

# GITHUB

## ÍNDICE

- 1.- INTRODUCCIÓN
- 2.- INSTALACIÓN GIT
- 3.- CREAR CUENTA GITHUB
- 4.- EJERCICIO DE SUBIDA DE PROYECTO A GITHUB

## 1. INTRODUCCIÓN

GitHub es un servicio basado en la nube que permite gestionar proyectos almacenados en repositorios.

Para establecer conexión entre nuestro equipo y GitHub necesitamos instalar Git que es un sistema de control de versiones distribuido. Los comandos de Git que vamos a utilizar son:

-git init → crea un repositorio local.

-git add → agrega los archivos al repositorio.

-git status → muestra el estado de los archivos.

-git commit → guarda los cambios en el repositorio. Con -m añadimos una etiqueta.

-git push → para subir al repositorio

-git pull → fusiona los cambios que se han hecho en el repositorio remoto con el directorio local.

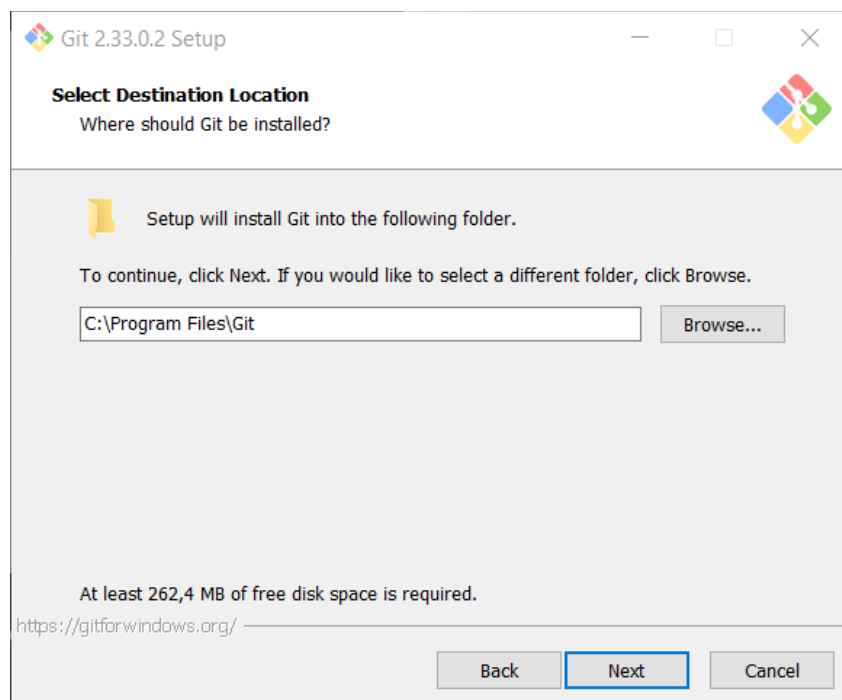
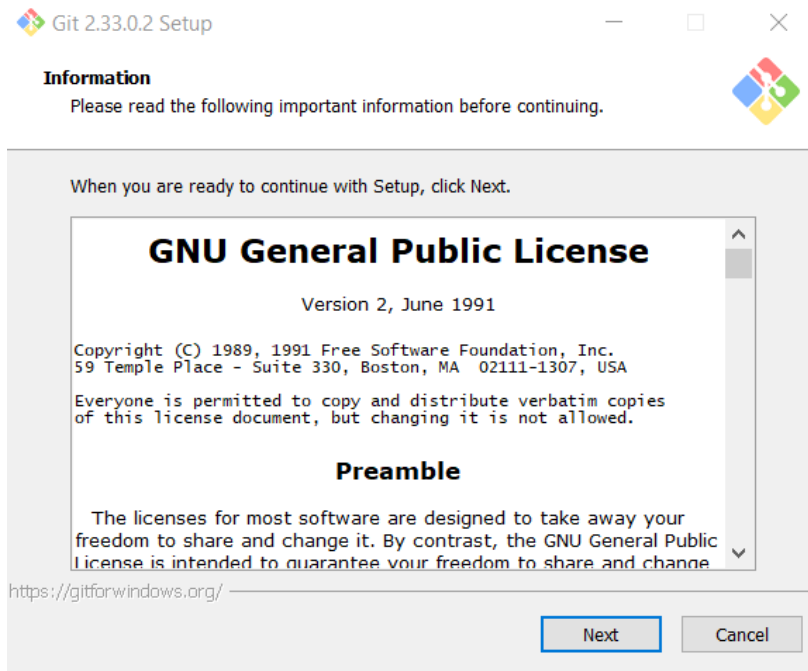
-git clone → para clonar

## 2. INSTALACIÓN GIT

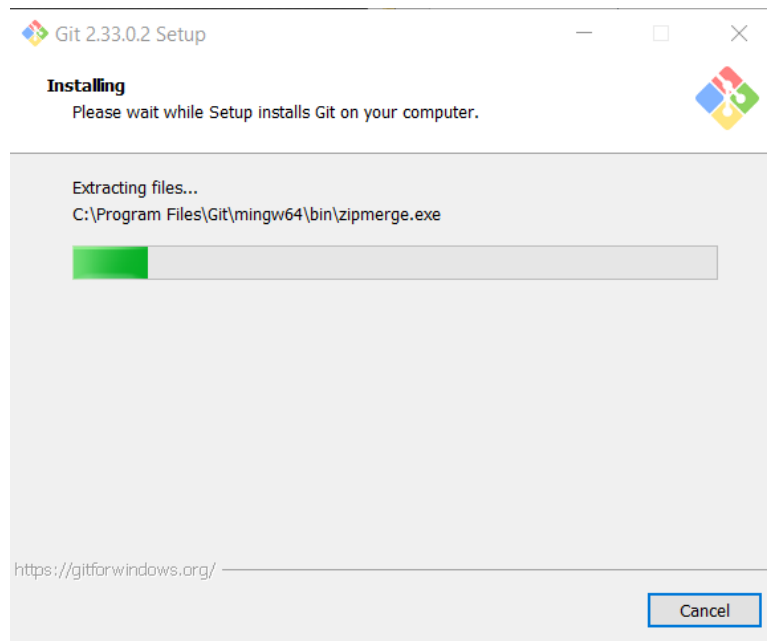
Para instalarlo accedemos a la página <https://git-scm.com/>



Y buscamos la última versión para descargar. Una vez descargada procedemos a instalarla.



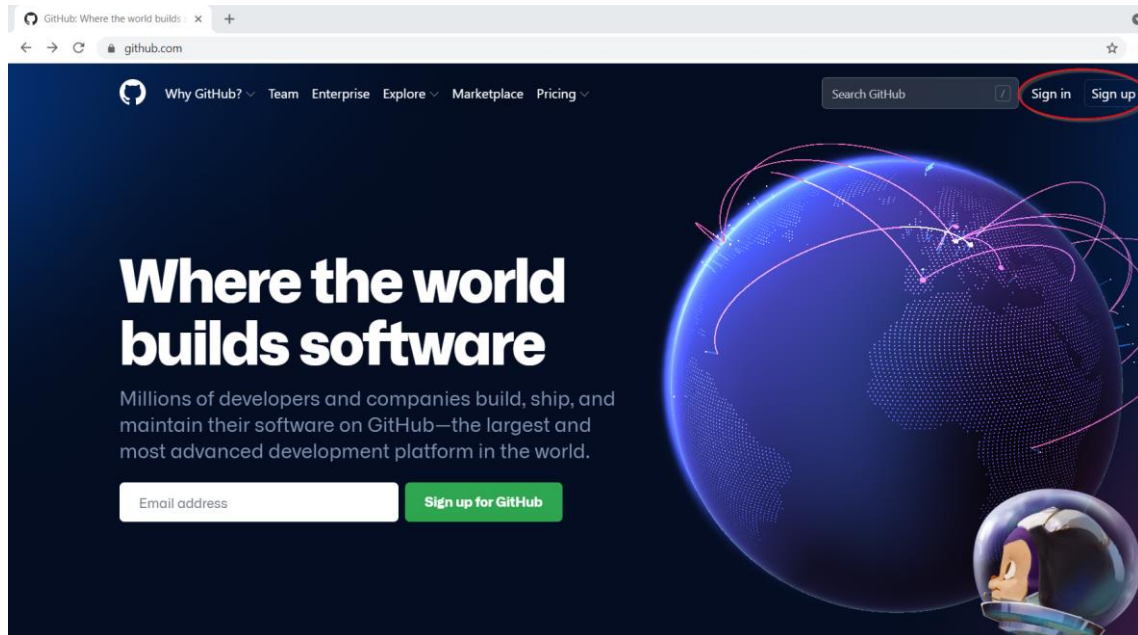
En nuestro caso la instalación va a ser con la configuración por defecto por lo que no hay que realizar ningún cambio y solo dar "Next" hasta que comience la instalación.



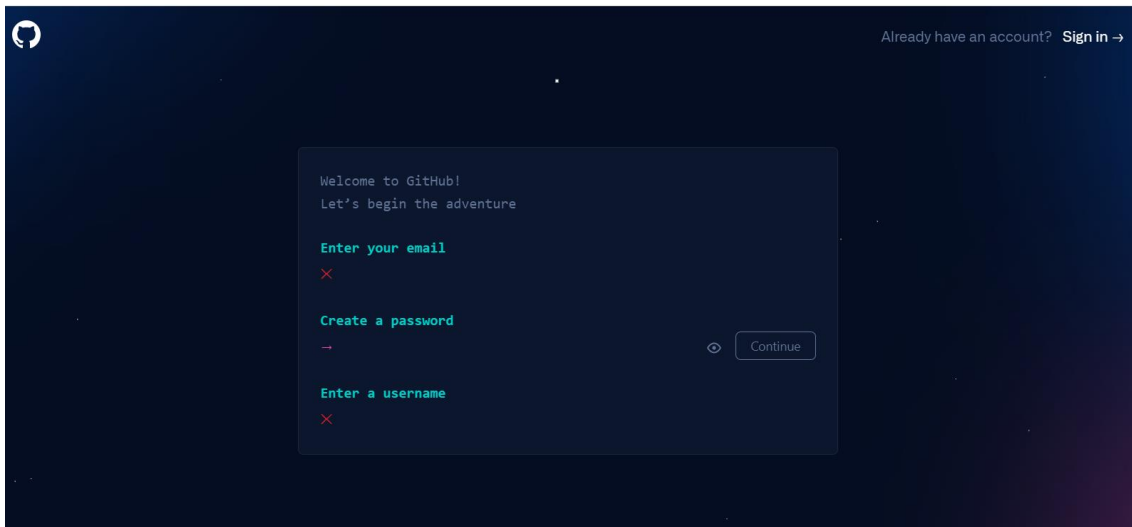
Una vez finalizada ya tenemos instalado Git.

### 3. CREAR CUENTA GITHUB

Vamos a crear una cuenta en GitHub. Para ello debemos ir a <https://github.com/> y nos dirigimos a "Sign up". Una vez creada la cuenta accederemos con "Sign in".

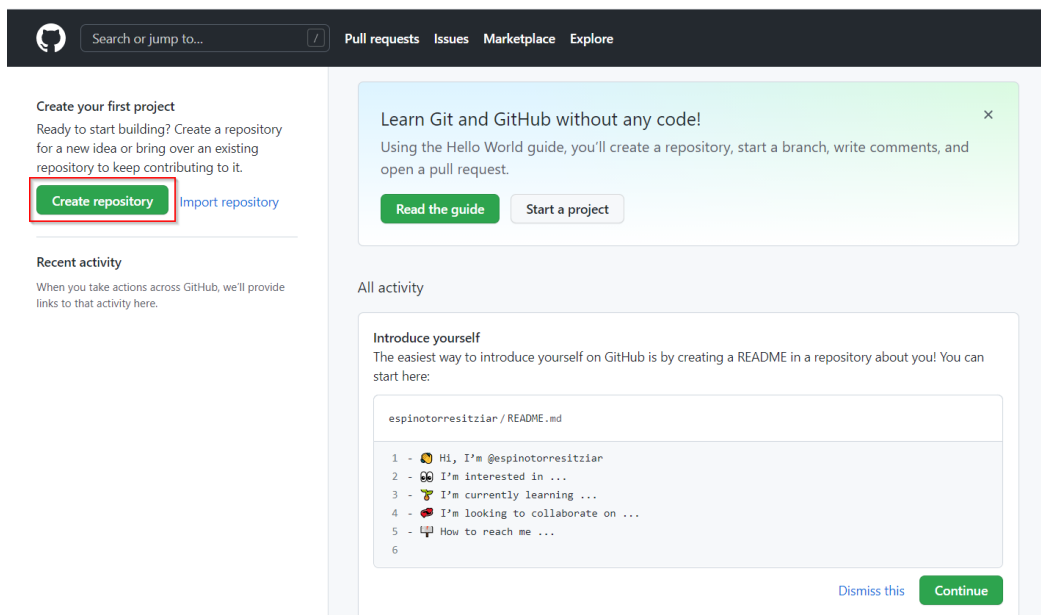


Para crear la cuenta lo que nos piden es un correo electrónico, un nombre de usuario y una contraseña.



Una vez verificada por email, la cuenta ya está creada y podemos acceder a ella.


Lo primero que vamos a hacer en nuestro GitHub es crear un repositorio.



## Create a new repository

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

Owner \*

 espinotorresitziar ▾

Repository name \*

/ prueba01 ✓

Great repository names are short and memorable. Need inspiration? How about [symmetrical-octo-parakeet?](#)

Description (optional)

☒  **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

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

Overview


**Repositories** 1

Projects

Find a repository...


**prueba01** Public

Updated 2 minutes ago



espinotorresitziar

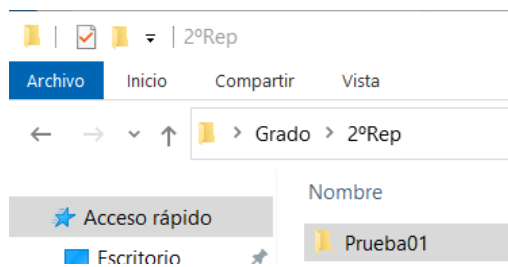
Edit profile

 Joined yesterday

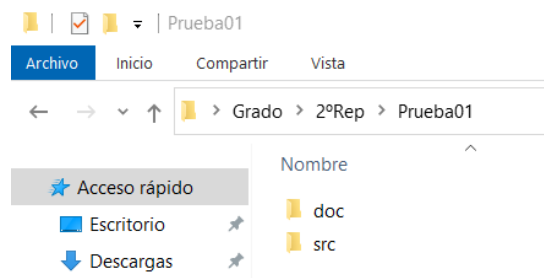
Cuando accedemos a nuestro repositorio nos aparecen una serie de comandos que son los que debemos usar para conectar nuestro equipo con GitHub y poder subir a este nuestro proyecto.

#### 4. EJERCICIO DE SUBIDA DE PROYECTO A GITHUB

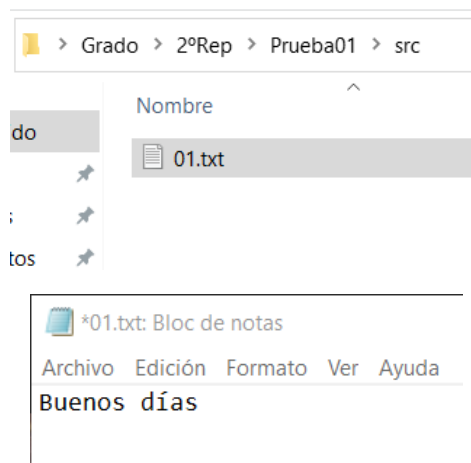
Creamos una carpeta del proyecto.



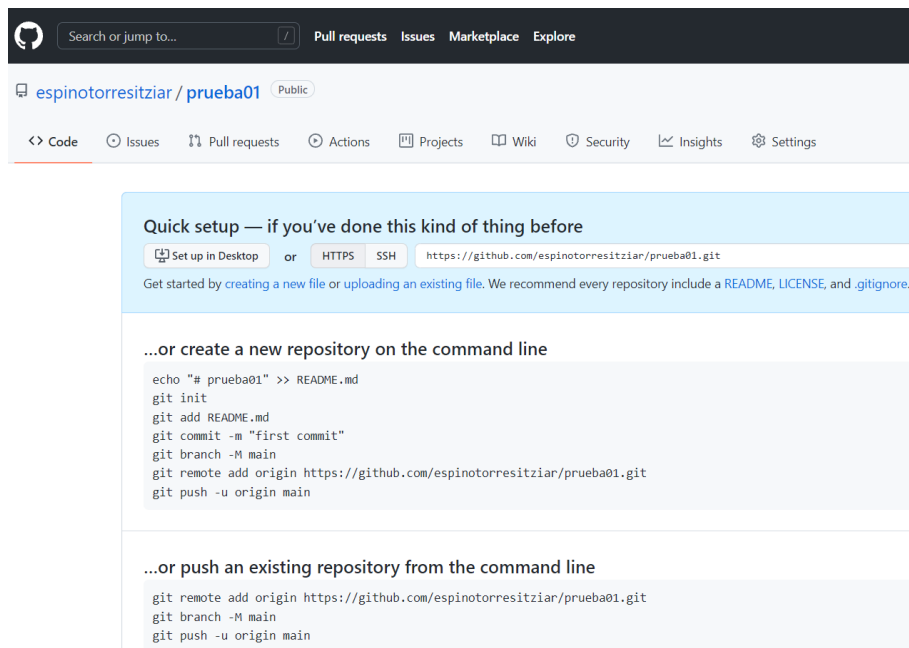
Dentro de ella creamos otras dos carpetas. Una carpeta “doc” en la que irá la documentación sobre Git y GitHub, y una carpeta “src” con los archivos.



En la carpeta src creamos un archivo con el contenido que queramos.



Accedemos a GitHub desde el navegador y entramos en el repositorio en el que queremos subir el proyecto. Al entrar, nos aparecen una serie de comandos que son los que debemos utilizar para subirlo.



Abrimos cmd y nos colocamos en la carpeta en la que está lo que queremos subir.

```
C:\> Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd C:\Users\torre\Desktop\Grado\2ºRep\Prueba01

C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>
```

Primero utilizamos “git init” para crear el repositorio local.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git init
Initialized empty Git repository in C:/Users/torre/Desktop/Grado/2ºRep/Prueba01/.git/
```

Luego debemos utilizar “git add .” para añadir al repositorio lo que hay en la carpeta. Y para ver que se ha añadido empleamos “git status”.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git add .
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git status
On branch master

No commits yet

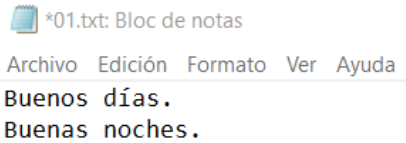
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   src/01.txt
```

Nos aparece que se ha añadido la carpeta con el archivo que hay en su interior.

Si ahora creamos un nuevo archivo o editamos el que acabamos de subir hay que volver a utilizar “git add .” para que se vuelva a añadir con los cambios. Realizamos una prueba:

-Primero vamos a editar el archivo “01.txt”.





Usamos "git status".

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/01.txt

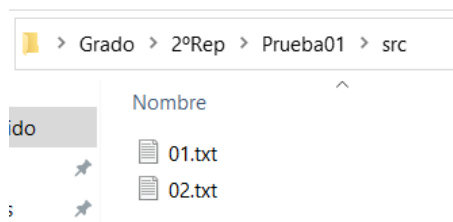
no changes added to commit (use "git add" and/or "git commit -a")
```

Nos informa de que el archivo ha sido modificado y debemos volver a añadirlo.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git add .

C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/01.txt
```

-Ahora vamos a crear otro archivo en la carpeta.



Ahora, con el comando "git status" nos informa de lo que hay añadido al repositorio y lo que no.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/01.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/02.txt
```

Por lo que debemos añadir lo nuevo que hemos creado.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git add .

C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/01.txt
        new file:   src/02.txt
```

Empleamos “git commit -m” seguido del nombre de etiqueta que queramos poner.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git commit -m "primer commit"
[master f5217fe] primer commit
2 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 src/02.txt
```

El siguiente comando a utilizar es “git branch -M main”.

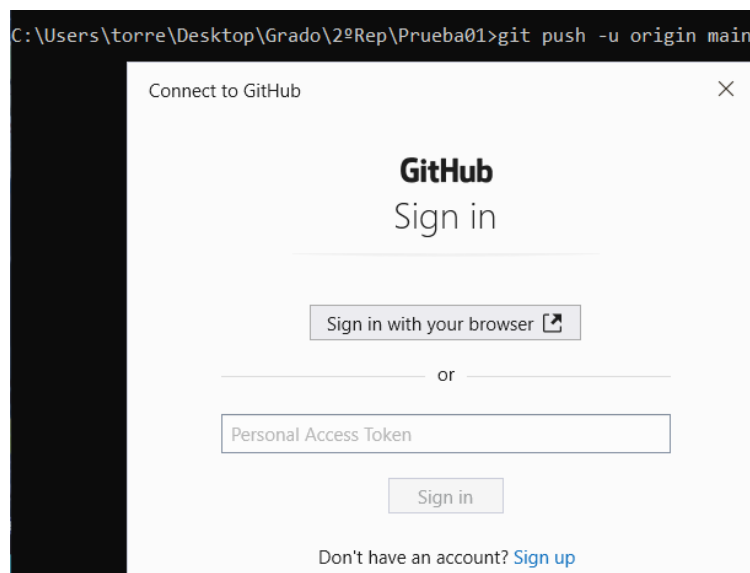
```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git branch -M main
```

Ahora vamos a añadir el repositorio de GitHub al que vamos a subir el proyecto, empleando “git remote add origin” seguido de la url del repositorio.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git remote add origin "https://github.com/espinatorresitziar/prueba01.git"
```

Y por último subimos al repositorio con “git push -u origin main”.

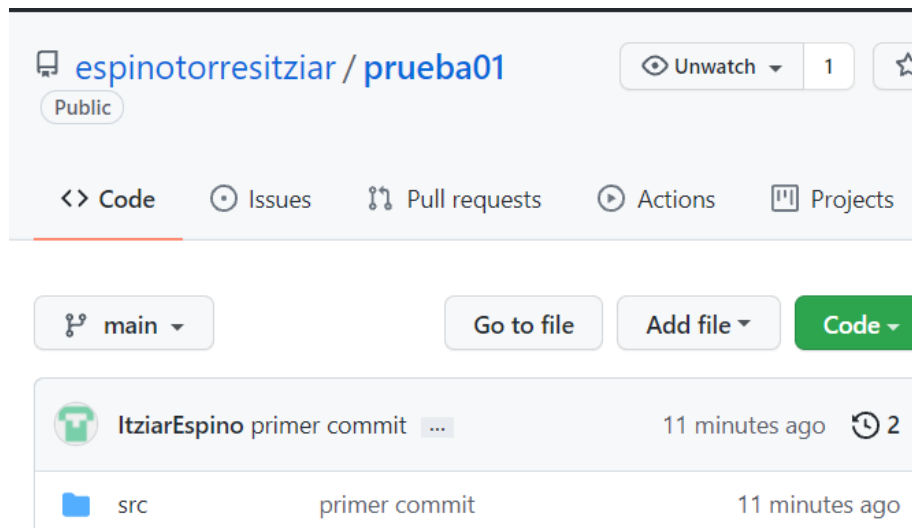
Si es la primera vez que subimos a GitHub al utilizar este comando solicita un inicio de sesión para poder subir el proyecto a tu GitHub.



Cuando iniciamos sesión volvemos a cmd y podemos ver que se ha completado la subida.

```
C:\Users\torre\Desktop\Grado\2ºRep\Prueba01>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 596 bytes | 198.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/espinatorresitziar/prueba01.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Vemos en GitHub que se ha subido correctamente.



Ahora lo que debemos hacer es subir la documentación, que deberemos introducirla en la carpeta “doc”. Para subirla realizamos utilizamos los mismos comandos desde cmd:

-git add .

-git commit -m

-git branch -M main

-git remote add origin

-git push -u

En GitHub creamos un archivo README.md en el que explicamos de lo que trata nuestro proyecto.



Si en nuestro proyecto hay algún archivo que no queremos que se suba a GitHub, creamos .gitignore en el que debemos escribir el nombre del archivo y este no se subirá con el proyecto.