

CHEATSHEET GIT-GITHUB

AYUDA

git help	Imprime la sinopsis y una lista de los comandos más usados
git help <cualquier_comando_git>	Imprime la sinopsis, una descripción y opciones a utilizar del comando de git especificado

MOSTRAR (REPOSITORIO LOCAL)

git status	Muestra cuáles archivos están en seguimiento y cuáles no
git status -s	Muestra los archivos en seguimiento y los que no, pero de forma recortada
git log	Muestra la información de los logs de los commits
git log --oneline	Muestra la información de los logs de los commits, pero solo una línea de información más entendible
git log --oneline grep <comentario>	Busca la información de un commit con un comentario en específico
git log -- <ruta de archivo>	Muestra el historial de un archivo específico
git log -- author=<usuario>	Muestra el historial de un usuario en particular

CONFIGURAR

git config --list	Muestra la configuración de Git
git config --global user.name "nombre de usuario"	Establece el usuario de forma global. Si --global no está presente, asigna el usuario a la carpeta correspondiente y las subcarpetas a la misma
git config --global user.email email@email.com	Establece el email de forma global. Si --global no está presente, asigna el email a la carpeta correspondiente y las subcarpetas a la misma
git config --global --unset user.name "nombre de usuario"	Elimina el registro que se refiere al usuario
git config --global --unset user.email email@email.com	Elimina el registro que se refiere al email

CREAR

git init	Crea un nuevo repositorio
git clone git@github.com:minombre/archivo-git.git	Clona un repositorio remoto de forma local
git remote add origin git@github.com:minombre/archivo-git.git	Enlaza el repositorio local con el repositorio remoto

AGREGAR/REGISTRAR	
git add <nombre de archivo>	Agrega un archivo que no tenga seguimiento o se haya modificado y pasa a ser seguido (tracked)
git add . / git add --all	Agrega todos los archivos que no tengan seguimiento o se hayan modificado, pasando a ser seguidos (tracked)
git commit <nombre de archivo> -m "mensaje del commit"	Registra los cambios de un archivo o directorio que se encuentra en seguimiento (staged)

DESHACER/REMOVER/RESTAURAR/CAMBIAR	
git checkout --archivo	Deshace los cambios locales realizados a un archivo en su directorio de trabajo local (debe usarse mientras el archivo no se haya añadido al área de trabajo temporal)
git reset HEAD <nombre de archivo>	Deshace los cambios locales realizados a un archivo en el área de trabajo temporal (stage area). Debe usarse cuando el archivo se haya añadido al área de trabajo temporal
git rm <nombre de archivo>	Remueve un archivo
git rm -r <nombre de directorio>	Remueve un directorio
git rm --cached <nombre de archivo> / git rm -cached .	Remueve un archivo o todos los archivos del área de trabajo temporal
git restore .	Restaura todos los archivos en disco pero no en Git
git restore --staged <nombre de archivo> / git restore --staged .	Saca del "staged area" un archivo o todos
git commit --amend -m "Mi nuevo mensaje"	Cambia los mensajes del commit

MOSTRAR (REPOSITORIO REMOTO)	
git remote	Muestra una lista de los repositorios remotos existentes
git remote -v	Muestra una lista de los repositorios remotos existentes con su URL después del nombre
git remote show origin	Muestra la información de los repositorios remotos

RENOMBRAR/REMOVER (REPOSITORIO REMOTO)	
git remote origin <nombre_nuevo>	Renombra un repositorio remoto
git remote rm <nombre_git>	Desvincula un repositorio remoto

ENVÍO/DESCARGA	
git push -u origin main	Envía y actualiza los cambios hechos localmente en el repositorio remoto para cada rama que esté actualizada o le fue hecho exitosamente un push
git push	Envía y actualiza los cambios hechos localmente en el repositorio remoto
git pull	Descarga y actualiza los cambios hechos en el repositorio remoto de forma local

RAMAS	
git branch	Lista las ramas
git branch -v	Lista ramas con información de los últimos commits
git branch --merged	Lista las ramas que ya se han unido (merged) con main
git branch --no-merged	Lista las ramas que no se han unido (merged) con main
git branch <nuevaBranch_nombre>	Crea una nueva rama
git checkout <nuevaBranch_nombre>	Cambia a una rama existente (el puntero HEAD apunta a la rama <nuevaBranch_nombre>)
git checkout -b <nuevaBranch_nombre>	Crea una nueva rama y apunta a ella
git checkout main	Vuelve a la rama principal-main
git merge <nuevaBranch_nombre>	Resuelve la unión (merge) entre ramas. Para realizar la unión debe estar en la rama que debe recibir los cambios.
git merge --abort / git reset --merge	Deshace la unión cuando se tiene problemas
git branch -d <nuevaBranch_nombre>	Apaga una rama
git pull origin <nombreBranch>	Saca los archivos de una rama existente
git push origin <nuevaBranch_nombre>	Crea una rama remota con el mismo nombre
git reset HEAD	Volver a un commit anterior. Si se quiere volver a más de un commit, poner el número de commits después de HEAD (Ejemplo: HEAD~2)