

Programación I - Tp2 Actividades 2 y 3

Martín Davidenco

Actividad 2 - Paso a paso:

- Crear un repositorio.
 - o Dale un nombre al repositorio.
 - o Elije el repositorio sea público.
 - o Inicializa el repositorio con un archivo.

Como se ve en la imagen creé el repositorio bajo el nombre "tp2-actividad2" . Público, y marqué la casilla para que se cree 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 *

martindavidenco / tp2-actividad2

✔ tp2-actividad2 is available.

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

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:

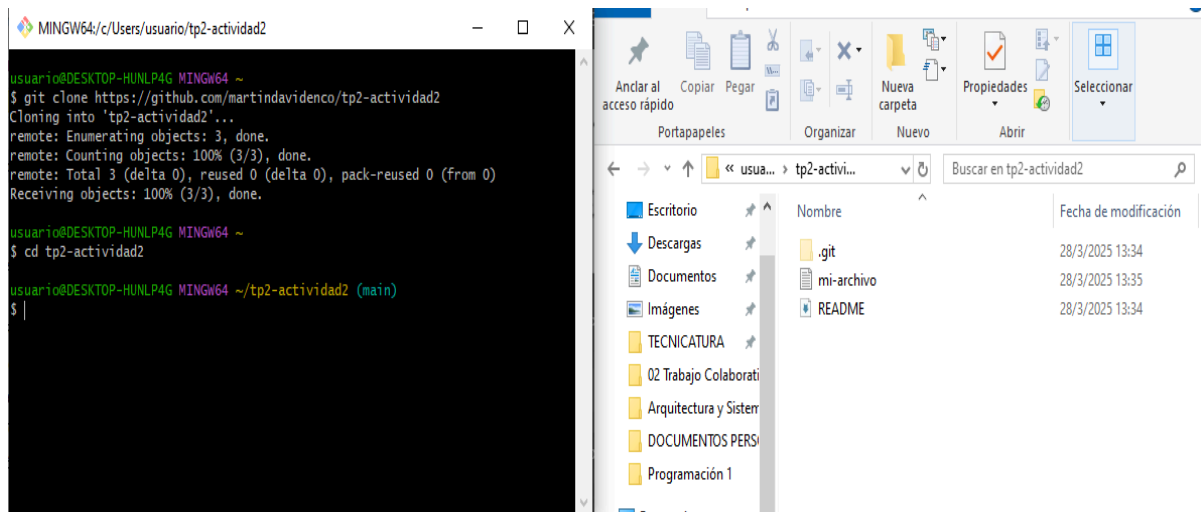
☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

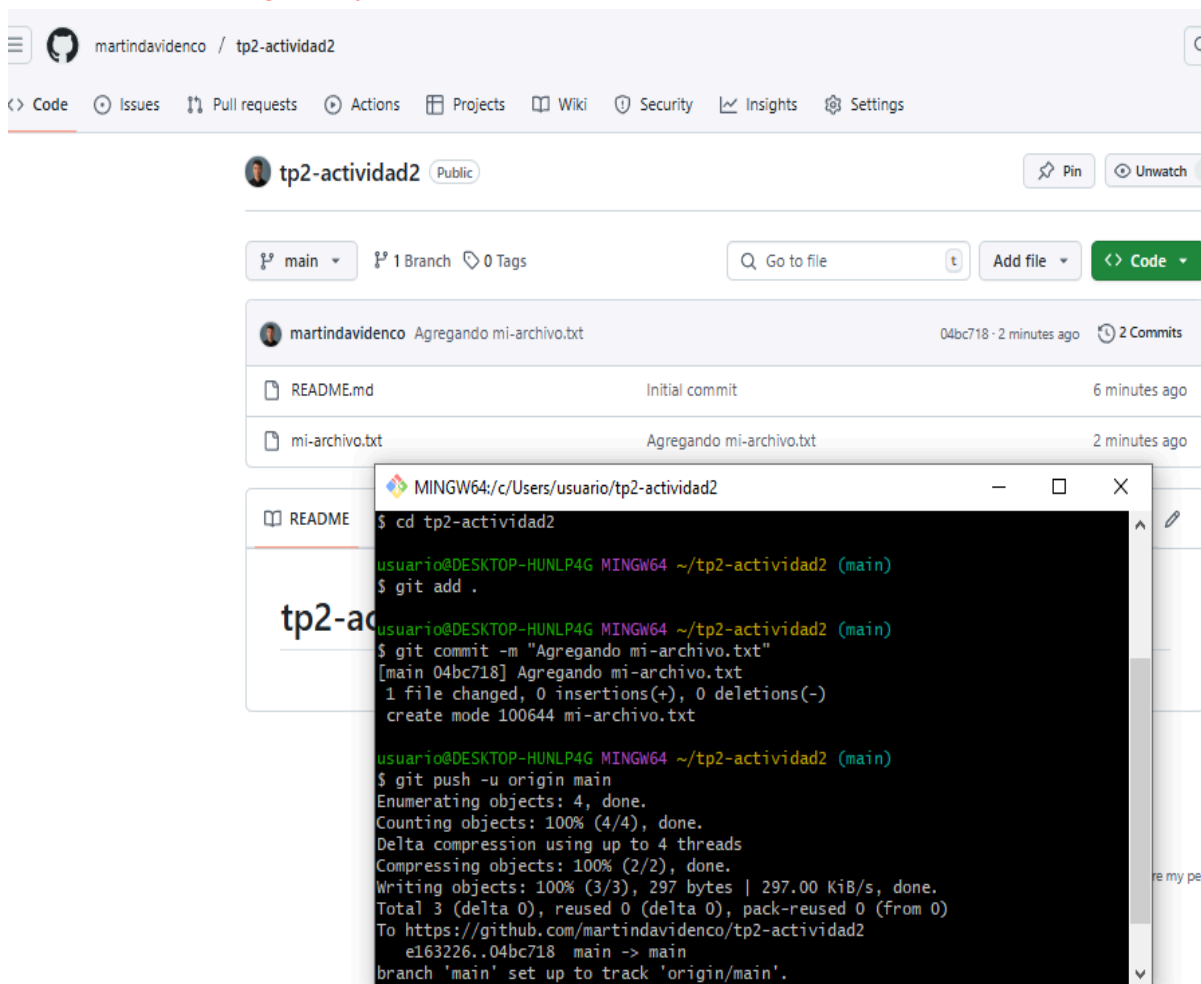
.gitignore template: None

- Agregando un Archivo
 - o Crea un archivo simple, por ejemplo, "mi-archivo.txt".
 - o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
 - o Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

En este caso primero clone desde git bash el proyecto, a continuacion fui a la ubicacion de la carpeta y le agregue el archivo "mi-archivo-txt"

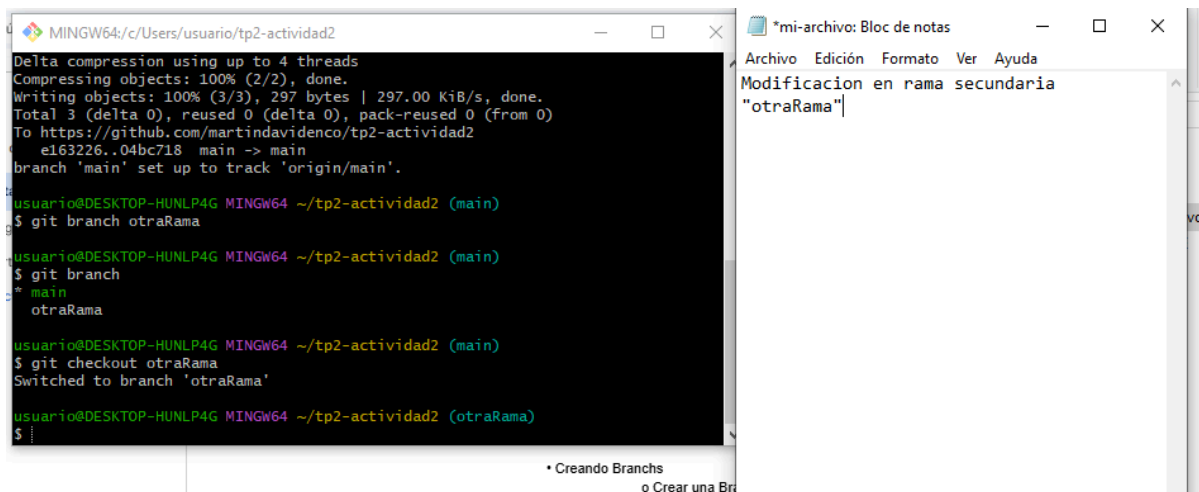


Cambios subidos a gitHub y comandos:

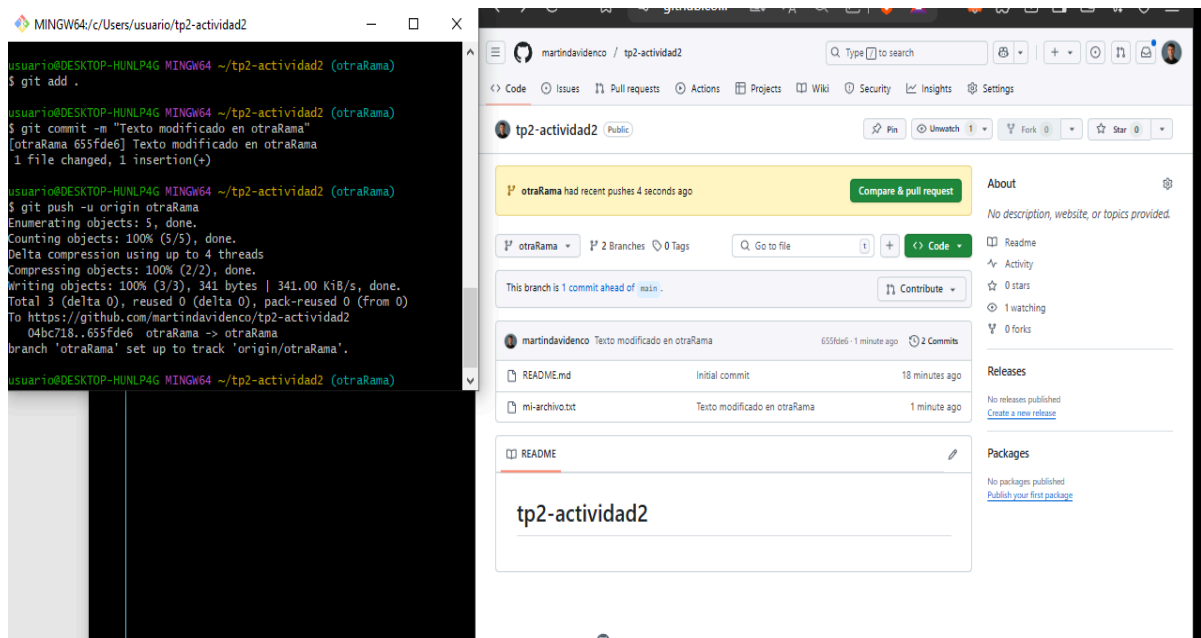


- Creando Branches
 - o Crear una Branch
 - o Realizar cambios
 - o agregar un archivo o Subir la Branch

Primero ejecuté los comandos de la izquierda para crear la nueva rama "otraRama" y posicionarme, y a la derecha se ve como modifique el archivo .txt en la rama nueva.



Hecho los cambios en la siguiente imagen se muestra los comandos para pushear esta nueva rama y tambien se ve el resultado subido a github donde se ve que hay 2 branches



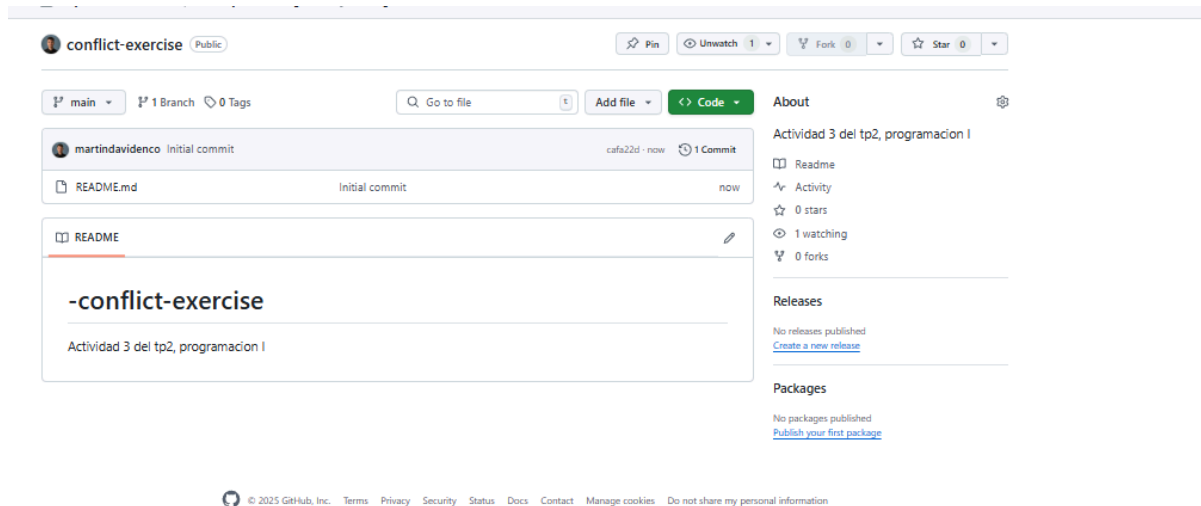
LINK REPOSITORIO:

<https://github.com/martindavidenco/tp2-actividad2>

Actividad 3 - Paso a paso:

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/tuusuario/conflict-exercise.git>)
- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando: `git clone https://github.com/tuusuario/conflict-exercise.git`
- Entra en el directorio del repositorio: `cd conflict-exercise`

```
MINGW64:/c/Users/usuario/conflict-exercise

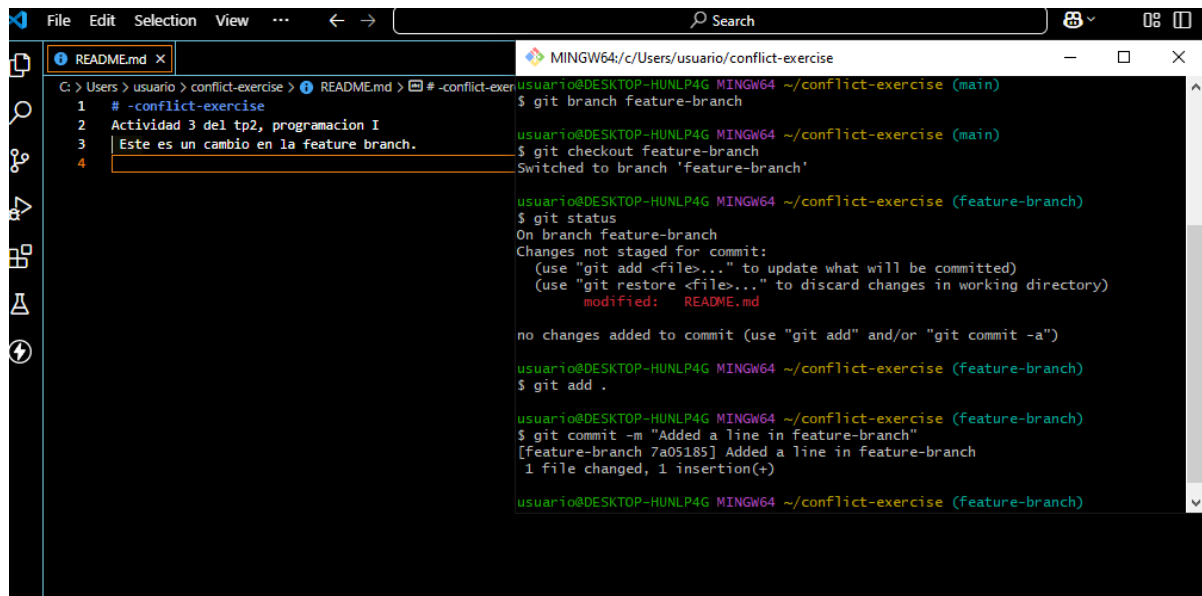
usuario@DESKTOP-HUNLP4G MINGW64 ~
$ git clone https://github.com/martindavidenco/conflict-exercise
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

usuario@DESKTOP-HUNLP4G MINGW64 ~
$ 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"`

La modificacion que se ve en el README esta hecha justo despues del comando:
“git checkout feature-branch”



The image shows a VS Code editor window with a file named README.md open. The file content is as follows:

```
1 # -conflict-exercise
2 Actividad 3 del tp2, programacion I
3 Este es un cambio en la feature branch.
4
```

The terminal window on the right shows the following commands and output:

```
usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (main)
$ git branch feature-branch

usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'

usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (feature-branch)
$ git status
On branch feature-branch
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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (feature-branch)
$ git add .

usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 7a05185] Added a line in feature-branch
1 file changed, 1 insertion(+)

usuario@DESKTOP-HUNLP4G MINGW64 ~/conflict-exercise (feature-branch)
```

Paso 4: Volver a la rama principal y editar el mismo archivo 3

- 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"`

```

C: > Users > usuario > conflict-exercise > README.md > # -conflict-exercise
1  # -conflict-exercise
2  Actividad 3 del tp2, programacion I
3  Este es un cambio en la main branch.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• PS C:\Users\usuario> cd conflict-exercise
• PS C:\Users\usuario\conflict-exercise> git branch
  feature-branch
* main
• PS C:\Users\usuario\conflict-exercise> git status
On branch main
Your branch is up to date with 'origin/main'.

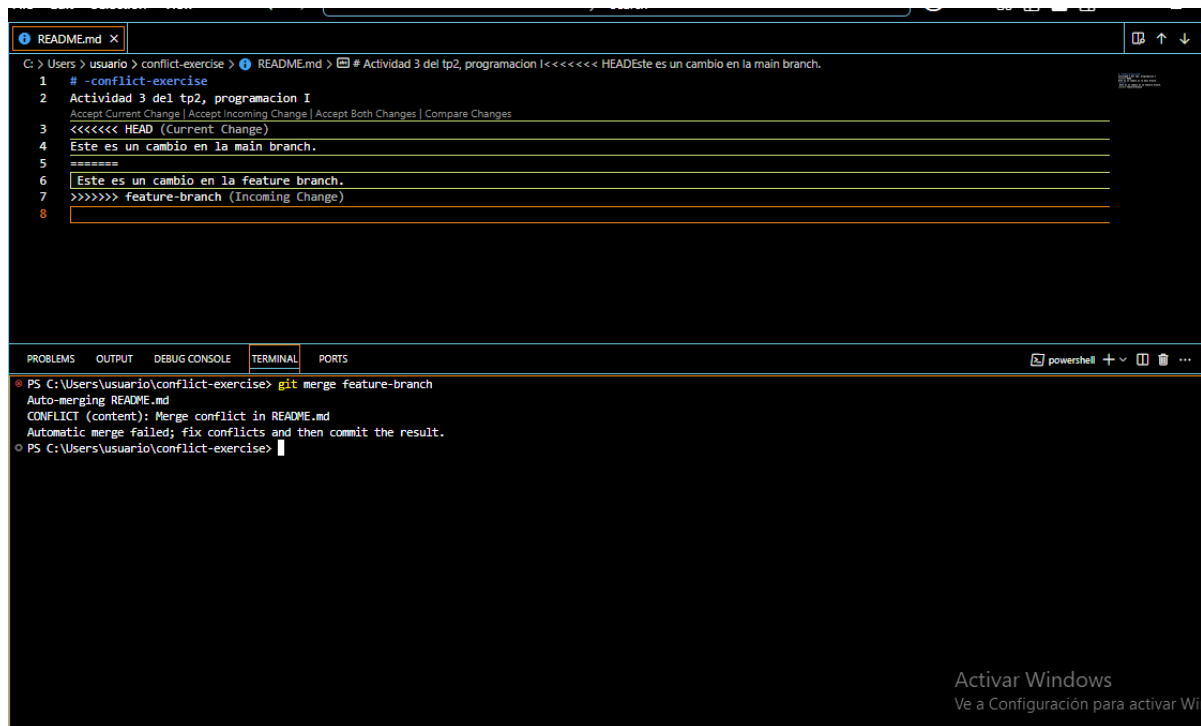
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

• PS C:\Users\usuario\conflict-exercise> git add .
• PS C:\Users\usuario\conflict-exercise> git commit -m "Added a line in main branch"
[main 624fa76] Added a line in main branch
 1 file changed, 1 insertion(+)
• PS C:\Users\usuario\conflict-exercise> 
```

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.

CONFLICTO GENERADO:



The image shows a Visual Studio Code editor window with a file named `README.md` open. The file content is as follows:

```
1 # -conflict-exercise
2 Actividad 3 del tp2, programacion I
3 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<<<< HEAD (Current Change)
5 Este es un cambio en la main branch.
6 =====
7 | Este es un cambio en la feature branch.
8 >>>>>> feature-branch (Incoming Change)
```

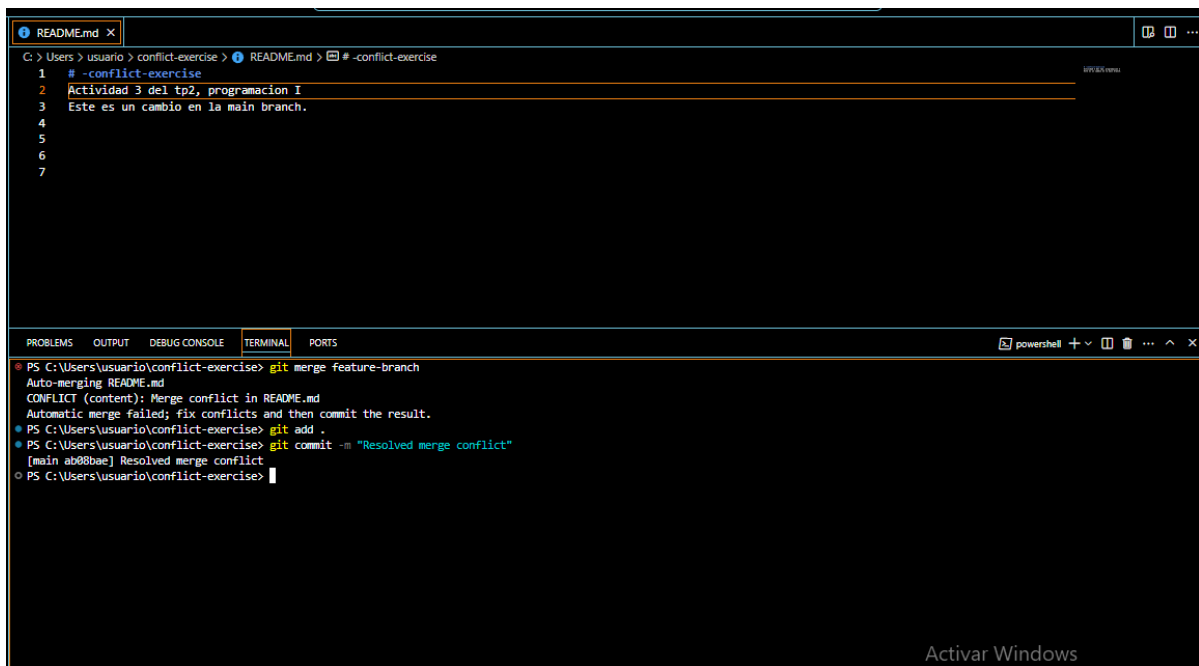
Below the editor, the `TERMINAL` tab is active, showing the output of a `git merge` command:

```
PS C:\Users\usuario\conflict-exercise> 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.
PS C:\Users\usuario\conflict-exercise>
```

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"`

En este caso decidí dejar solo lo del main



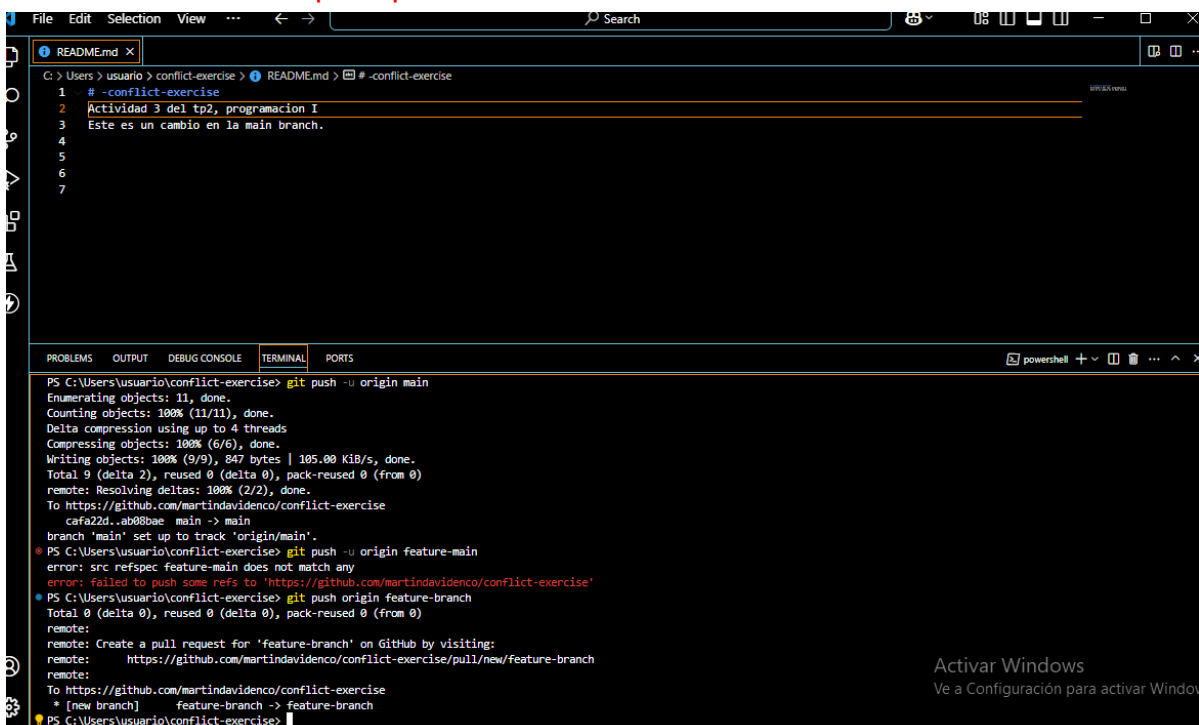
```
C:\Users\usuario> conflict-exercise > README.md > # -conflict-exercise
1 # -conflict-exercise
2 Actividad 3 del tp2, programacion I
3 Este es un cambio en la main branch.
4
5
6
7

PS C:\Users\usuario\conflict-exercise> 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.
PS C:\Users\usuario\conflict-exercise> git add .
PS C:\Users\usuario\conflict-exercise> git commit -m "Resolved merge conflict"
[main ab08bae] Resolved merge conflict
PS C:\Users\usuario\conflict-exercise>
```

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`

Pusheo tanto la rama principal como la secundaria:



```
PS C:\Users\usuario\conflict-exercise> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 847 bytes | 105.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/martindavidenco/conflict-exercise
   cafa22d..ab08bae  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\usuario\conflict-exercise> git push -u origin feature-main
error: src refspec feature-main does not match any
error: failed to push some refs to 'https://github.com/martindavidenco/conflict-exercise'
PS C:\Users\usuario\conflict-exercise> 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/martindavidenco/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/martindavidenco/conflict-exercise
 * [new branch]   feature-branch -> feature-branch
PS C:\Users\usuario\conflict-exercise>
```

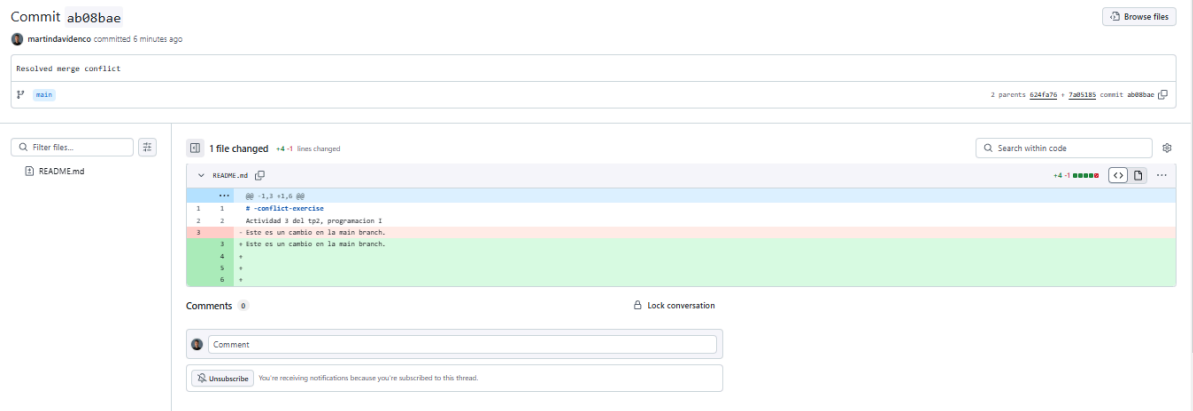
Paso 8: Verificar en GitHub

- 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.

Cambios subidos correctamente:



Historial de commits para ver conflicto y resolucion:



LINK REPOSITORIO:
<https://github.com/martindavidenco/conflict-exercise>