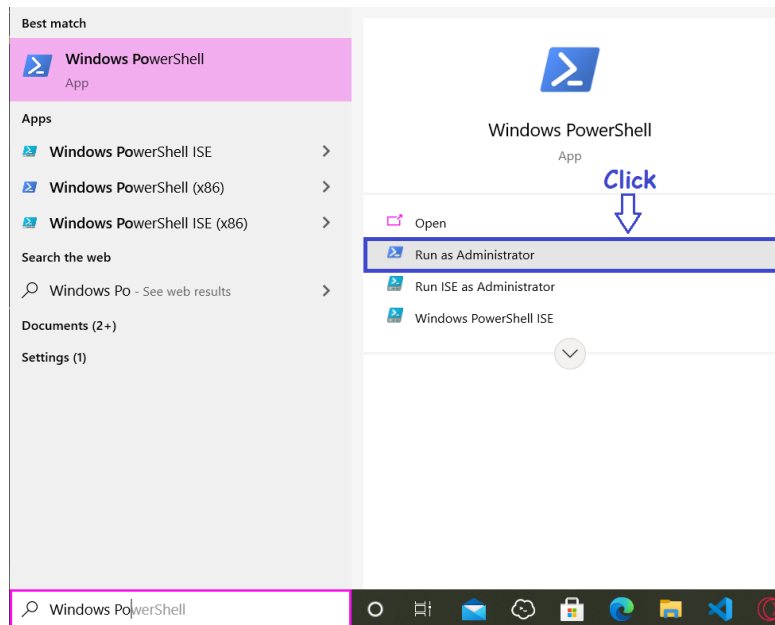


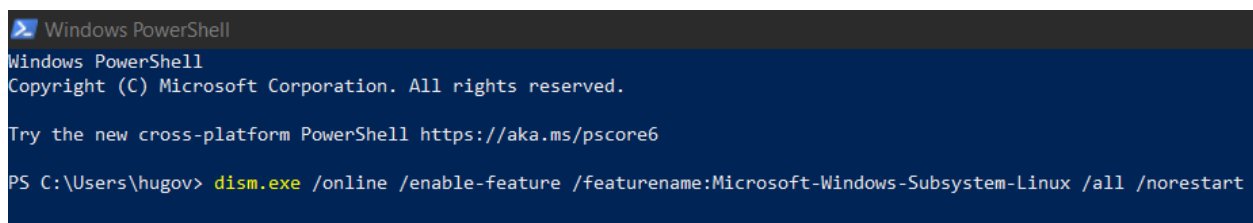
Entorno de Configuración para Windows

- 1) Buscar en windows la Terminal *Windows PowerShell* y *ejecutarla como administrador*.



- 2) Copiar el siguiente comando, pegarlo en la terminal y presionamos **Enter**.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```



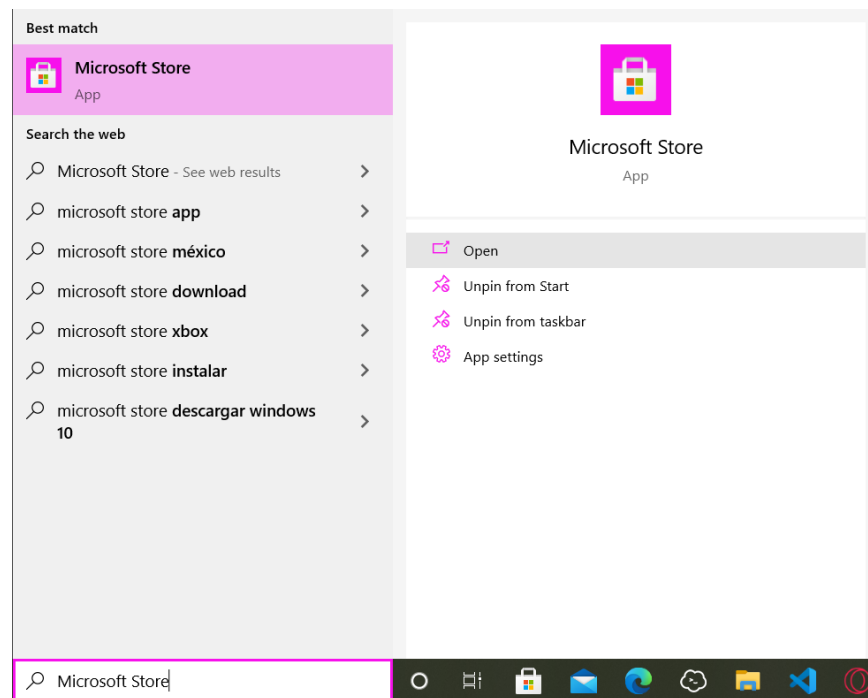
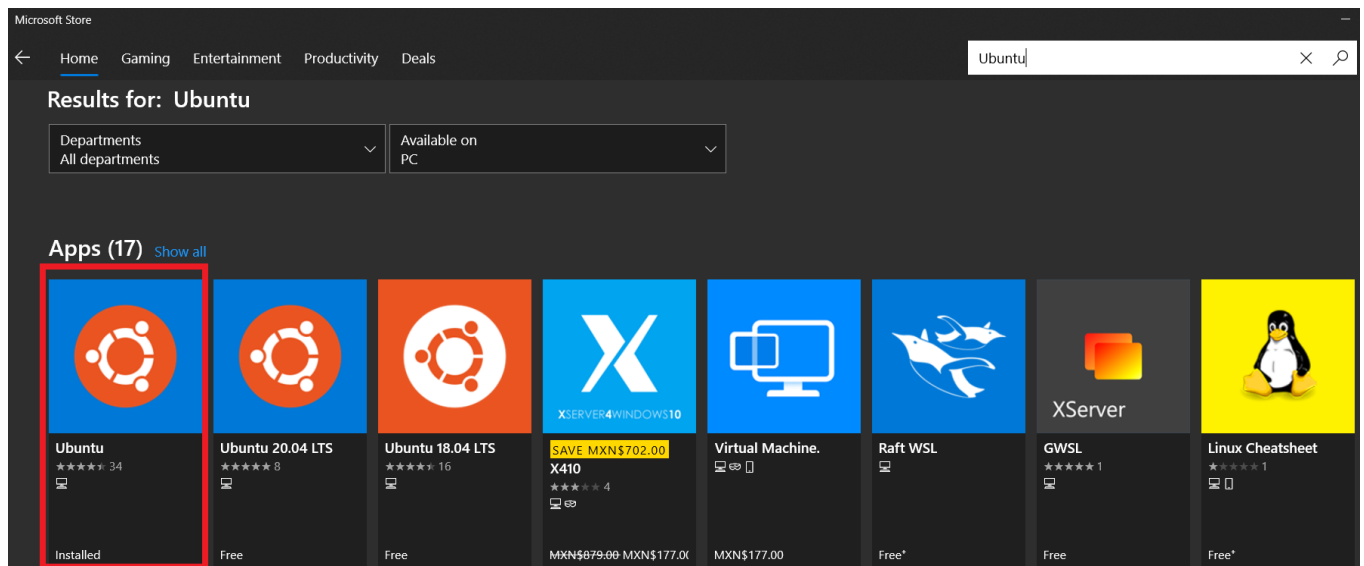
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

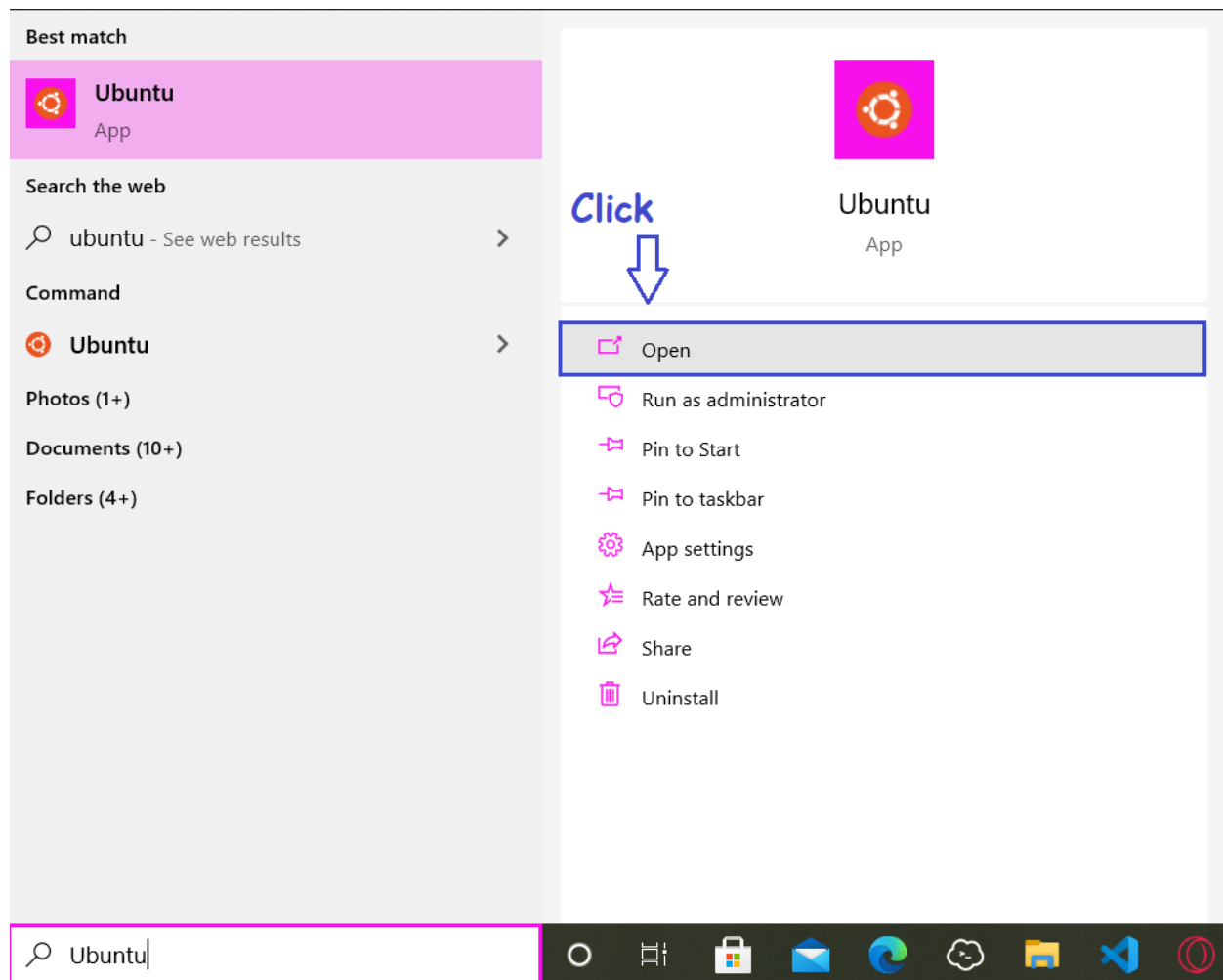
PS C:\Users\hugov> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

- 3) Cuando termine el proceso, reiniciar el sistema.

4) Abrimos la tienda de Microsoft

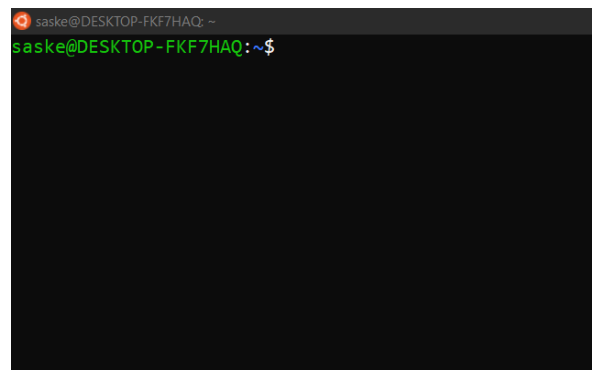
5) Buscamos **Ubuntu** e instalamos la aplicación.

6) Cuando termine el proceso de instalación, buscamos la aplicación y la abrimos.



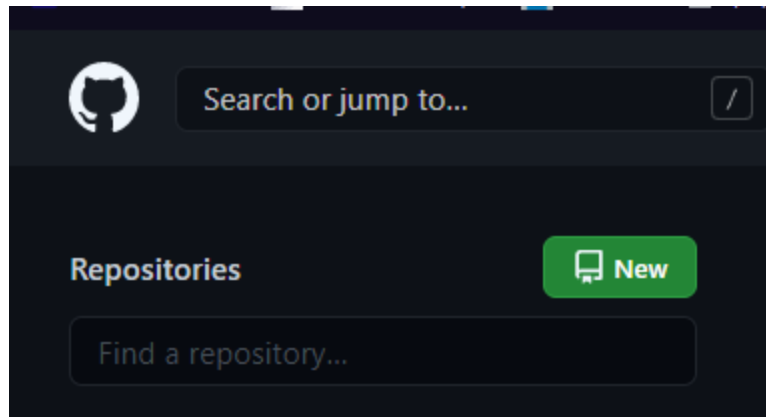
7) Configuramos el nombre de usuario y la contraseña.

8) Al término de la configuración, nuestra terminal se visualizará de la siguiente forma:

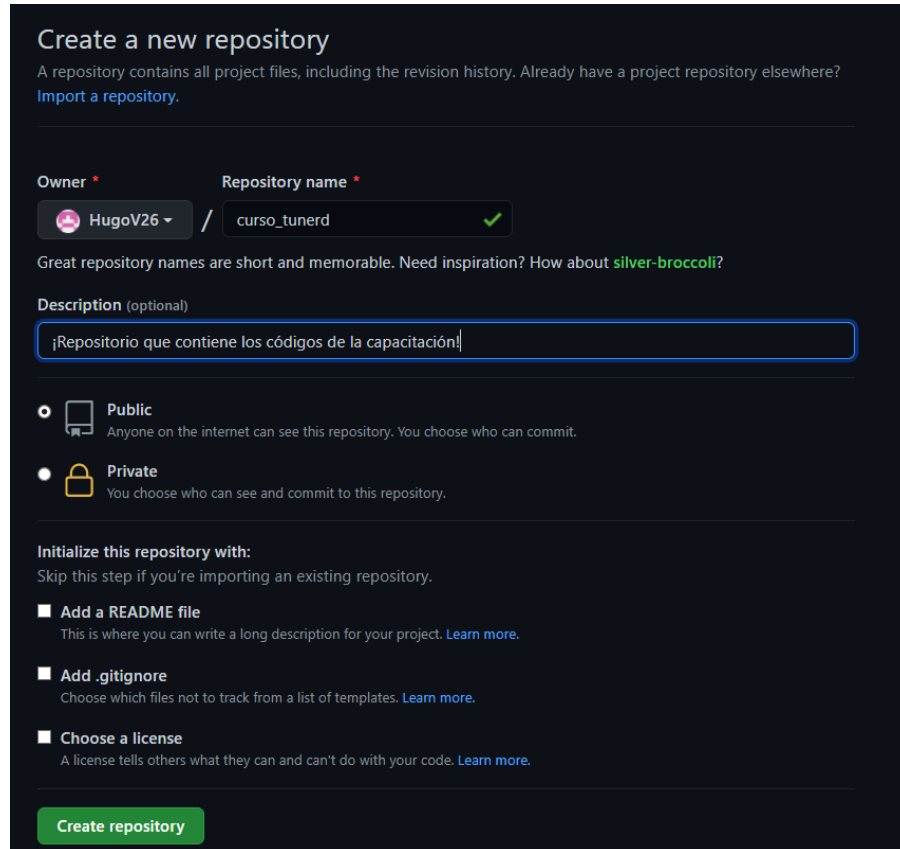


Creación de cuenta y repositorio en <https://github.com>

- 1) Crear una cuenta, llenar el formulario de registro.
- 2) Al terminar el proceso de registro, podremos visualizar el Dashboard principal de nuestra cuenta. Vamos a darle click al botón que dice **"New"**.



- 3) Se nos desplegará un formulario, llenamos los campos correspondientes:
 - a) Indicamos el nombre del repositorio
 - b) En el campo **Description**, colocaremos una explicación breve sobre qué es lo que contiene nuestro repositorio.
 - c) Podremos elegir si queremos que nuestro repositorio sea **público** (cualquiera puede tener acceso) o **privado** (únicamente podrán acceder a él mediante invitación).
 - d) Es posible inicializar el repositorio con cualquiera de las siguientes tres opciones:
 - i) **README file**: un archivo que contiene una explicación detallada del contenido de nuestro repositorio.
 - ii) **.gitignore**: dentro de este archivo podemos colocar el nombre de los archivos que no queremos subir a nuestro repositorio.
 - iii) **License**: una licencia le indicará a los demás qué es lo que pueden o no hacer con tu código.



Create a new repository

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

Owner * Repository name *

HugoV26 / curso_tunerd ✓

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

Description (optional)

¡Repositorio que contiene los códigos de la capacitación!

☒ 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:
Skip this step if you're importing an existing repository.

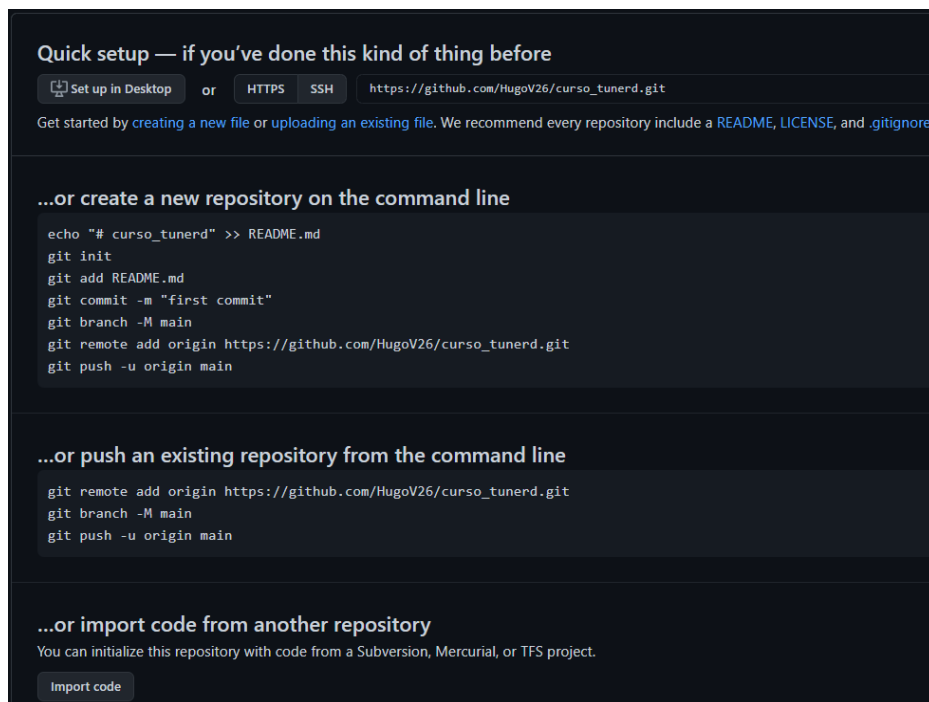
☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

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

Create repository

- 4) Cuando terminemos de llenar el formulario, daremos **click** al botón **Create repository** y nos mostrarán una serie de comandos que ejecutaremos con base a lo siguiente:



Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) https://github.com/HugoV26/curso_tunerd.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# curso_tunerd" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/HugoV26/curso_tunerd.git
git push -u origin main
```

...or push an existing repository from the command line

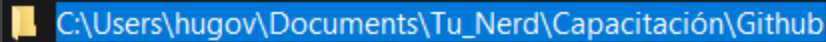
```
git remote add origin https://github.com/HugoV26/curso_tunerd.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

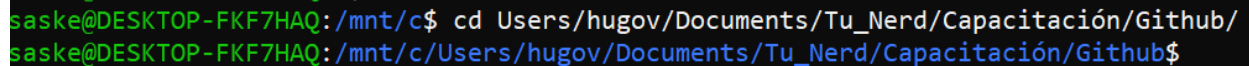
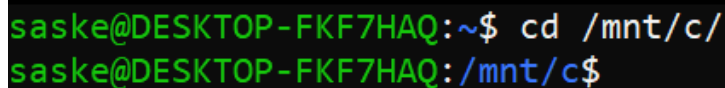
[Import code](#)

- a) Primero verificaremos la ruta en la que se encuentra nuestra carpeta que almacena todos nuestros códigos



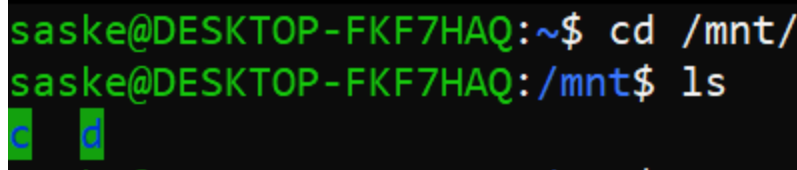
- b) Para este ejemplo, voy a vincular el repositorio que creé anteriormente con la carpeta **Github** que se encuentra en la ruta mostrada. Para esto, vamos a utilizar la terminal de Ubuntu.

- i) Lo primero que tenemos que hacer, es colocarnos en la carpeta **Github**. Para esto utilizaremos el comando **cd (Change Directory)** para posicionarnos en la carpeta mencionada.



- ii) Como se puede observar, nos hemos colocado dentro de la carpeta **Github**. Es importante tomar en cuenta que, la primera carpeta a la que debemos de ingresar es **/mnt/** luego ingresamos al disco en el que está almacenada nuestra carpeta, para mi caso, la tengo en el disco **C**.

- (1) Nota: mediante el comando **ls (List Directory)** podemos listar cuáles son las carpetas y archivos que se encuentran en la ruta en la que nos encontramos, por ejemplo:



Primero ingresamos a la carpeta **/mnt/** y luego listamos el contenido que hay dentro de esta carpeta con el comando **ls**. En mi caso, tengo dos discos, **c** y **d**.

- (2) Nota: podemos reducir el tiempo de escritura con la tecla de **Tabulación**. Por ejemplo, si quisiera ingresar a la carpeta **Users**, podría únicamente escribir **cd /mnt/c/Us** y presionar **Tabulador** para que la terminal de Ubuntu autocomplete el resto del contenido

- iii) Regresando al inciso i), estando posicionados dentro de nuestra carpeta **Github**, ejecutaremos los siguientes comandos:
- (1) **git init** → inicializa un repositorio vacío
 - (2) **git remote add origin**
https://github.com/HugoV26/curso_tunerd.git → apuntamos al repositorio que creamos anteriormente, (este comando se muestra después de haber creado el repositorio, revisar el punto 4).
 - (3) **git status** → este comando nos muestra el estado en el que se encuentran nuestros archivos
 - (4) **git add .** → agregamos **todos** los archivos
 - (5) **git commit -m "Creación de repositorio"** → realizamos un commit de los archivos que agregamos junto con un mensaje breve pero conciso que explique qué es lo que se va a subir.
 - (6) **git branch -M main** → modificamos el nombre de nuestra rama principal
 - (7) **git push -u origin main** → publicamos los cambios en nuestro repositorio

```
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git init
Initialized empty Git repository in /mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github/.git/
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git remote add origin https://github.com/HugoV26/curso_tunerd.git
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Codes/
    Images/
    main.js

nothing added to commit but untracked files present (use "git add" to track)
```

```
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git add .
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git status
On branch master

No commits yet

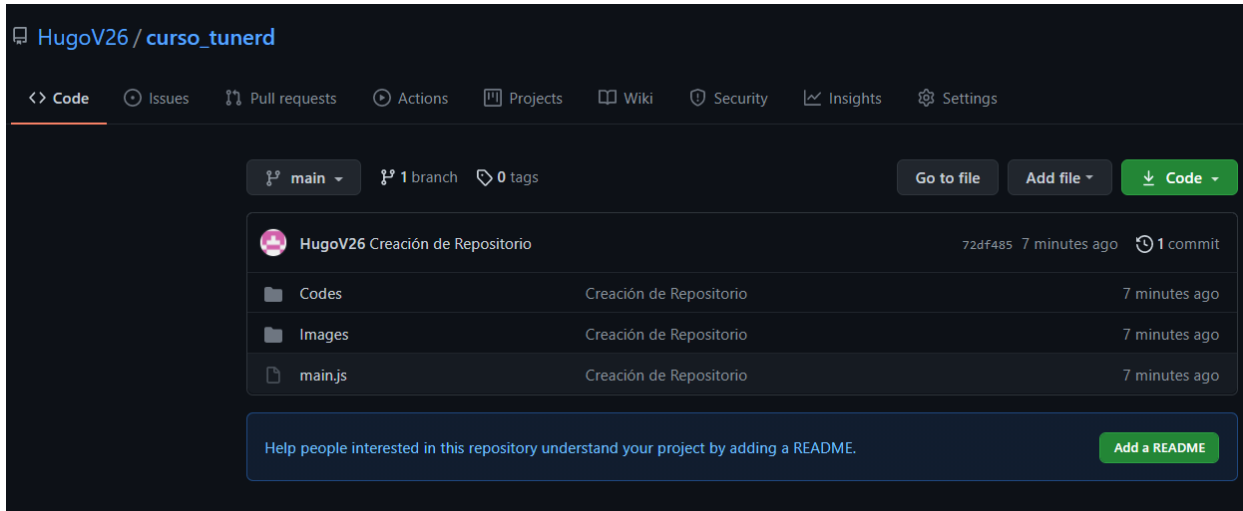
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Codes/main.py
        new file:   Images/step_1_windows_powershell.png
        new file:   Images/step_2_microsoft_store.png
        new file:   Images/step_3_ubuntu.png
        new file:   Images/step_3_ubuntu_app.png
        new file:   Images/step_4_route.png
        new file:   main.js

saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git commit -m "Creación de Repositorio"
[master (root-commit) 72df485] Creación de Repositorio
 7 files changed, 2 insertions(+)
 create mode 100644 Codes/main.py
 create mode 100644 Images/step_1_windows_powershell.png
 create mode 100644 Images/step_2_microsoft_store.png
 create mode 100644 Images/step_3_ubuntu.png
 create mode 100644 Images/step_3_ubuntu_app.png
 create mode 100644 Images/step_4_route.png
 create mode 100644 main.js
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git status
On branch master
nothing to commit, working tree clean

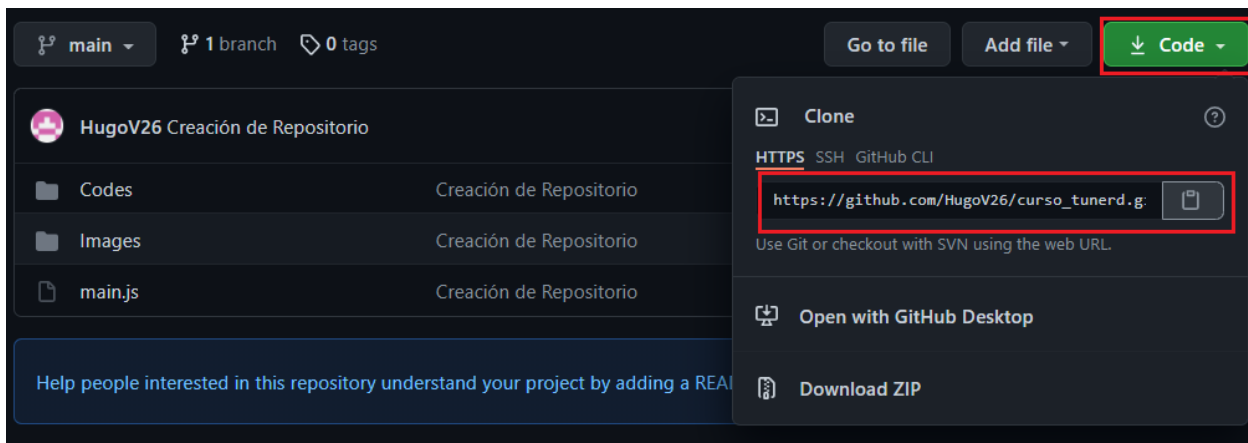
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git branch -M main
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git push -u origin main
Username for 'https://github.com': HugoV26
Password for 'https://HugoV26@github.com':
Counting objects: 11, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 338.74 KiB | 19.92 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To https://github.com/HugoV26/curso_tuner.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

- iv) Refrescamos la página de nuestro repositorio y nuestros archivos ya deberían de aparecer.



Clonar un repositorio

- 1) Si quisiéramos clonar un repositorio de alguien más o incluso uno propio, damos **click** en el botón de **Code** y copiamos la **url del repositorio**.



- 2) Para este ejemplo, voy a clonar el repositorio en mi **Escritorio**.

```
saske@DESKTOP -FKF7HAQ:/mnt/c/Users/hugov/Desktop$ git clone https://github.com/HugoV26/curso_tunerd.git
```

- 3) Ingresamos a la carpeta que tiene **el mismo nombre que le asigne al repositorio** y listo del contenido.

```
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Desktop$ cd curso_tunerd/  
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Desktop/curso_tunerd$ ls  
Codes  Images  main.js  
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Desktop/curso_tunerd$
```

- 4) Ya hemos clonado el repositorio, si quisiéramos publicar nuevos cambios desde esta nueva ruta, los únicos comandos que debemos de ejecutar son:
- a) **git add .**
 - b) **git commit -m "Clon del repositorio y nuevos archivos"**
 - c) **git push -u origin main**
- 5) Ahora, si quisiéramos descargar los cambios que subimos desde **esta nueva ruta** en la ruta donde **originalmente configuramos nuestro repositorio**, lo único que necesitamos es ejecutar el comando **git pull** pero posicionándonos en la ruta original.

```
saske@DESKTOP-FKF7HAQ:/mnt/c/Users/hugov/Documents/Tu_Nerd/Capacitación/Github$ git pull
```