**Manual Git**

* El primer comando que se usa es luego de estar posicionados en el directorio

Git init: que crea dos áreas “área de ensayo” y “repositorio local”

Git add: para hacer seguimiento del archivo, lleva del directorio al área de ensayo

Git commit –m “”: traslada el archivo del área de ensayo al repositorio local

Git status –s: nos dice los archivos y directorios que están en el directorio

NOTA: si no deja subir el archivo puede ser que le solicite el usuario / email y se hace con los siguientes comandos:

Git config –global user.name “Edward rosales”

Git config –global user.email : [ed@gmail.com](mailto:ed@gmail.com)

NOTA: Si queremos actualizar un archivo que ya está guardado en el repositorio, pero se le ha hecho una actualización primero se le lleva al área de ensayo con el git add y luego se le hace un commit y se le agrega la información realizada para que todo sea ordenado

* Podemos ver con el siguiente comando las copias o las fotos realizadas a nuestras versiones

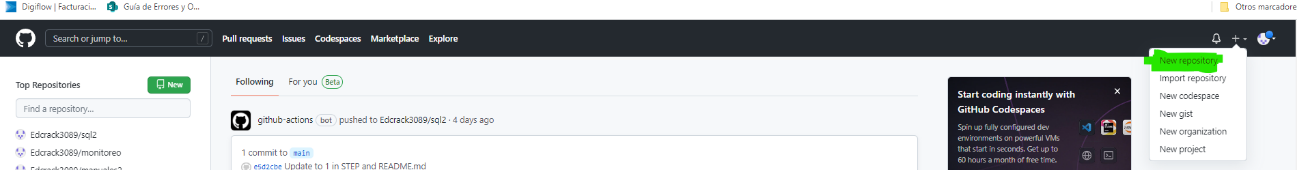
Git log --oneline: log del proyecto

Git reset --hard + el código del log al quie se quiere regresar: se hace para devolver a la versión anterior del proyecto que queremos usar.

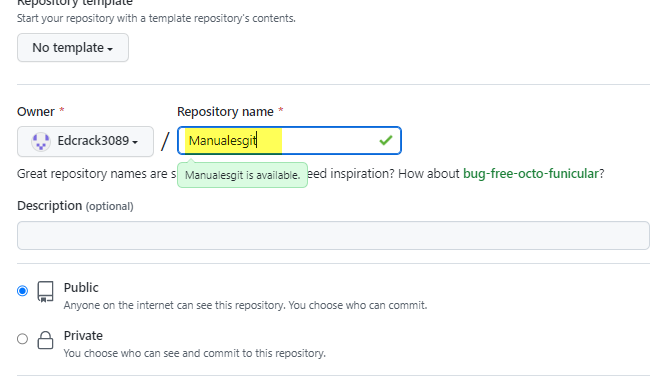
Git commit -- am: se puede hacer el add y el commit simultáneamente

**Manual para GitHub**

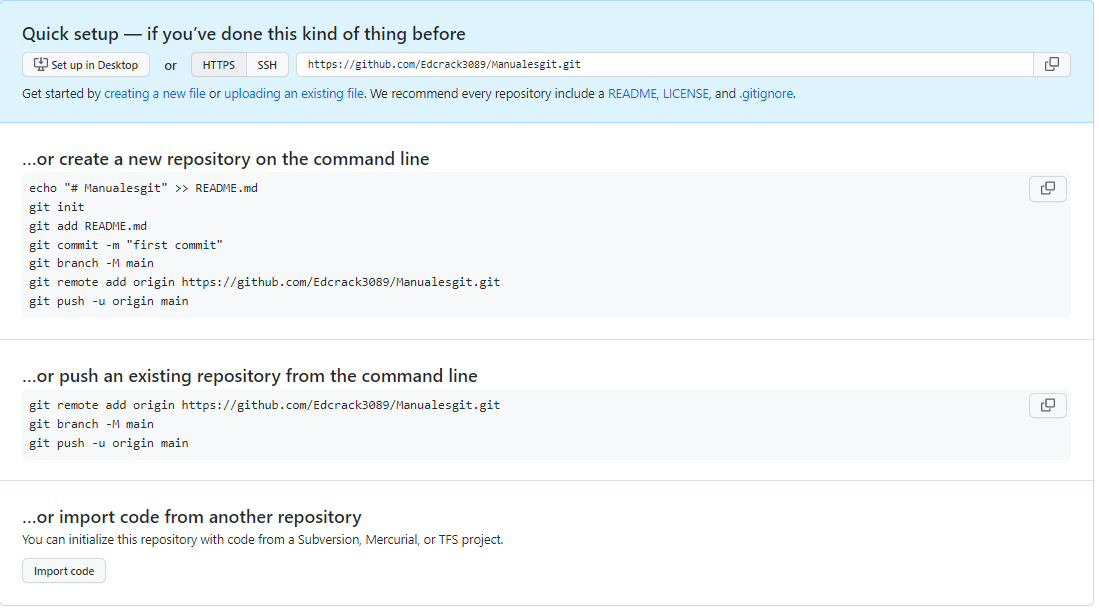
* El primer paso es crear un repositorio



* Luego se coloca el nombre al repo



* Se puede elegir si es público o privado
* Te crea el código que debe ingresar para enviar de git a github



Git remote add origin <https://github.com/Edcrack3089/Manualesgit.git>: este se usa para subir de local git a github

Git branch -M main: para cambiarte al main y subir las actualizaciones

Git push -u origin main: para ir cargando las modificaciones

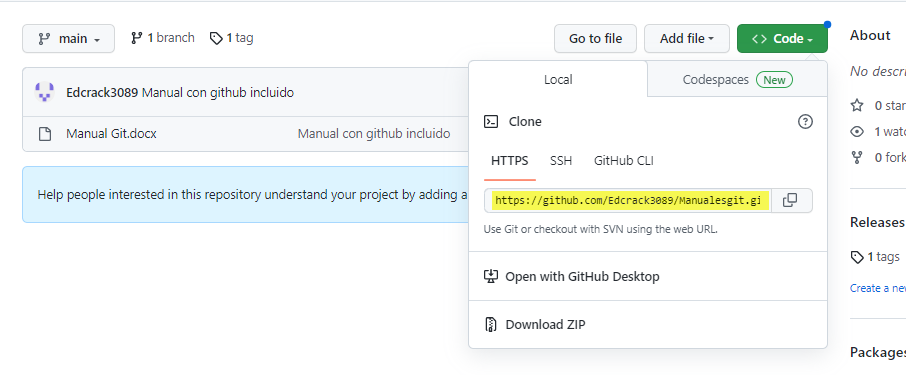
Git pull: detecta si se ha hecho algún cambion en git hub y te trae la información cambiada al repositorio local

Git tag +nombre de la etiqueta+ -m “descripcion”: se puede crear etiquetas que nos ayudan a descargar las versiones terminadas de nuestro proyecto

Git push --tags: sube la etiqueta que nos permite descargar los archivos

* Si llega a ocurrir algún desastre y se pierde la información de la maquina hay una opción para clonar lo que este guardo en github

Git clone +URL: se copia la dirección de la URL de git hub



**Ramas o branch**

* Estas se usan si se quieren trabajar en un proyecto de forma paralela lo cual uno se puede mover en diferentes ramas sin afectar la principal, primero se debe crear una rama adicional esto se hace con los siguientes comandos:

Git branch + nombre que se le quiere dar a la rama: este crea la rama

Git checkout + nombre de la rama creada: te mueves en a esa rama y comienzas a trabajar en una rama paralela que no afecta a la ppal.

* Indicar que su haces algúna actualización estando en esta rama si cambias a la ppal los cambios no se han guardado en la ppal si no en la rama creada con el cual con un nuevo comando se puede unir a la rama ppal con el siguiente comando:

Git merge + la rama que quieres que se fusione: hace que los cambios que estabas trabajando se fusionen.