FORMACION@AdminSena MINGW64 ~

$ git version

git version 2.49.0.windows.1

FORMACION@AdminSena MINGW64 ~

$ gti config --global user.name "Jerry Colorado"

bash: gti: command not found

FORMACION@AdminSena MINGW64 ~

$ git config --global user.name "Jerry Colorado"

FORMACION@AdminSena MINGW64 ~

$ git config --global user.mail jerrycol@gmail.com

FORMACION@AdminSena MINGW64 ~

$ git config --global core.editor "code --wait"

FORMACION@AdminSena MINGW64 ~

$ git config --global -e

FORMACION@AdminSena MINGW64 ~

$ git config --global -l

user.name=Jerry Colorado

user.mail=jerrycol@gmail.com

core.editor=code --wait

FORMACION@AdminSena MINGW64 ~

$ git config --global core.autoclrf true

FORMACION@AdminSena MINGW64 ~

$ git config --global -l

user.name=Jerry Colorado

user.mail=jerrycol@gmail.com

core.editor=code --wait

core.autoclrf=true

FORMACION@AdminSena MINGW64 ~

$ ls

**AppData**/ **Desktop**/ **Impresoras**@ NTUSER.DAT **Pictures**/ **SendTo**@

**'Configuración local'**@ **Documents**/ **Links**/ NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf **Plantillas**@ **Videos**/

**Contacts**/ **Downloads**/ **'Menú Inicio'**@ NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000001.regtrans-ms **Reciente**@ ntuser.dat.LOG1

**Cookies**@ **'Entorno de red'**@ **'Mis documentos'**@ NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000002.regtrans-ms **'Saved Games'**/ ntuser.dat.LOG2

**'Datos de programa'**@ **Favorites**/ **Music**/ **OneDrive**/ **Searches**/ ntuser.ini

FORMACION@AdminSena MINGW64 ~

$ pwm

bash: pwm: command not found

FORMACION@AdminSena MINGW64 ~

$ pmw

bash: pmw: command not found

FORMACION@AdminSena MINGW64 ~

$ pwd

/c/Users/FORMACION

FORMACION@AdminSena MINGW64 ~

$ cd

FORMACION@AdminSena MINGW64 ~

$ cd downloads

FORMACION@AdminSena MINGW64 ~/downloads

$ mkdir pruebagit

FORMACION@AdminSena MINGW64 ~/downloads

$ ls

**ChromeSetup.exe**\* **Git-2.49.0-64-bit.exe**\* **VSCodeUserSetup-x64-1.99.2.exe**\* **arduino-ide\_2.3.6\_Windows\_64bit.exe**\* desktop.ini **pruebagit**/

FORMACION@AdminSena MINGW64 ~/downloads

$ cd pruebagit

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit

$ git init

Initialized empty Git repository in C:/Users/FORMACION/Downloads/pruebagit/.git/

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ ls -a

**.**/ **..**/ **.git**/

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ touch prueba\_1.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ ls

prueba\_1.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ echo prueba\_2.txt

prueba\_2.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ ls

prueba\_1.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ touch prueba\_2.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ ls

prueba\_1.txt prueba\_2.txt

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit (master)

$ cd .git

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit/.git (GIT\_DIR!)

$ code .

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit/.git (GIT\_DIR!)

$ code .

FORMACION@AdminSena MINGW64 ~/downloads/pruebagit/.git (GIT\_DIR!)

$

**1. Subir los cambios al repositorio remoto (si tienes uno configurado):**

Si ya tienes un repositorio remoto (como en GitHub o GitLab) vinculado a tu proyecto, puedes usar:

bash

CopiarEditar

git push origin main

o si estás trabajando en una rama diferente (por ejemplo, desarrollo):

bash

CopiarEditar

git push origin nombre-de-tu-rama

**2. Verificar el estado de tu repositorio:**

Si quieres asegurarte de que todo esté en orden, puedes usar:

bash

CopiarEditar

git status

Este comando te muestra el estado actual de tu repositorio: archivos modificados, archivos sin seguimiento, etc.

**3. Crear una nueva rama:**

Si quieres crear una nueva rama para seguir trabajando en otra característica:

bash

CopiarEditar

git checkout -b nueva-rama

Si todo va bien y no hay más problemas, ¡felicitaciones por el avance! ¿Te gustaría saber algo más sobre Git o algún otro tema?

**Tú dijiste:**

pero primero indicame que pasos hacer en git hub

**ChatGPT Plus**

¡Claro! Ahora que ya tienes tu primer commit local, te guiaré para **subir tu repositorio a GitHub** y empezar a trabajar con el control de versiones de manera remota. Aquí están los pasos detallados:

**1. Crear un repositorio en GitHub**

1. **Inicia sesión en GitHub**: Dirígete a [github.com](https://github.com) y entra con tu cuenta.
2. **Crear un nuevo repositorio**:
   * Haz clic en el botón verde de **"New"** (Nuevo) en la página principal o en la parte superior izquierda, junto a tu avatar.
   * Completa los siguientes campos:
     + **Repository name**: Pon un nombre para tu repositorio.
     + **Description** (opcional): Una breve descripción de tu proyecto.
     + **Public/Private**: Elige si quieres que el repositorio sea público o privado.
     + **Initialize this repository with a README**: Desmarcado, ya que ya tienes tu repositorio local y no quieres que cree un archivo README.md que entre en conflicto con lo que ya tienes.
   * Haz clic en **"Create repository"**.

**2. Vincular tu repositorio local con GitHub**

Después de crear el repositorio en GitHub, deberías ver una página con instrucciones para vincular tu repositorio local con GitHub. Aquí están los comandos que necesitas ejecutar en tu terminal:

1. **Vincula tu repositorio local al remoto**:

Copia la URL de tu repositorio en GitHub. Esta URL la puedes encontrar en la página de tu repositorio bajo **"Clone or download"** (puedes elegir HTTPS o SSH).

Si usas HTTPS, el comando sería:

bash

CopiarEditar

git remote add origin https://github.com/tu-usuario/tu-repositorio.git

Si usas SSH, sería algo como:

bash

CopiarEditar

git remote add origin git@github.com:tu-usuario/tu-repositorio.git

Asegúrate de reemplazar tu-usuario y tu-repositorio con el nombre correcto.

1. **Verifica que el remoto se haya agregado correctamente**:

Puedes comprobar que Git esté usando el remoto correctamente con:

bash

git remote -v

Deberías ver algo como esto:

bash

origin https://github.com/tu-usuario/tu-repositorio.git (fetch)

origin https://github.com/tu-usuario/tu-repositorio.git (push)

**3. Subir tu commit al repositorio remoto**

Ahora que tienes configurado el remoto, puedes hacer un **push** de tus cambios al repositorio de GitHub:

bash

git push -u origin main

**Explicación**:

* -u es una opción que establece la rama remota predeterminada para git push, de modo que no necesites escribir el nombre de la rama en futuros git push.
* main es la rama principal. Si estás usando una rama diferente (como master o develop), cámbiala por el nombre de tu rama.

**4. Verifica en GitHub**

Ve a tu repositorio en GitHub, y deberías ver tu código, el commit y todos los archivos subidos correctamente.

**5. Hacer cambios adicionales y subirlos**

Ahora, cada vez que hagas un nuevo commit en tu repositorio local, solo tendrás que hacer git push para subir los cambios a GitHub.

1. **Hacer cambios en tu código**.
2. **Añadir los cambios** con git add:

bash

git add .

(Esto añade todos los archivos modificados.)

1. **Hacer un nuevo commit**:

bash

git commit -m "Descripción del cambio"

1. **Subir los cambios a GitHub**:

bash

git push