Crear un repositorio y normalmente un archivo.

Click derecho en el repositorio y se abre con git bash, aplicamos el comando git init

**Git init:** Inicia el proyecto

Hacer cambios:

**Git status:**

muestra los cambios **SOLO** los realizados dentro del proyecto lo que permite ver si se hicieron modificaciones desde el ultimo commit, si un archivo ya se hizo git add entonces no se muestra, excepto si se usa el siguiente comando.

Con git status -s Podemos visualizar **TODOS** los archivos del root y las carpetas, si la carpeta o archivo tiene un **verde** es que todo su contenido está salvado, y rojo es que aún no se salva.

**Git add <archivo>:**

Para salvar un archivo enviándolo al staging área.

con git add . envía todos los cambios.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Git commit -m “<mensaje>”:**

Realiza un commit, enviando todo nuestro staging area a local y guardando esta versión del proyecto

Si es la primera vez toca configurar el username y el email así:

git config global user.username “JUAN”

git config global user.email [correo@gmail.com](mailto:correo@gmail.com)

**Git log --oneline:**

Con este commando Podemos ver todos los commits realizados.

Texto

Descripción generada automáticamente

**Git reset --hard <el ID en amarillo del log>:**

Para restaurar el commit anterior.

**EJEMPLO:** Antes del comando

Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Después del comando:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Efectivamente lo ultimo a lo que se le hizo commit

**NOTA**: Se perderán los logs hasta donde se regrese

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ya no aparece el commit en el que estaba

En caso de querer cambiar la descripción del commit realizado se puede usar VIM en la consola git bash

**Git commit --amend**

:i modo edición

Borrar esa descripción antigua

Esc

Luego comenzar a escribir

Esc nuevamente

:wq comando para guardar y salir

Texto

Descripción generada automáticamente

En este punto, o desde el inicio, puedes generar tu proyecto en github y utilizar los comando que aparecen para proyectos existentes por línea de comandos

En caso de este ejemplo puntualmente fue:

git remote add origin <URLdada>

(No se usó) git branch -M master

git push -u origin master

**Git push origin master:**

Envía mis cambios de local a mi repositorio en remoto en github.

**Git pull:**

Envía los cambios de remoto a mi repositorio local.

**”:**

Crea un tag EN LOCAL

Tag: versión alguna del proyecto 100% utilizable dentro de lo que contiene.

**Git push --tags:**

Sube el tag al repositorio en remoto

**Git branch <nombre>:**

Crea una rama con el nombre a partir del punto en el que estás

**Git Branch:**

Muestra las ramas y señala la rama en la que estamos ubicados

**Git checkout <nombre rama>:**

Nos movemos a la rama a la que indicamos

**Git merge <rama a fusionar>:**

Parados en una de las 2 ramas, utilizamos este código para hacer merge con la rama descrita arriba, en caso de presentar conflictos se informará.

**Git branch -d <nombre rama a borrar>:**

Borra una rama, se recomienda hacer esto una vez que se hace merge exitosamente