

CHEATSHEET Sofi Bustabad >>>> TERMINAL CLI, GIT & GITHUB

COMANDOS + usados en TERMINAL

- MAN + nombre de comando. Me indica lo que significa el comando
- Pwd muestra el directorio donde estoy parada.
- MKDIR + nombre crea carpeta
- CD + nombre carpeta me posiciona en directorio/carpeta
- CD .. Entra al directorio padre del directorio en el que estoy
- TOUCH Crea archivos con cualquier extension
- MV para mover ficheros

Comandos + básicos

Git --version	INDICA VERSION DE GIT INSTALADA
Git init	GENERA REPOSITORIO LOCAL, en la carpeta en la que tendré archivos. Este repo inicialmente esta vacío.
git config user.name "nombre de usuario"	REGISTRA por única vez NOMBRE DE QUIEN HACE LOS CAMBIOS
git config user.email "micorreo"	REGISTRA por única vez EMAIL DE QUIEN HACE LOS CAMBIOS
git config --global user.name "usuario"	EVITA tener que CONFIGURAR CADA VEZ EL NOMBRE DE QUIEN REALIZA CAMBIOS
git config --global user.email "email"	EVITA tener que CONFIGURAR CADA VEZ EL EMAIL DE QUIEN REALIZA CAMBIOS
Ls -a	MUESTRA LOS ARCHIVOS OCULTOS (ej el repo local)

AGREGAR ARCHIVOS AL REPOSITORIO LOCAL (ej: en escritorio personal) & VIENDO STATUS

Git add	INDICO EL O LOS ARCHIVOS A AGREGAR
Git add .	AGREGA todos LOS ARCHIVOS AL REPO
Git status	ESTADO DE MIS ARCHIVOS

CONFIRMANDO ARCHIVOS

Git commit -m "agrego comentario explicativo"	AGREGA COMENTARIO EXPLICATIVO DE LOS CAMBIOS QUE VOY REALIZANDO ej: "agregue archivo .js" "modifique tal cosa"
Git log	REGISTRA HISTORIAL DE CAMBIOS

SUBIENDO ARCHIVOS a GIT HUB (Repositorio REMOTO, o en la nube)

*Para subir archivos tengo que tenerlos commiteados !!

Git push	SOLICITA A GIT QUE ENVIE ARCHIVOS AL REPO REMOTO
Git push origin master o main	ENVIA ARCHIVOS DEL REPO local AL REPO remoto

BAJANDO ARCHIVOS desde GIT HUB

Git clone <i>url del repo</i>	PERMITE crear copia exacta en la compu, de todos los archivos en el repo remoto. Se hace una sola vez para obtener los archivos del repo a una compu que por ejemplo es nueva y no tengo el trabajo alli alojado.
Git pull	
Git pull origin master o main	Mantener sincronizados (actualizados) los archivos en la compu.

RAMAS o BRANCH

- El comando para crear branch es **git branch nombre_rama**
- Para posicionarse en esa rama el comando es **git checkout nombre_rama**
- Para subir cambios a la rama creada es **git push origin nombre_rama**