

```
PS C:\Users\Ximena>  
git checkout feature-  
branch
```



git



Nombre y Apellido: Ximena Sosa
Carrera: Tecnicatura Univ.en Programación
Repositorios utilizados:
<https://github.com/xms44/UTN-TUPaD-P1>
<https://github.com/xms44/XimenaSosaTP2>
<https://github.com/xms44/conflict-exercise>

ÍNDICE

ACTIVIDAD 1	3
¿Qué es GitHub?	3
¿Cómo crear un repositorio en GitHub?	3
¿Cómo crear una rama en Git?	3
¿Cómo cambiar a una rama en Git?	3
¿Cómo fusionar ramas en Git?	3
¿Cómo crear un commit en Git?	3
¿Cómo enviar un commit a GitHub?	4
¿Qué es un repositorio remoto?	4
¿Cómo agregar un repositorio remoto a Git?	4
¿Cómo empujar cambios a un repositorio remoto?	4
¿Cómo tirar cambios de un repositorio remoto?	4
¿Qué es un fork de repositorio?	4
¿Cómo crear un fork de un repositorio?	4
¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?	5
¿Cómo aceptar una solicitud de extracción?	5
¿Qué es una etiqueta en Git?	5
¿Cómo crear una etiqueta en Git?	5
¿Cómo enviar una etiqueta a GitHub?	5
¿Qué es un historial de Git?	5
¿Cómo ver el historial de Git?	5
¿Cómo buscar en el historial de Git?	6
¿Cómo borrar el historial de Git?	6
¿Qué es un repositorio privado en GitHub?	6
¿Cómo crear un repositorio privado en GitHub?	6
¿Cómo invitar a alguien a un repositorio privado en GitHub?	6
¿Qué es un repositorio público en GitHub?	6
¿Cómo crear un repositorio público en GitHub?	7
¿Cómo compartir un repositorio público en GitHub?	7
ACTIVIDAD 2	7
Crear un repositorio	7
Agregando un Archivo	8
Agregando un Archivo	9
Creando Branchs	10
ACTIVIDAD 3	11
Paso 1: Crear un repositorio en GitHub	11
Paso 3: Crear una nueva rama y editar un archivo	11
Paso 4: Volver a la rama principal y editar el mismo archivo	12
Paso 5: Hacer un merge y generar un conflicto	12
Paso 6: Resolver el conflicto	13
Paso 7: Subir los cambios a GitHub	13
Paso 8: Verificar en GitHub	14

ACTIVIDAD 1

¿Qué es GitHub?

GitHub es una plataforma online que permite almacenar y gestionar proyectos que usan Git. Es como una "nube" donde los desarrolladores pueden subir sus repositorios, colaborar con otros, hacer revisiones de código y llevar un control de versiones. También permite trabajar en equipo de manera remota y organizar tareas mediante herramientas como issues, pull requests y proyectos.

¿Cómo crear un repositorio en GitHub?

1. Iniciar sesión en GitHub.
2. Hacer clic en el botón "New" o "Create repository".
3. Ingresar el nombre del repositorio.
4. Elegir si será público o privado.
5. (Opcional) Inicializar con un archivo README.
6. Hacer clic en "Create repository".

¿Cómo crear una rama en Git?

Utilizando el siguiente comando: **git branch nombre-de-la-rama**

¿Cómo cambiar a una rama en Git?

Utilizando el comando: **git checkout nombre-de-la-rama**

¿Cómo fusionar ramas en Git?

Primero, cambiar a la rama donde queremos traer los cambios (por ejemplo, master):

git checkout master

Después, hacer el merge: **git merge nombre-de-la-rama**

¿Cómo crear un commit en Git?

Después de agregar archivos al área de preparación (git add), se usa:

git commit -m "Mensaje descriptivo del cambio"

¿Cómo enviar un commit a GitHub?

Primero se hace el commit local y luego se sube con:

git push origin nombre-de-la-rama

¿Qué es un repositorio remoto?

Es un repositorio que está alojado en un servidor (por ejemplo, GitHub). Permite guardar el proyecto en la nube y compartirlo con otros colaboradores. A diferencia del repositorio local (que está en tu computadora), el remoto facilita el trabajo en equipo y el respaldo del código.

¿Cómo agregar un repositorio remoto a Git?

Se usa el siguiente comando: **git remote add origin URL-del-repositorio**

Por ejemplo:

git remote add origin https://github.com/xms44/UTN-TUPaD-P1

¿Cómo empujar cambios a un repositorio remoto?

Después de hacer el commit local, se usa: **git push origin nombre-de-la-rama**

¿Cómo tirar cambios de un repositorio remoto?

Se utiliza: **git pull origin nombre-de-la-rama**

¿Qué es un fork de repositorio?

Un fork es una copia completa de un repositorio que se crea dentro de tu cuenta de GitHub. Sirve para probar cambios sin afectar el proyecto original, proponer mejoras o trabajar en proyectos a los que no tenés permisos de edición directo.

¿Cómo crear un fork de un repositorio?

1. Ir al repositorio público que querés copiar.
2. Hacer clic en el botón "Fork" (arriba a la derecha en GitHub).
3. Elegir tu cuenta para crear una copia en tu propio GitHub.

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

Después de hacer cambios en tu rama o en tu fork, en GitHub:

1. Ir a la página del repositorio original.
2. Hacer clic en "New pull request".
3. Elegir la rama donde hiciste los cambios.
4. Completar el título y una breve descripción.
5. Hacer clic en "Create pull request".

¿Cómo aceptar una solicitud de extracción?

1. Ir al repositorio donde recibiste el pull request.
2. Revisar los cambios propuestos.
3. Si estás de acuerdo, hacer clic en "Merge pull request".
4. Confirmar con "Confirm merge".

¿Qué es una etiqueta en Git?

Es una marca que se usa para identificar un punto específico en la historia de un repositorio, normalmente para señalar versiones importantes o hitos.

¿Cómo crear una etiqueta en Git?

Se usa: **git tag nombre-de-la-etiqueta**

Para agregar un mensaje: **git tag -a nombre-de-la-etiqueta -m "Descripción"**

¿Cómo enviar una etiqueta a GitHub?

Primero, asegurarse de haber creado la etiqueta localmente. Luego:

git push origin nombre-de-la-etiqueta

¿Qué es un historial de Git?

Es el registro de todos los commits que se hicieron en el repositorio, incluyendo el autor, la fecha y el mensaje de cada commit. Permite revisar el progreso y volver a versiones anteriores si es necesario.

¿Cómo ver el historial de Git?

Se ejecuta: **git log**

¿Cómo buscar en el historial de Git?

Ejecutando: **git log --grep="palabra-clave"**

Esto busca commits que tengan esa palabra en el mensaje.

¿Cómo borrar el historial de Git?

Ejecutando los siguientes pasos:

1. git checkout --orphan nueva-rama
2. git add .
3. git commit -m "Nuevo inicio"
4. git branch -D main
5. git branch -m main
6. git push -f origin main

¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo puede ser visto o modificado por vos y las personas a las que invites. Es útil cuando querés mantener tu código o proyecto oculto al público.

¿Cómo crear un repositorio privado en GitHub?

1. Iniciar sesión en GitHub.
2. Hacer clic en "New" o "Create repository".
3. Ingresar el nombre y descripción.
4. Elegir la opción "Private".
5. Hacer clic en "Create repository".

¿Cómo invitar a alguien a un repositorio privado en GitHub?

1. Ir a tu repositorio privado.
2. Hacer clic en "Settings".
3. Seleccionar "Collaborators" o "Manage access".
4. Hacer clic en "Invite a collaborator".
5. Escribir el nombre de usuario o correo de la persona y enviar la invitación.

¿Qué es un repositorio público en GitHub?

Es un repositorio visible para cualquier persona. Cualquiera puede ver el contenido, pero solo quienes tengan permisos pueden hacer cambios. Es útil para compartir proyectos de código abierto o tareas públicas.

¿Cómo crear un repositorio público en GitHub?

1. Iniciar sesión en GitHub.
2. Hacer clic en "New" o "Create repository".
3. Ingresar el nombre y descripción.
4. Elegir la opción "Public".
5. Hacer clic en "Create repository".

¿Cómo compartir un repositorio público en GitHub?

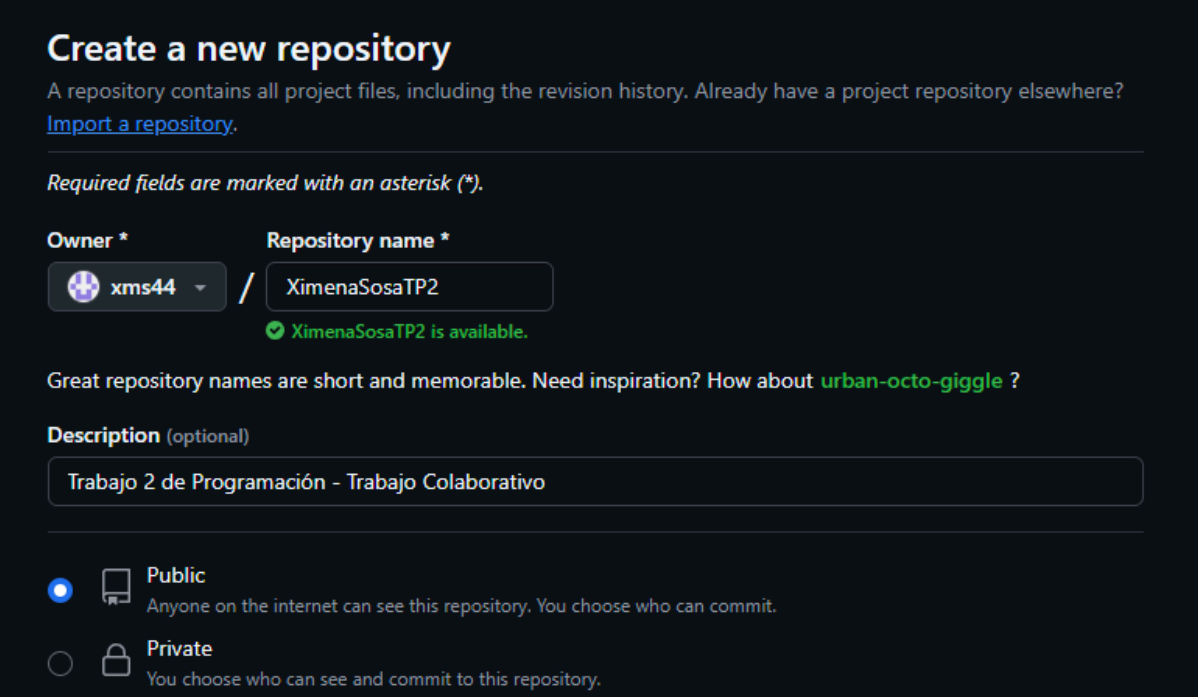
Simplemente copiando y compartiendo la URL del repositorio, por ejemplo:

<https://github.com/xms44/UTN-TUPaD-P1>

ACTIVIDAD 2

Crear un repositorio


El primer paso fue seleccionar “New”, luego agregar un nombre, una descripción, tildar la opción add a README file y finalmente “Create repository”




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 *  xms44 / **Repository name ***

 XimenaSosaTP2 is available.

Great repository names are short and memorable. Need inspiration? How about [urban-octo-giggle](#) ?

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** ▾

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](#).

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

Create repository

Repositorio creado:

The screenshot shows the GitHub interface for a newly created repository named 'XimenaSosaTP2' by user 'xms44'. The repository is public and has 1 branch (main) and 0 tags. The 'About' section describes it as 'Trabajo 2 de Programación - Trabajo Colaborativo'. The 'Releases' section shows no releases published. The 'README' file is visible, containing the title 'XimenaSosaTP2' and the subtitle 'Trabajo 2 de Programación - Trabajo Colaborativo'.

Agregando un Archivo

Iniciamos git bash y luego ejecutamos git clone junto con la url del repositorio creado:

```
NoxiePC@Xime MINGW64 ~
$ git clone https://github.com/xms44/XimenaSosaTP2
Cloning into 'XimenaSosaTP2'...
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.

NoxiePC@Xime MINGW64 ~
$ cd XimenaSosaTP2/

NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$
```


Agregando un Archivo

Primero utilizo el comando **echo** para escribir un texto dentro del archivo `pruebagit.txt`. Luego ejecuto **git add** para agregar el archivo al área de preparación, **git commit** con un mensaje descriptivo para registrar el cambio, y finalmente **git push** para subirlo al repositorio en GitHub:

```
MINGW64:/c:/Users/NoxiePC/XimenaSosaTP2
$ cd XimenaSosaTP2/

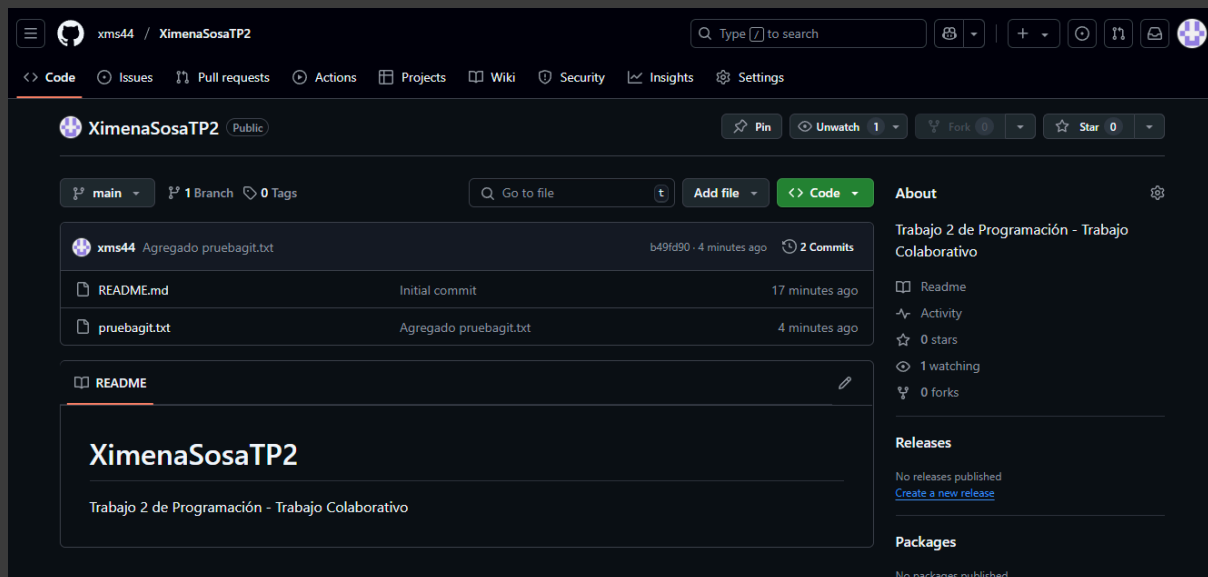
NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$ echo "Este archivo es de práctica para el tp2 de Git" > pruebagit.txt

NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$ git add pruebagit.txt
warning: in the working copy of 'pruebagit.txt', LF will be replaced by CRLF the
next time Git touches it

NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$ git commit -m "Agregado pruebagit.txt"
[main b49fd90] Agregado pruebagit.txt
1 file changed, 1 insertion(+)
 create mode 100644 pruebagit.txt

NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 333 bytes | 333.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xms44/XimenaSosaTP2
 odb3dc9..b49fd90  main -> main

NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$
```



Creando Branchs

Creo una nueva rama llamada **nueva-rama** con el comando `git checkout -b`. Luego agrego una línea nueva al archivo `pruebagit.txt`, la preparo con `git add`, confirmo los cambios con `git commit` y finalmente subo la rama al repositorio remoto usando `git push origin nueva-rama`:


```
NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (main)
$ git checkout -b nueva-rama
Switched to a new branch 'nueva-rama'


NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (nueva-rama)
$ echo "Línea agregada desde nueva-rama." >> pruebagit.txt


NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (nueva-rama)
$ git add pruebagit.txt
warning: in the working copy of 'pruebagit.txt', LF will be replaced by CRLF the next time
Git touches it


NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (nueva-rama)
$ git commit -m "Cambio de rama + línea nueva"
[nueva-rama e5a2179] Cambio de rama + línea nueva
1 file changed, 1 insertion(+)


NoxiePC@Xime MINGW64 ~/XimenaSosaTP2 (nueva-rama)
$ git push origin nueva-rama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 375 bytes | 375.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nueva-rama' on GitHub by visiting:
remote:   https://github.com/xms44/XimenaSosaTP2/pull/new/nueva-rama
remote:
To https://github.com/xms44/XimenaSosaTP2
 * [new branch]   nueva-rama -> nueva-rama
```


 **XimenaSosaTP2** Public Pin Unwatch 1

 **nueva-rama** had recent pushes 14 seconds ago Compare & pull request

 main


 2 Branches


 0 Tags

Go to file 

Add file


Code

 **xms44** Agregado pruebagit.txt b49fd90 · 30 minutes ago 2 Commits

 README.md


Initial commit

43 minutes ago

 pruebagit.txt

Agregado pruebagit.txt

30 minutes ago

README 

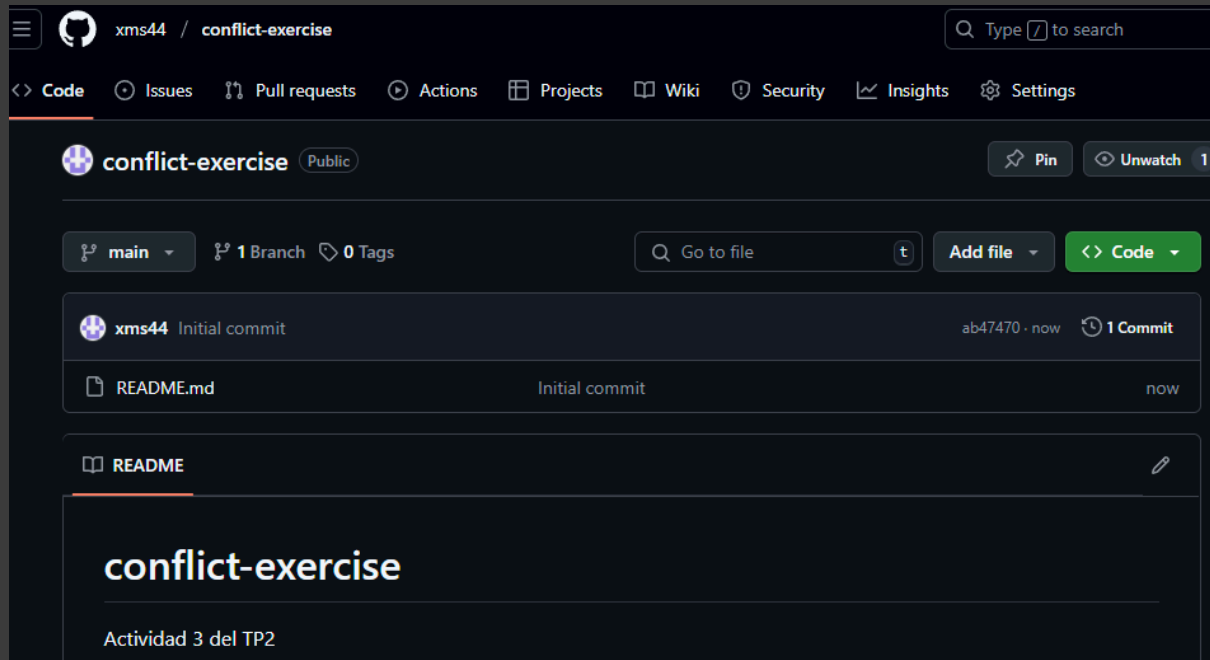
XimenaSosaTP2

Trabajo 2 de Programación - Trabajo Colaborativo

ACTIVIDAD 3

Paso 1: Crear un repositorio en GitHub

Repositorio creado:



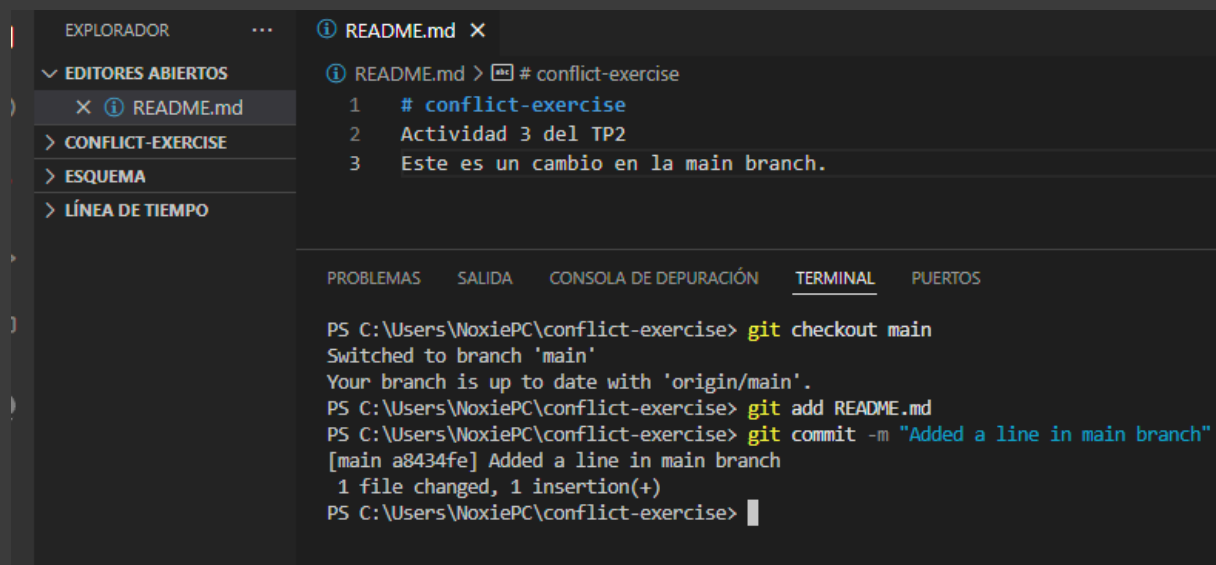
Paso 3: Crear una nueva rama y editar un archivo

```
1 # conflict-exercise
2
3 Actividad 3 del TP2
4 Este es un cambio en la feature branch.
5
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

```
PS C:\Users\NoxiePC\XimenaSosaTP2> cd ..
PS C:\Users\NoxiePC> git clone https://github.com/xms44/conflict-exercise
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS C:\Users\NoxiePC> cd .\conflict-exercise\
PS C:\Users\NoxiePC\conflict-exercise> git checkout -b feature-branch
Switched to a new branch 'feature-branch'
PS C:\Users\NoxiePC\conflict-exercise> git add README.md
PS C:\Users\NoxiePC\conflict-exercise> git commit -m "Added a line in feature-branch"
[feature-branch 58bd527] Added a line in feature-branch
1 file changed, 2 insertions(+)
```

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



EXPLORADOR ...

EDITORES ABIERTOS

- × README.md

CONFLICT-EXERCISE

ESQUEMA

LÍNEA DE TIEMPO

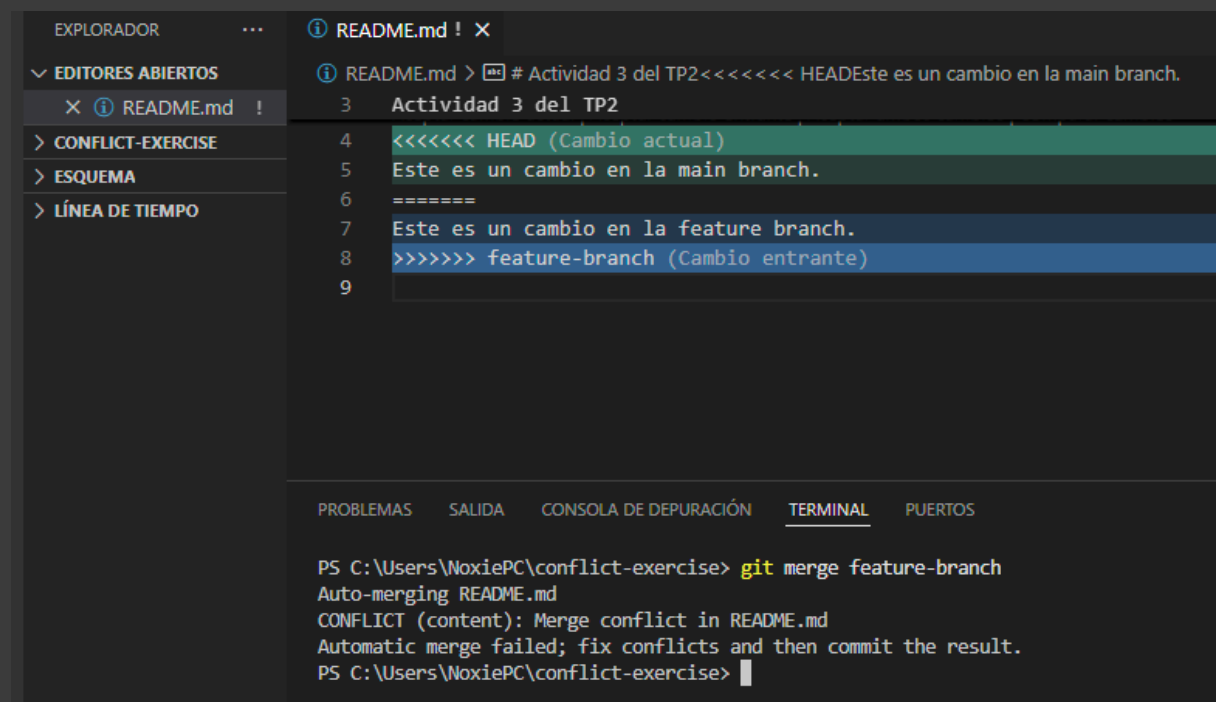
README.md

```
1 # conflict-exercise
2 Actividad 3 del TP2
3 Este es un cambio en la main branch.
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL** PUERTOS

```
PS C:\Users\WoxiePC\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\WoxiePC\conflict-exercise> git add README.md
PS C:\Users\WoxiePC\conflict-exercise> git commit -m "Added a line in main branch"
[main a8434fe] Added a line in main branch
1 file changed, 1 insertion(+)
PS C:\Users\WoxiePC\conflict-exercise>
```

Paso 5: Hacer un merge y generar un conflicto



EXPLORADOR ...

EDITORES ABIERTOS

- × README.md !

CONFLICT-EXERCISE

ESQUEMA

LÍNEA DE TIEMPO

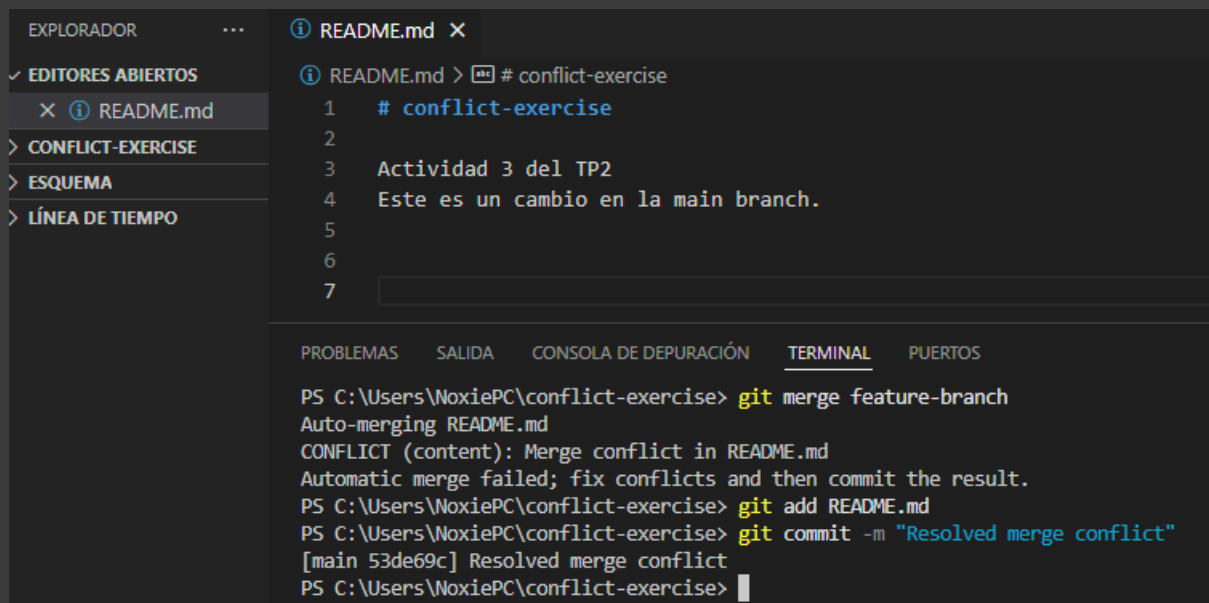
README.md !

```
3 Actividad 3 del TP2
4 <<<<<< HEAD (Cambio actual)
5 Este es un cambio en la main branch.
6 =====
7 Este es un cambio en la feature branch.
8 >>>>>> feature-branch (Cambio entrante)
9
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL** PUERTOS

```
PS C:\Users\WoxiePC\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\WoxiePC\conflict-exercise>
```

Paso 6: Resolver el conflicto



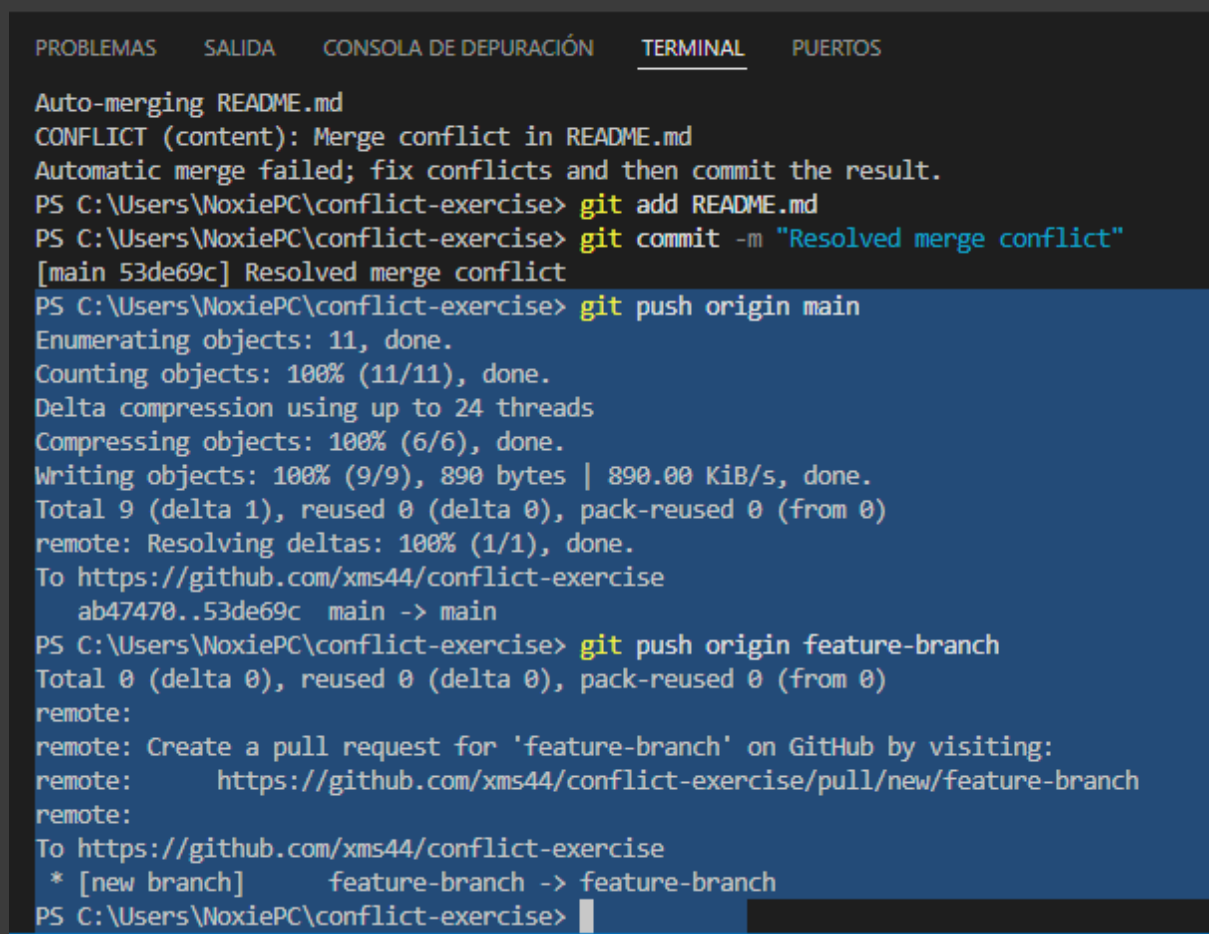
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the project structure with 'CONFLICT-EXERCISE' selected. The Editor pane shows 'README.md' with a conflict. The file content is as follows:

```
1 # conflict-exercise
2
3 Actividad 3 del TP2
4 Este es un cambio en la main branch.
5
6
7
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
PS C:\Users\NoxiePC\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\NoxiePC\conflict-exercise> git add README.md
PS C:\Users\NoxiePC\conflict-exercise> git commit -m "Resolved merge conflict"
[main 53de69c] Resolved merge conflict
PS C:\Users\NoxiePC\conflict-exercise>
```

Paso 7: Subir los cambios a GitHub



The screenshot shows the 'TERMINAL' tab in VS Code with the following output:

```
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\NoxiePC\conflict-exercise> git add README.md
PS C:\Users\NoxiePC\conflict-exercise> git commit -m "Resolved merge conflict"
[main 53de69c] Resolved merge conflict
PS C:\Users\NoxiePC\conflict-exercise> 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), 890 bytes | 890.00 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/xms44/conflict-exercise
   ab47470..53de69c  main -> main
PS C:\Users\NoxiePC\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/xms44/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/xms44/conflict-exercise
 * [new branch]   feature-branch -> feature-branch
PS C:\Users\NoxiePC\conflict-exercise>
```

Paso 8: Verificar en GitHub

This screenshot shows the GitHub interface for the 'conflict-exercise' repository. The 'main' branch is selected, showing a commit by 'xms44' titled 'Resolved merge conflict' from 6 minutes ago. The README file is highlighted, displaying the title 'conflict-exercise' and the text 'Actividad 3 del TP2 Este es un cambio en la main branch.'

This screenshot shows the same GitHub repository but on the 'feature-branch'. A message indicates 'This branch is 2 commits behind main'. A new commit by 'xms44' is shown, titled 'Added a line in feature-branch' from 33 minutes ago. The README file is updated to show 'Actividad 3 del TP2 Este es un cambio en la feature branch.'