

## GIT y GITHUB

**Git:** Git es un sistema de control de versiones que rastrea los cambios en los archivos de un proyecto.

**GitHub:** GitHub es una plataforma en línea que utiliza Git para alojar y compartir proyectos de software, facilitando la colaboración entre desarrolladores.

### Instalación de GIT y Creación de cuenta en GITHUB.

Algunos videos.

<https://www.youtube.com/watch?v=IGfuztkRr4k>

<https://www.youtube.com/watch?v=WcYTcttEf50>

Página en la que se tiene que crearse una cuenta de Github <https://github.com/>

Página de descarga de Git <https://git-scm.com/downloads>

### Configuración Básica GIT

git config --global user.name "your name"	Configurar el usuario
git config --global user.email "your email"	Configurar el Email
git config --list	Listado de configuraciones
git config --global --list	Lista todas las configuraciones del global
git config --global core.editor "code --wait"	Se configura como editor el Visual Studio Code --wait se espera que los cambios se guarden hasta que se cierre el archivo
git config --global core.autocrlf=true	Configuración de salto de line y retorno de carro (Default true)
git config --local --list	Lista las configuraciones de nuestro repositorio local

ls -a (Para ver los archivos ocultos)

rm -r .git (Eliminar el repositorio local)

## Iniciando repositorio

git init	Iniciamos GIT en la carpeta donde está el proyecto
git clone <url>	Clonamos el repositorio de github

## Git add

git add . git add *.txt git add [archivo1] [archivo2] git add directorio/[archivo3]	Añadimos todos a la area de preparacion Añadimos solo los archivos .txt a la area de preparacion Añadimos [archivo1 ] y [archivo2] a la area de preparacion Añadimos [archivo3 ] ala area de preparacion
git mv antiguo.txt nuevo.txt	Cambiamos el nombre de archivo y luego se agrega a la area de preparacion

## Git commit

git commit -m "Esto es mi primer commit"	Agrega a repositorio
git commit -a	Agrega a repositorio sin necesidad de pasar por la área de preparación
git checkout [file]	Se recupera el archivo [file] del último commit
git restore [file]	Se recupera deshace los cambios del último archivo [file]
git log	Para ver la identificador de los commits o lista de los commit realizadas
git log --oneline git log --oneline --all git log --oneline --all --graph	Listado de los commit con una información resumida Lista todos los commit en todas las ramas Lista todos los commit en forma gráfica
git commit --amend	Modificar el último commit y agregando los archivos en la area de preparacion
git reset --soft [hash]	Deshacer un commit y los archivos de ese commit lo llegan al área de preparación.
git reset --soft head~1	Mueve el head a 1 commit anterior

git reset --hard [hash]	Se mueve el head al commit que tiene [hash]
-------------------------	---

git show nombre.txt muestra el contenido del archivo en el repositorio

## Git diff

Hash1

Hash2

git diff Hash2 Hash1

git diff --staged	Compara los archivos de la área de preparación con el último commit
git diff [hash1] [hash2]	Compara los dos commits
git diff --name-only [hash1] [hash2]	
git diff --word-diff [hash1] [hash2]	

## Git branch

git branch	Lista de ramas
git branch [nombre]	Se crea una nueva rama [nombre]
git checkout [nombre-rama] git switch [nombre-rama]	Se mueve a la rama [nombre-rama]
git checkout -b [nombre-rama] git switch -c [nombre-rama]	Crear y mover a la rama [nombre-rama]
git branch -d [nombre-rama]	Eliminar la rama [nombre-rama]
git branch -m [nombre-rama] [nuevo-nombre-rama] git branch -m [nuevo-nombre-rama]	Cambia el nombre de la rama Cambia el nombre de la rama actual

## Git merge

git merge [nombre-rama]	Se funciona la rama actual con la rama [nombre-rama]
-------------------------	--

## Conexión a github

### Crear un nuevo repositorio

```
echo "# prueba2" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin <url>  
git push -u origin main
```

### Conectar de un repositorio ya existente

```
git remote add origin <url>  
git branch -M main  
git push -u origin main
```

### Alias

git config --global alias.mi-comando "log --oneline --all --graph"

### git reflog

Lista todos los [hash] o sea historial de commit

Github es el repositorio remoto

### git clone

git push origin master

git push -u origin master (Para que siempre esta conectado en caso de que no este conectado )

git pull origin master

**git remote -v (Para ver la lista de los remotos )**

### Eliminar la configuración global de Git:

git config --global --unset-all user.name

git config --global --unset-all user.email

### Eliminar la configuración de credenciales:

git credential-manager uninstal