

Angel Adolfo Méndez Orozco



Manual GitFlow



Paso 1: Instalar Git



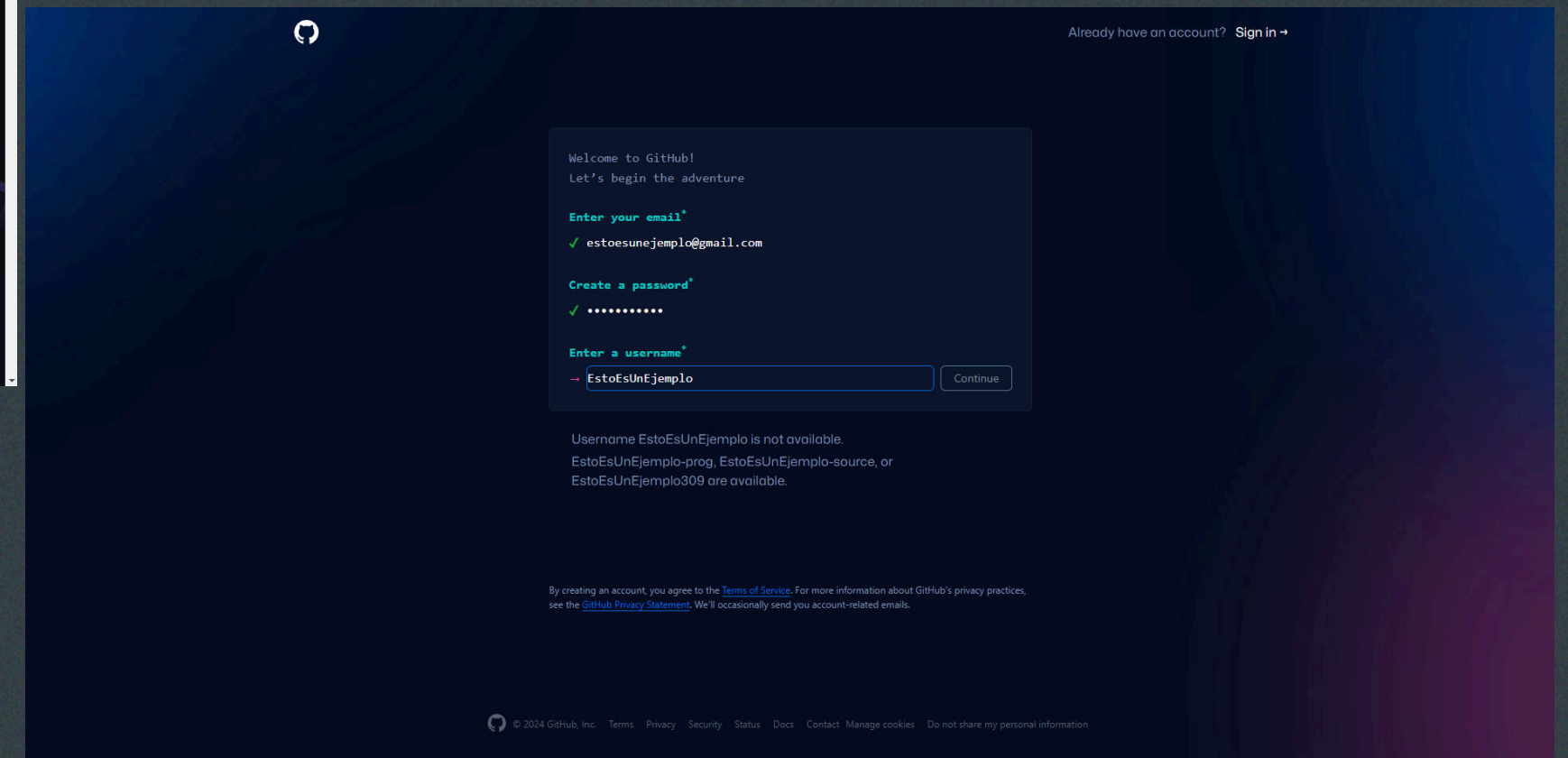
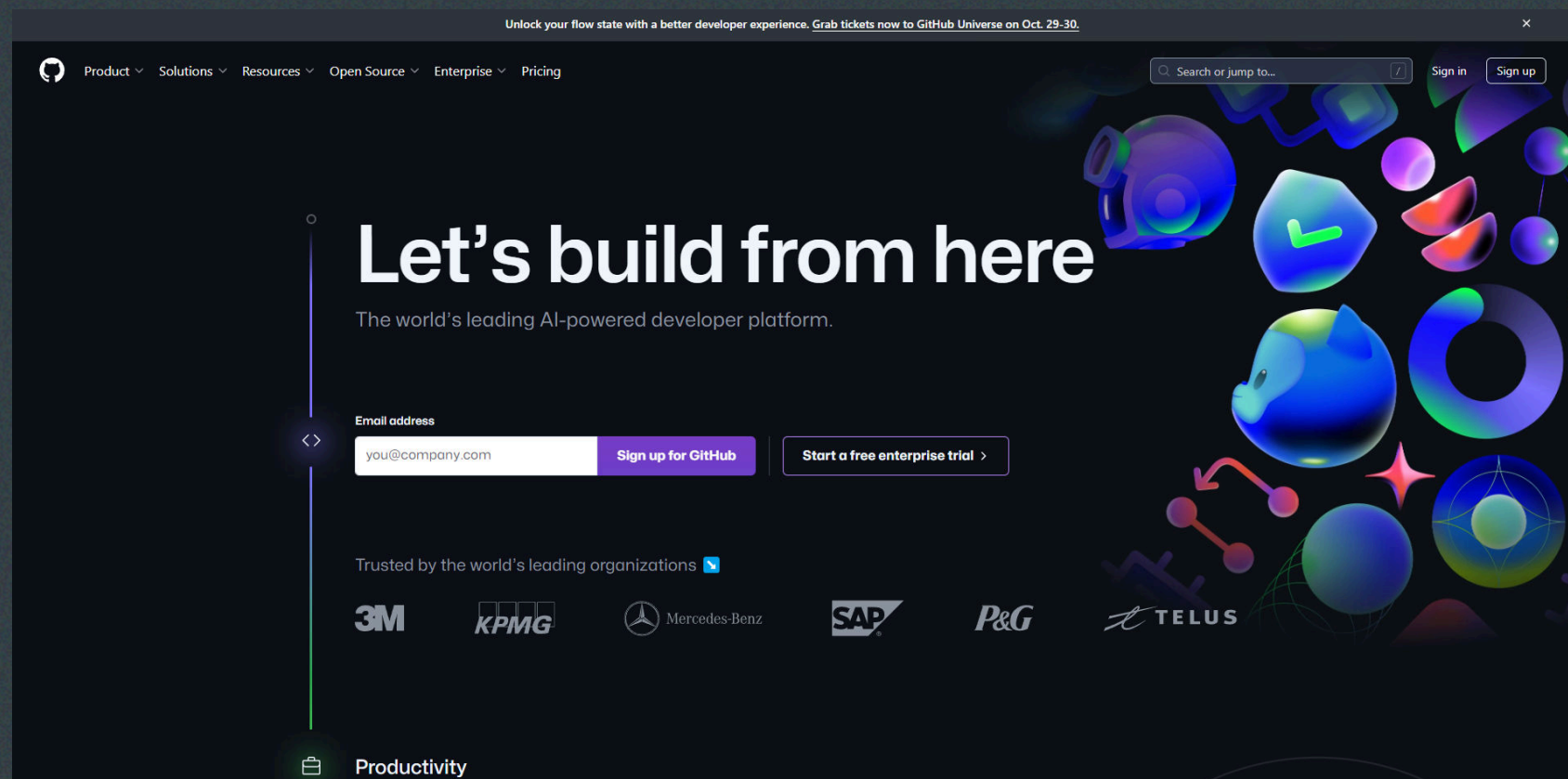
- Ve a "git-scm.com" y descarga la versión de Git para tu sistema operativo (Windows, macOS, o Linux).
- Sigue las instrucciones de instalación. Si no sabes cómo, selecciona las opciones predeterminadas en cada paso.
- Para verificar que Git se ha instalado correctamente, abre la consola de comandos y escribe: "git --version"

```
Símbolo del sistema
C:\Users\Angel>git --version
git version 2.43.0.windows.1
C:\Users\Angel>
```


Paso 2: Crear una cuenta en GitHub



- Entra a "github.com"
- Haz clic en Sign Up o Registrarse.
- Completa los campos solicitados: correo electrónico, nombre de usuario y contraseña.
- Confirma tu correo electrónico para activar tu cuenta.



Paso 3: Configurar Git



Abre tu consola (puede ser cmd en Windows, Terminal en macOS o Linux).

Configura tu nombre de usuario y correo electrónico en Git con los siguientes comandos:

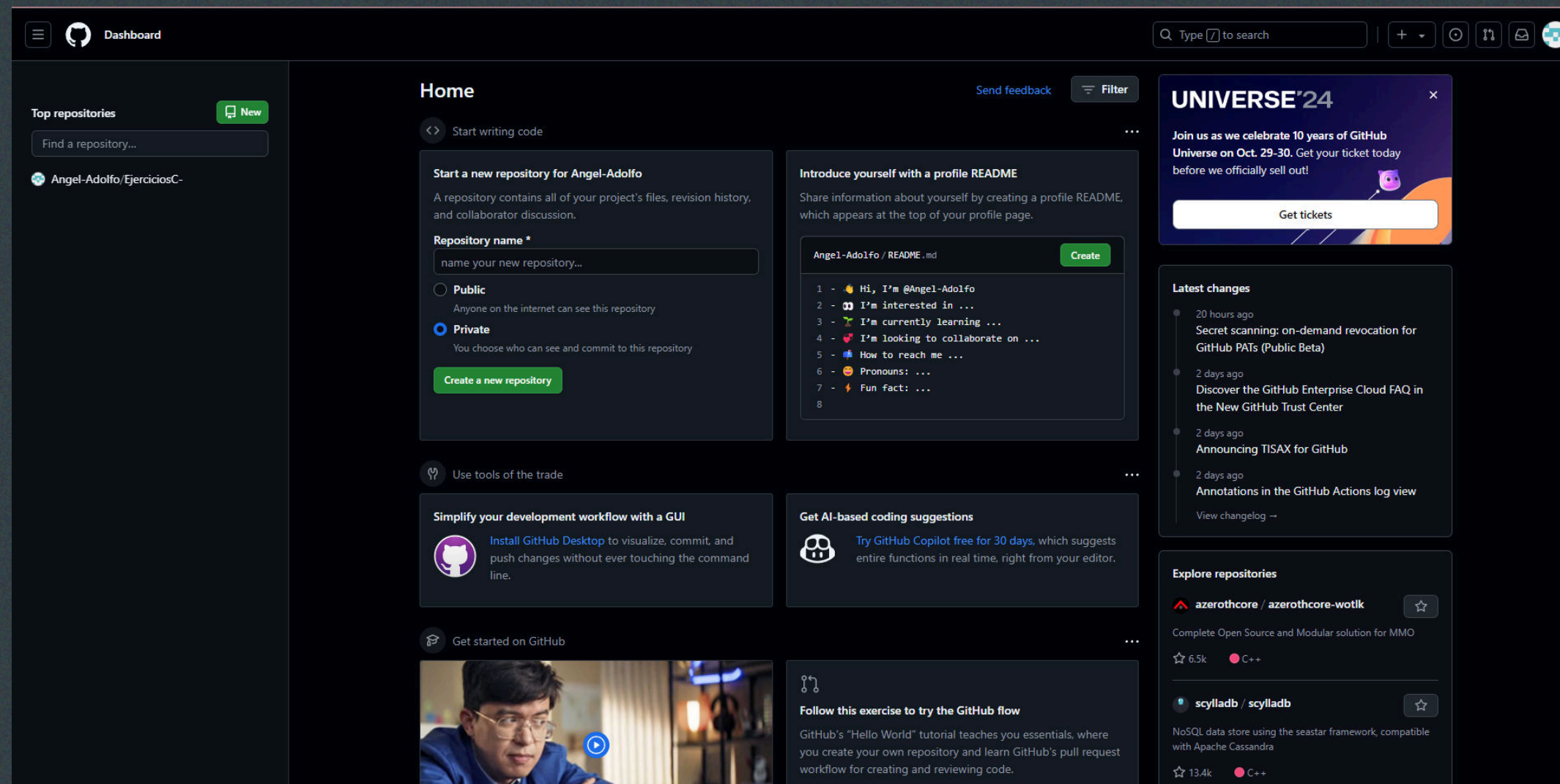
1. "git config --global user.name Tu Nombre"
2. "git config --global user.email tu-email@ejemplo.com"

```
Símbolo del sistema
C:\Users\Angel>git config --global user.name "EstoEsUnEjemplo"
C:\Users\Angel>git config --global user.email "estoesunejemplo@gmail.com"
```


Paso 4: Crear repositorio en GitHub



- Entra a tu cuenta de GitHub.
- Haz clic en el ícono de tu avatar en la parte superior derecha y selecciona Your repositories.
- Haz clic en el botón New para crear un nuevo repositorio.
- Llena el nombre del proyecto, agrega una descripción si lo deseas, y selecciona si será público o privado.
- Marca la opción para añadir un archivo README si quieres.
- Haz clic en Create repository para finalizar.



Paso 4: Crear repositorio en GitHub




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 *

 Angel-Adolfo ▾

Repository name *

EstoEsUnEjemplo

✔ EstoEsUnEjemplo is available.

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

Description (optional)

Aquí va una descripción



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.](#)

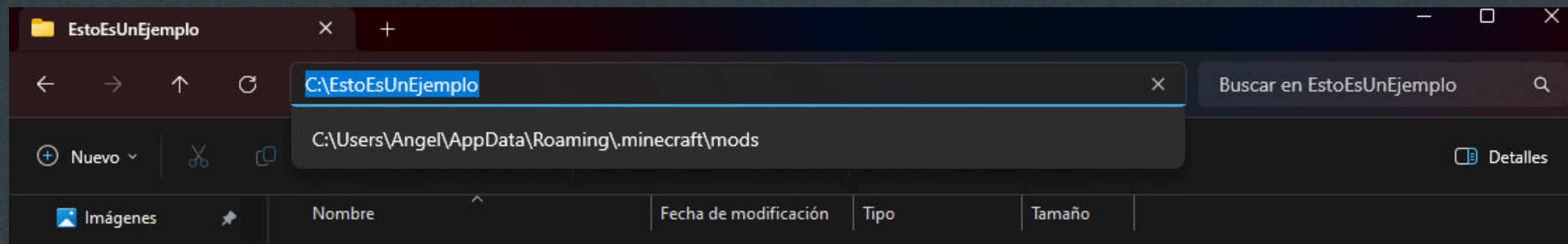
This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

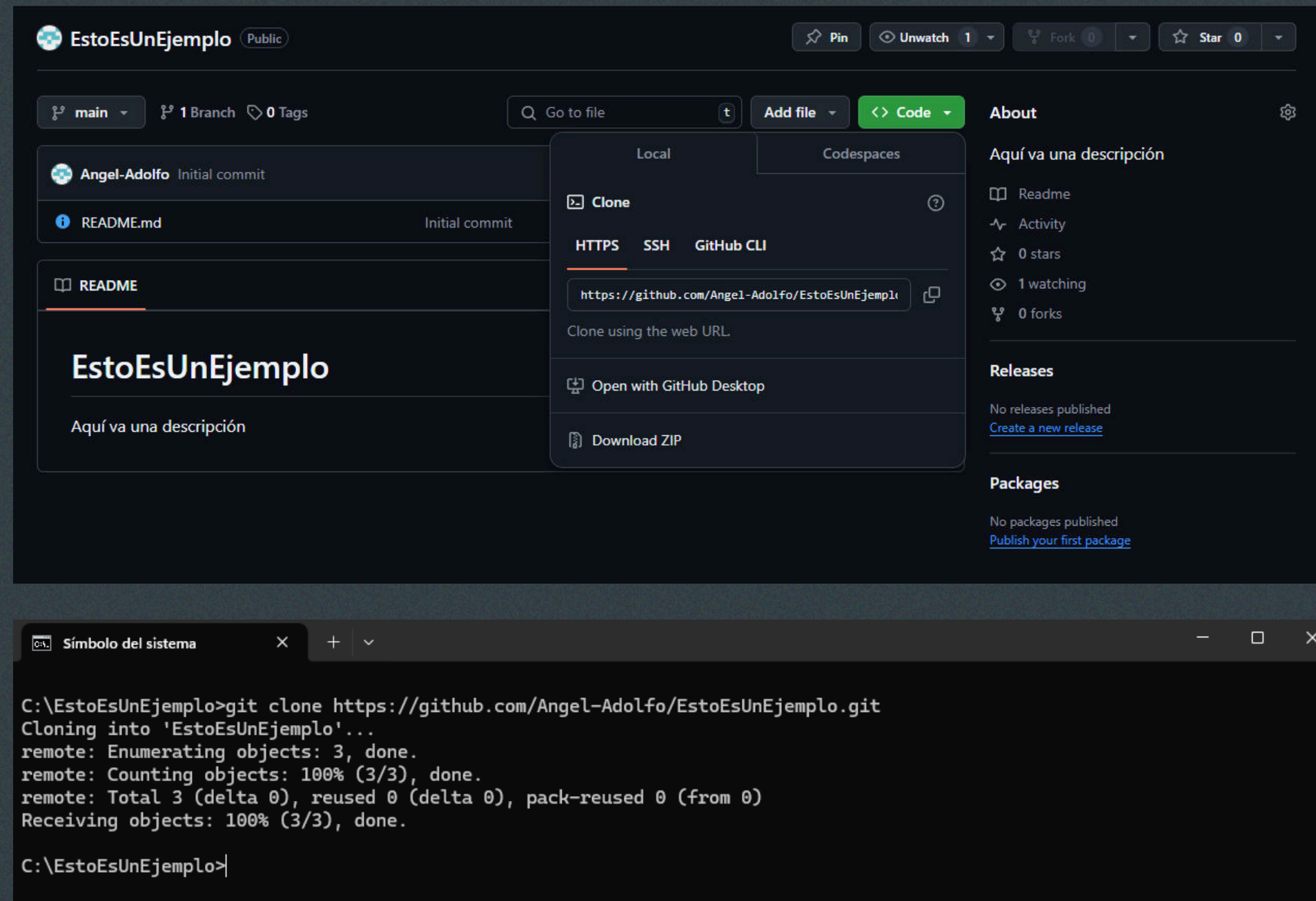
Create repository

Paso 5: Clonar el Repositorio a tu Computadora

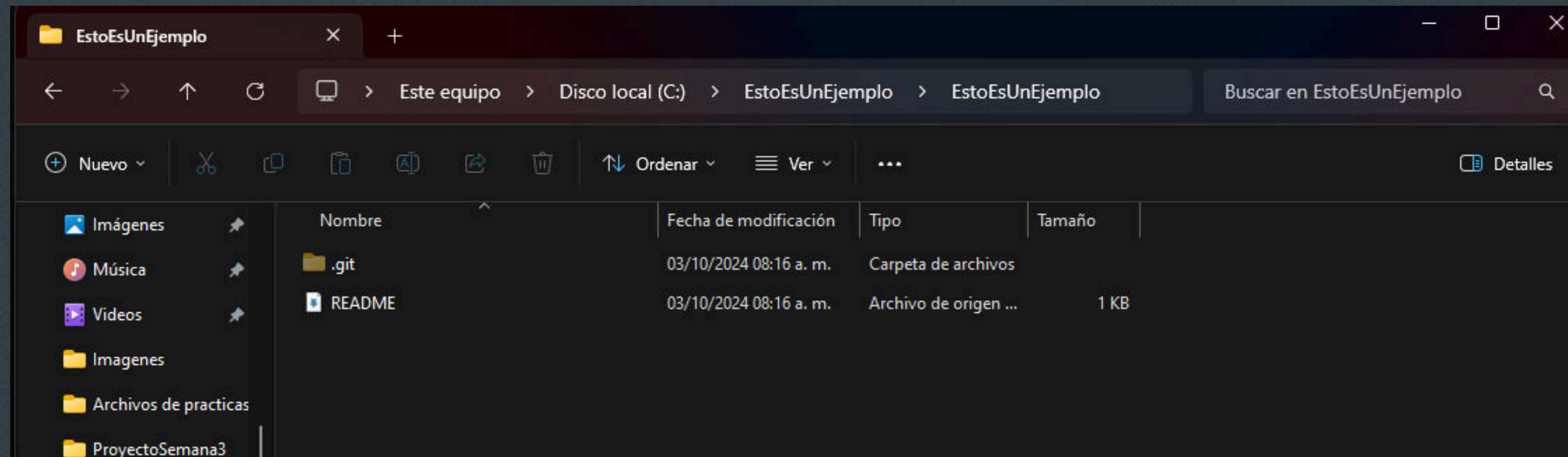
- Abre el repositorio que acabas de crear en GitHub.
- Haz clic en el botón verde Code y copia la URL que aparece.
- En la consola de tu computadora, navega hasta la carpeta donde quieres guardar tu proyecto usando el comando cd: `"cd ruta/de/tu/carpeta"`
- Clona el repositorio con el siguiente comando: `"git clone<URL-del-repositorio>"`



Paso 5: Clonar el Repositorio a tu Computadora



Paso 5: Clonar el Repositorio a tu Computadora



Paso 6: Instalar GitFlow



- Abre tu consola y ejecuta el siguiente comando para instalar GitFlow:

Windows: Ya puede venir preinstalado con Git.

macOS/Linux: Ejecuta `brew install git-flow` si tienes Homebrew instalado.

- Para verificar la instalación, escribe en la consola:

```
Símbolo del sistema
C:\Users\Angel>git flow version
1.12.3 (AVH Edition)
C:\Users\Angel>
```


Paso 7: Inicializar GitFlow en el Repositorio

- Navega a la carpeta de tu proyecto clonado: "cd<nombre-del-repositorio>"

```
Windows PowerShell
PS C:\EstoEsUnEjemplo> cd EstoEsUnEjemplo
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```

- Inicializa GitFlow en el repositorio con: "git flow init"

```
Windows PowerShell
PS C:\EstoEsUnEjemplo> cd EstoEsUnEjemplo
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> 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:/EstoEsUnEjemplo/EstoEsUnEjemplo/.git/hooks]
```


Paso 8: Crear una Rama

- Supón que quieres agregar una nueva funcionalidad al proyecto.
- Crea una nueva rama llamada feature con el siguiente comando:
"gitflowfeature start <nombre-de-la-feature>"
- Esto crea una rama separada donde puedes trabajar en la nueva funcionalidad sin afectar las demás.

```
Windows PowerShell
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git flow feature start RamaFeatureEjemplo
Switched to a new branch 'feature/RamaFeatureEjemplo'

Summary of actions:
- A new branch 'feature/RamaFeatureEjemplo' was created, based on 'develop'
- You are now on branch 'feature/RamaFeatureEjemplo'

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

    git flow feature finish RamaFeatureEjemplo

PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git flow feature finish RamaFeatureEjemplo
Switched to branch 'develop'
Already up to date.
Deleted branch feature/RamaFeatureEjemplo (was def5561).

Summary of actions:
- The feature branch 'feature/RamaFeatureEjemplo' was merged into 'develop'
- Feature branch 'feature/RamaFeatureEjemplo' has been locally deleted
- You are now on branch 'develop'
```

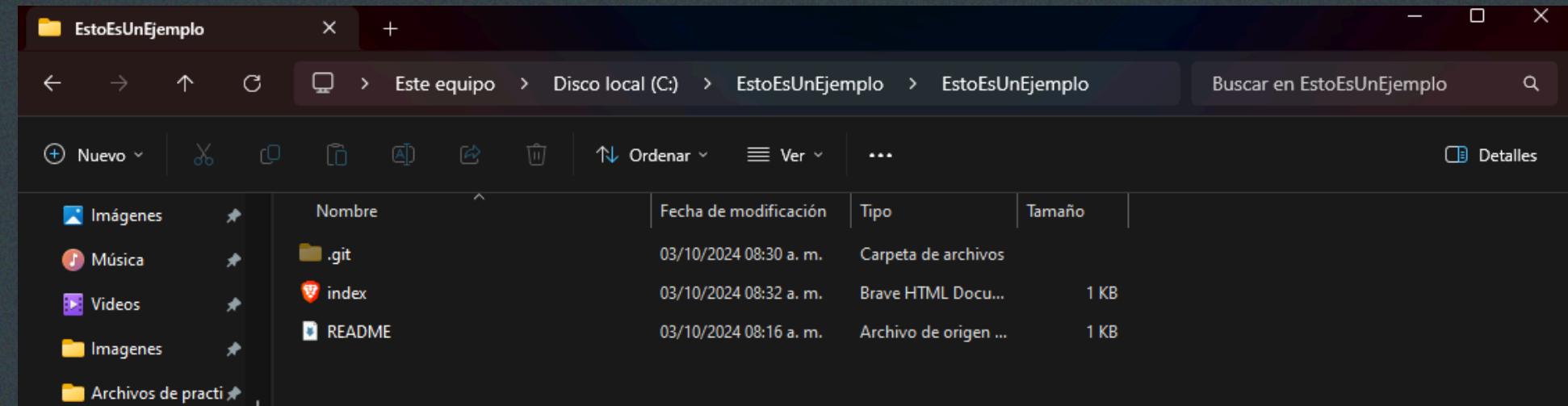
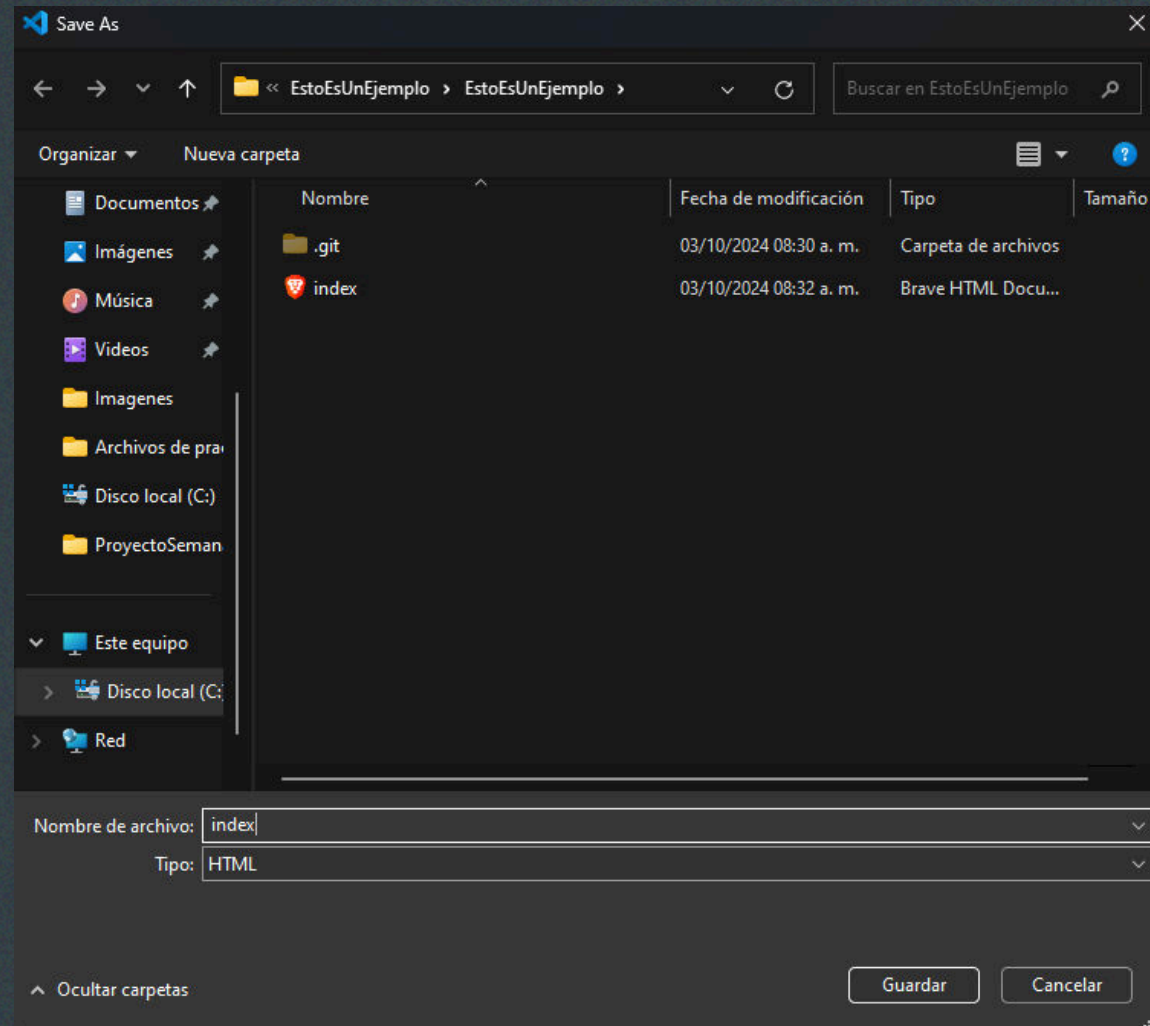

Paso 9: Trabajar en la Rama Feature



- Abre tu editor de código (como Visual Studio Code).
- Haz cambios en los archivos del proyecto, por ejemplo, crea un nuevo archivo .html o modifica uno existente.
- Guarda los cambios y revisa que todo funcione correctamente.



Paso 9: Trabajar en la Rama Feature □ ×



Paso 10: Revisar el estado de los

Archivos

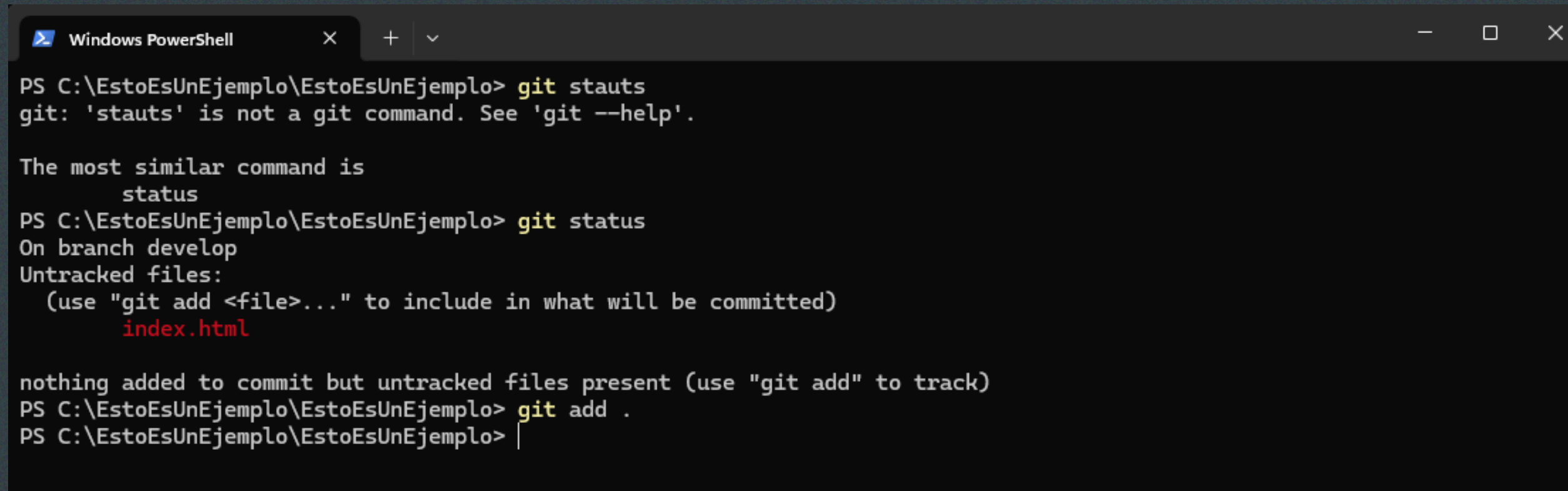
- En la consola, revisa qué archivos han sido modificados con el comando: "git status"
- Veras una lista de archivos que has cambiado, en color rojo si no han sido añadidos al repositorio

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```


Paso 11: Añadir los archivos modificados

- Para añadir los cambios al repositorio, usa el comando: "git add ."



```
Windows PowerShell
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git stauts
git: 'stauts' is not a git command. See 'git --help'.

The most similar command is
    status
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git add .
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```

- Esto agrega todos los archivos modificados.

Paso 12: Crear un Commit



- Un commit es como tomar una "foto" de los cambios que has hecho.
- Haz un commit con un mensaje descriptivo usando el comando:
"git commit -m "Descripción de los cambios realizados""

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git add .
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git commit -m "Se hicieron cambios en el archivo index.html"
[develop ddbb92f] Se hicieron cambios en el archivo index.html
1 file changed, 12 insertions(+)
create mode 100644 index.html
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> D|
```


Paso 13: Subir cambios a GitHub

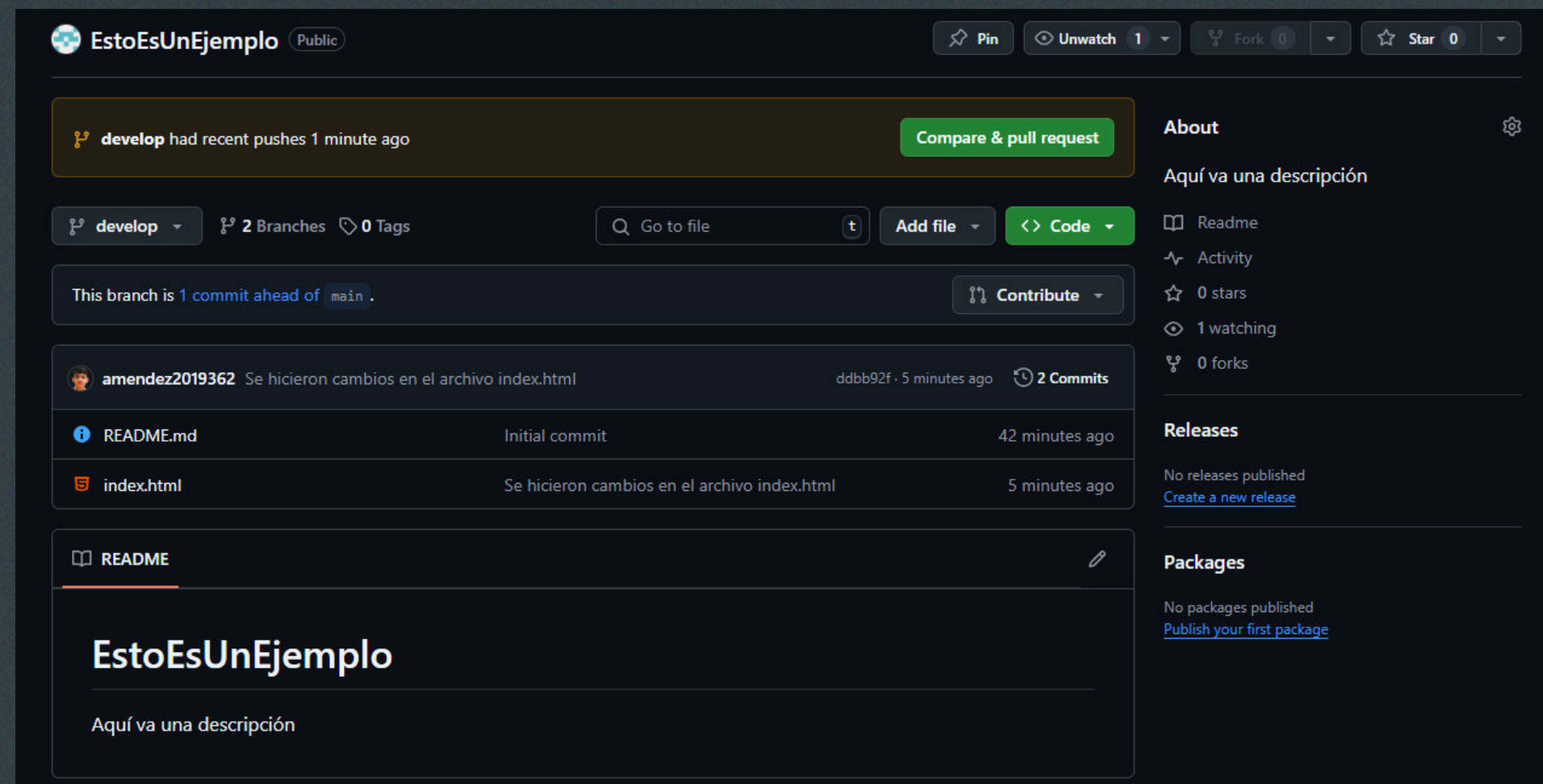


- Para que los cambios se reflejen en GitHub, usa el siguiente comando para subirlos: "git push origin develop"

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git push origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 484 bytes | 484.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Angel-Adolfo/EstoEsUnEjemplo/pull/new/develop
remote:
To https://github.com/Angel-Adolfo/EstoEsUnEjemplo.git
 * [new branch]      develop -> develop
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```


Paso 14: Revisar cambios en Github □ ×

- Entra a tu repositorio en GitHub y verifica que los cambios se hayan subido correctamente.



Paso 15: Crear una rama Hotfix para □ ×

corregir Errores

- Si surge un error en el código que ya está en producción, crea una rama:
`"git flow hotfix start <nombre-del-hotfix>"`

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git flow hotfix start SolucionIndex  
Switched to a new branch 'hotfix/SolucionIndex'
```

Summary of actions:

- A new branch 'hotfix/SolucionIndex' was created, based on 'main'
- You are now on branch 'hotfix/SolucionIndex'

Follow-up actions:

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

```
git flow hotfix finish 'SolucionIndex'
```

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```

- Esta rama te permite corregir rápidamente los errores sin afectar las demás funcionalidades.

Paso 16: Corregir y Finalizar el Hotfix — □ ×

- Realiza las correcciones necesarias y sigue el proceso de add y commit.



```
index.html > html
You, 1 second ago | 1 author (You)
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>Esto es un ejemplo</title>
7 </head>
8 <body>
9   <h1>Hola mundo</h1>
10  Esto es una corrección de la rama developer
11 </body>
12 </html> You, 14 minutes ago • Se hicieron cambios en el archivo index.html
```


Paso 16: Corregir y Finalizar el Hotfix

```
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git status
On branch hotfix/SolucionIndex
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:\EstoEsUnEjemplo\EstoEsUnEjemplo> git add .
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> git commit -m "Se arreglaron cambio en el index"
[hotfix/SolucionIndex c5c6301] Se arreglaron cambio en el index
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\EstoEsUnEjemplo\EstoEsUnEjemplo> |
```

- Finaliza la rama: `git flow hotfix finish <nombre-del-hotfix>`

Gracias!

