

# INSTALACION Y CONFIGURACION DE UN REPOSITORIO EN GITHUB



Miguel García Holgado  
Gestión de Base de Datos

# INDICE

## 1. Preparando el entorno local

---

- 1.1 Instalación del paquete git
- 1.2 Creación del directorio a subir

## 2. Preparando entorno en github

---

- 2.1 Crear cuenta de github
- 2.2 Crear un repositorio

## 3. Subiendo nuestros archivos

---

- 3.1 Usar la powershell para subir un archivo

## 4. Comandos en git

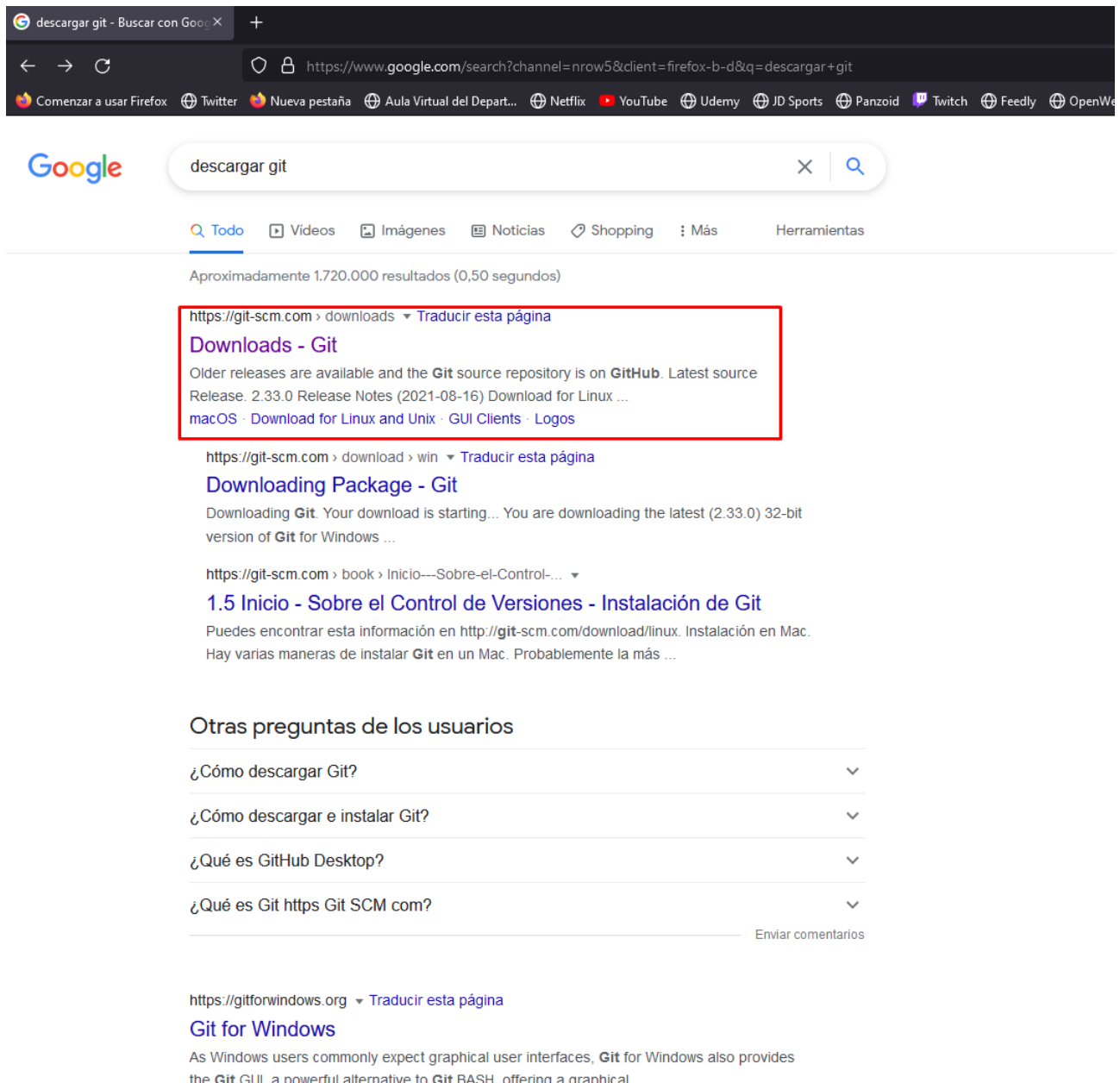
---

- 4.1 Comandos principales
- 4.2 Comando ignore
- 4.3 Comando diff
- 4.4 Comando clone
- 4.5 Comando pull

# 1. Preparando el entorno local

## 1.1 Instalación del paquete git

Primero empezaremos yendo al navegador y buscaremos: “Descargar Git”



The screenshot shows a web browser window with the address bar displaying `https://www.google.com/search?channel=nrow5&client=firefox-b-d&q=descargar+git`. The search bar contains the text "descargar git". Below the search bar, the Google logo is visible on the left, and navigation links for "Todo", "Videos", "Imágenes", "Noticias", "Shopping", "Más", and "Herramientas" are on the right. The search results indicate approximately 1,720,000 results in 0.50 seconds.

The first search result is highlighted with a red box and contains the following information:

- URL: <https://git-scm.com/downloads> with a link to "Traducir esta página".
- Title: **Downloads - Git**
- Snippet: "Older releases are available and the **Git** source repository is on **GitHub**. Latest source Release. 2.33.0 Release Notes (2021-08-16) Download for Linux ... macOS · Download for Linux and Unix · GUI Clients · Logos"

Below the highlighted result, another result is visible:

- URL: <https://git-scm.com/download/win> with a link to "Traducir esta página".
- Title: **Downloading Package - Git**
- Snippet: "Downloading **Git**. Your download is starting... You are downloading the latest (2.33.0) 32-bit version of **Git** for Windows ..."

Further down, there is a result from the Git SCM book:

- URL: <https://git-scm.com/book/Inicio---Sobre-el-Control-...> with a link to "Traducir esta página".
- Title: **1.5 Inicio - Sobre el Control de Versiones - Instalación de Git**
- Snippet: "Puedes encontrar esta información en <http://git-scm.com/download/linux>. Instalación en Mac. Hay varias maneras de instalar **Git** en un Mac. Probablemente la más ..."

Below these results is a section titled "Otras preguntas de los usuarios" (Other user questions) with a list of questions and expandable answers:

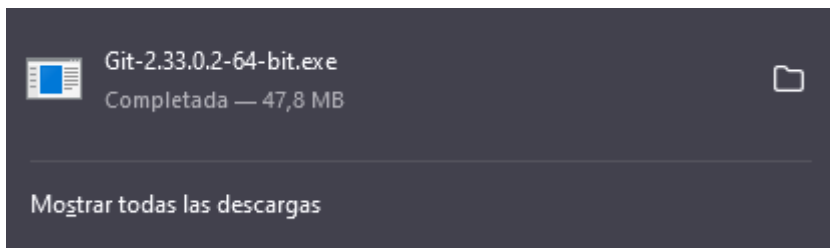
- ¿Cómo descargar Git?
- ¿Cómo descargar e instalar Git?
- ¿Qué es GitHub Desktop?
- ¿Qué es Git https Git SCM com?

At the bottom of the page, there is a link to <https://gitforwindows.org> with a link to "Traducir esta página", followed by the title **Git for Windows** and a snippet: "As Windows users commonly expect graphical user interfaces, **Git** for Windows also provides the **Git GUI** a powerful alternative to **Git BASH** offering a graphical

Seguido de esto daremos click en descargar



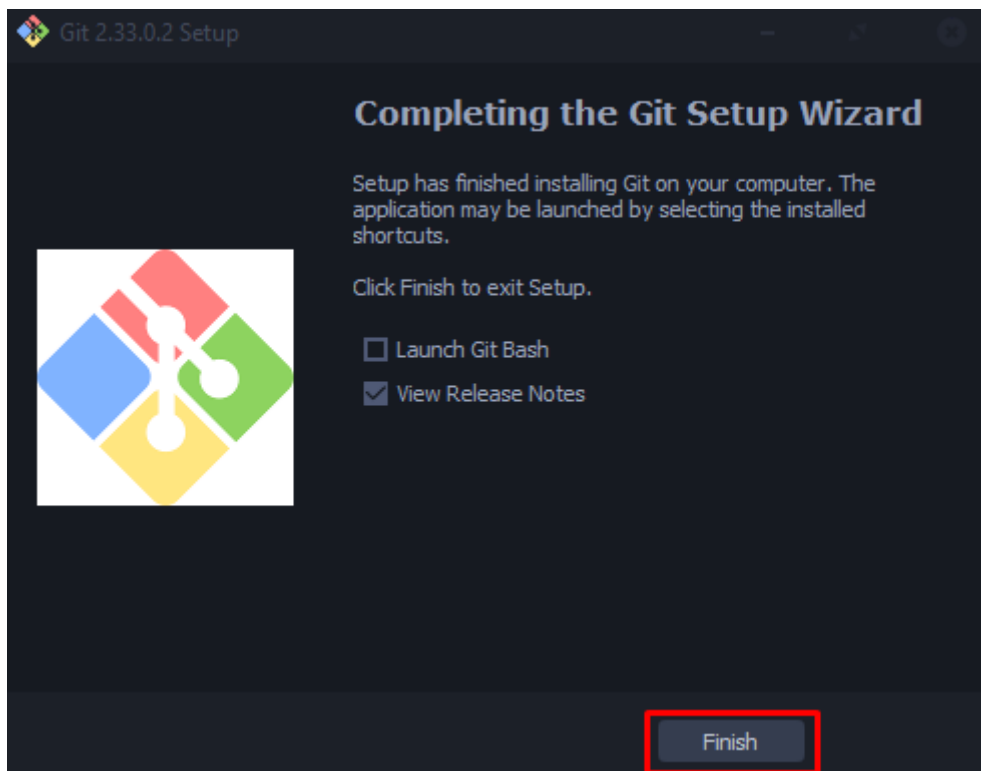
Una vez finaliza le damos doble click al ejecutable y lo instalamos



Una vez hayamos empezado la instalacion daremos todo next hasta terminar con ella



Una vez finalizada la instalación le damos en Terminar



Para verificar que todo esta correcto, vamos a la powershell y escribimos lo siguiente:

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

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

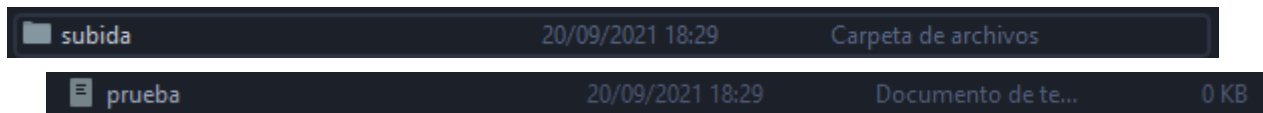
PS C:\Users\Mike> git version
git version 2.33.0.windows.2
PS C:\Users\Mike>
```

Comos vemos esta instalado y con esa versión

## 1.2 Creación del directorio a subir

Ahora vamos a crear una carpeta con un fichero para la prueba de subida

Esta es la carpeta con su archivo

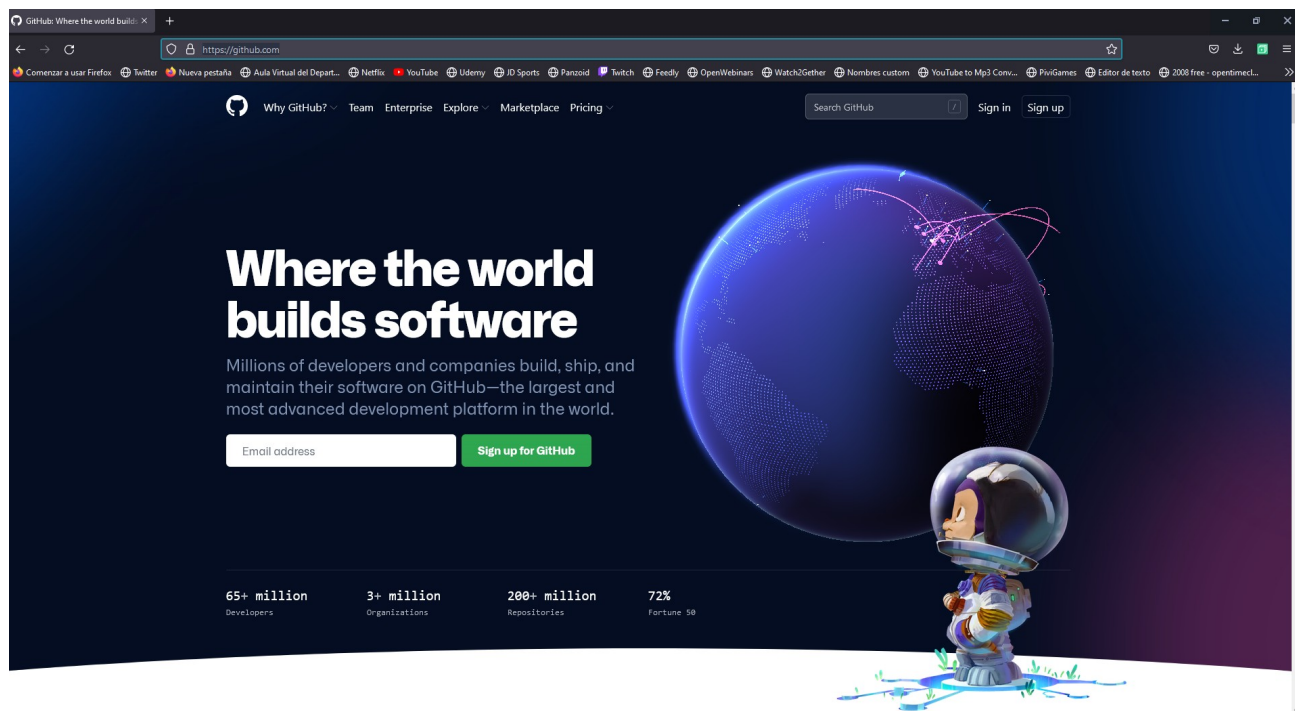


Ya estaría lista nuestra carpeta

## 2. Preparando entorno en github

### 2.1 Crear cuenta de github

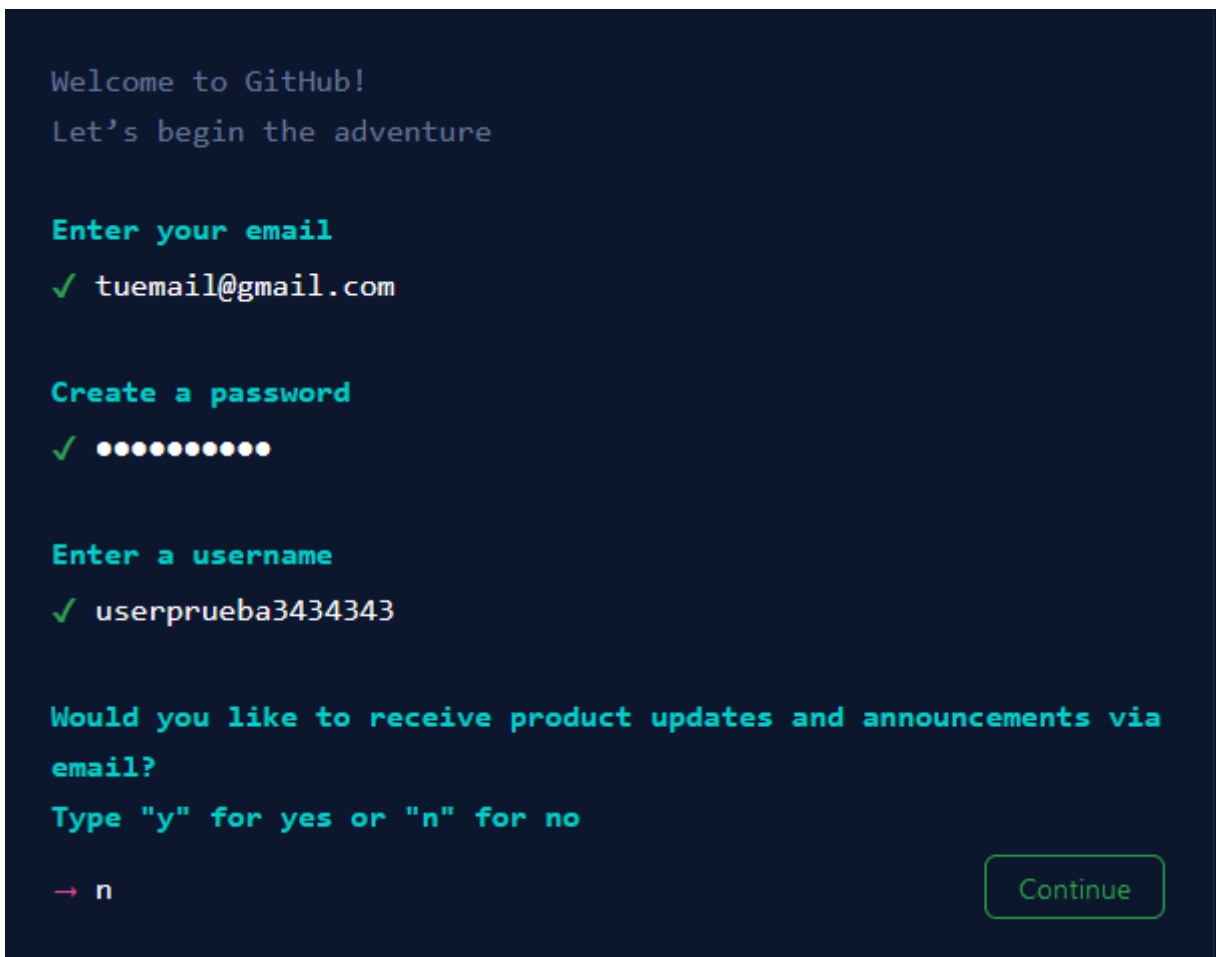
Para entraremos en <https://github.com/>



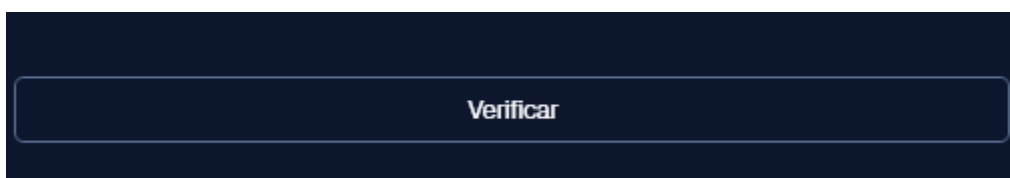
Le daremos click en registrarnos



Una vez dentro nos empezará a preguntar por nuestros datos y los rellenaremos



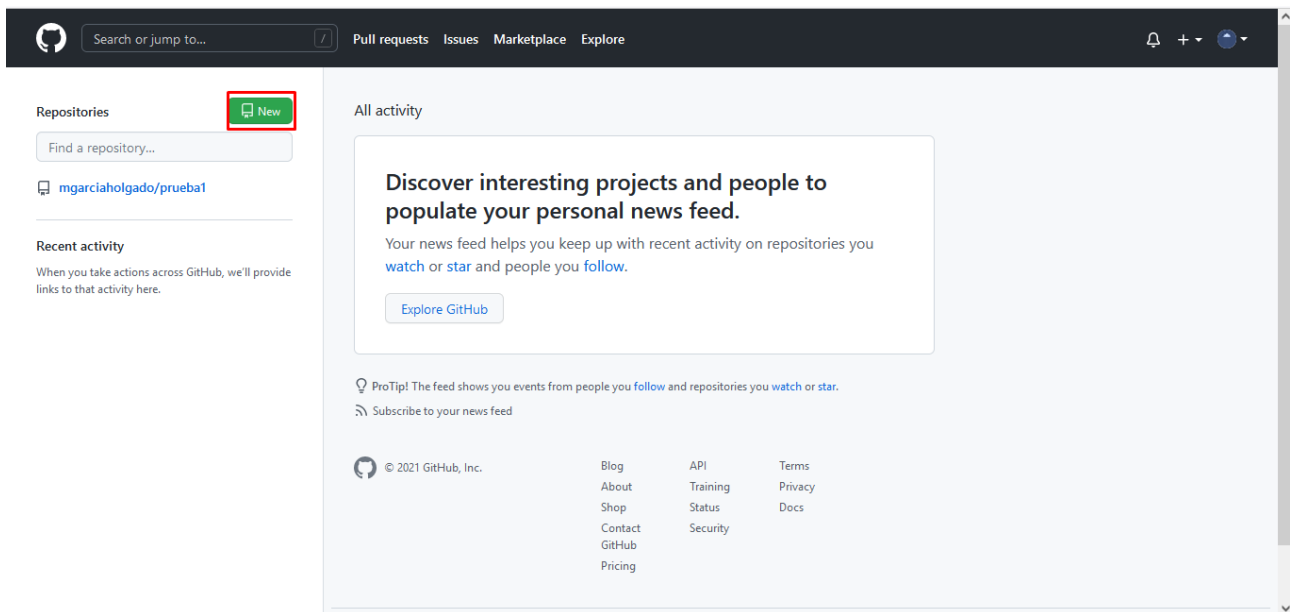
Una vez nos hayamos rellenado todo habrá que verificar la cuenta



Nos llegará un código a nuestro correo y lo verificaremos con el

## 2.2 Crear un repositorio

Para crear un repositorio en github nos dirigimos hacia la web de github → <https://github.com/> y seguimos los siguientes pasos para crear nuestro primer repositorio.



1. Nombre del repositorio
2. Descripción del repositorio
3. Asignaremos al repositorio si es publico o privado

### 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 \*

/

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

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. 3

Initialize this repository with:

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

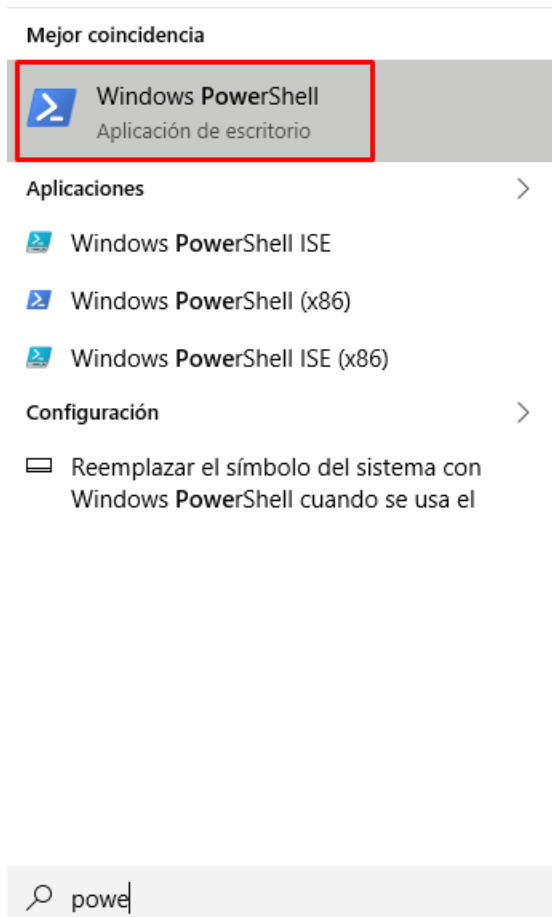
☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)



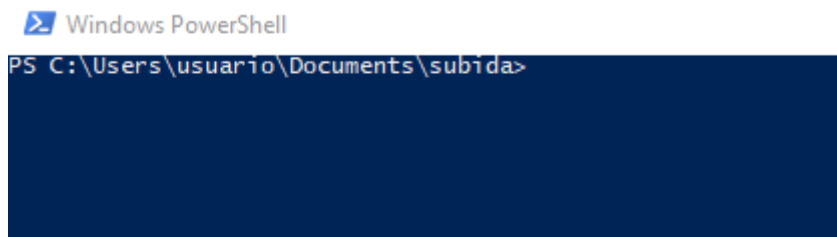
## 3. Subiendo nuestros archivos

### 3.1 Usar PowerShell para subir nuestra carpeta

Abrimos la powershell para empezar con nuestra subida de la carpeta

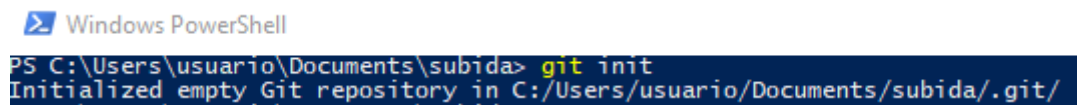


Una vez hayamos entrado a la powershell nos iremos a la carpeta que queremos subir



Una vez posicionado en la carpeta vamos a usar el comando git init **(todos los comandos explicados están en el txt de comandos, miralos si te interesa conocer el uso de cada comando)**

Una vez puesto el comando, nos sale un mensaje de que se ha inicializado git en la carpeta

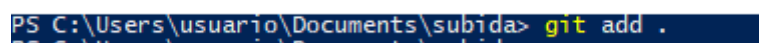


```
Windows PowerShell
PS C:\Users\usuario\Documents\subida> git init
Initialized empty Git repository in C:/Users/usuario/Documents/subida/.git/
```

Seguido de esto usaremos la guía de comandos que nos proporciona github al crear un nuevo repositorio

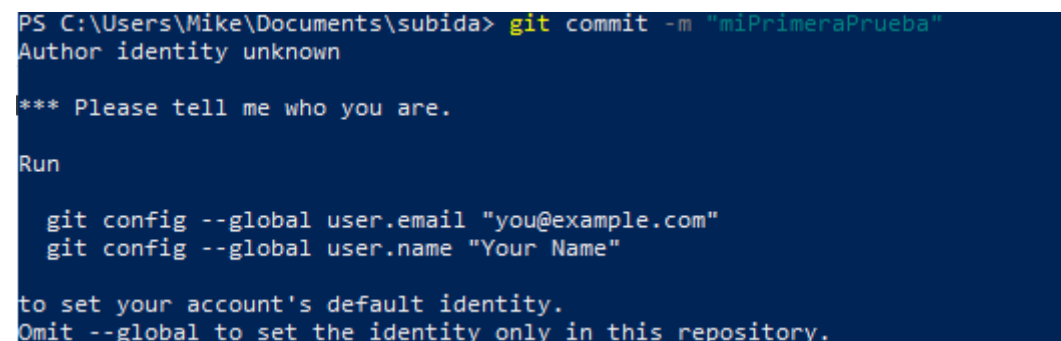
```
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/mgarciaholgado/practica1.git
git push -u origin main
```

Estos son los comandos, estarán explicados en el txt de comandos pero ahora nos dedicaremos a subir la carpeta, así que si ya hemos usado git init pues pasaremos al siguiente comando, brevemente explicado sirve para añadir los archivos que quieras subir, usaremos el "." para indicar que queremos subir todo el contenido.



```
PS C:\Users\usuario\Documents\subida> git add .
PS C:\Users\usuario\Documents\subida>
```

Pasamos al siguiente comando, este sirve para añadir un comentario, más info sobre el comando en el txt de comandos.



```
PS C:\Users\Mike\Documents\subida> git commit -m "miPrimeraPrueba"
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.
```

Nos salta este pequeño error, el cual nos indica que le tenemos que autenticar mediante esos comandos.

Lo haremos de la siguiente manera:

```
PS C:\Users\Mike\Documents\subida> git config --global user.mail "mgarholb@g.educaand.es"
PS C:\Users\Mike\Documents\subida> git config --global user.name "mgarciaholgado"
PS C:\Users\Mike\Documents\subida>
```

Una vez puestos estos datos vamos a seguir con el siguiente comando.

El siguiente nos indicara con que rama vamos a subirlo en nuestro casa al tratarse de un repositorio nuevo usaremos la principal (main)

```
PS C:\Users\Mike\Documents\subida> git branch -M main
PS C:\Users\Mike\Documents\subida>
```

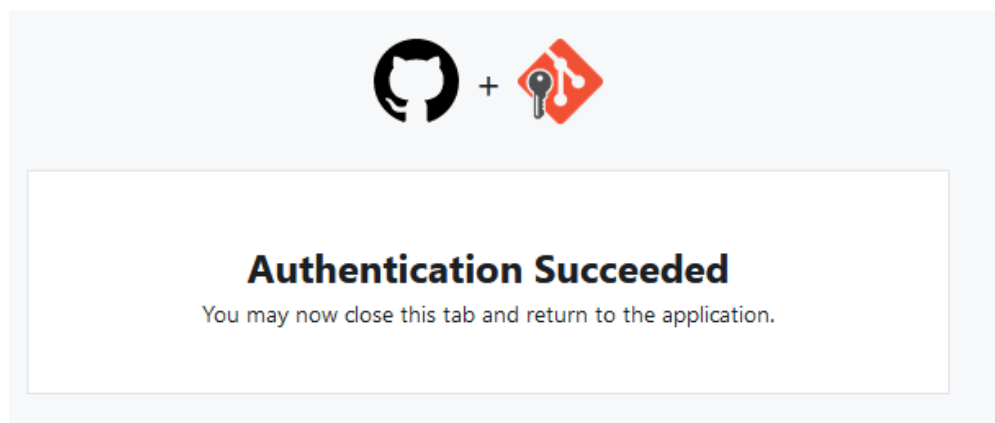
Seguido de esto ya estaremos a punto de finalizar con la subida al repositorio, ahora vamos a indicarle el nombre del repositorio al que vamos a subirlo, lo haremos así:

```
PS C:\Users\Mike\Documents\subida> git remote add origin https://github.com/mgarciaholgado/practical1.git
PS C:\Users\Mike\Documents\subida>
```





Ahora que ya hemos seleccionado el repositorio vamos a lanzar los archivos hacia el mediante un push:

```
PS C:\Users\Mike\Documents\subida> 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 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 235 bytes | 235.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mgarciaholgado/practical1.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'
PS C:\Users\Mike\Documents\subida>
```

Después de haber hecho esto y si todo a salido bien, se nos debe abrir el navegador web para iniciar sesión en github



Ahora vamos a ver si se ha subido correctamente recargando la página en github

	mgarciaholgado miPriemraPractica
	prueba.txt miPriemraPractica
	prueba1.txt miPriemraPractica
	prueba2.txt miPriemraPractica

## 4. Comandos en git

### 4.1 Comandos principales

- **git init:** El comando git init crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git, o para inicializar un nuevo repositorio vacío.

 Windows PowerShell

```
PS C:\Users\usuario\Documents\subida> git init
Initialized empty Git repository in C:/Users/usuario/Documents/subida/.git/
```

- **git add:** es el comando que mueve al índice las modificaciones que hayamos realizado.

```
PS C:\Users\usuario\Documents\subida> git add .
PS C:\Users\usuario\Documents\subida>
```

En este caso uso el punto porque quiero subir todo, si solo queremos subir un archivo o varios bastará con poner la siguiente estructura:

```
PS C:\Users\Mike> git add <nombre archivo>
```

- **git commit:** este comando nos crea algo parecido a una instantánea del repositorio.

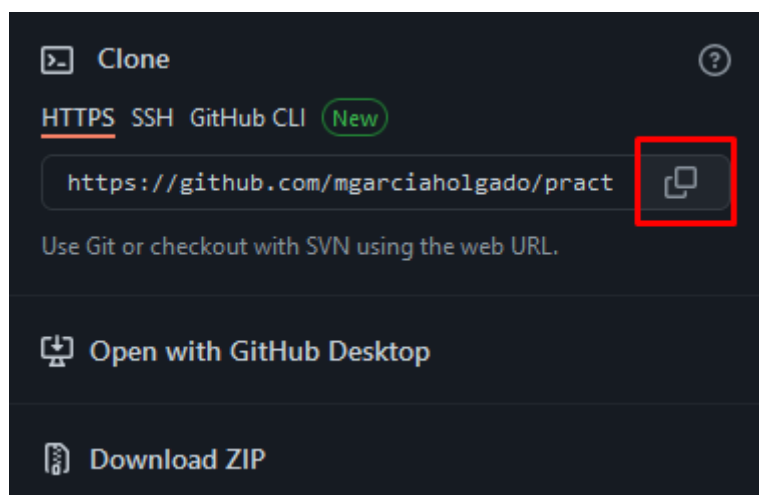
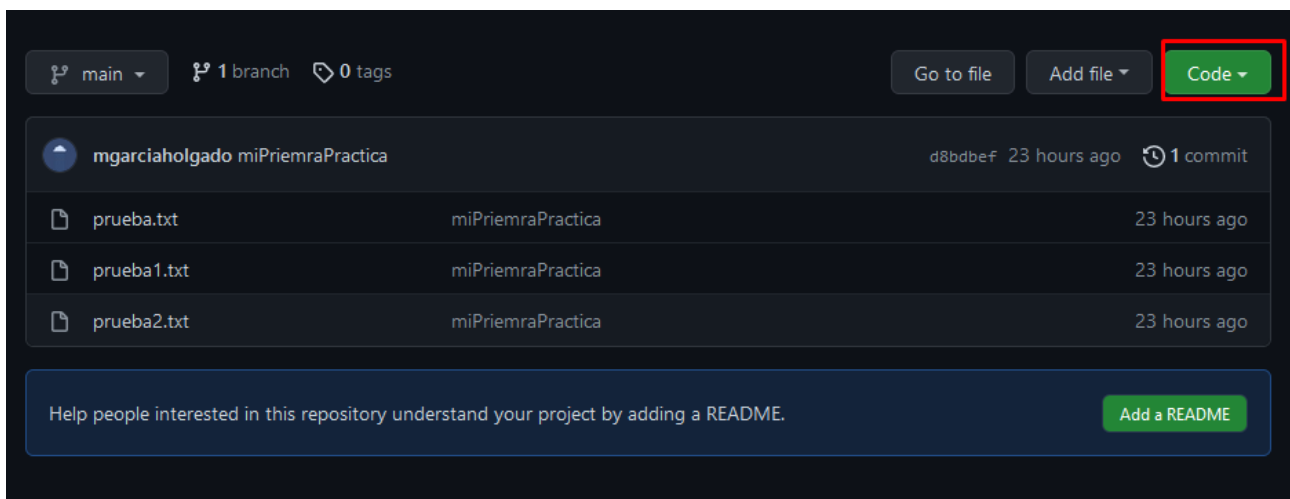
El comando sin ninguna opción abrirá un editor de texto que te pedirá un mensaje para la confirmación.

Para ahorrarnos todo esto pues usaremos la opción -m, se usa de esta manera:

```
PS C:\Users\Mike> git commit -m "commit message"
```

- **git remote add origin <url-repositorio>** : este comando te conectará al repositorio que le hayas indicado en la URL que se puede conseguir de la siguiente manera:

1.



- **git push:** se suele usar para publicar y cargar cambios locales a un repositorio central.

El modo estándar que se usa este comando es con la siguiente estructura:

```
PS C:\Users\Mike> git push origin master
```

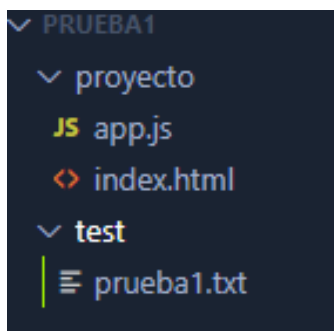
Una vez hallamos realizado el commit usaremos este comando para terminar con la subida a nuestro repositorio, si eres nuevo en tu ordenador tendrás que iniciar sesión en github, se te abrirá una ventana en el buscador para iniciar sesión.

Si previamente has usado git, se te subirá sin problemas.

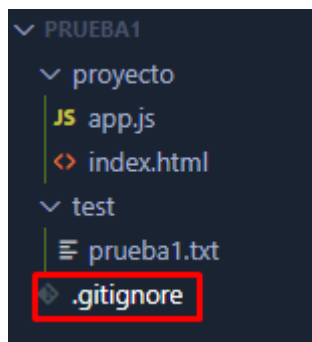
## 4.2 Comando ignore

Este comando se usa para ignorar archivos los cuales no queremos que se suban a nuestro repositorio pero queremos que estén ahí, por a lo mejor es una carpeta con archivos de pruebas.

Vamos a ver como se usa este comando con unos ejemplos, para ello empezaremos con crear una nueva carpeta:

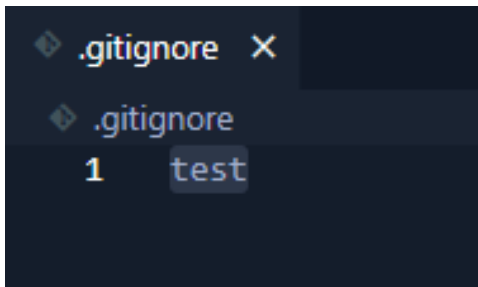


Tenemos esta estructura de directorios, ahora vamos a añadir dentro la carpeta raíz vamos a crear el archivo → **.gitignore**



En ese archivo vamos a añadir las carpetas o archivos que no queremos que se suban.

En nuestro caso vamos a ignorar la carpeta test, pues vamos al archivo y escribimos dentro lo siguiente:



Una vez hemos añadido lo que no queremos subir vamos a una terminal y hacemos los pasos que anteriormente hemos indicado, iniciar git, añadir archivos etc..

Cuando en mi caso he hecho un git add . (añadir todo el contenido), vemos que la carpeta test la ignora, lo podremos comprobar una vez añadido los archivos con un **git status**.

```
PS C:\Users\Mike\Documents\prueba1> git add .
PS C:\Users\Mike\Documents\prueba1> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   proyecto/app.js
    new file:   proyecto/index.html
```

## 4.3 Comando diff

---

Este comando sirve para ver las diferencias hechas en los archivos.

Vamos a ver un ejemplo añadiendo texto en el archivo html, nuestro archivo estaba vacío anteriormente, pues vamos a añadirle código para ver como se usa el comando.

Archivo original.....

```
proyecto > < index.html
1
```

Archivo con html....

```
proyecto > < index.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     |
11  </body>
12  </html>
13
```



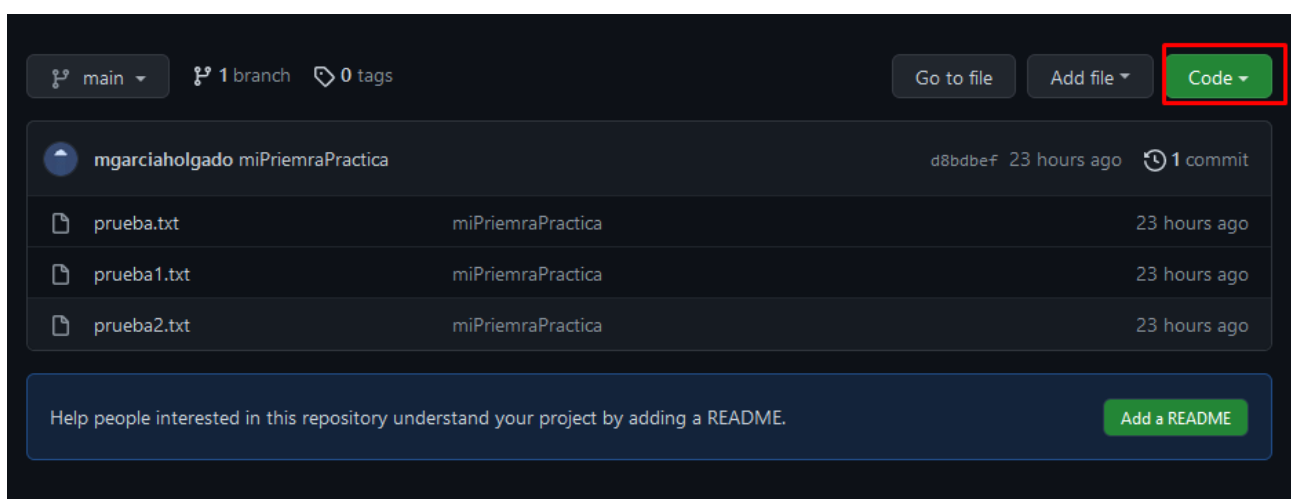
Una vez hayamos modificado el archivo vamos a la shell y escribimos lo siguiente...

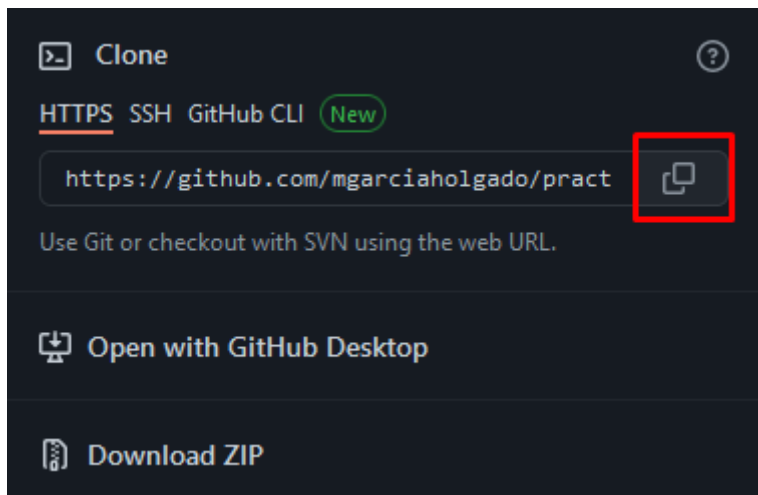
```
PS C:\Users\Mike\Documents\prueba1\proyecto> git diff index.html
diff --git a/proyecto/index.html b/proyecto/index.html
index e69de29..11f93dc 100644
--- a/proyecto/index.html
+++ b/proyecto/index.html
@@ -0,0 +1,12 @@
+<!DOCTYPE html>
+<html lang="en">
+<head>
+  <meta charset="UTF-8">
+  <meta http-equiv="X-UA-Compatible" content="IE=edge">
+  <meta name="viewport" content="width=device-width, initial-scale=1.0">
+  <title>Hola</title>
+</head>
+<body>
+  [REDACTED]
+</body>
+</html>
```

Una vez escrito el comando nos indica mediante los signos de + que se han añadido esas líneas al archivo.

## 4.4 Comando clone

El comando clone sirve para descargar tu repositorio remoto en tu máquina local mediante el enlace que ya vimos anteriormente como se conseguía, pero lo recordamos en un momento:





Una vez tengamos esa URL vamos a una shell y bastará con poner lo siguiente:

1º Vamos a crear una carpeta donde se descargará el repositorio

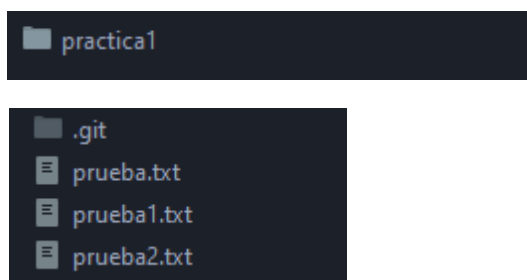
2º Vamos a entrar a ese directorio

3º Vamos a descargar el repositorio con **git clone**

```
PS C:\Users\Mike\Desktop\Clonacion>
```

```
PS C:\Users\Mike\Desktop\Clonacion> git clone https://github.c
Cloning into 'practical'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Users\Mike\Desktop\Clonacion> _
```

Una vez finalizada la clonación vamos a ver si a sido con éxito.



La clonación a salido bien y ya tenemos nuestro repositorio clonado.

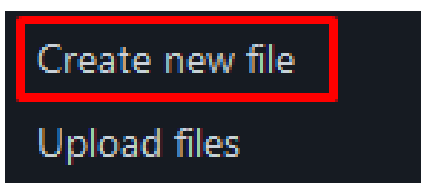
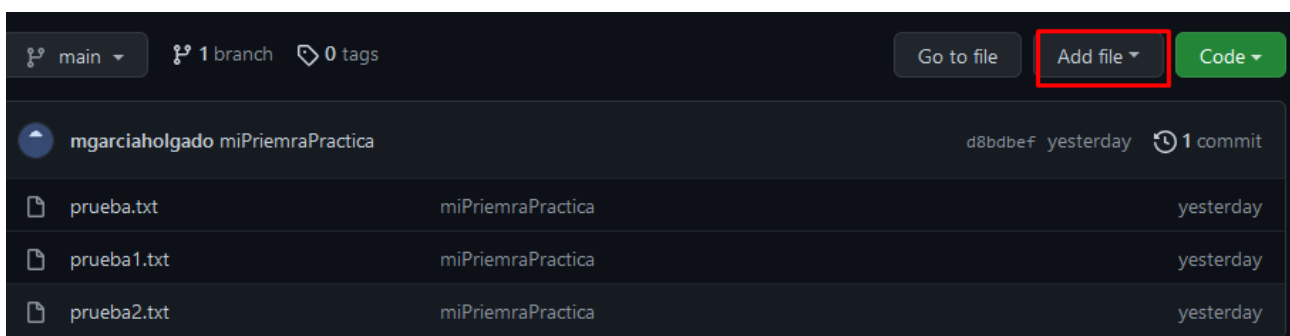
## 4.5 Comando pull

Este comando sirve por si has creado de manera gráfica archivos en la web de github, traerte esos archivos a tu máquina local para que todo este actualizado, veamos un ejemplo.

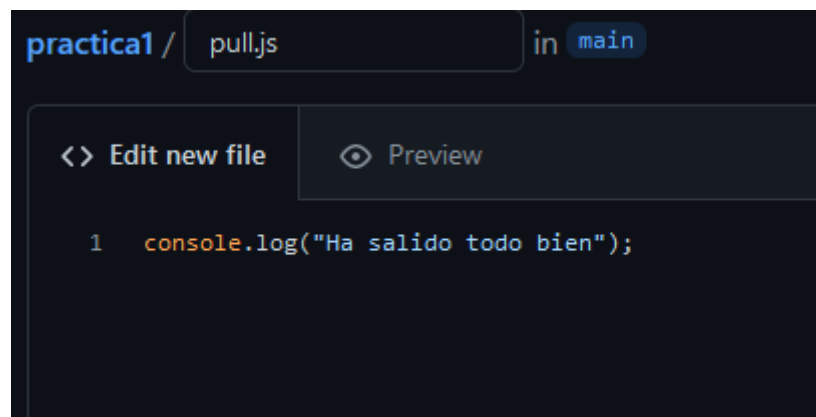
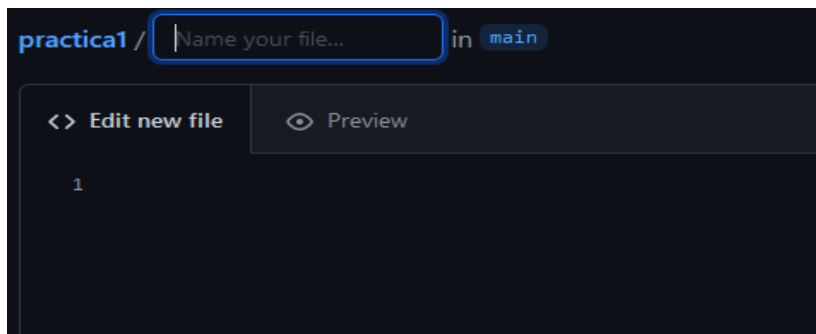
Nos dirigimos hacía la web de github



Como podemos ver tenemos esos archivos, así que vamos a crear uno nuevo:



Una vez estamos en esta pestaña, lo unico que tendremos que hacer será rellenar nuestro fichero, en mi caso haré un .js



Una vez creado vamos a nuestra shell para traernos ese archivo:

```
PS C:\Users\Mike\Documents\subida> git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 691 bytes | 230.00 KiB/s, done.
From https://github.com/mgarciaholgado/practical
   d8bdbef..0fc4b90  main       -> origin/main
Updating d8bdbef..0fc4b90
Fast-forward
 pull.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pull.js
```

Como vemos nos a añadido el archivo creado en web a nuestra carpeta

