

GIT COMMANDS

BASICOS

Git init = inicia un repositorio (es la base de datos donde se guardan esos cambios que hacemos)

Git add. = El repositorio sabe que existe el archivo y esto permite en cierta forma arrancarlo

Git commit -m “(nombre del commit)”= Es el que nos permite enviar los últimos cambios del archivo al repositorio

Git add .= Nos permite agregar todos los archivos en la carpeta que hayan cambiado

Git status = Permite ver el estado de nuestro repositorio si hemos hecho algún cambio o si no lo has añadido aun (add y commit)

Git log nombre_archivo.ext = Me muestra toda la historia entera de ese archivo

touch nombre_archivo.ext = Crea un archivo en esa carpeta

ls = muestra los archivos que hay en esa carpeta

ls -al = ver archivos ocultos

history = muestra todos los comandos que has hecho en la terminal

rm nombreakivo.ext = Borrar un archivo

pwd=Muestra en que parte de los directorios nos encontramos

INTERMEDIOS

Git diff = Compara dos o más commits

Git log --stat= Veo los cambios específicos de cada archivo

Git config -l= Nos permite ver la configuración por defecto de nuestro repositorio

Git config --global = Permite ver comandos más específicos

Git branch= Muestra las ramas que tengo

Git branch nombre_de_la_rama= Permite crear una rama

Git checkout nombre_de_la_raiz= Me permite trasladarme a otras ramas

Git merge nombre_de_la_rama= Me permite fusionar los cambios de esta rama con la rama master (rama principal) (Debo estar en la rama que vaya a hacer el merge por ejemplo en la master)

Git remote add “nombre” URL de https o ssh(llave) = Sirve para establecer esa conexión entre nuestro repositorio local y nuestro git hubb para hacer pull y push

Git pull (nombre de nuestro remote) master(nombre de la rama)

Git pull (nombre de nuestro remote) master(nombre de la rama) --allow --unrelated -histories

Git push (nombre de nuestro remote) master(nombre de la rama)

AVANZADO

Creación de llaves

ssh-keygen -t rsa -b 4096 -c ""

eval \$(ssh-agent -s) = Esto nos permite saber si el servidor ssh si esta corriendo

ssh -add (directorio en donde están las llaves (debe ser la llave privada)) =Esto nos permite agregarla a nuestro sistema para que este sepa que existe esta llave

Conectar github con ssh

Git remote -v = Me permite visualizar todos mis remote creados

Git remote set -url (nombre del remote que vas a cambiar) url de ssh

Git log --all --graph --decorate --oneline = Nos permit ver los commits de una forma mas grafica y detallada

Alias sobrenombre = "línea de comando"

Git clone url del repositorio de github= Me permite clonar un repositorio que no es mío (si es publico obviamente)

Git stash= Nos sirve para guardar cambios para después, estos cambios temporales los puedo ver con **git stash list**

Para abrir los cambios vamos al branch donde lo hicimos y escribimos **git stash pop**, luego hacemos el commit

Y si quieres pasar esos cambios a otro branch haces esto **git stash branch** nombre de la raíz

Git stash drop = Permite borrar un stash

Git clean = Limpia tu Proyecto de archivos no deseados

Git clean --dry -run = Es una simulación de lo que voy a borrar

Git clean -f =borra todo lo que habíamos dicho

Git cherry -pick (numero del commit) = Trae commits viejos al head de un branch

Git commits amend = Los cambios que hice los va a pegar al commit anterior

Para esto primero agregamos con add y luego si con commit --amend

Git reflog = Git nunca olvida

rest --soft= Hacer un borrado suave

rest --Hard =Hacer un borrado duro

Git grep palabra = Sirve para mostrarme donde use esa palabra

Git grep -n palabra = Me muestra en que línea de código use esa palabra

Git log -s “ ”= Muestra cuantas veces use esa palabra en un commit

Git short log -sn = Me muestra cuantos commits han hecho cada miembro del equipo

Añadiéndole el --all me muestra incluso los commits borrados

Git blame -c nombre del archivo = Me muestra quien hizo cada cosa línea por línea