0.0"`


Insertar un tag en un commit:

git tag -a v0.1.0 43809d4 -m 'Primera versión'

Ver los tags:

 git tag


Eliminar un tag:

 git tag -d v0.1.0

Crear un stash:

 git stash

Ver los stash:


 git stash list

Github


Una vez que creado el repositorio, lo añadimos como servidor remoto:

 git remote add origin tuRepositorio.git


Ver las fuentes remotas de nuestro repositorio:

 git remove -v


Push a github:

 git push -u origin master

Subir los tags

 git push --tags

Ignorar archivos y carpetas:

 .gitignore: con los nombres de los archivos y carpetas que no queremos que se tengan en cuenta a la hora de subir el repositorio.


Actualizar o bajar nuestro repositorio de Github:

 git pull

Clonar un repositorio:

 git clone repositorio.git

Clonar repositorio en una carpeta específica

 `git clone tuRepositorio.git miProyecto`