



Apuntes Módulo 2 - Git, GitHub, Hardware

Git y GitHub

Git:

Sistema de control de versiones.

Basics:

Default branch: `main`

Default remote name: `origin`

Referencia a branch actual: `HEAD`

Hijos: `HEAD: HEAD^` o `HEAD~1`

Nietos: `HEAD: HEAD^^` o `HEAD~3`

Head siempre apunta al último cambio.

`git init <directory>` ← Inicia un nuevo seguimiento.

(*si no es especificado donde, se crea donde estamos parados)

`git status` ← estado del repositorio

`git log` ← commit history

`git clone <url>` ← clona un repositorio remoto

`git add .` ← el flag `.` puede ser cambiado por el nombre del archivo

`git commit -m "<message>"` ← Commit y nota o mensaje sobre la actualización.

(*Debe ir entre comillas.)

Add + Commit:

`git commit -a -m "<message>"`

o

`git commit -am "<message>"`

`git reset <file>` <- Quitar de la staging area

`git mv <current path> <new path>` <- Mover o renombrar archivo

`git branch` <- muestra branches

`git checkout -b <branch>` <- crea y va hacia la nueva branch

`git checkout <branch>` <- va hacia una branch existente

`git branch -d <branch>` <- borrar una rama

`git branch -D <branch>` <- eliminado forzado

`git merge <rama>` <- fusiona la rama escrita en la actual

```

git rebase <branch> <- fusiona todos los commits en esta rama

git checkout <commit id> <- volver a un commit previo
git revert <commit id> <- revertir un commit
git reset <commit id> <- resetear un commit

git diff <- mostrar cambios a archivos unstaged
git diff - -staged <- mostrar cambios a archivos staged
git diff <commit id 01> <commit id 02> <- mostrar cambios entre dos commit

git stash <- permite guardar cambios temporalmente en stash sin hacer commit
git stash save "<message>" <- se puede agregar un mensaje al stash
git stash list <- lista los stashes

git remote add <remote name> <url> <- agregar un repositorio remoto

git remote add origin <url> ← agregar el origen

git remote -v ← chequear si ya esta coordinado el remoto con el local
git remote <- mostrar repositorio remoto
git remote -v <- esta flag muestra la url del repositorio remoto
git remote remove <remote name> <- borrar repositorio remoto

git fetch <remote name> <- actualiza cambios desde un repositorio remoto
git pull origin main <- actualiza los cambios en el repositorio local clonado con el de la nube
git push <remote name> <branch> <- push changes to a particular branch

git config user.name 'name' ← ingresar un nombre de usuario en el repositorio, sin las comillas
git config user.email 'email' ← ingresar un email relacionado al usuario en el repositorio, con comillas

*Consultando a estos pero sin ingresar datos nos devuelve el ya ingresado.

git config - - global user.name ← al igual que el anterior, pero define en TODA la pc tu usuario
git config - - global user.email ← al igual que el anterior, pero define en TODA la pc tu email

```

Subida

```

git add .
git commit -m "message"
git push origin main
git status

```

Clonado

```

git clone <url>
git pull <url>

```

Repositorio local ← directorio donde se guardan los archivos que estarán respaldados, con historial de modificaciones y cambios.

Commit ← paquetes de modificaciones que nos permiten hacer un seguimiento de los cambios