

MANUAL DE GIT Y DE GITHUB

ÍNDICE

1.¿Qué es GIT y GITHUB?

2.¿Para qué sirven?

3.Instalación de GIT

4.Comprobación de la correcta instalación

5.Como hacer una cuenta de GITHUB

6.Creación del repositorio en línea de comando

7.Webgrafía

1.¿Qué es GIT y GITHUB?

Git es un sistema de control de versiones, Un sistema de control de versiones nos va a servir para trabajar en equipo de una manera mucho más simple y óptima cuando estamos desarrollando software.

Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador.

2.¿Para qué sirven?

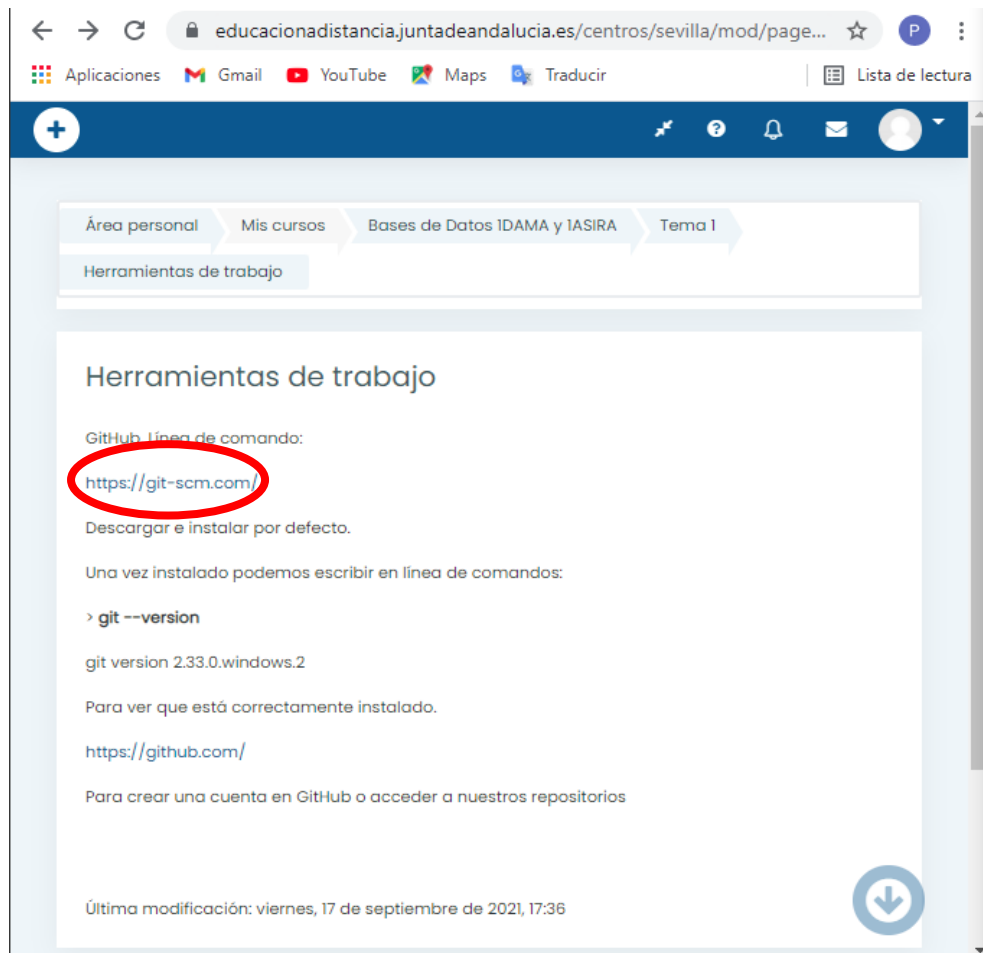
Git sirve para poder controlar todos los cambios que se realizan en nuestras aplicaciones y en los códigos, también vamos a tener control absoluto de todo lo que pasa en el código, pudiendo abrir diferentes ramas de desarrollo, etc. Además tendremos la posibilidad de trabajar con ramas de desarrollo, que nos van a permitir desarrollar cosas que divergen mucho del programa principal.

Unas de sus utilidades también es poder trabajar en equipo de una manera muy sencilla y optimizada, de forma que si tenemos dos o tres personas trabajando en ciertas funcionalidades del proyecto y nosotros podamos estar trabajando en nuestra parte del código. Nos proporciona también un listado (commits) y podemos volver atrás en el tiempo a cualquiera de esos cambios commints.

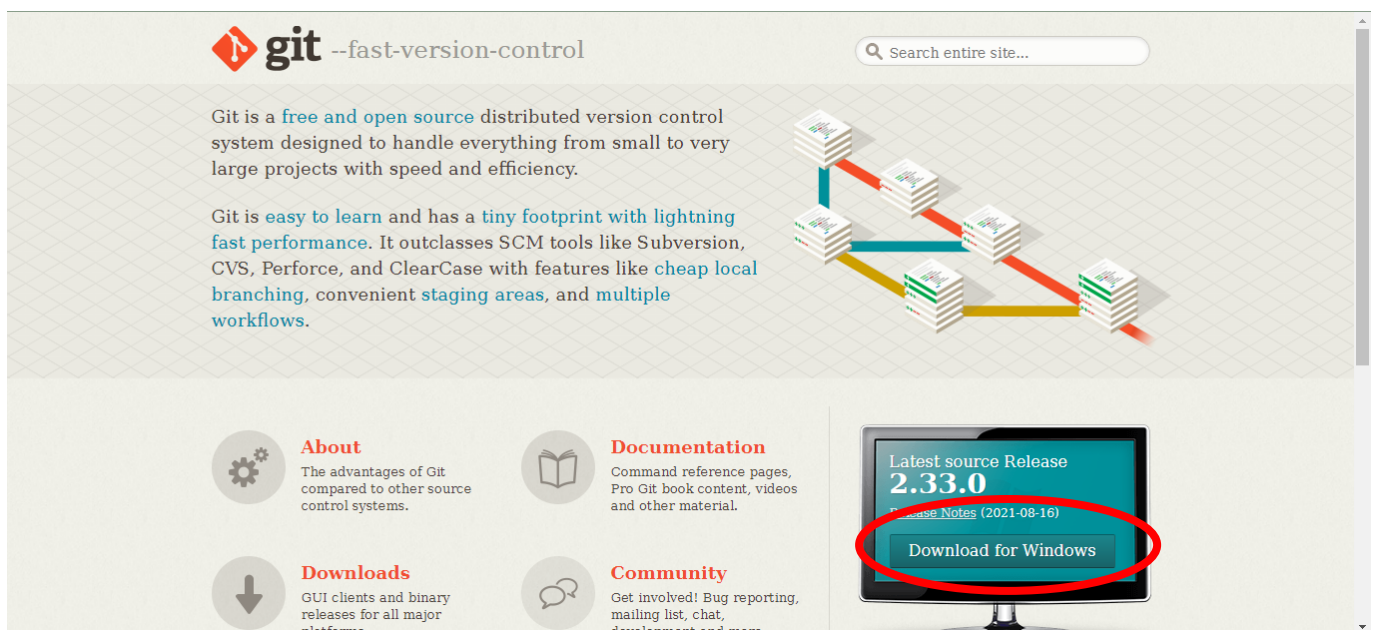
La plataforma de Github sirve para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

3.Instalación de GIT

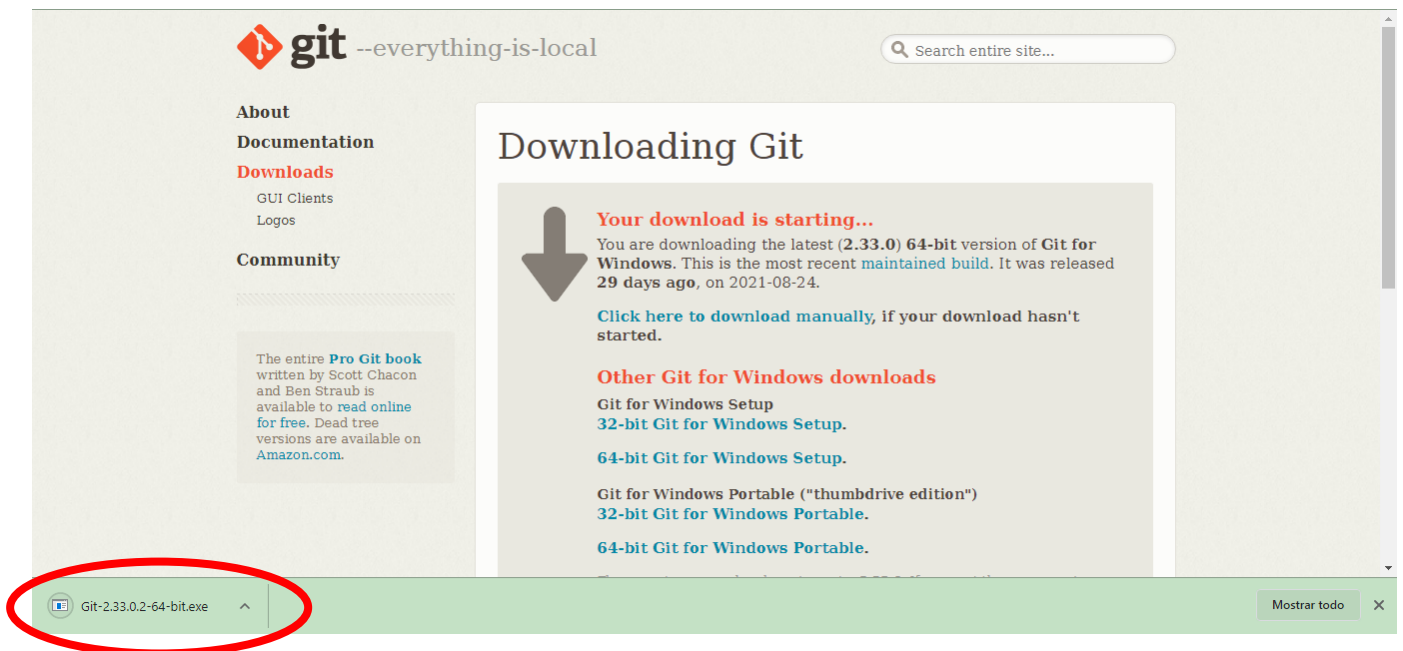
Para realizar la instalación de git tendríamos que dirigirnos a la moodle y seleccionar el enlace que te redirige directamente a la página de <https://git-scm.com/>



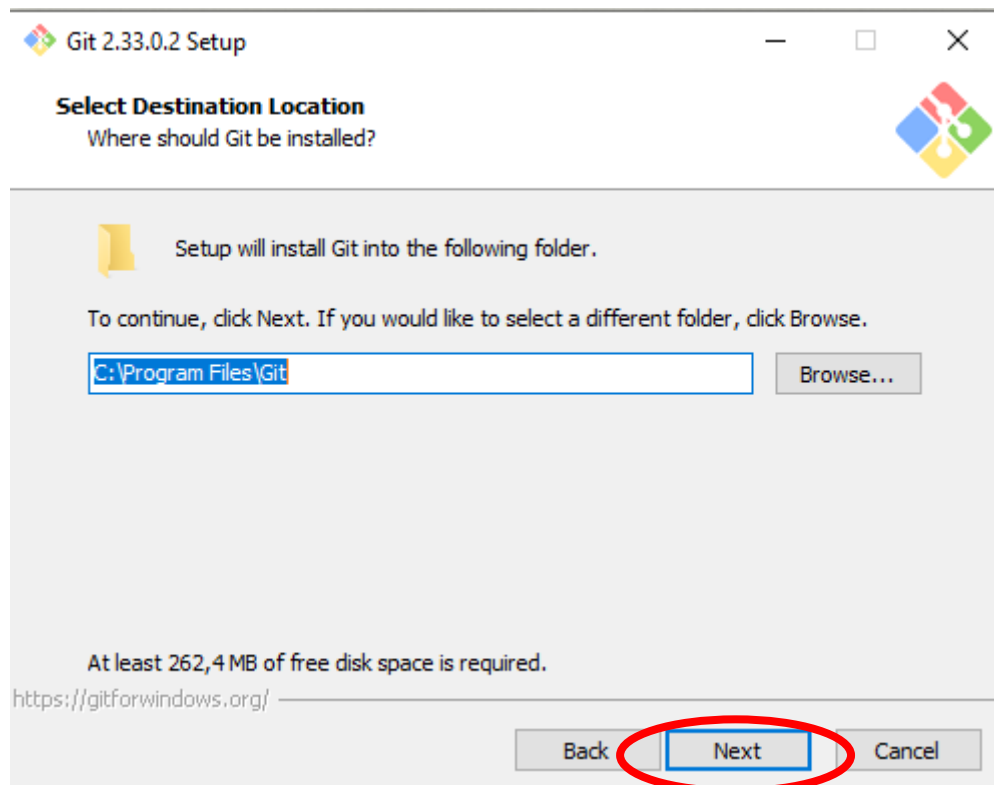
Una vez dentro de la página de GIT hacemos click en “Download for Windows” y se nos empieza a descargar el programa.



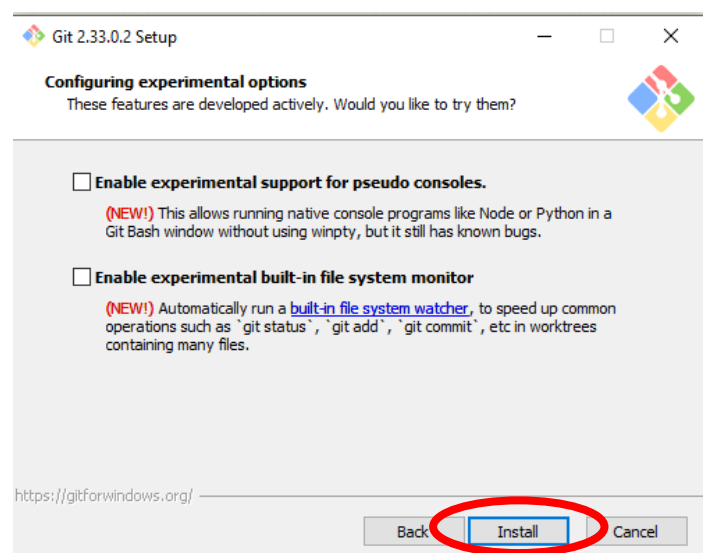
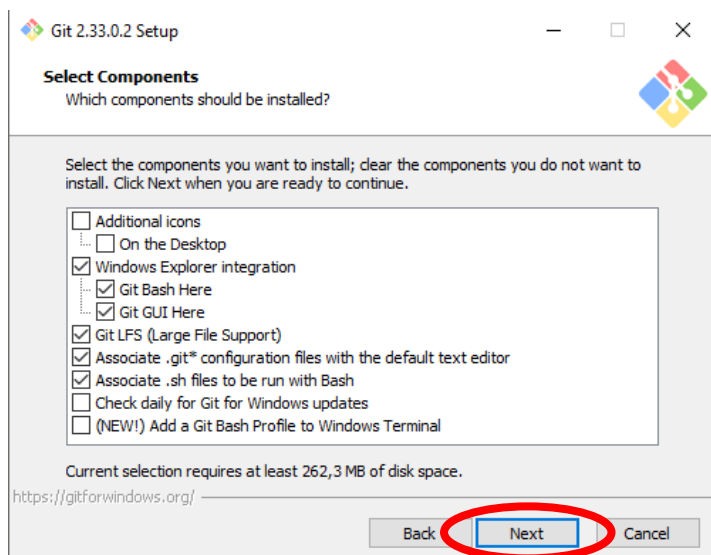
Cuando haya terminado la descarga el instalador hacemos click y se nos abre en una ventana externa donde le damos a “Next”.



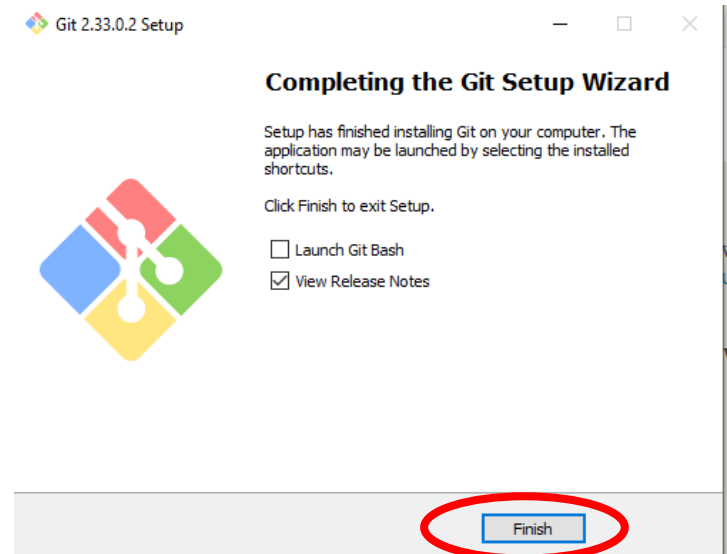
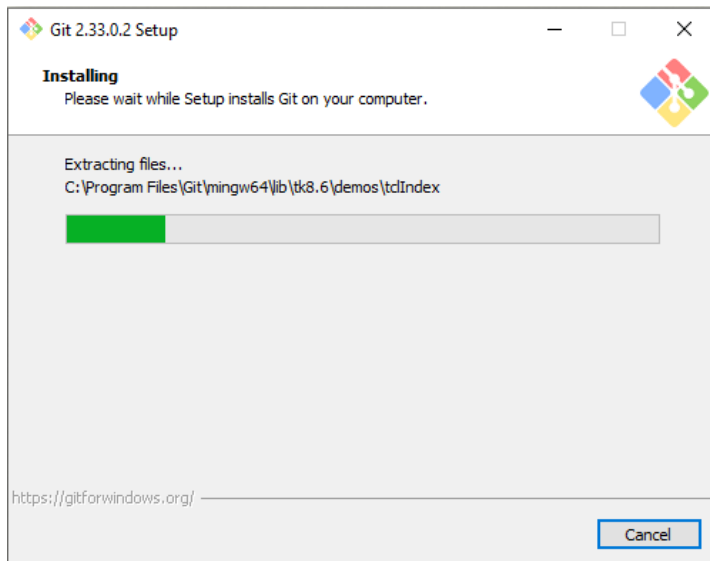
Aquí puedes elegir donde quieres que se descargue Git y una vez que lo elegimos le seguimos dando a “Next”.



A partir de este paso le seguimos dando a “Next” con las opciones ya seleccionada por defecto hasta llegar a la parte que nos dice “Install” y le damos. Hacemos esto ya que estamos GIT por defecto.

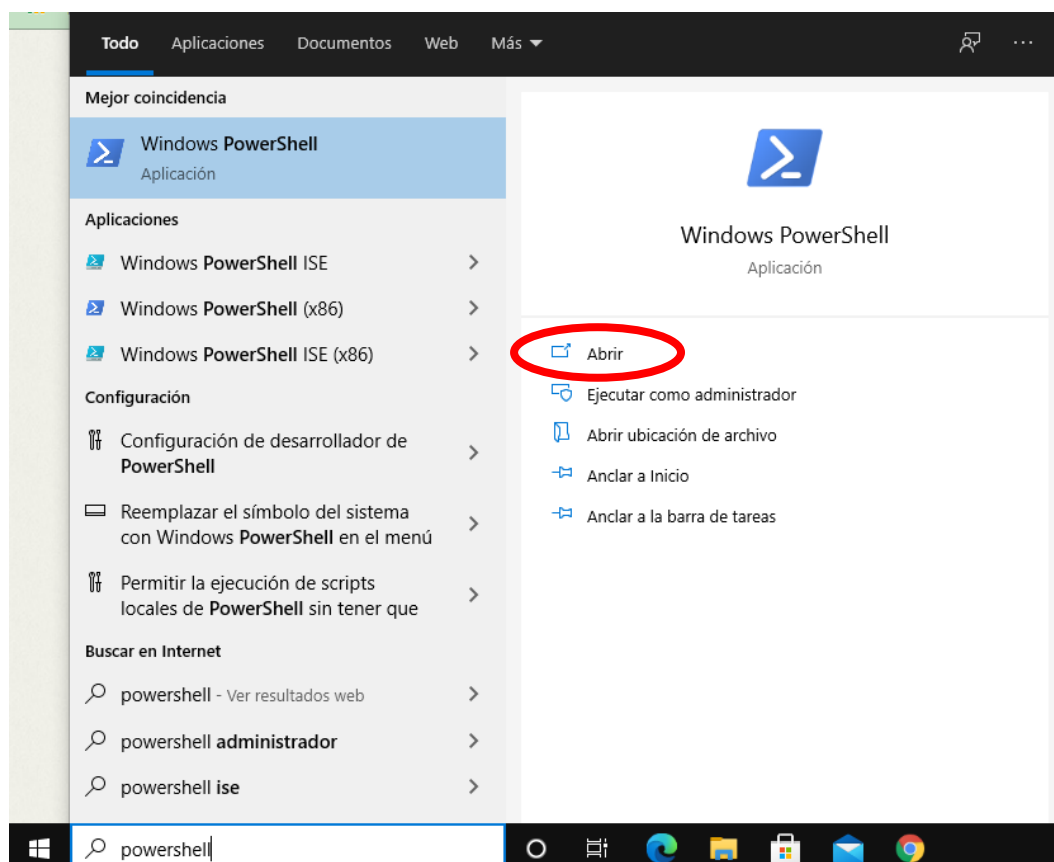


Aquí podemos observar que se está instalando y cuando termine le damos a “Finish” y ya estaría instalado

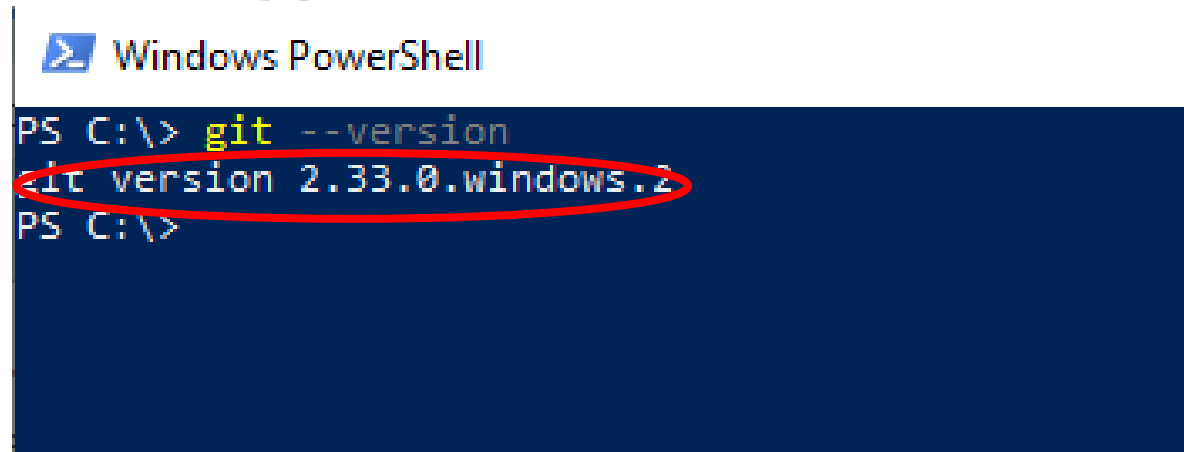


4. Comprobación de la correcta instalación y herramientas

Para comprobar que está instalado correctamente escribimos en el menú de windows Powershell y lo abrimos.



Y una vez dentro escribimos “git --version” y nos da la versión que está instalada en el equipo.

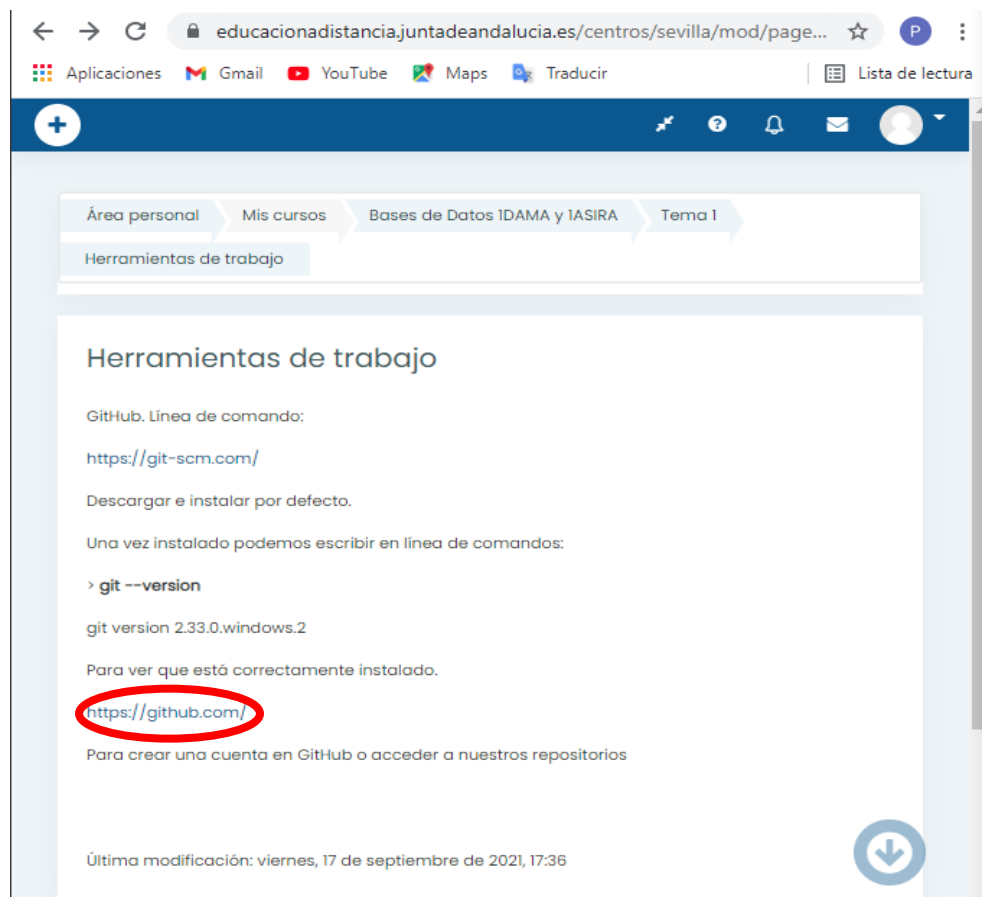


```
Windows PowerShell

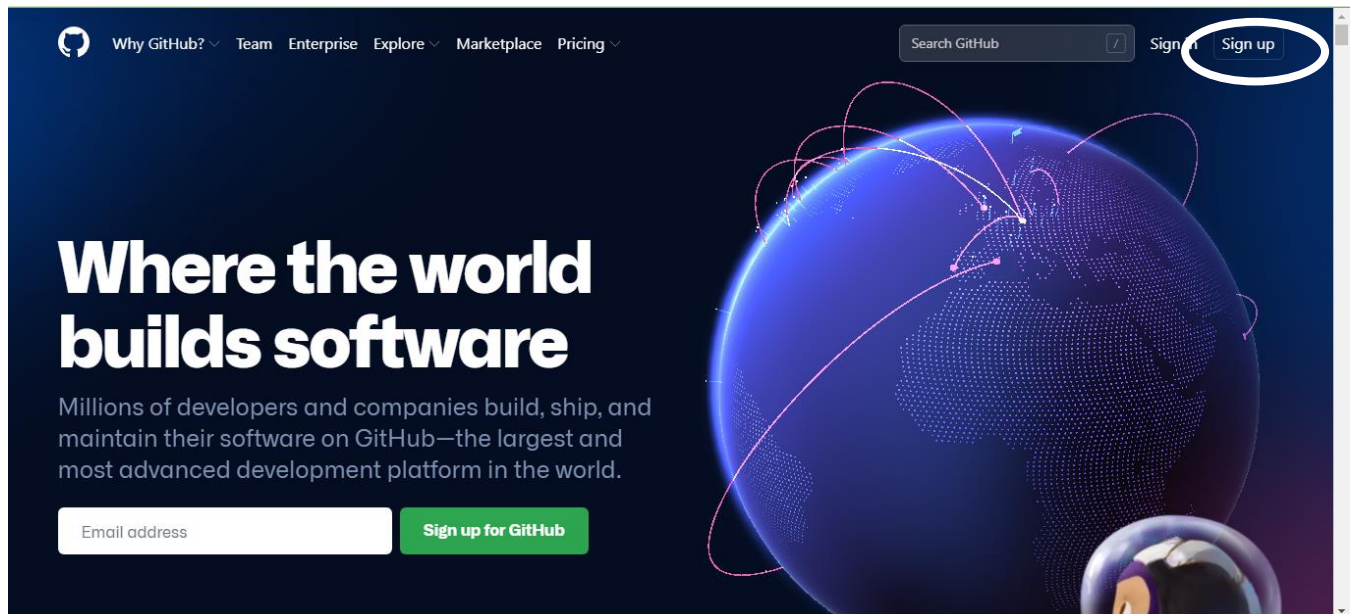
PS C:\> git --version
git version 2.33.0.windows.2
PS C:\>
```

5. Como hacer una cuenta de GITHUB

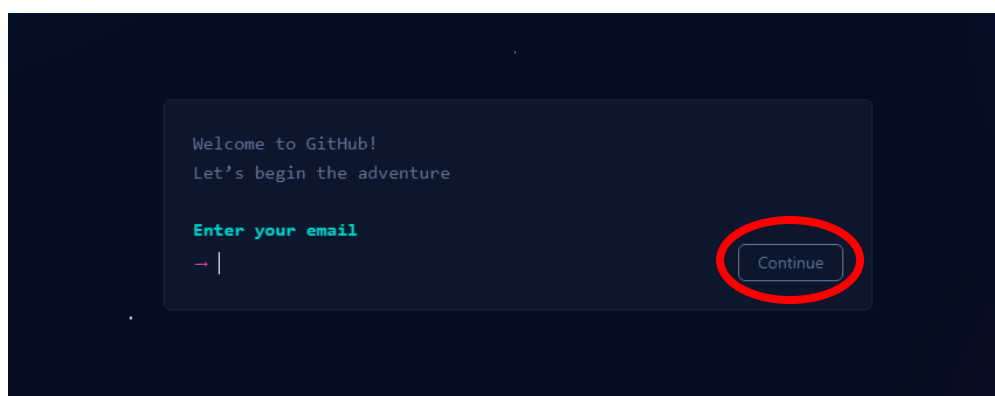
Lo primero que tenemos que hacer es dirigirnos a la moodle y hacer click en el enlace de GITHUB.



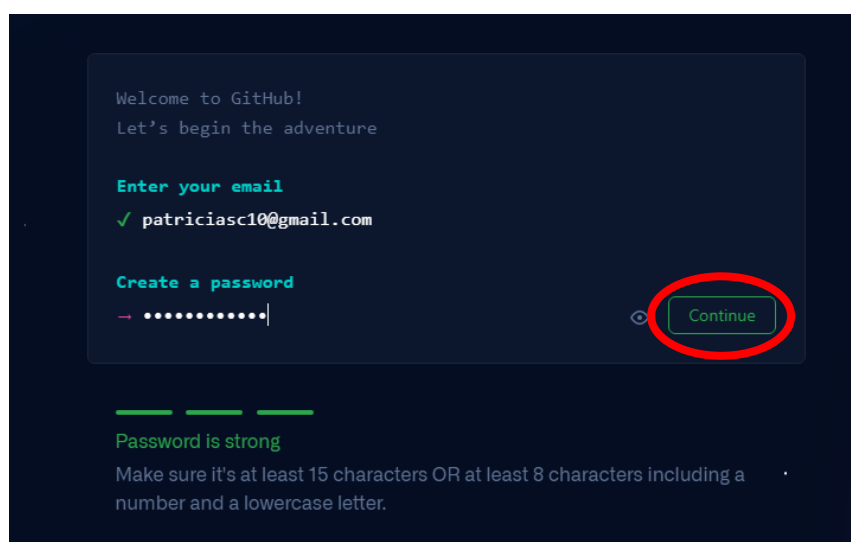
Dentro de la página web de Github hacemos click en “Sing up” para crearnos la cuenta.



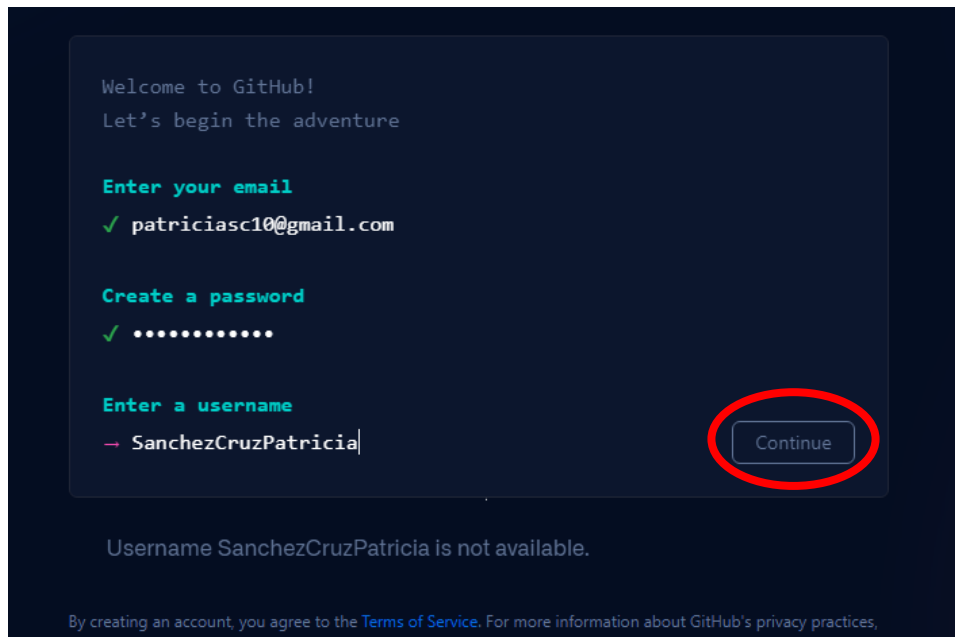
Lo que nos pide al principio es nuestro correo electrónico, lo introducimos y le damos a “Continue”



Al introducir nuestro correo electrónico nos pide la contraseña de la cuenta donde tienes que incluir números en ella y le volvemos a dar a “Continue”



Y por último nos pide el nombre de usuario que vamos a tener y de nuevo otra vez a “Continue”



Wellcome to GitHub!
Let's begin the adventure

Enter your email
✓ patriciasc10@gmail.com

Create a password
✓


Enter a username
→ SanchezCruzPatricia

Continue

Username SanchezCruzPatricia is not available.

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Policy](#).

A continuación, nos mandan un correo con un código para verificar la cuenta.

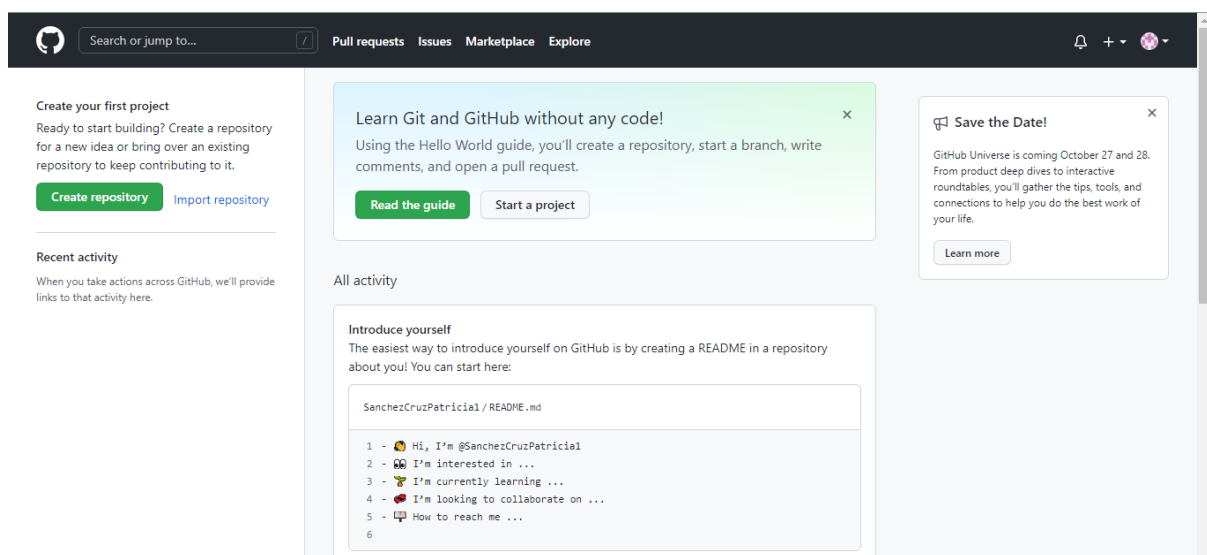


You're almost done!
We sent a launch code to patriciasc10@gmail.com

→ Enter code

Input fields for the verification code.

Y finalmente tenemos ya la cuenta en Github



Search or jump to... Pull requests Issues Marketplace Explore

Create your first project
Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.
[Create repository](#) [Import repository](#)

Recent activity
When you take actions across GitHub, we'll provide links to that activity here.

Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.
[Read the guide](#) [Start a project](#)

Save the Date!
GitHub Universe is coming October 27 and 28. From product deep dives to interactive roundtables, you'll gather the tips, tools, and connections to help you do the best work of your life.
[Learn more](#)

All activity

Introduce yourself
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

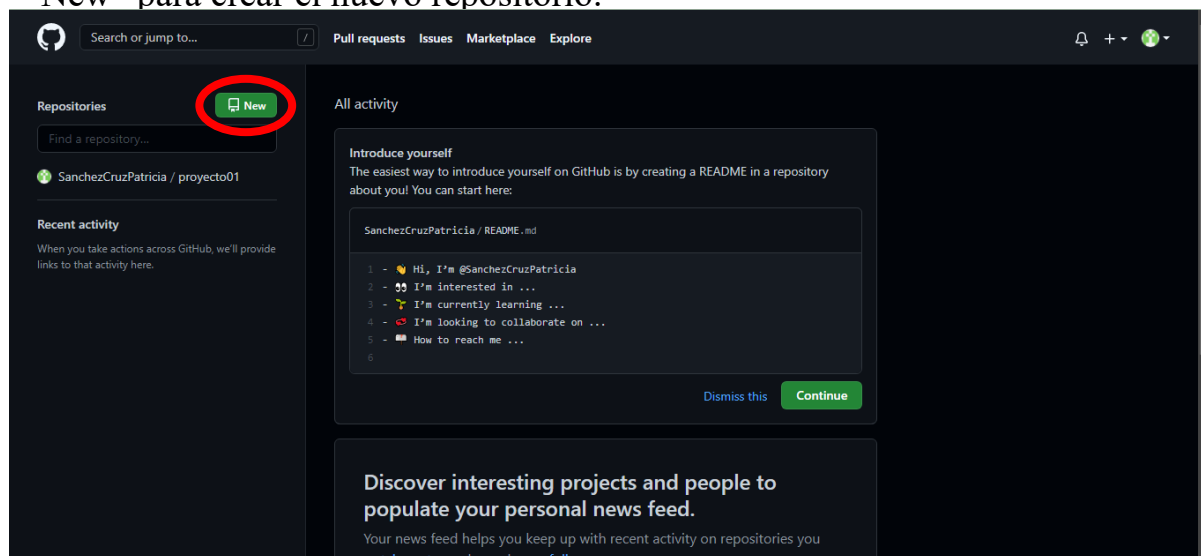
SanchezCruzPatricia / README.md

```
1 - 🙋 Hi, I'm @SanchezCruzPatricia
2 - 🗨️ I'm interested in ...
3 - 📖 I'm currently learning ...
4 - 🤝 I'm looking to collaborate on ...
5 - 📍 How to reach me ...
6
```

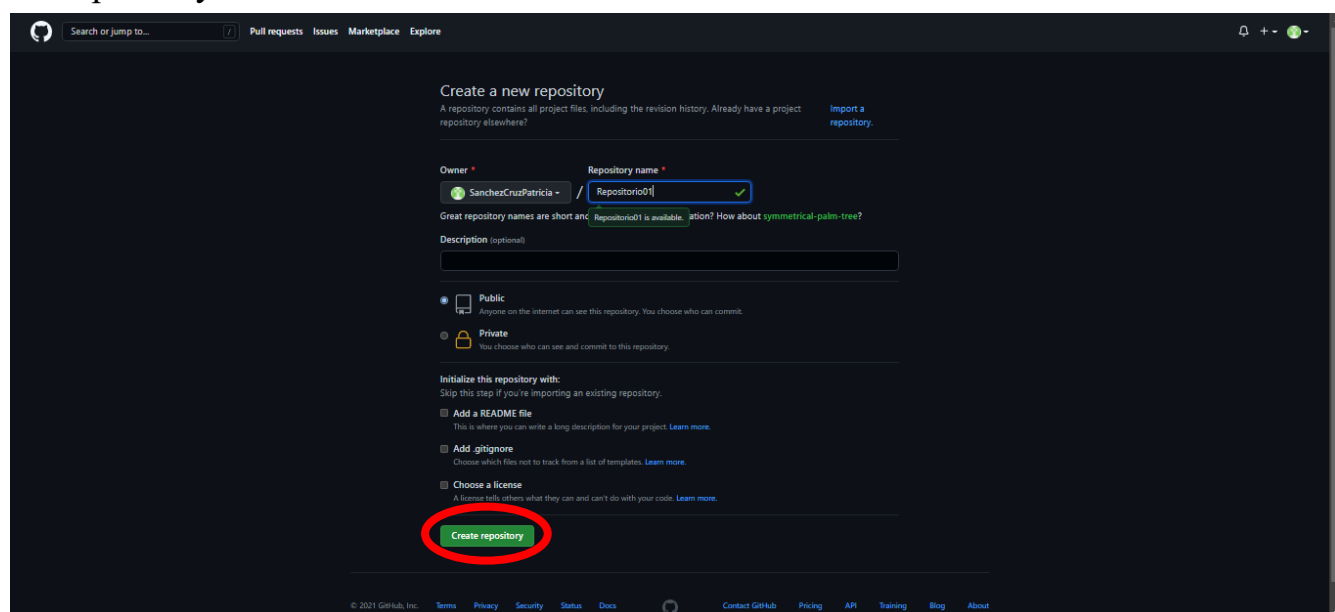
6. Creación del repositorio en línea de comando

En este apartado vamos a ver la creación del repositorio a través de Powershell y las herramientas que vamos a utilizar para ello.

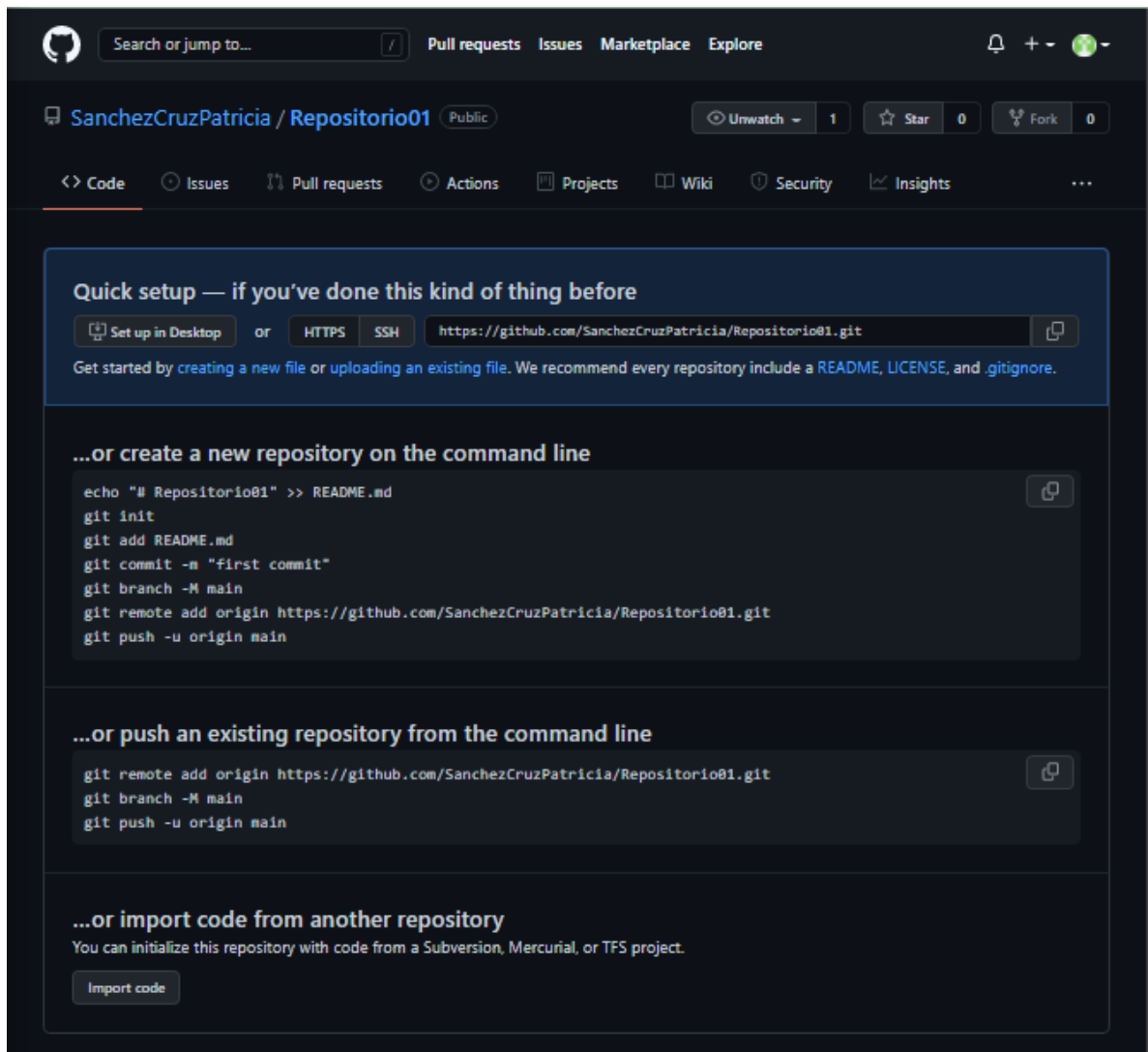
Primero nos tenemos que dirigir a nuestra cuenta de Github donde le damos a “New” para crear el nuevo repositorio.



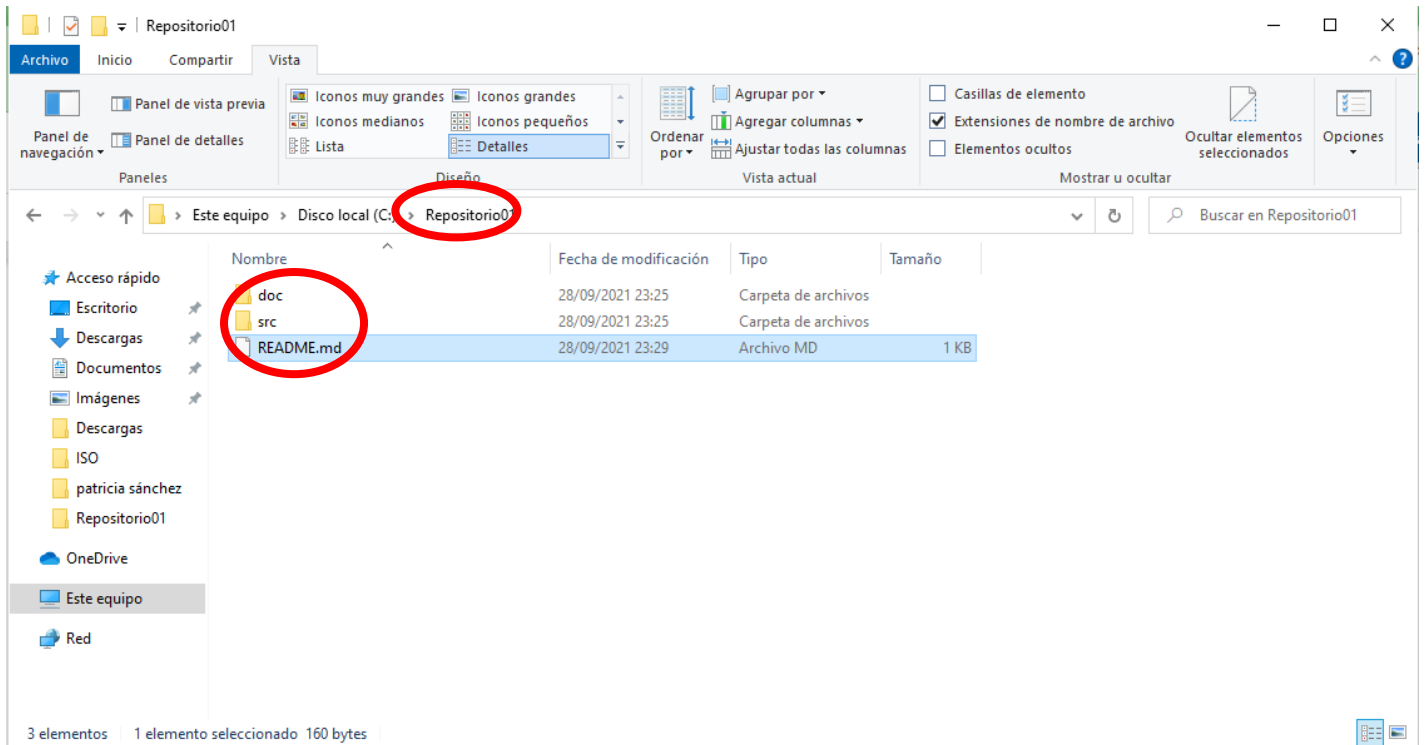
Al darle nos aparece la configuración del nuevo repositorio donde vamos a escribir el nombre y lo dejamos por defecto. Al terminar le damos a “Create repository”



Al crearlo nos aparecen comandos para poder enlazar nuestro repositorio con nuestro equipo. Estos comandos son las herramientas que vamos a utilizar.



Antes de dirigirnos a Powershell para utilizar las herramientas, vamos a crear las carpetas que vamos a subir a nuestro repositorio. Como podemos ver he creado una carpeta llamada como el repositorio de Github y dentro de él las carpetas que voy a subir y un archivo llamado “README.md” donde esta la explicación resumida de lo que va haber en el repositorio.



Una vez creado ya nos podemos dirigir a Powershell. Lo primero que vamos a hacer es dirigirnos a nuestra carpeta con el comando “cd” y la dirección de ella.

```
Windows PowerShell
PS C:\> git --version
git version 2.33.0.windows.2
PS C:\> cd C:\Repositorio01
PS C:\Repositorio01>
```

A continuación, introducimos “git init” para que inicie Git en la dirección de la carpeta que hemos creado.

```
Windows PowerShell
PS C:\> git --version
git version 2.33.0.windows.2
PS C:\> cd C:\Repositorio01
PS C:\Repositorio01> git init
Initialized empty Git repository in C:/Repositorio01/.git/
PS C:\Repositorio01>
```

Después de lo anterior escribimos “git add README.md”, “git add src” y “git add doc” para agregar los archivos al nuevo repositorio local

```
PS C:\Repositorio01> git add README.md
PS C:\Repositorio01> git add src
PS C:\Repositorio01> git add doc
PS C:\Repositorio01>
```

Luego introducimos el comando "git commit -m "mi primer repositorio """. Commit es un comando que va creando distintas versiones del proyecto. Conforme vayamos avanzando, creando nuevos archivos y mejorando el proyecto, podemos ir guardando las distintas versiones en github, para esto será necesario que hagamos un commit cada vez que queramos crear una nueva versión.

```
PS C:\Repositorio01> git commit -m "mi primer repositorio"
[main (root-commit) e8fd86a] mi primer repositorio
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Ahora, escribimos "git branch -M main", con este comando puedes crear la rama donde se va a almacenar todo lo que subamos.

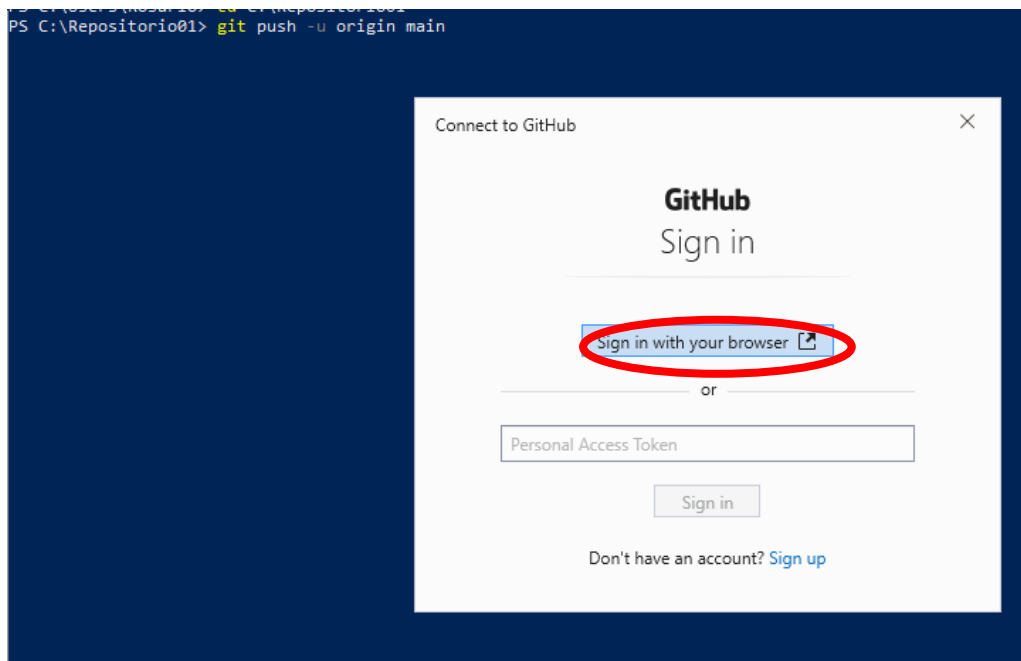
```
PS C:\Repositorio01> git branch -M main
```

Después, es necesario ir al repositorio git (la carpeta del proyecto) y estando dentro pegar el siguiente código.

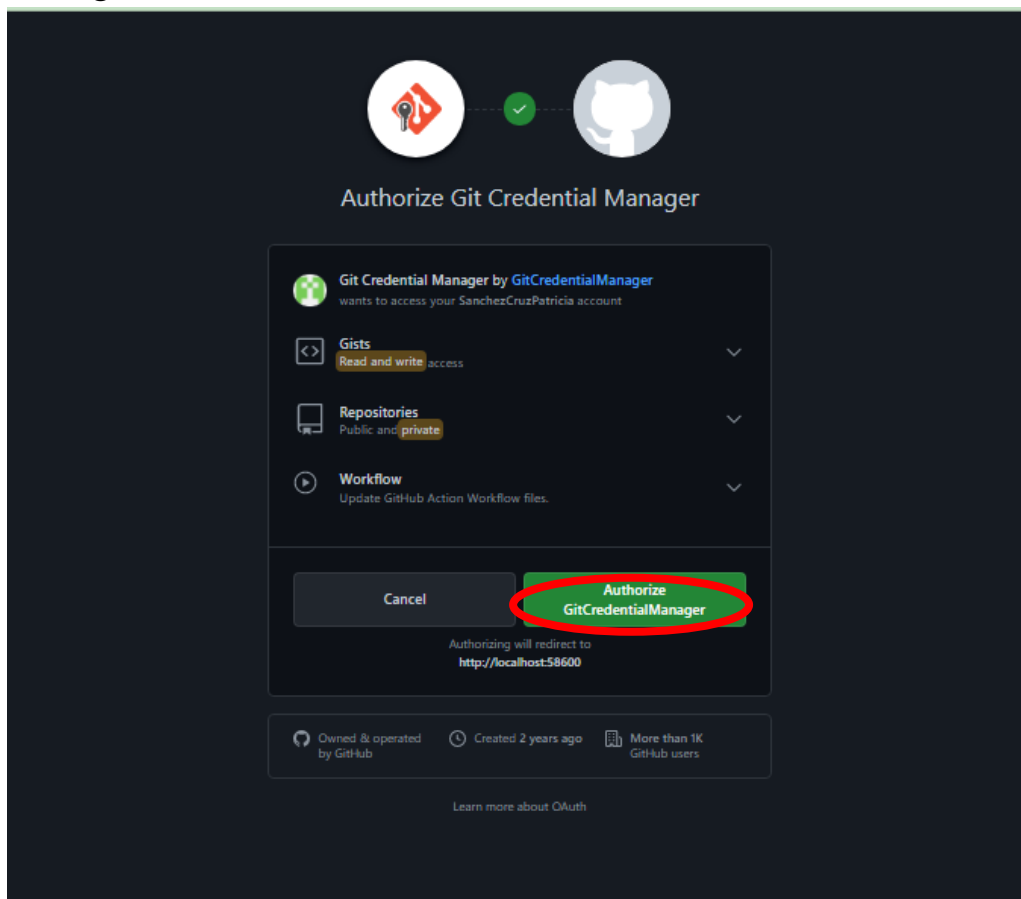
```
PS C:\Repositorio01> git remote add origin https://github.com/SanchezCruzPatricia/Repositorio01.git
```

A continuación ,escribimos "git push -u origin main" te permite subir los commits desde tu rama (branch) local en tu repositorio git local al repositorio remoto. Para poder subir a tu repositorio remoto, te debes asegurar de hacer commit a todos tus cambios al repositorio local.

Al introducir el comando nos aparece una pantalla externa donde tenemos que hacer click en "Sign in with your browser" para poder emparejar nuestra cuenta de Github con el Git local.

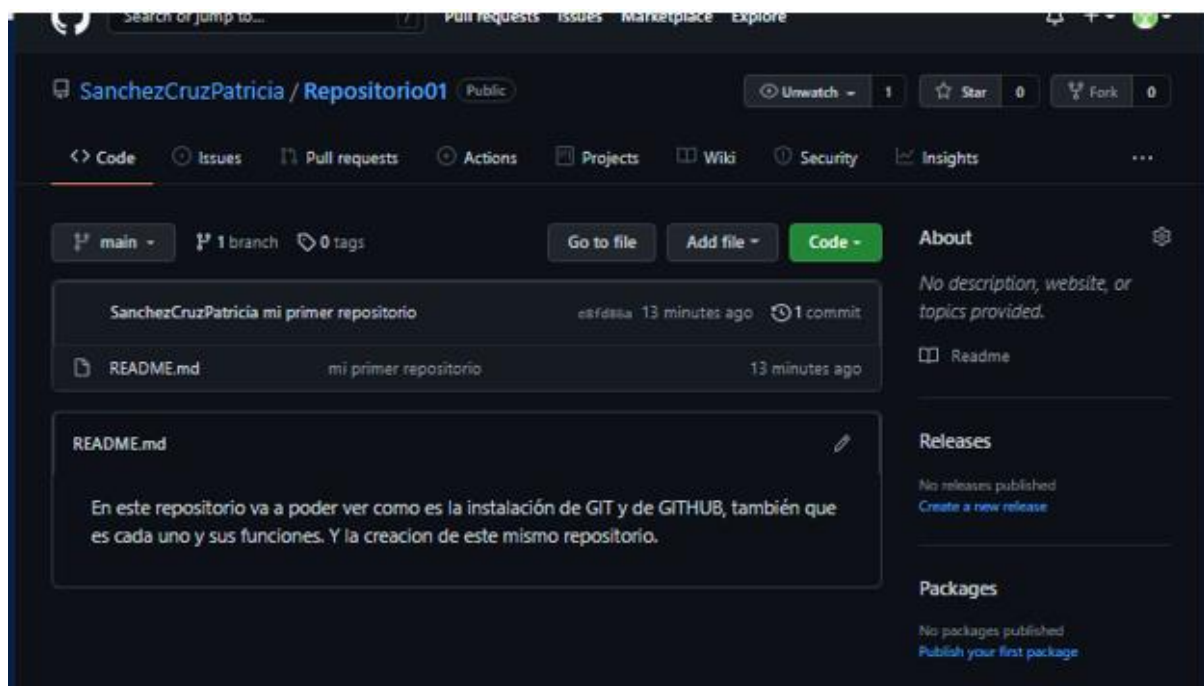


Al hacer click en lo anterior se nos abre el navegador donde lo único que tenemos hacer es autorizar el vínculo dándole en “Authorize Get Credential Manager”



Y por último nos aparecen las comprobaciones de la subida de los archivos a Github ya al entrar en Github nos aparece.

```
PS C:\Repositorio01> git push -u origin main
info: please complete authentication in your browser...
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), 353 bytes | 176.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SanchezCruzPatricia/Repositorio01.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Repositorio01>
```



7. Webgrafía

- <https://victorrobleweb.es/2018/04/28/que-es-git-y-para-que-sirve/>
- <https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>