# Guía Básica de Git

git init >> crea repositorio .git

seguimiento: es un estado en el que git ya considera los archivos para añadirlos al repositorio

<variables> : todo lo que esta adentro <> son variables

'file name.exmp' : los archivos y carpetas que contengan espacio en su nombre deben ser agregados colocando su nombre y formato dentro de comillas simples 'ejemplo.ejm'

## /\*\*Actualizar\*\*\*\*\*\*/

git add \* >> Agrega todos los archivos a seguimiento

git add . >> Agrega todos los archivos a seguimiento

git add --a >> Agrega todos los archivos a seguimiento

git add <file.name> >> Agrega el archivo seleccionado a seguimiento

git rm --cached <file.name> >> Quita el archivo seleccionado de seguimiento

git commit -m "mensaje" >> Actualiza el repositorio local, añadiendo los cambios de seguimiento incluyendo un mensaje

git commit -m "mensaje Titulo" -m "mensaje descripción" >> Agrega una actualización al repositorio local con título y descripción

## /\*\*Repositorio Remoto (online)\*\*\*\*\*/

git remote add origin <repositorio.dominio/ruta> >> agrega la ruta como repositorio remoto

git push -u <origin> <master> >> actualiza la rama master del repositorio remoto con lo que contiene origin del repositorio local

git clone <repositorio.dominio.git> >> Clona el repositorio en la carpeta de trabajo donde se estan ejecutando los comandos

git fetch <repositorio.dominio.git> >> Trae los archivos del repositorioa tu carpeta de trabajo y actualiza con tu repositorio local

git merge <repositorio.dominio.git> >> Actualiza el repositorio local con las actualizaciones del servidor remoto

## /\*\*Revisión\*\*\*\*\*\*\*\*/

git status >> muestra los archivos en seguimiento indicando en rojo los que no estan en seguimiento y fueron editados y en verde los editados y bajo seguimiento

git log >> Muestra historial de commits

## /\*\*Editar ir Atrás\*\*\*\*\*/

git commit --amend >> Edita el ultimo commit (es para enmendar algo)

{

este comando te lleva a otra pantalla donde Decides si Editar el mensaje o lo dejas así,

+Si No editas el Mensaje del Commit{

Aprieta **Ctrl + C** para culminar el enmendado y luego escribe **:qa!** y apretar **[ENTER]** para terminar

}

+Si editaste el mensaje{

Luego de Editar el mensaje

Aprieta **Ctrl + C** para culminar el enmendado, luego escribe **:wa!** y aprieta **[ENTER]** para actualizar el mensaje,

luego escribe **:qa!** y apretar **[ENTER]** para terminar}

}

git rm --force <file.name> >> Elimina el archivo indicado del disco duro y el repositorio local

git restore --staged <file> >> Para quitar del seguimiento de un archivo eliminado

git restore <file> >> Deshace la eliminación de un archivo (lo regresa a la carpeta)

git reset --hard <commitCode> >> Para ir atrás hasta el codigo de ese commit eliminando todos los cambios hasta el momento de ese commit

git reset --soft

git reset --mixed

## /\*\*Ramas\*\*\*\*\*\*\*/

git branch -M <nombre> >> crea una rama con ese nombre

git checkout -b <branchName> >>Crea una nueva rama y te mueve a ella

git checkout <branchName> >> Te mueve hacia la rama indicada

git branch -d <branchName> >> Borra la rama indicada

git branch >> Para ver todas las ramas (la rama en verde indica la rama en la que estas situado)

## /\*\*Tags\*\*\*\*\*/

git tag <tagToAdd> <code> >> Agrega un Tag (etiqueta) al commit del codigo code