

## **Intro informática:**

Git init: genera un repositorio local vacío

Repositorio local: es el que tiene todos los archivos (que hayas guardado en él) en nuestra computadora.

Commits: son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor.

Git commit -m "" (mensaje de la línea del tiempo) (volver un guardado atrás)

Git log historial de los cambios del commit

Los commits van a ser nuestro historial de cambios que se fueron haciendo en el proyecto.

Agregar nombre de usuario: git config user.name "Nombre de usuario"

Agregar mail: git config user.mail "mail"

Sincronizar repositorio git remote add origin

git remote -v Ver si está bien

Git add . (agrega todos los archivos presentes en el repositorio)

Git status estado de archivos

Git push origin master : solicita a git mandar los archivos del repositorio local al remoto Los sube

Git fetch origin master

Git clone "url del repositorio" permite copiar todos los archivos en cualquier computadora de todos los archivos existentes del repositorio remoto

Git pull origin master: Los trae actualiza los archivos que clonaste los sincroniza

- Git init:
- git config user.name "Nombre de usuario"
- git config user.mail "mail"
- Crear el repositorio por github
- Agregar el README puedes explicarte
- Code me dice el url
- git remote add origin \*pego url\*
- touch archivos
- git add archivos
- git status
- Git commit -m ""
- Git log (historial e actividad)
- Git push origin master : solicita a git mandar los archivos del repositorio local al remoto

- `git status`

**Rama:** línea paralela dentro de un repositorio sin afectar al código (versión 2 para probar cosas nuevas y después si quieres unificas)

- `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.

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`).

`git push origin <branch>`

Así también, para traer los cambios de esa rama utilizamos el `git pull` agregando desde donde queremos traer los cambios:

- `git pull origin <branch>`