

Del tipo informativo

pwd / Mostrar el nombre de la carpeta en el que uno se encuentra situado (print working directory).

cd / Cambiar la carpeta de trabajo: Con este comando nos podemos mover entre diferentes directorios. Si queremos ir a un directorio en particular.

ls / Listar el contenido de directorios (list): Este comando lista los ficheros y carpetas.

Crear o eliminar carpetas

mkdir / Crear un carpeta (make directory).

rmdir / Borrar un carpeta (remove directory)

Crear o eliminar archivos

gedit / Crear y editar archivos de texto.

rm / Borrar archivos.

Copiar archivos y directorios

cp / Copiar un archivo o carpeta en el directorio especificado (copy)

cp -r / Copiar carpetas

Lmv / Mover un archivo o carpeta a un archivo o carpeta (move)

Comandos

`git branch -m master main`

`Ls -a`: repositorio `git`

`Git add`: archivos a agregar

`Git add.`: agrego todos

`Git status`

`Git commit -m ""`

`Git log`: ver `commits`

`Git log --oneline`: resumen

`Git checkout xxxxx`: volver al `commit`

`Git checkout + nombre rama`: cambiar rama

`Git checkout -b`: cambiar nombre y cambiar de rama

`Git checkout main`: retornar

`Git restore . : ctrl z`

`Git config --list`: usuarios que tengo

`Rm -rf .git`: carpeta oculta

- `git branch`

Enumera todas las ramas de tu repositorio, es similar a `git branch --list`.

- `git branch <branch>`

Crea una nueva rama llamada `<branch>`.

- `git branch -d <branch>`

Elimina la rama llamada `<branch>`. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.

- `git branch -D <branch>`

Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.

Git checkout

Para moverse de una rama a otra, se ejecuta el comando

- `git checkout nombre_rama`

Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si tenemos cambios, para cambiarnos de rama, debemos:

1. Eliminarlos (deshaciendo los cambios).
2. Confirmarlos (haciendo un git commit).