

Comandos

GIT

- `git init`: Es como decir que vamos a empezar a utilizar en ese proyecto git
- `git add <file>`: Es para poder pasar los archivos del working directory al staging area (si se pone un punto en vez del nombre, agrega todos los archivos)
- `git status`: Es para ver en que estado estan los archivos
- `git diff <file>`: Es para ver las diferencias entre el archivo ya cargado y el nuevo
- `git commit`: Es para pasar los archivos del staging area al repositorio
- `git reset`: Es para desañadir un archivo que tengamos en el staging area, vuelve al working directory
- `git branch`: Muestra las versiones que hay del proyecto
- `git branch <nombre>`: Crea una nueva version del proyecto (serian como carpetas con diferentes versiones por ejemplo)
- `git branch -d <nombre>`: Elimina la rama
- `git branch -D <nombre>`: Fuerza la eliminacion de la rama, por mas que hayan cambios sin fusionarse
- `git checkout <nombre>`: Cambia de branch al especificado
- `git push`: Es para subirlo a un repositorio remoto
- `git pull`: Es para traer los cambios que hicieron los otros desarrolladores
- `git fetch`: Obtiene los cambios pero no los aplica a la rama actual
- `git clone`: Es para hacer una copia del servidor donde estan los archivos, a la computadora
- `git status`: Es para ver las snapshots
- `git clone <url>`: Copia todos los archivos desde el repositorio
- `git restore <nombre>`: Reustaura el archivo hasta el ultimo add
- `git merge <rama>`: Fusiona en la rama en la que estamos parados, con la rama que elegimos (no borra las ramas)
- Crear un archivo llamado `.gitignore`, dentro de este, van a ir los nombres de los archivos o carpetas que queremos que git ignore

GITHUB:

- `git remote add origin <url>` Agregamos la url de donde va a estar el repositorio
- `git push -u origin master` Subimos los archivos a la nube