WEEK 2 SOLUTION.

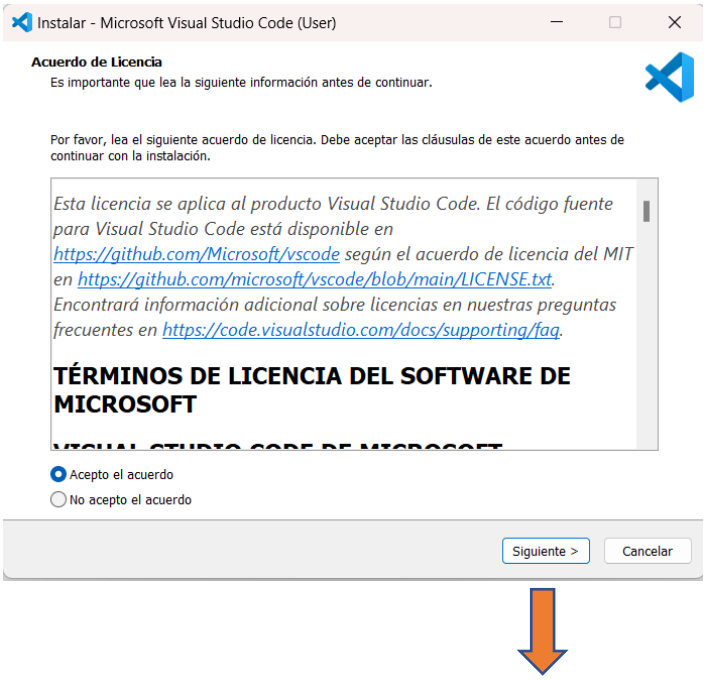
Student: Emanuel Nguema Oyono Avoro

1. Select Your Operating System (OS): Choose an operating system that best suits your preferences and project requirements. Download and Install Windows 11. https://www.microsoft.com/software-download/windows11

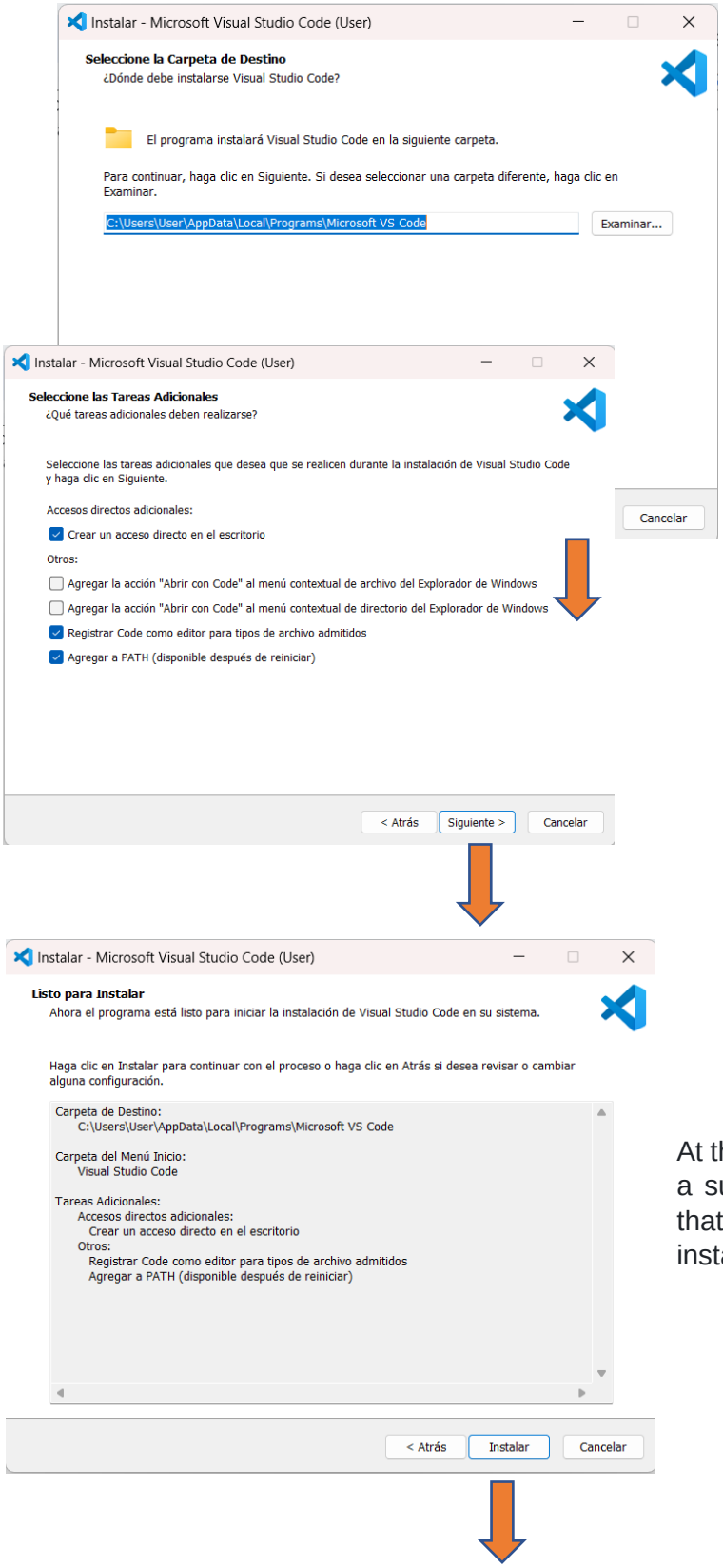
In my case I will use Windows 10.

2. Install a Text Editor or Integrated Development Environment (IDE): Select and install a text editor or IDE suitable for your programming languages and workflow. Download and Install Visual Studio Code. https://code.visualstudio.com/Download

To install VScode code, we go to the link provided in this task for download. After downloading we proceed with the installation as indicated in the official documentation:

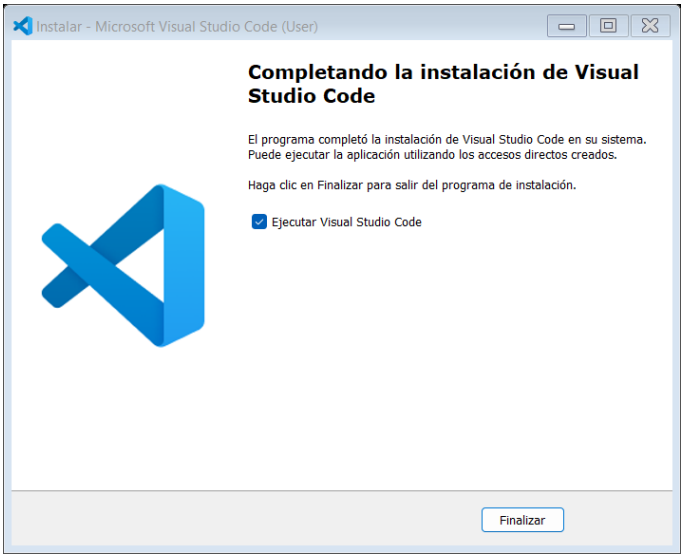


We double click on the installer and accept the license agreement.



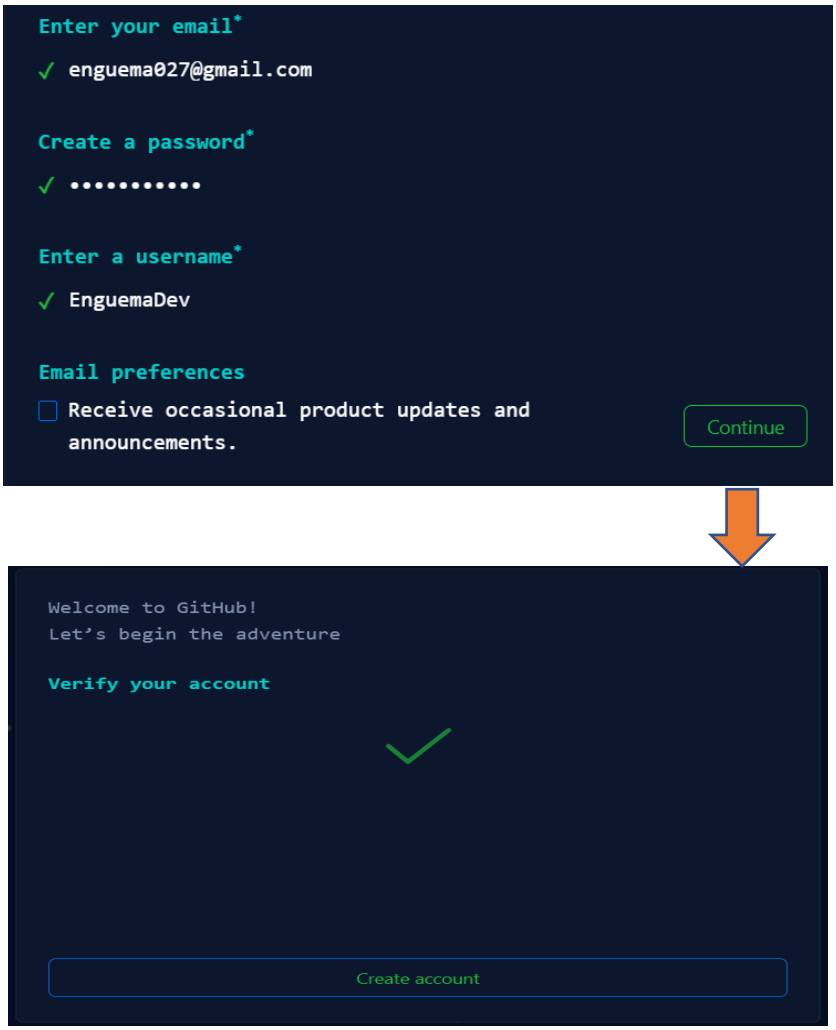
We indicate the installation destination folder. In our case it will be the one that comes by default and we give the following.

At the end of everything, it will show us a summary of the installation options that we have selected and we click install.



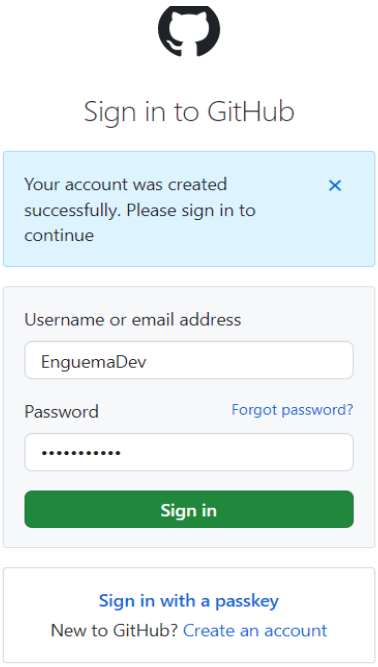
We finish with the installation.

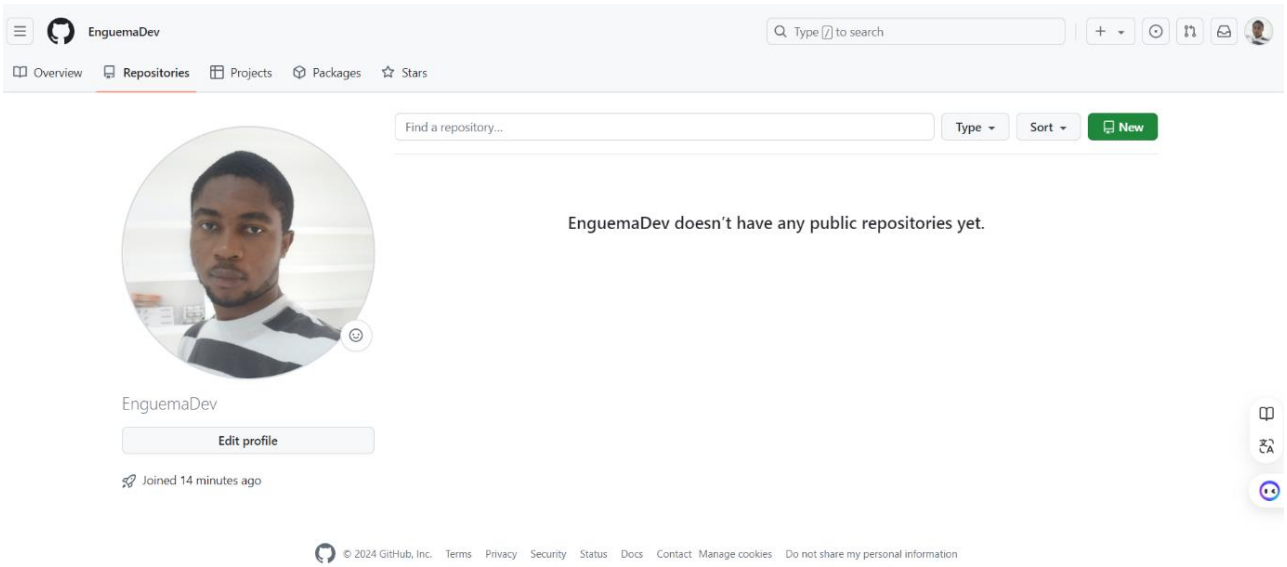
3. Set Up Version Control System: Install Git and configure it on your local machine. Create a GitHub account for hosting your repositories. Initialize a Git repository for your project and make your first commit. https://github.com



We will start by creating an account on github. To do this, we access the link provided, go to the register tab and fill out the registration form as requested.

After successfully completing the account creation process, we finally access my new Github account.





I proceed with the creation of my first repository.

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 * / Repository name *

EnguemaDev / PLP-My-First-Repository

PLP-My-First-Repository is available.

Great repository names are short and memorable. Need inspiration? How about urban-computing-machine ?

Description (optional)

Mi repositorio para PLP Academy

- 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.

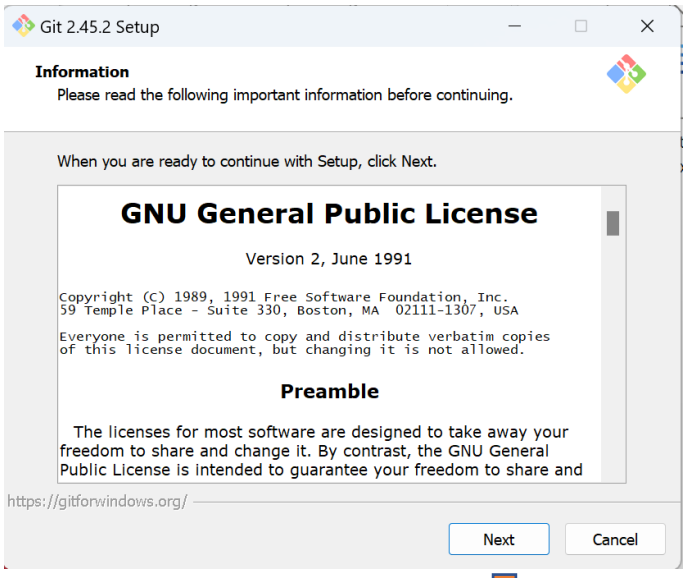
Add .gitignore

.gitignore template: None

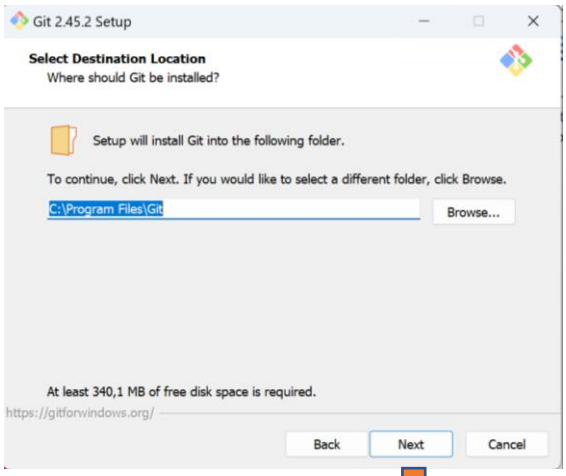
Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

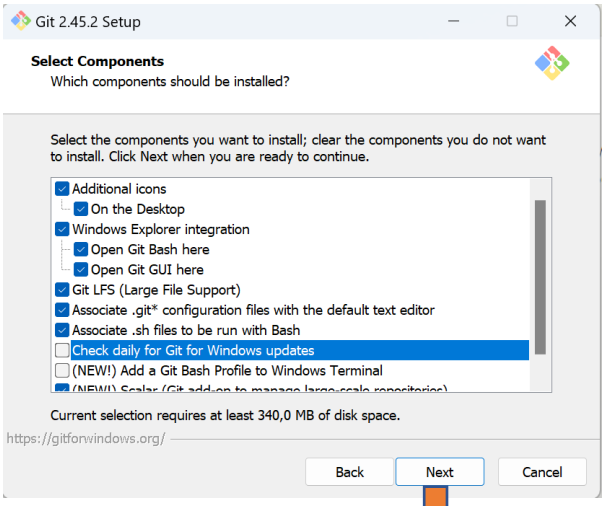
Next we proceed with the installation of Git.



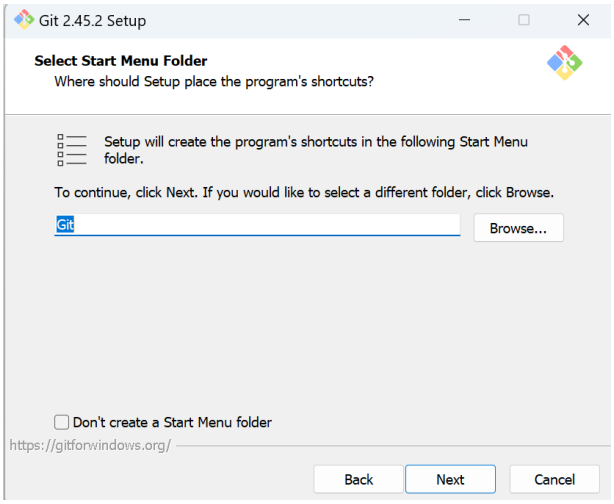
To do this, we double click on the downloaded installer.

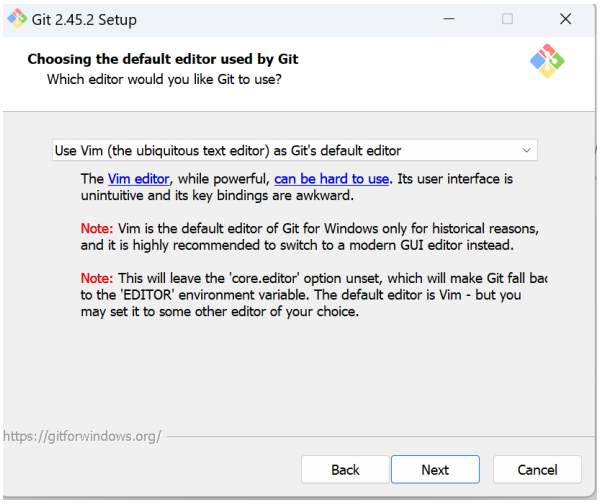


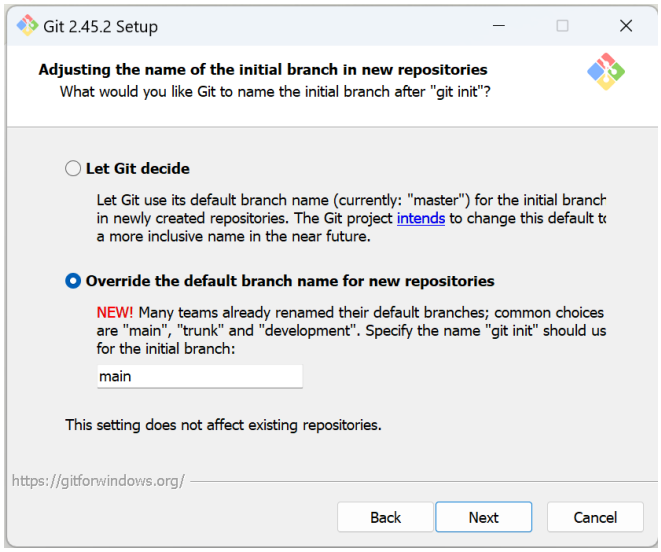
We leave the installation folder by default.

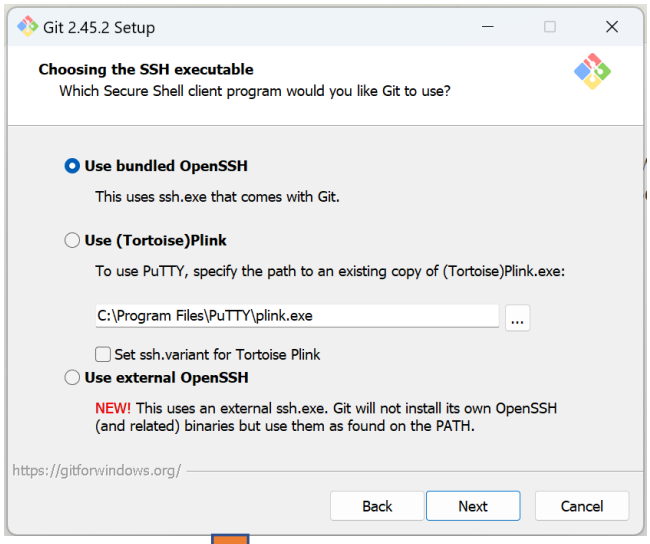


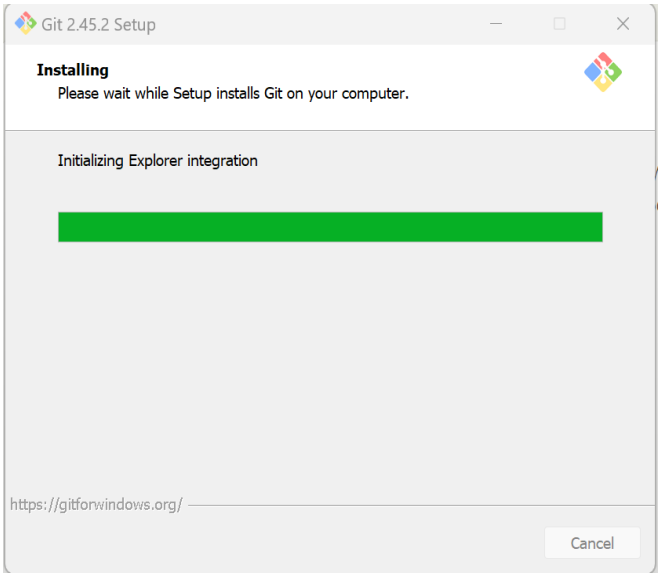
We select the installation options that fit our needs

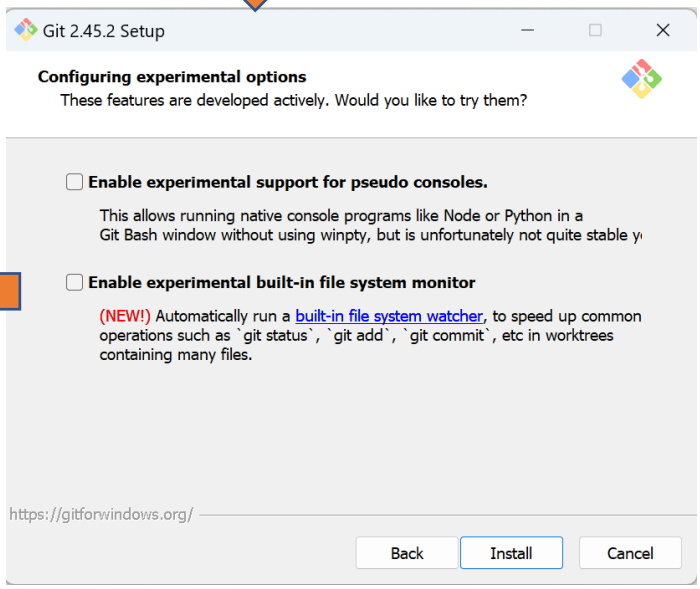


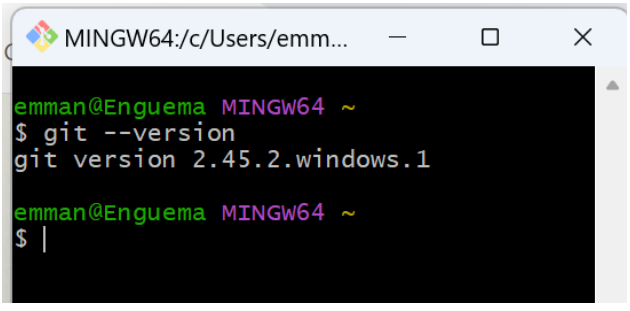



Git 2.45.2 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch
in newly created repositories. The Git project intends to change this default t
a more inclusive name in the near future.
Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices
are "main", "trunk" and "development". Specify the name "git init" should us
for the initial branch:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.45.2 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 (Tortoise)Plink
To use PuTTY, specify the path to an existing copy of (Tortoise)Plink.exe:
C:\Program Files\PuTTY\plink.exe
Set ssh.variant for Tortoise Plink
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.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.45.2 Setup
Installing
Please wait while Setup installs Git on your computer.
Initializing Explorer integration
https://gitforwindows.org/
Cancel


Git 2.45.2 Setup
Configuring experimental options
These features are developed actively. Would you like to try them?
Enable experimental support for pseudo consoles.
This allows running native console programs like Node or Python in a
Git Bash window without using winpty, but is unfortunately not quite stable y
Enable experimental built-in file system monitor
(NEW!) Automatically run a built-in file system watcher, to speed up common
operations such as `git status`, `git add`, `git commit`, etc in worktrees
containing many files.
https://gitforwindows.org/
Back
Install
Cancel

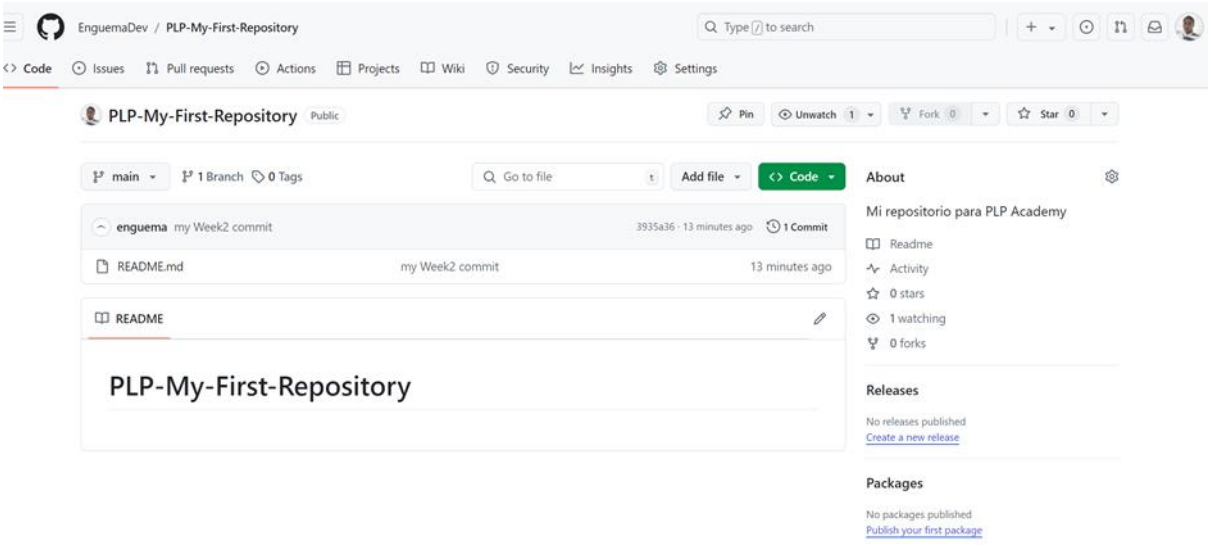

MINGW64:/c/Users/emm...
emman@Enguema MINGW64 ~
$ git --version
git version 2.45.2.windows.1
emman@Enguema MINGW64 ~
$

```
emman@Enguema MINGW64 /d/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository

emman@Enguema MINGW64 /d/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository (main)
$ echo "# PLP-My-First-Repository" >> README.md

emman@Enguema MINGW64 /d/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository (main)
$ git init
Reinitialized existing Git repository in D:/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository/.git/
```

```
emman@Enguema MINGW64 /d/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

emman@Enguema MINGW64 /d/PlpCod/SoftEngenering/my-first-repo/PLP-My-First-Repository (main)
$ git commit -m "my Week2 commit"
[main (root-commit) 3935a36] my Week2 commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

```
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/EnguemaDev/PLP-My-First-Repository.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

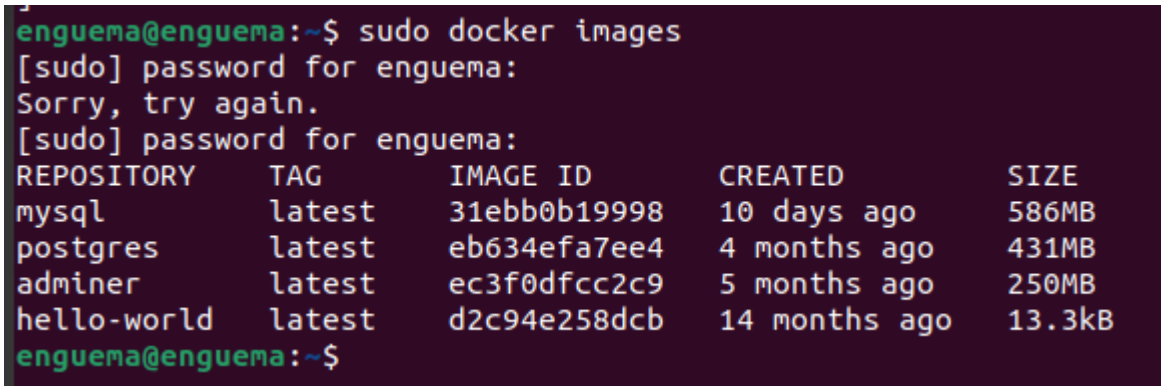
We access the repository again and observe that the changes have been applied



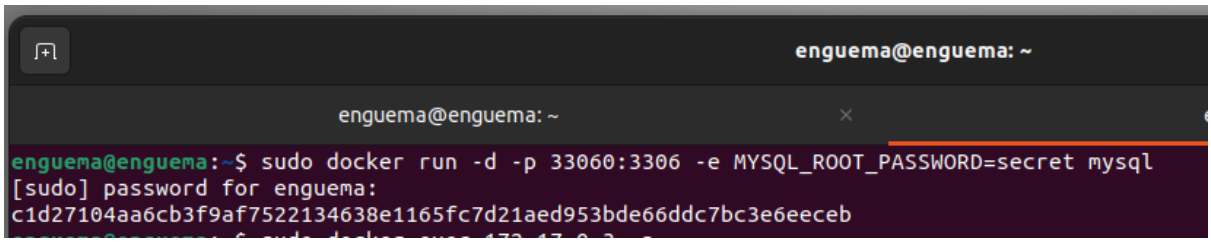
4. Install Necessary Programming Languages and Runtimes: Instal Python from http://wwww.python.org **programming language required for your project and install their respective compilers, interpreters, or runtimes. Ensure you have the necessary tools to build and execute your code.**

6. Configure a Database (MySQL): Download and install MySQL database. https://dev.mysql.com/downloads/windows/installer/5.7.html

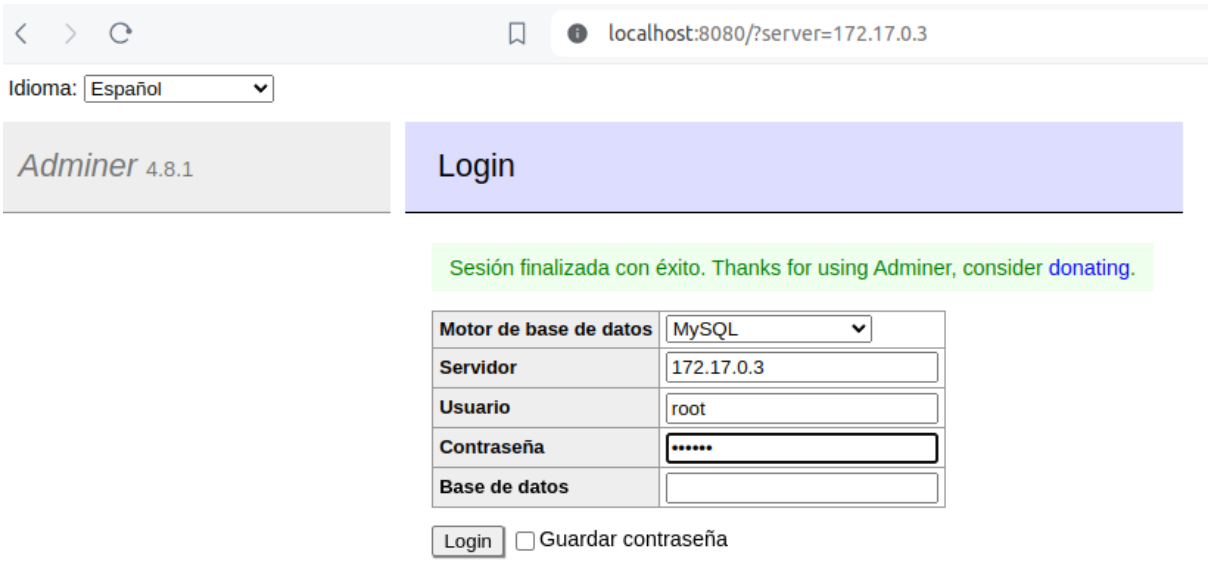
In my case, instead of downloading the mysql installer itself, I will download a mysql docker image with the following command ***docker pull mysql***



After downloading the image, we proceed to create a container based on said image.



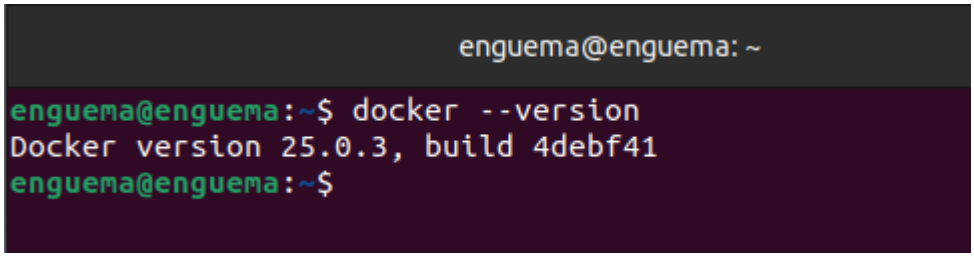
With the container running we access our mysql container.



By logging in successfully I can create my test database.

7. Set Up Development Environments and Virtualization (Optional): Consider using virtualization tools like Docker or virtual machines to isolate project dependencies and ensure consistent environments across different machines.

In my case I will use Docker



PROBLEMS FOUND

Among the difficulties encountered I highlight:
Denial of access to my new github repository when I want to upload content. The error message is shown below:

remote: Permission to EnguemaDev/PLP-My-First-Repository.git is denied to EnguemaDev..
fatal: unable to access https://github.com/EnguemaDev/PLP-My-First-Repository.git': The requested URL returned error: 403

How have I solved it?

Investigating, I have found that this problem is due to the fact that, when uploading changes to a GitHub repository, access credentials to said account are required and Windows automatically searches for them first in the Windows credentials manager in case there is startup data. previous sessions to the account in question, otherwise, it automatically redirects us to the GitHub login screen.

In my specific case, there was login information, but it corresponded to a GitHub account different from the one I use to take these classes and therefore, when trying to use these credentials automatically, it clearly showed me the error indicated above.

As a final solution, I have had to delete the data of my other GitHub account in the Windows credentials manager. Once this is done, when doing a push again

push https://github.com/EnguemaDev/PLP-My-First-Repository.git, It redirected me directly to the GitHub login screen where I entered my credentials and the changes were uploaded to my repository.