**Cuenta github**

JohanC97

1997Johancampos

**Cuenta gitLab**

[jecampos@uned.cr](mailto:jecampos@uned.cr)

iAfjmDc5B7s:!zm

**Crear repositorio remoto**

* Se crea en la página de git, en el usuario.

**Enlazar con repositorio remoto**

Al crear un repositorio se generan instrucciones, copiamos la sexta línea algo así:

(git remote add origin https://github.com/JohanC97/prueba-2.git)

* Git init (para inicializar un retositorio local en la carpeta especificada)
* Git remote add origin <https://github.com/JohanC97/prueba-2.git> (para enlazar al repositorio remoto)

También podemos clonar un repositorio remoto a uno local existente

* Git clone <url del reposotorio>

**Cargar cambios al repositorio**

* Git status (opcional, (para ver si hay cambios))
* Git add . (con punto para agregar todo o con el nombre del archivo que quiero agregar)

Para remover los archivos o datos agregados

* Git rm – -cached <archivo>
* Git commit -m “mensaje”

Para remover un comit o cambiar el mensaje de ese comit

* Git reset –soft HEAD~1
* Git push
* Or Git push origin master (master es la rama)

Si al cargar cambios a repositorio remoto y nos pide credenciales

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Podemos generar un token en la cuenta de git para poder subir los cambios (cuenta, perfil, settimgs, developer settings)

**-----------------------------Bajar novedades del repositorio--------------------------------**

* Git pull

Para clonar un repositorio

* Git clone (link ssd or normal link)

**------------------------------------------------Extras----------------------------------------------**

Si al hacer alguna solicitud en git desde la consola, nos pide credenciales

* git config --global credential.helper manager
* luego seguimos con el push normal

Si al hacer push sale el siguiente error

fatal: The current branch main has no upstream branch.

To push the current branch and set the remote as upstream, use

* git push --set-upstream origin main

Se puede crear o escribir en un archivo txt con el siguiente comando

* echo “texto a escibir” >> nombre de archivo .txt

Se puede ver un archivo txt con el siguiente comando

* Cat nombre de archivo .txt

Para ver la información de cambios

* Git log

Para establecer un nombre para identificar quien hace los cambios

* Git config –global user.name “nombre”

Para establecer un correo de usuario

Git config –global user.email “correo”

**--------------------------------------------Ramas--------------------------------------------------**

Para ver las ramas existentes

* Git Branch

Para crear una nueva rama

* Git Branch <nombre de la rama>

Para cambiar de rama

* Git checkout <nombre de la rama>

Para crear y pasarse de una ves a la rama

* Git checkout -b <nombre de la rama>

**Para crear ramas se hace desde la rama main sino seria una subrama**

Para cambiar el nombre de una rama (se debe estar en la rama a modificar)

* Git Branch -m <nuevo nombre de la rama>

Para cambiar el nombre de una rama sin estar dentro de la rama

* Git Branch -m <rama a cambiar> <nuevo nombre de la rama>

Para eliminar una rama (no estar dentro de la rama a eliminar)

* Git Branch -d <rama a borrar>

Para combinar ramas (si creamos una rama para subir cosas no oficiales, pero luego se hacen oficiales y queremos combinar esa rama con la rama principal o la que tenga todo el proyecto correcto) **desde la rama donde queremos combinar**

* Git merge <rama aux>

**----------------------------------Comandos básicos--------------------------------------------**

Para entrar a una carpeta

* Cd <nombre carpeta>

Para crear una carpeta

* Mkdir <nombre de archivo o carpeta>

Para ver el contenido de una carpeta

* Ls

Para eliminar un archivo de la carpeta

* Rmdir <nombre del archivo>

Para regresar a carpetas anteriores

* Cd .. o solo cd y vuelve a la raíz o al principio