

Práctico 2: Git y GitHub

Actividades

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

- ¿Qué es GitHub? Es una plataforma para gestionar proyectos de software
- ¿Cómo crear un repositorio en GitHub? Creas una cuenta en github, hago clic en el icono + , escribo el nombre y descripción, publico/ privado y luego créate repository.
- ¿Cómo crear una rama en Git? Con el comando git branch
- ¿Cómo cambiar a una rama en Git? Con el comando git checkout
- ¿Cómo fusionar ramas en Git? Con el comando git merge
- ¿Cómo crear un commit en Git? Comando: git add . / git commit -m "..."
- ¿Cómo enviar un commit a GitHub? Comando: git push origin main
- ¿Qué es un repositorio remoto? Es una versión de mi repositorio de git almacenada en un servidor en línea ej: github
- ¿Cómo agregar un repositorio remoto a Git? Comando: git clone
- ¿Cómo empujar cambios a un repositorio remoto? Comando: git push
- ¿Cómo tirar de cambios de un repositorio remoto? Comando: git pull
- ¿Qué es un fork de repositorio? Es una copia de un repositorio remoto que me permite hacer cambios sin afectar el original.
- ¿Cómo crear un fork de un repositorio? Voy al repositorio original y hago clic en el botón fork, luego clono el repositorio en mi maquina local con el comando git clone, realizo los cambios (con git add . y git commit -m) y luego con git push lo subo a mi repositorio remoto.
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio? Hago clic pull requests y luego envio la solicitud haciendo clic en créate pull requests
- ¿Cómo aceptar una solicitud de extracción? Comando: merge pull request
- ¿Qué es un etiqueta en Git? Es una referencia para señalar un punto especifico en la historia de un repositorio.

- ¿Cómo crear una etiqueta en Git? Comando : git tag (nombre de la etiqueta)
- ¿Cómo enviar una etiqueta a GitHub? Comando: git push origin (nombre de la etiqueta)
- ¿Qué es un historial de Git? Es un registro de todos los cambios realizados en un repositorio.
- ¿Cómo ver el historial de Git? Comando: git log
- ¿Cómo buscar en el historial de Git? Comando: git log --grep= "palabra clave" (por ej para buscar por palabra/mensaje).
- ¿Cómo borrar el historial de Git? Elimino el archivo .git
- ¿Qué es un repositorio privado en GitHub? Es un repositorio al que solo pueden acceder personas que tiene un permiso explícito.
- ¿Cómo crear un repositorio privado en GitHub? Al crearlo selecciono : private.
- ¿Cómo invitar a alguien a un repositorio privado en GitHub? Settings > Manage Access > Invite a collaborator
- ¿Qué es un repositorio público en GitHub? es un repositorio cuyo contenido es accesible para cualquier persona en la web, sin necesidad de permisos especiales.
- ¿Cómo crear un repositorio público en GitHub? Al crearlo selección : public.
- ¿Cómo compartir un repositorio público en GitHub? crear el repositorio, verás la URL en la parte superior de la página de tu repositorio. Puedes compartir esta URL con otras personas para que accedan al repositorio.

2) Realizar la siguiente actividad:


- Crear un repositorio.
 - Dale un nombre al repositorio. ○ Elige el repositorio sea público.
 - Inicializa el repositorio con un archivo.

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 *



 Astrid-Schneider / primer-repo-git

✓ primer-repo-git is available.

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

Description (optional)

Repositorio de Programación 1 - tp 2

- ☒  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.](#)

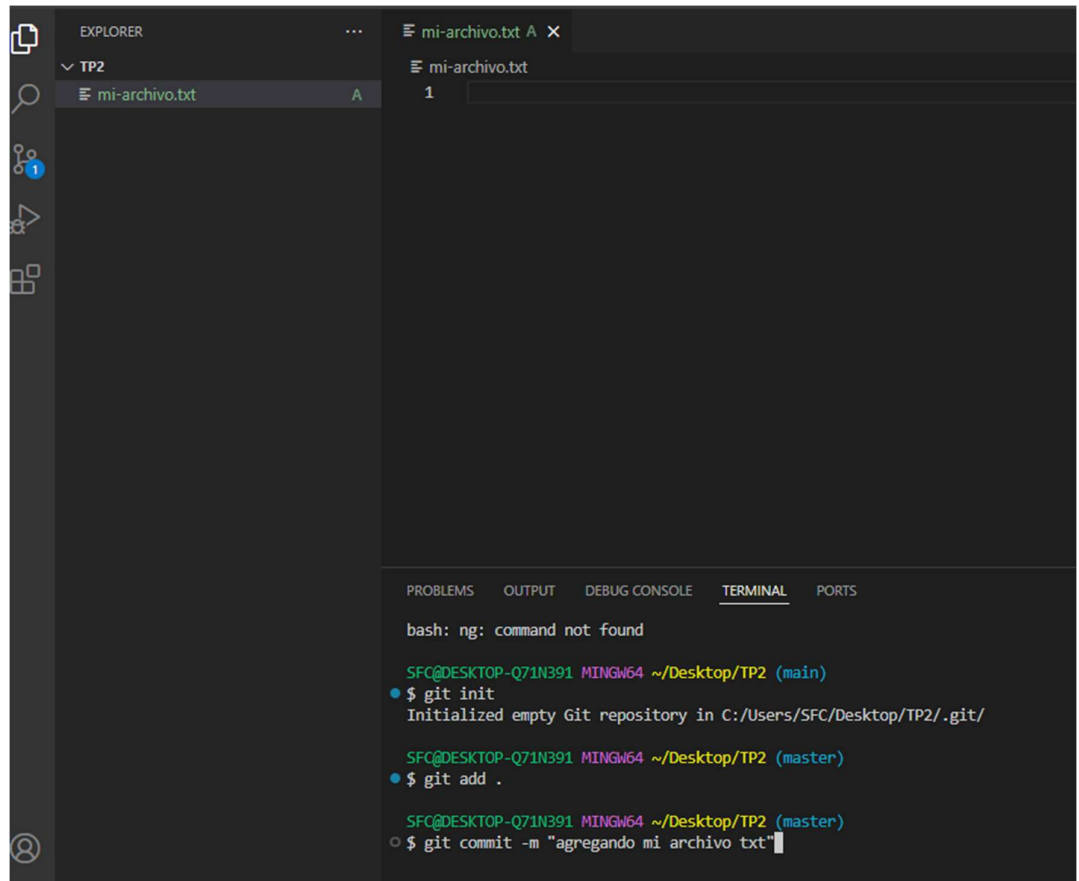
This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

- 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



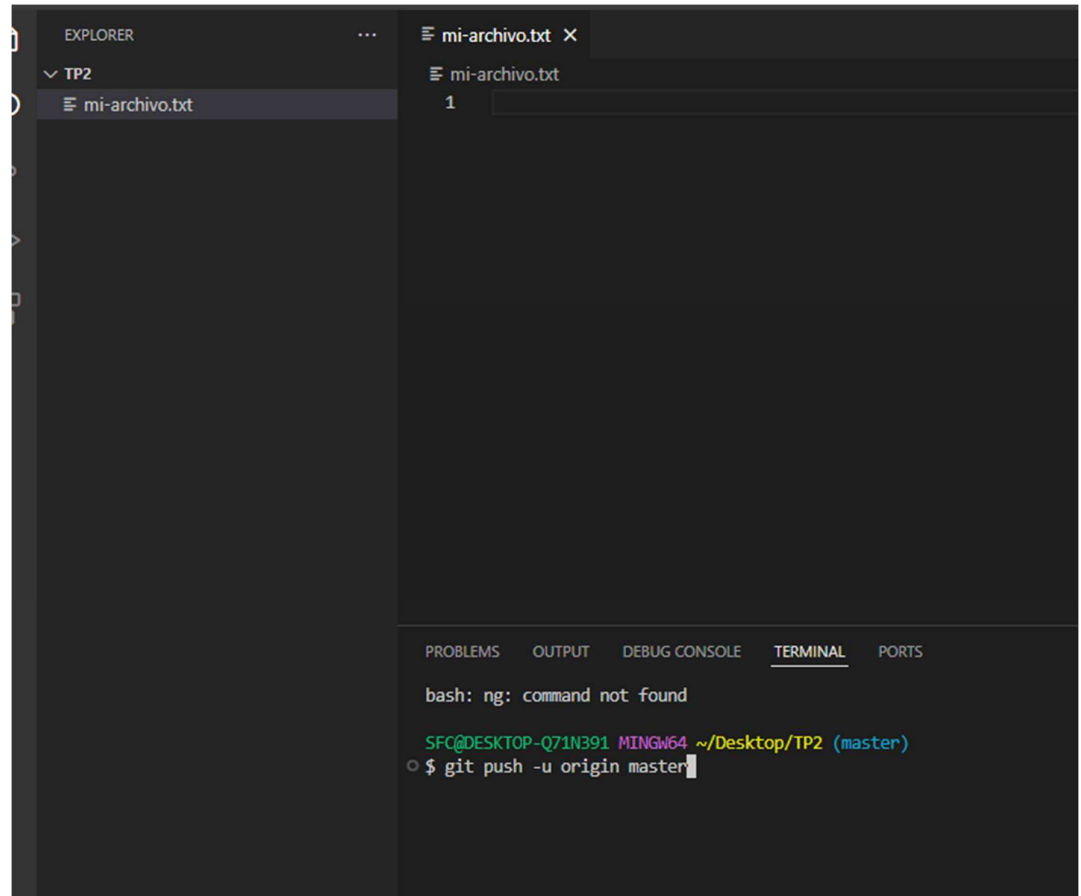
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a folder named 'TP2' containing a file 'mi-archivo.txt'. The file is open in the editor, showing a single line with the number '1'. At the bottom, the Terminal pane is active, displaying the following commands and output:

```
bash: ng: command not found

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (main)
• $ git init
  Initialized empty Git repository in C:/Users/SFC/Desktop/TP2/.git/

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (master)
• $ git add .

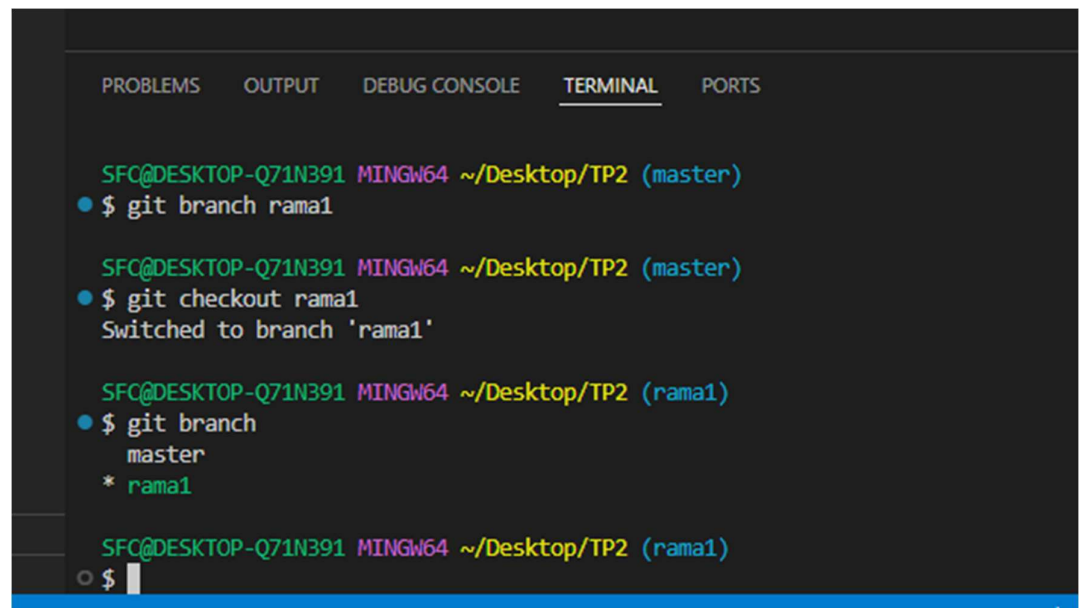
SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (master)
○ $ git commit -m "agregando mi archivo txt"
```



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a folder named 'TP2' containing a file 'mi-archivo.txt'. The main editor area displays the content of 'mi-archivo.txt', which is a single line: '1'. At the bottom, the Terminal panel is active, showing the following output:

```
bash: ng: command not found

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (master)
$ git push -u origin master
```



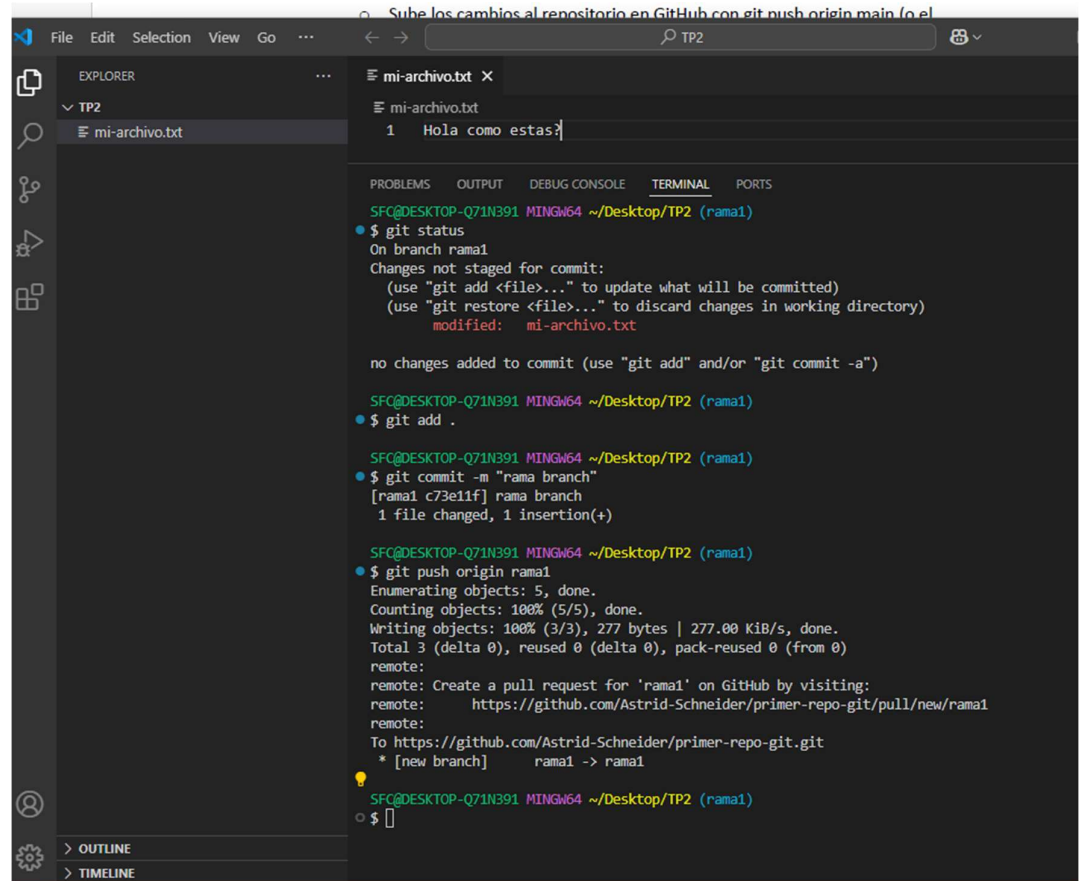
The screenshot shows the Visual Studio Code Terminal panel with the following output:

```
SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (master)
$ git branch rama1

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (master)
$ git checkout rama1
Switched to branch 'rama1'

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
$ git branch
  master
* rama1

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
$
```



```
File Edit Selection View Go ... TP2
EXPLORER
  TP2
    mi-archivo.txt
mi-archivo.txt
  1 Hola como estas?

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
• $ git status
On branch rama1
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:   mi-archivo.txt

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

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
• $ git add .

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
• $ git commit -m "rama branch"
[rama1 c73e11f] rama branch
1 file changed, 1 insertion(+)

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
• $ git push origin rama1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'rama1' on GitHub by visiting:
remote:   https://github.com/Astrid-Schneider/primer-repo-git/pull/new/rama1
remote:
To https://github.com/Astrid-Schneider/primer-repo-git
 * [new branch]      rama1 -> rama1

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/TP2 (rama1)
o $
```

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/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`

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:

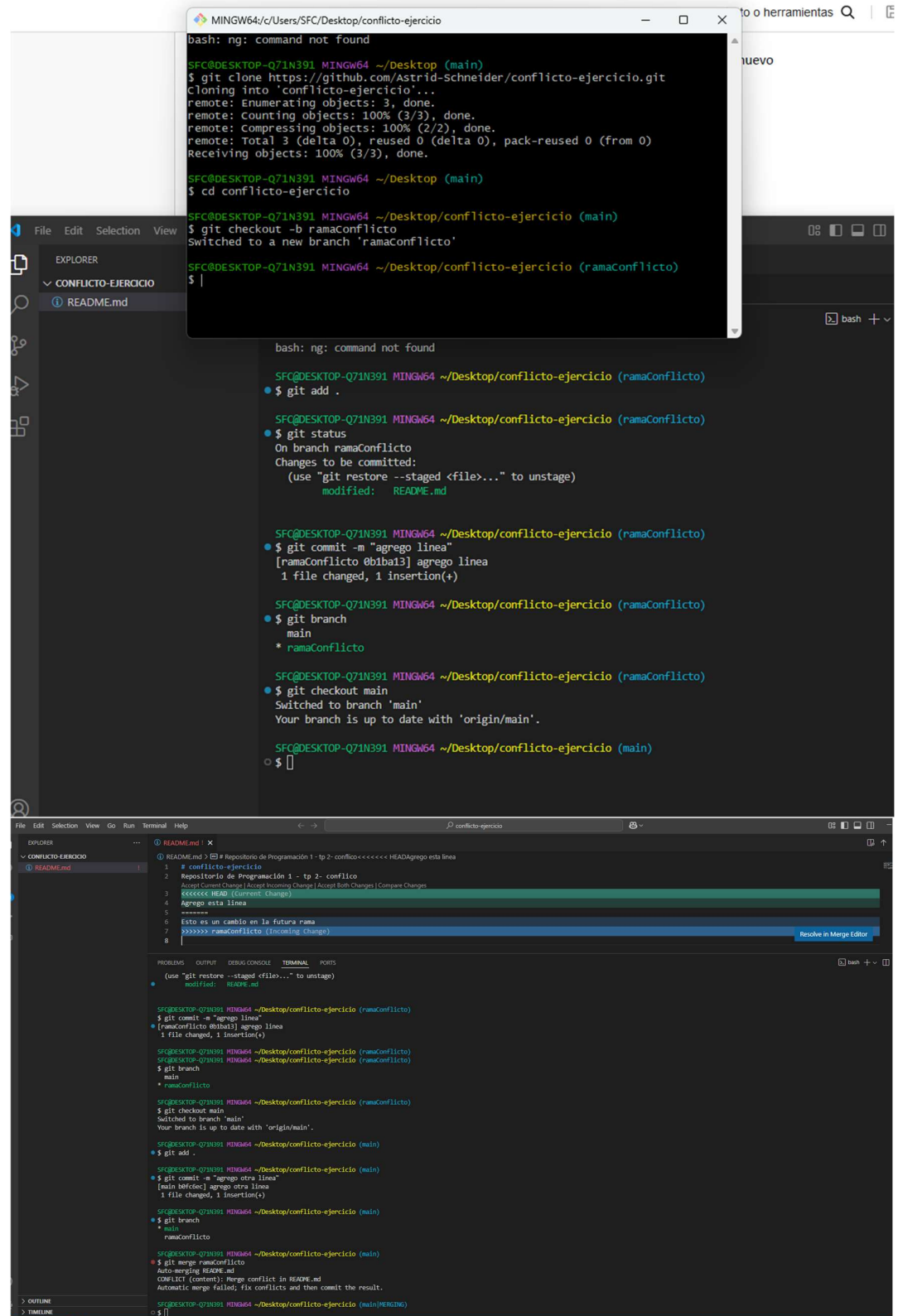
```
git push origin main
```

- 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.



```
MINGW64/c/Users/SFC/Desktop/conflicto-ejercicio
bash: ng: command not found

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop (main)
$ git clone https://github.com/Astrid-Schneider/conflicto-ejercicio.git
Cloning into 'conflicto-ejercicio'...
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.

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop (main)
$ cd conflicto-ejercicio

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ git checkout -b ramaConflicto
Switched to a new branch 'ramaConflicto'

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ |

bash: ng: command not found

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git add .

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git status
On branch ramaConflicto
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git commit -m "agrego linea"
[ramaConflicto 0b1ba13] agrego linea
1 file changed, 1 insertion(+)

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git branch
* ramaConflicto
main

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ |
```

Explorer: CONFLICTO-EJERCICIO, README.md

Terminal: bash

README.md

```
1 # Repositorio de Programación 1 - tp 2- conflicto
2 # Repositorio de Programación 1 - tp 2- conflicto
3 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 Agrego esta linea.
5 =====
6 Esto es un cambio en la futura rama
7 >>>>>> ramaConflicto (incoming change)
8
```

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

```
SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git commit -m "agrego linea"
[ramaConflicto 0b1ba13] agrego linea
1 file changed, 1 insertion(+)

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git branch
* ramaConflicto
main

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (ramaConflicto)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

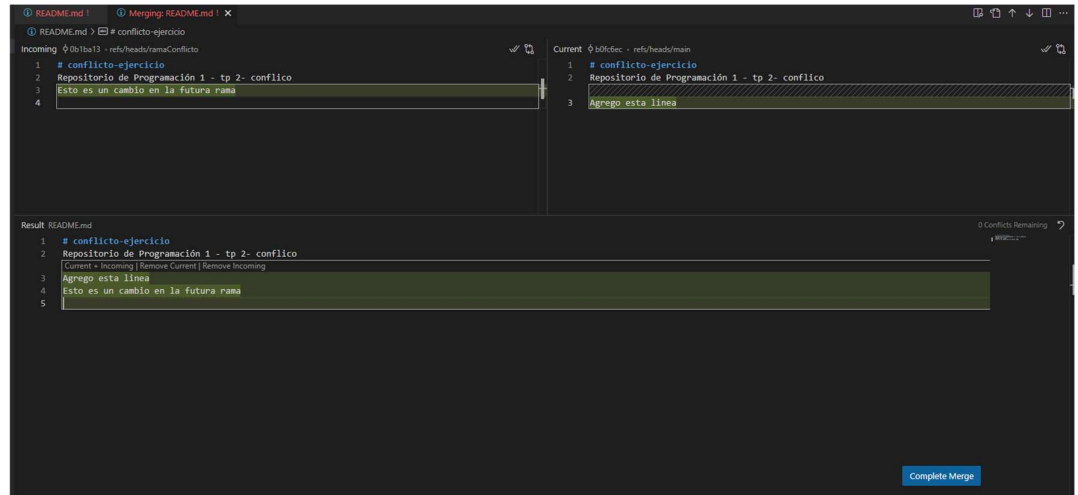
SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ git add .

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ git commit -m "agrego otra linea"
[main b1bf0ec] agrego otra linea
1 file changed, 1 insertion(+)

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ git branch
* main
ramaConflicto

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main)
$ git merge ramaConflicto
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

SFC@DESKTOP-Q71N391 MINGW64 ~/Desktop/conflicto-ejercicio (main/MERGEING)
$ |
```



```
README.md 1 x Merging README.md 1 x
# conflicto-ejercicio
1 # conflicto-ejercicio
2 Repositorio de Programación 1 - tp 2- conflicto
3 Esto es un cambio en la futura rama
4

Current refs/heads/main
1 # conflicto-ejercicio
2 Repositorio de Programación 1 - tp 2- conflicto
3 Agrego esta linea

Result README.md
1 # conflicto-ejercicio
2 Repositorio de Programación 1 - tp 2- conflicto
3 Agrego esta linea
4 Esto es un cambio en la futura rama
5

0 Conflicts Remaining
Complete Merge
```