git

# git init .- inicia un repositorio y crea una carpeta .git (oculta)

# git add . .- Agrega todos los archivos que se cambiaron en la carpeta (aquí pasa a staging)

# git commit -m “mensaje” .- Lo envía al repositorio .git (siempre de entrada es master/main)

# git rm --cached archivo(s) .- Los quita del repositorio y los reenvía a staging

# gi checkout .- Regresa a un commit y puedes revisar los cambios

# git log .-

# git reset --hard .- Elimina todos los commits y apunta el HEAD y main hacia el commit que le hayamos dicho

# git reset --soft .- Sólo apunta al commit que le hayamos dicho pero no borra los otros commits, conserva lo de staging.

# git commit -am “mensaje” .- Agrega a staging (add) y hace un commit con mensaje

# git log –stat .- Visualiza todos los comitts y sus cambios

# git diff commit commit .- compara los commits y muestra los cambios que se hicieron en cada uno

# git clone .- Descarga todo el contenido del repositorio remoto al repositorio local (laptop, pc)

# git push .- Sube los commits al repositorio remoto y lo actualiza

# git pull .- Trae los cambios del remoto y actualiza mi local

# git show .- Muestra las diferencias entre el commit actual con el commit anterior

# git branch *nombre* .- Te permite crear una nueva rama

# git branch .- Te muestras las ramas y te señala en cual estas

# git checkout nombre .- Te permite cambiar de rama (ejemplo)

- # git checkout firstbranch .- Te cambia a la rama firstbranch

- # git checkout master .- Te cambia a la rama master

El merge siempre se debe de realizar estando situado en la rama master/main, para que traigamos los cambios de las demás ramas (branches) de lo contrario, si nos situamos en cualquier rama (branch) y hacemos el merge, traeremos los cambios de master/main a esa rama.

staging .- área aislada del proyecto en RAM y ahí se guardan los cambios