

Pasos para configurar Github

Universidad del Valle

Monitorias Programacion Imperactiva

Alejandro Sierra

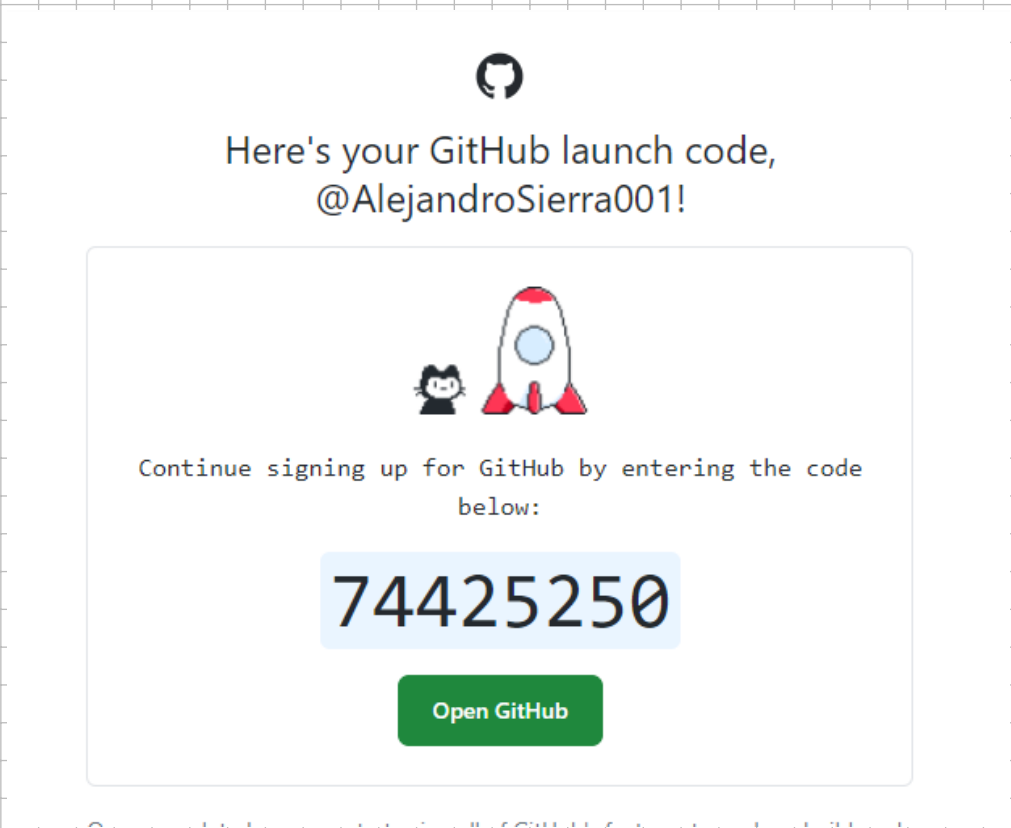
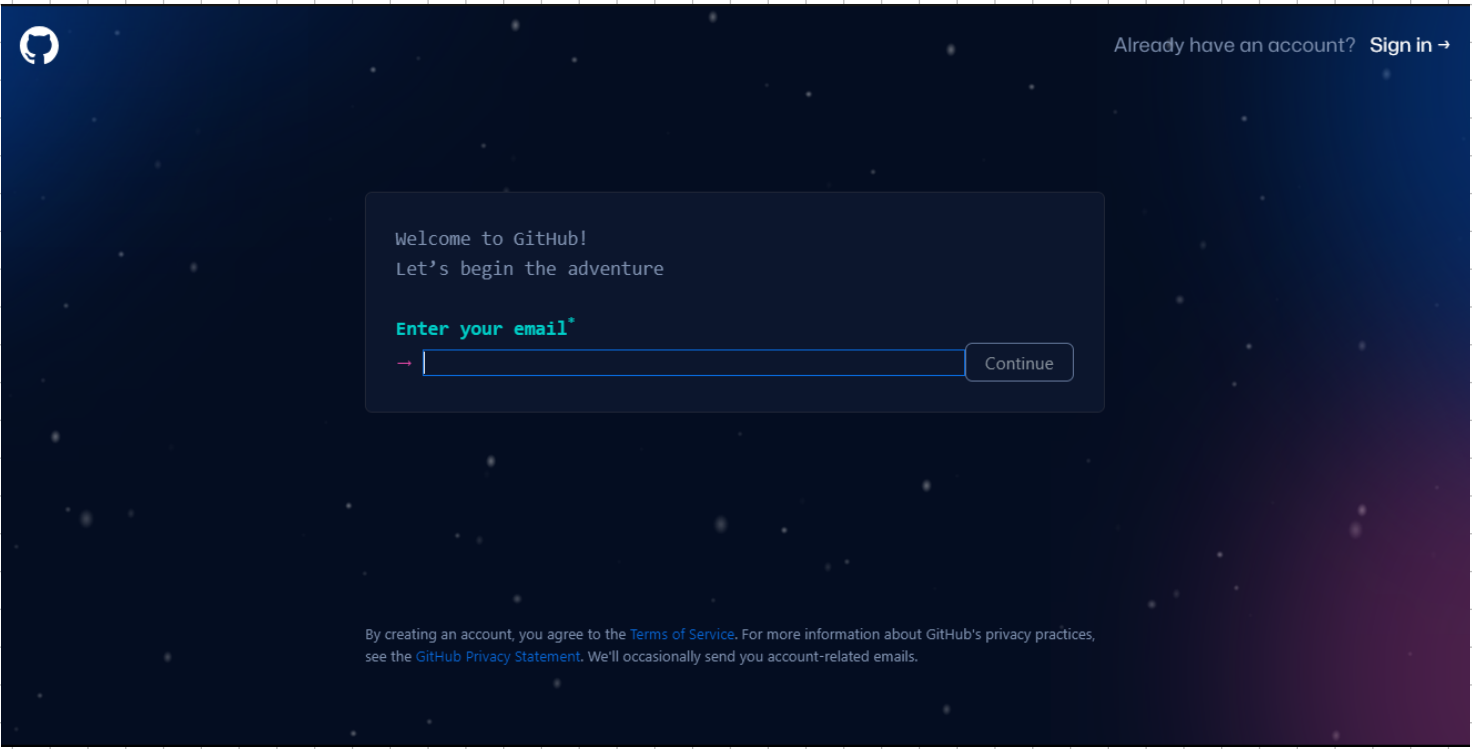
Ingeneria en Sistemas

Links de aplicaciones necesarias:

<https://git-scm.com/download/win>

Link donde se explica la creacion de un repositorio :

[https://www.youtube.com/watch?v=L\\_IWQZNhN7w](https://www.youtube.com/watch?v=L_IWQZNhN7w)



Paso 1 :

Debemos registrarnos en la plataforma de github, la idea es que sea con el correo institucional.

Completar todos los campos, una vez termina el formulario se solicita un codigo de verificación.

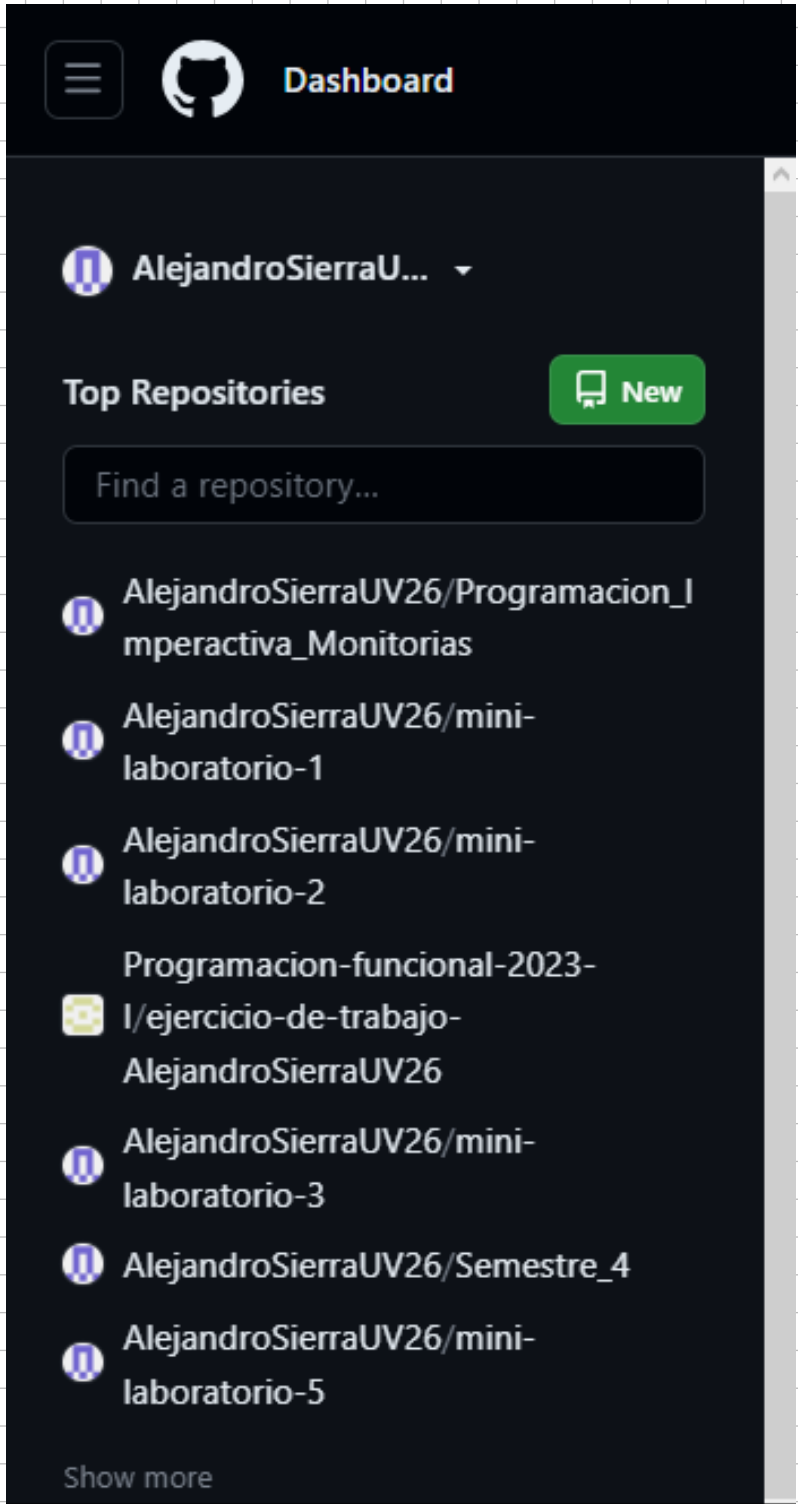
Paso 2 :

Una vez se ha validado el correo y se logro crear la cuenta, nos pondremos a realizar el paso a paso de un repositorio y conectarlo a nuestra zona de trabajo.

Paso 3:

Para crear un nuevo repositorio debemos primero centrarnos en las opciones que aparecen en la zona izquierda. Lugar donde seran rutas de acceso para nuestros repositorios existentes.

Debemos presionar el boton verde "New".



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

Owner \*

AlejandroSierraUV26

Repository name \*

Great repository names are short and memorable. Need inspiration? How about [supreme-goggles](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None**

Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

You are creating a public repository in your personal account.

Create repository

Paso 4: Se nos genera un formulario donde debemos completar la informacion pertinente a nuestro repositorio.

Si inicio sesion con la cuenta institucional nos pertimitira hacer nuestro repositorio privado. Sino fue asi, solo esta disponible por limitaciones de repositorios privados.

Como tú apenas estas aprendiendo a manejar la plataforma, se recomienda que los campos solo se llene:

- \*Nombre del repositorio
- \*Descripcion
- \*Public Repository

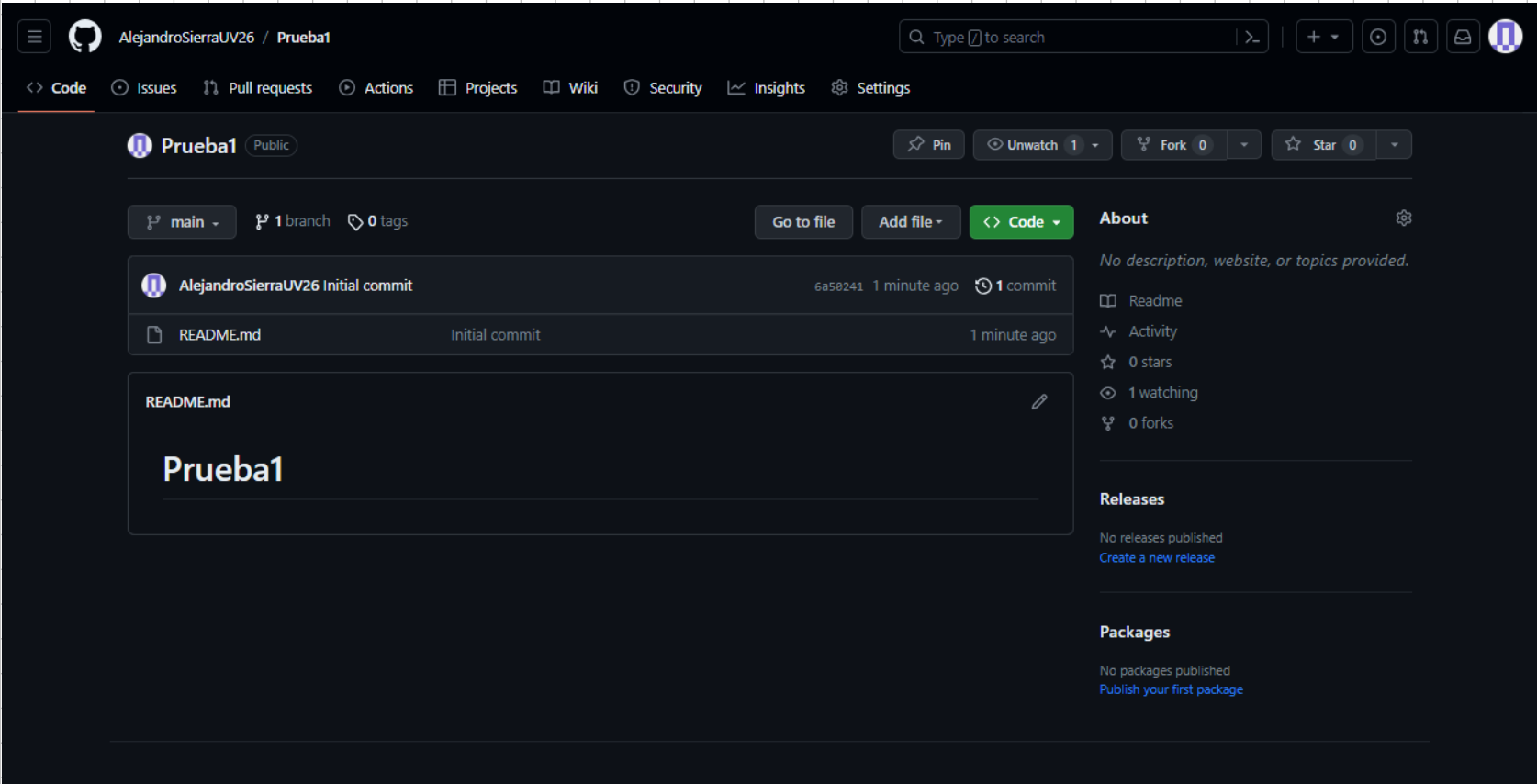
De resto no es necesario por el momento.

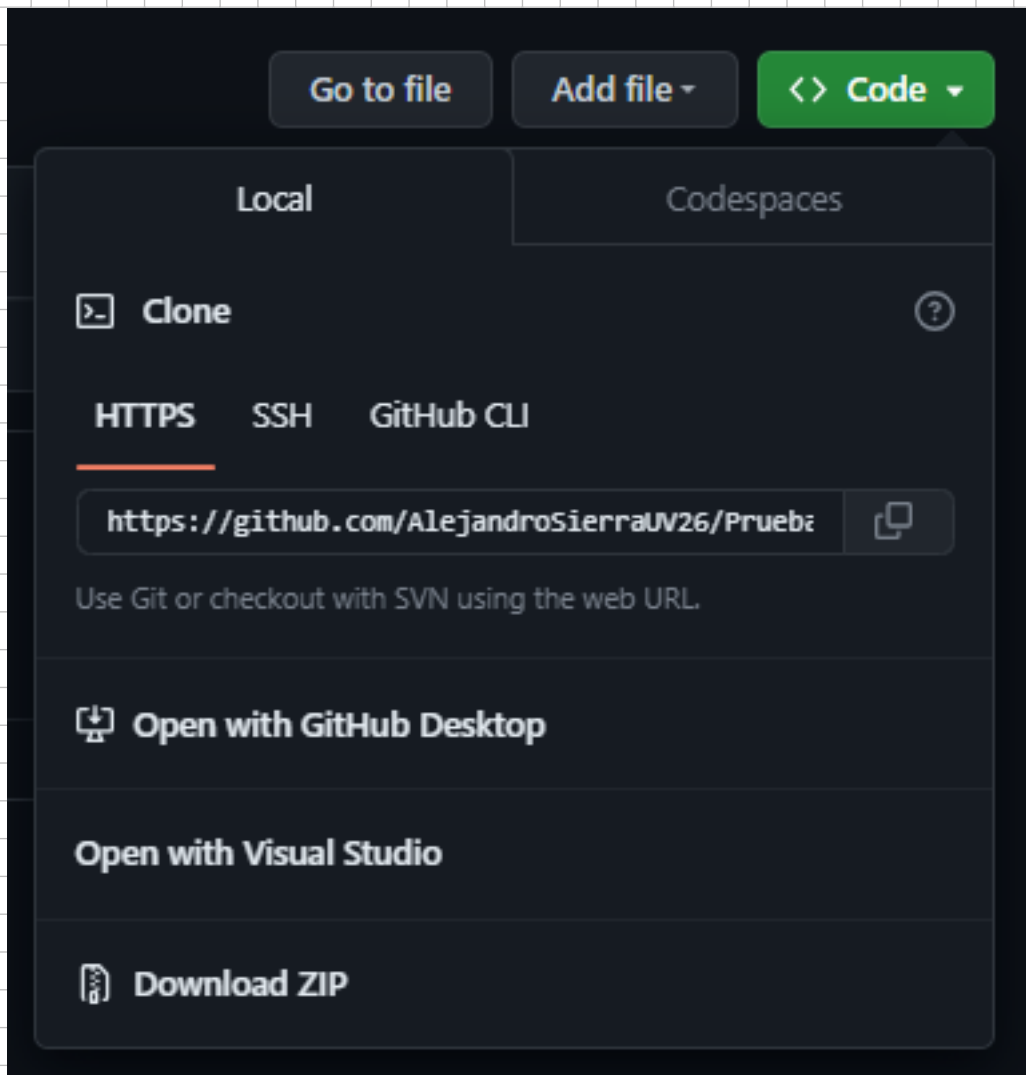
Paso 5:

El repositorio esta hecho, ahora debemos conectarlo a nuestra carpeta que queremos utilizar para modificar, crear y eliminar archivos..

Lo recomendable es crear una carpeta donde podamos encontrarla facilmente, esta carpeta la usaremos para enlazarla en el repositorio.

Usaremos la consola de Visual Studio Code, con la terminal de PowerShell.





Paso 6 :

Para tener la ruta de conexion debemos ir al boton verde, e el apartado Local, y debemos copiar la ruta HTTPS. Esta sera la que permita que nuestro repositorio tenga una carpeta remota.

Paso 7:

En VSC vamos a iniciar con los siguientes comandos. Tener en cuenta que esto solo se debe hacer una vez para la configuración del repositorio.

```
echo "# Prueba1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/AlejandroSierraUV26/Prueba1.git
git push -u origin main
```

- >git init ----> Inicia un archivo (.git) en nuestra carpeta que permitira la conexion.
- >git add "Nombre\_Archivo"--> actualiza la informacion del archivo para subirlo.
- >git commit -m -->"Mensaje" Permite darle un mensaje a la version del codigo.
- >git branch -M main -->Prestablece la rama principal del repositorio
- >git remote add origin <Ruta\_Repositorio> ----> Conecta el repositorio con la carpeta local.
- >git push -u origin main ----> Este comando hace el cambio en el repositorio, la primera vez debe ser de esta manera.  
Una vez se realiza, solo es necesario poner >git push y se actualizara solo.

Tener en cuenta que los pasos para hacer un cambio en un repositorio una vez se configura son solo 3 en el mismo orden.

( git add . ) ----> Actualiza todos los elementos de la carpeta

(git commit -m "Mensaje") ----> Les da un titulo al cambio realizado

(git push) ----> Sube el respectivo cambio al repositorio.



Aqui hay un ejemplo de como deberia mostrarse el codigo, en Visual Studio Code.

```
PS C:\Users\Alejandro\Desktop\Github> git init
Initialized empty Git repository in C:/Users/Alejandro/Desktop/Github/.git/
PS C:\Users\Alejandro\Desktop\Github> git add .
PS C:\Users\Alejandro\Desktop\Github> git commit -m "Implementacion Ejemplo"
[master (root-commit) 885a774] Implementacion Ejemplo
 1 file changed, 9 insertions(+)
 create mode 100644 Ejemplo.py
PS C:\Users\Alejandro\Desktop\Github> git branch -M main
PS C:\Users\Alejandro\Desktop\Github> git remote add origin https://github.com/AlejandroSierraUV26/Prueba1.git
PS C:\Users\Alejandro\Desktop\Github> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 392 bytes | 392.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AlejandroSierraUV26/Prueba1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Alejandro\Desktop\Github> █
```

Por cada "Push" realizado se vera reflejado en el repositorio creado.

