COMANDOS TERMINAL GIT BASH

Mostrar	
ls	lista de archivos en el directorio o carpeta
ls -a	lista todos los archivos, incluyendo los archivos ocultos
ls -l	Muestra toda la informacion de una carpeta: usuario, grupo, permisos, tamaño, fecha y hora de creacion
ls -R	Muestra las carpetas y archivos contenidos en ellos de manera recursiva
pwd	Muestra la carpeta en la que se esta trabajando actualmente
more {nombre del archivo}	Muestra el contenido de un archivo

Copiar/Mover/Renombrar	
mv {ruta/archivo1} {ruta/archivo2}	Renombra archivos (archivo2 no debe existir o sera sobrescrito)
mv {ruta/carpeta1} {ruta/carpeta2}	Renombra la carpeta1 como carpeta2 (carpeta2 no debe existir)
mv {ruta/carpeta1} {ruta/carpeta2}	Mueve contenido de carpeta1 a carpeta2 (carpeta2 debe existir)
cp {ruta/archivo1} {ruta/archivo2} **cp {ruta/carpeta1} {ruta/carpeta2}**	Copia un archivo o carpeta
opcion: -r	Indica que compie recursivamente el contenido de las subcarpetas

Crear	
mkdir {carpeta}	crea un nuevo directorio o carpeta
touch {nombre del archivo/s}	Crea un nuevo archivo/s
Navegacion entre carpetas	
cd ..	sube un nivel de carpeta
cd	Cambia de carpeta
cd/xxx/xx	Cambia a una carpeta especifica
Caracteres especiales	
""(comillas)	Nos permite utilizar terminos que consistan en mas de una palabra
. (el punto)	Permite hacer referencia al directorio donde estamos ubicados actualmente

Eliminar	
rm {nombre del archivo}	Elimina un archivo
rmdir {nombre de la carpeta	Elimina una carpeta vacia
rm -r {nombre de la carpeta}	elimina una carpeta y su contenido
Atajos de teclado	
crtl + c	Finaliza un proceso vigente que esta corriendo en la terminal
crtl + l	Limpia la pantalla de la terminal
Otros comandos	
clear	Limpia la pantalla de la terminal
comando --help	Muestra ayuda del comando

GIT HUB COMANDOS

Configuracion Basica	
git config --global user.name "nombre"	Configurar nombre que salen en los commits
git config --global user.email xxxxxx@gmail.com	Configurar el email
git config --global color.ui true	Marco de colores para los comandos
Iniciando Repositorio	
git init	Iniciamos git en la carpeta donde esta el proyecto
git clone (url)	Clonamos el repositorio de github o bitbucket
git add .	Añadimos todos los archivos para el commit
git commit -m "texto de que se hizo en el commit"	Hacemos el commit de los archivos agregados anteriormente
git push origin (nombre de la rama)	Subimos el repositorio
Git add	
git add .	Añadimos todos los archivos para el commit
git add (archivo)	Añadimos el archivo para el commit
git add --all	Añadimos todos los archivos para el commit omitiendo los nuevos
git add *.txt	Añadimos todos los archivos con la extencion especificada
git add docs/*.txt	Añadimos todos los archivos dentro de un directorio y de una extencion especifica
git add docs/	Añadimos todos los archivos dentro de un directorio
Git commit	
git commit -m "texto de que se hizo en el commit"	Carga en el HEAD los cambios realizados
git commit -a -m "texto de que se hizo en el commit"	Agregar y cargar en el HEAD los cambios realizados
git commit -a	Si hay conflictos los muestra
git commit --amend -m "texto de que se hizo en el commit"	Agrega al ultimo commit, este no se muestra como un nuevo commit en los logs.Se puede especificar un nuevo mensaje
Git push	
git push (origen) (branch)	Subimos el repositorio
git push --tags	Subimos un tag
Git log	
git log	Muestra los logs de los commits
git log --oneline --stat	Muestra los cambios en los cmmits
git log --oneline --graph	Muestra graficos de los commits
Git diff	
git diff **git diff --staged**	Muestra los cambios realizados a un archivo
Git head	
git reset HEAD (archivo)	Saca un archivo del commit

git reset --soft HEAD^	Devuelve el ultimo commit que se hizo y pone los cambios en staging
git reset --hard HEAD^	Devuelve el ultimo commit y todos los cambios
git reset --hard HEAD^^	Devuelve los 2 ultimos commit y todos los cambios
git log git reset --hard (commit_sha)	Rollback merge/commit
Git remote	
git remote add origin (url)	Agregar repositorio remoto
git remote set-url origin (url)	Cambia de remote
git remote rm (name/origin)	Remover repositorio
git remote -v	Muestra lista de repositorios
git remote show origin	Muestra los branches remotos
git remote prune origin	Limpiar todos los branches eliminados
Git branch	
git branch (nameBranch)	Crea una rama
git branch	Lista de las ramas
git branch -d <nameBranch>	Comando -d elimina la rama y lo mezcla al master
git branch -D <nameBranch>	Elimina sin preguntar
Git tag	
git tag	Muestra una lita de todos los tags
git tag -a <verison> - m "esta es la versión x"	Crea un nuevo tags
Otros comandos	
git status	Lista un estado actual del repositorio con lista de archivos modificados o agregados
git checkout --(file)	Quita del HEAD un archivo y le pone el estado de no trabajado
git checkout -b newlocalbranchname origin/branch-name	Crea un branch en base a uno online
git pull origin <nameBranch>	Busca los cambios nuevos y actualiza el repositorio
git checkout <nameBranch/tagname>	Cambia de branch
git merge (nombreBranch)	Une el branch actual con el especificado
git fetch	Verifica cambios en el repositorio online con el local
git rm (archivo)	Borra un archivo del repositorio
Fork	
git remote add upstream (url)	Descargar remote de un fork
git fetch upstream **git merge upstream/master**	Merge con master de un fork
Git rebase	
git rebase	Los rebase se usan cuando trabajamos con branches esto hace que los branches se pongan al día con el master sin afectar al mismo.Une el branch actual con el mastar, esto no se puede ver como un merge.
git rebase --continue	Cuando se produce un conflicto no das las siguientes opciones: cuando resolvemos los conflictos --continue continua la secuencia del rebase donde se pauso
git rebase --skip	Omite el conflicto y sigue su camino
git rebase --abort	Devuelve todo al principio del rebase
git rebase (nombre de la rama)	Para hacer un rebase a una rama en especifico