

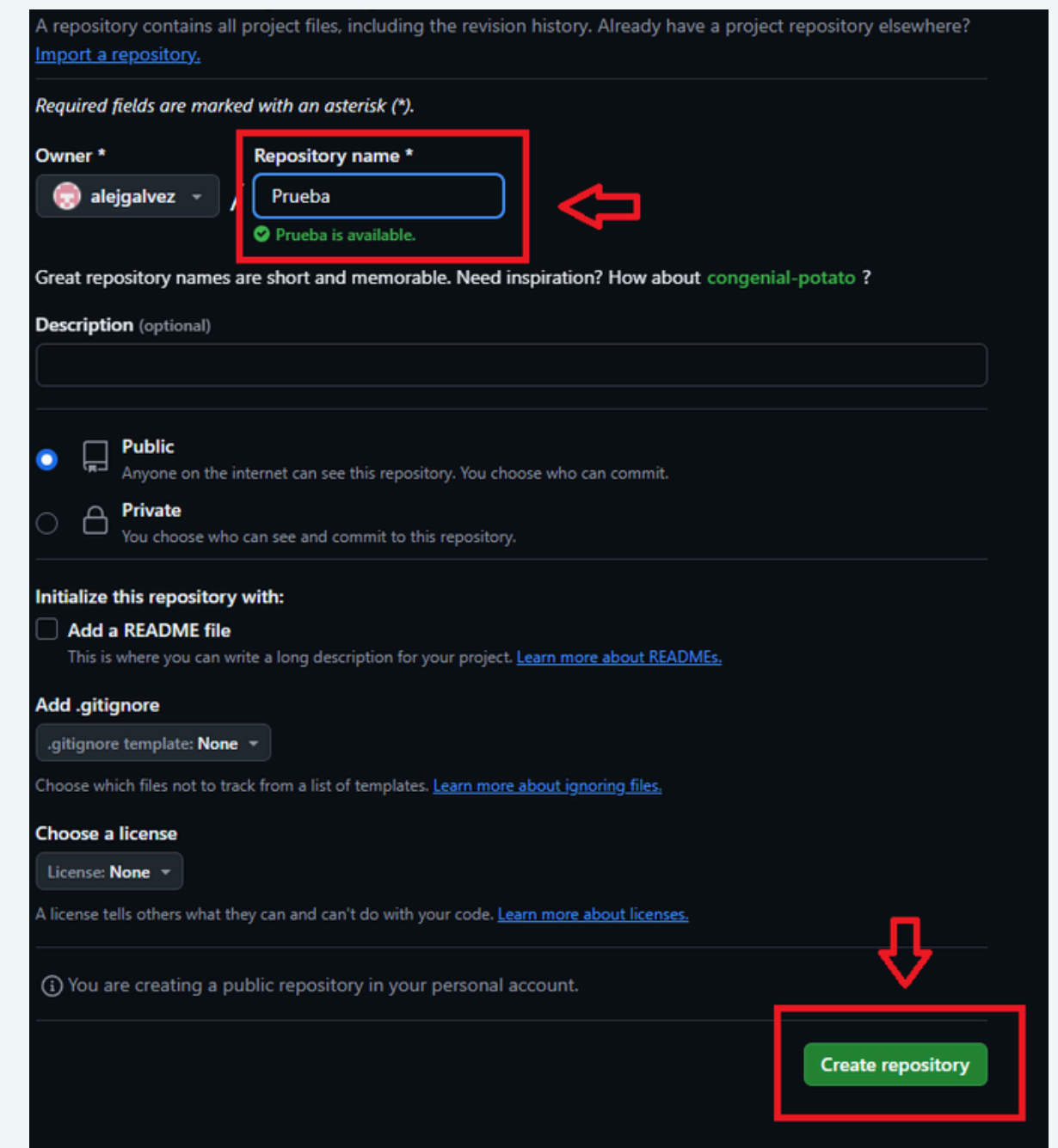
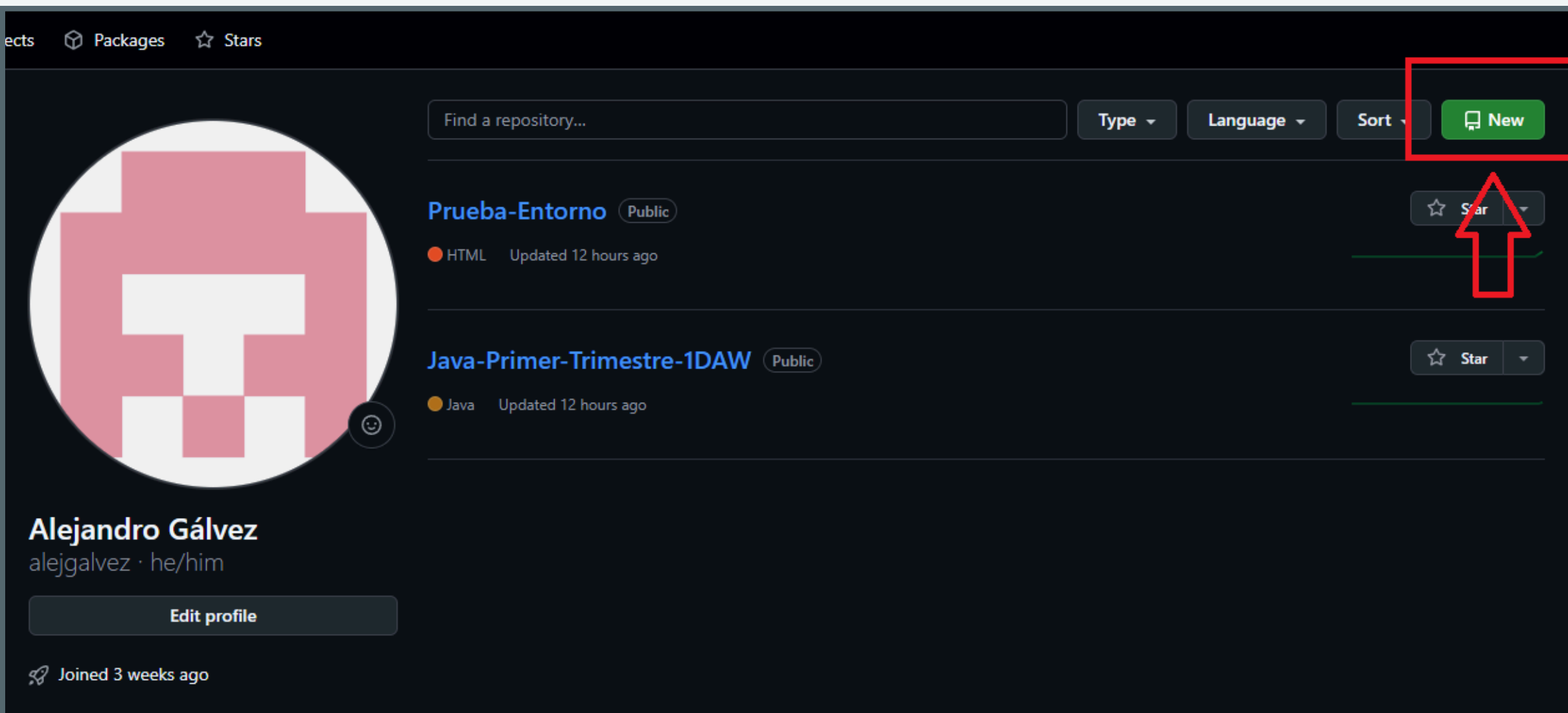
Alejandro Gálvez

GITHUB



Paso 1.



Creamos un repositorio en GitHub.



Paso 2.

Una vez creado nos generará una ruta a seguir en nuestra consola.


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

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/alejgalvez/Prueba.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 "# Prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/alejgalvez/Prueba.git
git push -u origin main
```

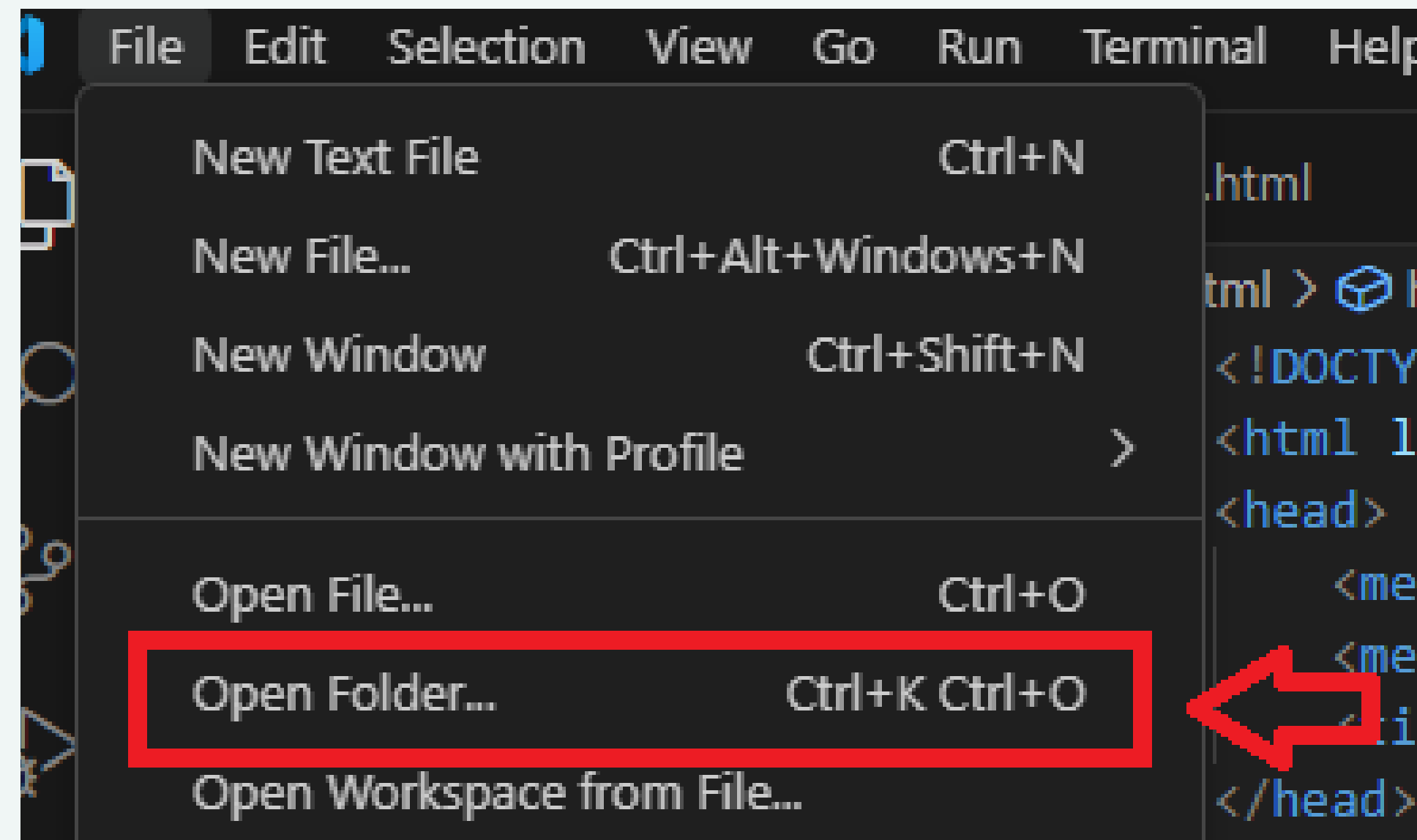
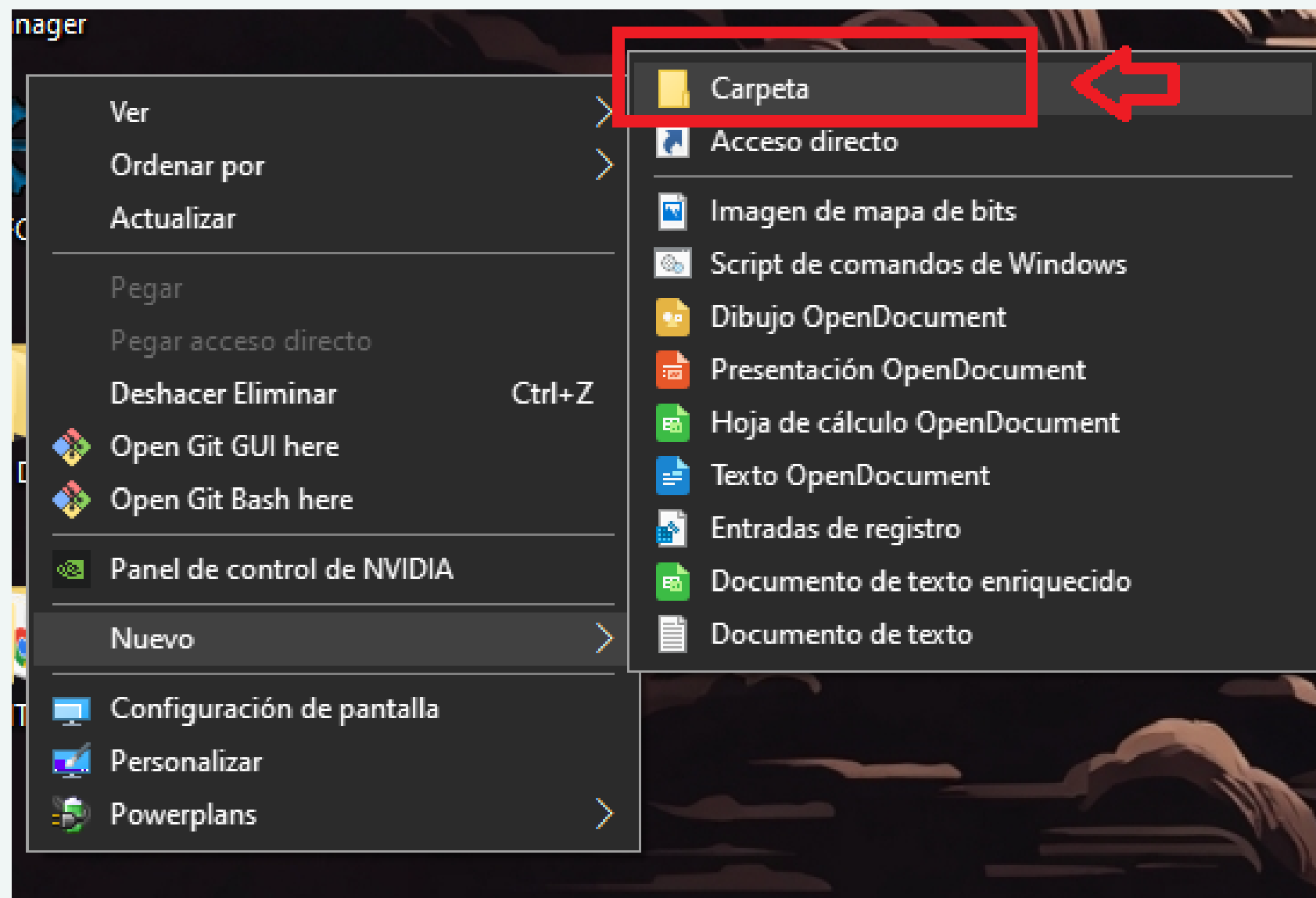


...or push an existing repository from the command line

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

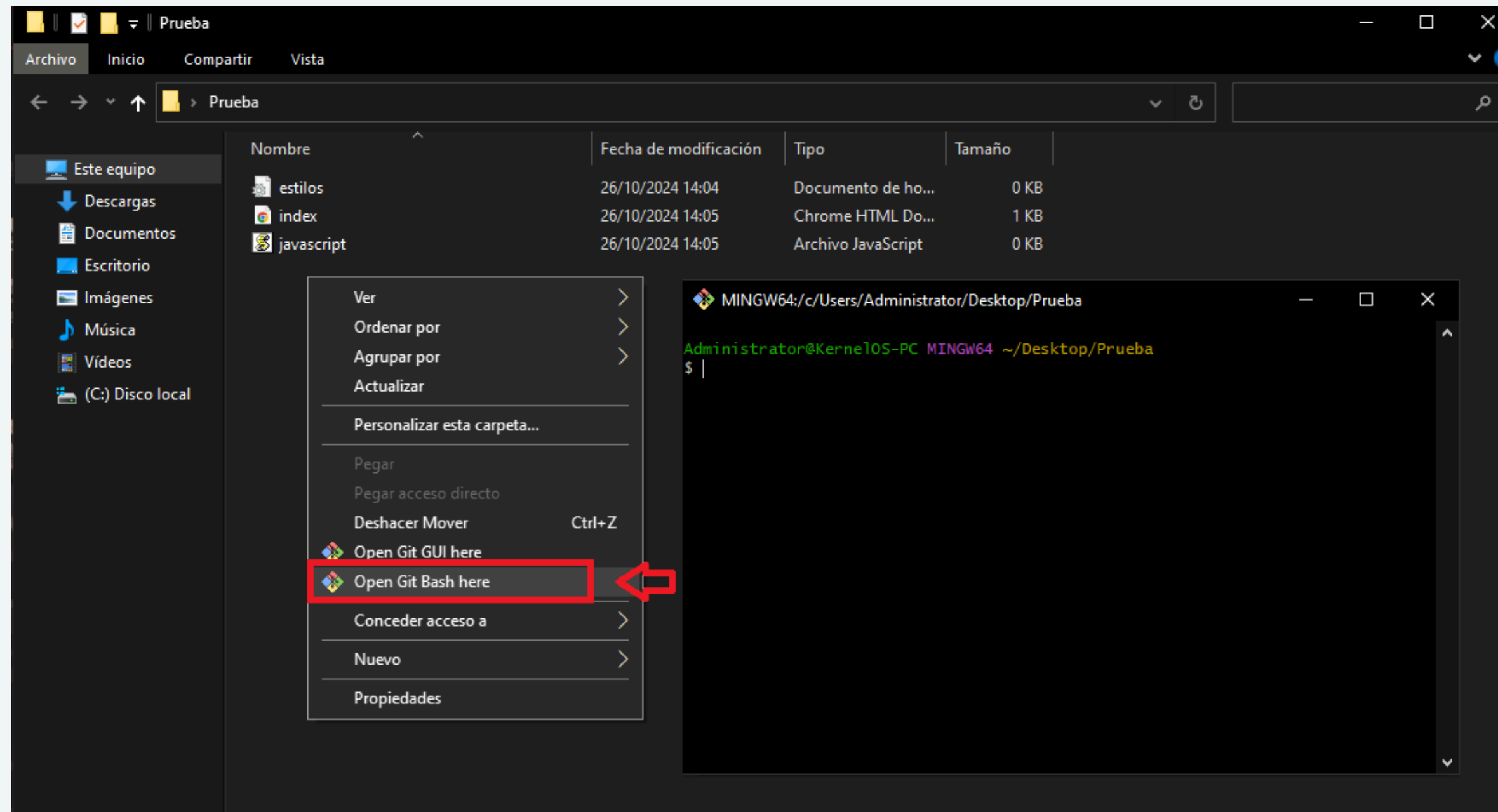
Paso 3.

Creamos una carpeta dentro de nuestro equipo y procedemos a abrirla en VSC para empezar a trabajar.



Paso 4.

Ya guardado nuestro trabajo en VSC, procedemos a entrar en la carpeta creada y abrimos GIT Bash para entrar en la consola y subirlo a GitHub.



Paso 5. COMANDOS

1- Git init (para iniciar la consola)

2- Git add . (para añadir nuestro trabajo)

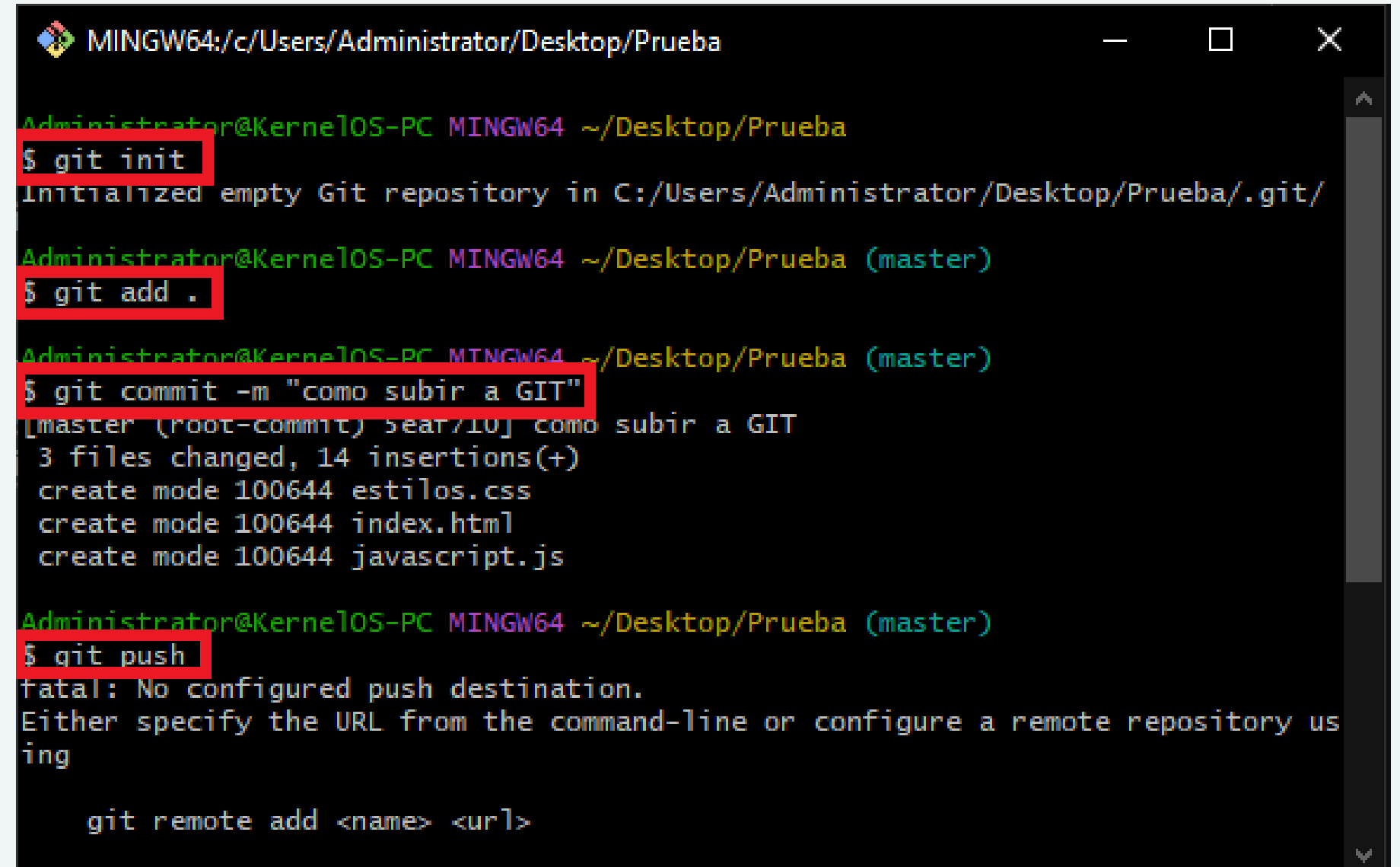
3- Git commit -m "" (para comentar nuestro trabajo)

4- Git push (para subir)

5- <enlace a nuestro repositorio>

6- Git branch -m Main <Para convertir a main>

7- Git push -u origin main <último paso para subirlo a nuestra nube de GitHub>



```
MINGW64:/c/Users/Administrator/Desktop/Prueba
Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba
$ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/Prueba/.git/

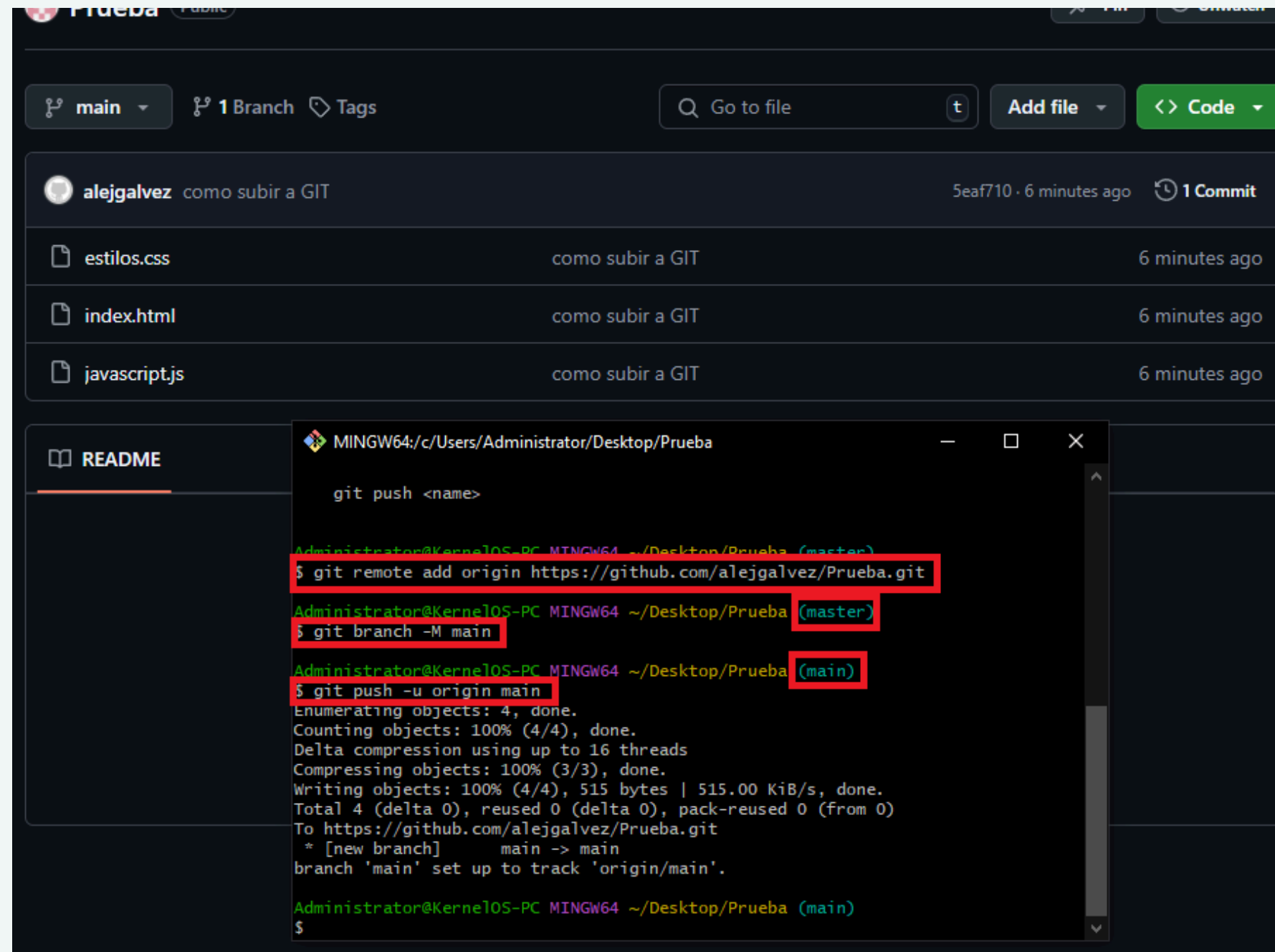
Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (master)
$ git add .

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (master)
$ git commit -m "como subir a GIT"
[master (root-commit) 5eaf710] como subir a GIT
 3 files changed, 14 insertions(+)
 create mode 100644 estilos.css
 create mode 100644 index.html
 create mode 100644 javascript.js

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (master)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>
```

Paso 6. Una vez terminamos de poner los comandos anteriores en nuestra consola, ya tendremos nuestro trabajo subido a la nube de GitHub.



The screenshot shows a GitHub repository named 'Prueba' with a commit by 'alejgalvez' titled 'como subir a GIT'. The commit includes three files: 'estilos.css', 'index.html', and 'javascript.js'. Below the repository view, a terminal window is open, displaying the following commands and their output:

```
MINGW64:/c/Users/Administrator/Desktop/Prueba

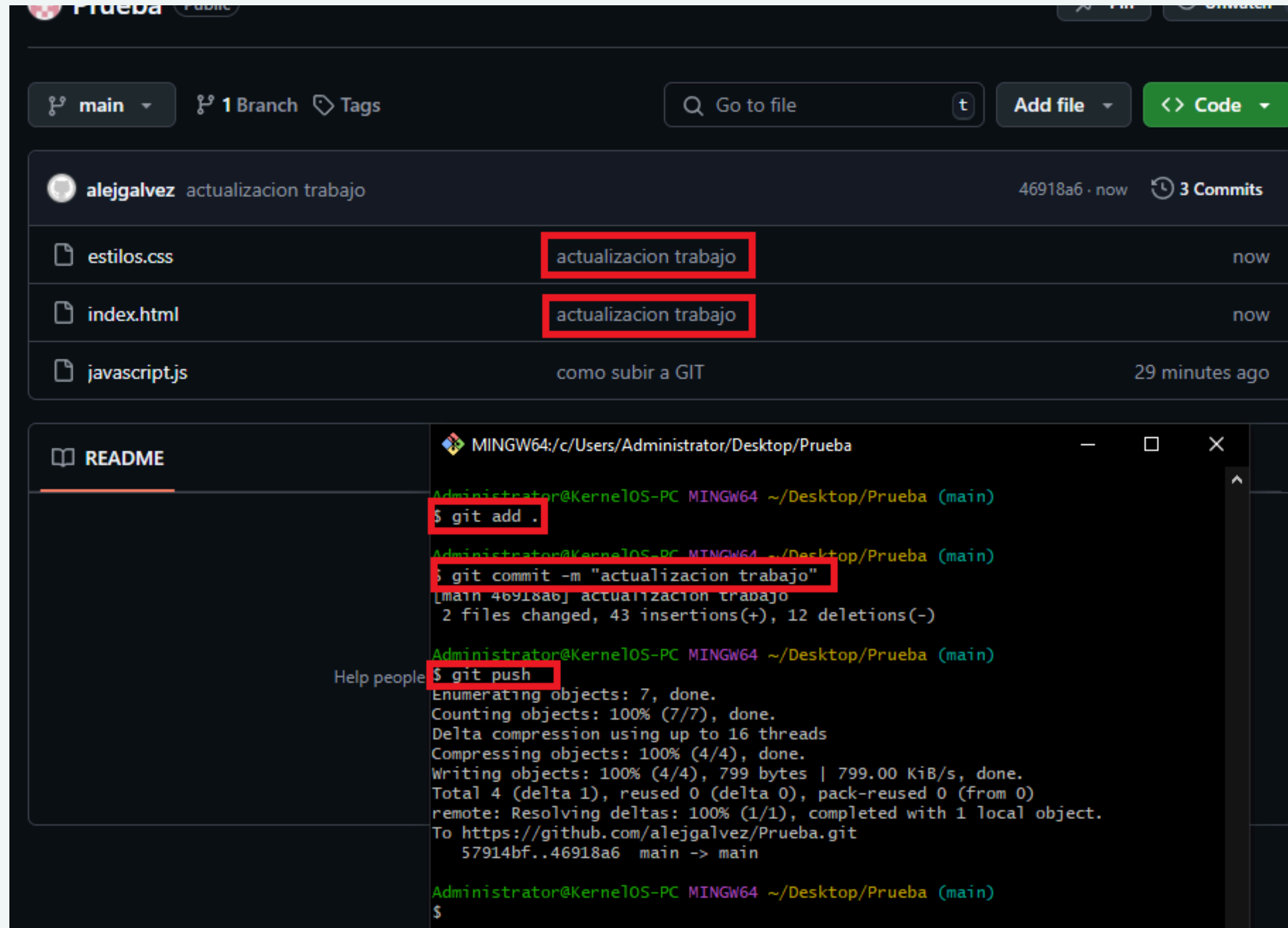
git push <name>

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (master)
$ git remote add origin https://github.com/alejgalvez/Prueba.git
Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (master)
$ git branch -M main
Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 515 bytes | 515.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/alejgalvez/Prueba.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$
```


Paso 7. Para actualizar nuestro trabajo y que los cambios se vean en GitHub haremos lo siguiente.

Entramos a la carpeta donde se encuentra nuestro trabajo y escribimos:



The screenshot shows a GitHub repository interface for a user named 'alejgalvez' with the repository name 'actualizacion trabajo'. The repository has 1 branch (main) and 3 commits. The file list shows 'estilos.css' and 'index.html' both with the commit message 'actualizacion trabajo' and 'now' status, and 'javascript.js' with the commit message 'como subir a GIT' and '29 minutes ago' status. Below the file list, there is a terminal window titled 'MINGW64: c:/Users/Administrator/Desktop/Prueba' showing the following commands and output:

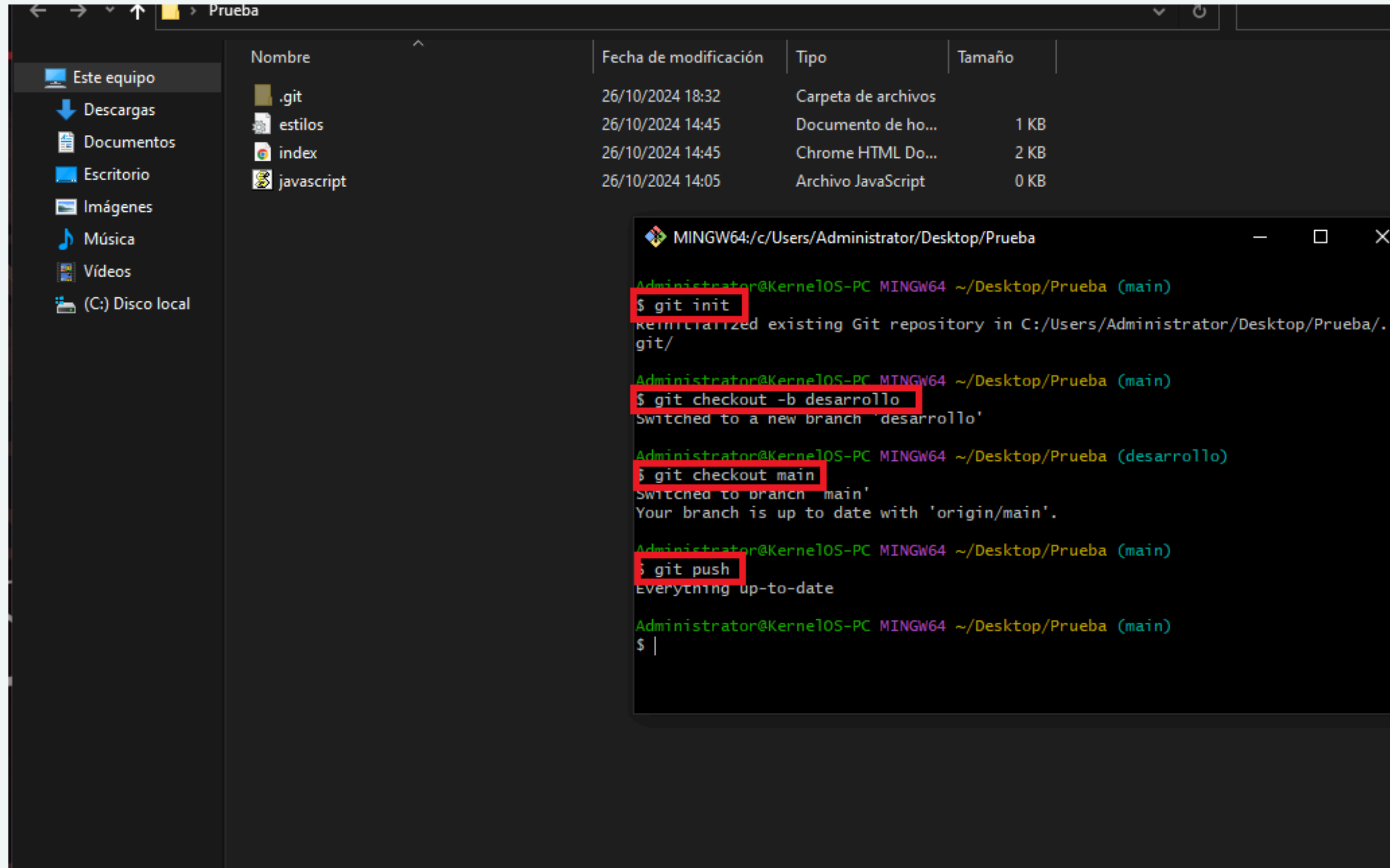
```
Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$ git add .

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$ git commit -m "actualizacion trabajo"
[main 46918a6] actualizacion trabajo
2 files changed, 43 insertions(+), 12 deletions(-)

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 799 bytes | 799.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/alejgalvez/Prueba.git
57914bf..46918a6 main -> main

Administrator@KernelOS-PC MINGW64 ~/Desktop/Prueba (main)
$
```


Cómo crear una rama en GitHub dentro de nuestra carpeta.



Una vez creamos la rama y modificamos cualquier cosa para actualizar nuestro trabajo (en este caso VSC) volvemos a repetir el paso número 7.

