

## Reporte Técnico

**Asunto:** Sistema de Información Web **Postgrado:** Maestría en Ingeniería en Electrónica

**Tutores:** Dr. Julio Mello / MSc. Marcos Benítez

**Autor:** Aldo Reyes Mella

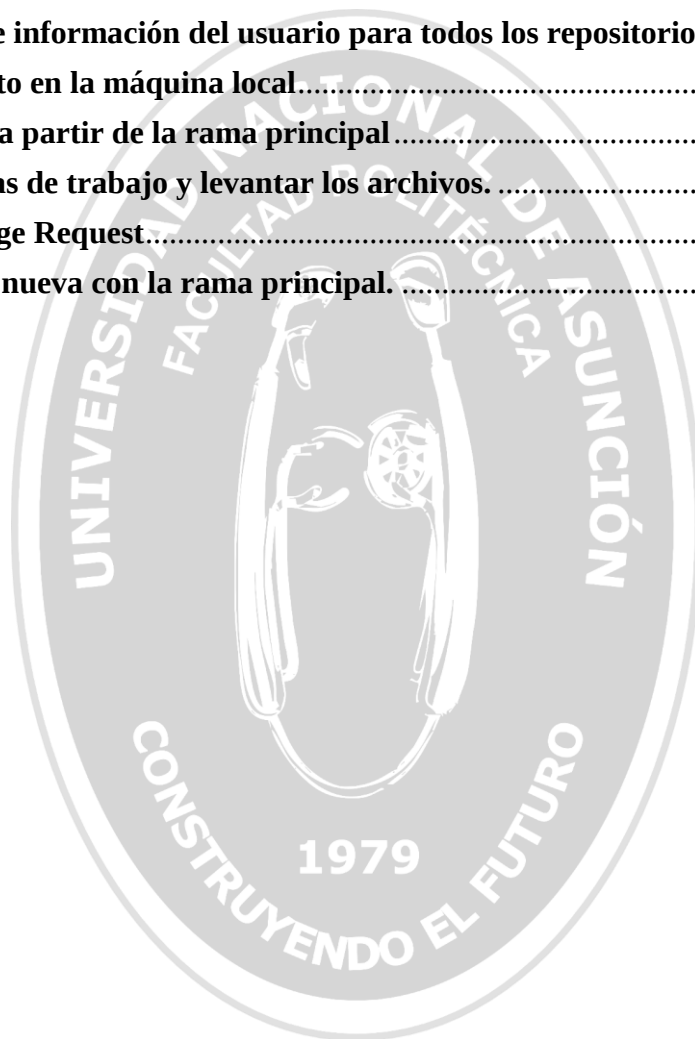
**Fecha:** 23 de marzo del 2025

**Clasificación del documento:** Público



## TABLA DE CONTENIDO

Reporte Técnico .....	1
<b>1. Instalación del cliente git en el ordenador. ....</b>	<b>3</b>
<b>2. Configuración de información del usuario para todos los repositorios locales .....</b>	<b>4</b>
<b>3. Clonar el proyecto en la máquina local.....</b>	<b>7</b>
<b>4. Crear una rama a partir de la rama principal .....</b>	<b>7</b>
<b>5. Crear las carpetas de trabajo y levantar los archivos. ....</b>	<b>8</b>
<b>6. Solicitud de Merge Request.....</b>	<b>9</b>
<b>7. Mezclar la rama nueva con la rama principal. ....</b>	<b>10</b>






## 1. Instalación del cliente git en el ordenador.

Se ingresa a <https://git-scm.com/downloads> y se descarga acorde al sistema operativo del ordenador, para nuestro caso se descargó el de 64bit, luego se debe ejecutar, leer la licencia y darle next para continuar con el proceso de instalación.

Se visualizará la barra de descarga y cuando esté completo se desplegará el mensaje "Completing the Git Setup Wizard con el botón de finalizar, se desplegará la consola.

### Downloads

 macOS
  Windows
  Linux/Unix

Latest source Release

## 2.49.0

Release Notes (2025-03-14)

Download for Windows

Older releases are available and the Git source repository is on GitHub.

### Download for Windows

Click here to download the latest (2.49.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 7 days ago, on 2025-03-17.

**Other Git for Windows downloads**

Standalone Installer

32-bit Git for Windows Setup.


64-bit Git for Windows Setup.

Portable ("thumbdrive edition")

32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

▼ hoy

 Git-2.49.0-64-bit

Git 2.49.0 Setup

**Information**

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

### GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

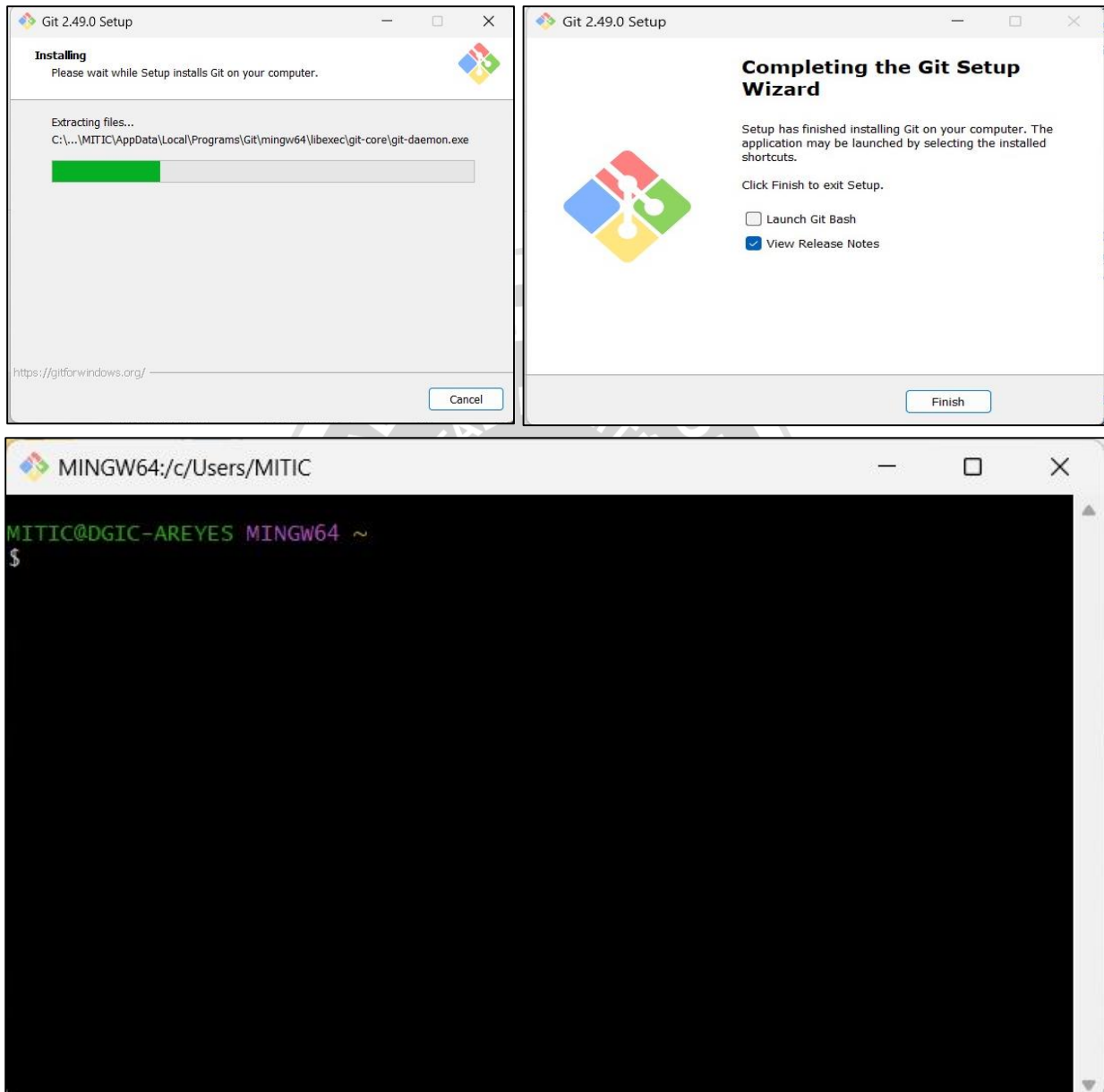
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>

Next Cancel



- Crear una rama a partir de la rama principal
- Crear la carpeta Semana 01 y levantar todos los ejercicios de la tarea 1 y tarea 2
- Crear la carpeta Semana 02 y levantar todos los ejercicios de la diapositiva de material de javascript
- Levantar la rama
- Solicitar Merge Request
- Mezclar la rama con la rama principal

## 2. Configuración de información del usuario para todos los repositorios locales

Se configura las credenciales básicas para nuestro entorno Git de trabajo, a través de los siguientes comandos:

```
git config --global user.name "ajreyes"
git config --global user.email "ajreymel@gmail.com"
```

De esta forma se obtiene lo básico para utilizarlo.

- ✓ Crear una cuenta en GitHub (<https://github.com/>): Para crear la cuenta en la plataforma Github, accedemos la página <https://github.com/> y nos registramos. Una vez registrado procederemos a autenticarnos para iniciar la carga de nuestro proyecto.
- ✓ Crear un proyecto denominado Sistemas de Información Web

	<p>Iniciamos sesión en Github desde la página <a href="https://github.com/login">github.com/login</a></p> <p>Se coloca email y contraseña con la que se configuró la cuenta Github.</p>
	<p>En la esquina superior derecha se encuentra un signo más en el cual se despliegan varias opciones, entre ellas “New repository” para crear un entorno de trabajo Git.</p>

### 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 \*  / Repository name \*

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

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:

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

Add .gitignore

.gitignore template:

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Nombre del Proyecto: “Sistemas de información Web”

En descripción “Public” para acceso de usuario externo a nuestro repositorio.

Se selecciona para agregar un archivo README.

### 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 \*  / Repository name \*

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

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:

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

Add .gitignore

.gitignore template:

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License:

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

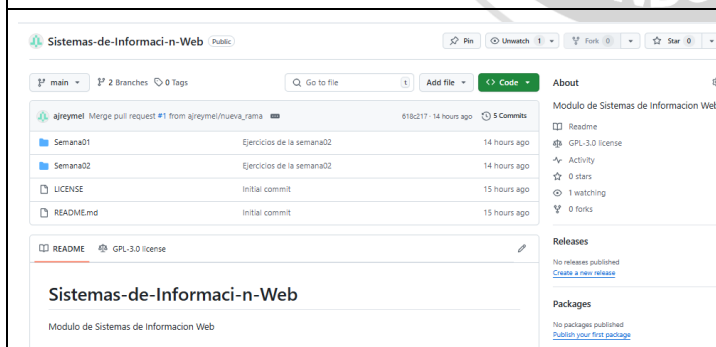
This will set `main` as the default branch. Change the default name in your [settings](#).

☐ You are creating a public repository in your personal account.

Podemos agregar una descripción del repositorio y seleccionar la opción "Public" para permitir el acceso a usuarios externos.

Además, marcamos la opción para incluir un archivo README, el cual sirve para documentar y explicar el propósito, estructura y detalles relevantes del proyecto a otros colaboradores o interesados.

Finalmente, seleccionamos "Create repository" para completar el proceso.



The screenshot shows a GitHub repository page for 'Sistemas-de-Informaci-n-Web' (Public). The repository is owned by 'ajreymel' and has 2 branches and 0 tags. The README file is visible, showing the title 'Sistemas-de-Informaci-n-Web' and the description 'Modulo de Sistemas de Informacion Web'. The repository is initialized with a README file and a .gitignore file. The license is set to 'None'.

El repositorio se encuentra listo para ser cargado, en la imagen se visualiza el que ya tengo con los archivos pero al inicio se encuentra vacío.

### 3. Clonar el proyecto en la máquina local

En el ordenador, nos posicionamos en la carpeta de trabajo que utilizaremos y ejecutamos en la terminal de comandos

```
git clone https://github.com/ajreymel/Sistemas-de-Informaci-n-Web.git
```

La ejecución da el siguiente resultado en la consola:



```
MINGW64/c/Users/Usuario/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO
$ git clone https://github.com/ajreymel/Sistemas-de-Informaci-n-Web.git
Cloning into 'Sistemas-de-Informaci-n-Web'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), 12.75 KiB | 3.19 MiB/s, done.
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO
$ cd Sistemas-de-Informaci-n-Web
bash: cd: Sistemas-de-Informaci-n-Web: No such file or directory
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO
$ cd Sistemas-de-Informaci-n-Web
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
```

Además se accede a la carpeta “Sistemas de Información Web” a través del comando cd y corroboramos el estatus.

### 4. Crear una rama a partir de la rama principal

El proyecto descargado se encuentra actualmente en la rama principal (main/master). Utilizando Git, podemos implementar un flujo de trabajo eficiente creando una rama secundaria (feature branch) para realizar modificaciones, lo que nos permite:

Aislar los cambios en desarrollo

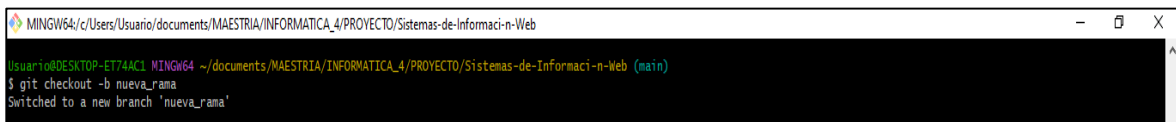
Proteger la rama principal de modificaciones directas

Mantener un entorno estable mientras trabajamos en nuevas características

Este enfoque es particularmente útil incluso cuando la rama principal no contiene aún un proyecto completo, ya que establece buenas prácticas desde el inicio del desarrollo.

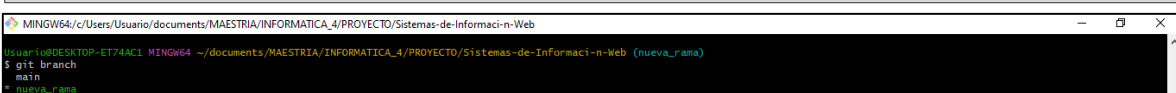
Creamos una rama a partir de la rama principal, el comando es el siguiente:

```
git checkout -b nueva-rama
```



```
MINGW64/c/Users/Usuario/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web (main)
$ git checkout -b nueva-rama
Switched to a new branch 'nueva-rama'
```

```
git branch
```

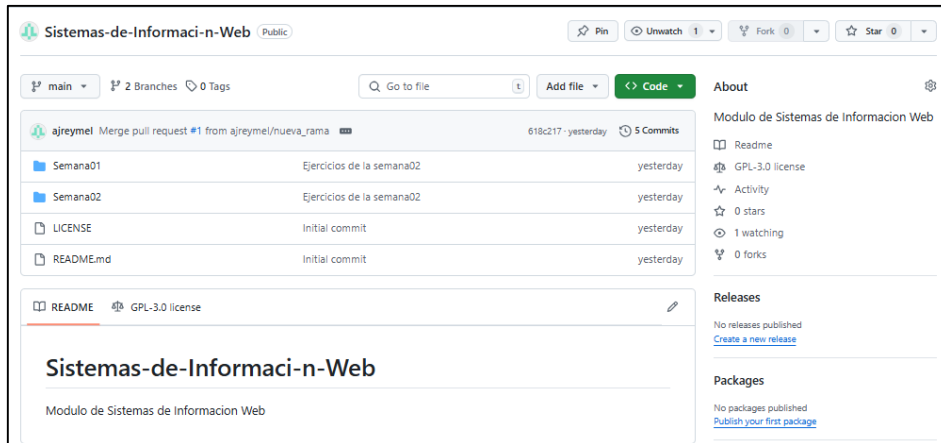


```
MINGW64/c/Users/Usuario/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web
Usuario@DESKTOP-ET74AC1 MINGW64 ~/documents/MAESTRIA/INFORMATICA_4/PROYECTO/Sistemas-de-Informaci-n-Web (nueva-rama)
$ git branch
main
* nueva-rama
```

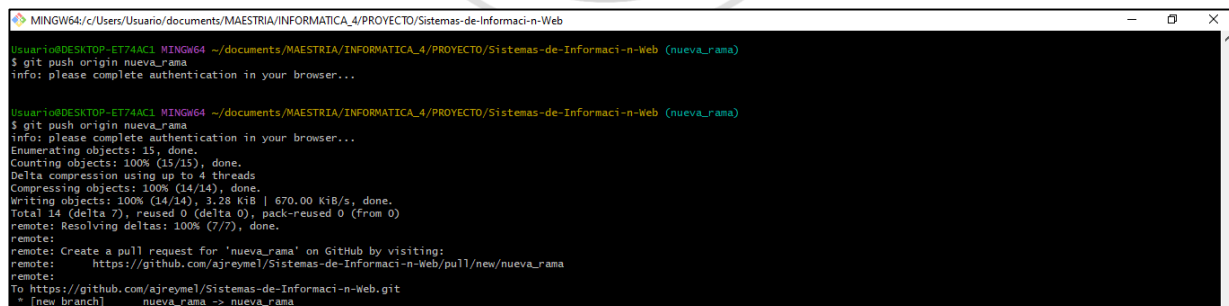
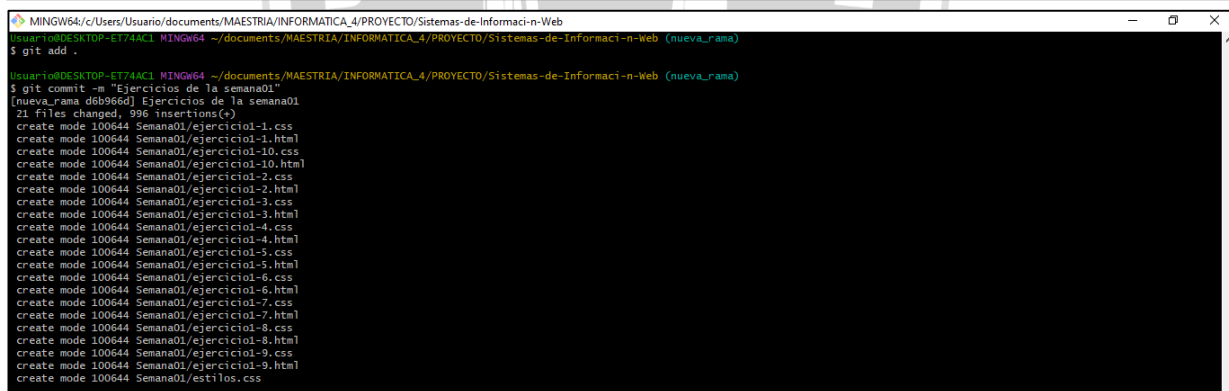
## 5. Crear las carpetas de trabajo y levantar los archivos.

Dentro del directorio local del proyecto clonado desde Git, creamos las carpetas Semana01 y Semana02 en el escritorio.

Sincronizaremos estos directorios con el repositorio remoto en GitHub mediante los comandos estándar de Git (add, commit, push):



```
git add .
git commit -m "Ejercicios de Semana 01 y Semana 02"
git push origin nueva-rama
```

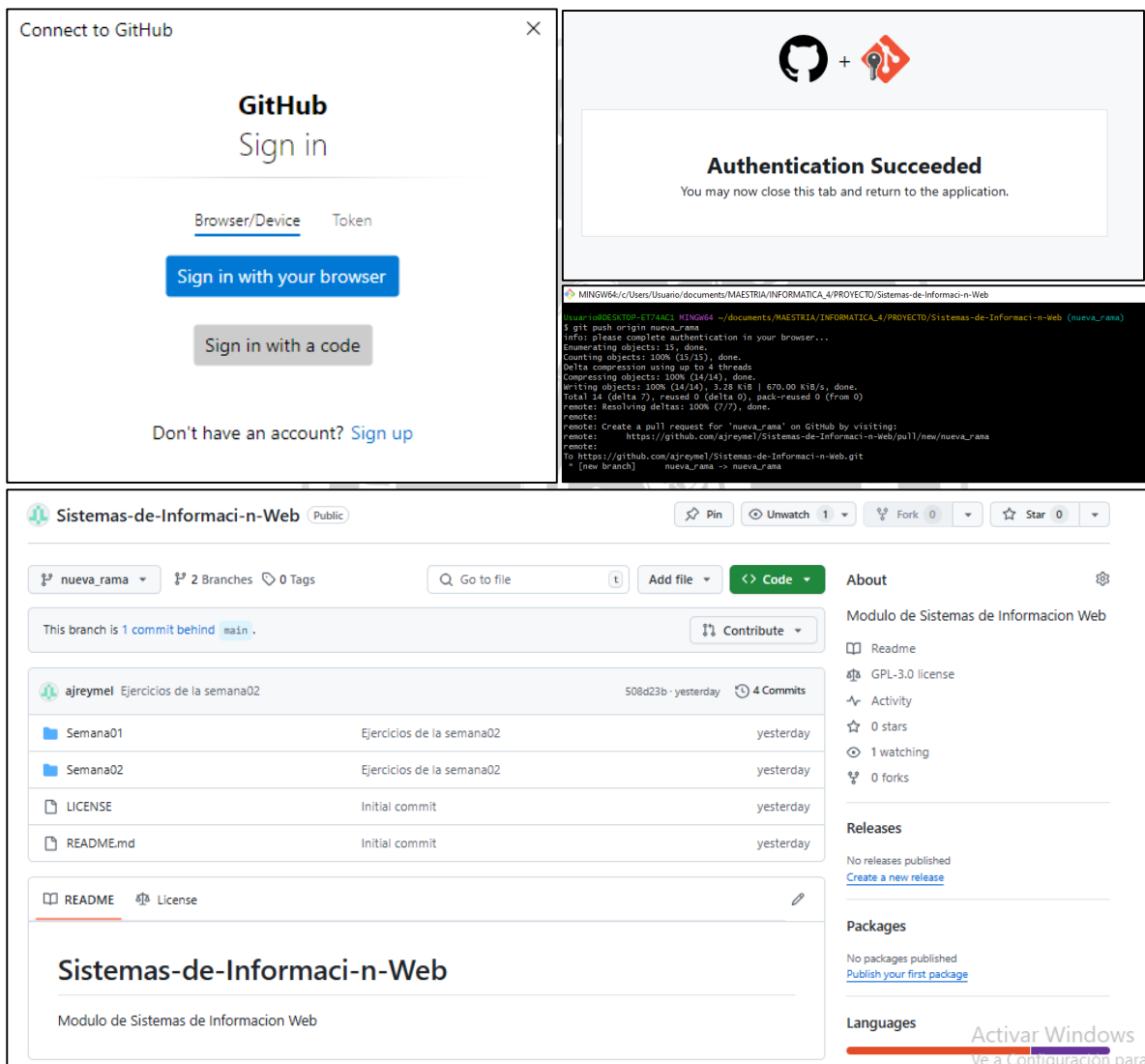




Al ser la primera vez que accedemos, el sistema requerirá autenticación. Para ello:  
Seleccionamos la opción "Sign in with your browser".

Seguimos los pasos de autenticación que indica la plataforma.

Una vez completado el proceso, el sistema cargará automáticamente nuestro proyecto en GitHub.  
El resultado final se visualizará como se muestra en la siguiente imagen:



## 6. Solicitud de Merge Request

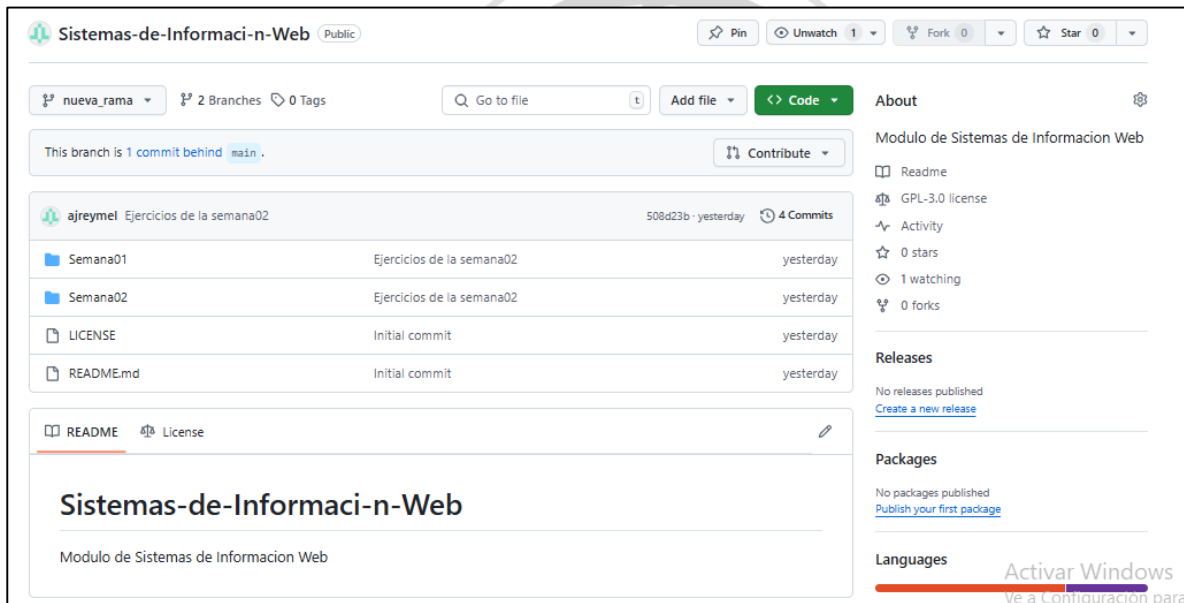
- ✓ Accede al repositorio deseado en GitHub.
- ✓ Navega a la pestaña "Pull Requests".
- ✓ Selecciona el botón "New Pull Request".
- ✓ Especifica las ramas:
  - Rama de origen (compare): nueva-rama
  - Rama de destino (base): main

- ✓ Proporciona una descripción detallada de los cambios realizados y confirma con "Create Pull Request".

Esta acción solo genera una solicitud de revisión del merge. La fusión no ocurre automáticamente. Si no existen conflictos, puede ejecutar "Merge Pull Request" para integrar los cambios.

## 7. Mezclar la rama nueva con la rama principal.

Se verifica que la rama main con los ejercicios de las semanas 01 y 02 se fusionaron.



El repositorio se encuentra en: <https://github.com/ajreymel/Sistemas-de-Informaci-n-Web>