

TUTORIAL BASICO DE GITHUB-GIT

Por: Yomin Jaramillo

OBJETIVOS

1. Crear un repo en GitHub.
2. Clonarlo en WSL (Ubuntu).

Este tutorial parte de la base de que usted tiene instalado WSL en su computador, sigalo como una introducción a Git y GitHub, si tiene dudas y o errores en su reproducción, verifique que haya seguido los pasos de forma adecuada, en caso de que persista, intente solucionar su dificultad con asistencia de IA, y si su dificultad persiste, acuda al docente.

1. INSTALA Y CONFIGURA GIT EN WSL

Actualiza paquetes (se te solicitaran tus credenciales de Ubuntu)

```
sudo apt update
```

Instala git y (opcional) GitHub CLI

```
sudo apt install -y git gh
```

Identidad (reemplaza por tus datos)

```
git config --global user.name "Yomin"
```

```
git config --global user.email "tu_email@dominio.com"
```

Rama por defecto = main y manejo de saltos de línea

```
git config --global init.defaultBranch main
```

```
git config --global core.autocrlf input
```

2. REGISTRAR CLAVE SSH

Registrar una **clave SSH** en GitHub sirve principalmente para **autenticación sin contraseña** cada vez que trabajas con un repositorio remoto.

Tú creas un **par de llaves**: una privada (guardada en tu máquina) y una pública (la registras en GitHub). Cuando haces git push o git pull, tu computadora "firma" con la llave privada.

GitHub reconoce esa firma porque tiene tu llave pública registrada → y te autentica automáticamente.

No tienes que escribir credenciales nunca más (siempre que uses esa máquina).

#Verifica si ya tienes una llave, en tu Ubuntu WSL (abre la terminal de Ubuntu, no PowerShell):

```
ls -al ~/.ssh
```

#Crea una nueva llave (usa tu propio correo obviamente)

```
ssh-keygen -t ed25519 -C "tu_correo@ejemplo.com"
```

Te pedirá dónde guardar la llave. Presiona Enter para dejar la ruta por defecto:

/home/tu_usuario/.ssh/id_ed25519

Luego pedirá una *passphrase* (contraseña opcional para proteger la llave). Puedes dejarlo vacío o poner una.

Esto genera dos archivos:

- `id_ed25519` → tu llave privada (NO la compartas).
- `id_ed25519.pub` → tu llave pública (esta sí la usaremos en GitHub).

#Iniciar el *ssh-agent* y añadir la llave

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_ed25519
```

#Copiar tu llave pública

```
cat ~/.ssh/id_ed25519.pub
```

Te mostrará algo como: `ssh-ed25519 AAAAC3Nz...etc...` tu_correo@ejemplo.com

Copia toda la clave (como lo que esta resaltado en el ejemplo)

#Registrar la llave en GitHub

Ve a [GitHub → Settings → SSH and GPG keys](#)

Haz clic en **New SSH key**

Ponle un título (ejemplo: "Ubuntu WSL laptop")

Pega la llave pública copiada

Guarda.

#Probar la conexión

```
ssh -T git@github.com
```

Si todo está bien, te saldrá algo como:

Hi Yomin! You've successfully authenticated, but GitHub does not provide shell access.

YA NO VAS A TENER QUE AUTENTICARTE CADA VEZ QUE QUIERAS HACER UN PUSH O PULL A TU REPOSITORIO REMOTO EN GITHUB!!!!

3. CREAR EL REPOSITORIO DE GITHUB

Ve a tu perfil de github.com → New repository.

Nombre: Automatizacion1.

Público/Privado (Recomiendo publico), y (opcional) marca Add README y .gitignore (Python).

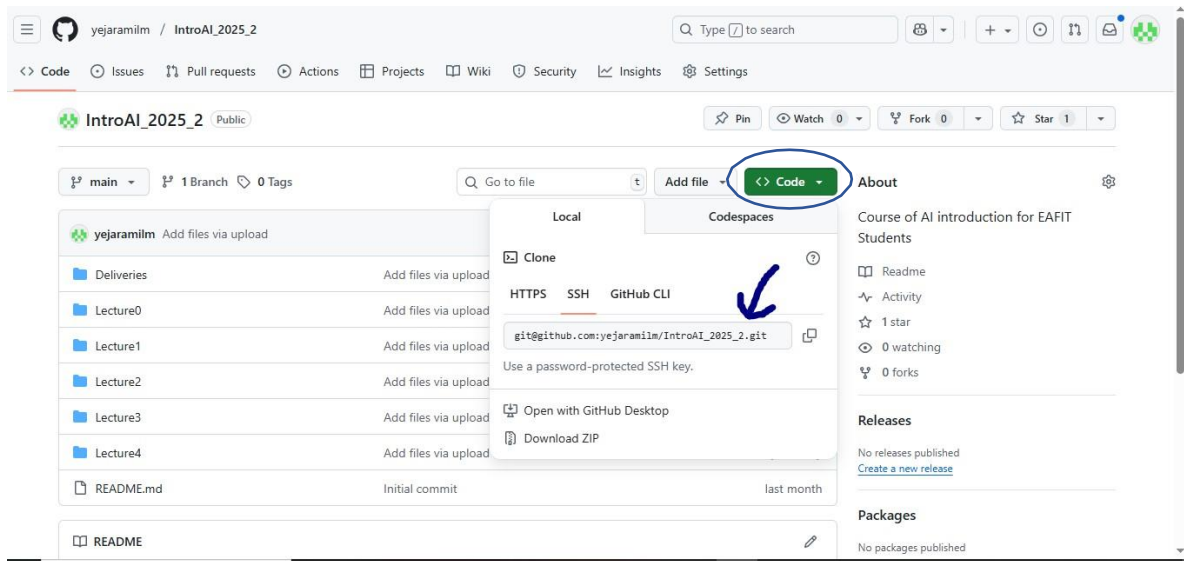
Crea el repo.

Crea una carpeta en tu computador con los archivos del laboratorio llamada Laboratorio1. Luego, agrega la carpeta a tu repositorio de Github.

4. CLONA EL REPO EN WSL

```
cd ~ //esto te lleva a la carpeta raíz, usalo si no estas parado ahí
```

En el code de tu repositorio, vas a encontrar el link de ssh que te permite generar la clonación a nivel local



Copia el link y ejecuta lo siguiente para generar la clonación:

```
git clone git@github.com:<tu-usuario>/mi-proyecto.git
```

Ejemplo:

```
yominjaramillo@DESKTOP-86CFT2K:~$ git clone git@github.com:yejaramilm/IntroAI_2025_2.git
Cloning into 'IntroAI_2025_2'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 49 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (49/49), 20.14 MiB | 1.30 MiB/s, done.
Resolving deltas: 100% (6/6), done.
yominjaramillo@DESKTOP-86CFT2K:~$
```

Despues de esto, para ver la carpeta del proyecto usa:

```
ls
```

Y para entrar usa:

```
cd nombre_de_tu_proyecto
```

5. EXPLORACION DE COMANDOS BASICOS PARA TRABAJO EN GIT

Ver estado

```
git status
```

Agregar archivos (todo o selectivo)

```
git add .
```

Confirmar (commit) con mensaje corto y claro
git commit -m "Texto descriptivo de los cambios realizados"

Subir a GitHub (primera vez con upstream)
git push -u origin main

Traer cambios antes de trabajar
git pull --rebase