

MANUAL PARA GIT

Realizado por Elmer Santos

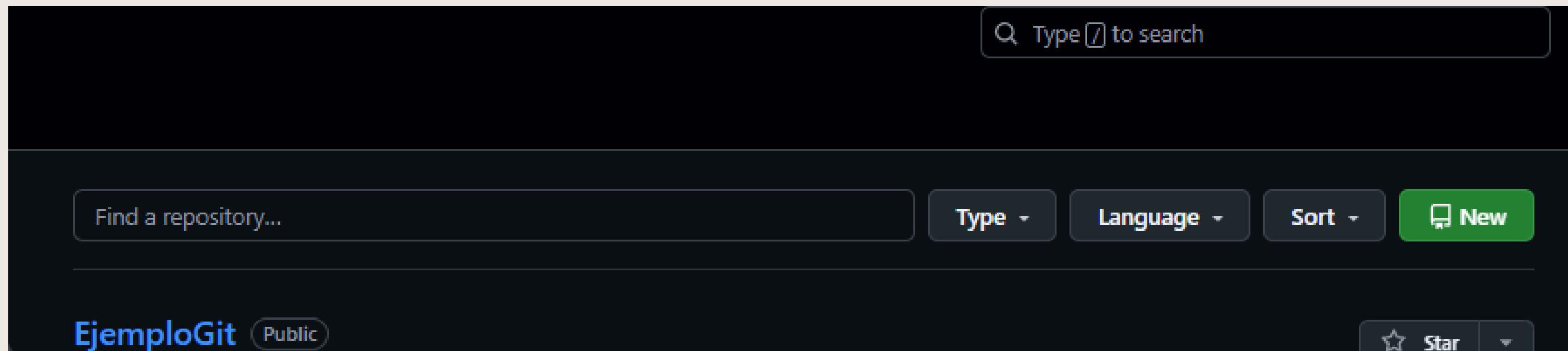
PASO 1

Crear un repositorio en Git Hub. Para ello nos dirigimos a la parte derecha de la pantalla, dándole click a nuestro avatar. Procedemos a dar click a Your repositories.

The screenshot shows the GitHub Home page for the user 'esantos2019254'. On the left, there's a sidebar with links like 'Dashboard', 'Repositories', and a 'New' button. The main area has sections for creating a new repository and introducing自己 with a profile README. On the right, a vertical navigation menu is open, showing options such as 'Your profile', 'Your repositories' (which is highlighted in green), 'Your Copilot', 'Your projects', etc. The 'Your repositories' section in the menu includes a sub-section for 'Unlock experience event or ...'.

PASO 2

Veremos esta parte en específico en nuestra pantalla, en donde le daremos click a New



PASO 3

Después del paso anterior veremos esta pantalla, en donde crearemos nuestro repositorio, le daremos un nombre, podemos agregar una descripción lo podemos poner publico o privado. Además podemos agregar el archivo README si así lo deseamos. Finalizamos dándole a Create repository.

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * esantos2019254 / **Repository name *** EjemploGit EjemploGit is available.

Great repository names are short and memorable. Need inspiration? How about [musical-spoon](#) ?

Description (optional) ejemplo 1 git

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

Add .gitignore **.gitignore template: None** Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

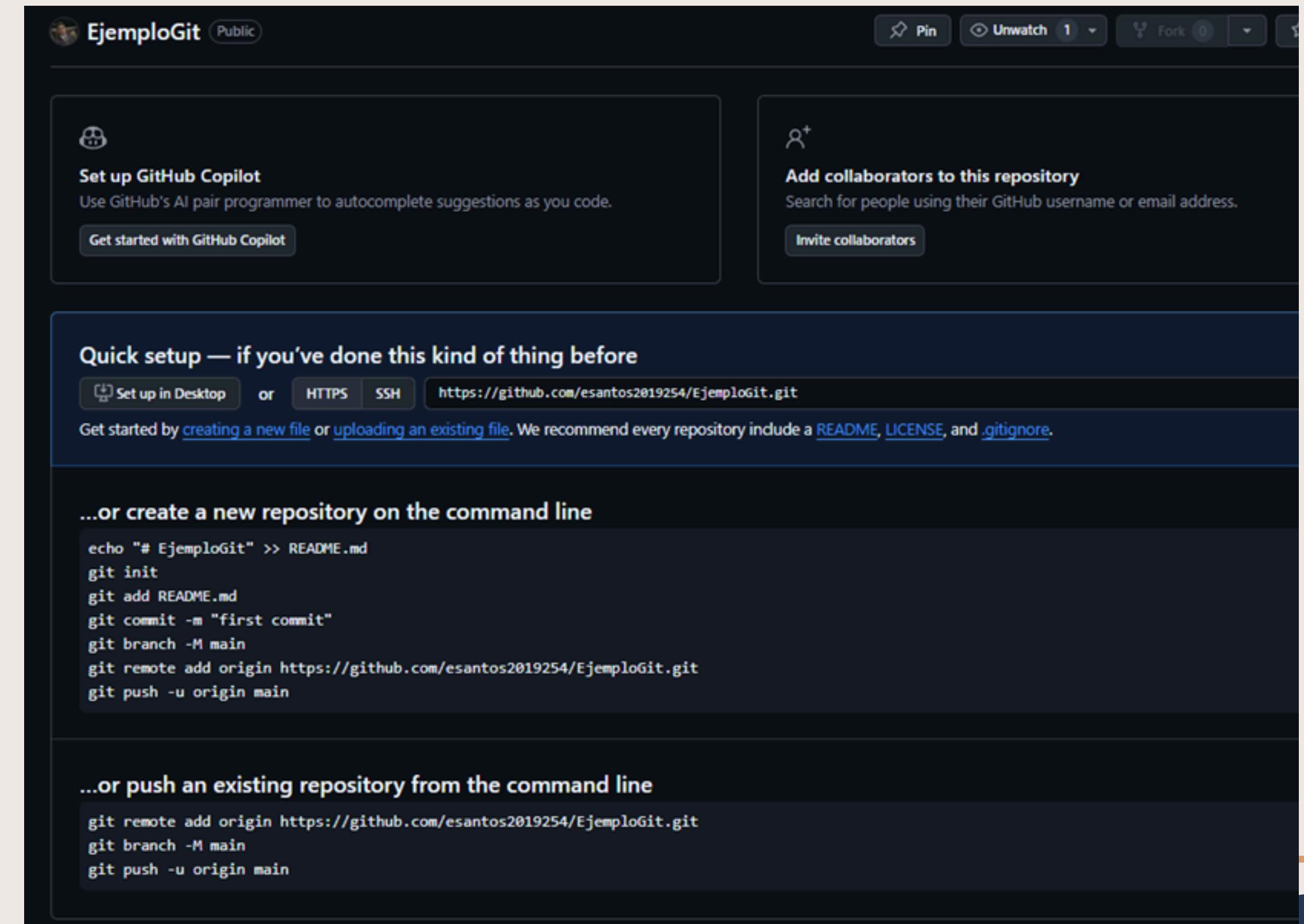
Choose a license **License: None** A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

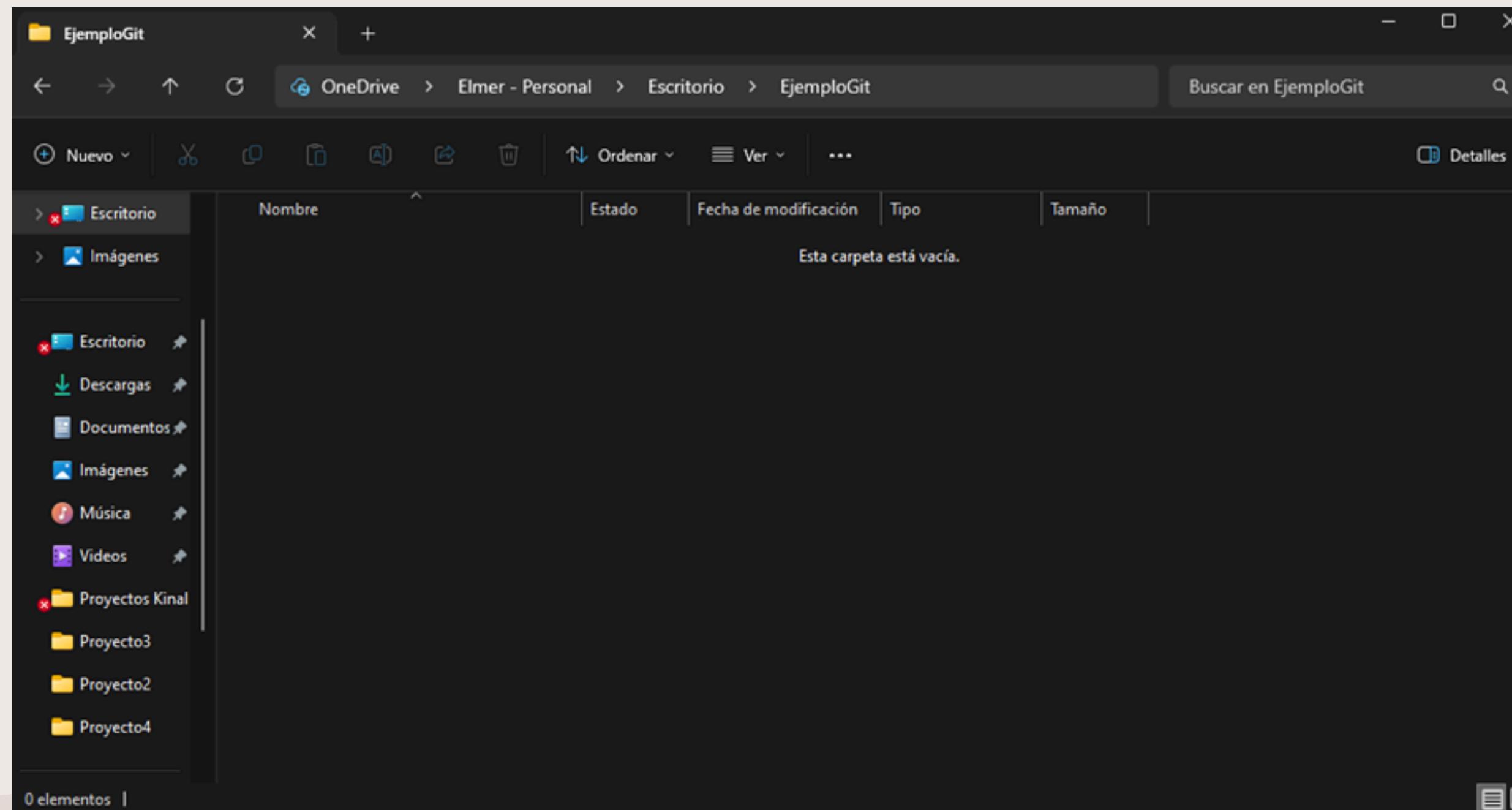
PASO 4

Como en este caso no agregamos nuestro README, no tenemos ningún commit inicial por lo que deberemos crearlo tal cual se nos indica en estos pasos. Tenemos 2 opciones la primera cuando aun no tenemos nuestro repositorio creado y la otra cuando ya tenemos nuestro repositorio creado. En este caso optaremos por la opción 2 ya que no hay diferencia entre ambas.



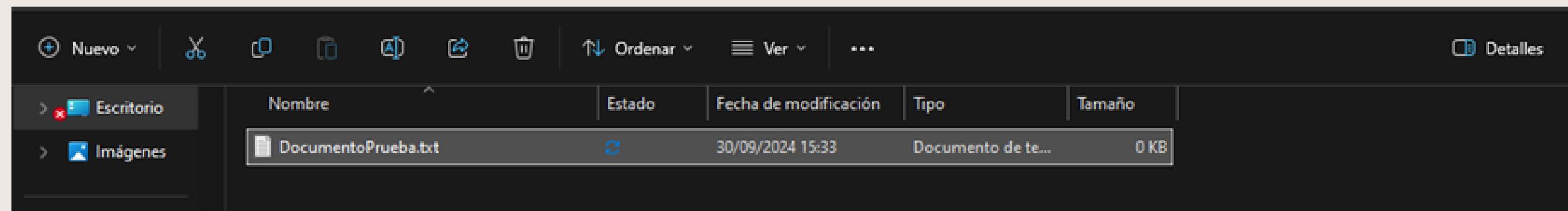
PASO 5

Procederemos a crear nuestro repositorio local en nuestra pc. Para ello nos dirigimos a una carpeta vacía creada en cualquier parte de nuestra pc.



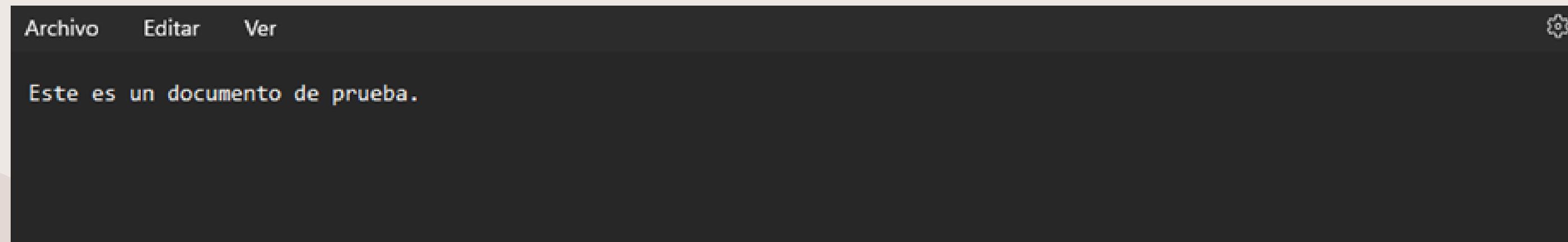
PASO 6

Acá en esta carpeta podemos agregarlos nuestros archivos que deseamos subir.



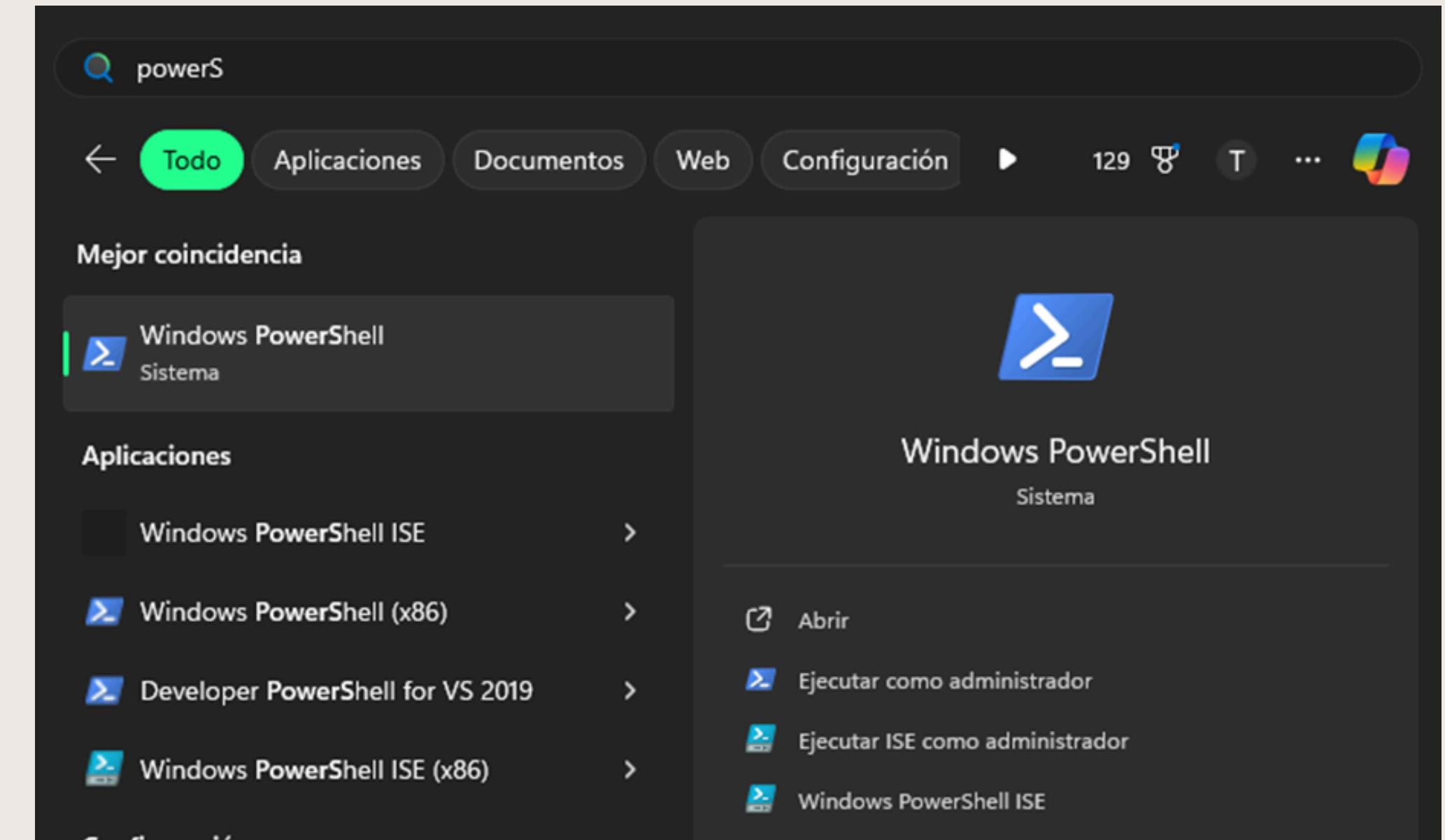
PASO 7

En este caso, como pudimos ver cree un archivo de texto en el cual adentro agregue texto.



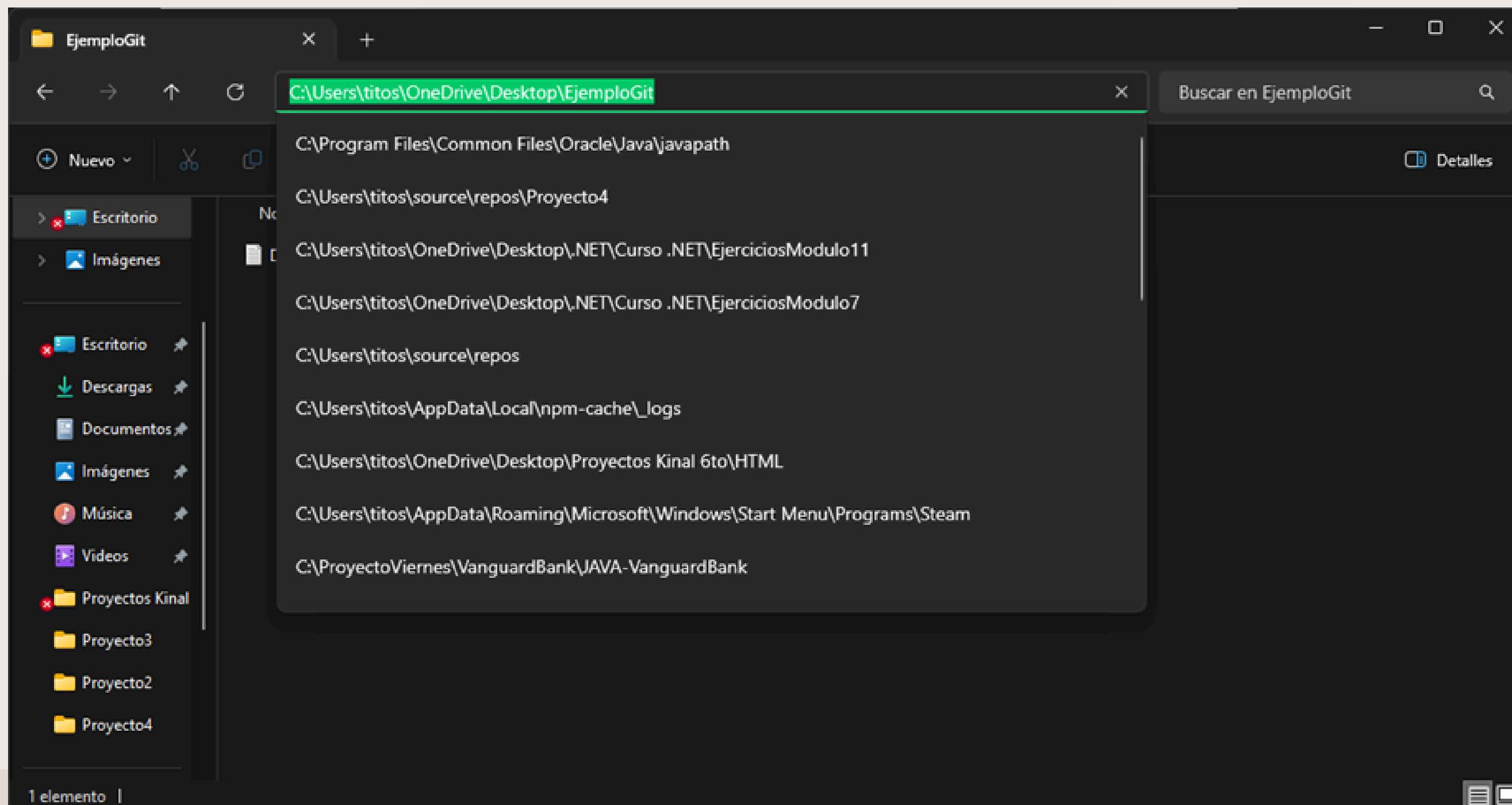
PASO 8

Ahora en nuestro sistema operativo buscaremos cualquier consola que deseamos utilizar, como cmd, cmder, git Bash entre otras. En este caso usaremos PowerShell y procedemos abrir nuestra consola.



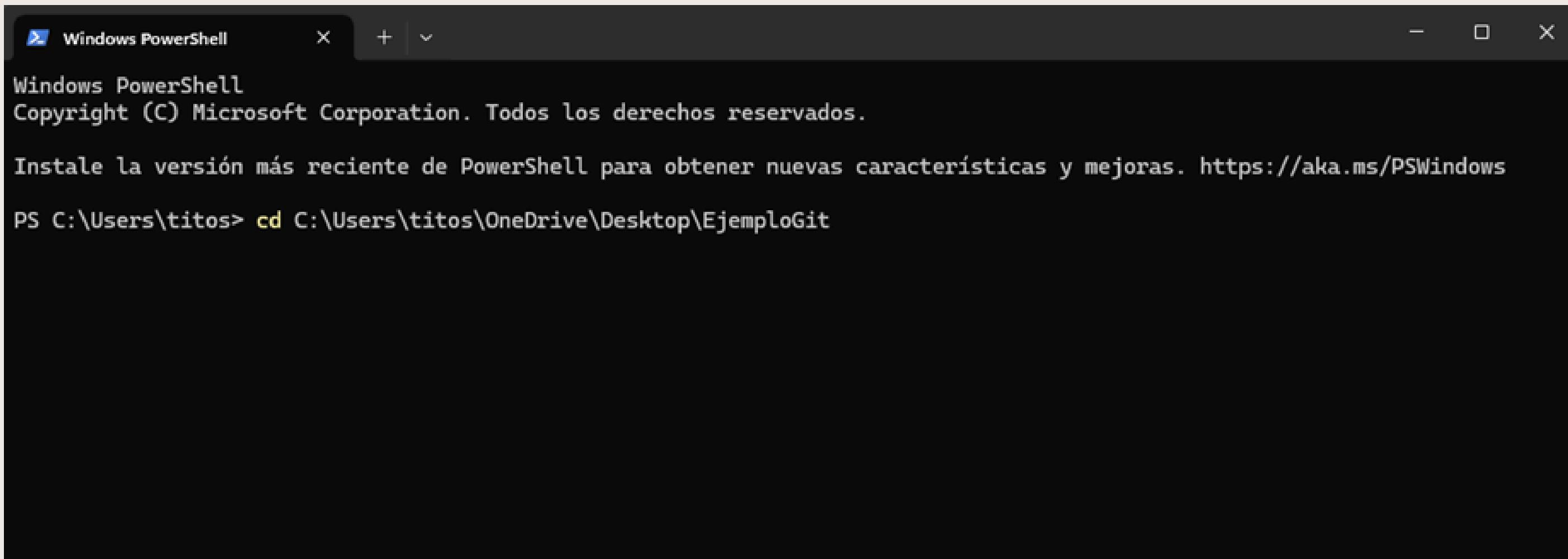
PASO 9

En este paso iniciaremos, copiando la dirección en donde se encuentra ubicada nuestra carpeta, como lo podemos ver a continuación.



PASO 10

Ahora en nuestra consola, colocaremos el siguiente comando: cd “dirección”. Así como miramos a continuación y le damos enter para ubicarnos en nuestra carpeta.



The screenshot shows a Windows PowerShell window with a dark theme. The title bar reads "Windows PowerShell". The console output is as follows:

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

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\titos> cd C:\Users\titos\OneDrive\Desktop\EjemploGit
```

PASO 11

En este paso agregaremos a nuestras credenciales para poder acceder a nuestro repositorios de GitHub y poder subir nuestros repositorios locales. Para ellos colocaremos los siguientes comandos:

- git config --global user.name "nombre"
- git config --global user.email "correo"

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

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\titos> cd C:\Users\titos\OneDrive\Desktop\EjemploGit
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git config --global user.name "esantos2019254"
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git config --global user.email "esantos-2019254@kinal.edu.gt"
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 12

Ahora comprobaremos que nuestras credenciales si estan de manera correcta para ello lo podemos ver con el comando.

- git config --list

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=esantos-2019254@kinal.edu.gt
user.name=esantos2019254
color.ui=auto
credential.helperselector.selected=manager
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> |
```

PASO 13

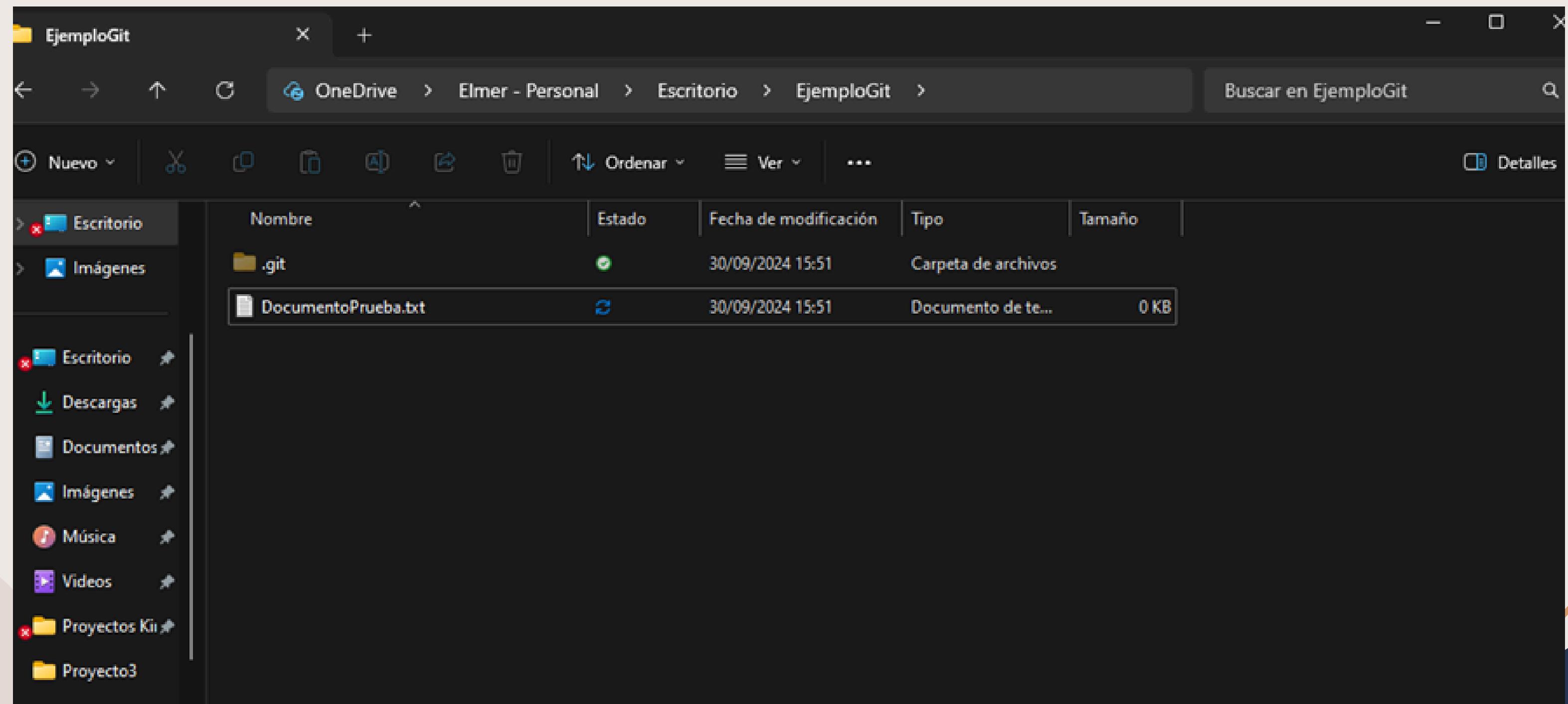
Con todo eso ya podemos iniciar a crear nuestro repositorio local para ellos iniciamos colocando el siguiente comando:

- git init

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git init
Initialized empty Git repository in C:/Users/titos/OneDrive/Desktop/EjemploGit/.git/
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

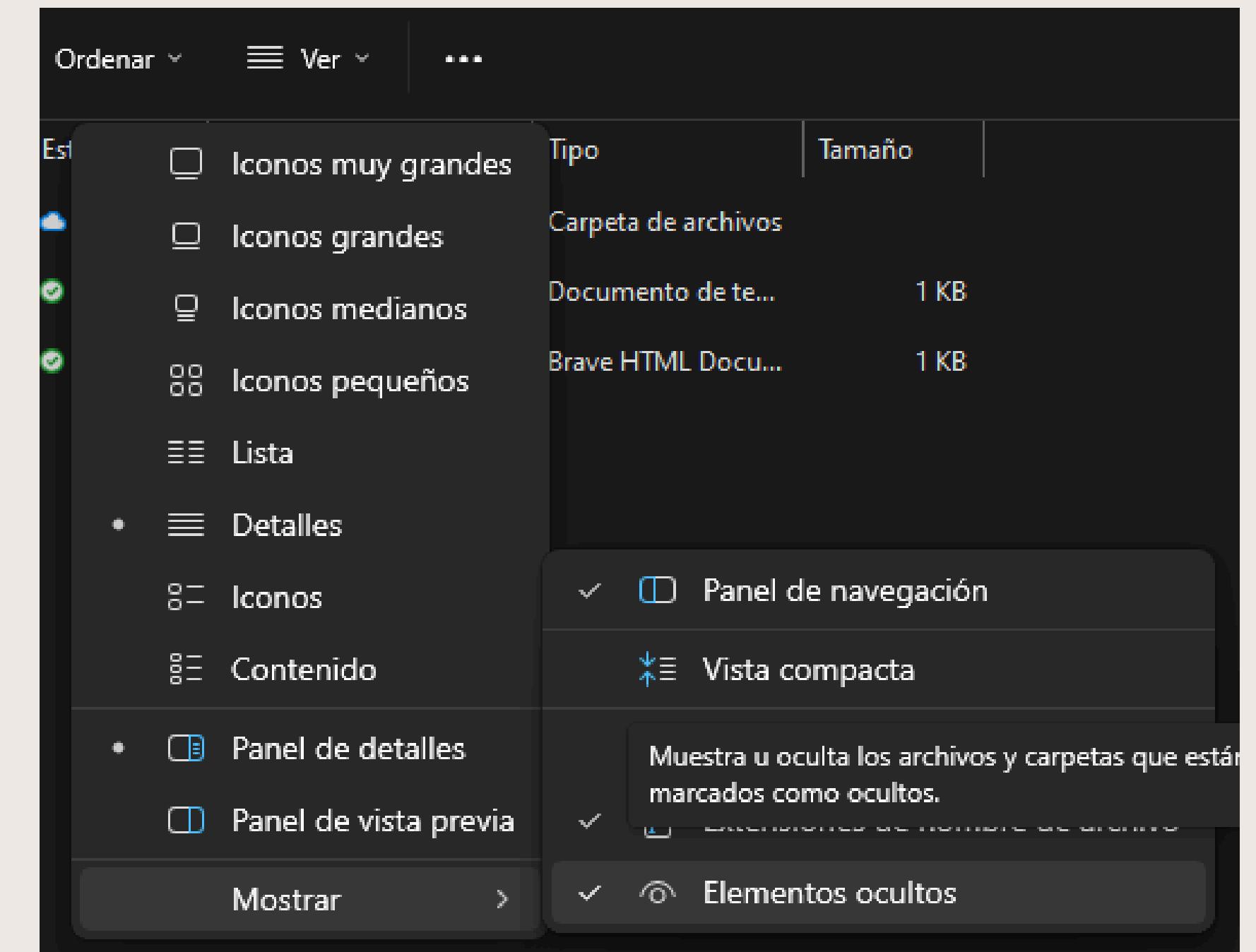
PASO 14

Para ver si colocamos bien el comando y todo esta bien en nuestra carpeta nos tiene que aparecer una carpeta oculta que se llama .git como lo podemos ver a continuación.



PASO 15

En el caso que no nos aparezca la carpeta tenemos que habilitar las carpetas ocultas para esto le daremos click a "ver" luego a "mostrar" y por ultimo "Elementos ocultos"



PASO 16

Ahora vamos a ver el estado de nuestros archivos que están en nuestra carpeta, como tenemos uno, este lo tenemos que añadir para que pueda ser subido. Para poder ver que archivo es el que esta modificado usamos el siguiente comando:

- git status

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 17

Como pudimos ver nuestro archivo esta marcado en rojo, eso quiere decir que fue modificado y necesita ser añadido por ello usamos el siguiente comando:

- git add . (Para añadir todos los archivos)

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
```

PASO 18

Si ahora miramos el estado de nuestro archivo cambia a un color verde, eso quiere decir que fue añadido con éxito.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch master

No commits yet

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

PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 19

En este paso nos queda hacer un “commit” que este se refiere a realizar un comentario acerca de lo que hicimos en el cambio del archivo para ello usamos el siguiente comando:

- git commit -m “comentario”

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "Creando archivo 1"
[master (root-commit) 3906f50] Creando archivo 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 DocumentoPrueba.txt
```

PASO 20

Volvemos hacer un git status para ver como esta nuestro archivo, como podemos ver ya no nos parece nada solo el mensaje de que esta listo para que se pueda subir nuestros cambios.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch master
nothing to commit, working tree clean
```

PASO 21

Ahora ¿Qué pasa si realizamos un nuevo cambio? bueno entonces si lo hacemos pasara lo siguiente:

- Realizamos nuestro cambio.
- Si hacemos un git status otra vez aparece en rojo. (Siguiente diapositiva)
- Eso quiere decir que lo tenemos que añadir de nuevo. (Siguiente diapositiva)
- Procedemos a realizar nuestro comentario del cambio que hicimos. (Siguiente diapositiva)

The screenshot shows a dark-themed code editor window. At the top, there is a menu bar with options "Archivo", "Editar", and "Ver". In the top right corner, there is a small circular icon with a double arrow symbol. The main area of the editor contains two lines of text in a light color:
Este es un documento de prueba.
Esta es un línea nueva.

PASO 22

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> 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:   DocumentoPrueba.txt

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

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "Nueva linea agregada"
[master 44d55c9] Nueva linea agregada
 1 file changed, 3 insertions(+)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 23

Después de realizar nuestros cambios, vamos a ver el historial de commit que hemos echo para ello podemos usar el siguiente comando:

- git log

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git log
commit 82613b3ce1d75631f5adc92d262ca6068ad43419 (HEAD -> master)
Author: esantos2019254 <esantos-2019254@kinal.edu.gt>
Date:   Mon Sep 30 16:08:27 2024 -0600

    Nueva linea agregada

commit 022768c05eb20b7e0a9a94b506e0e29849000c13
Author: esantos2019254 <esantos-2019254@kinal.edu.gt>
Date:   Mon Sep 30 16:07:51 2024 -0600

    Este es un archivo de prueba
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

*Con esto podemos ver que cambios hicimos y si necesitamos regresar a un commit anterior o borrar alguno.(Investiga como realizar esto.)

PASO 24

Ahora usando los comandos que vimos al crear nuestro repositorio los usaremos ahora para poder subir todos nuestros cambios. Recuerda que todo esto que hicimos esta de manera local por lo que ahora debemos subirlo a GitHub. Como ya tenemos creado nuestro repositorio usaremos la segunda opción de comandos.

The screenshot shows the GitHub 'Create repository' wizard with three sections:

- Quick setup — if you've done this kind of thing before**:
 - Set up in Desktop** or **HTTPS** or **SSH**
 - Clipboard icon: <https://github.com/esantos2019254/EjemploGit.git>
 - Text: 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 "# EjemploGit" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/esantos2019254/EjemploGit.git
git push -u origin main
```
- ...or push an existing repository from the command line**:

```
git remote add origin https://github.com/esantos2019254/EjemploGit.git
git branch -M main
git push -u origin main
```

PASO 25

Para poder conectarnos de manera remota a nuestro repositorio de Git Hub usaremos el siguiente comando:

- git remote add origin “url”

Y para poder ver que estamos conectados a nuestro repositorio remoto podemos usar este otro:

- git remote -v

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git remote add origin https://github.com/esantos2019254/EjemploGit.git  
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git remote -v  
origin https://github.com/esantos2019254/EjemploGit.git (fetch)  
origin https://github.com/esantos2019254/EjemploGit.git (push)  
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 26

Ahora crearemos nuestra rama principal sustituyendo a la rama en la que estábamos trabajando por nuestra rama main o master.

- git branch -M “nombre rama”

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch -M main  
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 27

Por ultimo para subir nuestros cambios con el siguiente comando para subir nuestros cambios:

- git push -u origin “rama principal” (establecemos una relación de seguimiento entre la rama local y la remota)
- git push origin “rama” (subir cambios a la rama que deseamos)

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 555 bytes | 277.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/esantos2019254/EjemploGit.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 28

Por último podremos ver en Git hub nuestros cambios y todo lo que realizamos anteriormente.

The screenshot shows a GitHub repository page for 'EjemploGit'. The repository is public and has 1 branch and 0 tags. A recent commit by 'esantos2019254' was pushed 6 minutes ago, adding a new line to 'DocumentoPrueba.txt'. The repository has 2 commits in total. On the right side, there's an 'About' section for 'ejemplo 1 git' with metrics: 0 stars, 1 watching, and 0 forks. Below that are sections for 'Releases' (no releases published) and 'Packages' (no packages published). At the bottom, there's a 'Add a README' button.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

EjemploGit Public

main · 1 Branch 0 Tags

esantos2019254 Nueva linea agregada 62613b3 · 6 minutes ago 2 Commits

DocumentoPrueba.txt Nueva linea agregada 6 minutes ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

Type ⌘ to search

Pin Unwatch Fork Star

About

ejemplo 1 git

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

PASO 29

Ahora vamos a trabajar un nuevo proyecto, para ello usaremos nuestro ide favorito para poderlo hacer, en este caso usaremos Visual Studio Code, una manera de abrir nuestro proyecto desde consola es ingresar el siguiente comando:

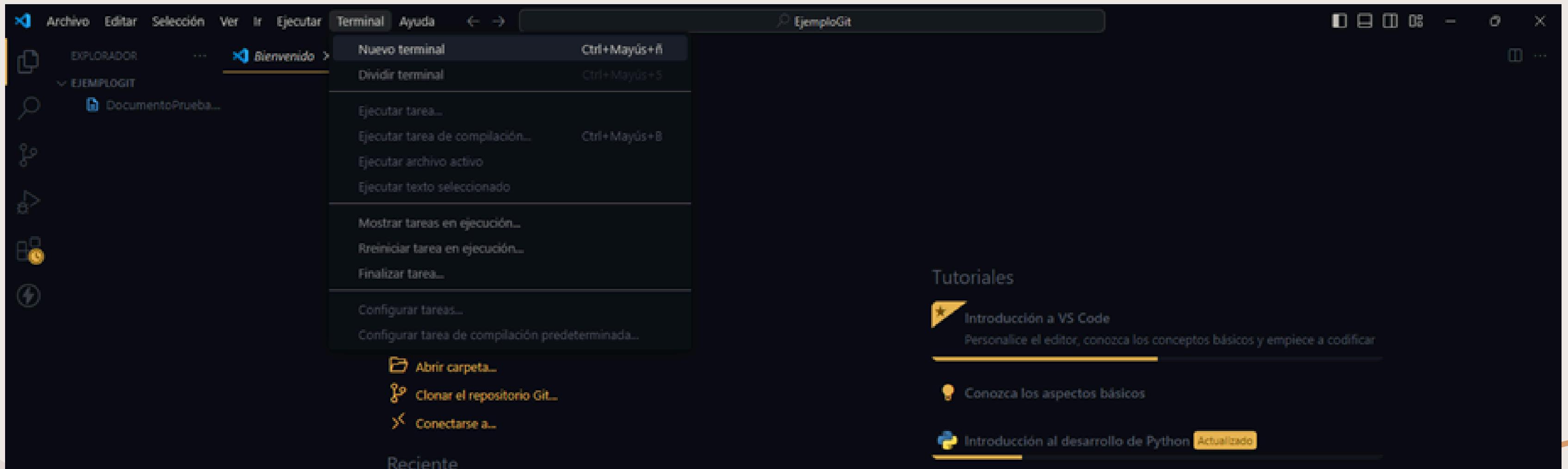
- code .

Ahora para mayor facilidad de trabajo usaremos la consola que nos ofrece visual (es opcional puedes seguir trabajando en otra consola).

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> code .
```

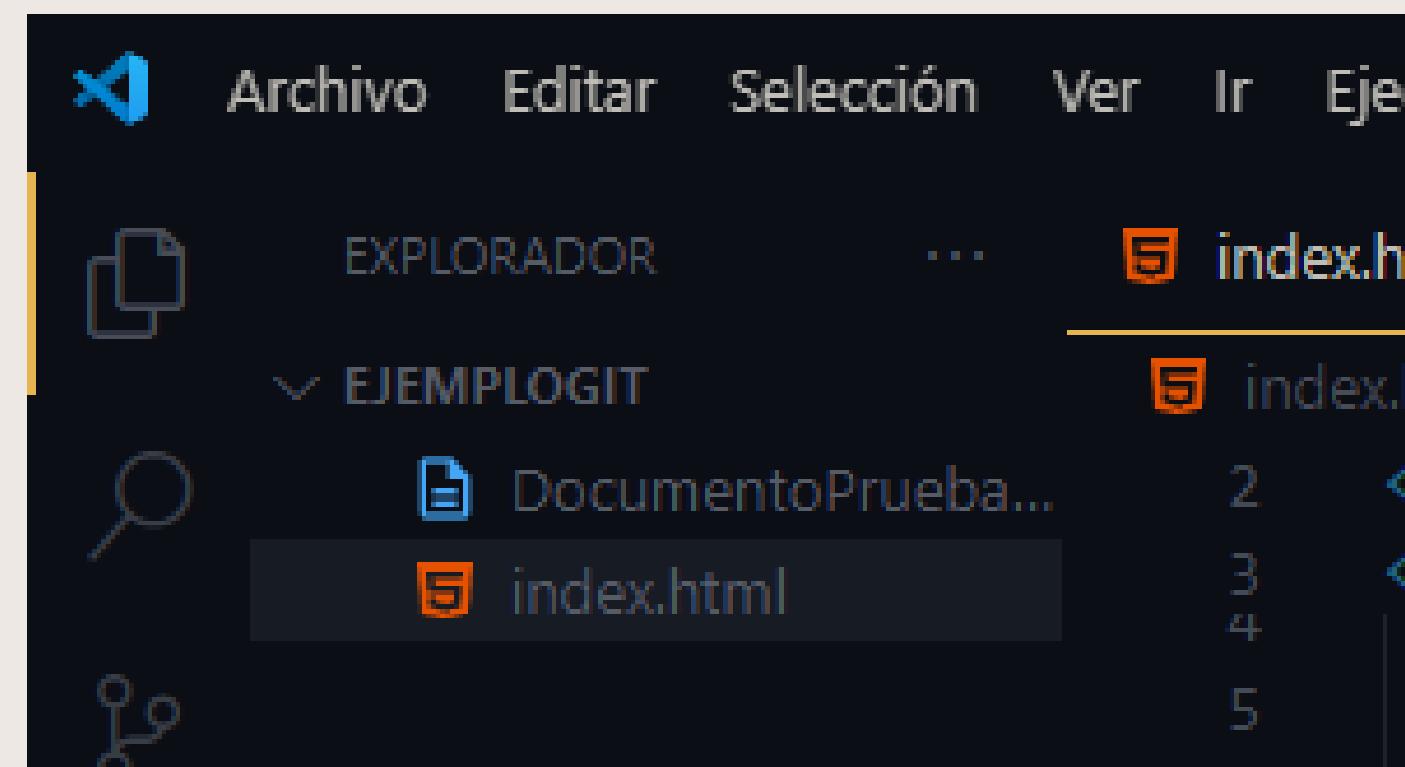
PASO 30

Si no sabes donde se encuentra la consola en visual simplemente dirígete a “terminal” y click a “nuevo terminal”



PASO 31

Después de esto, crearemos nuestro proyecto, para el ejemplo se uso un archivo html pero puede ser algun otro. Despues lo subiremos a nuestro repositorio.



The terminal window shows a PowerShell session with the following commands and output:

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "archivo index.html creado"
[main 5dc0a4e] archivo index.html creado
 1 file changed, 11 insertions(+)
 create mode 100644 index.html
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 449 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/esantos2019254/EjemploGit.git
 82613b3..5dc0a4e main -> main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 32

Después de ver como subir y crear nuestro repositorio remoto y local, vamos a trabajar con el flujo de trabajo de GitFlow, el cual nos va ayudar a poder llevar un mejor orden y control en nuestro proyecto. Para ello el comando inicial para empezar es el siguiente:

- git flow init

Procedemos dar a la tecla enter hasta terminar la ejecución

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/titos/OneDrive/Desktop/EjemploGit/.git/hooks]
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 33

Ahora vamos a observar la nueva rama que se nos creo al ejecutar el comando anterior para ello vamos a ingresar:

- git branch (se usa para ver las ramas existentes y en cual se esta ubicado).

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main
```

PASO 34

Ahora que ya estamos en la rama develop apartir de alli ya podremos hacer todos los cambios en nuestros proyectos que estemos trabajando entonces para esto vamos a iniciar usando este comando:

- git flow feature start "nombre-de-la-feature"

Usaremos features para realizar cualquier funcionalidad en nuestros proyectos para que luego sea unido a la rama develop

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow feature start title
Switched to a new branch 'feature/title'
```

Summary of actions:

- A new branch 'feature/title' was created, based on 'develop'
- You are now on branch 'feature/title'

Now, start committing on your feature. When done, use:

```
git flow feature finish title
```

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 35

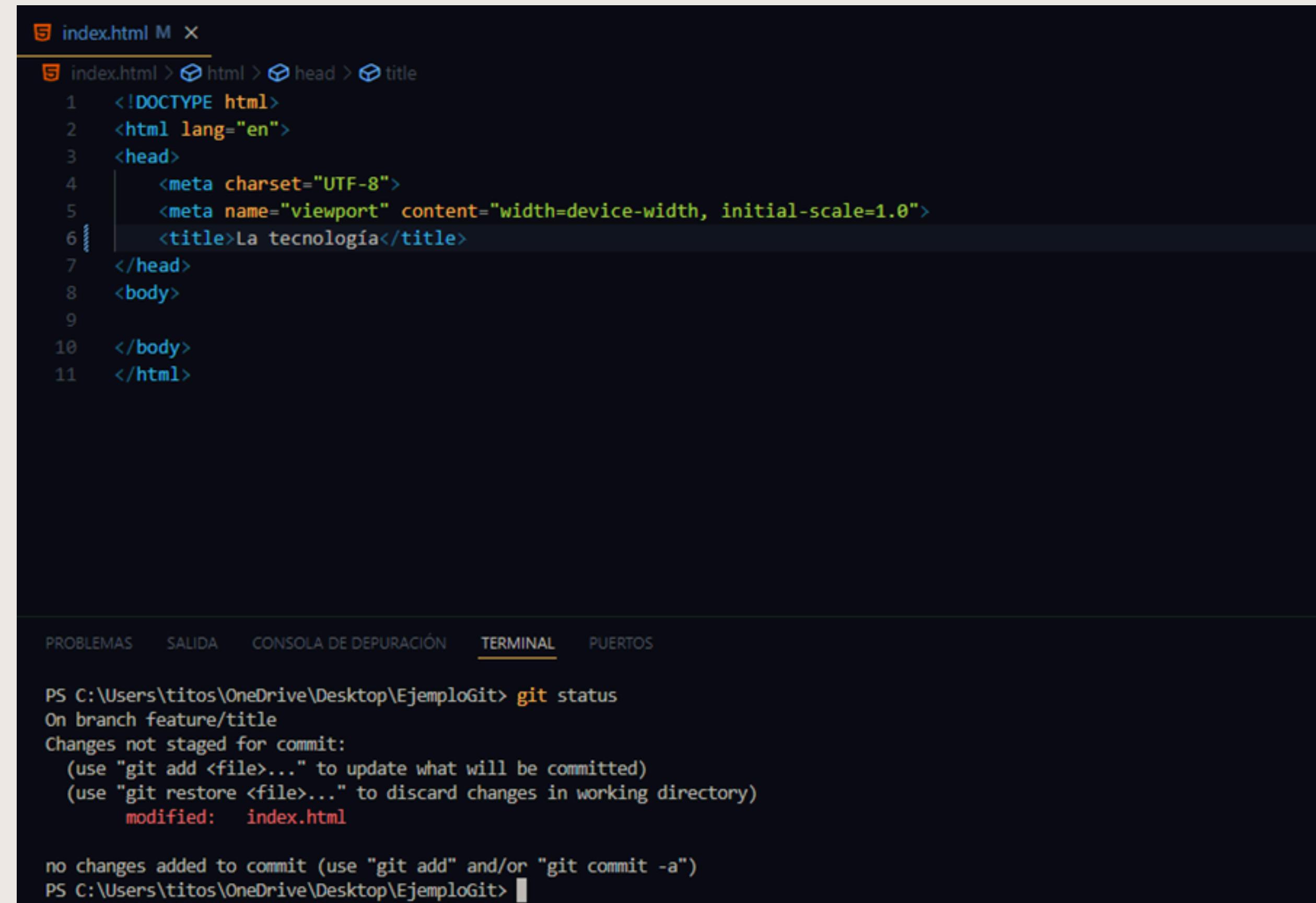
Hecho lo anterior, podemos ver de nuevo que se creo una nueva rama para ello ejecutamos el comando ya visto.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
  develop
* feature/title
  main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 36

Ya estando en nuestra rama a trabajar, realizaremos todos los cambios que corresponden a la funcionalidad a hacer.

Por ultimo realizamos un status y como podemos ver hay cambios que hay que agregar y subir.



The screenshot shows a terminal window with the following content:

```
index.html M X
index.html > html > head > title
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>La tecnología</title>
7  </head>
8  <body>
9
10 </body>
11 </html>

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch feature/title
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 37

Ahora vamos a agregar nuestros cambios que hicimos y hacemos nuestro commit describiendo lo que hicimos.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "Titulo modificado"
[feature/title 84a5af] Titulo modificado
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 38

Por ultimo, terminado de trabajar en esta rama vamos a finalizar para ellos usamos el siguiente comando:

- git flow feature finish "nombre-de-la-feature"

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow feature finish title
Switched to branch 'develop'
Updating 5dc0a4e..84a5afd
Fast-forward
  index.html | 2 +-  
  1 file changed, 1 insertion(+), 1 deletion(-)
Deleted branch feature/title (was 84a5afd).
```

Summary of actions:

- The feature branch 'feature/title' was merged into 'develop'
- Feature branch 'feature/title' has been locally deleted
- You are now on branch 'develop'

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

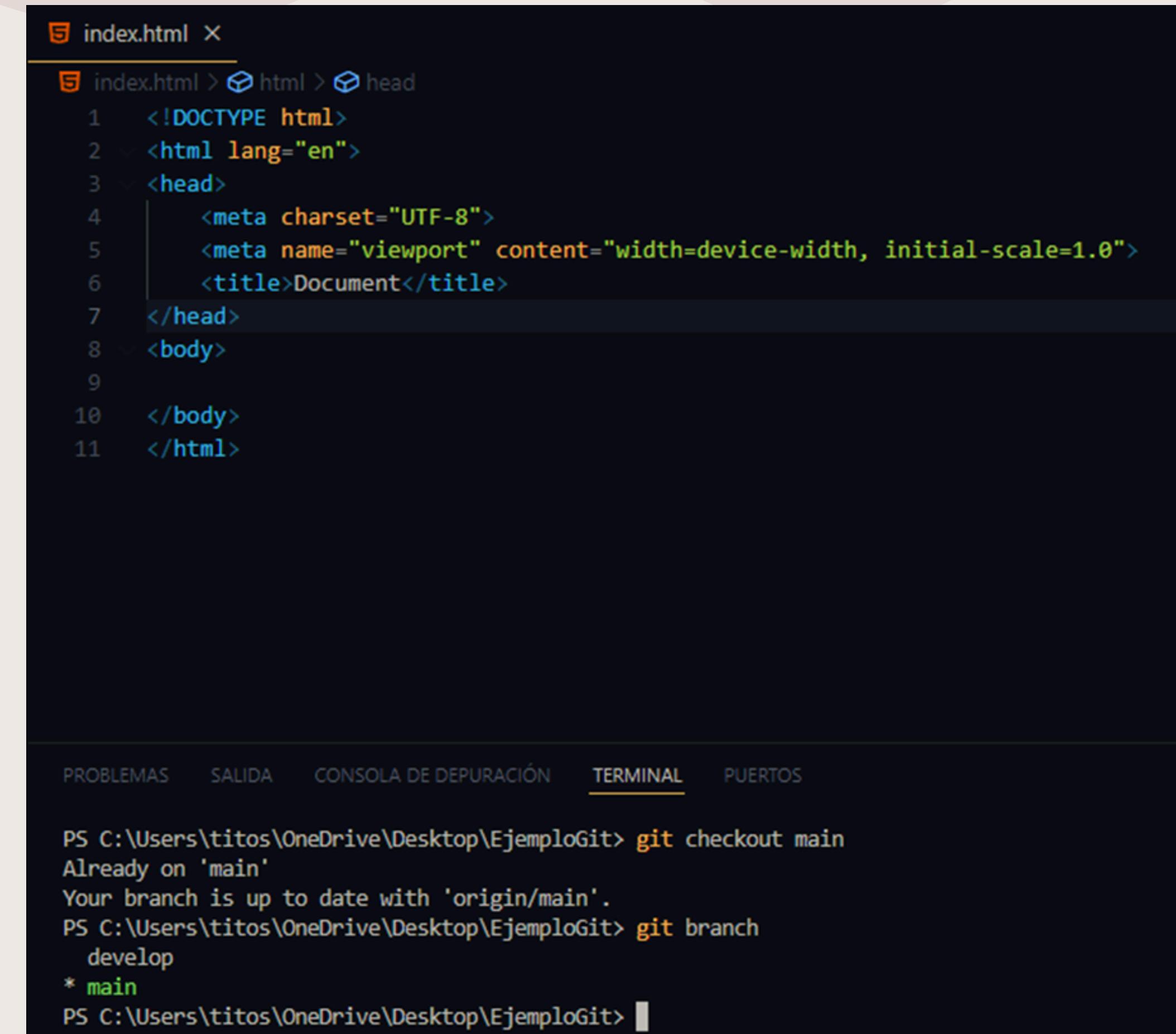
PASO 39

Ahora si observamos con git branch la rama feature a desparecido, esto porque se fusiono con la rama develop.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main
```

PASO 40

Ahora nos cambiaremos a la rama principal para ver que los cambios solo se realizaron en la rama develop y no en la rama principal, que esto es el objetivo de este flujo de trabajo.



The screenshot shows a terminal window with the following content:

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

Below the code editor, the terminal tab is selected, showing the following command history:

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
  develop
* main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 41

Ahora regresamos a nuestra rama develop y si observamos el archivo cambio a lo que hicimos anteriormente.

The screenshot shows a terminal window with the following content:

```
index.html ×
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>La tecnología</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Below the code editor, the terminal tab is selected, showing the following command history:

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git checkout develop
Switched to branch 'develop'
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 42

Visto lo anterior es momento de subir nuestros cambios a nuestro repositorio para ello usaremos el comando

- git push origin “rama”

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 340 bytes | 170.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/esantos2019254/EjemploGit/pull/new/develop
remote:
To https://github.com/esantos2019254/EjemploGit.git
 * [new branch]      develop -> develop
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 43

Después de esto podemos ver los cambios subidos a nuestro repositorio.

The screenshot shows a GitHub repository page for 'EjemploGit'. At the top, there's a banner indicating 'develop had recent pushes 46 seconds ago'. Below this, the repository name 'EjemploGit' is shown as 'Public'. The main area displays a list of recent commits:

- esantos2019254 Titulo modificado - 84a5af0 · 5 minutes ago · 4 Commits
- DocumentoPrueba.txt Nueva linea agregada - 17 hours ago
- index.html Titulo modificado - 5 minutes ago

On the right side, there are sections for 'About', 'Activity', 'Contributors', 'Releases', 'Packages', and 'Languages'. The 'Languages' section shows HTML at 100.0%. A large orange button at the bottom encourages users to 'Add a README'.

PASO 44

Ahora vamos a hacer el uso de la rama release que nos sirve, para corregir errores y revisión antes que el proyecto sea subido a producción. Para ello primero debemos estar en la rama develop para ejecutar el siguiente comando:

- git flow release start “nombre-de-la-release”

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main

PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow release start v1.0.0
Switched to a new branch 'release/v1.0.0'

Summary of actions:
- A new branch 'release/v1.0.0' was created, based on 'develop'
- You are now on branch 'release/v1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

  git flow release finish 'v1.0.0'

PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 45

Ahora si observamos una nueva rama a sido creada. Con ello podemos pasar a realizar los cambios que se consideren en nuestro proyecto

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
  develop
  main
* release/v1.0.0
```



The screenshot shows a terminal window at the top and a code editor window below it.

Terminal:

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git status
On branch release/v1.0.0
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

Code Editor:

```
index.html M X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>La tecnología</title>
7  </head>
8  <body>
9    <h1>Aprendiendo GitFlow</h1>
10 </body>
11 </html>
```

PASO 46

Como pudimos ver ahora hay cambios que debemos de agregar para ello lo haremos como ya lo hemos visto.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "Agregando un h1 par mejor comprensión"
[release/v1.0.0 7c94c12] Agregando un h1 par mejor comprensión
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 47

Despues de los visto finalizaremos nuestra rama, para ello ejecutamos el siguiente comando:

- git flow release finish "nombre-de-la-release"

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow release finish v1.0.0
```

PASO 48

Ahora nos aparecerá la siguiente información en donde podremos agregar un comentario, para ello precionaremos la tecla “i” ingresaremos nuestro comentario y para salir le daremos a la tecla “esc” e ingresamos la combinación:

- :wq

Por ultimo presionamos enter.

PASO 49

Como podemos ver ahora nuestros cambios se reflejaron tanto en la rama main como en la develop, eso quiere decir que todo se realiza de manera correcta.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow release finish v1.0.0
Branches 'main' and 'origin/main' have diverged.
And local branch 'main' is ahead of 'origin/main'.
Already on 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)
Switched to branch 'develop'
Merge made by the 'ort' strategy.
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Deleted branch release/v1.0.0 (was 7c94c12).
```

Summary of actions:

- Release branch 'release/v1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/v1.0.0' has been locally deleted
- You are now on branch 'develop'

```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>La tecnología</title>
7  </head>
8  <body>
9      <h1>Aprendiendo GitFlow</h1>
10 </body>
11 </html>
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>La tecnología</title>
7  </head>
8  <body>
9      <h1>Aprendiendo GitFlow</h1>
10 </body>
11 </html>
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 50

Por último subiremos nuestros cambios para ello observamos que primero estemos en nuestra rama develop y lo subiremos.

```
PROBLEMAS    SALIDA    CONSOLA DE DEPURACIÓN    TERMINAL    PUERTOS

PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> █
```

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin develop
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 686 bytes | 98.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/esantos2019254/EjemploGit.git
  84a5af..6f51af3  develop -> develop
```

PASO 51

Ahora nos cambiaremos a nuestra rama principal y subiremos los cambios en esta rama. Para este ultimo usaremos el siguiente comando:

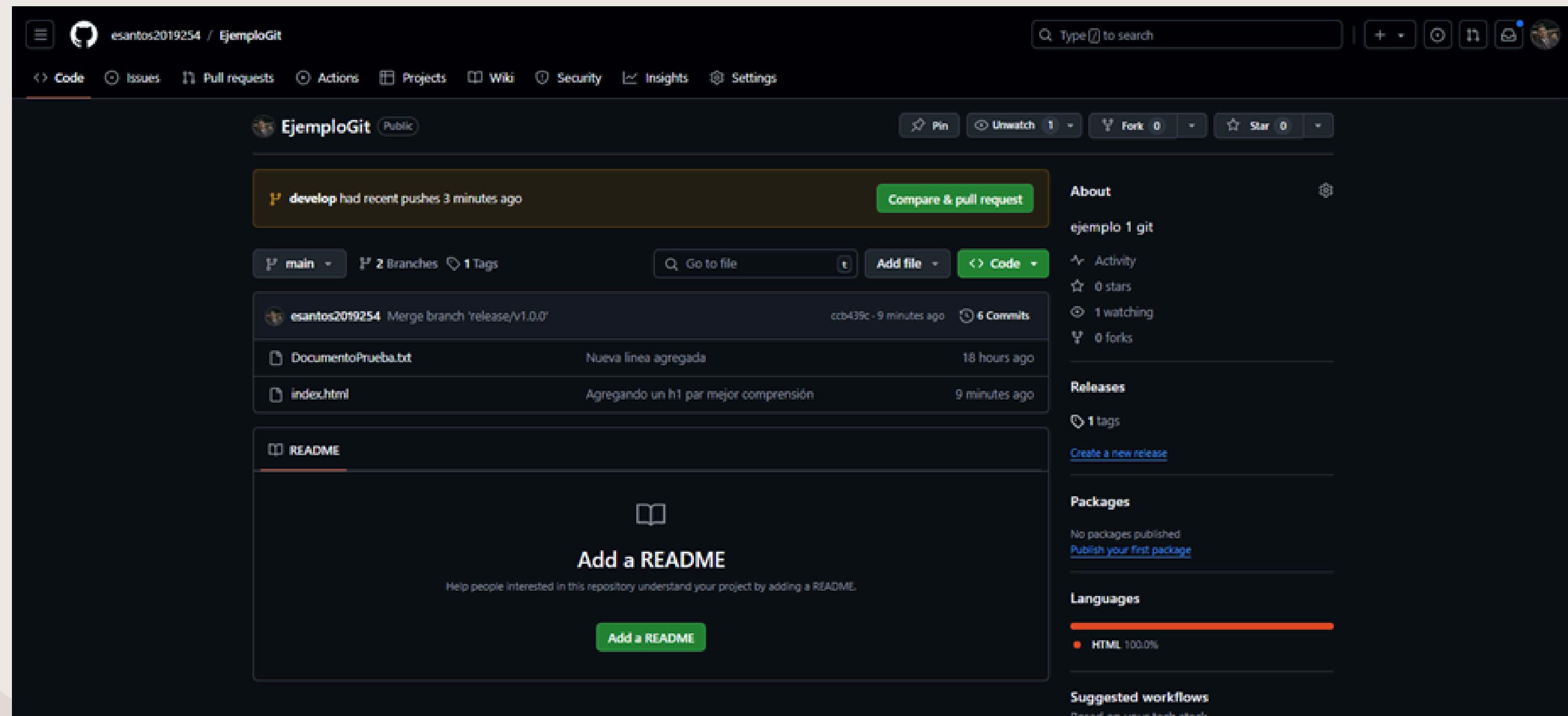
- git push origin “rama” --tags (indica que también se deben subir las etiquetas (tags) locales al repositorio remoto.)

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin main --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 166 bytes | 83.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/esantos2019254/EjemploGit.git
 * [new tag]          v1.0.0 -> v1.0.0
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 52

Ahora ya podemos ir a nuestro repositorio y ver todos los cambios subidos.



PASO 53

Por ultimo usaremos la rama hotfix, la cual se usa para corregir algun bug o error ya en producción, para ello primero debemos estar en la rama principal y luego ejecutaremos el siguiente comando:

- git flow hotfix start "nombre-de-la-hotfix"

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
  develop
* main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow hotfix start v1.0.1
Switched to a new branch 'hotfix/v1.0.1'

Summary of actions:
- A new branch 'hotfix/v1.0.1' was created, based on 'main'
- You are now on branch 'hotfix/v1.0.1'

Follow-up actions:
- Start committing your hot fixes
- Bump the version number now!
- When done, run:

  git flow hotfix finish 'v1.0.1'

PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 54

Como podemos ver ahora se nos creo nuestra rama. por lo que ahora ya podemos corregir algún problema que tengamos en nuestro proyecto. Entonces procedemos a realizar este.

The screenshot shows a code editor with a dark theme and a terminal window below it.

Code Editor (index.html):

```
index.html M X
index.html > html > body > h1
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" >
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" >
6     <title>La tecnología</title>
7   </head>
8   <body>
9     <h1>Aprendiendo GitFlow</h1>
10    <h1>Corrigiendo Errores</h1>
11  </body>
12 </html>
```

Terminal:

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
  develop
* hotfix/v1.0.1
  main
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> |
```

PASO 55

Hecho nuestros cambios si miramos hemos hecho cambios por lo que debemos de añadirlos de nuevo, con lo que procedemos a realizar lo visto.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git add .
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git commit -m "Corrigiendo errores en producción"
[hotfix/v1.0.1 42cd8ce] Corrigiendo errores en producción
 1 file changed, 1 insertion(+)
```

PASO 56

Ahora hecho lo anterior vamos a terminar de usar nuestra rama por lo que ejecutaremos el siguiente comando:

- git flow hotfix finish "nombre-de-la-hotfix"

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow hotfix finish v1.0.1
```

PASO 57

Ahora procederemos a agregar nuestro comentario, como vimos antes (presionar tecla “i”). Por último presionar “esc” seguido de :wq y enter para salir.

PASO 58

Por ultimo podemos ver que nuestros cambios se reflejaron en la rama develop como en la main.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git flow hotfix finish v1.0.1
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Merge made by the 'ort' strategy.
 index.html | 1 +
  1 file changed, 1 insertion(+)
Switched to branch 'develop'
Merge made by the 'ort' strategy.
 index.html | 1 +
  1 file changed, 1 insertion(+)
Deleted branch hotfix/v1.0.1 (was 42cd8ce).

Summary of actions:
- Hotfix branch 'hotfix/v1.0.1' has been merged into 'main'
- The hotfix was tagged 'v1.0.1'
- Hotfix tag 'v1.0.1' has been back-merged into 'develop'
- Hotfix branch 'hotfix/v1.0.1' has been locally deleted
- You are now on branch 'develop'

PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 59

Ahora debemos subir nuestros cambios en ambas ramas por lo que primero nos cambiamos a develop y subimos nuestros cambios.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git branch
* develop
  main

PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin develop
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 693 bytes | 77.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/esantos2019254/EjemploGit.git
  6f51af3..b8a1796  develop -> develop
```

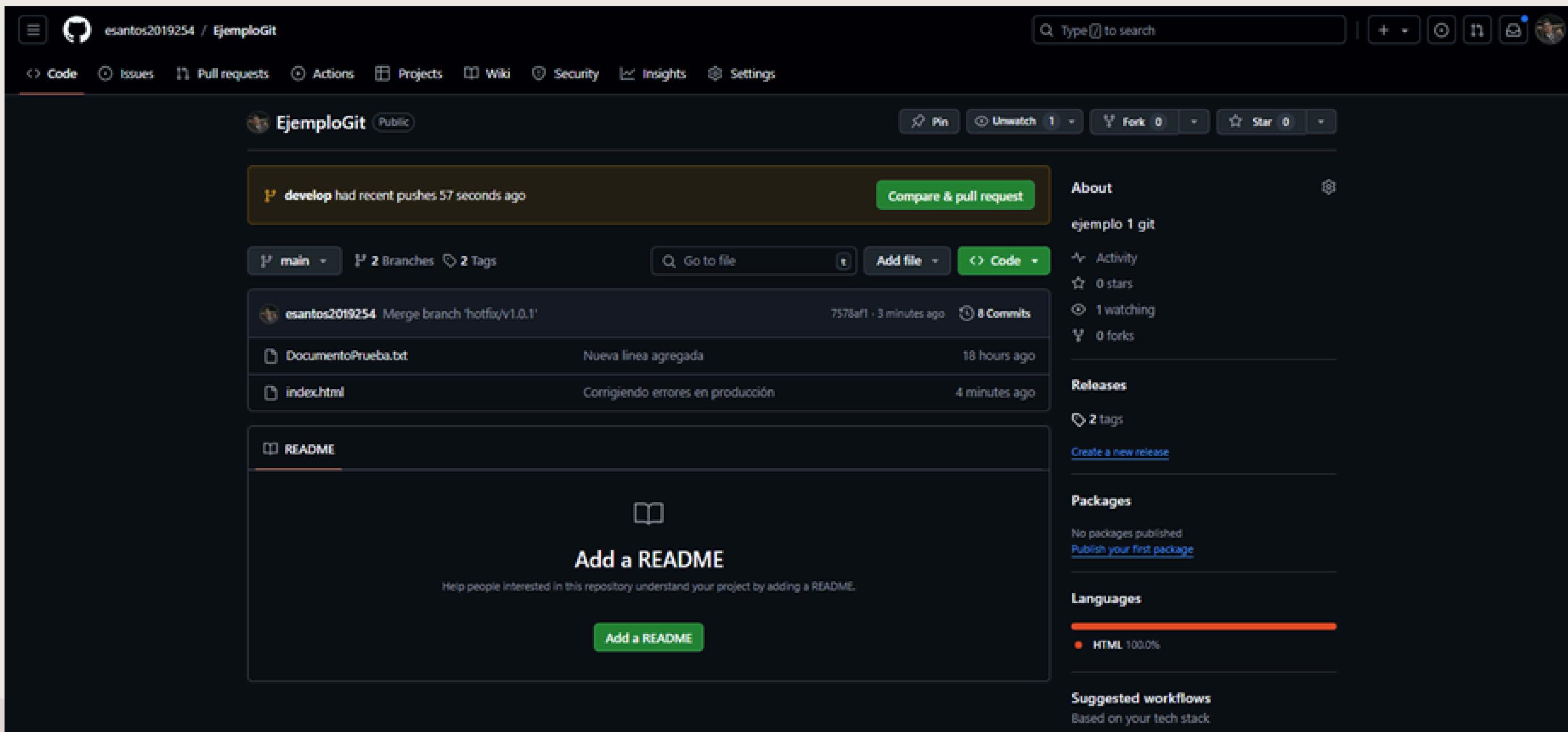
PASO 60

Procedemos de igual manera a cambiarnos a la rama main para subir nuestros cambios.

```
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
PS C:\Users\titos\OneDrive\Desktop\EjemploGit> git push origin main --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 170 bytes | 85.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/esantos2019254/EjemploGit.git
  ccb439c..7578af1  main -> main
* [new tag]          v1.0.1 -> v1.0.1
PS C:\Users\titos\OneDrive\Desktop\EjemploGit>
```

PASO 61

Por último ya podemos ver nuestros cambios hechos en nuestro repositorio.



MUCHAS
GRACIAS

