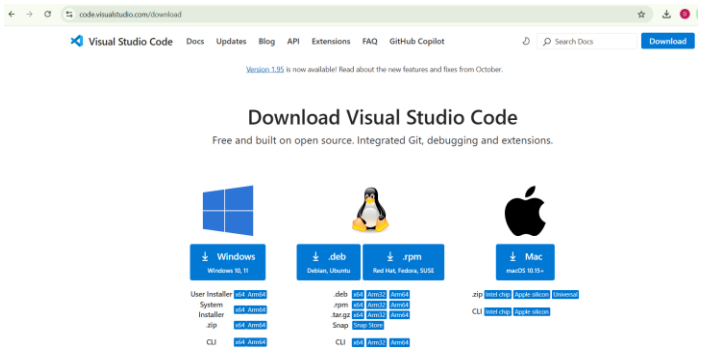
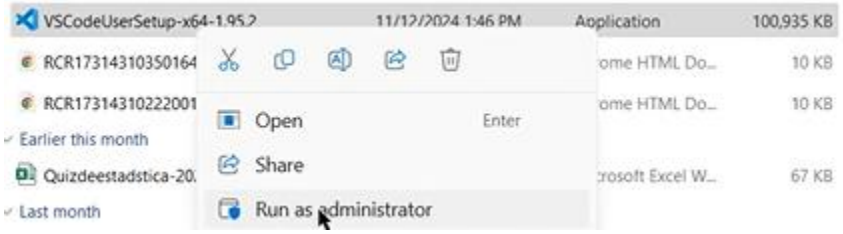
Instalación de entorno y uso de GitHub (versión 2024.11.16)

1. Instalar visual studio code

https://code.visualstudio.com/download

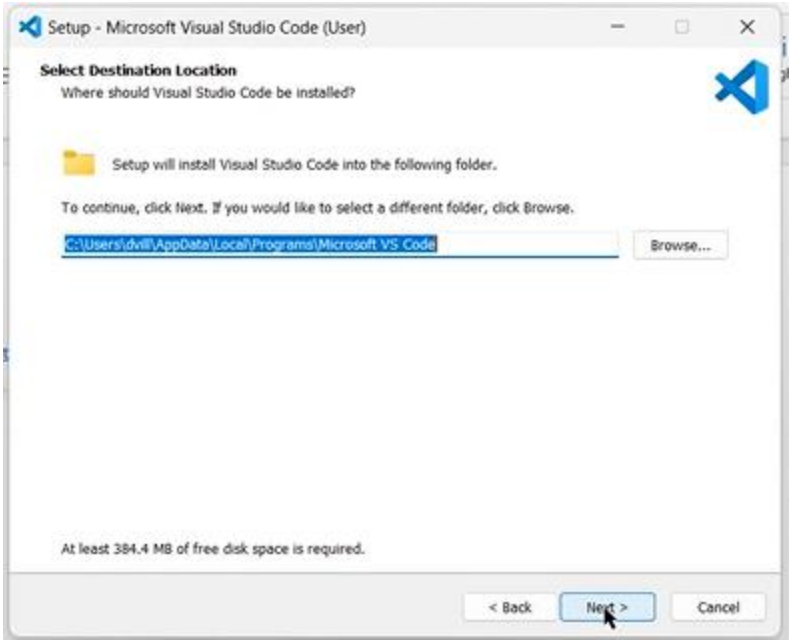


Ir a la carpeta de descargas y dar click en correr como administrador

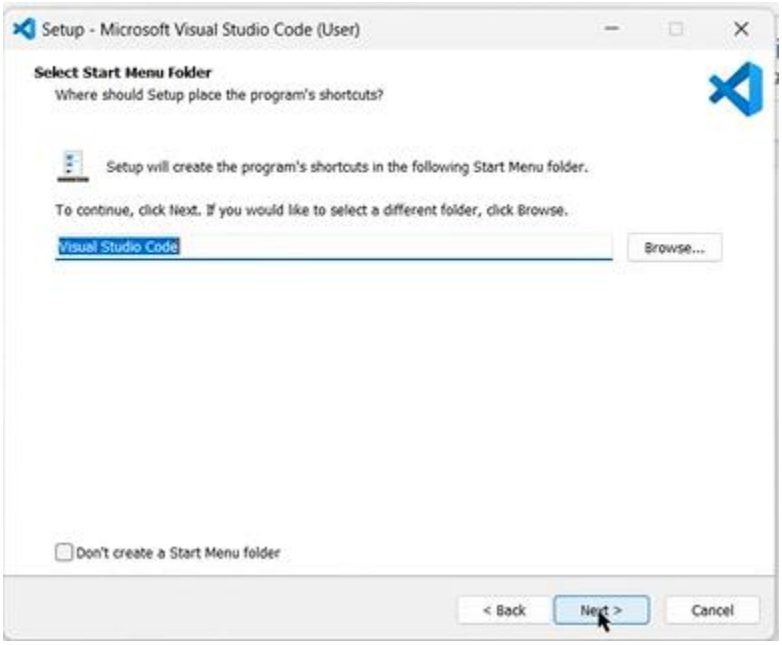


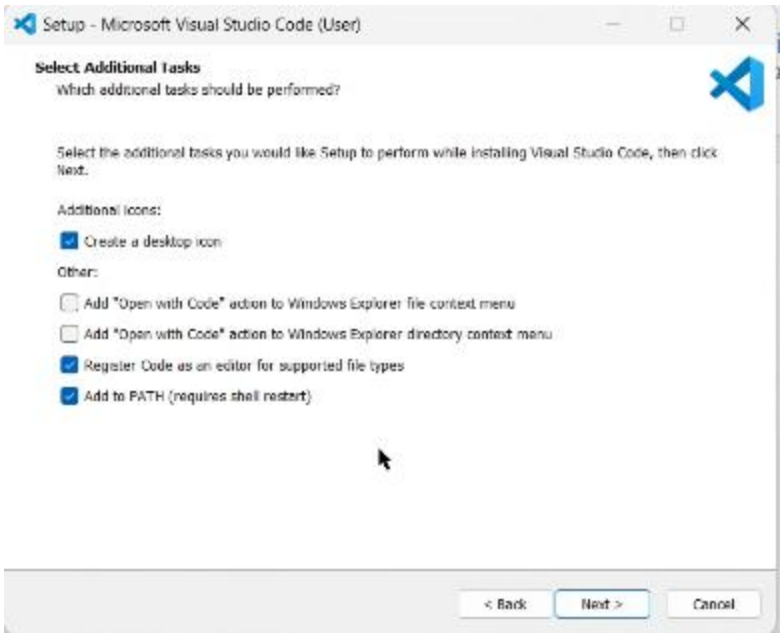
Aceptar la licencia

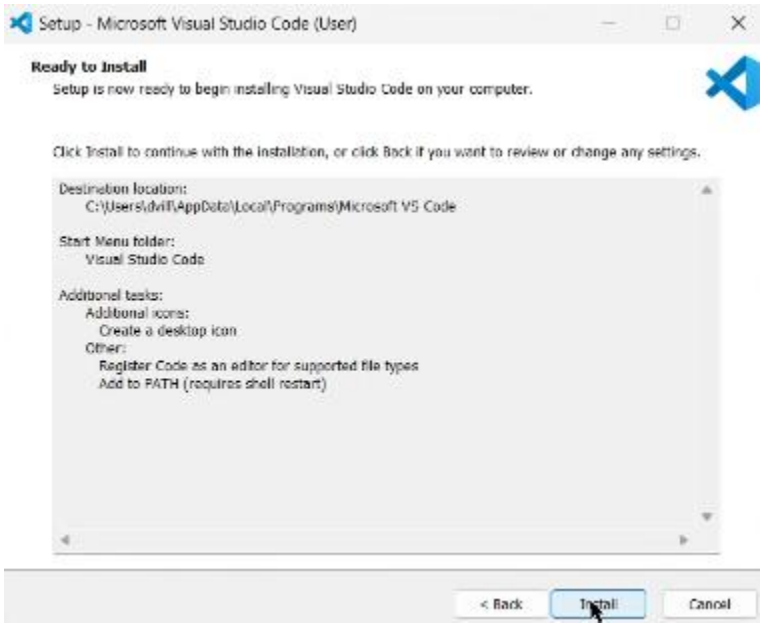
Click en Next o Continuar

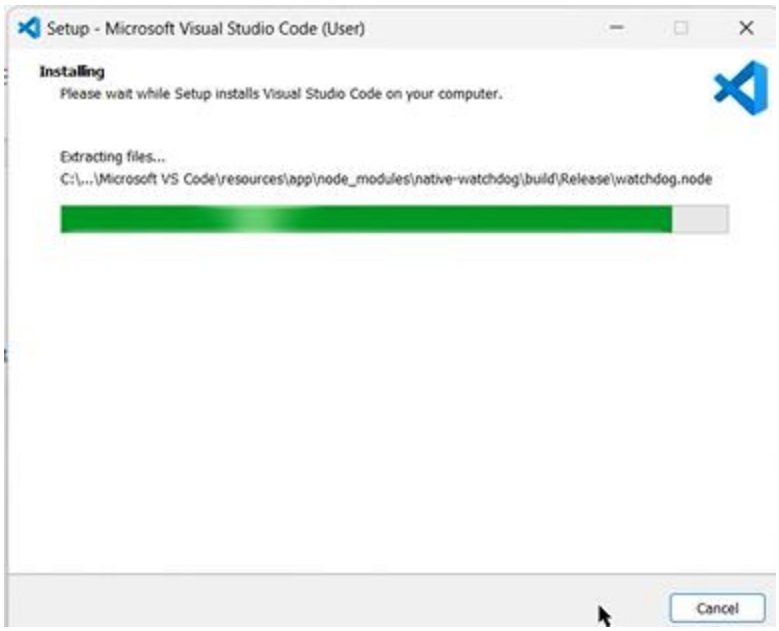


Dar clic en Next o Continuar



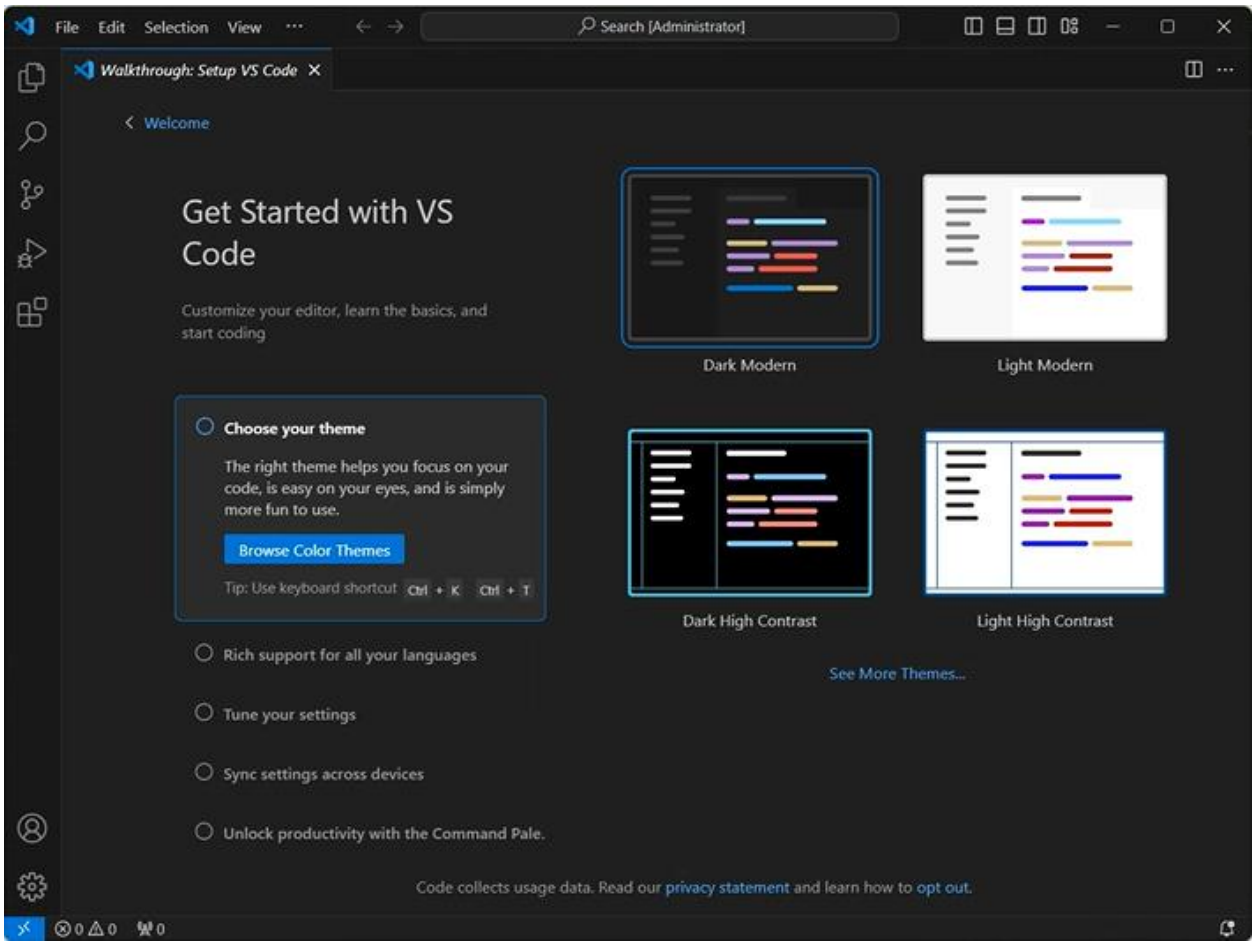

Setup - Microsoft Visual Studio Code (User)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.
Additional icons:
Create a desktop icon
Other:
Add "Open with Code" action to Windows Explorer file context menu
Add "Open with Code" action to Windows Explorer directory context menu
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
< Back
Next >
Cancel


Setup - Microsoft Visual Studio Code (User)
Ready to Install
Setup is now ready to begin installing Visual Studio Code on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Users\dvil\AppData\Local\Programs\Microsoft VS Code
Start Menu folder:
Visual Studio Code
Additional tasks:
Additional icons:
Create a desktop icon
Other:
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
< Back
Install
Cancel


Setup - Microsoft Visual Studio Code (User)
Installing
Please wait while Setup installs Visual Studio Code on your computer.
Extracting files...
C:\...\Microsoft VS Code\resources\app\node_modules\native-watchdog\build\Release\watchdog.node
Cancel

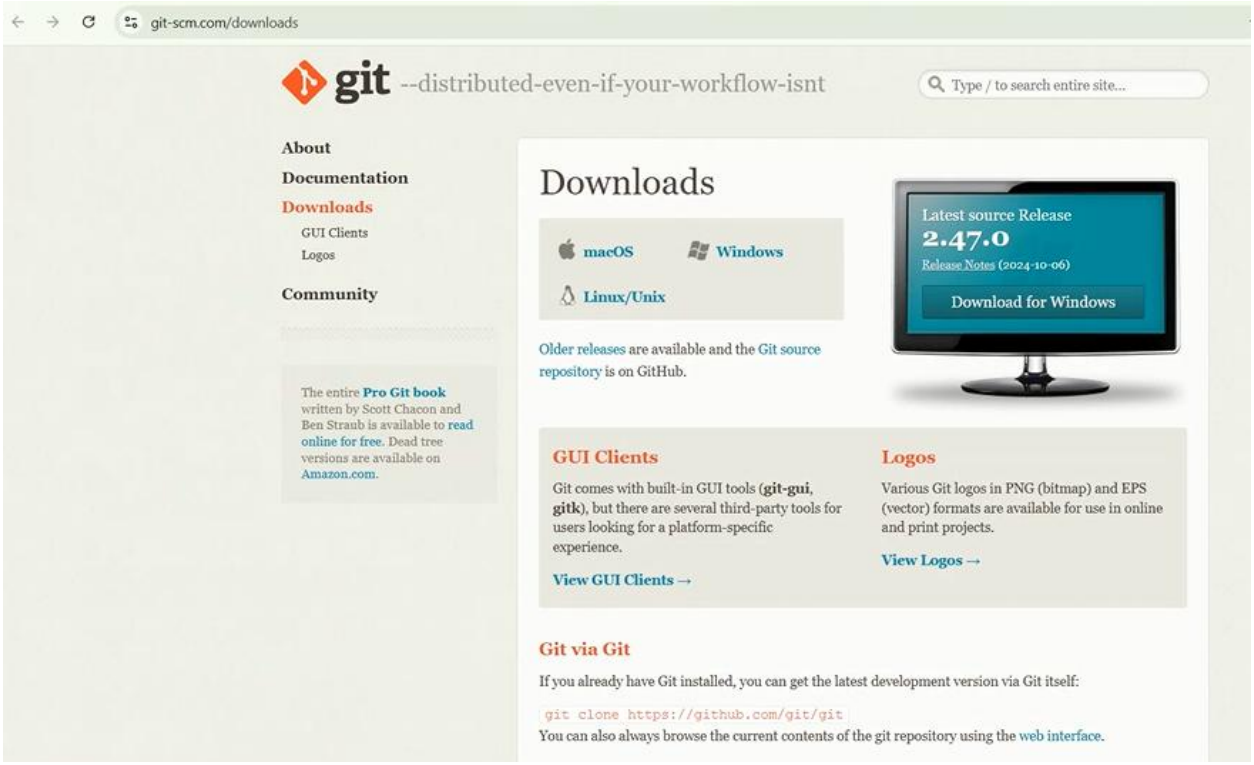

Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

Se abre VSC

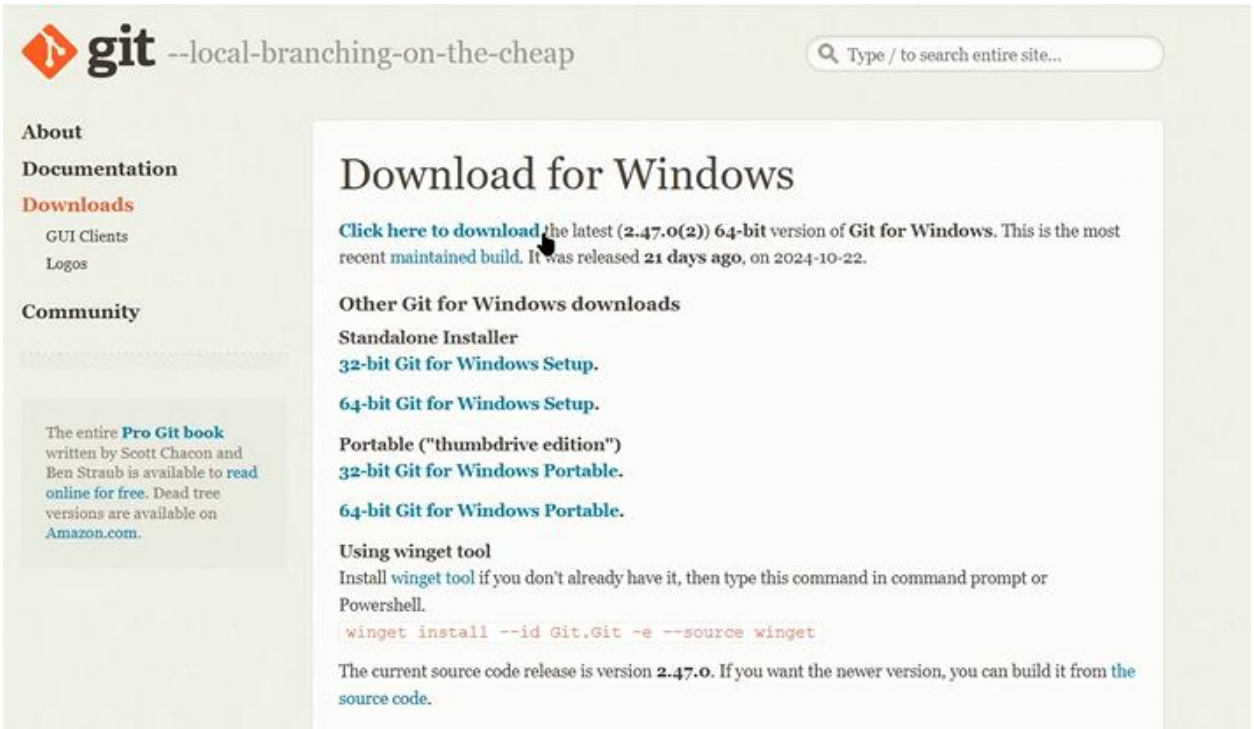

File
Edit
Selection
View
Search [Administrator]
Walkthrough: Setup VS Code
Welcome
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Dark Modern
Light Modern
Choose your theme
The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.
Browse Color Themes
Tip: Use keyboard shortcut Ctrl + K Ctrl + T
Dark High Contrast
Light High Contrast
See More Themes...
Rich support for all your languages
Tune your settings
Sync settings across devices
Unlock productivity with the Command Pale.
Code collects usage data. Read our privacy statement and learn how to opt out.

2. Instalar GIT

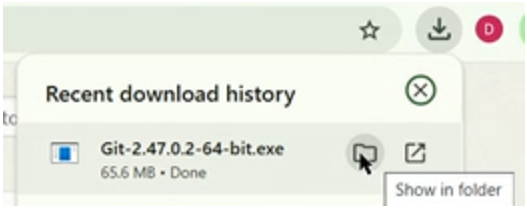
https://git-scm.com/downloads



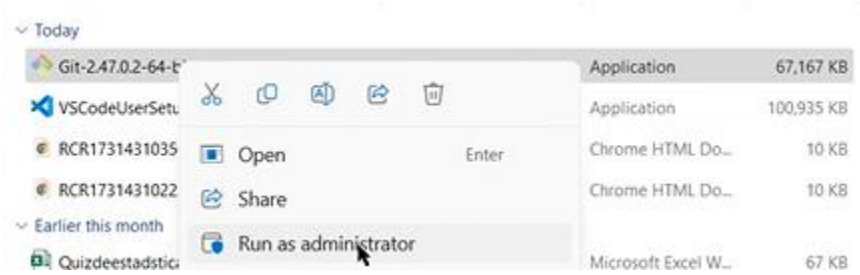
Dar click en sistema operativo por ejemplo Windows



Abrir folder descargas

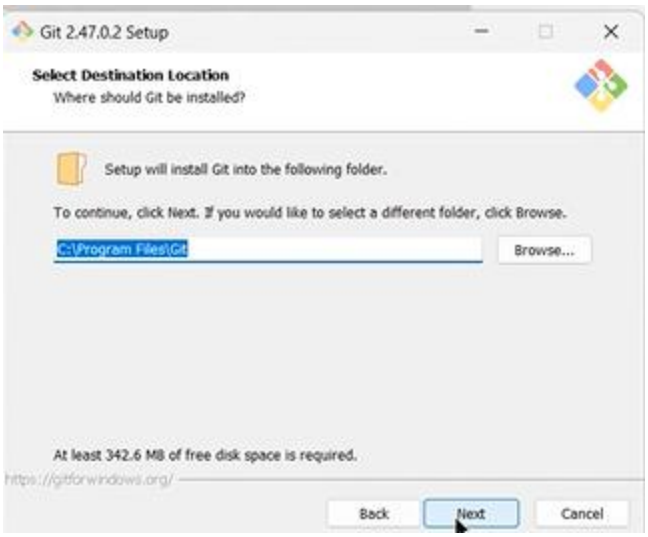


Ejecutar como administrador

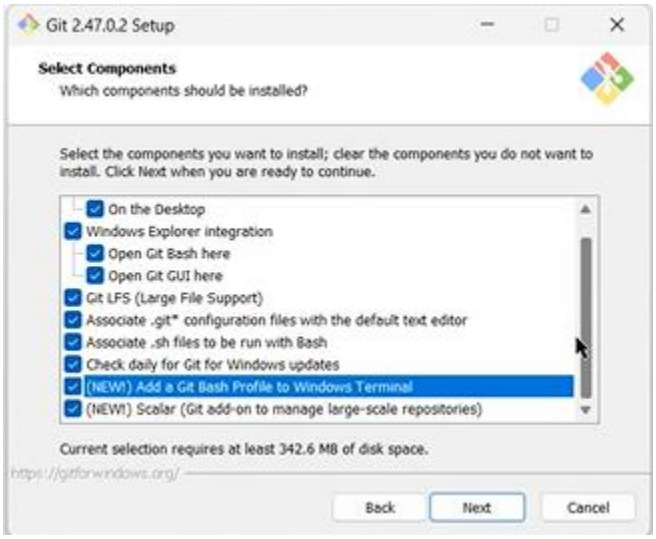


Next

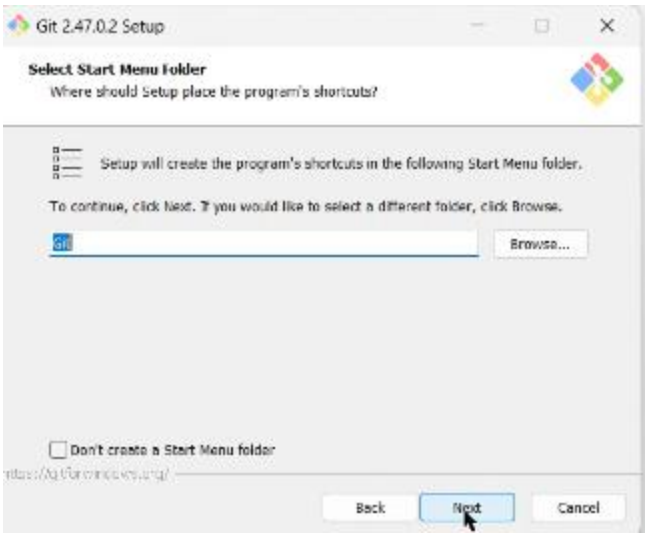
Next



Seleccionas las opciones

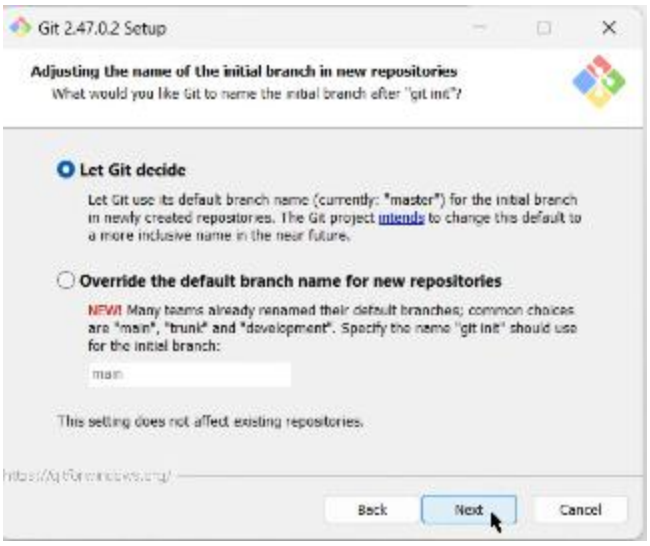


Next

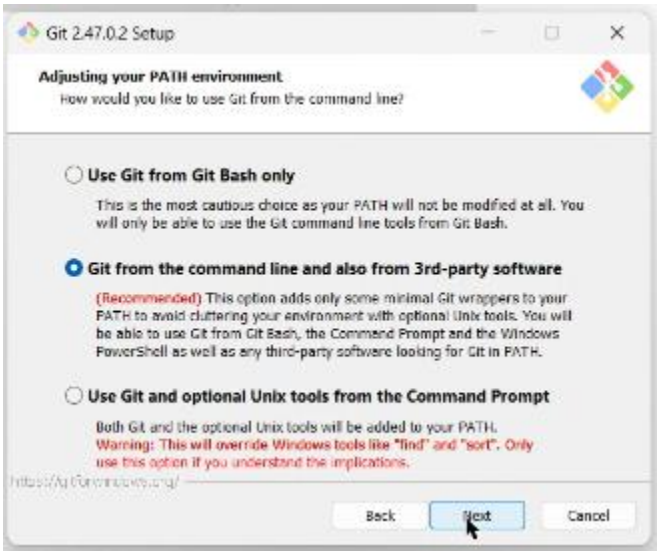


Seleccionar Visual Studio Code como editor default

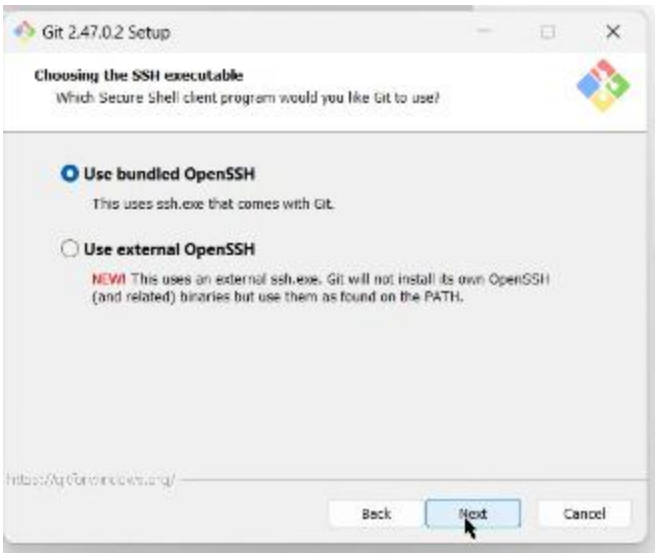
Next



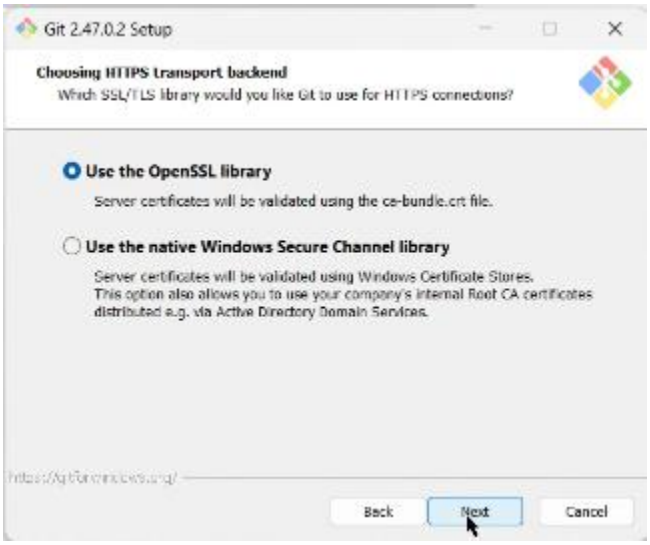
Escoger



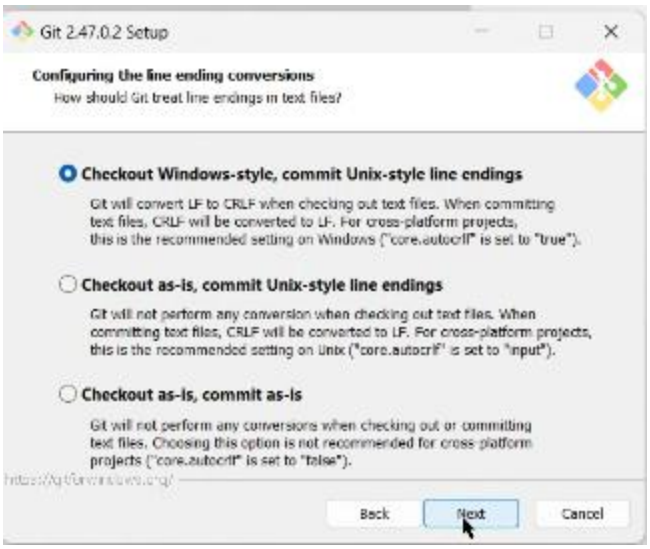
Next


Git 2.47.0.2 Setup
Choosing the SSH executable
Which Secure Shell client program would you like Git to use?
Use bundled OpenSSH
This uses ssh.exe that comes with Git.
Use external OpenSSH
NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.
Back
Next
Cancel

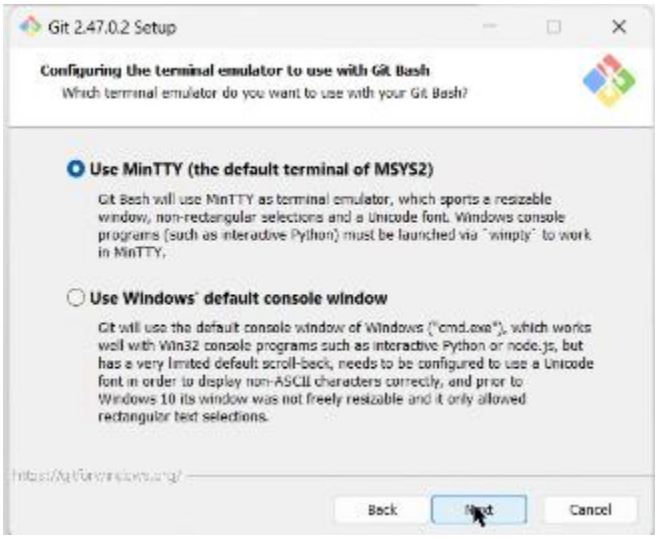
Next


Git 2.47.0.2 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
Back
Next
Cancel

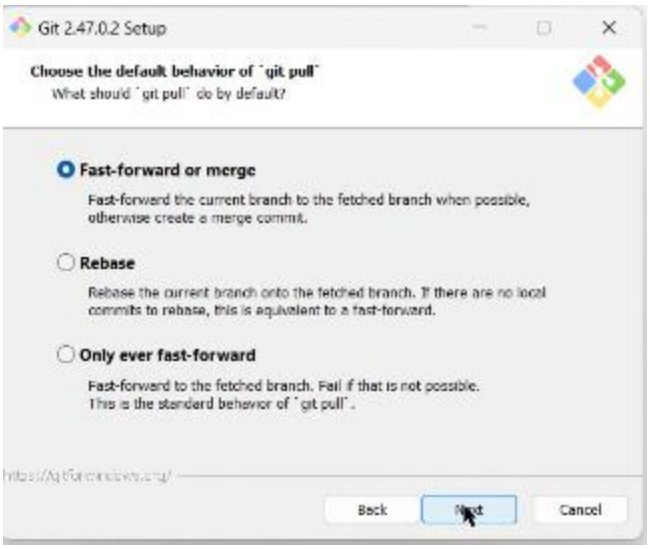
Next


Git 2.47.0.2 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
Back
Next
Cancel

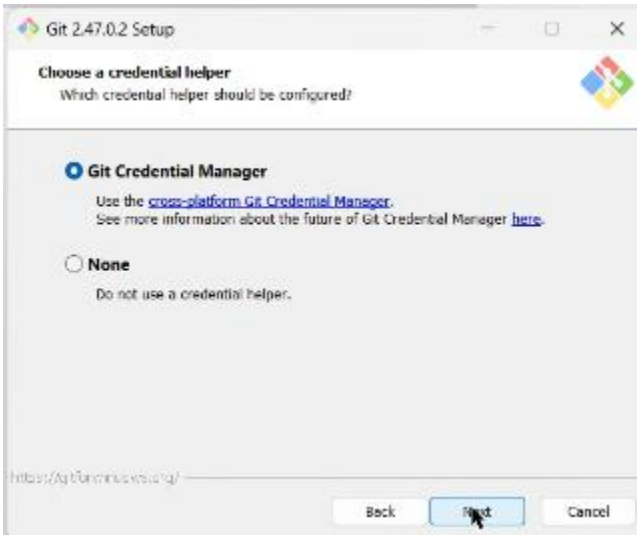
Next


Git 2.47.0.2 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
Back
Next
Cancel

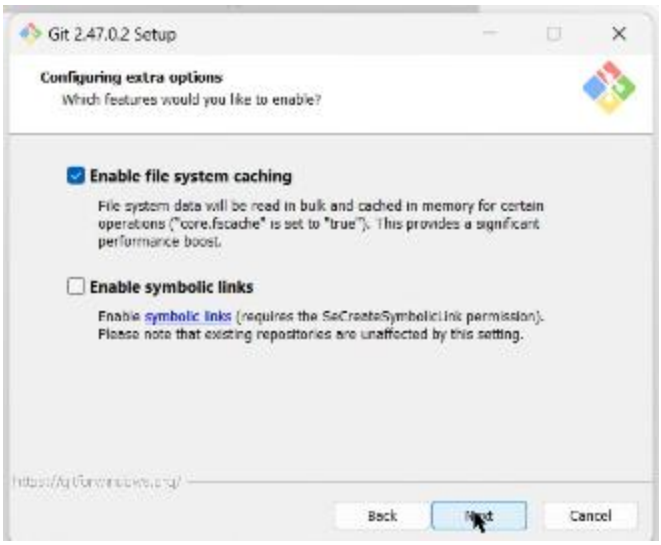
Next



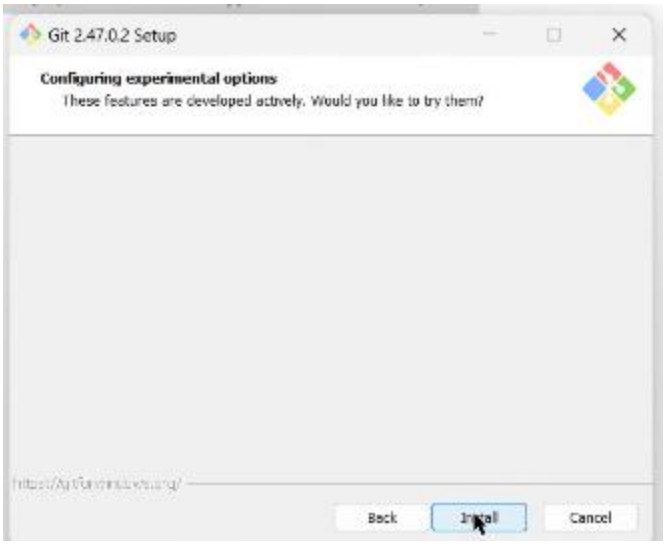
Next

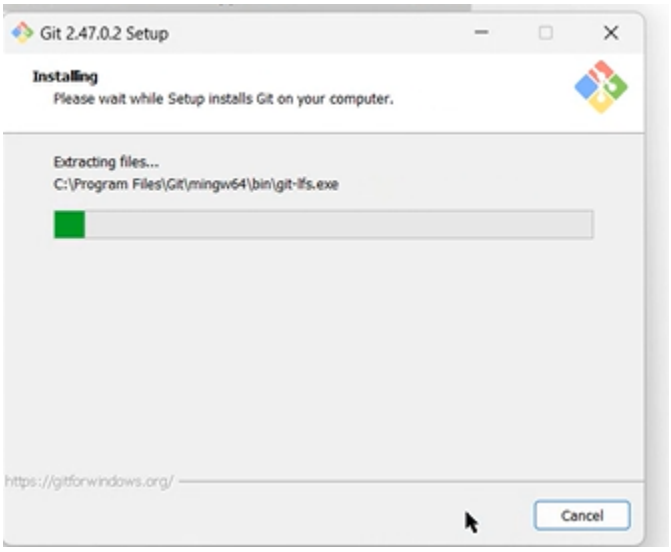


Enable file system caching


Git 2.47.0.2 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
Back
Next
Cancel

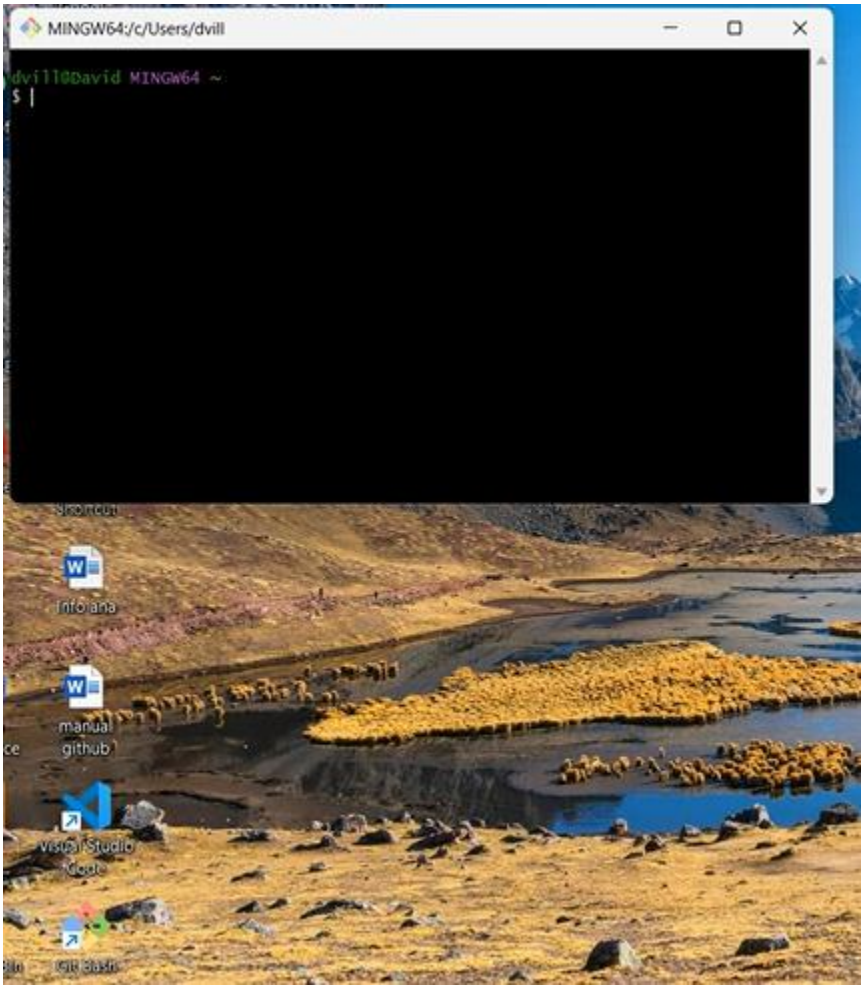
Install


Git 2.47.0.2 Setup
Configuring experimental options
These features are developed actively. Would you like to try them?
Back
Install
Cancel


Git 2.47.0.2 Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
C:\Program Files\Git\mingw64\bin\git-lfs.exe
https://gitforwindows.org/
Cancel

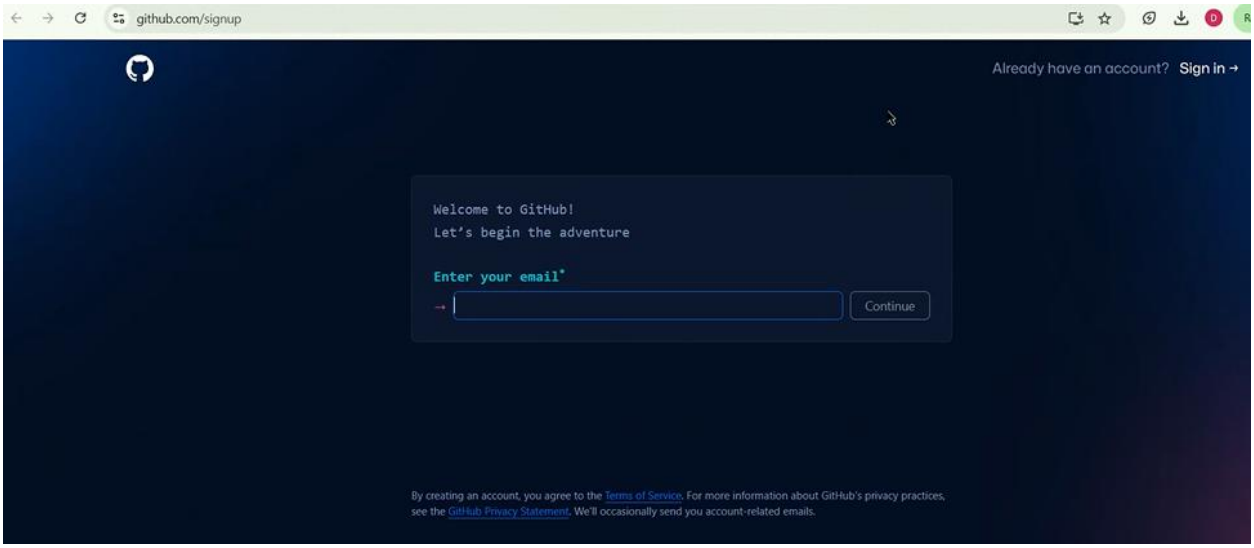
Finish

Instalación de GIT, se puede cerrar esta ventana

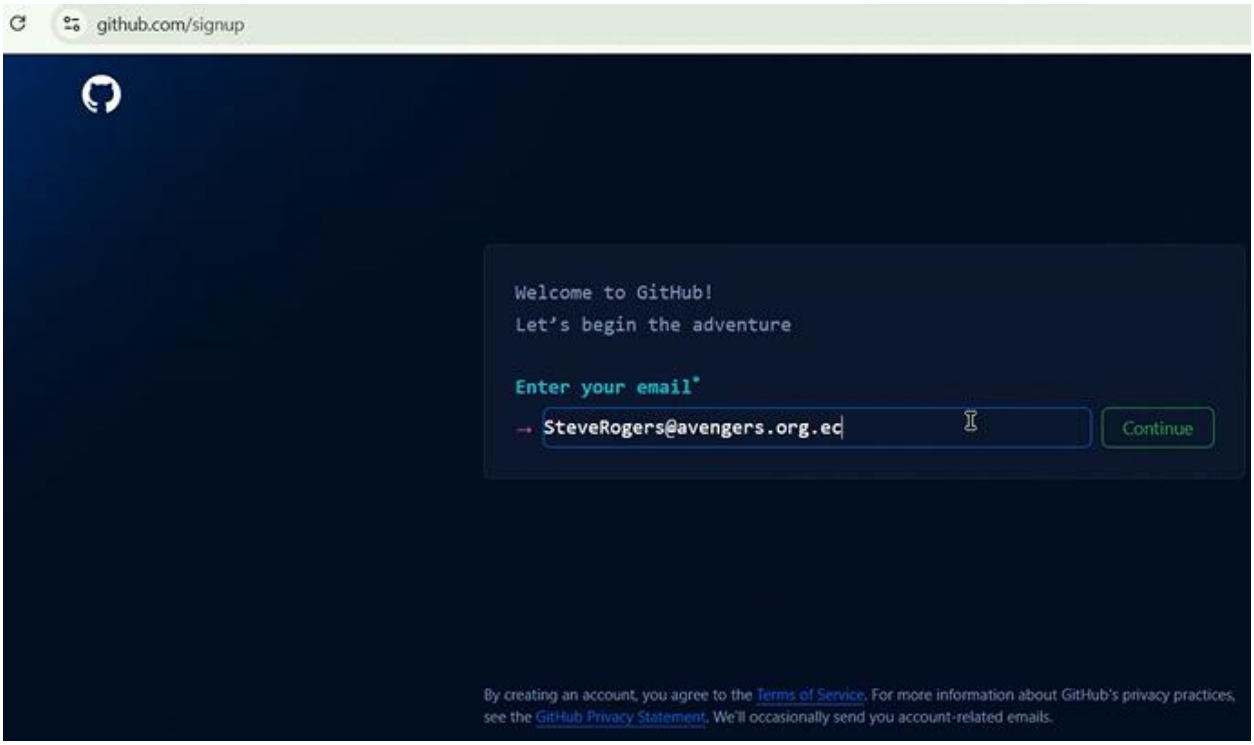


3. Crear cuenta de GitHub, ingresa a la siguiente página web

https://github.com/signup

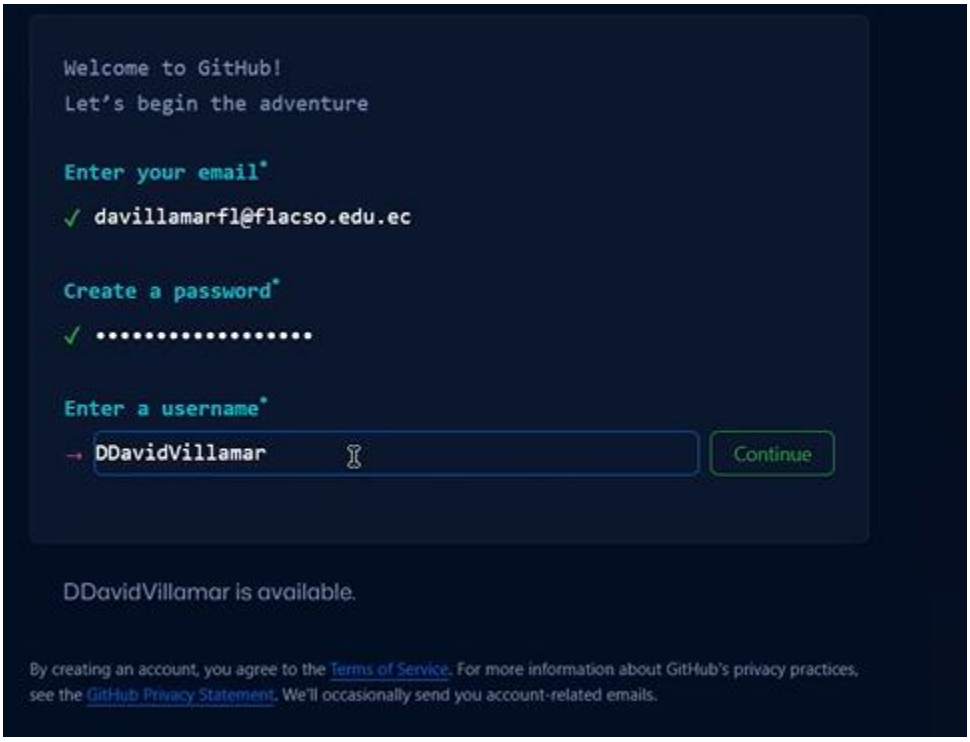


Ingresa el correo con el cual quieres usar GitHub y dar click en Continue

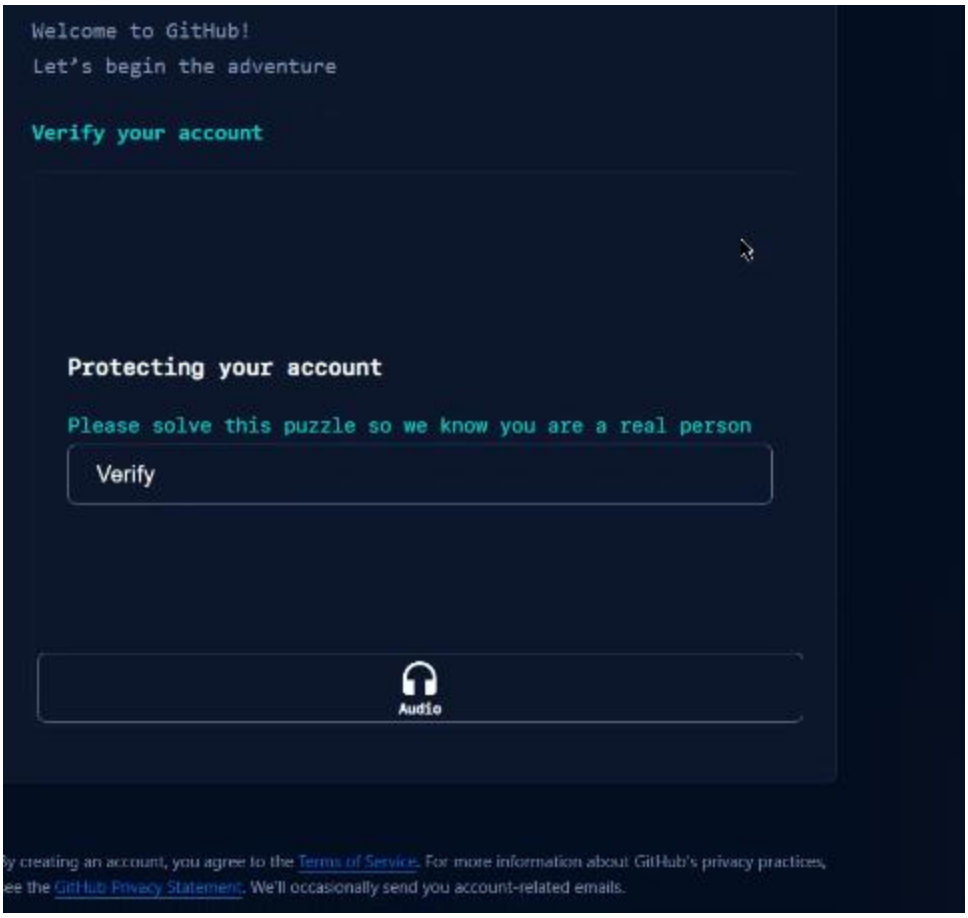


Ingresa un password que no hayas usado en otra web y que lo recuerdes para usarlo para ingresar a tu cuenta de GitHub

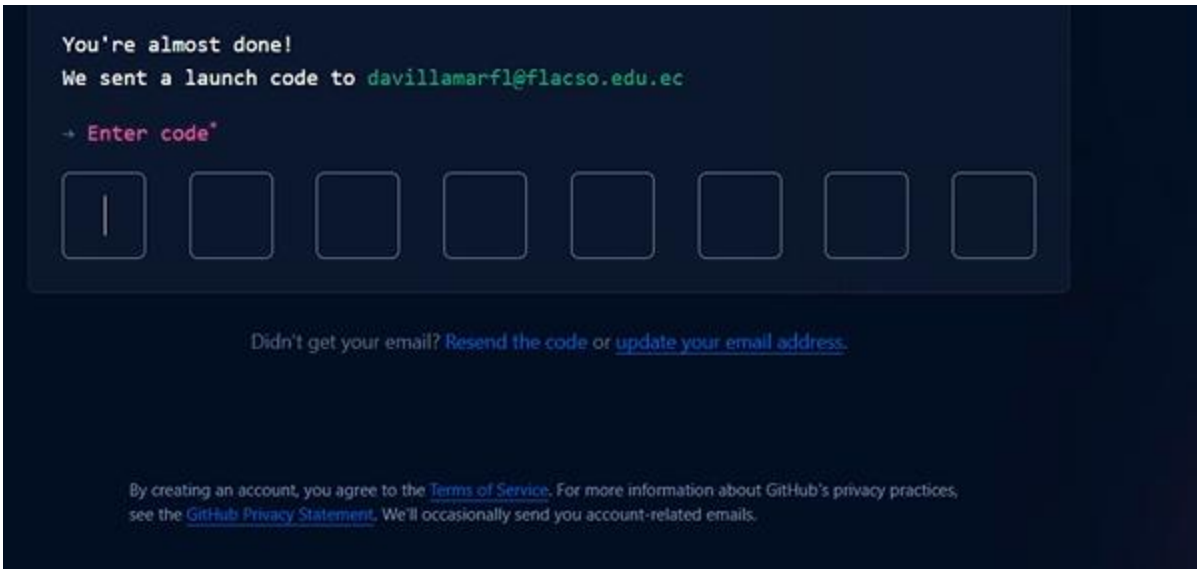
Ingresa un username



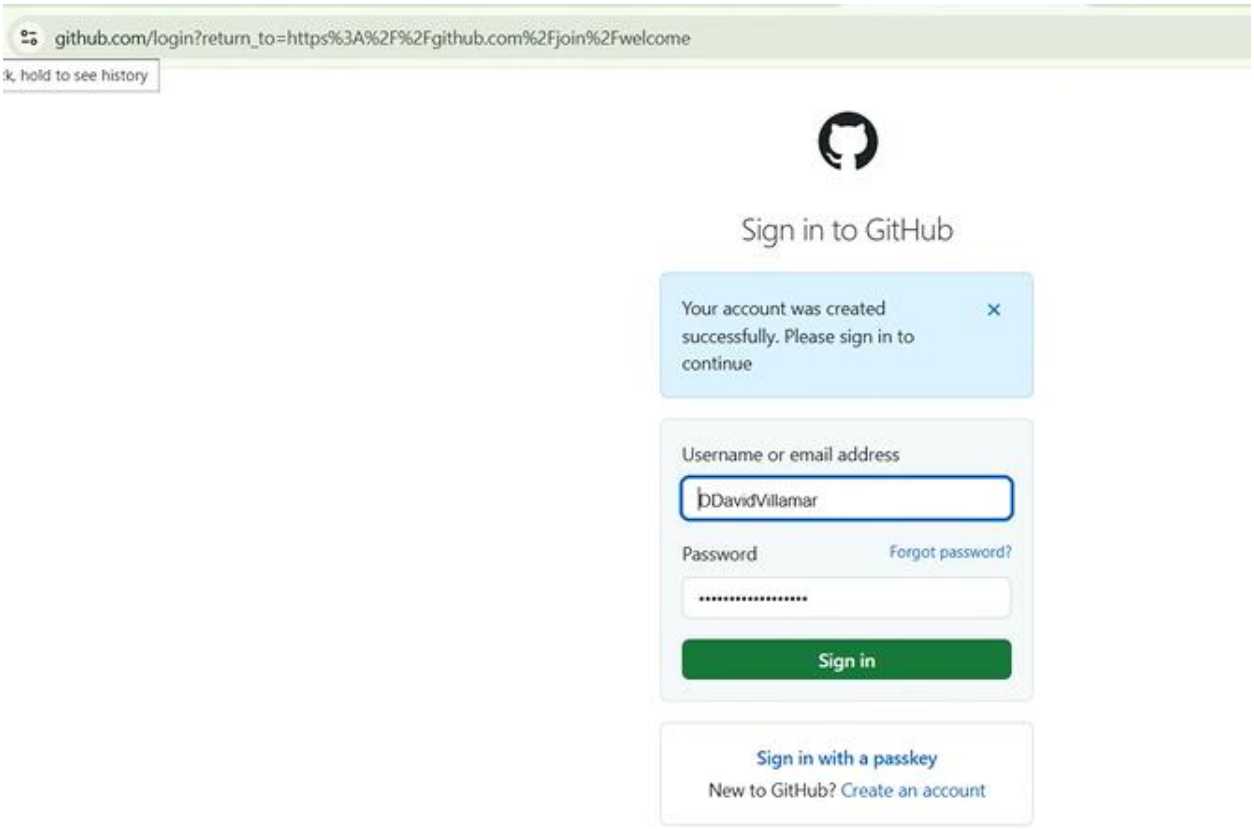
Verifica tu cuenta dando click en Verify



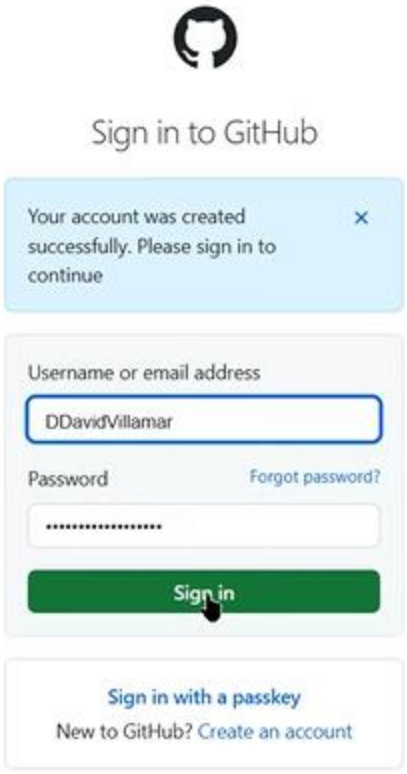
Ingresa el código que te llega al correo



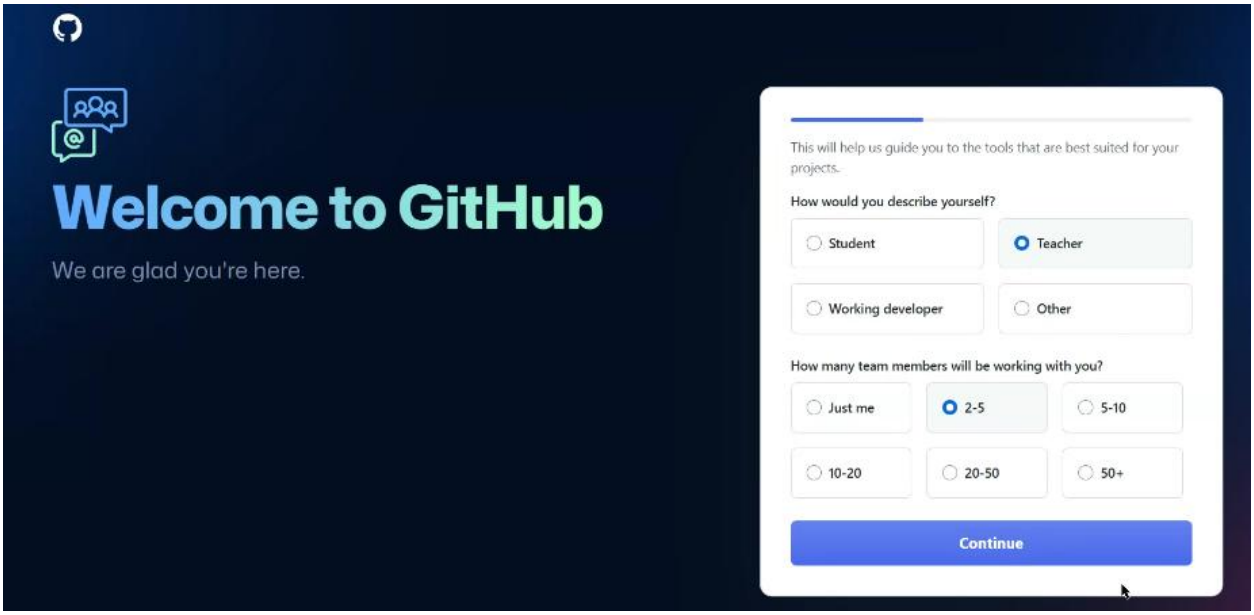
Luego de ingresar tu código vas al la página de inicio de tu nueva cuenta de GitHub



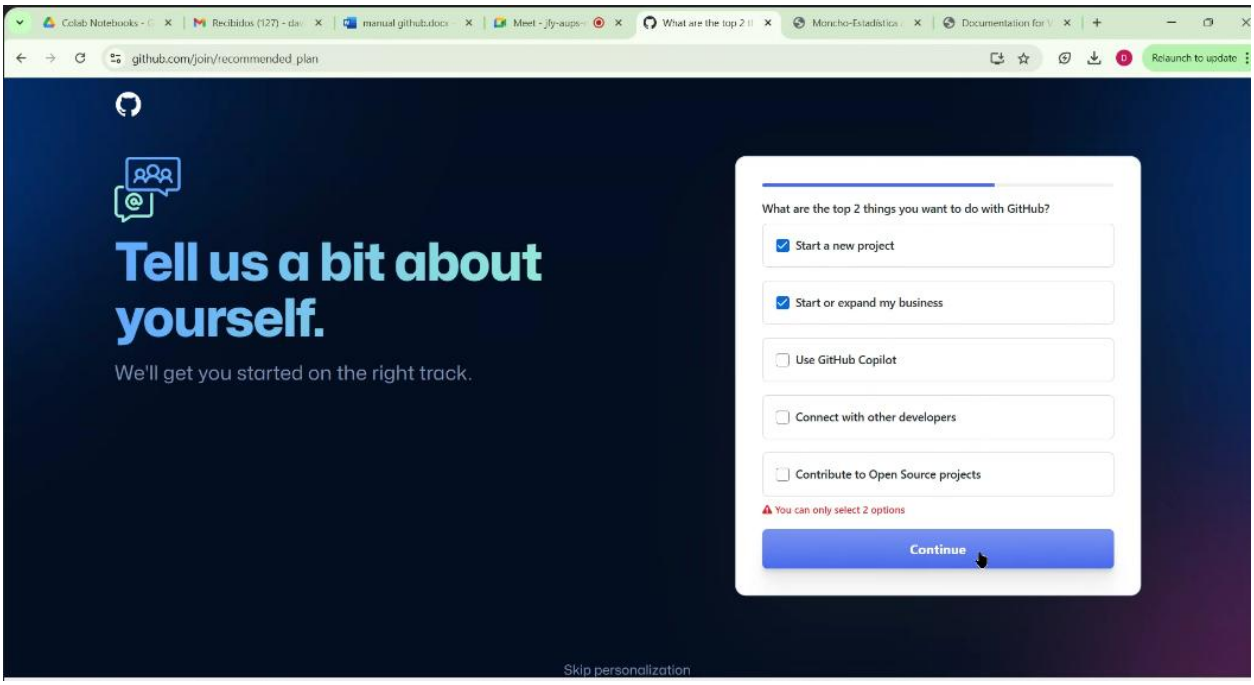
Dar click en Sign in



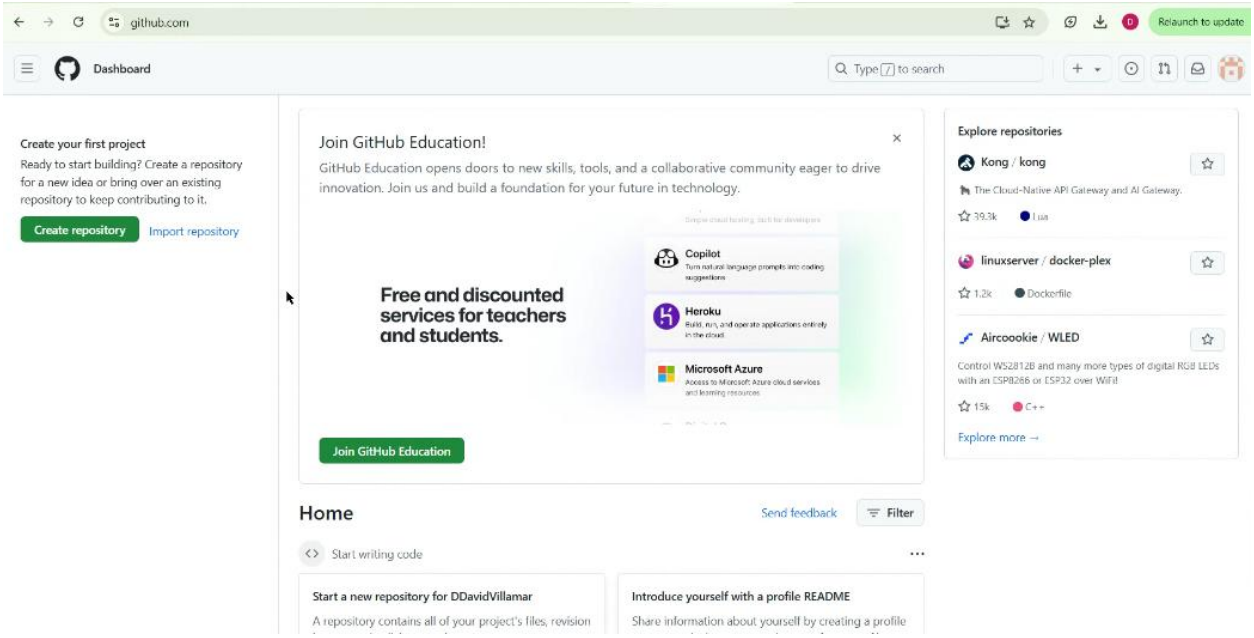
Escoger opciones de acuerdo a tu perfil y Continue



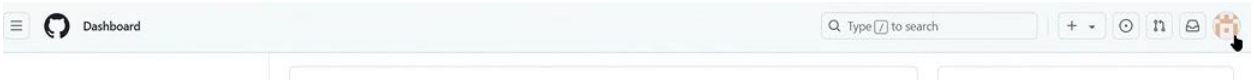
Elige 2 opciones



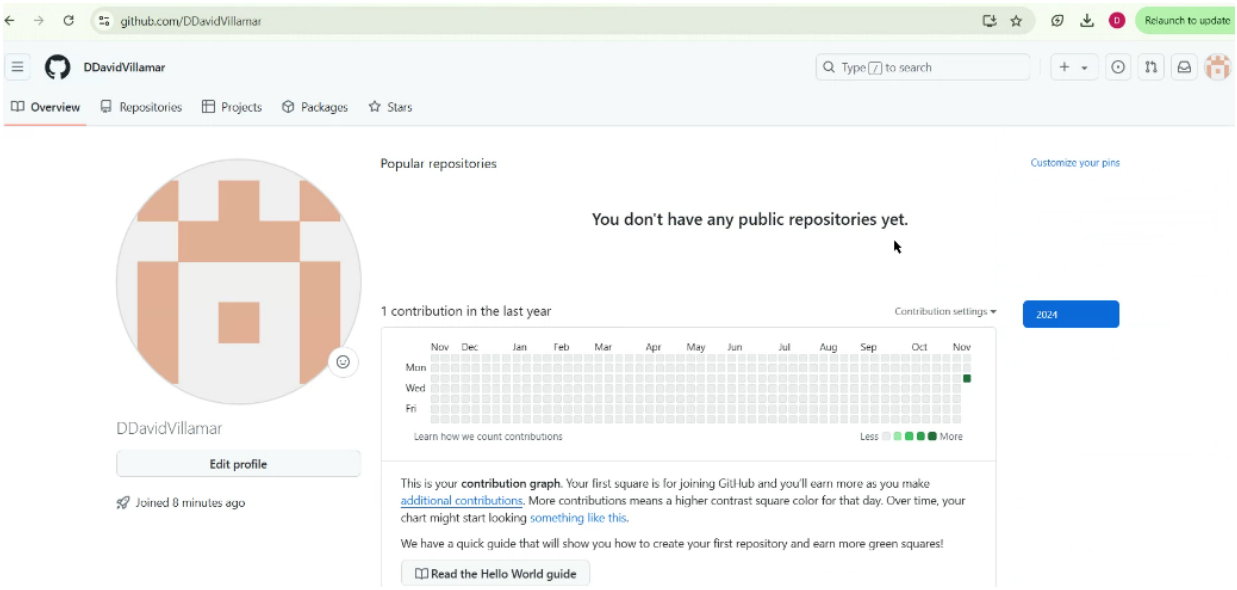
Elige el plan Free y se abre su nueva página de GitHub



Dar click en el ícono del lado derecho para poder abrir tu perfil

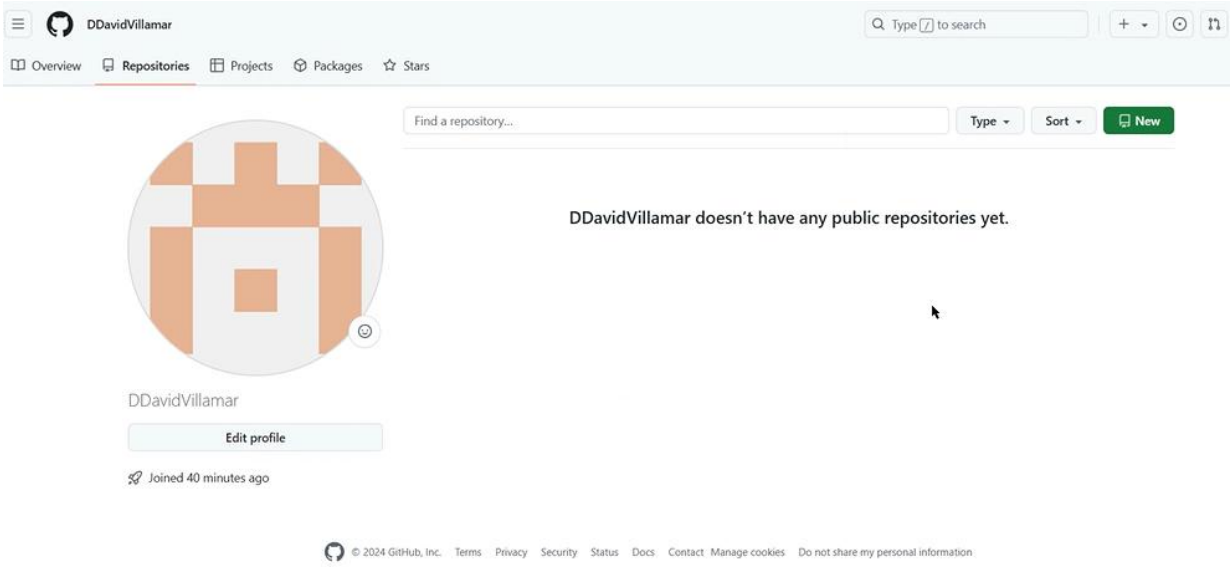


Aquí en Overview puedes personalizar poner una foto y alguna descripción y ver las contribuciones que has hecho el primer cuadrado verde es la fecha de creación de tu GitHub

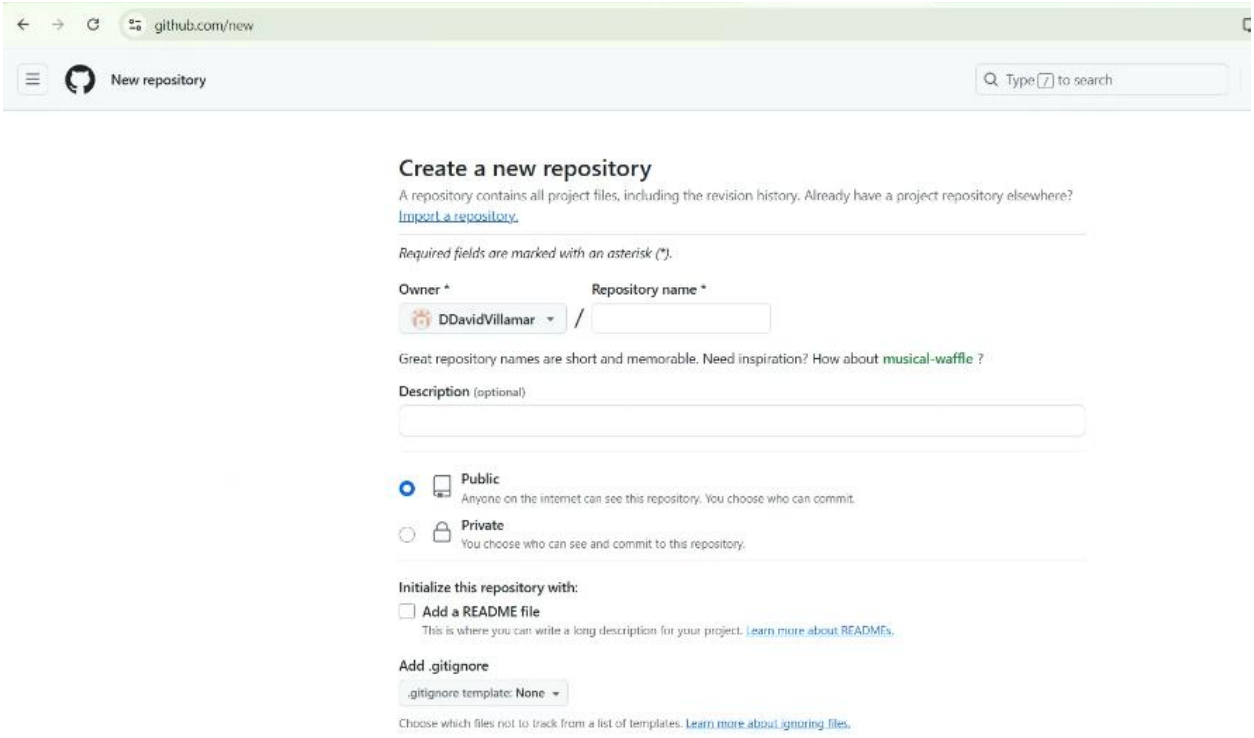


4. Crear un nuevo repositorio

Click en la pestaña repositorios. Luego en el botón verde de la parte superior izquierda dar click en New



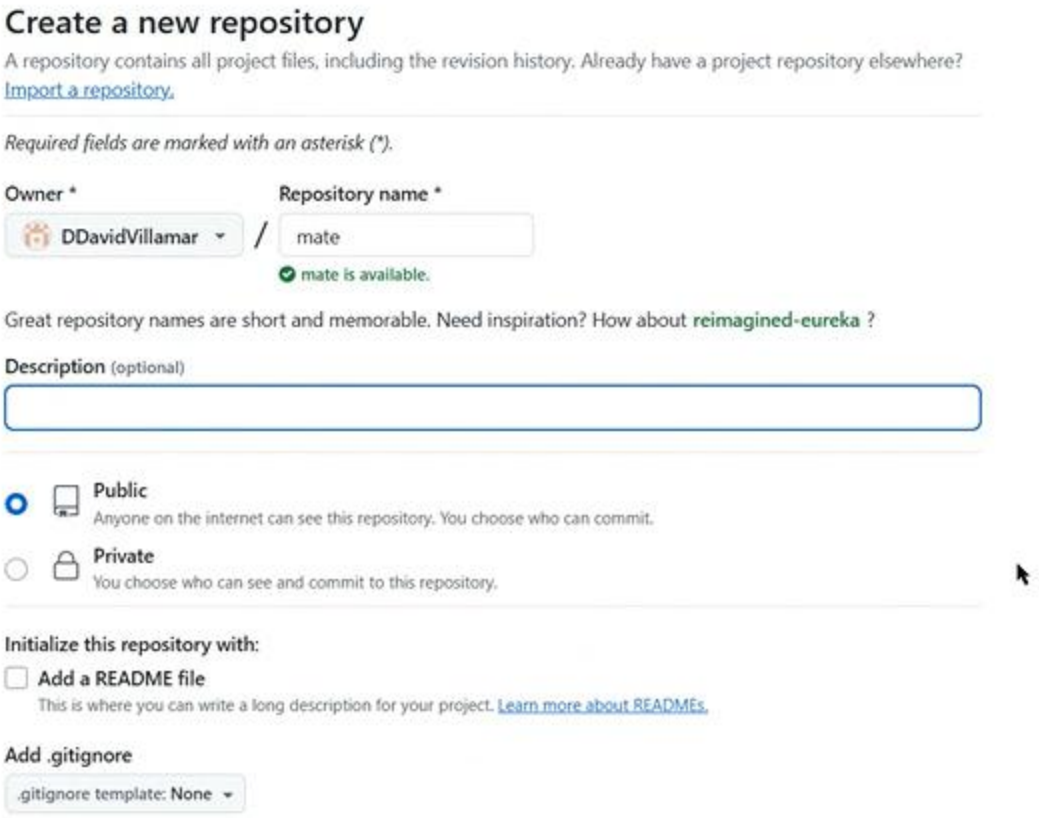
Se abre la página para crear un nuevo repositorio



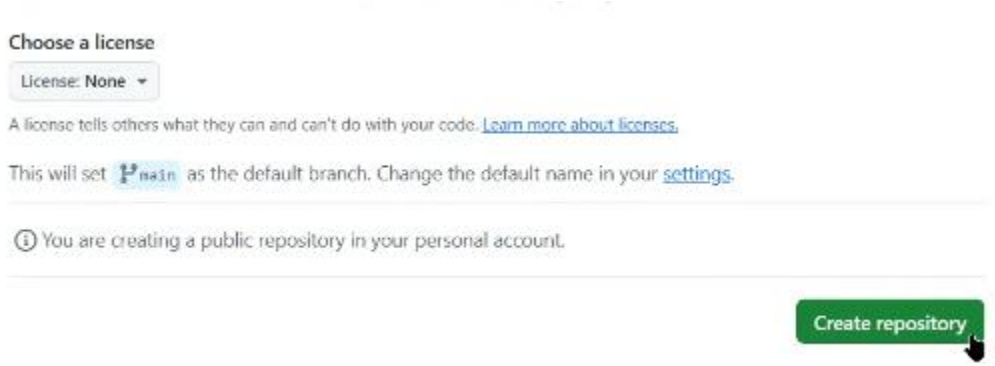
Se debe poner un nombre al repositorio preferible en minúsculas y sin caracteres especiales, sin espacios, elegir público (para que otros usuarios puedan ver) o privado (si solo usuarios con permiso pueden ver el repositorio).

En resumen para crear un repositorio en GitHub:

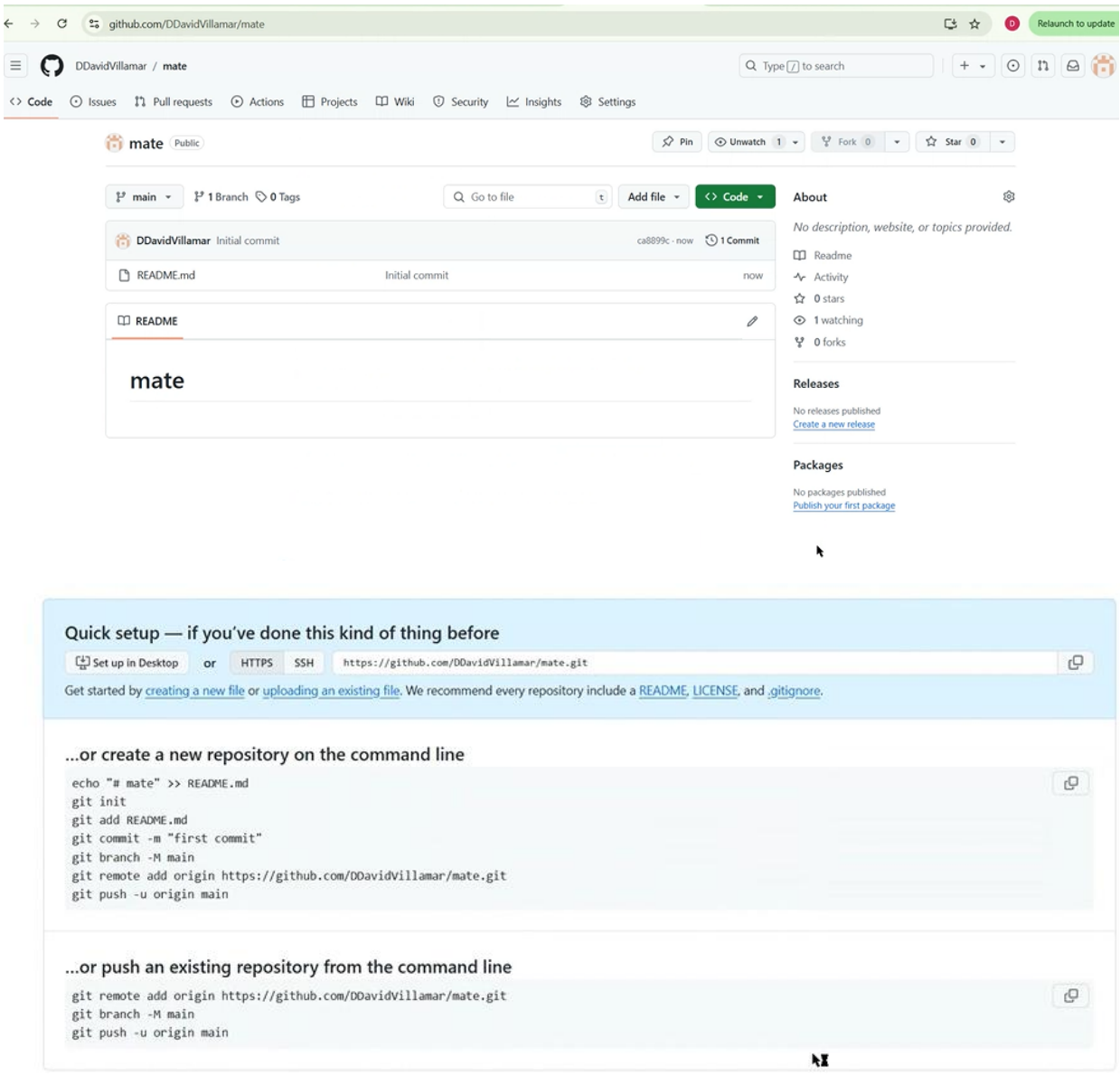
a) Inicia sesión en tu cuenta de GitHub.
b) Haz clic en el botón "New" o "Nuevo repositorio".
c) En "Repository name", escribe el nombre del repositorio por ejemplo “mate”.
d) Añade una descripción si lo deseas.
e) Selecciona la visibilidad del repositorio: Público o Privado.
f) No marques las opciones para inicializar el repositorio con un README, .gitignore o licencia, ya que ya tienes archivos en tu carpeta local.
g) Haz clic en "Create repository".



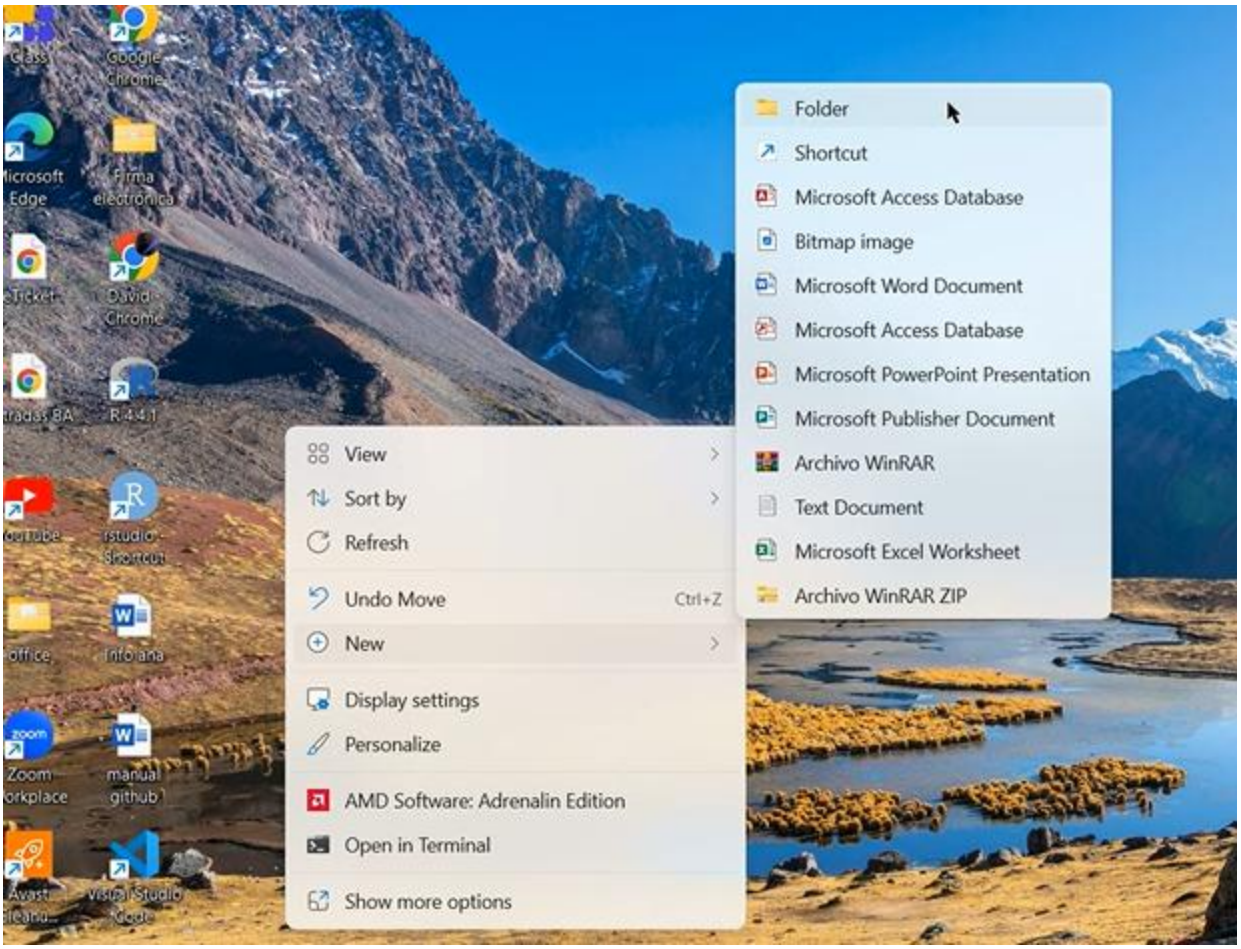
Dar clic en Crear repositorio al final en el ícono verde



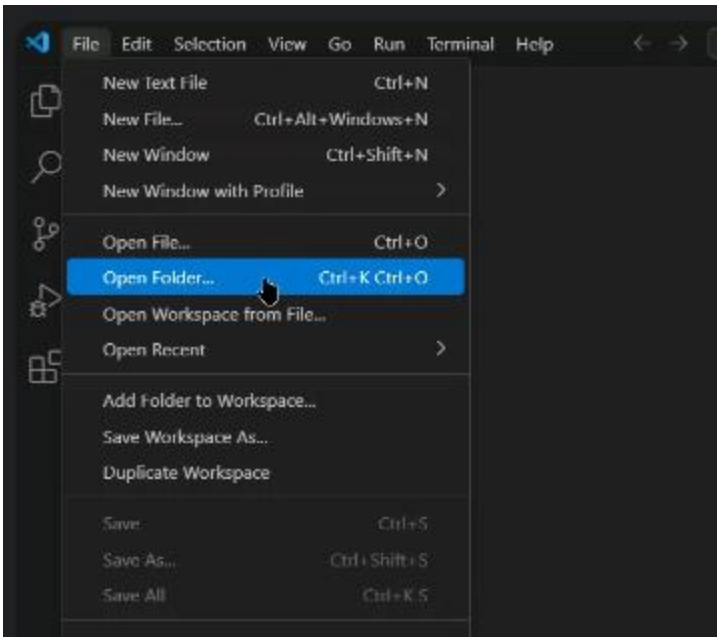
Está creado tu repositorio



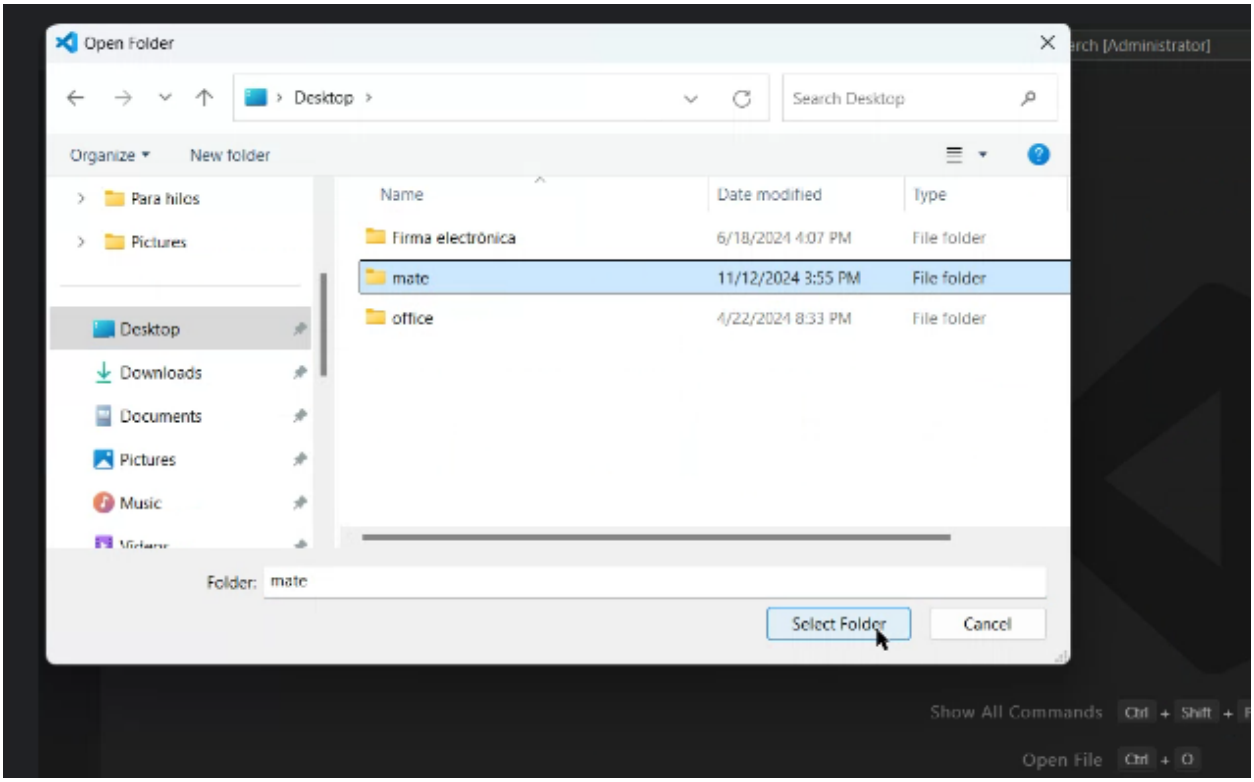
5. **Crear un directorio de trabajo en tu PC local, por ejemplo, en el escritorio o en algún directorio de tu computadora**



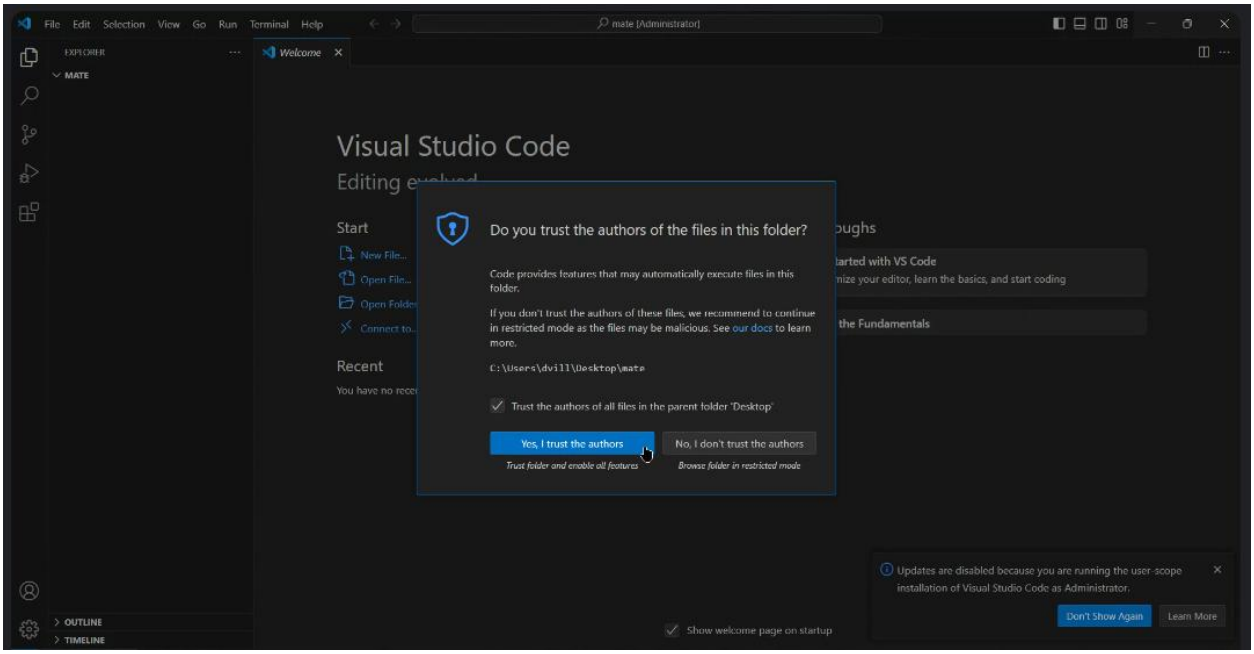
Abrir desde VSC el directorio creado



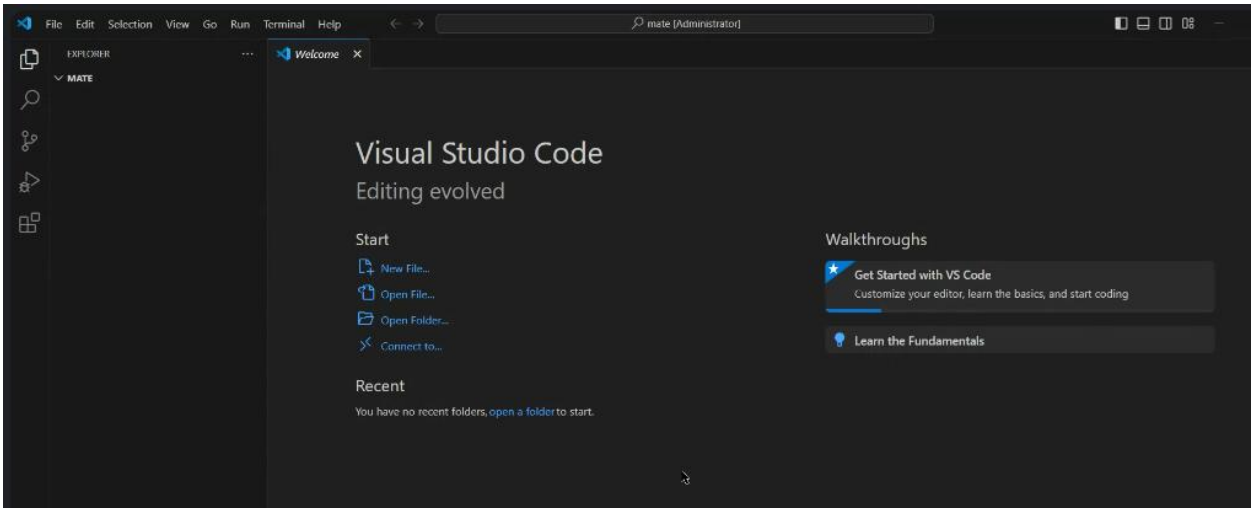
Seleccionas el Folder donde vamos a trabajar



Poner las opciones de confiar en los autores



Ya estás en el directorio de trabajo creado



Verificar que tener instalado Python y Anaconda Y también que se haya creado el entorno para trabajar

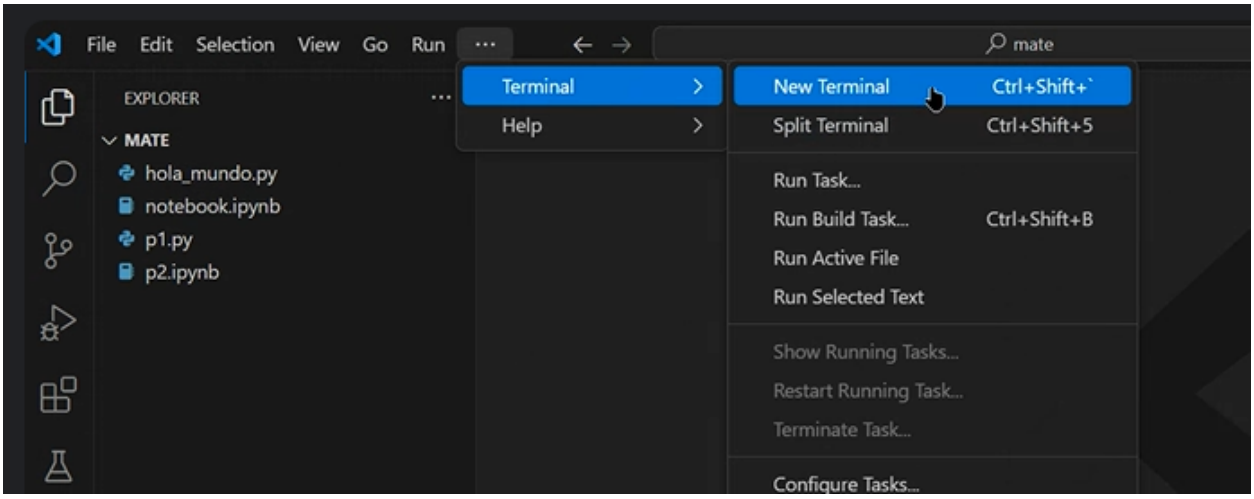
(Ver guía 1)

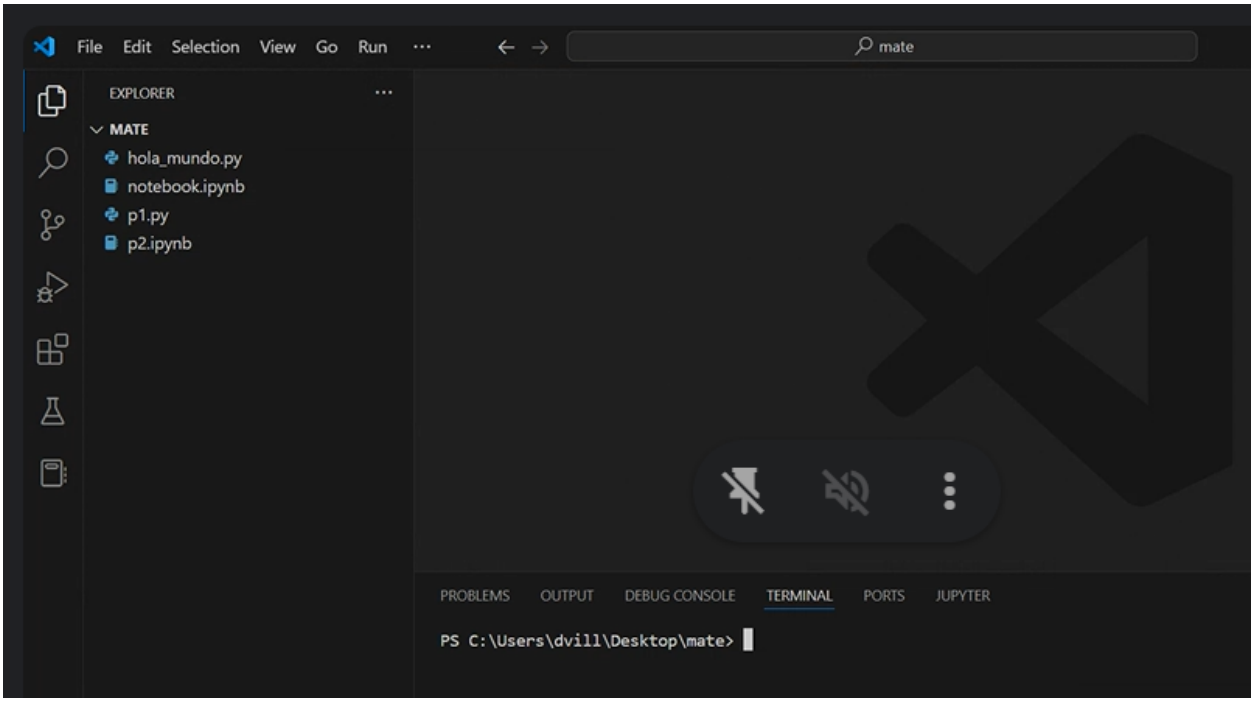
https://www.python.org/downloads/

https://anaconda.org/

1. **Subir archivos de tu directorio local a tu repositorio en GitHub**

En VSC abre una nueva terminal





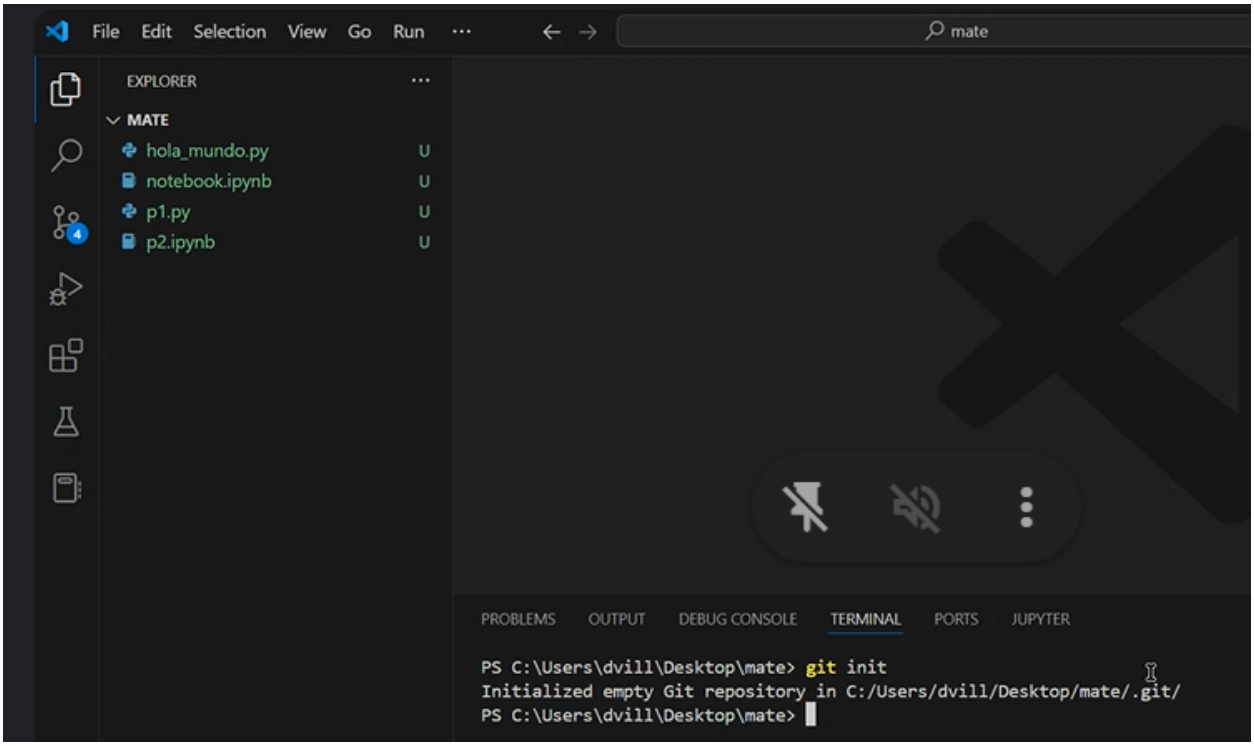
Inicializar el repositorio local

- Navega hasta la carpeta de tu proyecto en la terminal.
- Ejecuta el siguiente comando para inicializar un repositorio Git en esa carpeta:

git init

Incializar el repositorio local con

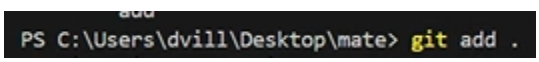
git init

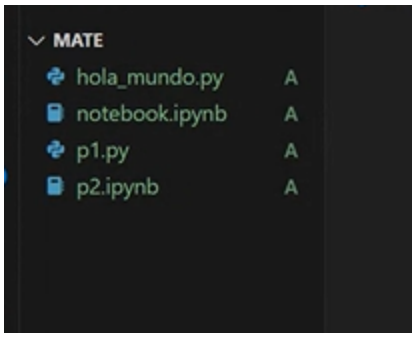


Agregar archivos al repositorio

- Puedes agregar archivos individuales usando el comando git add nombre_archivo.
- Para agregar todos los archivos modificados, utiliza el comando:

git add .





Vamos a configurar el repositorio en la terminal

echo "# mate" >> README.md

git init

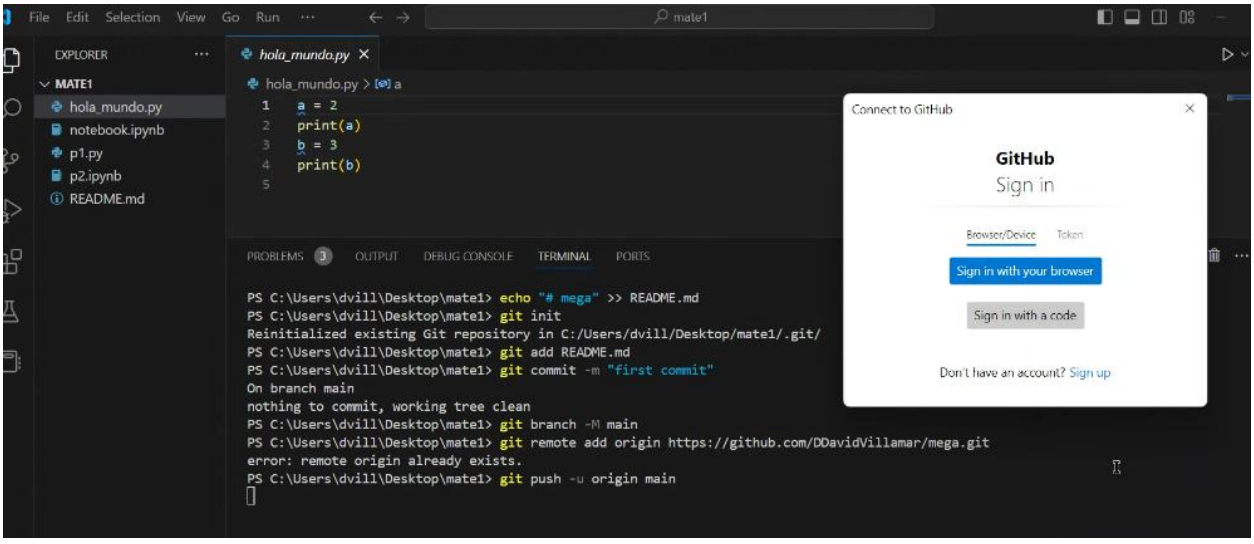
git add README.md

git commit -m "first commit"

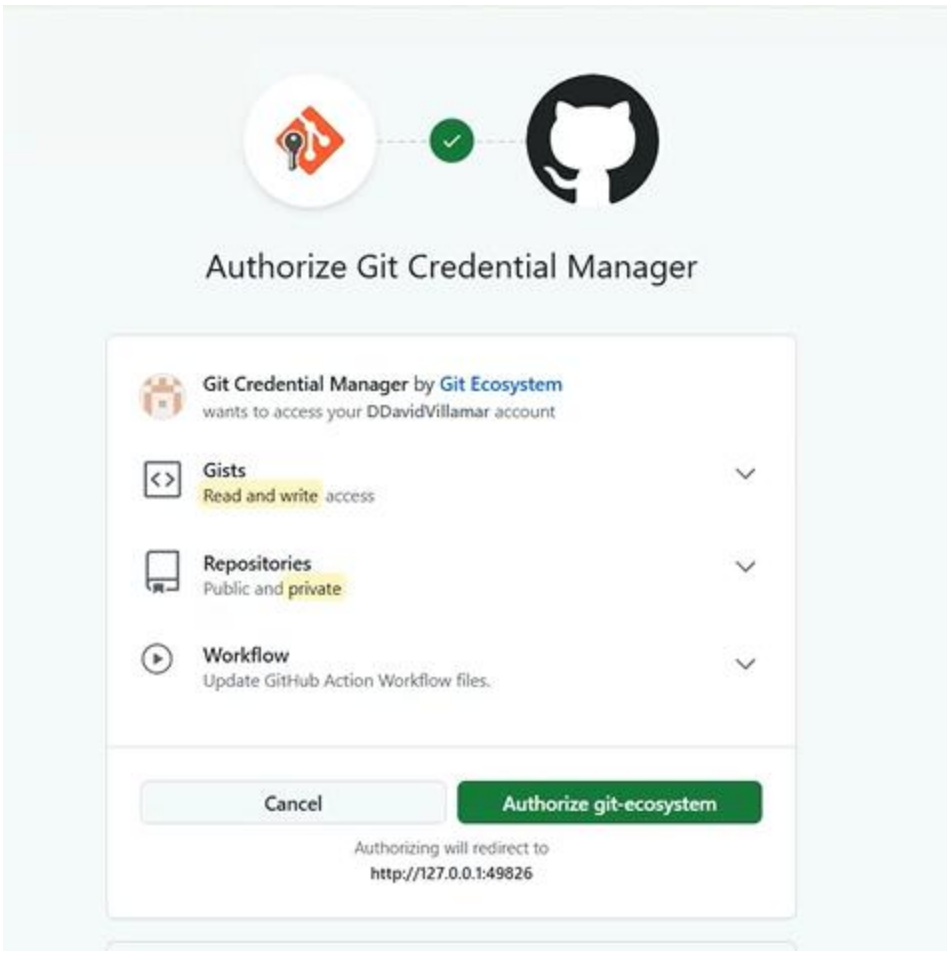
git branch -M main

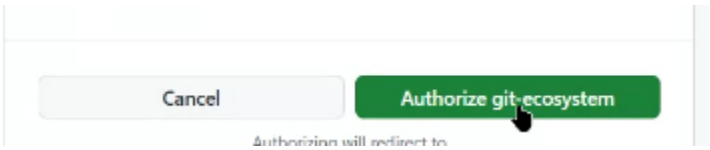
git remote add origin https://github.com/DDavidVillamar/mate.git

git push -u origin main



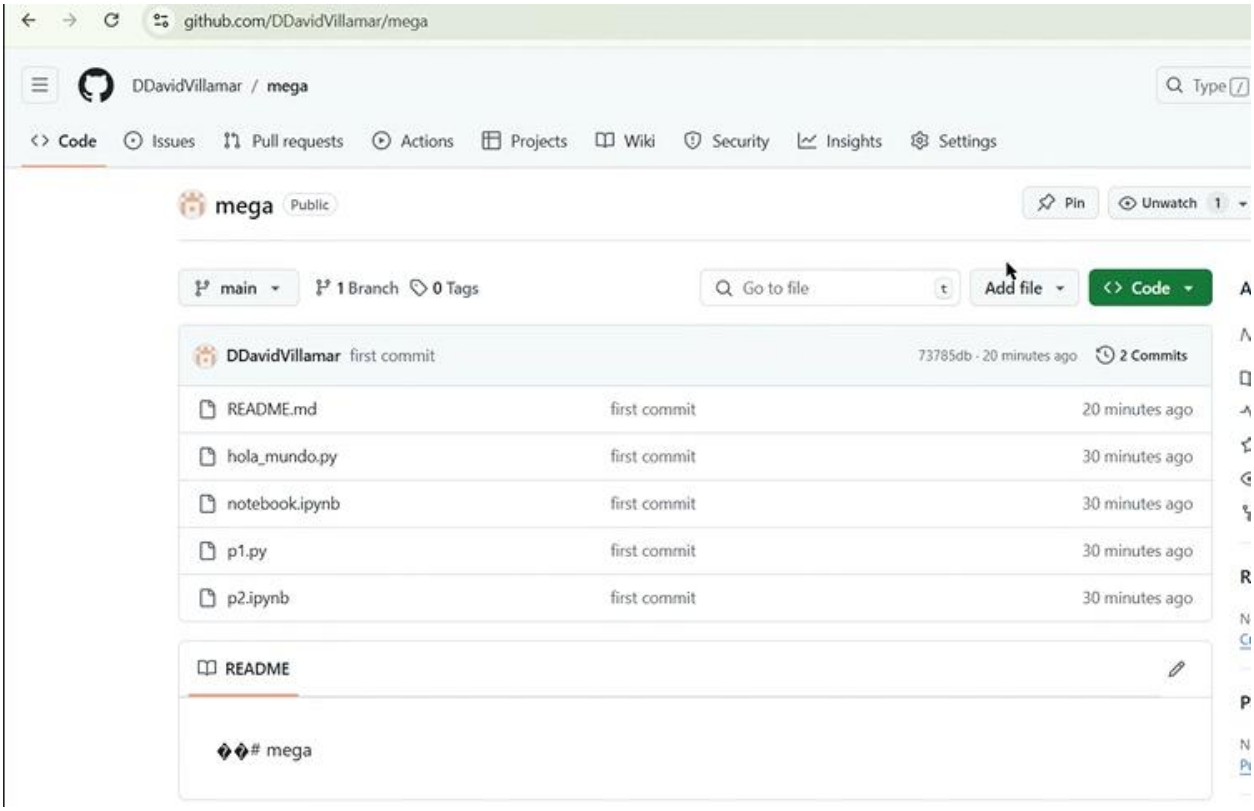
Te tienes que autentificar con tu clave de GitHub





Dar click en Authorize y poner credenciales de GitHub.

Ahora los archivos están respaldados en tu página de GitHub



Otros comandos útiles para GitHub en terminal

Verificar el remoto en el que me encuentro

git remote -v

Eliminar un remoto existente y agregar uno nuevo

 # Eliminar el remoto actual

git remote remove origin

 # Agregar el nuevo remoto

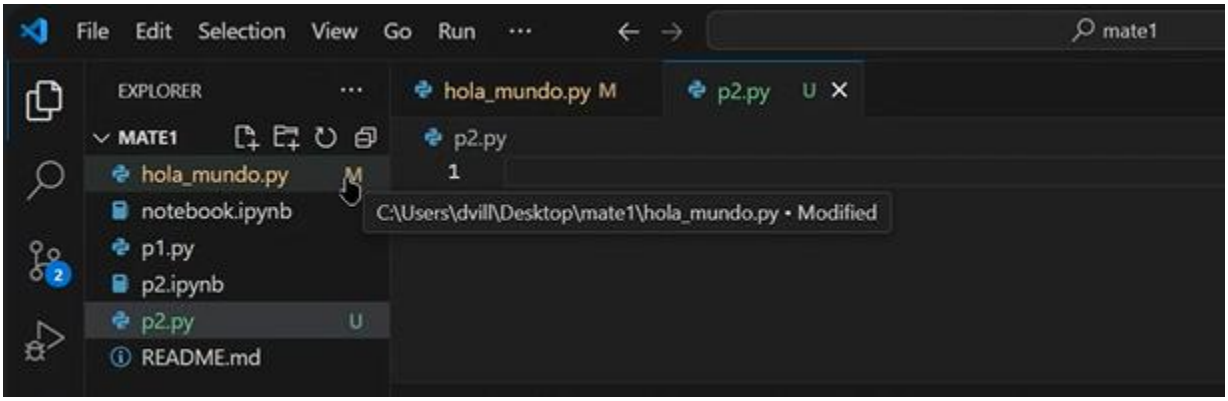
git remote add origin https://github.com/DDavidVillamar/mega.git

##Modificar la URL del remoto existente

 # Cambiar la URL del remoto existente

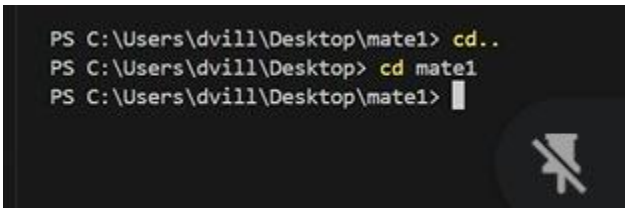
git remote set-url origin https://github.com/DDavidVillamar/mega.git

Si quieres subir cambios realizados en tu archivo local los puedes volver a cargar en GitHub

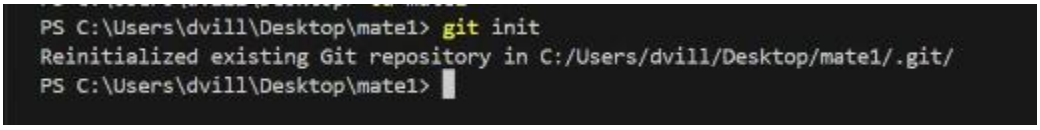


Abrir una nueva terminal

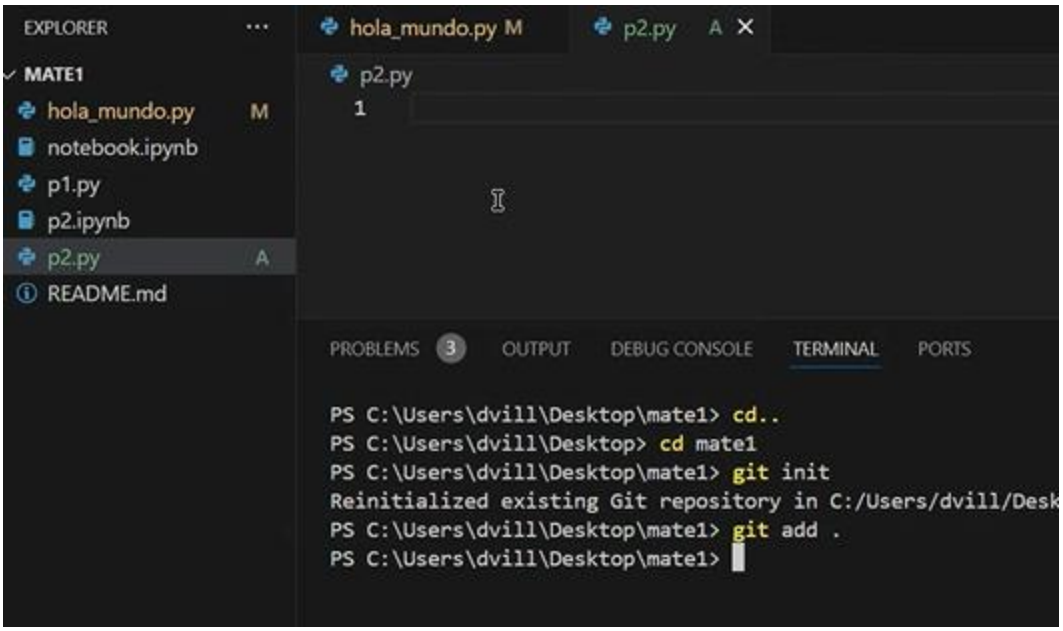
Ubicarse en el directorio adecuado con el comando cd (cd..)



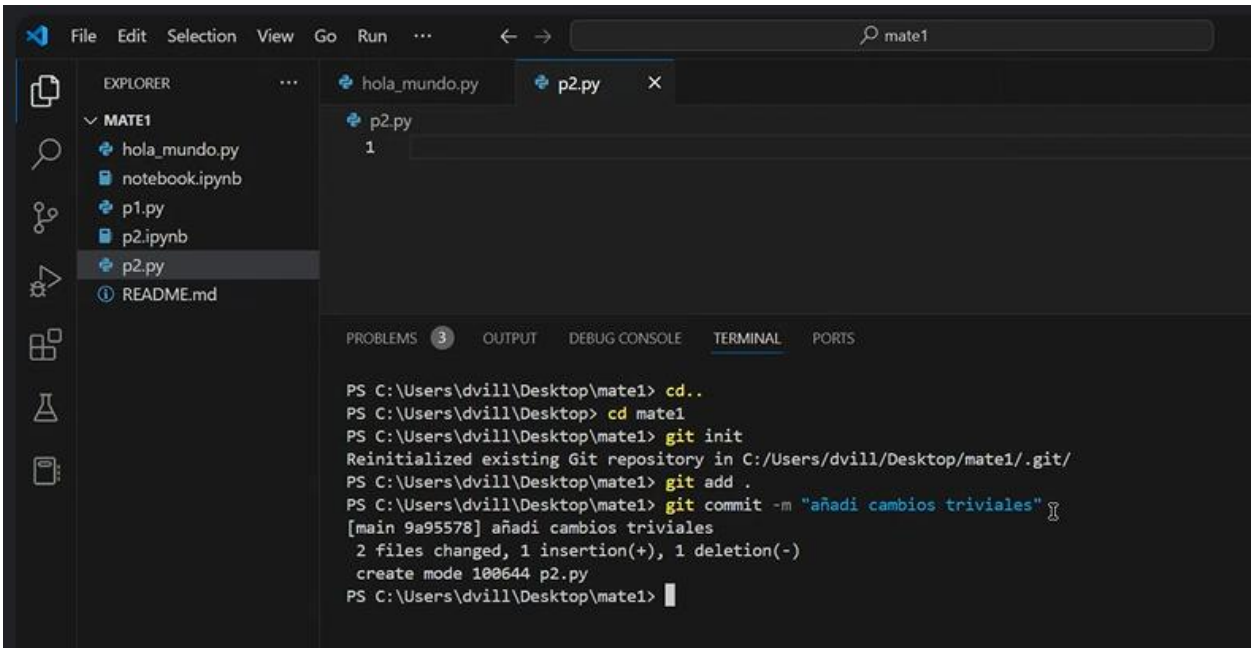
git init



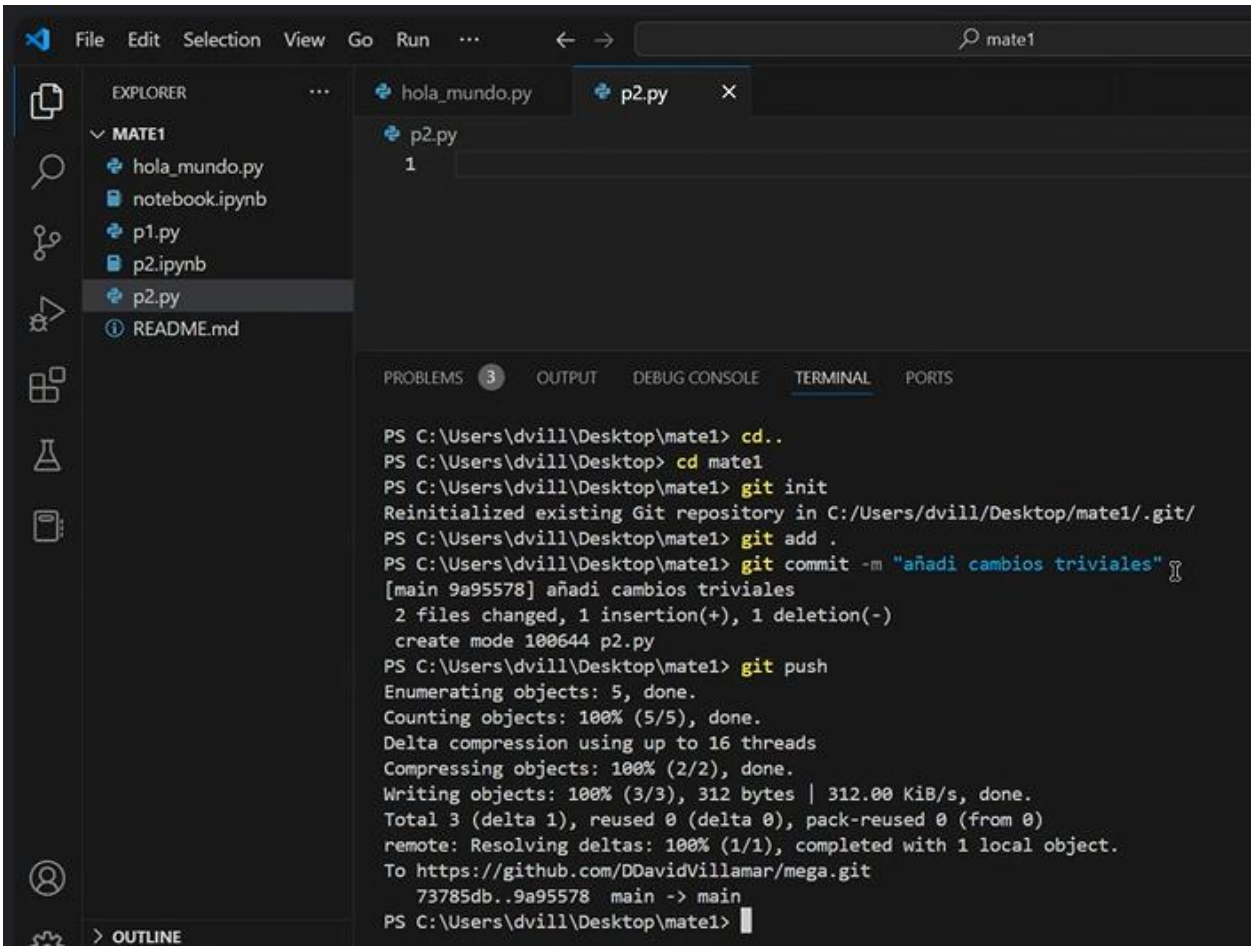
git add .



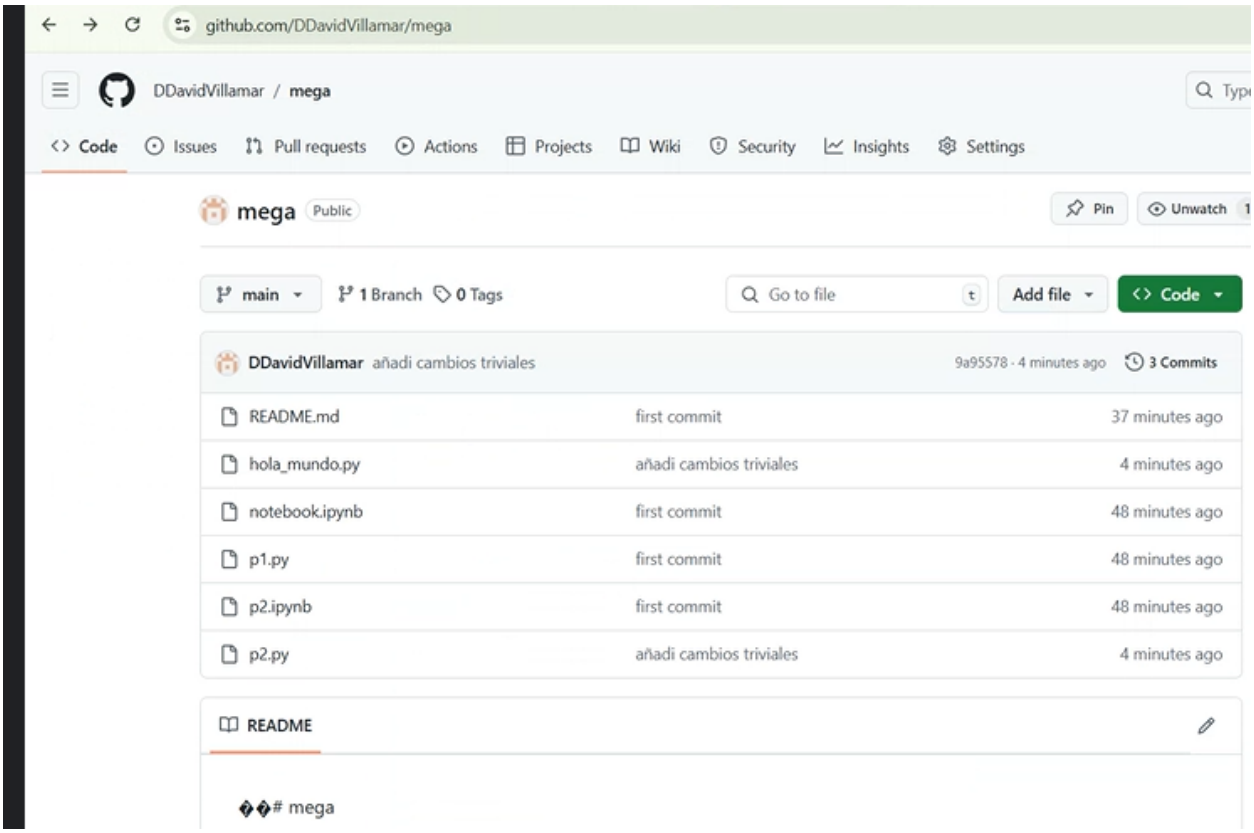
git commit -m “añadí cambios triviales”



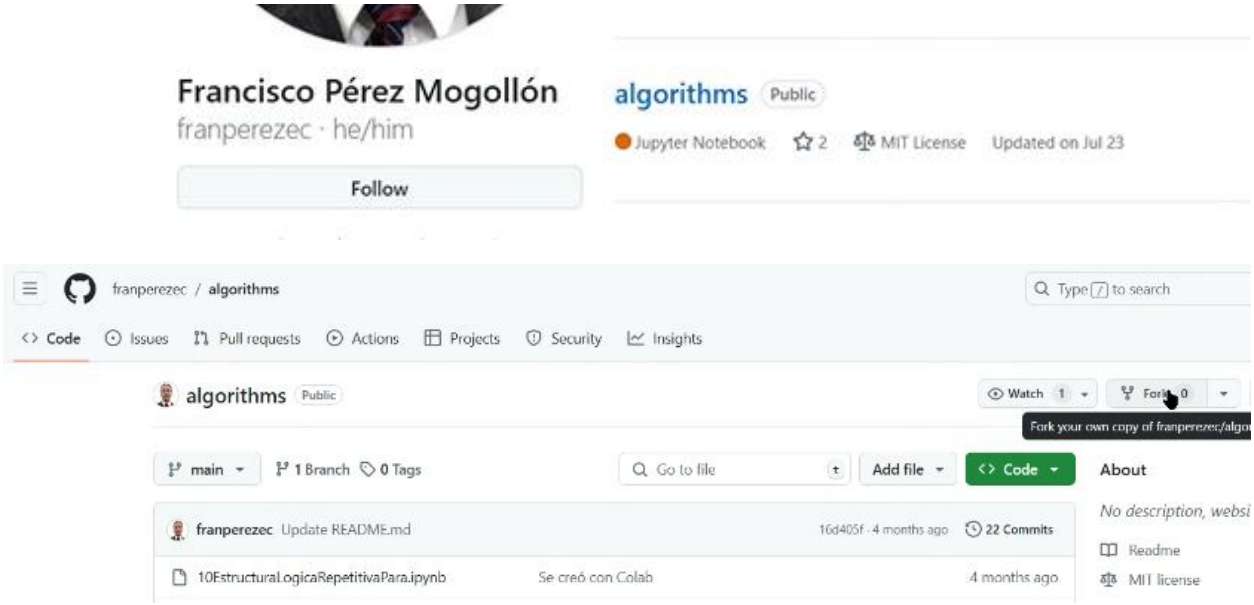
git push

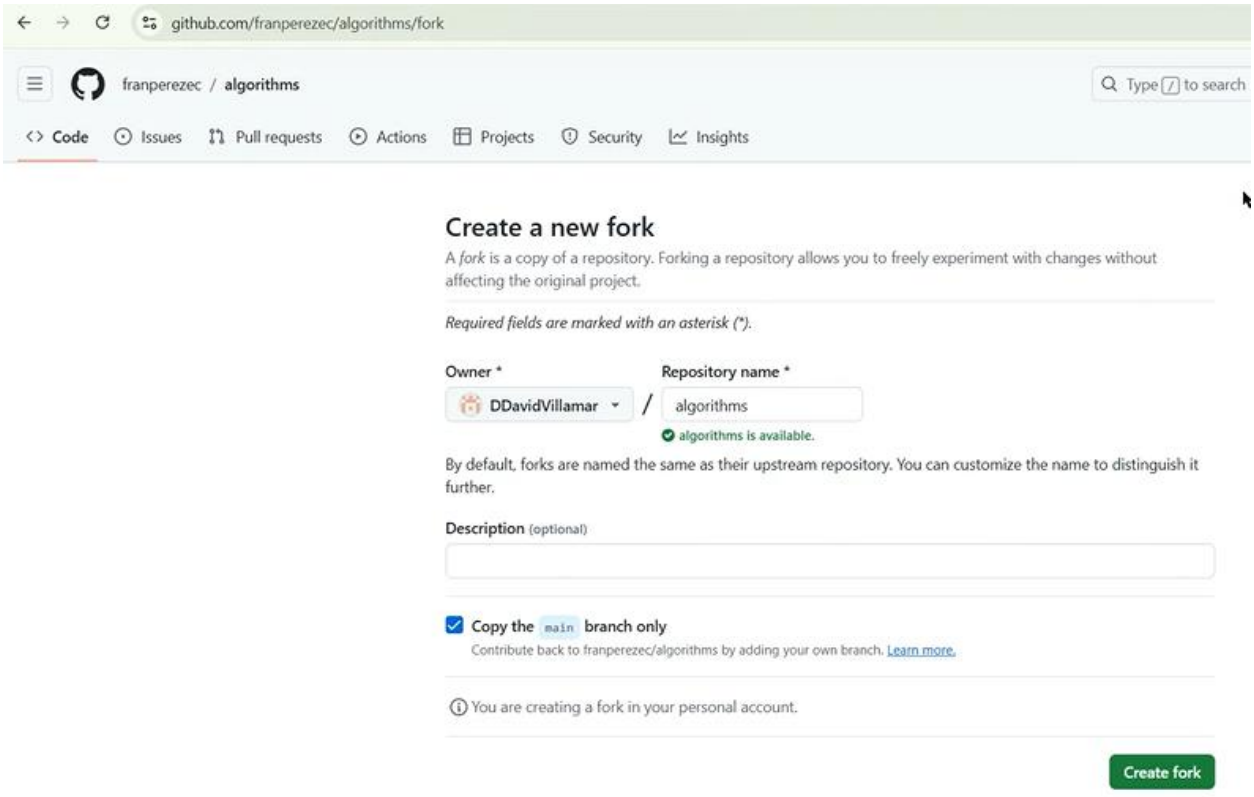


Ahora puedes ver los nuevos archivos en tu GitHub

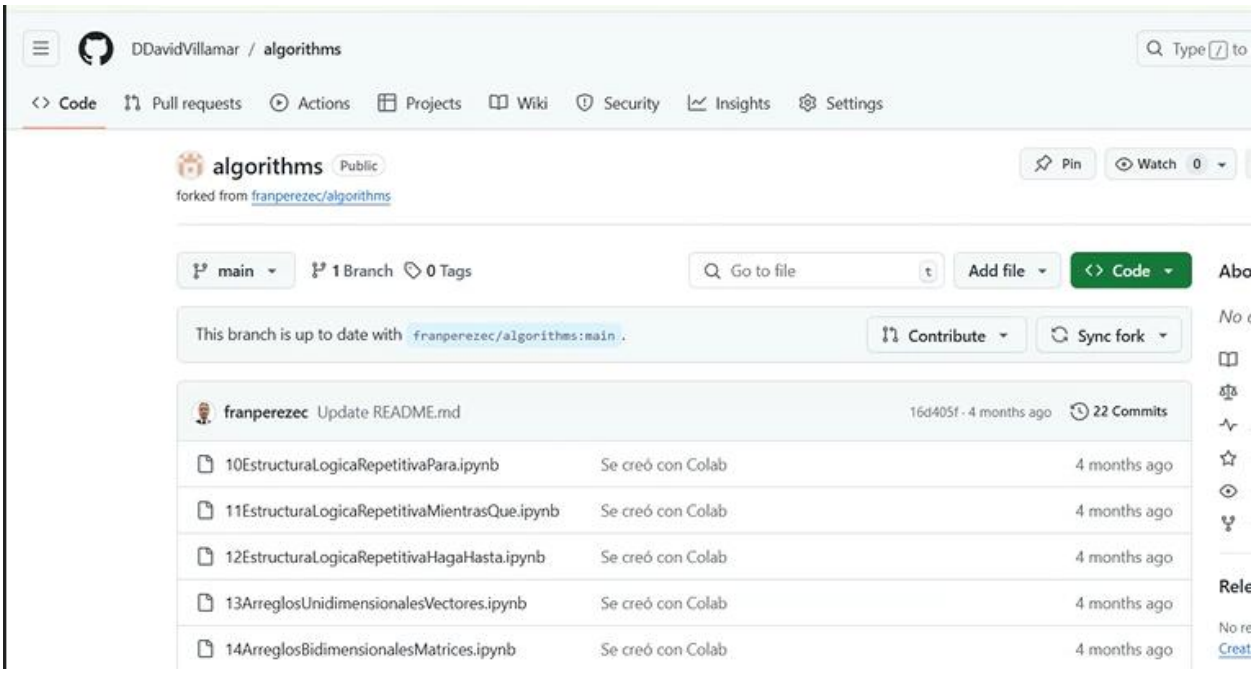


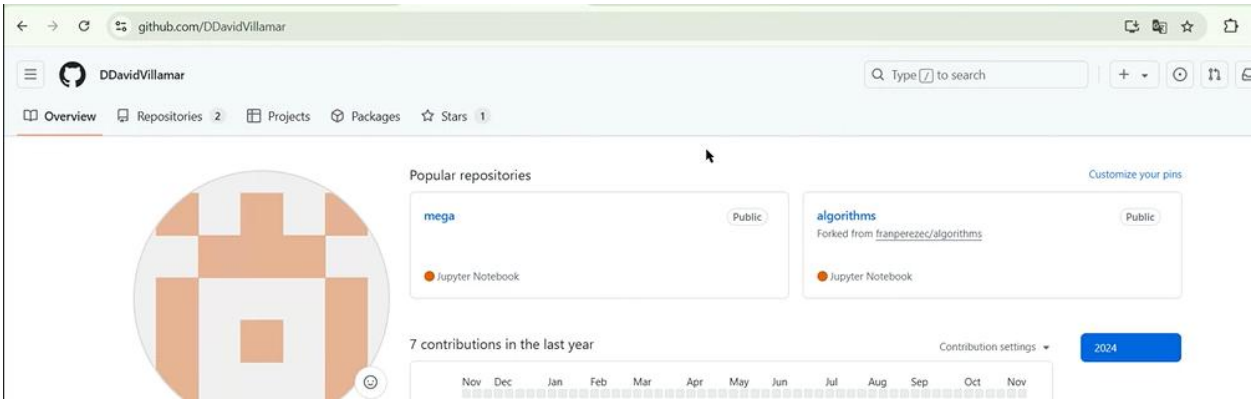
Hacer una copia de un repositorio de terceros





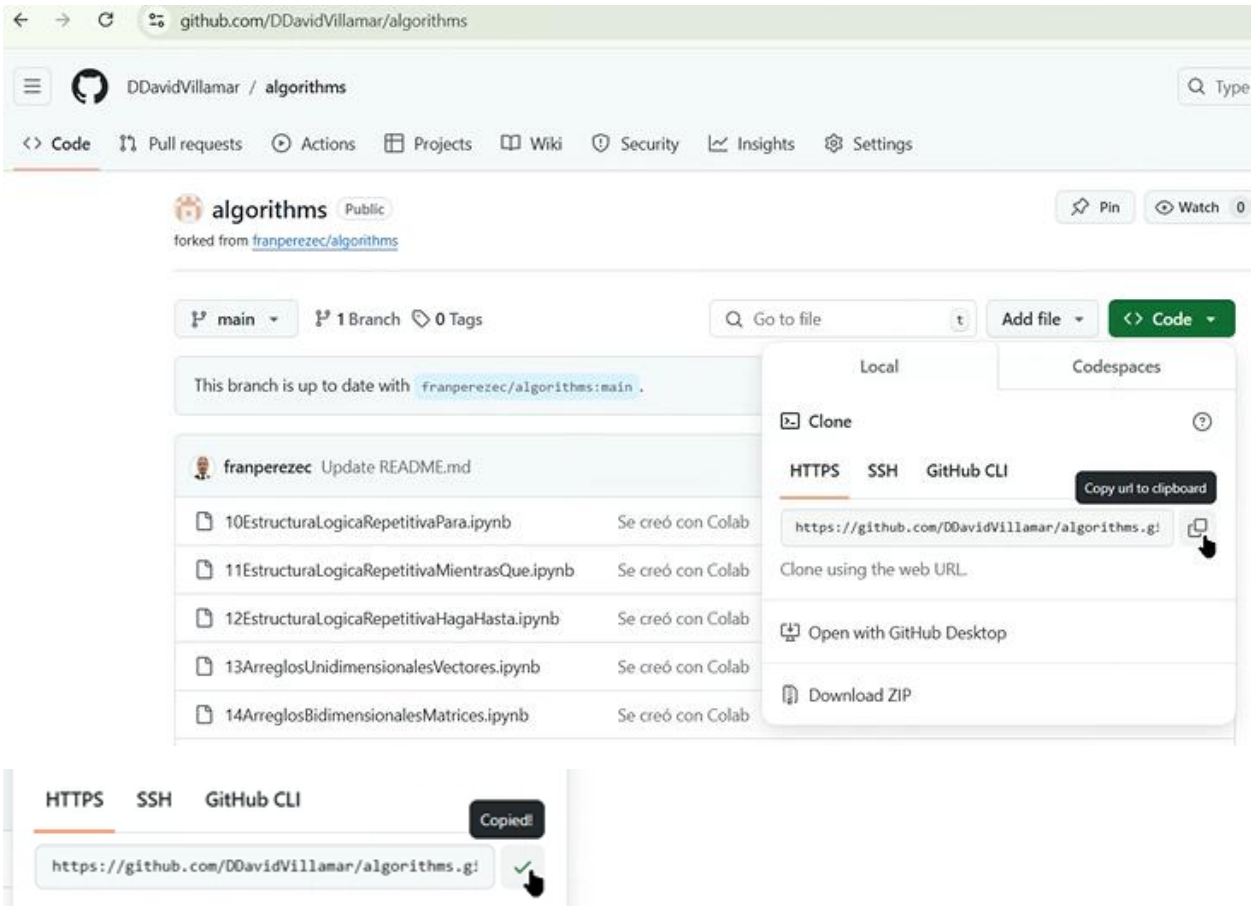
Se crea una copia del repositorio en mi perfil de GitHub



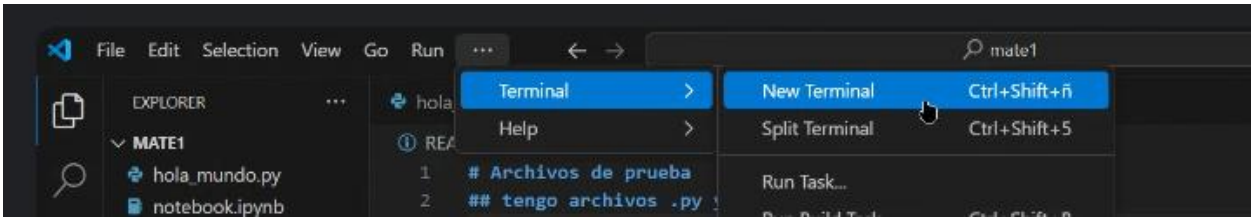


Pasos para clonar y trabajar con un repositorio del que has hecho Fork

Primero, ve a tu GitHub y copia la URL del repositorio fork (el que está en tu cuenta). El botón "Code" te dará la URL.

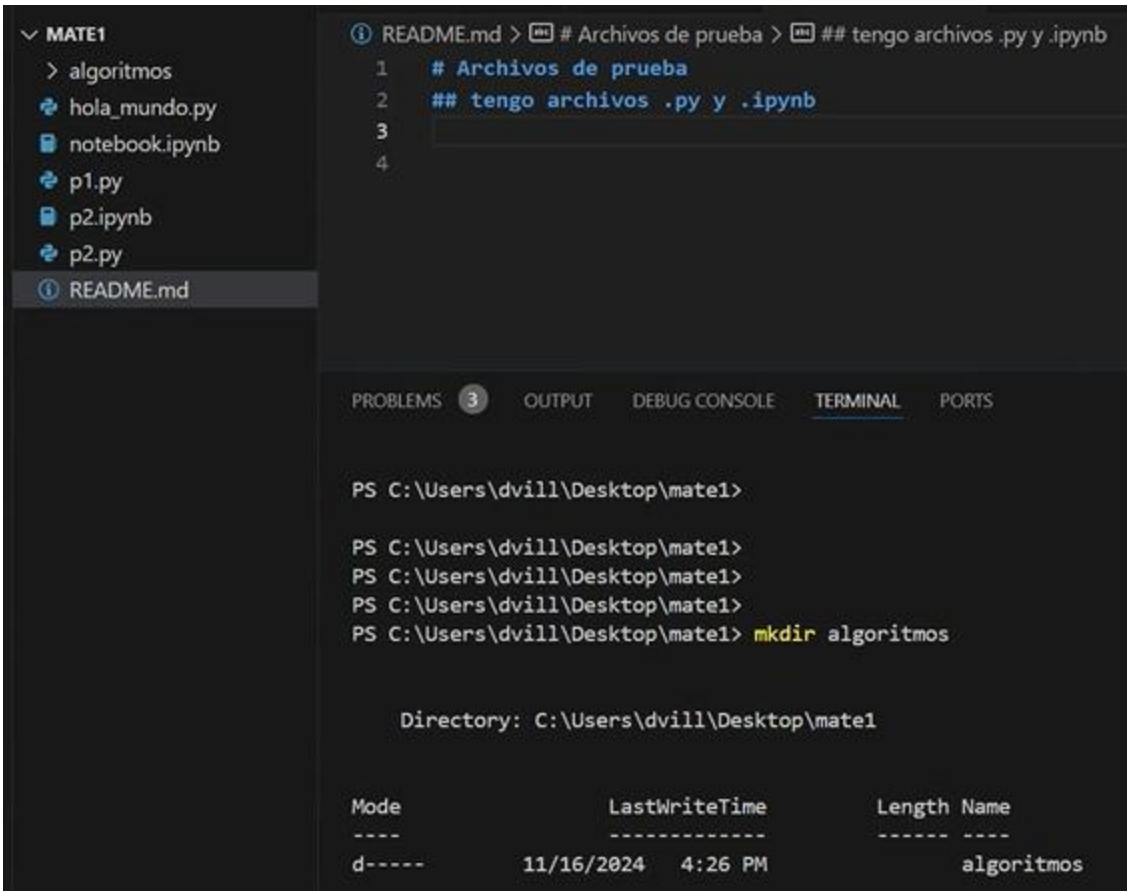


Abre la terminal en la carpeta donde quieres tener el proyecto y ejecuta:



 # Primero creas la carpeta

mkdir algoritmos

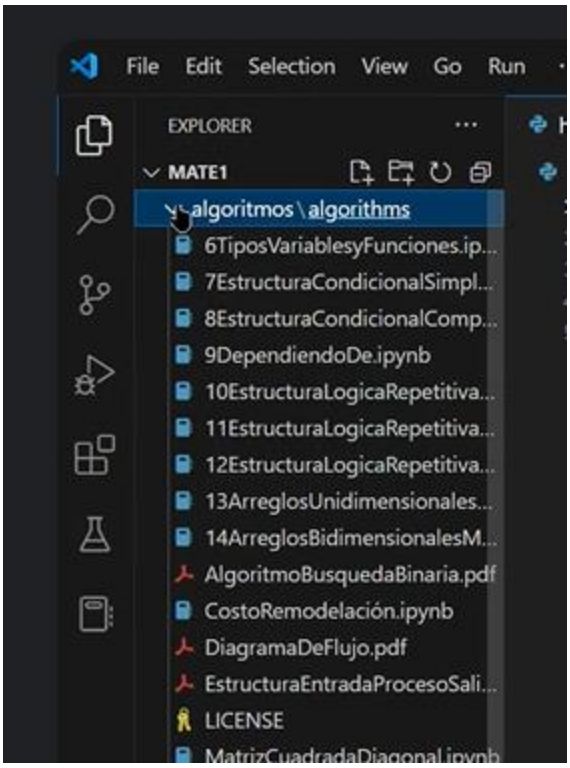


 # Entras a ella cd algoritmos

 # Y aquí ya puedes clonar tu repositorio

git clone https://github.com/tu-usuario/tu-repositorio.git

```
PS C:\Users\dvill\Desktop\mate1\algoritmos> git clone https://github.com/DDavidVillamar/algorithms.git
Cloning into 'algorithms'...
remote: Enumerating objects: 67, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 67 (delta 42), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (67/67), 1.77 MiB | 1.81 MiB/s, done.
Resolving deltas: 100% (42/42), done.
PS C:\Users\dvill\Desktop\mate1\algoritmos>
```



Comandos útiles para terminal

Ver en qué carpeta estás actualmente

pwd

Ver el contenido de la carpeta actual

dir # (en Windows)

ls # (en Linux/Mac)

Regresar a la carpeta anterior

cd ..

Ir a una ruta específica

cd C:\Users\nombre-usuario\Desktop\nombre-carpeta

Crear varias carpetas anidadas de una vez

mkdir carpeta1/carpeta2/carpeta3

elminar una carpeta

En Windows

rmdir nombre-carpeta # (si está vacía)

```
rmdir /s nombre-carpeta # (si tiene contenido)

# En Linux/Mac

rm -r nombre-carpeta
```