

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

- ¿Qué es GitHub?

Es una comunidad donde podemos compartir nuestros repositorios de forma pública o privada. Con una cuenta en GitHub disponemos de un perfil similar a una red social donde podemos cargar nuestros repositorios, seguir a otros usuarios, que nos sigan a nosotros, ver los repositorios de otras personas, poder clonarlos, guardarlos como favoritos, etc.

- ¿Cómo crear un repositorio en GitHub?

Una vez en nuestro perfil ingresamos a crear un nuevo repositorio, colocamos su nombre (convenientemente el mismo nombre que el repositorio local), añadimos una breve descripción y ponemos si queremos que sea público o privado.

- ¿Cómo crear una rama en Git?

Una rama es una línea del tiempo donde vamos a poder plasmar con puntos, los cambios significativos de nuestro proyecto. Para crear una nueva rama tenemos que ejecutar el comando *git branch nombre_rama_nueva*.

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama tenemos que ejecutar el comando *git checkout nombre_rama*.

- ¿Cómo fusionar ramas en Git?

Para fusionar dos ramas ejecutamos el comando *git merge nombre_rama_a_fusionar*. Cabe aclarar que esto va a volcar lo que esta en la rama a fusionar a la rama en la que nos encontramos en ese momento.

- ¿Cómo crear un commit en Git?

Un commit es la manera que Git tiene de registrar cambios en un proyecto. Para crear un commit debemos ejecutar el comando *git commit m- mensaje_descriptivo*.

- ¿Cómo enviar un commit a GitHub?

Para llevar un commit del repositorio local a GitHub necesitamos tener una cuenta en la página oficial de GitHub. Luego de crear nuestro repositorio remoto en GitHub y haberle llevado el código desde nuestra maquina local agregando la dirección de nuestro repositorio remoto haciendo un “push”, hacemos un commit como se explicó anteriormente.

- ¿Qué es un repositorio remoto?

Es una copia de un proyecto subido a internet, que pueden ser alojados en diferentes sitios. Ademas de servir para almacenar nuestros proyectos y permitirnos trabajarlos desde diferentes lugares, permite compartir y colaborar con otros programadores.

- ¿Cómo agregar un repositorio remoto a Git?

Para ello debemos usar el comando *git remote add* en la terminal. Este comando se debe ejecutar dentro del directorio donde está el repositorio local.

- ¿Cómo empujar cambios a un repositorio remoto?

Para poder realizar esto, debemos haber previamente guardado los cambios realizados al área de staging (*git add*), luego realizar un commit como se explicó anteriormente, y finalmente subir los cambios al repositorio remoto con el comando *git push origin nombre_repositorio*.

- ¿Cómo tirar de cambios de un repositorio remoto?

Para traer los cambios de un repositorio remoto a nuestro repositorio local, debemos usar el comando *git pull origin rama_a_traer*.

- ¿Qué es un fork de repositorio?

Esto es una copia exacta de un repositorio que se crea en nuestra cuenta de GitHub. Es totalmente independiente, por lo tanto los cambios que haga van a estar asociados únicamente a nuestro repositorio.

- ¿Cómo crear un fork de un repositorio?

Debemos iniciar sesión con nuestra cuenta de GitHub y buscamos el repositorio que queremos copiar, una vez dentro del repositorio, hacemos click en Fork, cambiamos si queremos la descripción del repositorio y aguardamos a que se cree una copia del original en nuestra cuenta.

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Para enviar una solicitud de extracción debemos ir a la página principal del repositorio en cuestión, luego de realizar los cambios y crear una nueva rama y hacer un commit, hacemos un push del repositorio local al remoto. Luego de hacer esto, en el repositorio remoto nos aparecerá una opción para comparar y realizar la solicitud de extracción, proporcionando los detalles de los cambios realizados.

- ¿Cómo aceptar una solicitud de extracción?

En el repositorio remoto nos aparecerá un apartado de “Solicitudes de extraccion”, ahí podremos la/s solicitud/es y revisarlas. Luego de ello podemos aprobar y fusión los cambios.

- ¿Qué es una etiqueta en Git?

Los tags o etiquetas son puntos específicos en la historia de nuestro proyecto y se usan para marcar la versión de un commit. Hay ligeras y anotadas, la diferencia general es que unas contienen mas información que las otras.

- ¿Cómo crear una etiqueta en Git?

Para crear una tag, en este caso anotada, debemos ejecutar el comando *git tag -a nombre_etiqueta -m 'mensaje'*.

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub debemos realizar un push con el comando *git push origin nombre_etiqueta* para una etiqueta especifica o *git push --tags* para incluir todas las etiquetas.

- ¿Qué es un historial de Git?

Nos permite ver de forma general las versiones que se van generando dentro del proyecto. Podemos ver quien fue el autor de los cambios, la fecha y los comentarios.

- ¿Cómo ver el historial de Git?

Para verlo utilizamos el comando *git log*, esto nos muestra una lista de todas las confirmaciones de la rama en la que estamos.

- ¿Cómo buscar en el historial de Git?

Otras formas de buscar en el historial es con el comando `git log -p`, el cual me indica los detalles de los cambios en cada commit. Tambien puedo filtrar la cantidad que quiero ver especificando `git log -p -numero_commit`. Otra opción es `git log --stat` y nos muestra los cambios resumidos. Otros más precisos son `git log --online` mostrando toda la información en una línea, `git log --format: "caracter"`, puedo filtrar por quien lo hizo, fecha, el mensaje que se agregó, etc.

- ¿Cómo borrar el historial de Git?

Podemos hacerlo de dos maneras.

La primera consiste en reemplazar todo el historial con un nuevo commit, manteniendo el estado actual del proyecto, pero perdiendo el historial para ti y otros colaboradores. Para ello creamos un nuevo branch sin historial `git checkout -orphan nueva_rama`, `git add`, `git commit -m "Nuevo inicio del historial"`, luego eliminamos la rama anterior y renombramos la nueva `git branch -D main`, `git branch -m main`, finalmente forzamos el push para sobreescribir el historial remoto, `git push origin main --force`.

La segunda opción consiste en borrar solo el historial local, manteniendo el remoto. Primero eliminamos el directorio `.git` que contiene el historial y la configuración del repositorio, `rm -rf .git`. Luego inicializamos un nuevo repositorio en la misma carpeta, con el historial desde cero, `git init`. Ahora vinculamos el repositorio local con el remoto `git remote add origin URL_del_repo` (en GitHub). Luego descargamos los cambios desde el remoto pero no los integramos, actualizando las referencias remotas (`origin/main`), `git fetch`. Finalmente creamos una rama a partir del remoto y cambiamos a ella `git checkout -t origin/main`.

- ¿Qué es un repositorio privado en GitHub?

Es un espacio para guardar código, archivos y revisiones de forma segura, solo visible para los usuarios con permiso.

- ¿Cómo crear un repositorio privado en GitHub?

En GitHub vamos a crear un Nuevo repositorio, damos un nombre, una descripción y en visibilidad elegimos que sea privado. Luego de crearlo, vamos a configurar las opciones. En colaboradores podemos seleccionar a quien o quienes les queremos compartir el proyecto.

- ¿Qué es un repositorio público en GitHub?

Es similar al repositorio privado, un espacio virtual para almacenar y compartir código, archivos y revisiones de forma gratuita y accesible para cualquier persona. En este caso sirven para compartir código abierto, software que se puede usar, modificar y distribuir libremente.

- ¿Cómo crear un repositorio público en GitHub?

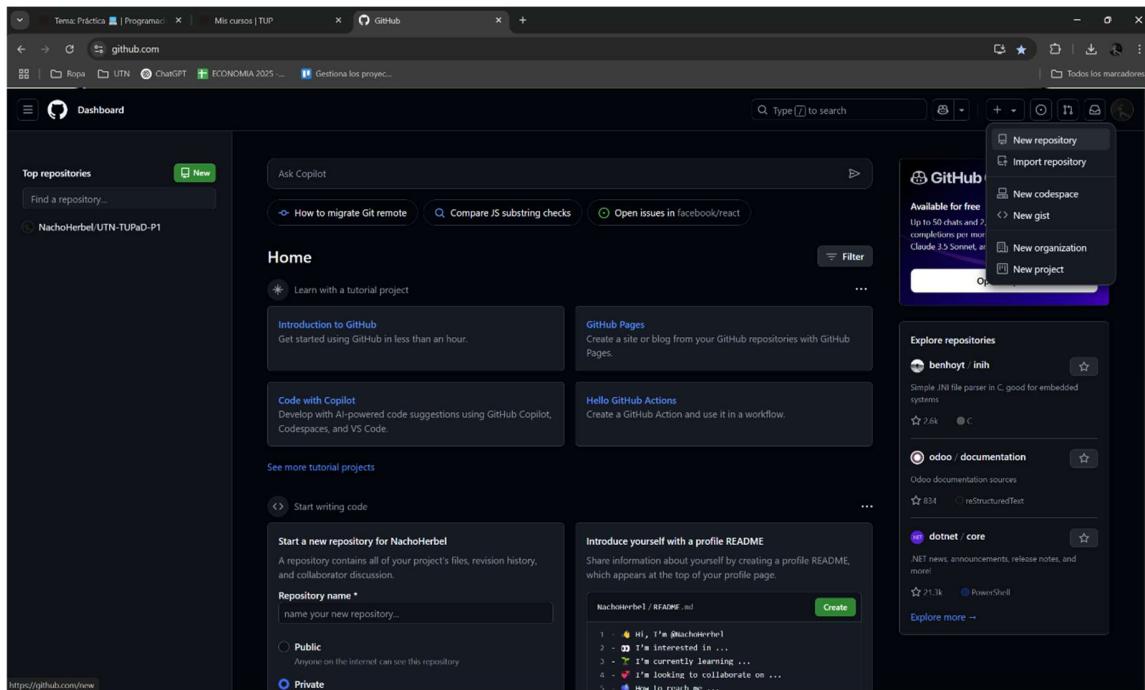
Del mismo modo que creamos un repositorio privado, pero en este caso en las opciones elegimos que sea público.

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público, puedes invitar colaboradores o simplemente configurarlo para que el mismo sea público, así cualquiera en internet y GitHub puede verlo.

2) Realizar la siguiente actividad:

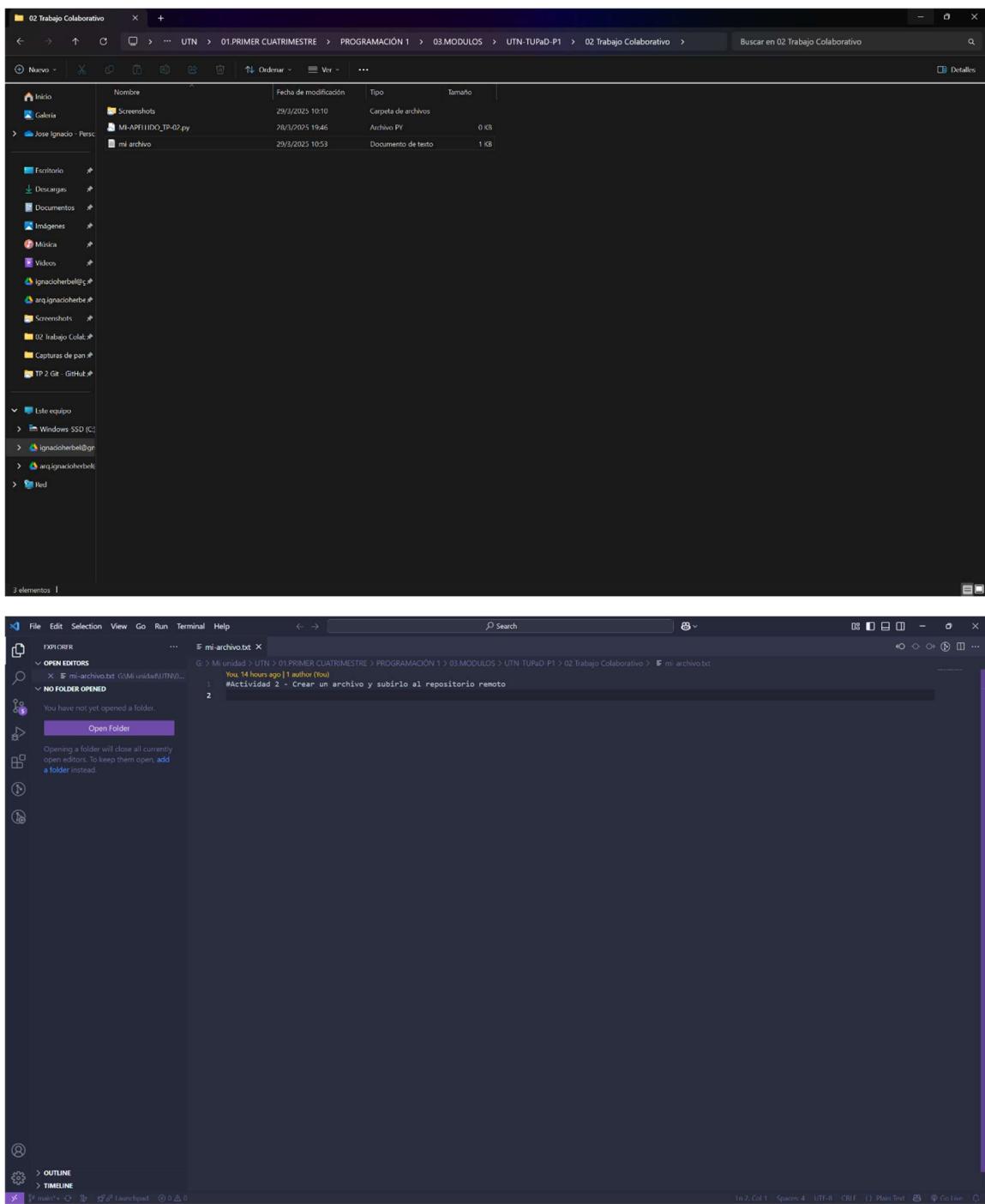
- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Escribe el repositorio sea público.
 - Inicializa el repositorio con un archivo.
- Agregando un Archivo
 - Crea un archivo simple, por ejemplo, "mi-archivo.txt".
 - Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
 - Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).
- Creando Branchs
 - Crear una Branch
 - Realizar cambios o agregar un archivo
 - Subir la Branch



Creando un repositorio.

The image shows two screenshots of a GitHub interface. The top screenshot is a 'Create a new repository' form. It has fields for 'Owner' (set to 'NachoHerbel'), 'Repository name' (set to 'Actividad 2'), and 'Description' (left empty). It includes options for 'Public' or 'Private' visibility, 'Add a README file' (checked), and 'Add .gitignore' (with a dropdown for 'None'). It also has sections for choosing a license ('None') and setting the default branch ('main'). The bottom screenshot shows the resulting repository page for 'Actividad 2'. It displays one commit by 'NachoHerbel' titled 'Initial commit' with a timestamp of '7e083c3 · now'. The repository has 1 branch and 0 tags. On the right side, there are sections for 'About' (with a note 'No description, website, or topics provided.'), 'Activity' (0 stars, 1 watching, 0 forks), 'Releases' (none published), and 'Packages' (none published).

Dándole nombre, eligiendo que sea público e inicializando el repositorio con un archivo.



Creando archivo *mi-archivo.txt*, y abriéndolo con VSC.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — □ ×
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git add .

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git commit -m "Agregando mi-archivo.txt"
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.

fatal: unable to auto-detect email address (got 'ignac@Notebook-Nacho.(none)')

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ |
```

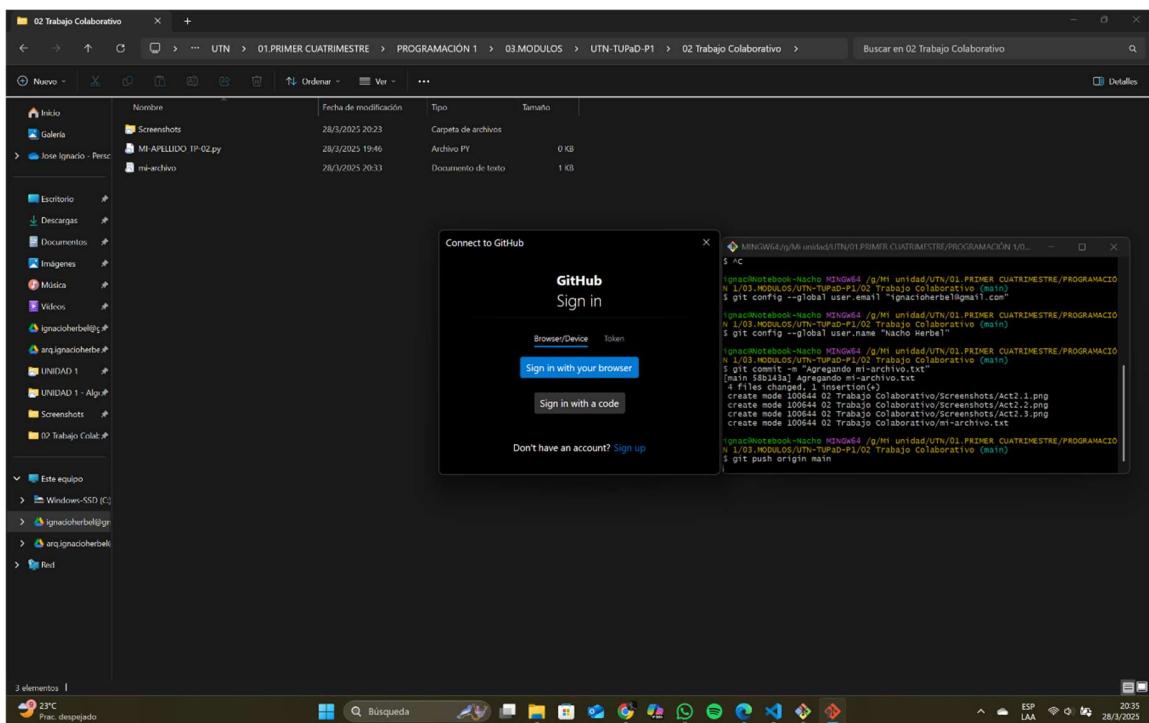
```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — □ ×
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ ^C

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git config --global user.email "ignacioherbel@gmail.com"

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git config --global user.name "Nacho Herbel"

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git commit -m "Agregando mi-archivo.txt"
[main 58b143a] Agregando mi-archivo.txt
 4 files changed, 1 insertion(+)
 create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.1.png
 create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.2.png
 create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.3.png
 create mode 100644 02 Trabajo Colaborativo/mi-archivo.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ |
```



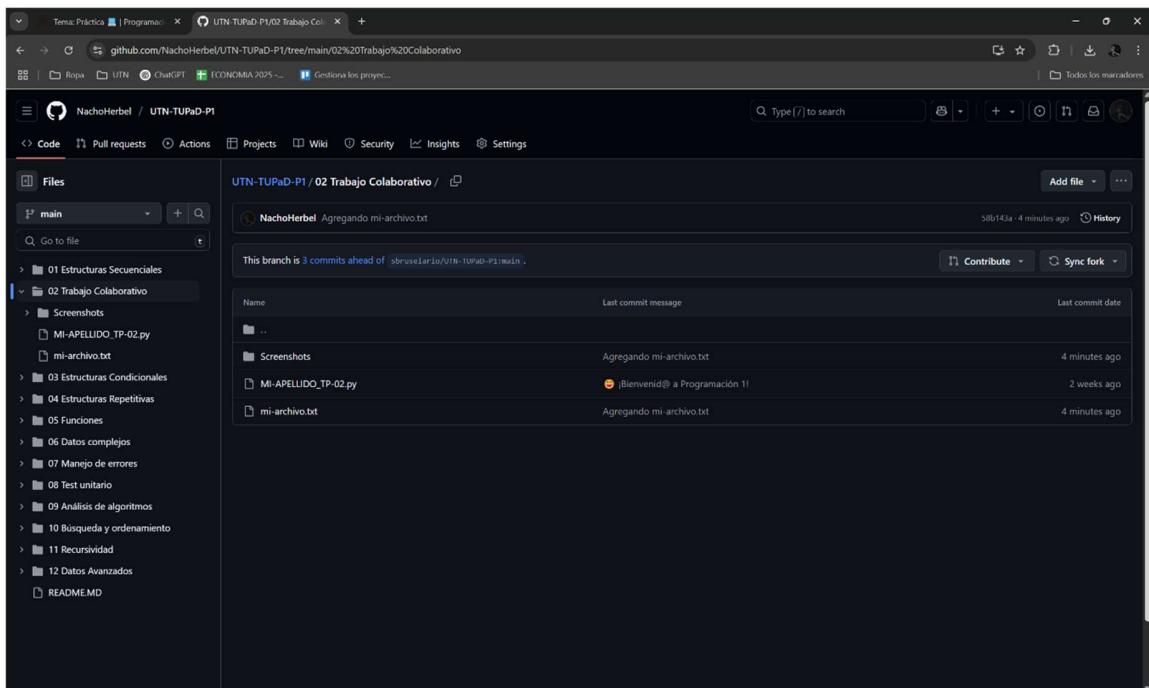
Moviendo los cambios al staging area, y agregando un commit. Previo inicio de sesión en GitHub.

```

MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
[main 58b143a] Agregando mi-archivo.txt
 4 files changed, 1 insertion(+)
    create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.1.png
    create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.2.png
    create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.3.png
    create mode 100644 02 Trabajo Colaborativo/mi-archivo.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ git push origin main
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ 

```



```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... ━ ━ X
create mode 100644 02 Trabajo Colaborativo/mi-archivo.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 24 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 690.20 KiB | 19.72 MiB/s, done.
Total 8 (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/NachoHerbel/UTN-TUPaD-P1
  6b6664f..58b143a  main -> main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git push origin main
Everything up-to-date

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ |
```

Subiendo cambios al repositorio remoto de GitHub.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — X
info: please complete authentication in your browser...
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 24 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 690.20 KiB | 19.72 MiB/s, done.
Total 8 (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/NachoHerbel/UTN-TUPaD-P1
  6b6664f..58b143a main -> main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git push origin main
Everything up-to-date

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch
* main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$
```

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — X
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git push origin main
Everything up-to-date

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch
* main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch nuevaBranch

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch
* main
nuevaBranch

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$
```

Creando una nueva Branch.

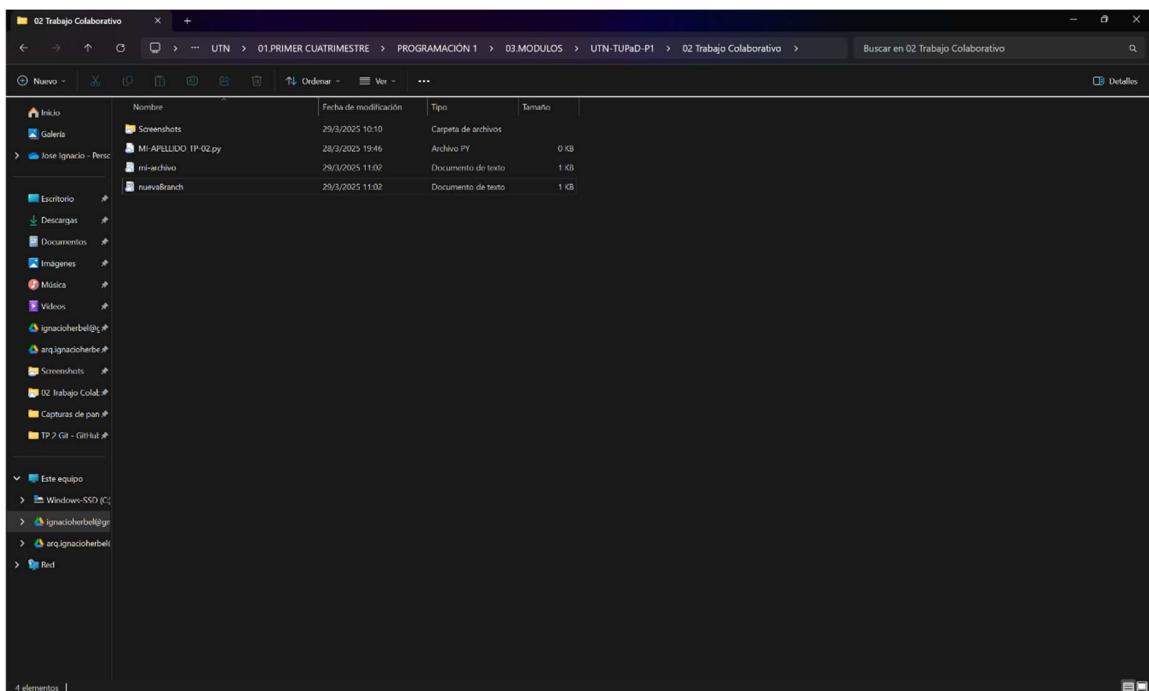
```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — □ X
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch
* main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch nuevaBranch

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git branch
* main
nuevaBranch

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git checkout nuevaBranch
M      02 Trabajo Colaborativo/mi-archivo.txt
Switched to branch 'nuevaBranch'

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ |
```



Cambiando a nueva branch y agregando un nuevo archivo llamado *nuevaBranch.txt*.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
M       02 Trabajo Colaborativo/mi-archivo.txt
Switched to branch 'nuevaBranch'

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ git add .

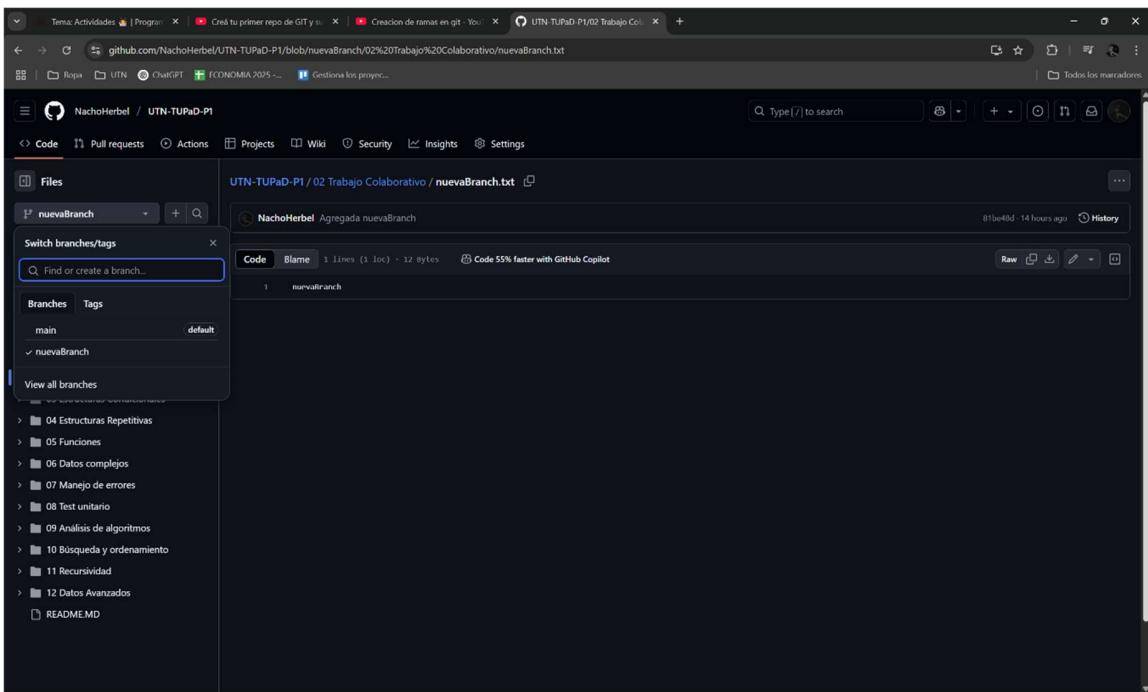
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ git status
On branch nuevaBranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Screenshots/Act2.10.png
    new file:   Screenshots/Act2.11.png
    new file:   Screenshots/Act2.12.png
    new file:   Screenshots/Act2.13.png
    new file:   Screenshots/Act2.14.png
    new file:   Screenshots/Act2.15.png
    new file:   Screenshots/Act2.4.png
    new file:   Screenshots/Act2.5.png
    new file:   Screenshots/Act2.6.png
    new file:   Screenshots/Act2.7.png
    new file:   Screenshots/Act2.8.png
    new file:   Screenshots/Act2.9.png
  modified:    mi-archivo.txt
    new file:   nuevaBranch.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ |
```

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
new file:   Screenshots/Act2.4.png
new file:   Screenshots/Act2.5.png
new file:   Screenshots/Act2.6.png
new file:   Screenshots/Act2.7.png
new file:   Screenshots/Act2.8.png
new file:   Screenshots/Act2.9.png
modified:   mi-archivo.txt
new file:   nuevaBranch.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ git commit -m "Agregada nuevaBranch"
[nuevaBranch 81be48d] Agregada nuevaBranch
14 files changed, 3 insertions(+)
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.10.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.11.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.12.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.13.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.14.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.15.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.4.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.5.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.6.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.7.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.8.png
create mode 100644 02 Trabajo Colaborativo/Screenshots/Act2.9.png
create mode 100644 02 Trabajo Colaborativo/nuevaBranch.txt

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaBranch)
$ |
```



Subiendo el nuevo archivo y la nueva branch al repositorio remoto.

3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

- Ve a GitHub e inicia sesión en tu cuenta.
- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".
- Haz clic en "Create repository".

Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuuario/conflict-exercise.git>).
- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando:

```
git clone https://github.com/tuusuuario/conflict-exercise.git
```

- Entra en el directorio del repositorio:

```
cd conflict-exercise
```

Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch:

```
git checkout -b feature-branch
```

- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:

Este es un cambio en la feature branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in feature-branch"
```

Paso 4: Volver a la rama principal y editar el mismo archivo

- Cambia de vuelta a la rama principal (main):

```
git checkout main
```

- Edita el archivo README.md de nuevo, añadiendo una línea diferente:

Este es un cambio en la main branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in main branch"
```

Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de la feature-branch en la rama main:

```
git merge feature-branch
```

- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

```
<<<<< HEAD
```

Este es un cambio en la main branch.

```
=====
```

Este es un cambio en la feature branch.

```
>>>>> feature-branch
```

- Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.

- Edita el archivo para resolver el conflicto y guarda los cambios (Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).

- Añade el archivo resuelto y completa el merge:

```
git add README.md
```

```
git commit -m "Resolved merge conflict"
```

Paso 7: Subir los cambios a GitHub

- Sube los cambios de la rama main al repositorio remoto en GitHub:

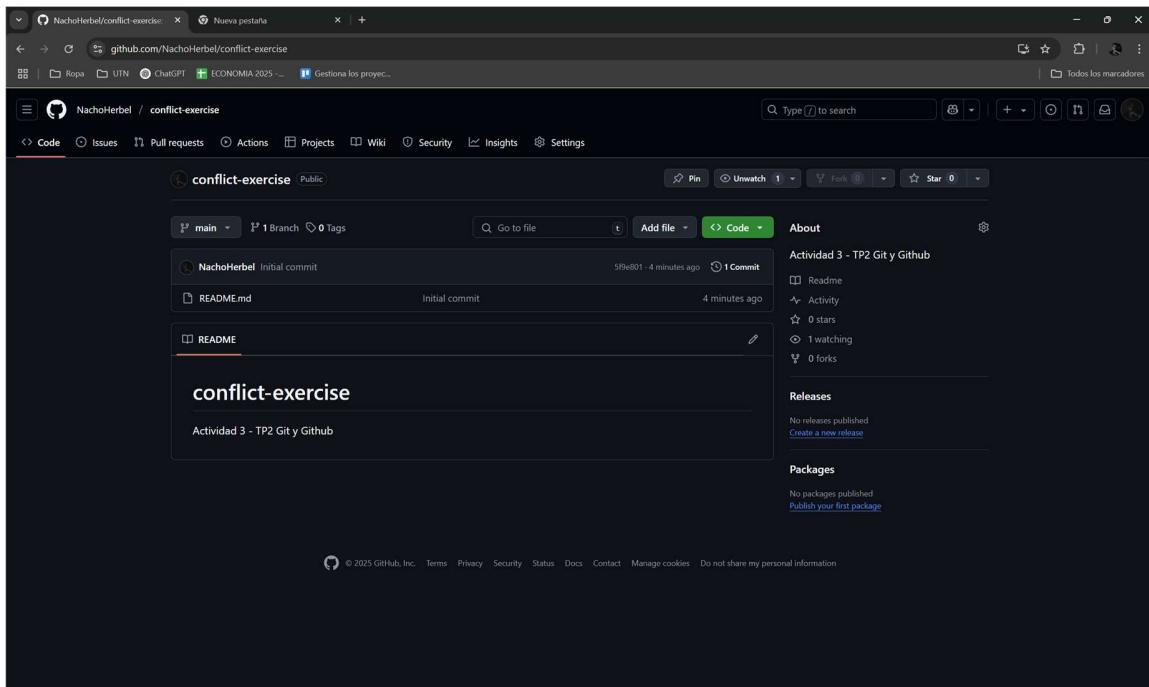
```
git push origin main
```

- También sube la feature-branch si deseas:

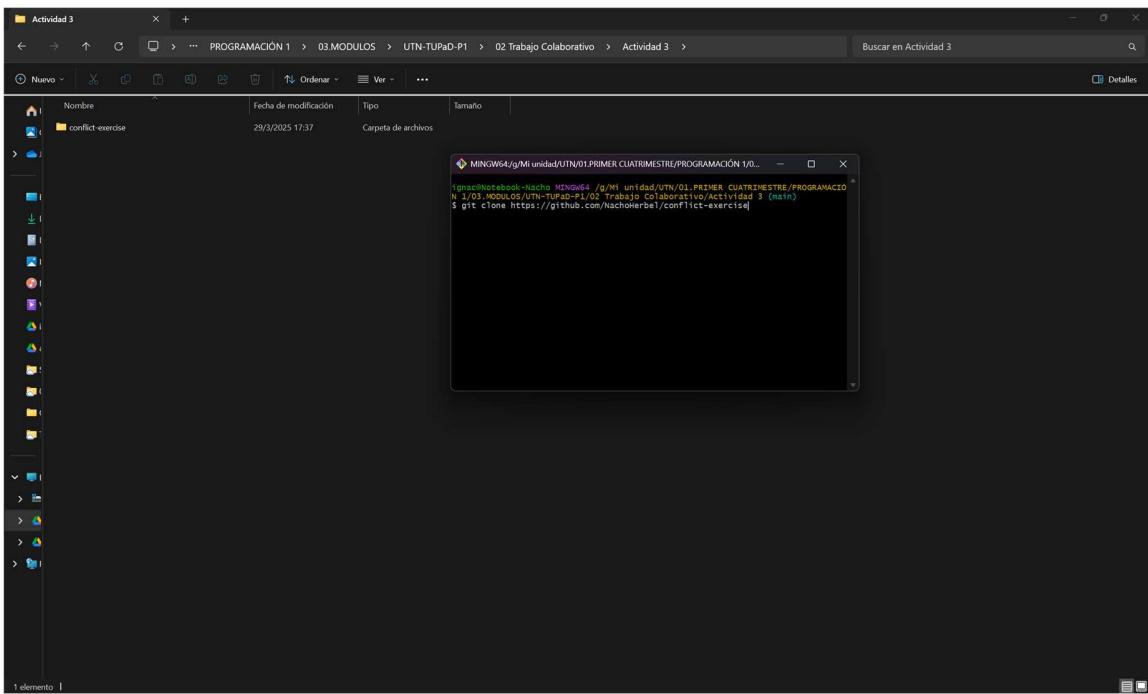
```
git push origin feature-branch
```

Paso 8: Verificar en GitHub

- Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.
- Puedes revisar el historial de commits para ver el conflicto y su resolución.



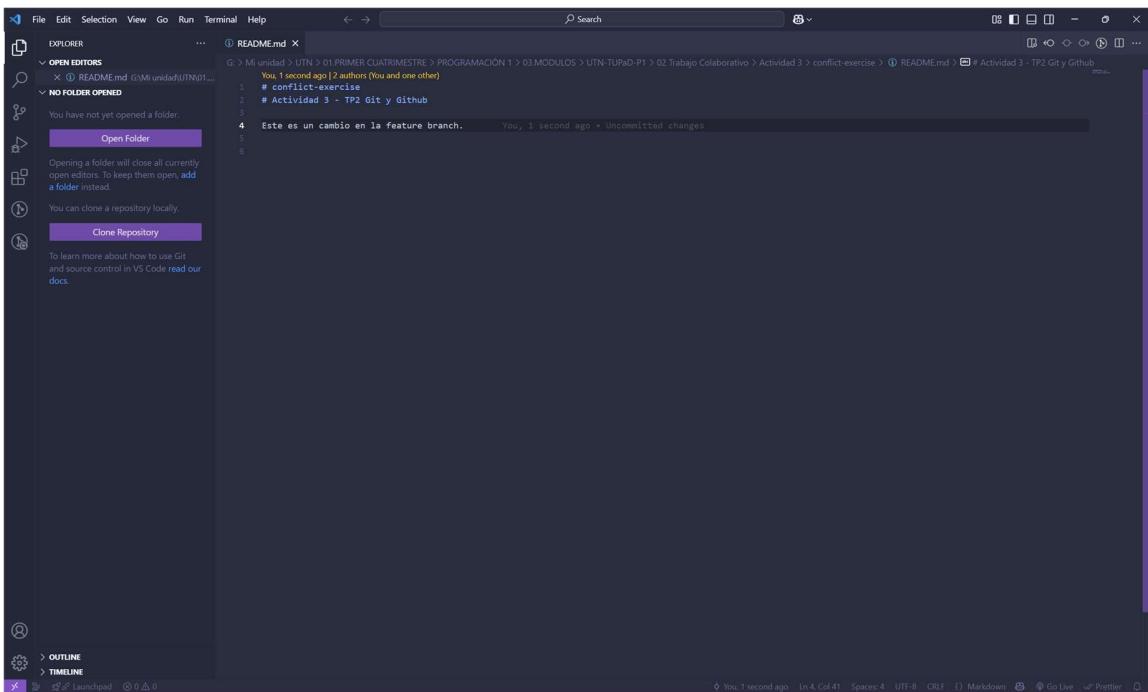
Creando un nuevo repositorio *conflict-exercise* con un archivo README.



Clonando el repositorio remoto en la maquina local.

```
ignaci@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3 (main) $ cd conflict-exercise ignaci@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main) $ git checkout -b feature-branch Switched to a new branch 'feature-branch' ignaci@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (feature-branch) $
```

Entrando en el directorio del repositorio. Creando una nueva rama llamada *feature-branch* y cambiando a ella.



Abriendo el archivo README.md en VSC y añadiendo una nueva línea.

```
$ cd conflict-exercise
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ git add README.md

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ git commit -m "Added a line in feature-branch"
[feature-branch 9aa3d25] Added a line in feature-branch
 1 file changed, 4 insertions(+), 1 deletion(-)

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...
$ |
```

A screenshot of a terminal window titled 'MINGW64:g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0...'. The window shows a series of commands being run in a shell environment. It starts with navigating to a directory, then creating a new branch named 'feature-branch'. After switching to this branch, it adds the 'README.md' file to the staging area and performs a commit with the message 'Added a line in feature-branch'. The commit hash '9aa3d25' is shown, along with the details of the change: one file changed, four insertions, and one deletion.

Guardando los cambios en la staging area y haciendo un commit.

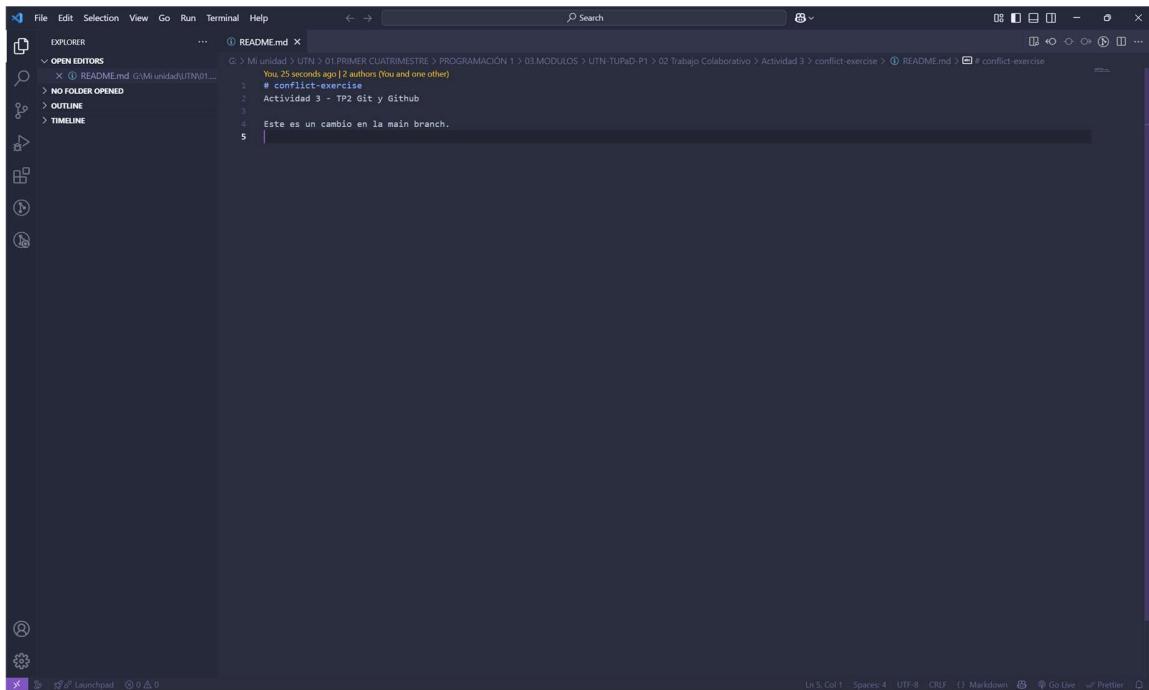
```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - □ X
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 9aa3d25] Added a line in feature-branch
 1 file changed, 4 insertions(+), 1 deletion(-)

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - □ X
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (feature-branch)
$ git branch
* feature-branch
  main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - □ X
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - □ X
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main)
$ |
```

Voviendo a la rama principal *main*.



Editando el archivo README.md, añadiendo una línea diferente.

```
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN  
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis  
e (feature-branch)  
$ git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
  
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN  
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis  
e (main)  
$ git add .  
  
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN  
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis  
e (main)  
$ git commit -m "Added a line in main branch"  
[main e26e942] Added a line in main branch  
 1 file changed, 2 insertions(+)  
  
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN  
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis  
e (main)  
$
```

Guardando los cambios en el archivo README.md y haciendo un nuevo commit.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — □ X
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main)
$ git add .

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main e26e942] Added a line in main branch
 1 file changed, 2 insertions(+)

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main)
$ git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercise (main|MERGING)
$ |
```

A screenshot of the Visual Studio Code interface. The title bar shows "README.md". The left sidebar has sections for "OPEN EDITORS", "NO FOLDER OPENED", "OUTLINE", and "TIMELINE". The main editor area displays the following code:

```
1 # conflict-exercise
2 <<<< HEAD (Current Change)
3 Actividad 3 - TP2 Git y Github
4
5 Este es un cambio en la main branch.
6 =====
7 # Actividad 3 - TP2 Git y Github
8
9 Este es un cambio en la feature branch.
10
11 >>>> feature-branch (Incoming Change)
12
```

The status bar at the bottom indicates "In 12 Col 1 Spaces 4 UTF-8 OUT Markdown Go Live Profile".

Haciendo un merge y generando un conflicto.

A screenshot of the Visual Studio Code interface. The title bar shows "README.md". The left sidebar has sections for "OPEN EDITORS", "NO FOLDER OPENED", "OUTLINE", and "TIMELINE". The main editor area displays the following code:

```
1 # conflict-exercise
2 Esta es la resolución del conflicto.
3
```

The status bar at the bottom indicates "In 2 Col 37 Spaces 4 UTF-8 OUT Markdown Go Live Profile".

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - X
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main|MERGING)
$ git commit -m "Resolved merge conflict"
U README.md
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main|MERGING)
$ git add README.md

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main 7fcf6cc] Resolved merge conflict

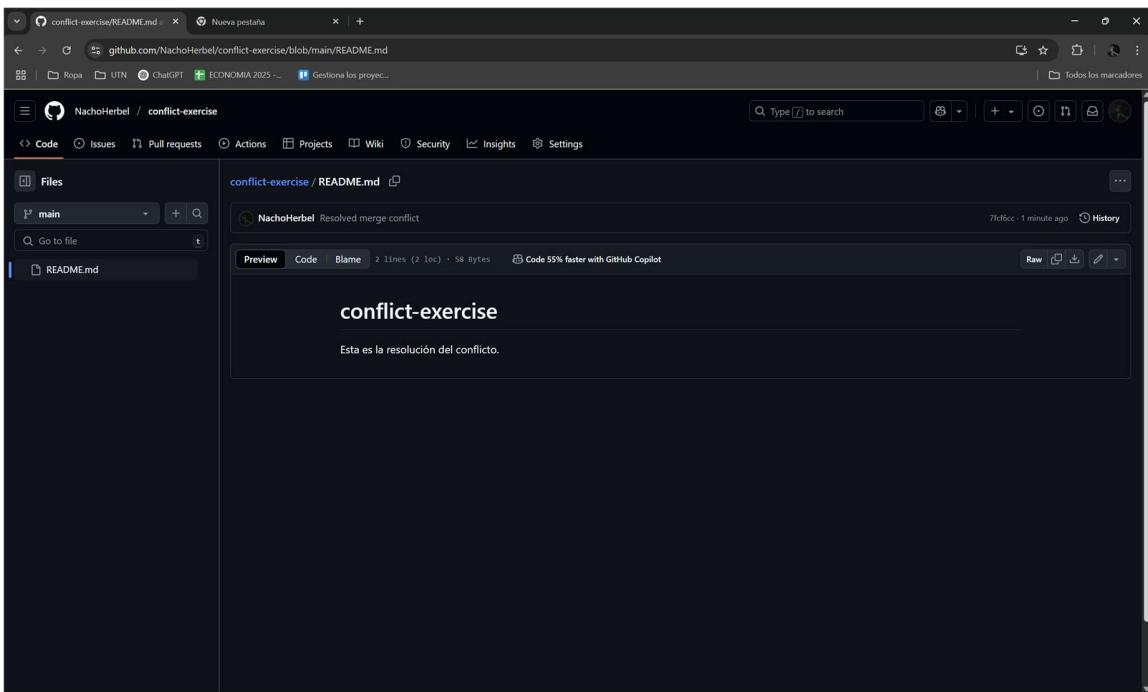
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main)
$
```

Resolviendo el conflicto.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... - X
ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main 7fcf6cc] Resolved merge conflict

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 24 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 900 bytes | 31.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/NachoHerbel/conflict-exercise
  5f9e801..7fcf6cc  main -> main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main)
$ |
```

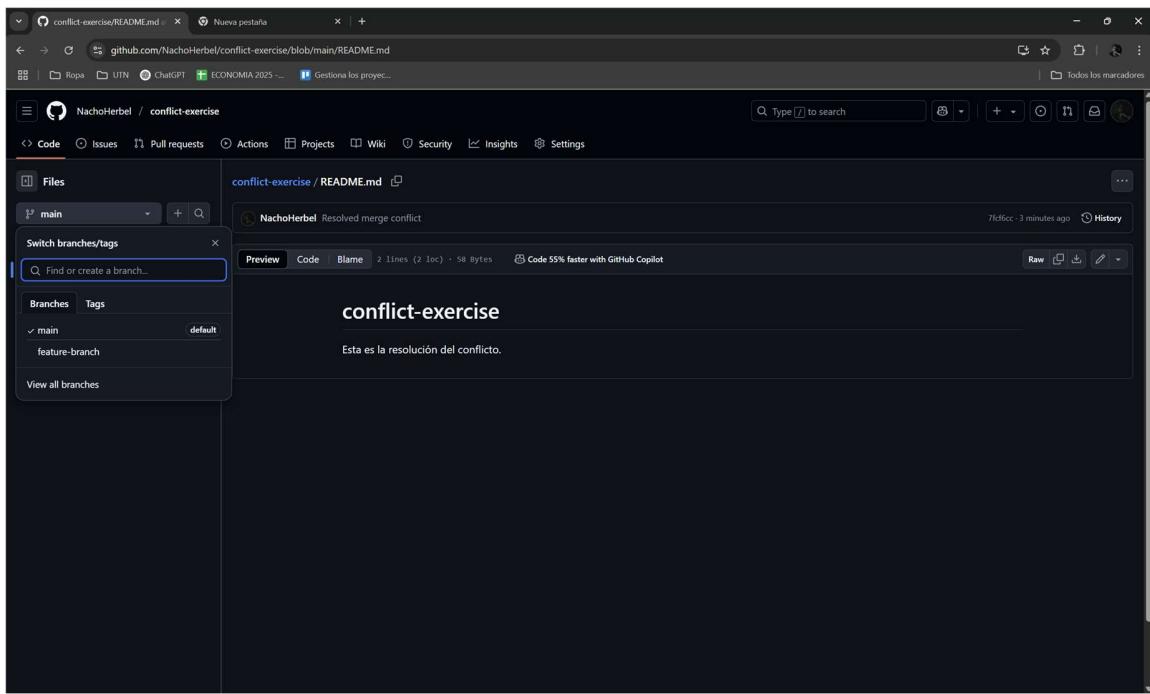


Subiendo los cambios de la rama *main* a GitHub.

```
MINGW64:/g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓN 1/0... — □ X
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 900 bytes | 31.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/NachoHerbel/conflict-exercise
  5f9e801..7fcf6cc  main -> main

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓ
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:     https://github.com/NachoHerbel/conflict-exercise/pull/new/feature-b
ranch
remote:
To https://github.com/NachoHerbel/conflict-exercise
 * [new branch]      feature-branch -> feature-branch

ignac@Notebook-Nacho MINGW64 /g/Mi unidad/UTN/01.PRIMER CUATRIMESTRE/PROGRAMACIÓ
N 1/03.MODULOS/UTN-TUPaD-P1/02 Trabajo Colaborativo/Actividad 3/conflict-exercis
e (main)
$ |
```



Subiendo a GitHub la *feature-branch* y verificando que los cambios se subieron correctamente.