LOPEZ ROBERTO JOSE

COMANDOS DE GIT:

Usar la flecha hacía arriba nos muestra el último comando utilizado

1. Configuración Inicial

git config --global user.name "Tu Nombre" #Configura tu nombre de usuario para todos tus repositorios.

git config --global user.email tuemail@example.com #Configura tu correo electrónico.

git config --global core.editor "tu_editor" #Configura tu editor de texto preferido.

Git config --list

2. Creación y Clonación de Repositorios

git init #Inicializa un nuevo repositorio Git en el directorio actual.

git clone URL #Clona un repositorio remoto a tu máquina local.

Git checkout hash #direccion o hash de commit que quiero cargar

3. Trabajando Con Carpetas

mkdir <<nombre_carpeta>> #Crea una carpeta, recordar que en window las mayúsculas no tienen relevancia, pero si en Linux

rmdir <<nombre_carpeta>> #borra la carpeta, cuando esta vacia

rm -rf <<nombre_carpeta>> #borra la carpeta, con los archivos dentrogit

git rm --cached -r <<nombre_carpeta>> #borra una carpeta del area preparación de con sus archivos dentro git

git reset

4. Trabajando con Archivos

git status	#Muestra el estado actual del repositorio (archivos modificados, agregados, etc.).
touch “nombre_archivo”	para crear archivos específicos
echo > nombre archivo	para crear archivos específicos
git add <<archivo>>	#Agrega un archivo específico al área de preparación (staging area).
git add .	#Agrega todos los archivos nuevos y modificados al área de preparación.
git commit -m "Mensaje del commit"	#Realiza un commit de los archivos en el área de preparación con un mensaje descriptivo.

Git commit –am “escribimos el comentario de nuestro commit”

#No se puede utilizar con archivos nuevos.

#Sólo con archivos modificados

git rm <<archivo>>	# Elimina un archivo del repositorio y lo registra para el próximo commit.
rm <<archivo>>	#elimina archivo
git rm --cached <<nombre_archivo>>	#(para eliminar un archivos del area de preparacion)
git restore --stage «nombre del archivo»	#vuelve el archivo al area de trabajo y poder modificarlos

5. Inspección y Comparación

git log	#Muestra el historial de commits del repositorio.
git log –oneline	#Muestra el historial de commits en una sola línea por commit.
git diff	#Muestra las diferencias entre los archivos en el directorio de trabajo y el área de preparación.
git show	#Muestra los cambios de un commit específico.
Git checkout –f	#nos va a permitir borrar todos los últimos cambios realizados

6. Ramas (Branches)

git Branch	#Lista todas las ramas en tu repositorio.
git branch <<nombre_rama>>	#Crea una nueva rama.
git checkout <<nombre_rama>>	#Cambia a la rama especificada.
git checkout -b <<nombre_rama>>	#Crea y cambia a una nueva rama.
git merge <<nombre_rama>>	#Fusiona la rama especificada con la rama actual.(ósea la rama en la que estoy posicionado con la rama q menciono en el comando)
git branch -d <<nombre_rama>>	#Elimina una rama
git merge <<mi-rama-secundaria>>	# Fusiona la rama secundaria en la rama principal: Utiliza el comando git merge seguido del nombre de la rama secundaria que deseas fusionar en la rama principal. Por ejemplo, si tu rama secundaria se llama mi-rama-secundaria, ejecutarías:(estando en la rama master)

7. Remotos (Remotes)

git remote add <<carpeta_local>> <<URL>>	#Agrega un repositorio remoto.
git remote -v	#Muestra los repositorios remotos configurados.
git fetch	#Descarga cambios desde el repositorio remoto sin fusionarlos.
git pull	#Descarga y fusiona cambios desde el repositorio remoto.
git push origin <<nombre_rama>>	#Envía tus commits a la rama especificada en el repositorio remoto.

git remote remove <<nombre_carpeta>>

Para comprender esto voy a llamar (tecnicaturagit) a la rama local y (tecnicaturagithub) a la rama online

La rama online tiene una ruta de acceso que lo saco de git hub
https://www.github.com/ejemplo/curso_git_github

git remote add carpeta + url	(para conectar una repositorio online con uno local)
git remote add Tecnicaturagit https://www.github.com/ejemplo/curso-git-github curso-git seria mi carpeta en la pc y el url es la dirección de mi repositorio online	
Git	Muestra cual es la carpeta que esta vinculada, en este ejemplo curso_git
git remote –v **Fetch** para descargar **Push** para subir	Para ver cual es la ruta del repositorio online en este ejemplo https://www.github.com/ejemplo/curso_git_github>>
git remote –verbose	Es igual al de arriba pero la forma completa

git fetch nombre_carpeta ejemplo	Para descargar lo q esta online a travez de esa dirrecicon q pusimos

git fetch tecnicaturagit	
git pull nombre_carpeta+nombre_rama_online ejemplo **git pull** tecnicaturagit master	Ahí lo q hacemos es un fetch_head , seria ponemos lo q tenemos en la carpeta por ejemplo rama master online en la rama master local descarga
git merge <<nombre donde se encuentra la conexión/ donde quiero q escriba lo que esta almacenado en esa conexion>> **git merge** curso-git/master	Otra forma de hacer las convinaciones de datos
git push - -set-upstream <<carpeta conectada + que es lo q deseo sincronizar o lugar donde estan las cosas q quiero sincronizar>> **git push --set-upstream** curso-git master	En este caso lo que realizo es cuando hago una modificacion en el repositorio local, sincronizarlo o subirlo al main que seria el online median git pushh

8. Revertir Cambios

git reset --hard HEAD	#Revierte todos los cambios en el directorio de trabajo al último commit.
git reset --soft HEAD~1	#Deshace el último commit, pero mantiene los cambios en el área de preparación.
git revert commit	#Crea un nuevo commit que revierte los cambios de un commit específico.

9. Stashing

git stash	#Guarda temporalmente los cambios no confirmados.
git stash list	#Lista los stashes guardados.

git stash apply	#Aplica el último stash guardado.
git stash pop	#Aplica el último stash guardado y lo elimina de la lista de stashes.

10. Etiquetas (Tags)

git tag	#Lista las etiquetas (tags) existentes.
git tag >>nombre_etiqueta<<	#Crea una nueva etiqueta.
git tag -a <<nombre_etiqueta>> -m "Mensaje de la etiqueta"	#Crea una nueva etiqueta anotada.
git push <<origin>> <<nombre_etiqueta>>	#Envía una etiqueta específica al repositorio remoto.

11. Búsqueda y Navegación

git grep "texto"	#Busca un texto específico en todo el repositorio.
git reflog	#Muestra el historial de referencias del repositorio (incluye commits eliminados).

---

pwd	#Vemos la ruta de la carpeta en la que estamos
cd	#Es para navegar a una carpeta: change directory -> cambiar de directorio
cd /	#Nos llava al home, en la raíz del disco
cd ~	#La virgulilla significa que estamos en el lugar de los documentos o del usuario

ls	#Esto es listar los archivos, nos muestra todos los archivos en la raíz
ls -al	#El espacio -al significa que es un argumento especial para ver archivos ocultos
ls -l	#Muestra casi todos los archivos sin los que están ocultos
ls -a lista	#Muestra el grupo de archivos pero no en una
clear	#Limpia la consola
ctrl + l	#Limpia la consola
cd ..	#Nos devuelve a la carpeta anterior
cd U + tab	#Esto se usa para un autocompletado o para buscar una referencia
df -h	#Muestra información de los discos, memorias en uso y memoria disponible