

## CHEATSHETT de COMANDOS GIT GIT HUB

Noelia Zurbriggen

**Git - -version:** me dice que versión tengo (en mi compu no funciona git -v)

**Git init:** posicionarse en la carpeta donde quiero hacer el repositorio y este comando genera el repo local.

**Git config user.name "NoeZur"**

**Git config user.email "noeliaz...@hotmail.com":** ambos es para decirle quien esta haciendo el repositorio para dp subirlo a GH tbn.

**Git config - - global user.name y global user.email** si trabajo siempre solo o en esta compu.

**Git add:** para subir un archivo al repo LOCAL. En realidad es para cambiar de untracked a tracked y listo para committear.

**Git add .:** para subir todos los archivos que hice al repo local.

**Git commit -m:** para hacer un commit y ponerle un titulo al mismo (es el cambio que lleva el commit. Ej: agrego archivo.js o error de inicio solucionado etc)

**Git log:** muestra los commits creados en esa carpeta con nombre, fecha y autor.

**Git status:** me muestra si hay archivos untracked o sin committear.

**Git rm** (es lo contrario de add) lo saco del seguimiento y no aparecera si hago un commit (no borra el archivo de la computadora)

**Git remote add origin:** para crear repositorio remoto en GH.

**Git remote add origin "https...."**(aqui va la direccion del repositorio remoto de git hub)" es para crear subir mis archivos al repositorio remoto desde la compu.

**Git remote -v:** para chequear que estoy en el repositorio correcto gitHub.

**Git push origin main:** para subir los archivos que ya les hice git add y git commit a la rama principal.

**Git push origin Zurbriggen\_Noelia:** para subir los archivos en la rama NZ del repo remoto

**Git clone https://... :** para clonar un repositorio remoto SOLO SE HACE UNA VEZ.

**Git pull origin main:** para actualizar la rama main de un repo ya clonado o con multiples usuarios.

**Github settings:** puedo elegir trabajar en otra rama.

**Git branch:** enumera la ramas

**Git branch >noelia>** Crea rama noelia

**Git branch -d<Noelia>:** elimina la rama Noelia siempre que todos los cambios esten fusionados con la MAIN (merge)

**Git branch -D<noelia>:** fuerza la eliminacion de la rama sin importar los cambios.

**Git checkout noelia:** para cambiar de rama (voy a noelia)

**Git push origin noelia:** para subir la nueva rama a GH

**Git merge:** para unir las ramas