

`git init` --> Iniciar un repositorio local

`git config user.name [nombre de usuario GitHub]` --> Define la identidad con la que trabajaremos en el repositorio local.

`git config user.name` --> Verificar la identidad, nombre de usuario.

`git config user.email [email registrado en GitHub]` --> Define el correo electrónico de quien trabajara en el repositorio local.

`git config user.email` --> Nos permite verificar el correo electrónico.

`git config --global user.name [nombre usuario en GitHub]` --> Nos permite configurar para que todos los repositorios sean creados con ese usuario.

`git config --global user.email [email registrado GitHub]` --> Todos los repositorios son configurados a partir de este email.

`git status` --> Muestra el estado de los archivos del repositorio.

`git add [nombre de archivo]` --> Agrega el archivo especificado al repositorio.

`git add .` --> Agrega todos los archivos modificados o creados al repositorio.

`git restore --staged [nombre archivo]` --> Remueve un archivo del added.

`git commit -m "comentario"` --> Creamos un punto en la línea de tiempo en nuestro proyecto. En el mensaje entre comillas la idea es escribir de manera resumida el trabajo hecho hasta el momento.

`git log` --> Muestra un historial con todos los commits realizados y detallados.

`git log --oneline` --> Muestra el historial de commits en una línea c/u, resumidos.

`git log --graph` --> Ver el historial en forma gráfica o de línea de tiempo.

`git checkout [codigo de commit]` --> Me permite volver en el tiempo a ese commit seleccionado.

`git checkout main` --> Vuelvo nuevamente a la main, al último estado.

`git commit --amend` --> Modifico el mensaje del último commit.

`git remote add origin [URL del repositorio remoto]` --> Conectar al repositorio remoto, que está creado en GitHub al que pertenece la URL.

`git remote -v` --> Verificar que el repositorio remoto y local estén sincronizados.

`git remote remove origin` --> Eliminar la conexión al repositorio remoto.

`rm -rf .git` --> Eliminar el repositorio local, forzándolo.

`git clone [URL de repositorio GitHub]` --> Clona en nuestra pc el repositorio remoto (única vez).

`git push origin main` --> Envía los archivos del repositorio local (commiteados) al repositorio remoto.

`git pull origin main` --> descargará los nuevos archivos del repositorio remoto al local y actualizará aquellos archivos que hayan sufrido algún tipo de modificación.

`git branch -M [nombre]` --> Cambio el nombre de la rama

`git config --global init.defaultBranch main` --> Modifico el nombre de la rama principal por defecto a main.

`git branch` --> Enumera todas las ramas del repositorio.

`git branch [nombre de rama]` --> Crea una rama

`git branch -d [nombre de rama]` --> Elimina la rama del nombre seleccionado, excepto que tenga cambios sin fusionar con la main.

`git branch -D [nombre rama]` --> Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.

`git checkout [nombre de rama]` --> Para moverse de una rama a otra.

`git push origin [nombre de rama]` --> subir archivos commiteados a la rama digitada.

`git pull origin [nombre de rama]` --> Descargar los cambios realizados en la rama digitada.

`git merge [nombre de rama]` --> Mergea, une, el contenido de la rama digitada a la rama en la cual nos encontramos ejecutando el comando.