

Comandos

Obtener ayuda sobre un comando: git clone --help // git commit --help

git reset

Si lo que queremos es deshacer todos los cambios que hemos hecho desde el último commit y quedarnos con el directorio "limpio de cambios", ejecutaremos: **git reset --hard**

Usando **git reset** en conjunto al comando anterior **git clean**, que nos quedará el directorio de trabajo igualito a lo que había hasta el último commit. Si queremos volver a commits previos, borrando los cambios realizados desde ese commit, podemos utilizar el siguiente comando:

git reset --hard [commit ID]: regresa hasta el commit de ID

Listar ramas remotas: git branch -r

git diff

Sirve para ver los ficheros modificados y sus diferencias desde el último commit. Aunque lo podemos usar para ver las diferencias entre dos commits: **git diff hash-commit-más-antiguo hash-commit-más-nuevo** ...donde hash-commit-más-antiguo y hash-commit-más-nuevo son los identificadores de commit. Se ven después de hacer **git commit** o al listarlos con **git log**. Los puedes copiar y pegar o escribir sólo las primeras 5 letras. También se puede usar para ver los cambios entre dos ramas: **git diff rama-1 rama-2** ...nos dirá los cambios de la rama-1 que no están en la rama-2 con un símbolo - al principio de la línea y los de la rama-2 que no están en la rama-1 con un símbolo +.

Enlace Adalab

git add -A

git commit -m "texto"

git push

git pull

git branch

git branch XXX

git checkout XXX

git merge XXX (desde la rama principal)

git revert

git log (historico de commits)

git clone xxxx

git status