

TUTORIAL:

Instalación y uso de: **Git y GitHub.**

1.0. Descarga e Instalación de Git.

1.1. Descarga de Git.

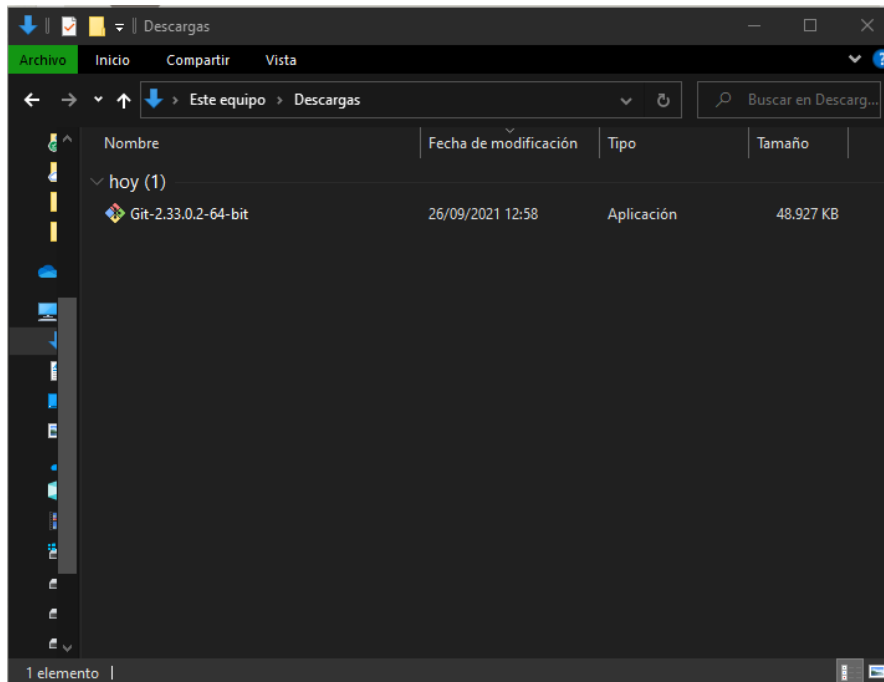
Para descargar la herramienta GIT debemos usar el siguiente hipervínculo: “[Git - Downloading Package](#)”.



Y seleccionaremos “Click here to Download manually” para descargar la última versión de Git.

1.2. Instalación de Git.

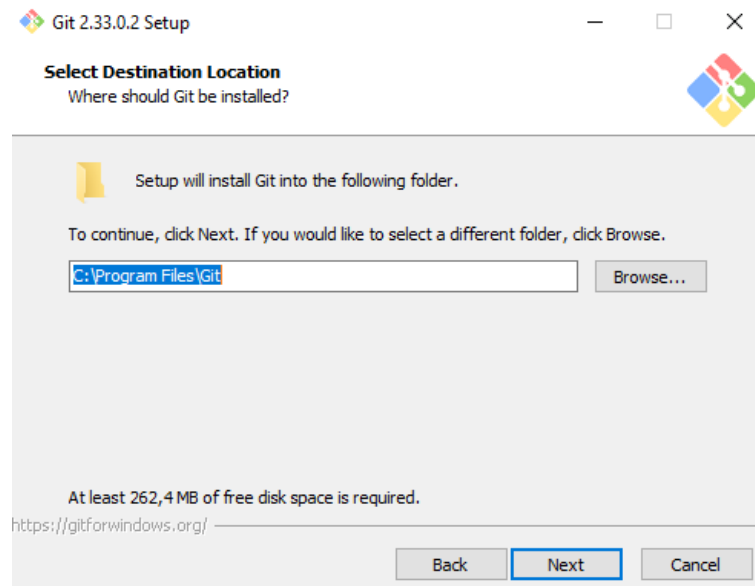
Ejecutaremos él .exe descargado.



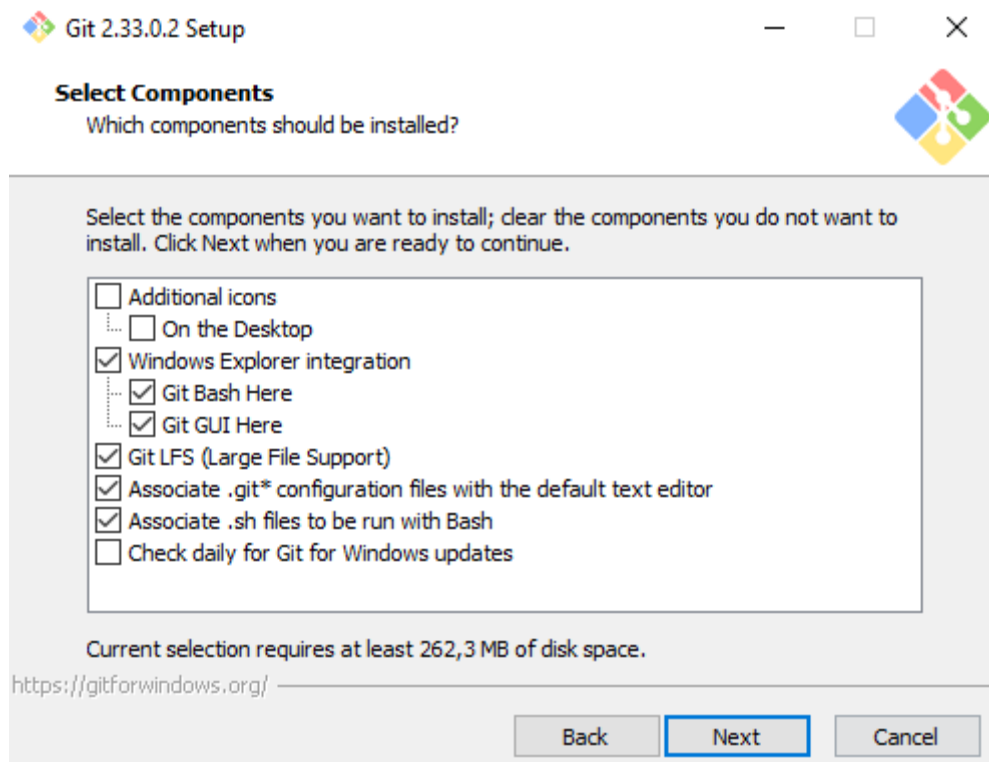
Y continuaremos los pasos del ejecutable.



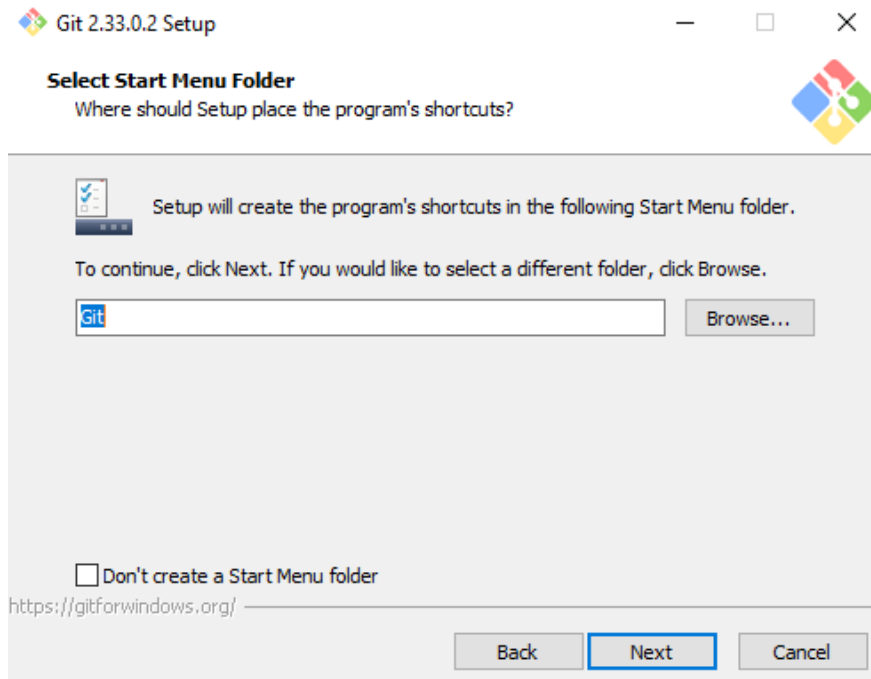
Seleccionaremos la ruta de instalación por defecto. Y pulsaremos “Next” para continuar.



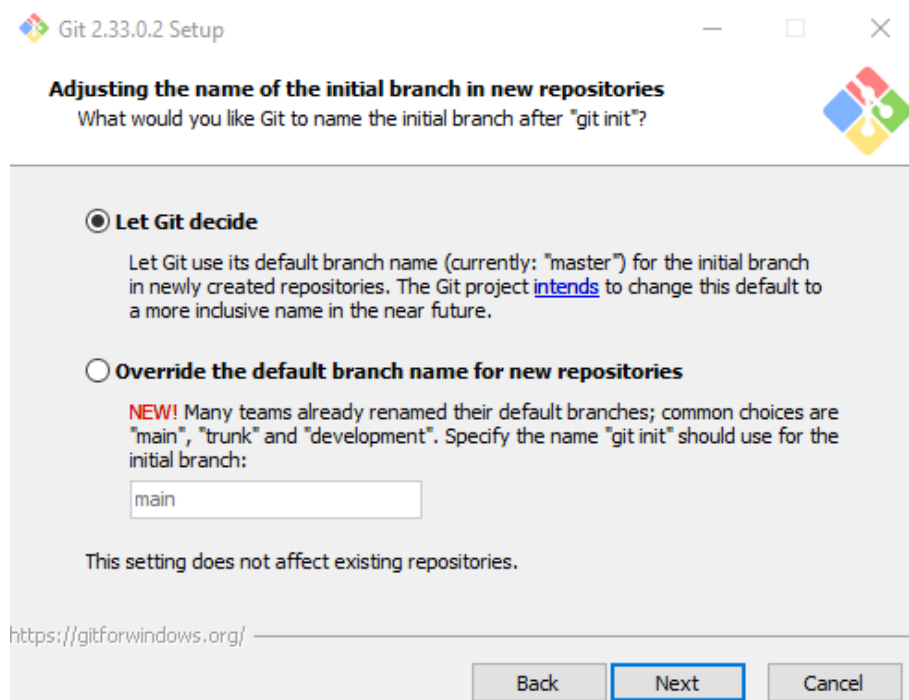
Se abrirá la ventana de elección de complementos, pero la dejaremos de manera preconfigurada, es decir, como viene por defecto. Y pulsamos “Next”.



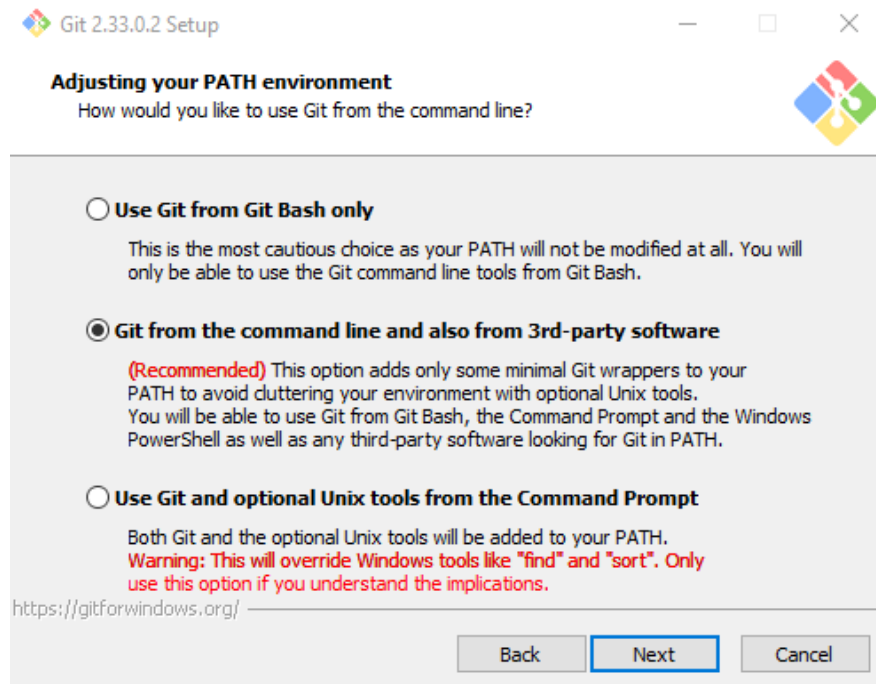
A continuación, nos solicitará la carpeta de inicio para el menú inicial, la dejaremos nombrada como “Git” como viene por defecto.



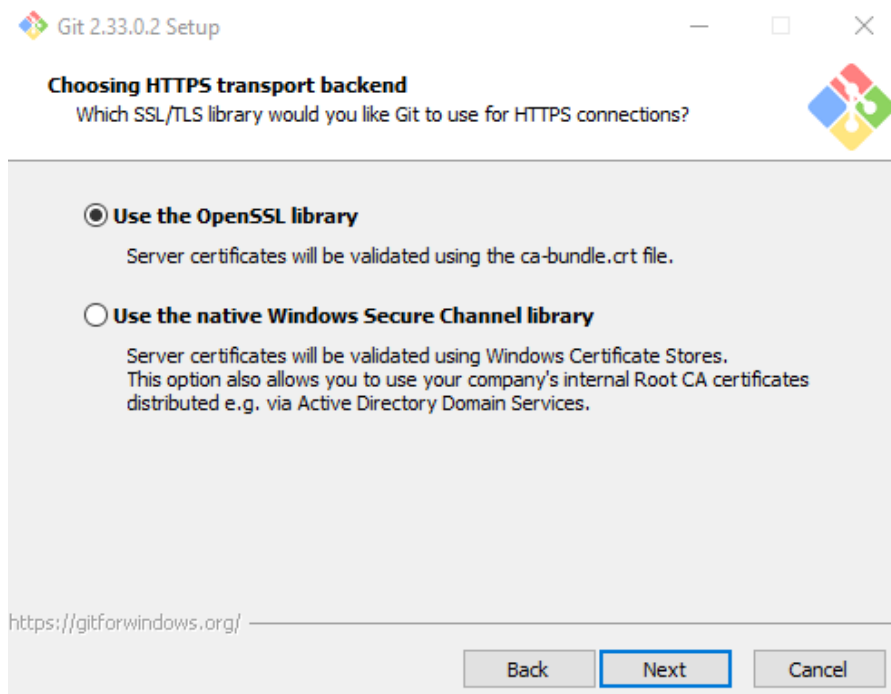
Nos pedirá que seleccionemos “El nombre inicial del Branch en los nuevos repositorios” y dejaremos por defecto como: “main”.



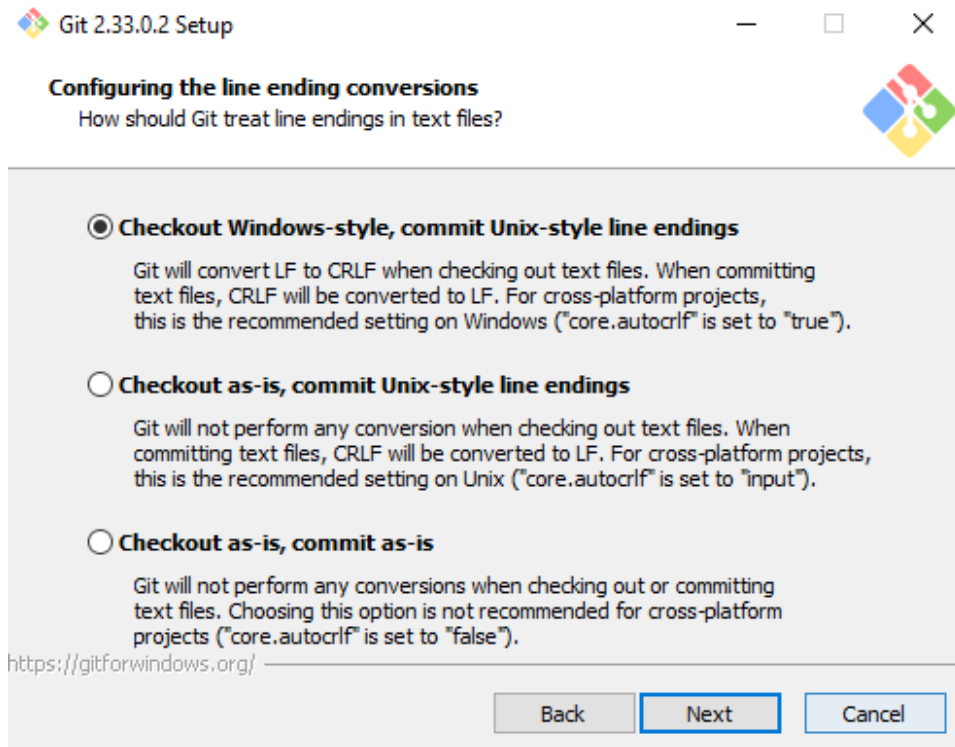
Seleccionaremos la opción para poder usar Git desde la línea de comandos, es decir, la opción predeterminada.



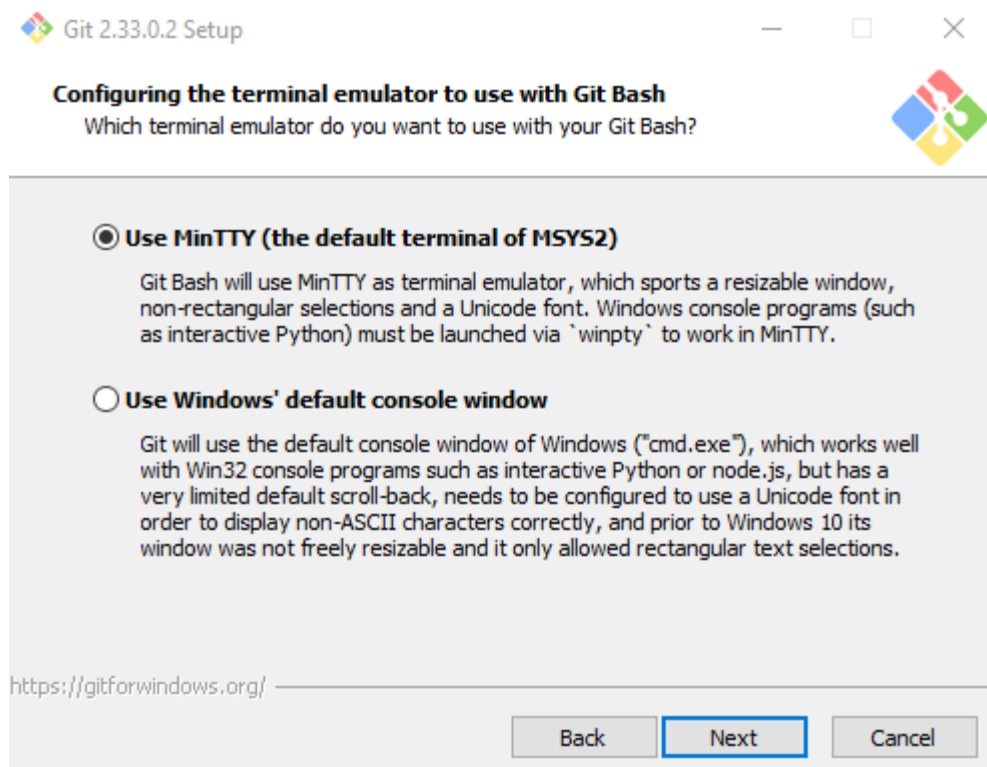
Seleccionaremos la opción por defecto en la ventana “Elegir transporte de HTTPS”.



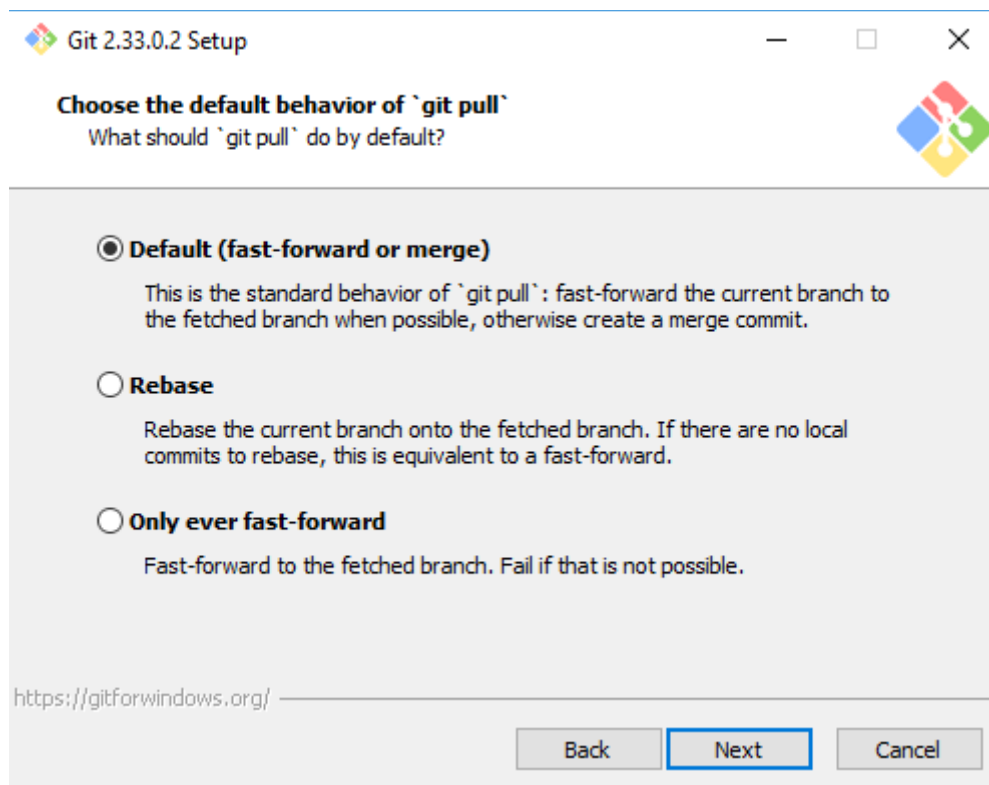
Dejaremos la opción predeterminada por defecto, en la siguiente ventana.



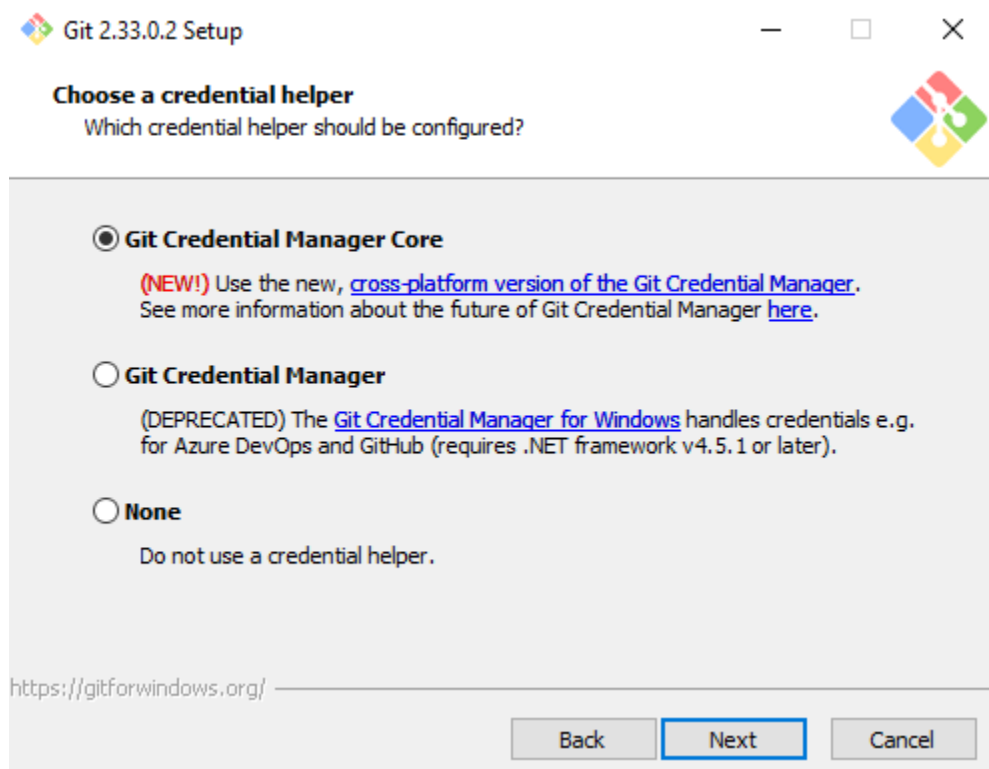
Igualmente, dejaremos la opción predeterminada por defecto, en la siguiente ventana. Que son herramientas adicionales al Bash de Git.



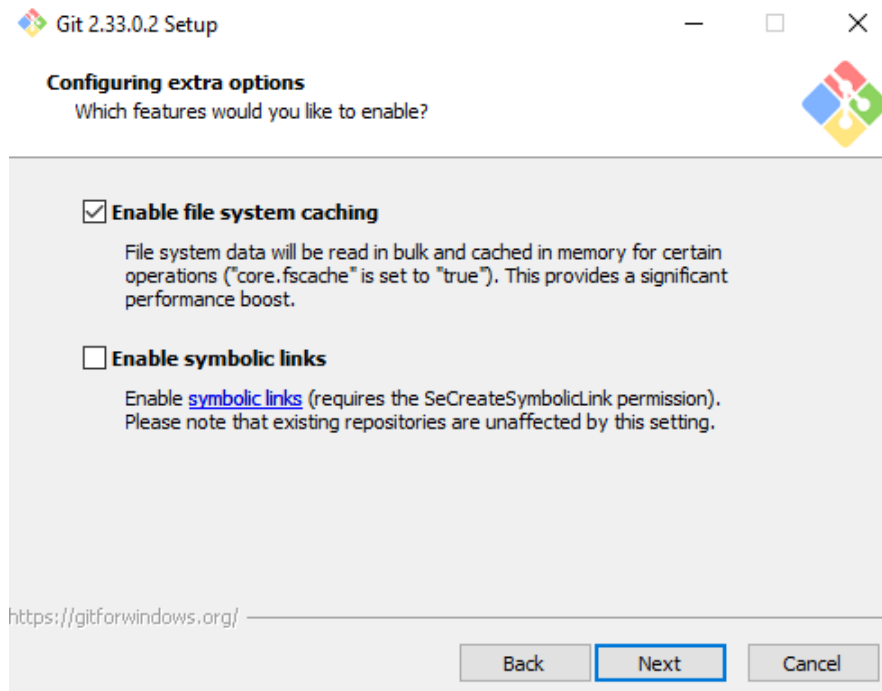
También dejaremos la opción por defecto “Default.”



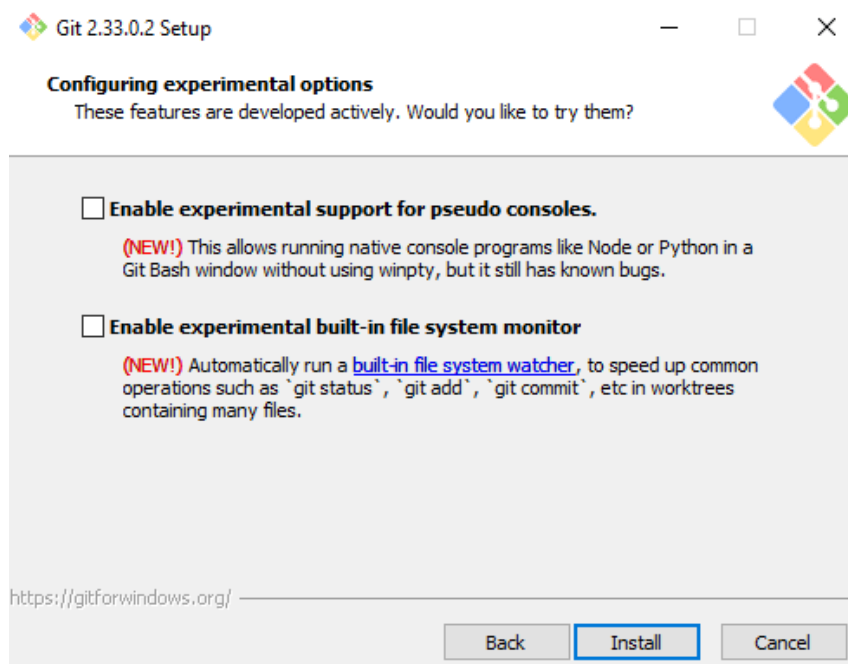
Dejaremos la opción predeterminada en la elección de credenciales.
“Git Credential Manager Core”.



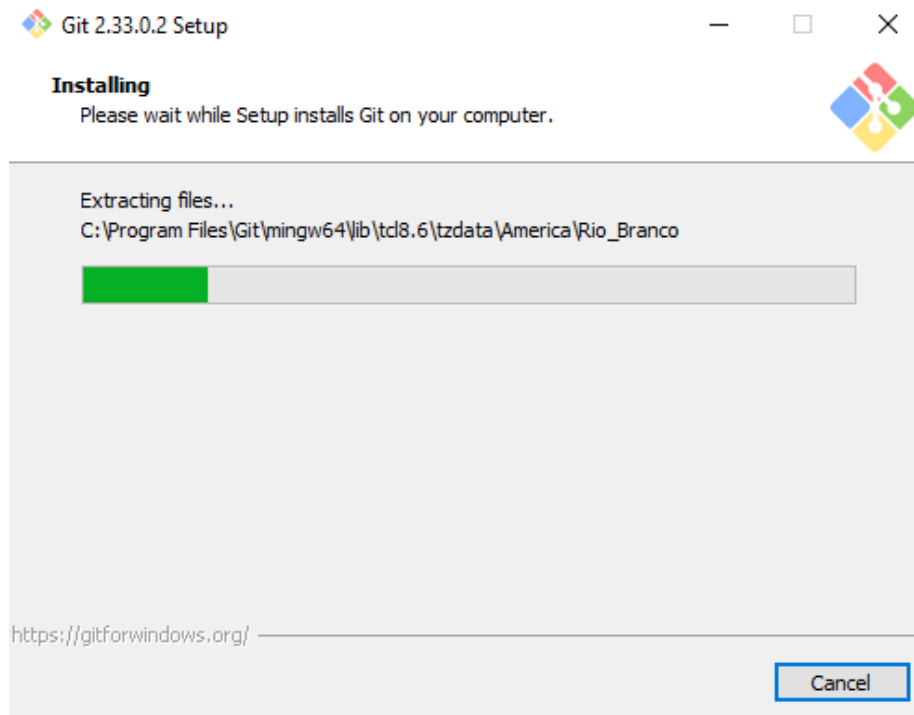
Dejaremos la configuración de opciones extra predeterminada por defecto.



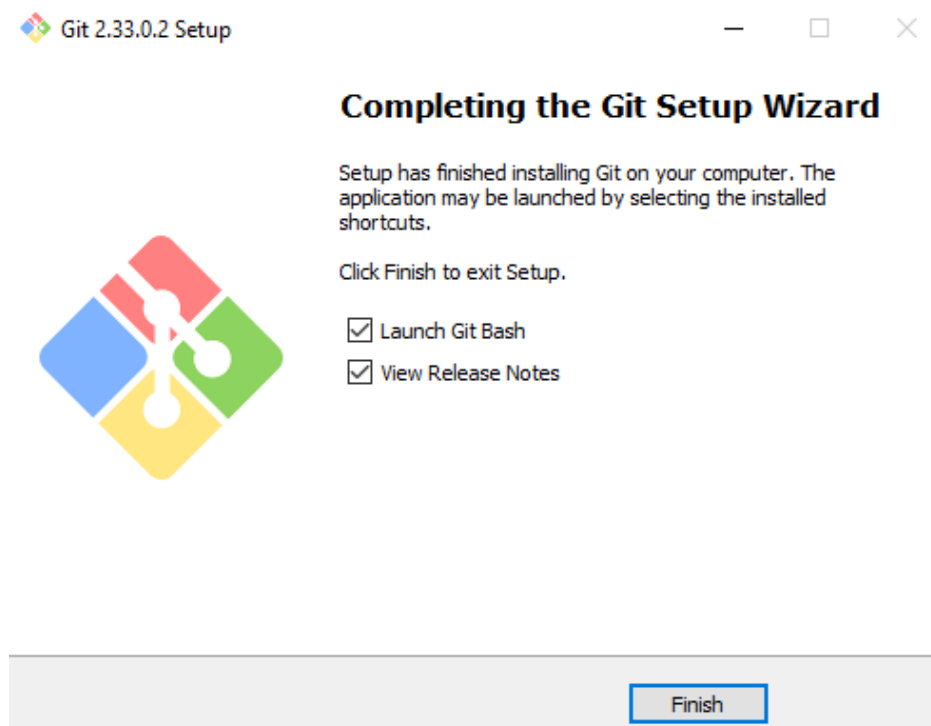
Dejaremos sin tocar las configuraciones de opciones experimentales para evitar errores inesperados.



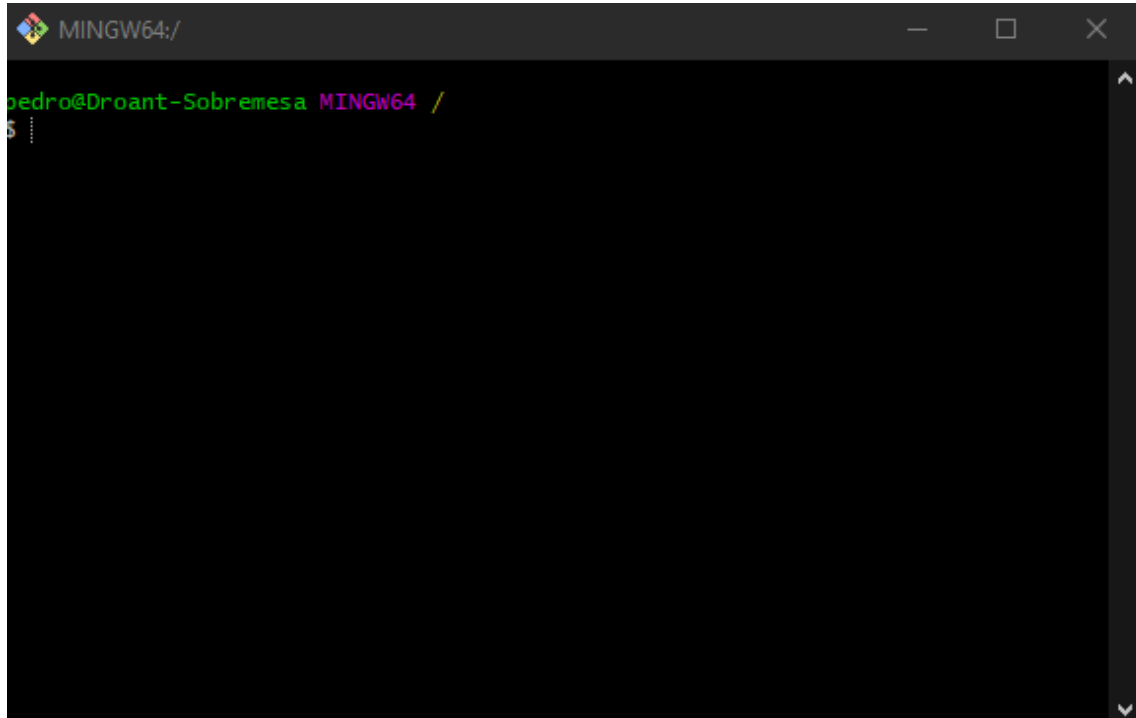
Esperaremos a que el instalador termine la instalación en nuestro equipo.



Al completarse la instalación, marcaremos el “tag” para lanzar Git y pulsaremos finalizar.



Y nos saldrá una ventana de líneas de código propia de Git si todo ha ido bien como la siguiente.

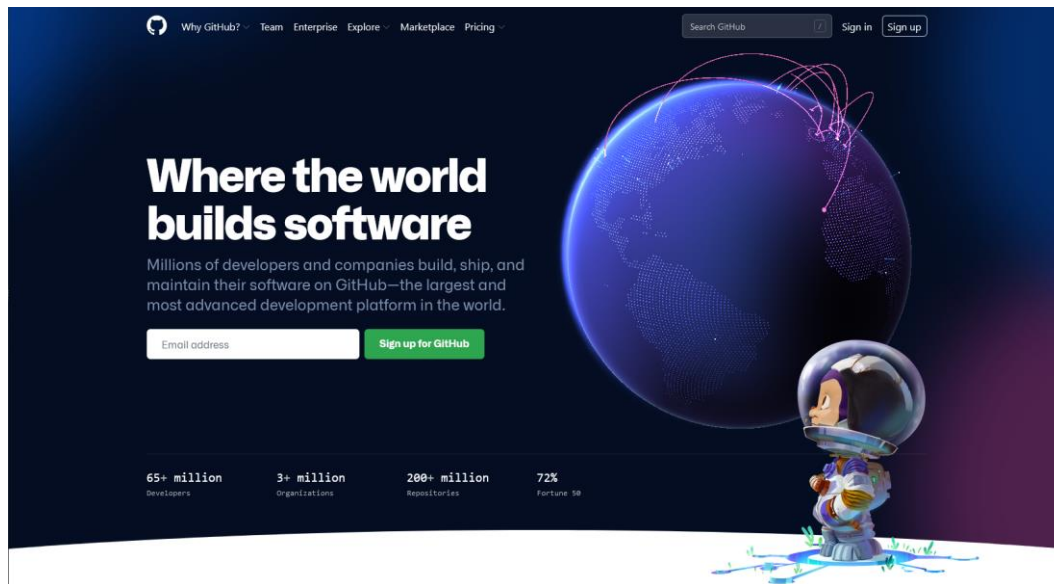
A screenshot of a terminal window titled "MINGW64:". The prompt shows the user "pedro@Droant-Sobremesa" in a "MINGW64" environment. The prompt is followed by a green "\$" and a vertical ellipsis "..." indicating a command has been entered or is being processed. The terminal background is black, and the text is in a monospaced font with some color coding (green for the prompt, pink for the environment).

Con esto se da por finalizada la instalación de Git en nuestro PC.

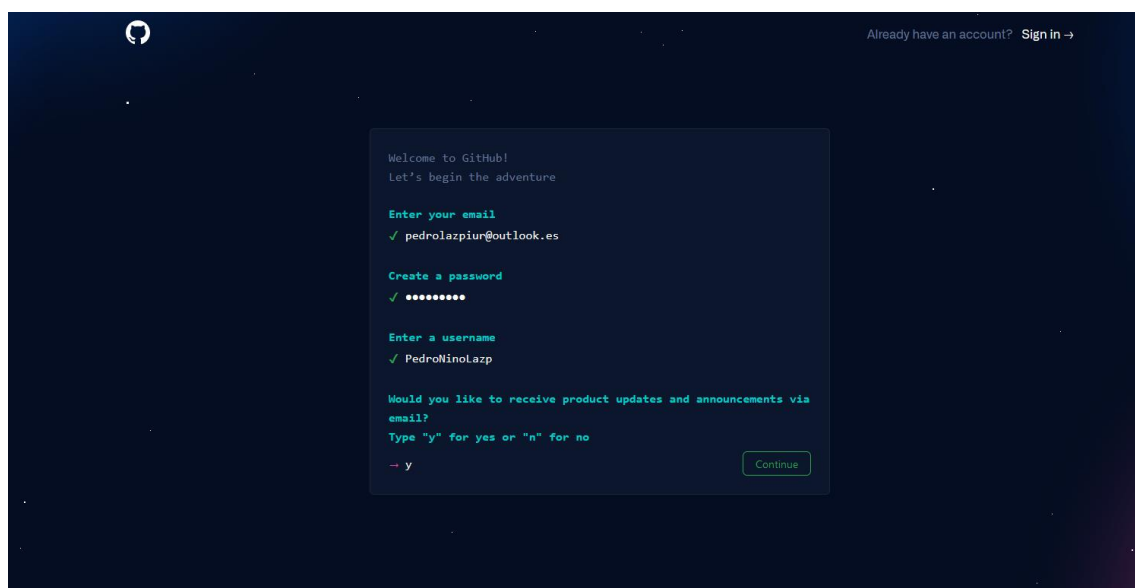
2.0. Creación y Utilización de un Nuevo Repositorio en GitHub.

2.1. Creación de Cuenta en GitHub.

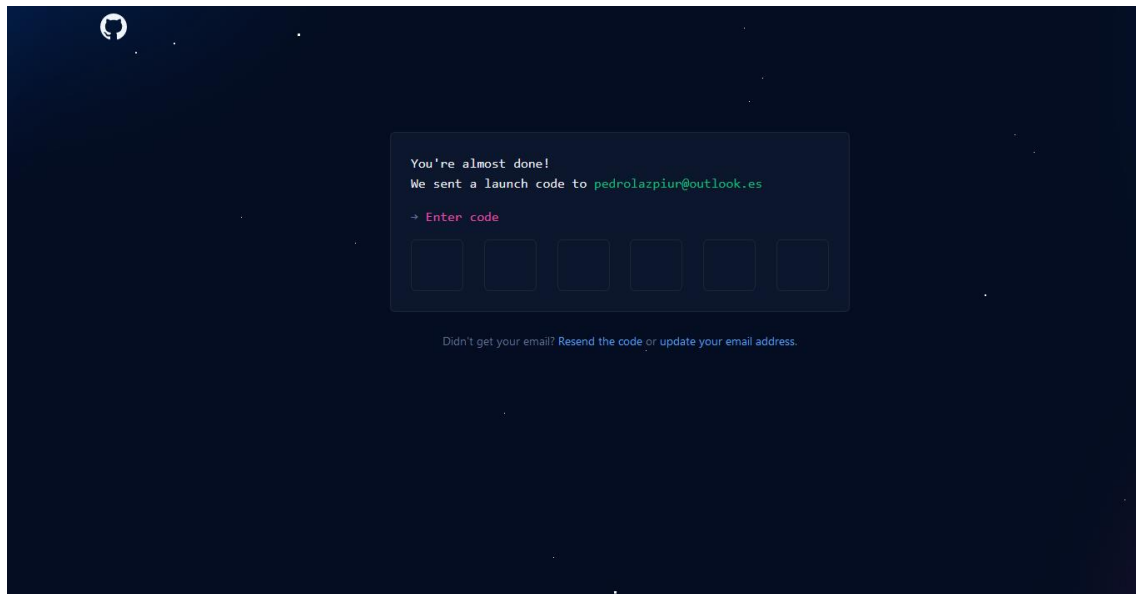
Para empezar, debemos ir al siguiente enlace: "<https://github.com/>", en el cual pulsaremos el botón de la parte superior izquierda "Sign up".



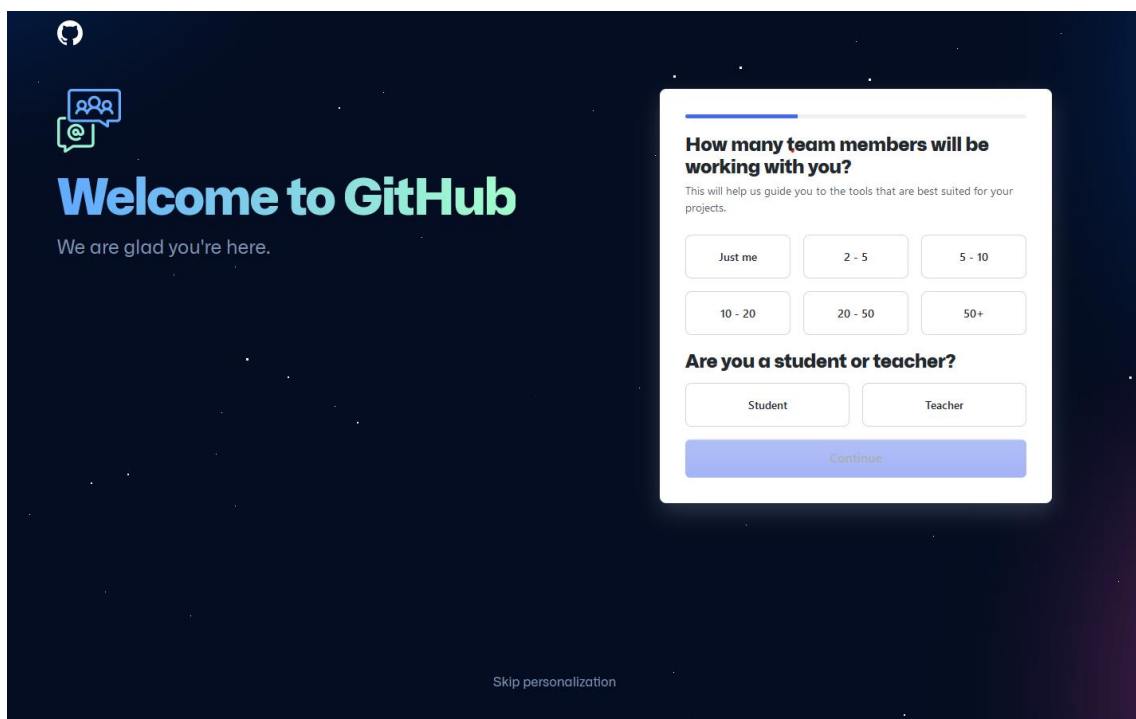
Nos abrirá la siguiente ventana y rellenaremos con nuestro email, la contraseña que deseemos y el nombre de usuario que tendremos en nuestra cuenta. (En la siguiente imagen no se muestran los datos actuales debido a que no realiza esa captura en el momento de crearlo y ahora produce error por estar ya ocupado por lo que he utilizado otra simplemente para poder mostrar el ejemplo).



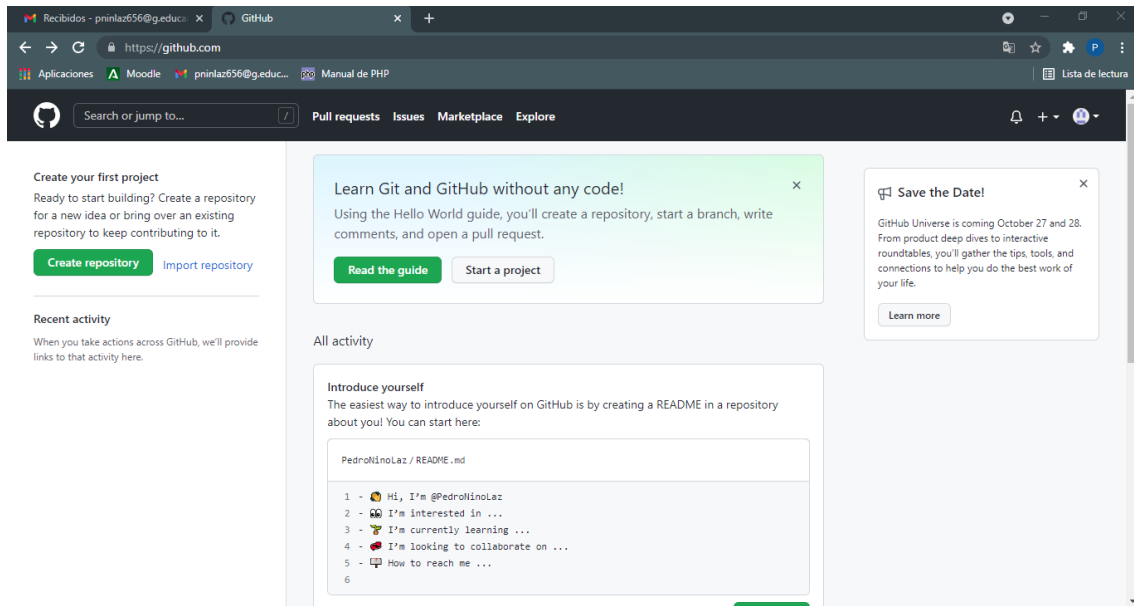
Nos solicitará una vez pulsado continuar en la página anterior resolver un captcha para demostrar que no somos un “bot”, y nos abrirá la siguiente página para introducir un código de 6 dígitos, que nos ha sido enviado a al correo electrónico con el que hemos creado la cuenta.



Continuara con una página para la personalización de la cuenta, podemos seleccionar nuestras preferencias, o bien saltaras dándole a “Skip personalization” en la parte inferior.

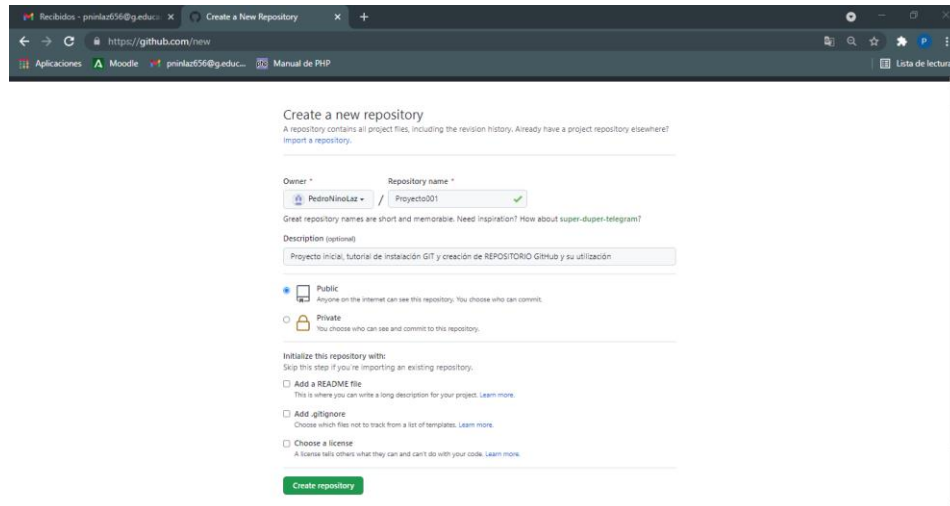


Una vez realizado los pasos anteriores ya habremos creado nuestra cuenta en GitHub.

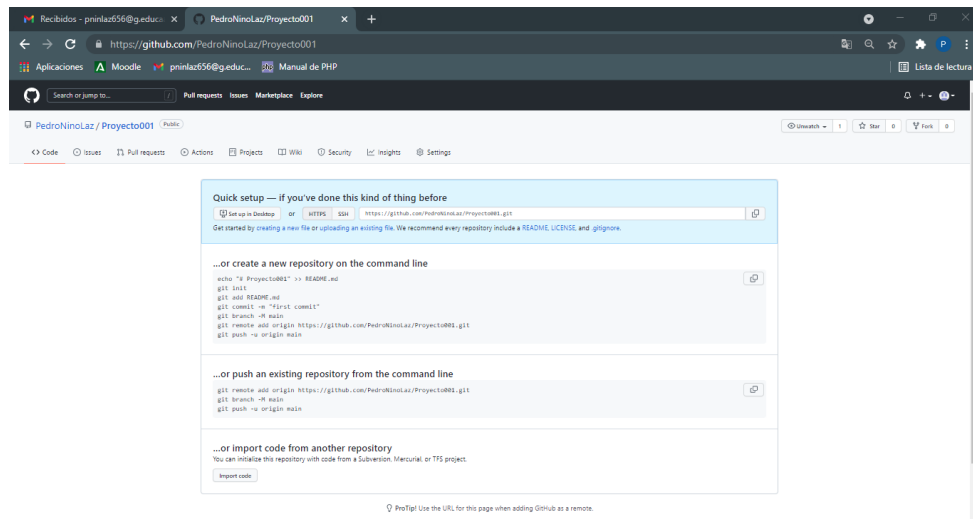


2.1. Creación de un Nuevo Repositorio en GitHub.

En la imagen anterior en la parte izquierda de la página tenemos un botón en verde que indica “Create repository” lo pulsaremos y se nos abrirá la siguiente página.



Elegiremos el Owner (“Dueño” o “Creador”) que por defecto es el nombre de usuario con el que hemos iniciado sesión, y pondremos el nombre a nuestro repositorio en este caso el “Owner: PedroNinoLaz” y el nombre del repositorio nuevo es: “Proyecto001”. Añadiremos una descripción si lo deseamos, seleccionaremos si es “Privado o Público” (Público en este caso) y los “tags” que deseamos, (ninguno en este repositorio). Y nos mostrará los comandos con los que podremos interactuar con Git en GitHub.

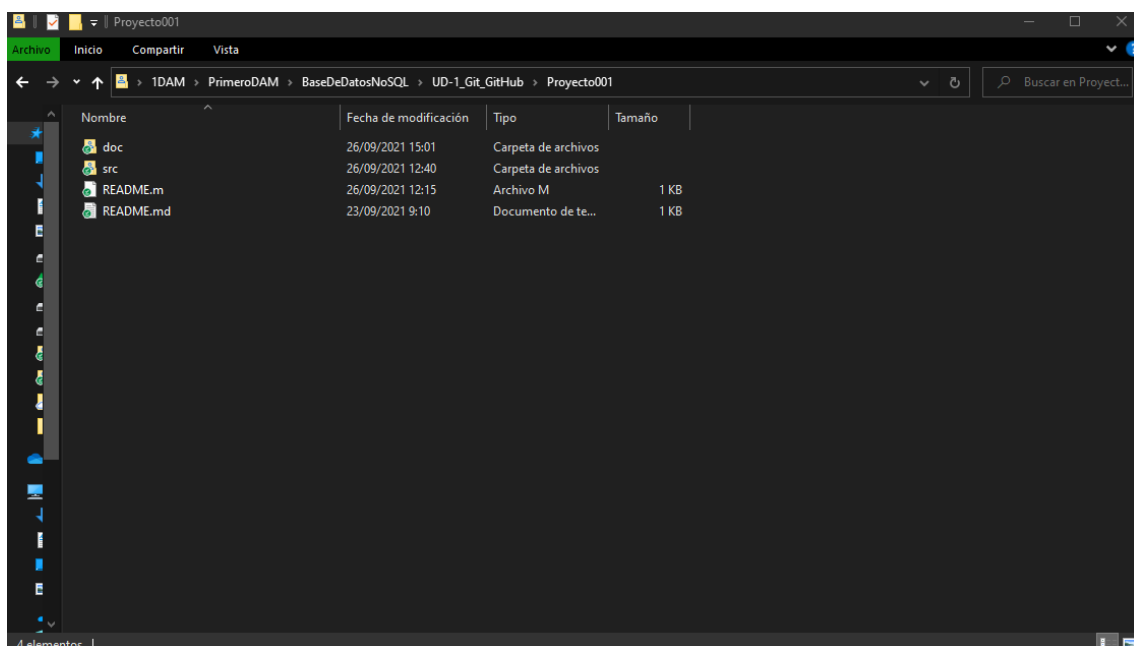


3.0. Creación de Archivos Locales, Uso de GitHub a través de PowerShell y Sincronización de GitHub con Archivos Locales.

3.1. Creación de Archivos Locales.

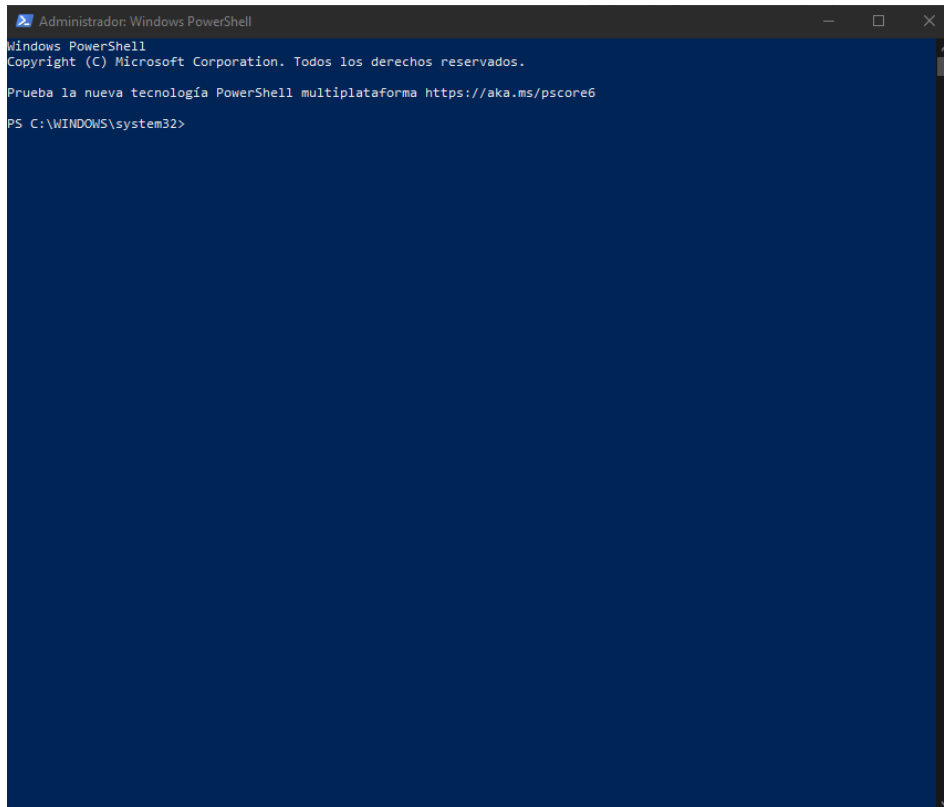
Los archivos locales que necesitaremos crear para esta práctica son los siguientes.

- Capeta destinada a guardar el proyecto → Proyecto001
- Dentro de la misma dos carpetas → doc y src.
- Y un documento de texto → README.md



3.2. Uso de GitHub a través de PowerShell.

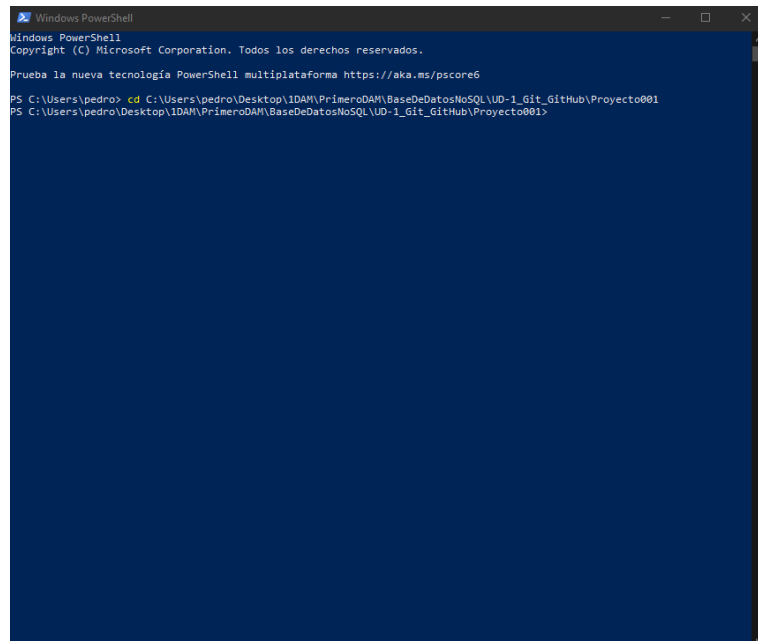
Antes de nada explicaremos que es “PowerShell”: Es un interfaz de consola con el cual se pueden escribir comandos por medio de instrucciones es el predecesor de la consola CRM de los anteriores sistemas operativos Windows.



A través de ella, nos comunicaremos con GitHub para subir archivos (push), a nuestro repositorio que hemos creado previamente. Iremos explicando cada parte del proceso a continuación.

Lo primero será acceder a la ruta específica de la carpeta dónde trabajaremos, es decir “Proyecto001” con el comando “cd”

cd C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\pedro> cd C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001>
```

A continuación, utilizaremos los comandos iniciales que nos mostro GitHub al crear el nuevo repositorio que son los siguientes.

echo “# Proyecto001” >> README.md

git init

git add README.md

git commit -m “first commit”

git branch -M main

git remote add origin <https://github.com/PedroNinoLaz/Proyecto001.git>

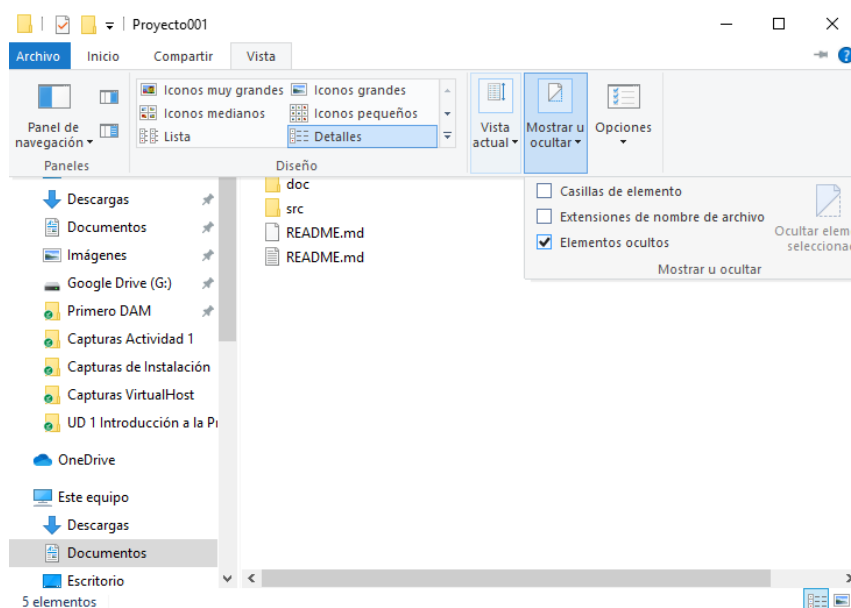
git push -u origin main

Usaremos el primer commando “echo “# Proyecto001” >> README.md. y el comando “git init”.

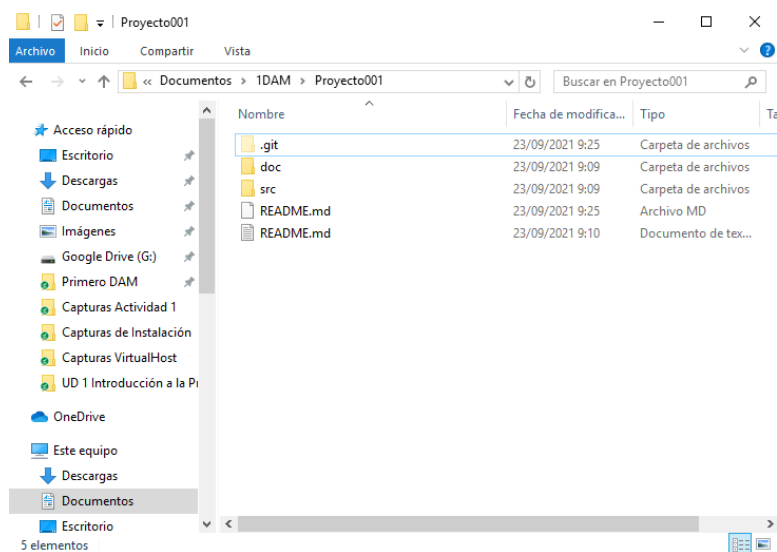
```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS C:\Windows\system32> cd C:\Users\alumno\Documents\1DAM\Proyecto001
PS C:\Users\alumno\Documents\1DAM\Proyecto001>
PS C:\Users\alumno\Documents\1DAM\Proyecto001> echo "# Proyecto001" >> README.md
PS C:\Users\alumno\Documents\1DAM\Proyecto001>
PS C:\Users\alumno\Documents\1DAM\Proyecto001> git init
Initialized empty Git repository in C:/Users/alumno/Documents/1DAM/Proyecto001/.git/
PS C:\Users\alumno\Documents\1DAM\Proyecto001>
```

Al utilizar el comando “git init” nos creará una carpeta dentro de la carpeta “Proyeccto001” con el nombre “.git” que estará oculta, eso se resuelve de la siguiente manera, en la barra de opciones de cascada seleccionaremos la ultima opción “Vista” se nos abrirá un desplegable con múltiples opciones y seleccionaremos “Mostrar u ocultar” que nos abrirá otro desplegable con tres opciones y marcaremos el “tag” “Elementos ocultos”.

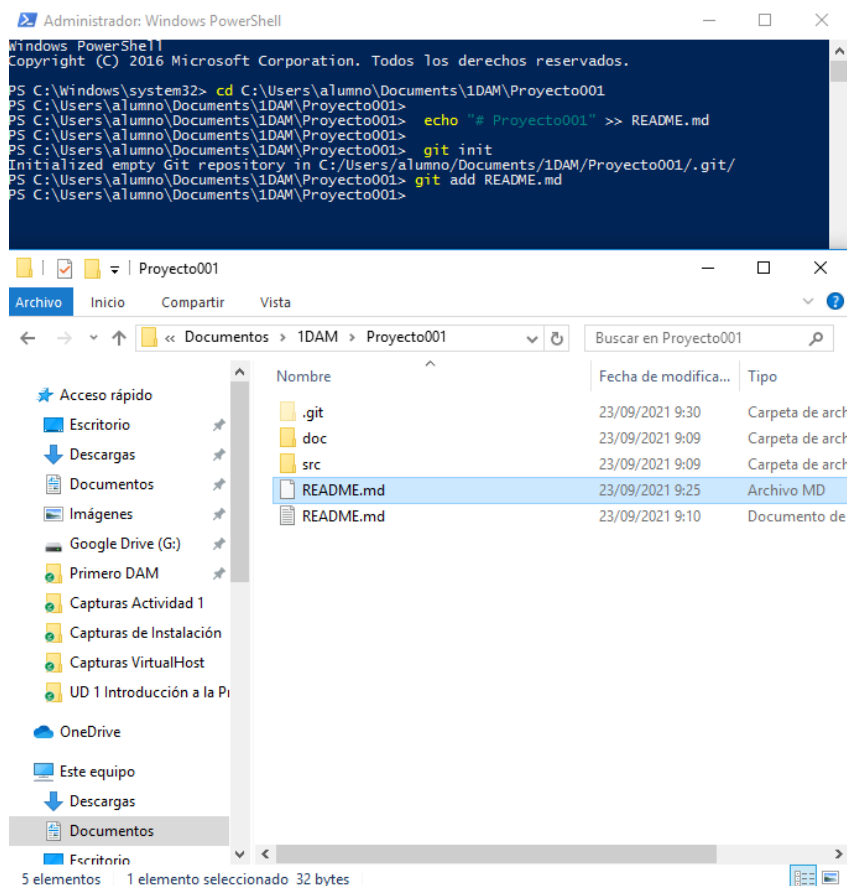


Y quedando de la siguiente manera:



Y siendo posible visualizar la carpeta “.git”.

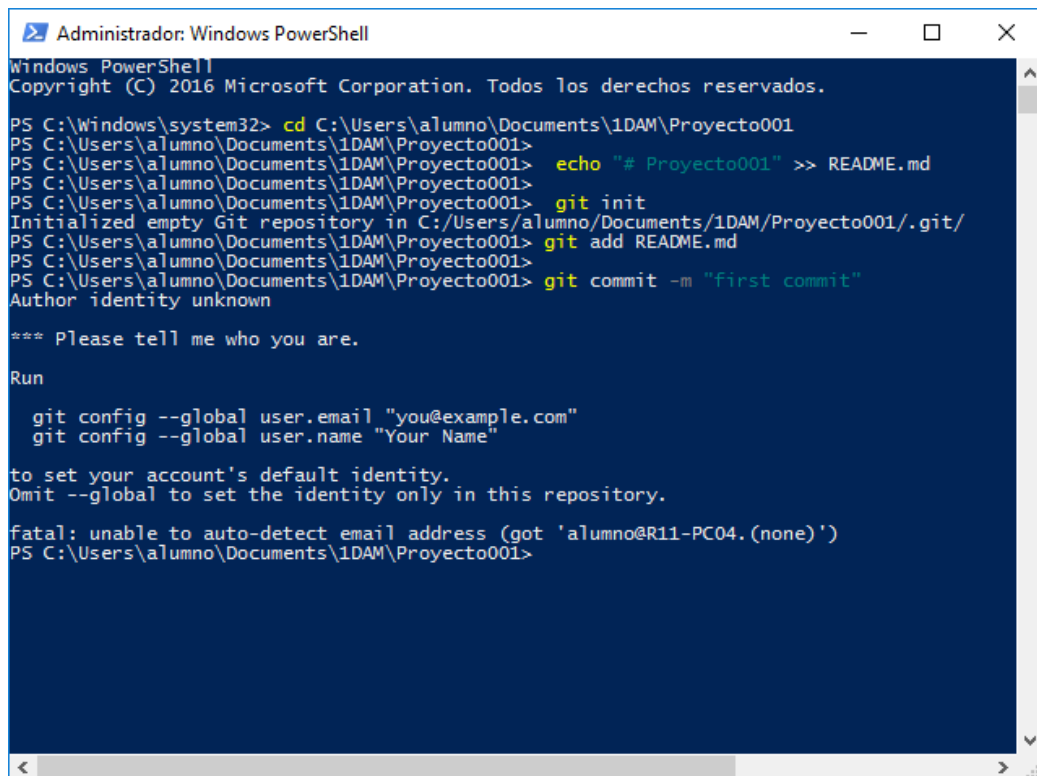
(Aún que en la imagen anterior ya aparece (porque se realizó la captura posteriormente). Vamos a utilizar el comando “**git add README.md**”. El cuál nos va a crear un archivo README.md dentro de nuestra carpeta Proyecto001.



Vamos a continuar con el comando `git commit -m "first commit"`, (la primera vez no pedirá que nos identifiquemos con los comandos:

```
git config --global user.email "you@example.com"
```

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



```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS C:\Windows\system32> cd C:\Users\alumno\Documents\1DAM\Proyecto001
PS C:\Users\alumno\Documents\1DAM\Proyecto001>
PS C:\Users\alumno\Documents\1DAM\Proyecto001> echo "# Proyecto001" >> README.md
PS C:\Users\alumno\Documents\1DAM\Proyecto001> git init
Initialized empty Git repository in C:/Users/alumno/Documents/1DAM/Proyecto001/.git/
PS C:\Users\alumno\Documents\1DAM\Proyecto001> git add README.md
PS C:\Users\alumno\Documents\1DAM\Proyecto001> git commit -m "First commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

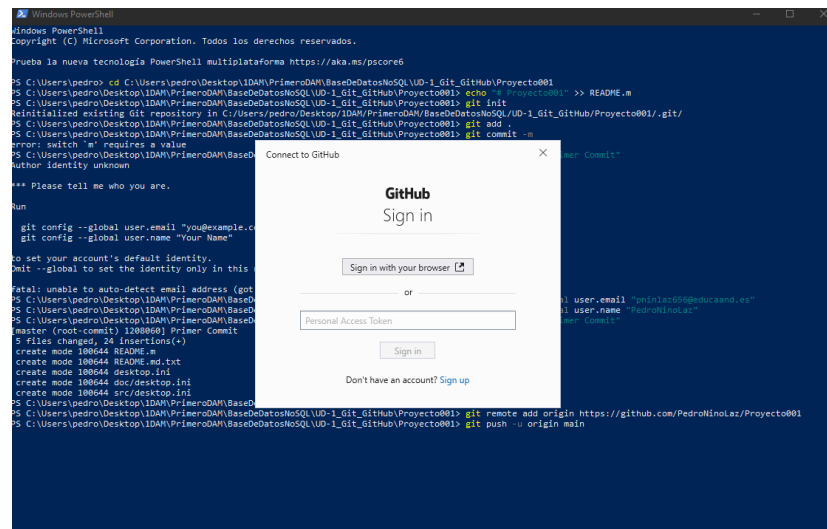
fatal: unable to auto-detect email address (got 'alumno@R11-PC04.(none)')
PS C:\Users\alumno\Documents\1DAM\Proyecto001>
```

Después usaremos

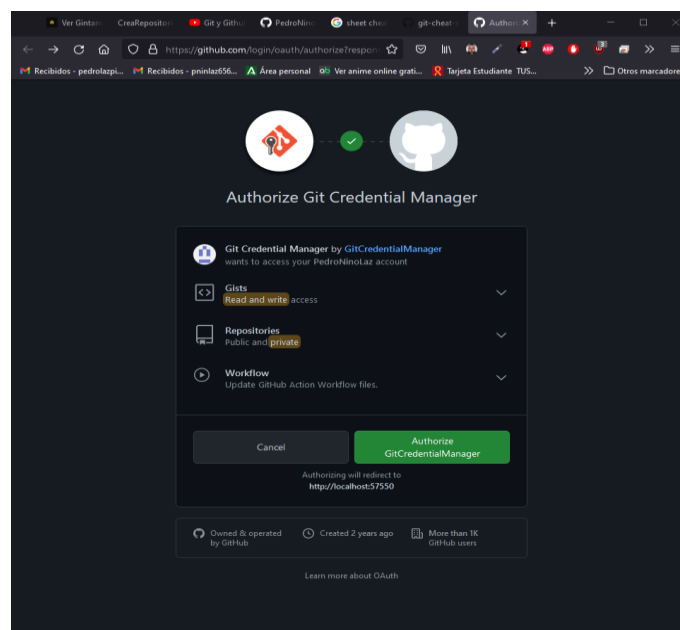
```
git remote add origin https://github.com/PedroNinoLaz/Proyecto001
```

```
git push -u origin main
```

Al introducir nuestras credenciales que nos ha solicitado nos abrirá una ventana para poder dar identificarnos en la página de GitHub.



Seleccionaremos “Sign in with your browser” (identificarse en tú navegador) y nos abrirá la siguiente página. En la cual nos identificaremos.



Y ya tendremos nuestro PowerShell sincronizado con GitHub para poder usarlo.

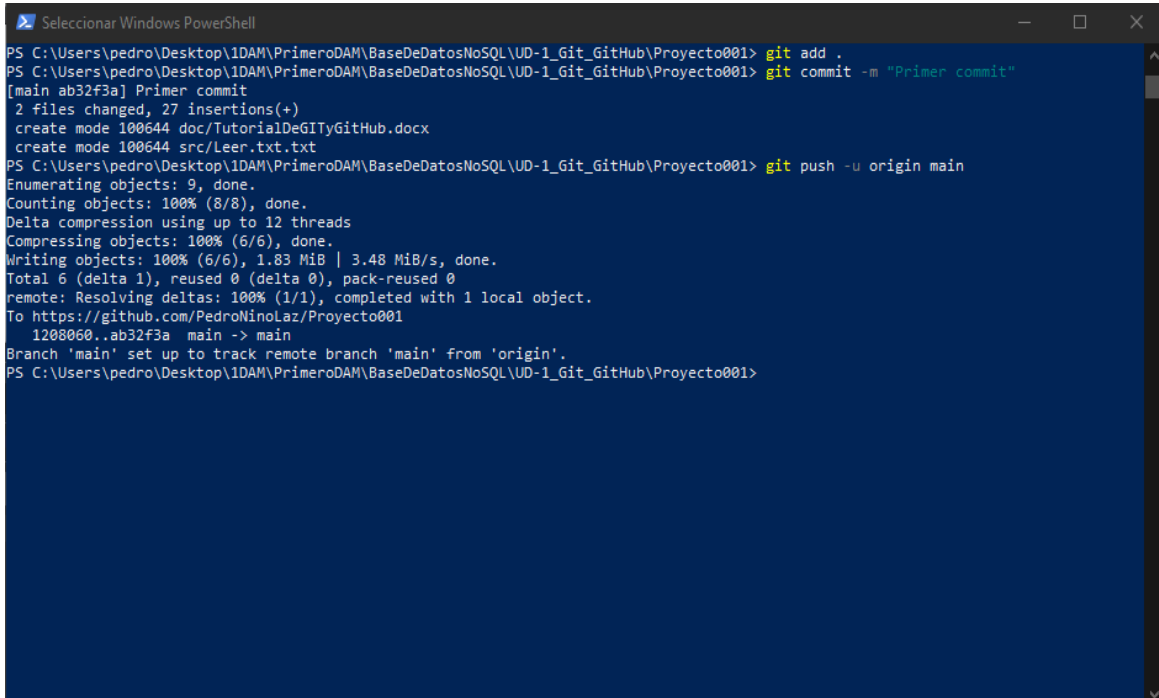
3.3. Sincronización de archivos a GitHub con PowerShell.

A continuación, vamos a utilizar los siguientes comandos para sincronizar nuestros archivos locales con el repositorio de GitHub, los comandos que vamos a usar son los siguientes:

- `git add .`
- `git commit -m "first commit"`
- `git push -u origin main`

Quedando el código de comandos de la siguiente manera para que sea más legible que en la imagen.

```
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git add .
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git commit -m "Primer commit"
[main ab32f3a] Primer commit
2 files changed, 27 insertions(+)
create mode 100644 doc/TutorialDeGITyGitHub.docx
create mode 100644 src/Leer.txt.txt
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.83 MiB | 3.48 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/PedroNinoLaz/Proyecto001
 1208060..ab32f3a  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001>
```

A screenshot of a Windows PowerShell terminal window. The title bar reads "Seleccionar Windows PowerShell". The terminal shows the following commands and output:
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git add .
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git commit -m "Primer commit"
[main ab32f3a] Primer commit
2 files changed, 27 insertions(+)
create mode 100644 doc/TutorialDeGITYGitHub.docx
create mode 100644 src/Leer.txt.txt
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.83 MiB | 3.48 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/PedroNinoLaz/Proyecto001
1208060..ab32f3a main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\pedro\Desktop\1DAM\PrimeroDAM\BaseDeDatosNoSQL\UD-1_Git_GitHub\Proyecto001>

Y con esto queda realizado el tutorial para empezar a utilizar las herramientas Git, GitHub y PoweShell.