

## Práctico 2: Git y GitHub

### Actividades

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

#### 1. ¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

Presentar o compartir el trabajo.

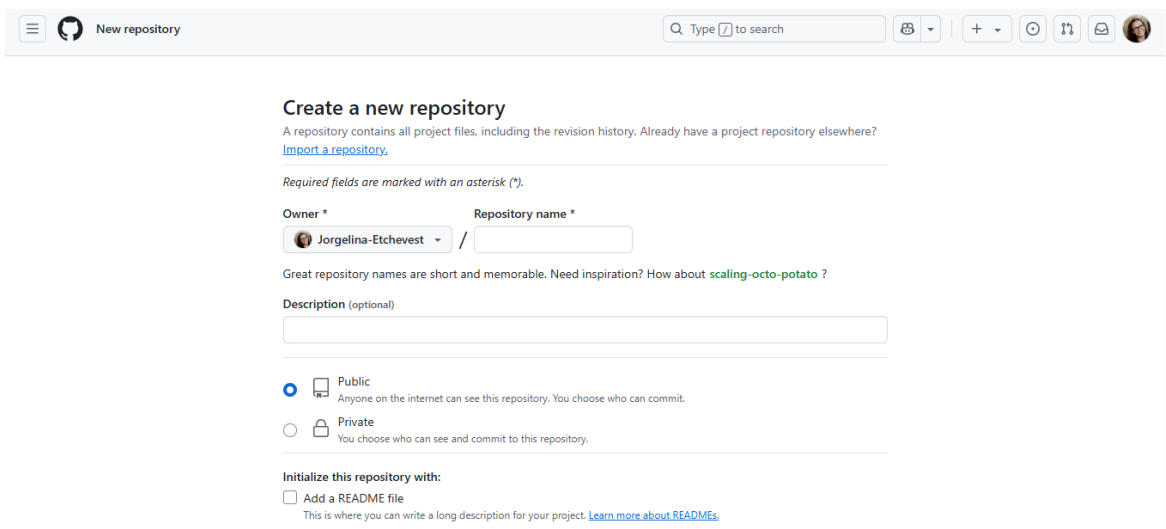
Seguir y administrar los cambios en el código a lo largo del tiempo.

Dejar que otros usuarios revisen el código y realicen sugerencias para mejorarlo.

Colaborar en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

#### 2. ¿Cómo crear un repositorio en GitHub?

Una vez creada y abierta la cuenta de GitHub, clickeando en “NEW” nos permite crear un repositorio.



New repository

Q Type [Z] to search

### 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 \*

Jorgelina-Etchevest /

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

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.](#)

Se debe colocar nombre, opcionalmente alguna descripción, además de elegir si será un repositorio público o privado.

### **3. ¿Cómo crear una rama en Git?**

En Git, una rama es una versión del código de un proyecto que permite trabajar en él sin afectar a la rama principal.

El comando `$ git init` crea por defecto la rama “master”.

Para crear una rama nueva en Git, se utiliza el comando

`$ git branch NombreRama`

### **4. ¿Cómo cambiar a una rama en Git?**

En Git, una rama es una línea de desarrollo independiente de un proyecto.

Para cambiar de rama, se utiliza el comando

`$ git checkout NuevaRama`

### **5. ¿Cómo fusionar ramas en Git?**

Primero, uno debe estar en la rama a la que se le quieren fusionar los cambios.

Haciendo

`$ git checkout master`

Nos paramos en la rama principal

Y haciendo el comando

`$ git merge NuevaRama`

Se fusiona NuevaRama en master.

### **6. ¿Cómo crear un commit en Git?**

Un commit en Git es una confirmación de cambios en un proyecto, que se guarda en el repositorio local. Antes de hacer un commit, se debe usar el comando

`$ git add .`

para seleccionar los cambios que se quieran incluir.

Para hacer un commit en GitHub, puedes usar el comando

`$ git commit -m “Mensaje que acompaña al commit”`

Incluir un mensaje con la opción -m. El mensaje debe describir los cambios que se están realizando.

### 7. ¿Cómo enviar un commit a GitHub?

Subir los cambios a GitHub con el comando

```
$ git push origin NombreRama
```

### 8. ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de un proyecto que se almacena en un servidor remoto, ya sea en internet o en una red. Se comparte entre varios miembros de un equipo.

### 9. ¿Cómo agregar un repositorio remoto a Git?

Se utiliza el siguiente comando:

```
$ git remote add origin URL
```

### 10. ¿Cómo empujar los cambios de un repositorio remoto?

```
$ git push origin master
```

### 11. ¿Cómo tirar los cambios de un repositorio remoto?

Para traer los cambios más recientes del repositorio remoto

```
$ git pull origin master
```

Esto descarga y fusiona los cambios desde la rama master del repositorio remoto origin.

### 12. ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio que se guarda en tu cuenta de GitHub. Te permite experimentar y modificar sin afectar el original. Es muy común en proyectos de código abierto.

### 13. ¿Cómo crear un fork de un repositorio?

Pasos para hacer un fork:

Ingresa al repositorio original en GitHub. Hacer clic en el botón "Fork" en la esquina superior derecha y GitHub va a crear una copia del repositorio en nuestra cuenta.

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

Un pull request en GitHub es una solicitud de revisión y aprobación de cambios en un repositorio. Es una forma de colaborar en proyectos de código abierto o grupales.

Al realizar cambios en un fork, en la solapa Pull Request, clickeamos en NEW PULL REQUEST, CREATE NEW PULL REQUEST.

#### 15. ¿Cómo aceptar una solicitud de extracción?

Quien haya desarrollado el repositorio al cual le realizamos un pull request, verá los mensajes que le enviaron y revisará los cambios. Si todo está bien, hará clic en "Merge pull request". Luego en "Confirm merge" para aceptar.

#### 16. ¿Qué es una etiqueta en Git?

Una etiqueta o tag es una marca que se le pone a un commit para señalar un momento importante, como una versión (v1.0, v2.3.1, etc.). No cambia con el tiempo y se usa mucho para versiones de software.

#### 17. ¿Cómo crear una etiqueta en Git?

Desde la terminal se escribe el siguiente comando, para generar una etiqueta con la versión v1.0

```
$ git tag v1.0
```

#### 18. ¿Cómo enviar una etiqueta a GitHub?

Una vez creada la etiqueta en el repositorio local debemos empujarla al repositorio remoto (GitHub), con el siguiente comando:

```
$ push origin v1.0
```

#### 19. ¿Qué es un historial de Git?

El historial de Git es un registro de los cambios realizados en un proyecto, es decir, es la lista de todos los commits hechos en un repositorio: qué se cambió, quién lo hizo, cuándo, y con qué mensaje.

#### 20. ¿Cómo ver el historial de Git?

Para ver el historial de Git es necesario utilizar el comando:

```
$ git log
```

#### 21. ¿Cómo buscar en el historial de Git?

Existen varias posibilidades para localizar los commits:

- a. Commits que tengan una palabra o frase específica:

```
git log --grep="palabra_clave"
```

- b. Commits realizados por autor

```
git log --author="nombre"
```

c. Commits por nombre del archivo

```
git log nombre_archivo
```

d. Commits realizados entre 2 fechas específicas

```
git log --since=<date-from> --until=<date-to>
```

## 22. ¿Cómo borrar el historial de Git?

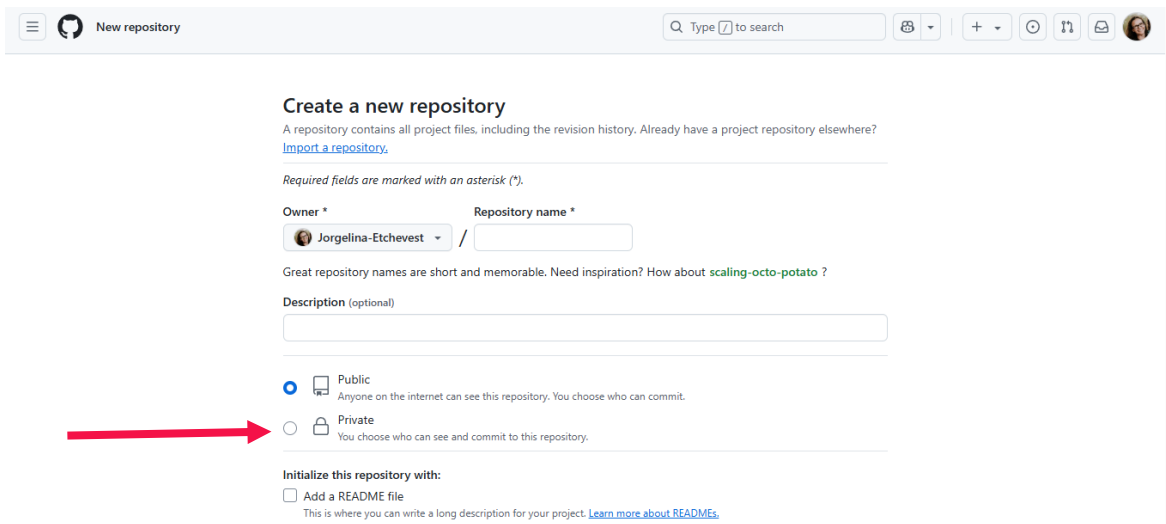
Con el comando

```
$ git reset
```

## 23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo vos o quienes han sido autorizados pueden ver o editar. No está disponible públicamente.

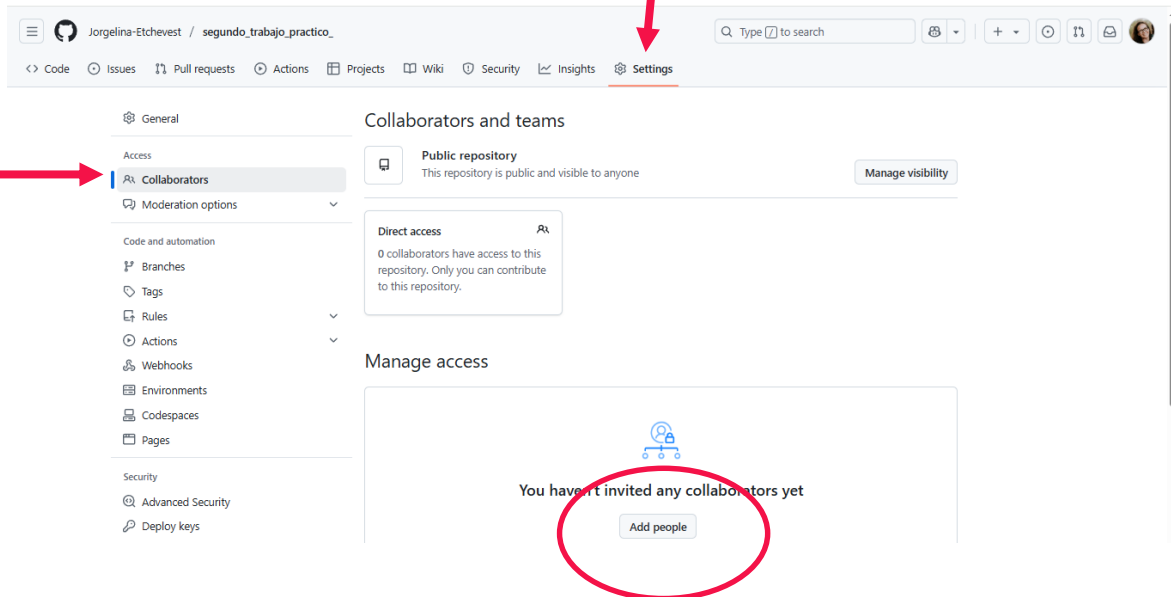
## 24. ¿Cómo crear un repositorio privado en GitHub?



The screenshot shows the GitHub 'Create a new repository' interface. At the top, there's a search bar and navigation icons. The main heading is 'Create a new repository'. Below it, a note says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A line of text states 'Required fields are marked with an asterisk (\*)'. There are two required fields: 'Owner \*' (a dropdown menu showing 'Jorgelina-Echevest') and 'Repository name \*' (an empty text box). Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about [scaling-octo-potato](#) ?'. There is an optional 'Description' text box. Under the 'Visibility' section, there are two radio buttons: 'Public' (selected) and 'Private'. A red arrow points to the 'Private' radio button. Below this, there is an 'Initialize this repository with:' section with a checkbox for 'Add a README file' and a link to 'Learn more about READMEs'.

En la configuración de la privacidad se marca repositorio Private.

## 25. ¿Cómo invitar a alguien a un repositorio privado en GitHub?



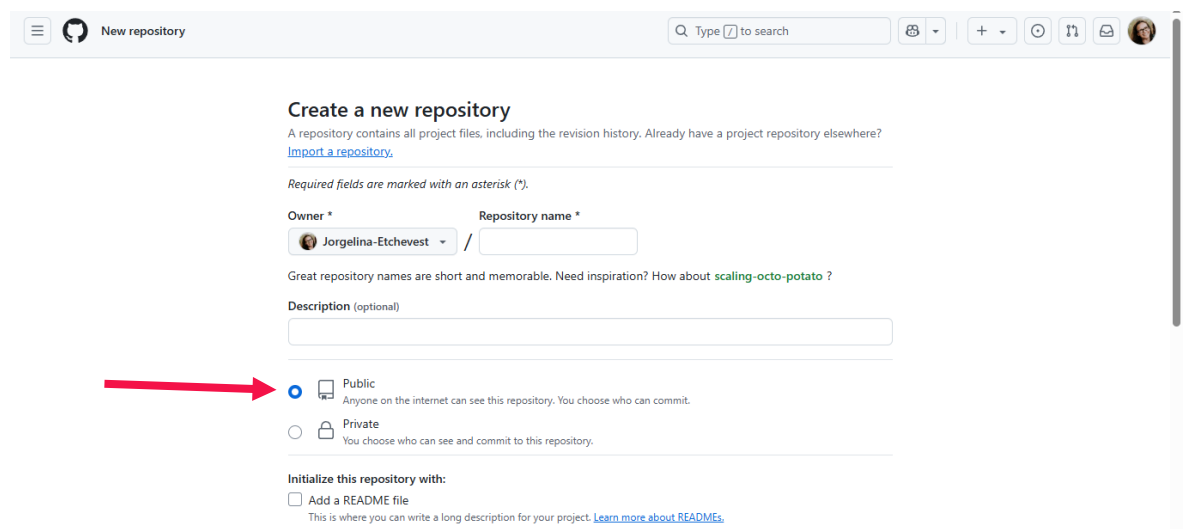
Una vez en el repositorio en el que queremos agregar colaboradores, vamos a:  
Settings  
Collaborators  
Add people

Se agrega el nombre de usuario o email de la persona y se envía la invitación.

## 26. ¿Qué es un repositorio público en GitHub?

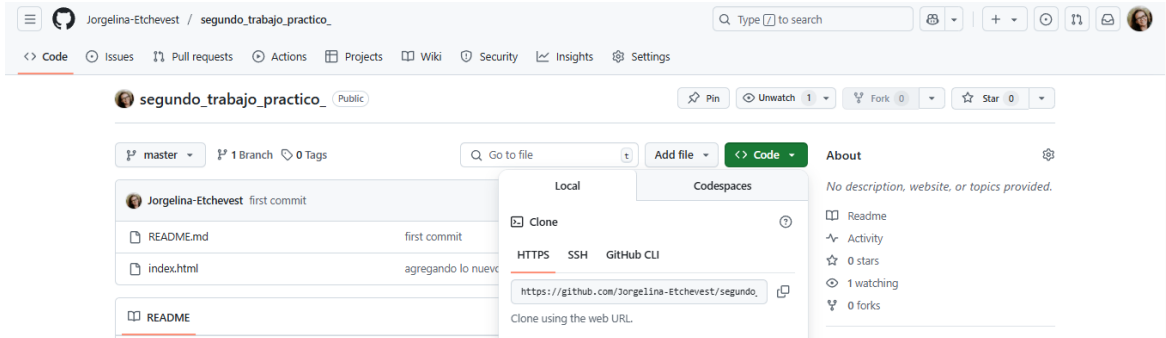
Un repositorio visible para todo el mundo. Cualquiera puede verlo, clonarlo o hacer forks (pero no modificarlo sin permiso).

## 27. ¿Cómo crear un repositorio público en GitHub?



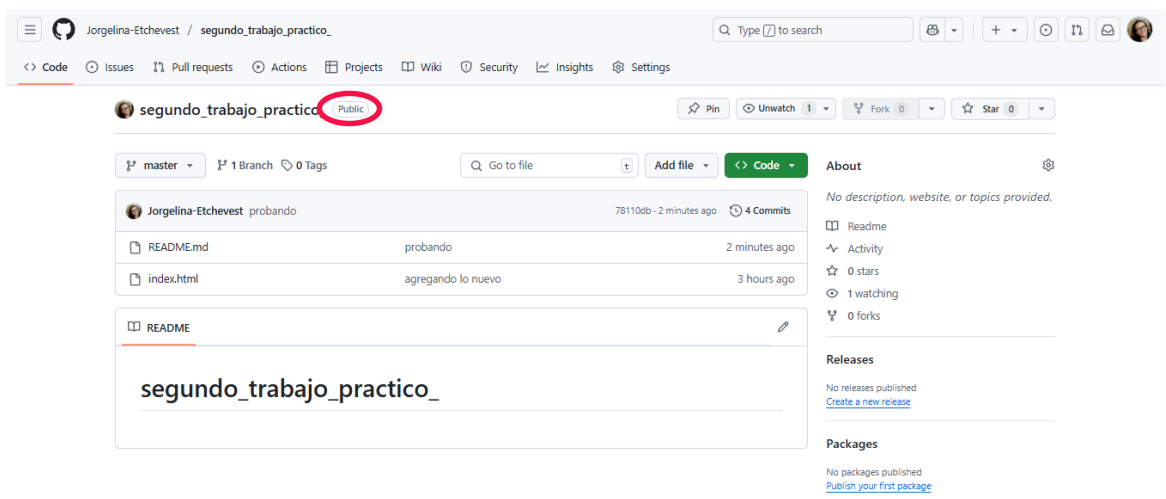
## 28. ¿Cómo compartir un repositorio público en GitHub?

Entrando a la pestaña < > Code y compartiendo la URL.

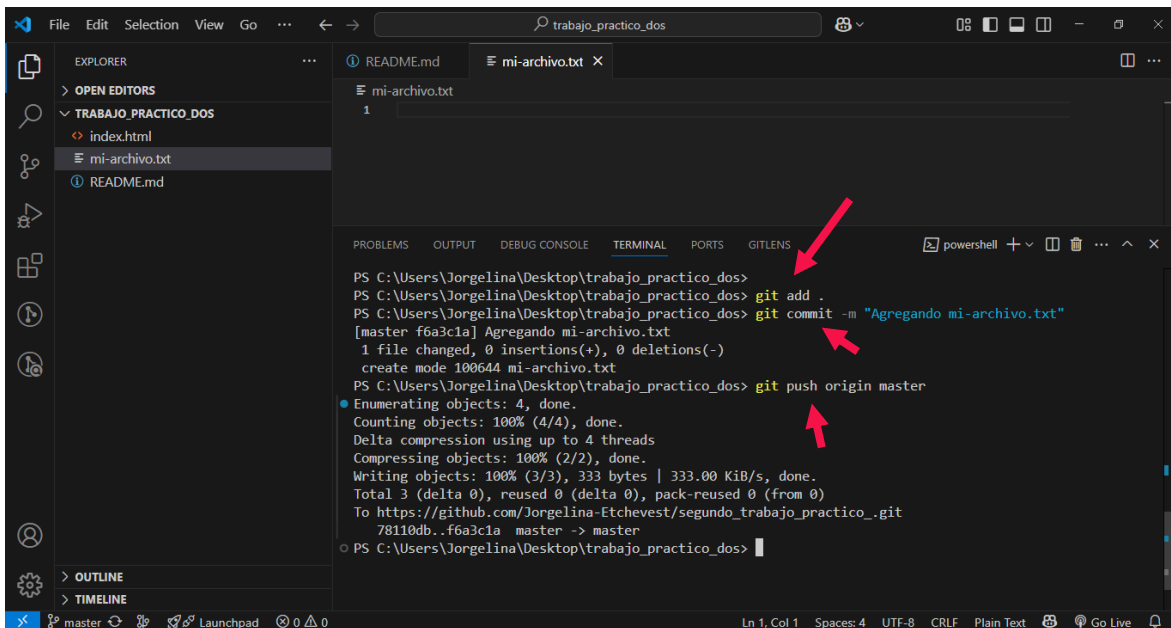


## 2. Realizar la siguiente actividad:

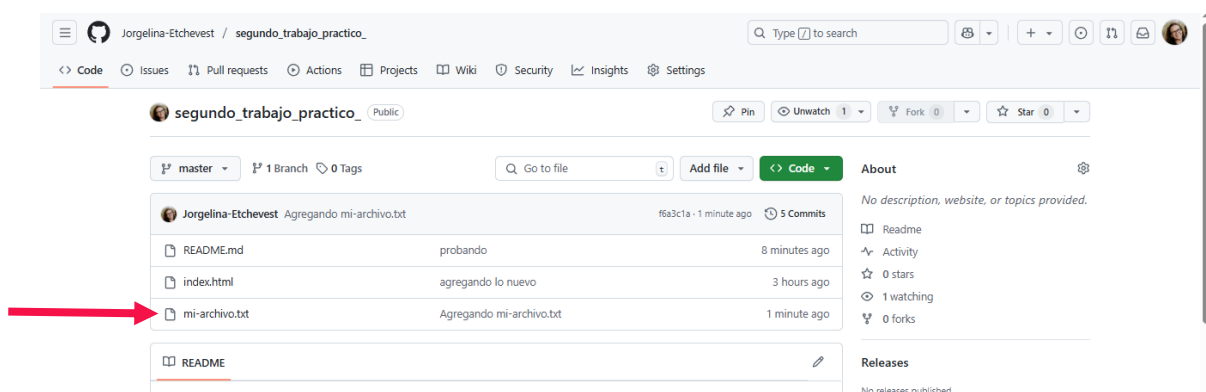
- Crear un repositorio.
  - ✓ Dale un nombre al repositorio. **segundo\_trabajo\_practico**
  - ✓ Elije el repositorio sea público.
  - ✓ Inicializa el repositorio con un archivo.



- 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).

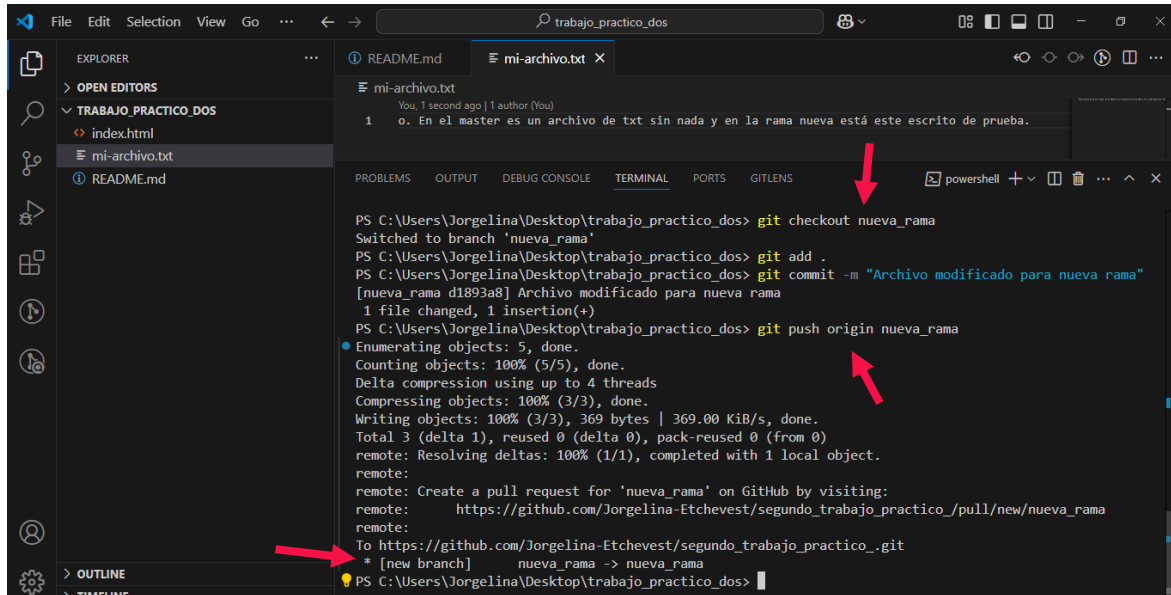


```
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos>
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git commit -m "Agregando mi-archivo.txt"
[master f6a3c1a] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 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/Jorgelina-Etchevest/segundo_trabajo_practico_git
78110db..f6a3c1a master -> master
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos>
```

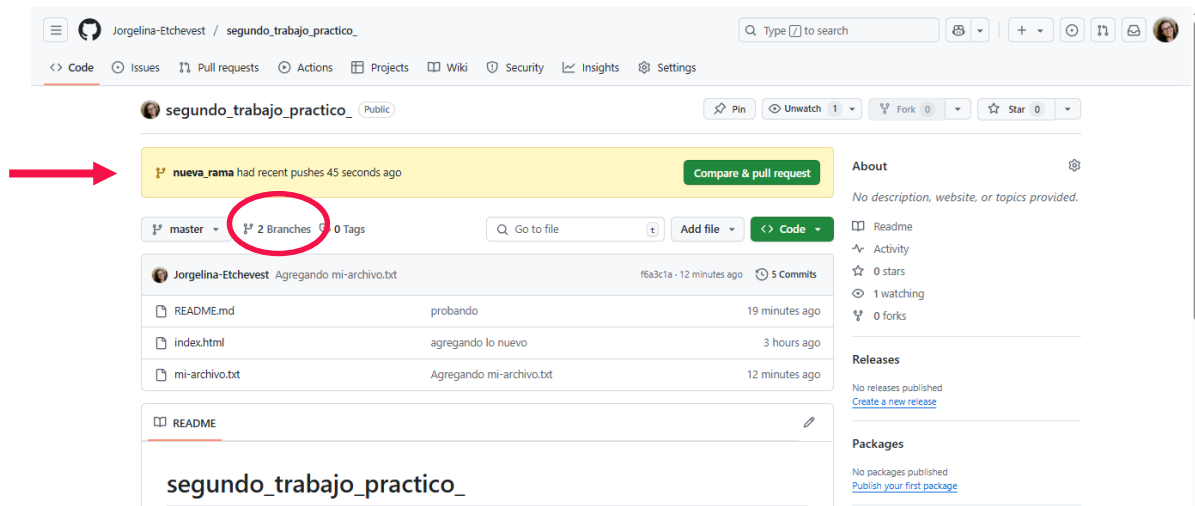




- Creando Branchs
  - ✓ Crear una Branch
  - ✓ Realizar cambios o agregar un archivo
  - ✓ Subir la Branch



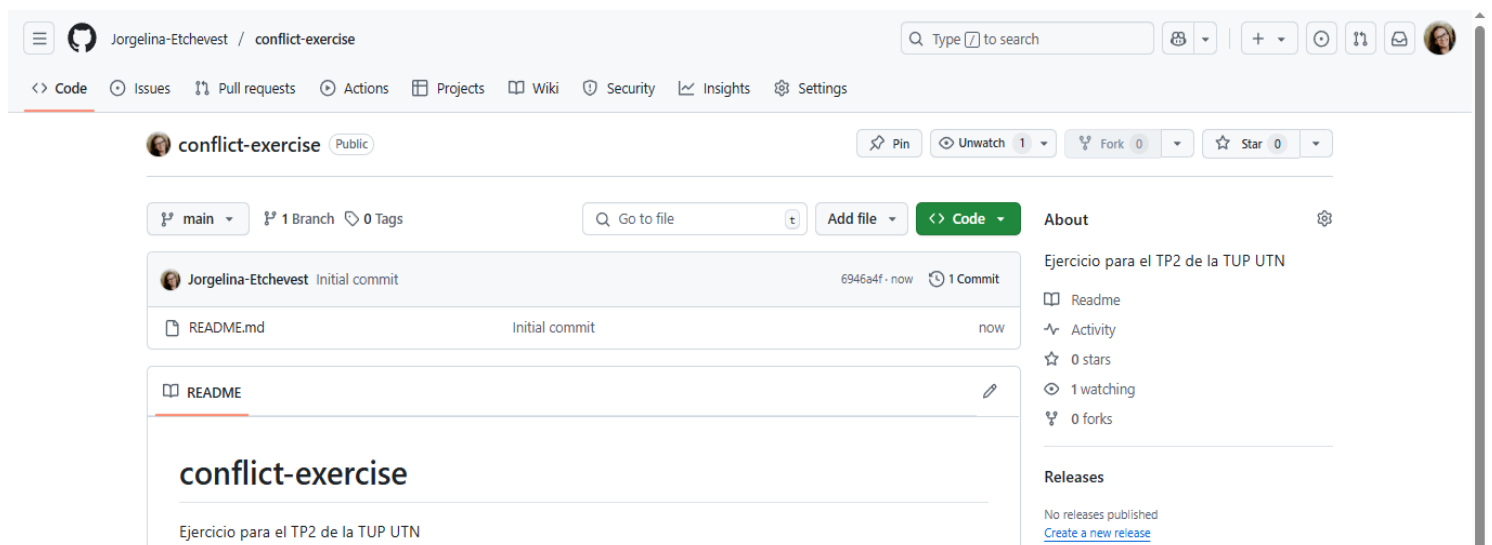
```
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git checkout nueva_rama
Switched to branch 'nueva_rama'
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git commit -m "Archivo modificado para nueva rama"
[nueva_rama d1893a8] Archivo modificado para nueva rama
1 file changed, 1 insertion(+)
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git push origin nueva_rama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 369 bytes | 369.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: Create a pull request for 'nueva_rama' on GitHub by visiting:
remote:   https://github.com/Jorgelina-Etchevest/segundo_trabajo_practico_/pull/new/nueva_rama
remote:
To https://github.com/Jorgelina-Etchevest/segundo_trabajo_practico_.git
 * [new branch]   nueva_rama -> nueva_rama
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos>
```



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
```

```
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> git clone https://github.com/Jorgelina-Etchevest/conflict-exercise.git
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.
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> cd conflict-exercise
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise>
```

### Paso 3: Crear una nueva rama y editar un archivo

- ✓ Crea una nueva rama llamada feature-branch:

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

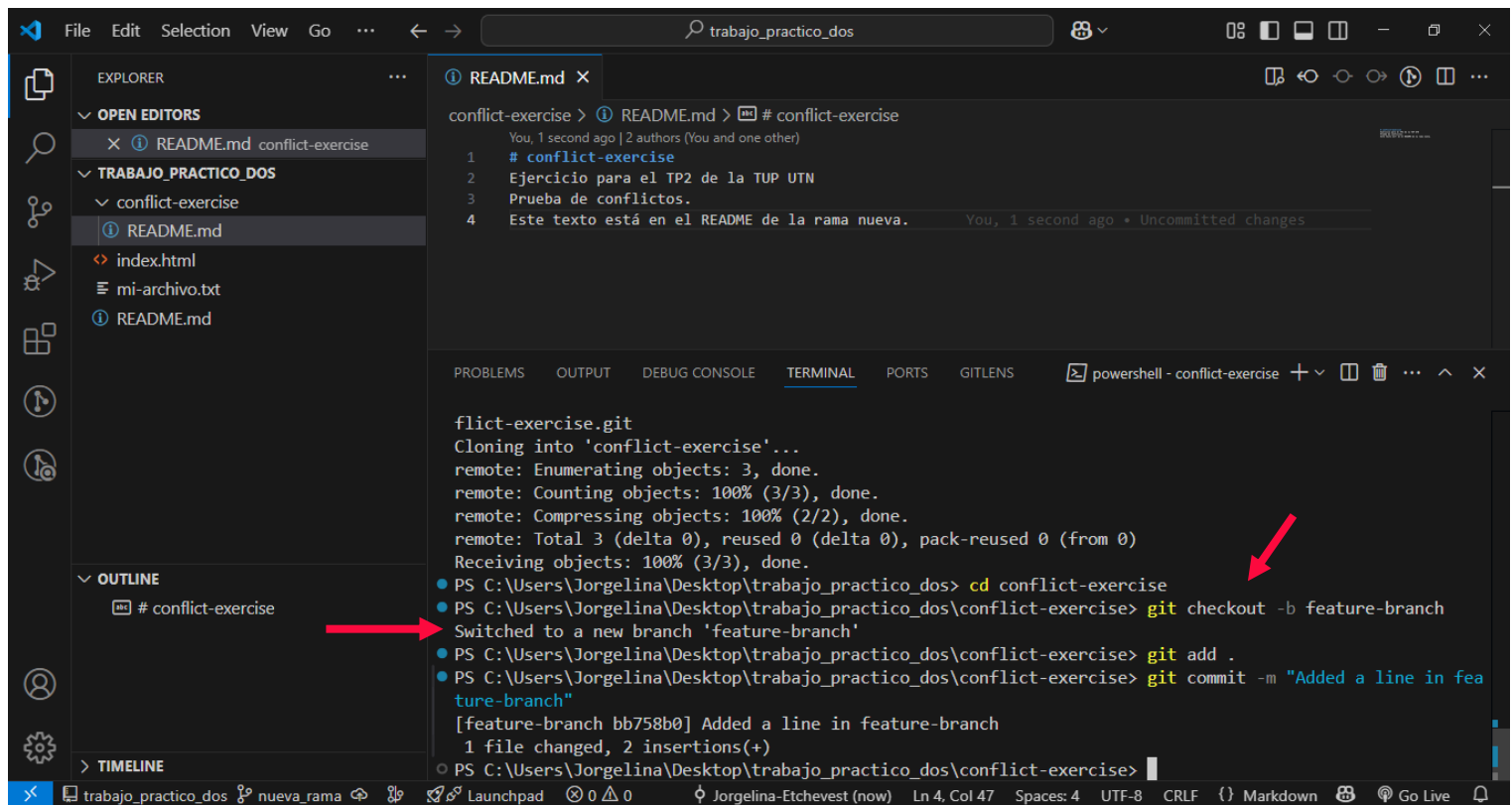
- ✓ Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:

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"
```



The screenshot shows the Visual Studio Code interface with the Explorer, Search, and Run and Debug views on the left. The Explorer view shows the file structure of the 'trabajo\_practico\_dos' project, including the 'conflict-exercise' folder and the 'README.md' file. The Search view shows the contents of 'README.md', which includes the text 'Este texto está en el README de la rama nueva.' The Run and Debug view shows the terminal output of the commands: `git checkout -b feature-branch`, `git add .`, and `git commit -m "Added a line in feature-branch"`. A red arrow points to the 'feature-branch' in the terminal output, and another red arrow points to the 'README.md' file in the Explorer view.

```
File Edit Selection View Go ... < -> trabajo_practico_dos
```

EXPLORER

- OPEN EDITORS
  - README.md conflict-exercise
- TRABAJO\_PRACTICO\_DOS
  - conflict-exercise
    - README.md
    - index.html
    - mi-archivo.txt
    - README.md
- OUTLINE
  - # conflict-exercise
- TIMELINE

README.md

```
conflict-exercise > README.md > # conflict-exercise
You, 1 second ago | 2 authors (You and one other)
1 # conflict-exercise
2 Ejercicio para el TP2 de la TUP UTN
3 Prueba de conflictos.
4 Este texto está en el README de la rama nueva. You, 1 second ago * Uncommitted changes
```

TERMINAL

```
powershell - conflict-exercise
flict-exercise.git
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.
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos> cd conflict-exercise
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git checkout -b feature-branch
Switched to a new branch 'feature-branch'
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git commit -m "Added a line in fea
ture-branch"
[feature-branch bb758b0] Added a line in feature-branch
1 file changed, 2 insertions(+)
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise>
```

#### 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"

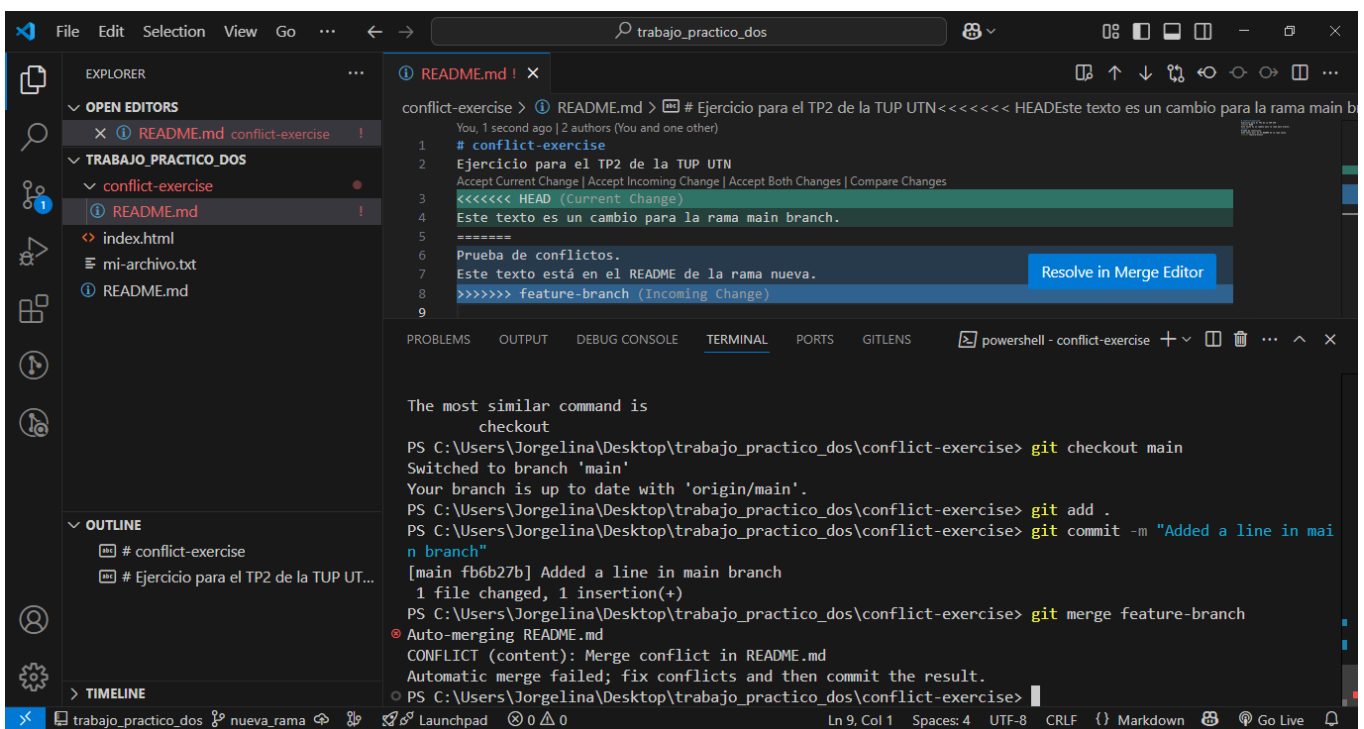
```
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git commit -m "Added a line in main branch"
[main fb6b27b] Added a line in main branch
1 file changed, 1 insertion(+)
```

#### 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.



```
File Edit Selection View Go ... trabajo_practico_dos
EXPLORER
OPEN EDITORS
X README.md conflict-exercise
TRABAJO_PRACTICO_DOS
conflict-exercise
README.md
index.html
mi-archivo.txt
README.md
OUTLINE
# conflict-exercise
# Ejercicio para el TP2 de la TUP UTN...
TIMELINE
trabajo_practico_dos nueva Rama Launchpad 0 0

conflict-exercise > README.md > # Ejercicio para el TP2 de la TUP UTN<<<<<< HEADEste texto es un cambio para la rama main b
You, 1 second ago | 2 authors (You and one other)
1 # conflict-exercise
2 Ejercicio para el TP2 de la TUP UTN
3 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<<<< HEAD (Current Change)
5 Este texto es un cambio para la rama main branch.
6 =====
7 Prueba de conflictos.
8 Este texto está en el README de la rama nueva.
9 >>>>>> feature-branch (Incoming Change)
Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell - conflict-exercise
The most similar command is
checkout
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git commit -m "Added a line in mai
n branch"
[main fb6b27b] Added a line in main branch
1 file changed, 1 insertion(+)
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\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\Jorgelina\Desktop\trabajo_practico_dos\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.

```
=====
```

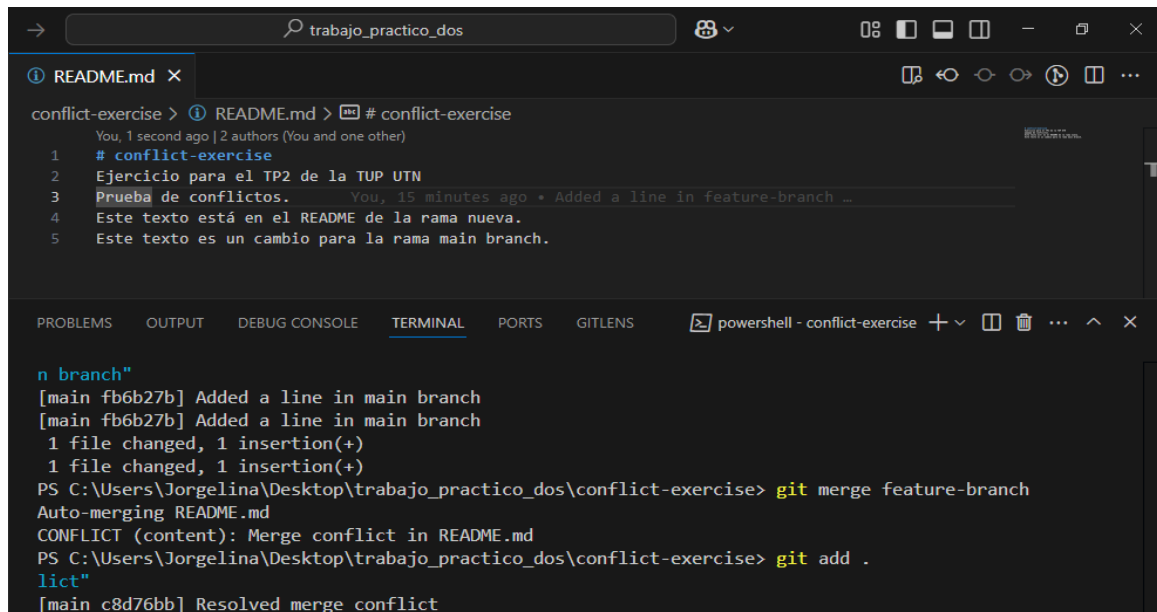
Este es un cambio en la feature branch.

```
>>>>>>> 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"
```



```
conflict-exercise > 1 README.md > # conflict-exercise
You, 1 second ago | 2 authors (You and one other)
1 # conflict-exercise
2 Ejercicio para el TP2 de la TUP UTN
3 Prueba de conflictos. You, 15 minutes ago * Added a line in feature-branch ...
4 Este texto está en el README de la rama nueva.
5 Este texto es un cambio para la rama main branch.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell - conflict-exercise
n branch"
[main fb6b27b] Added a line in main branch
[main fb6b27b] Added a line in main branch
1 file changed, 1 insertion(+)
1 file changed, 1 insertion(+)
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git add .
lict"
[main c8d76bb] Resolved merge conflict
```

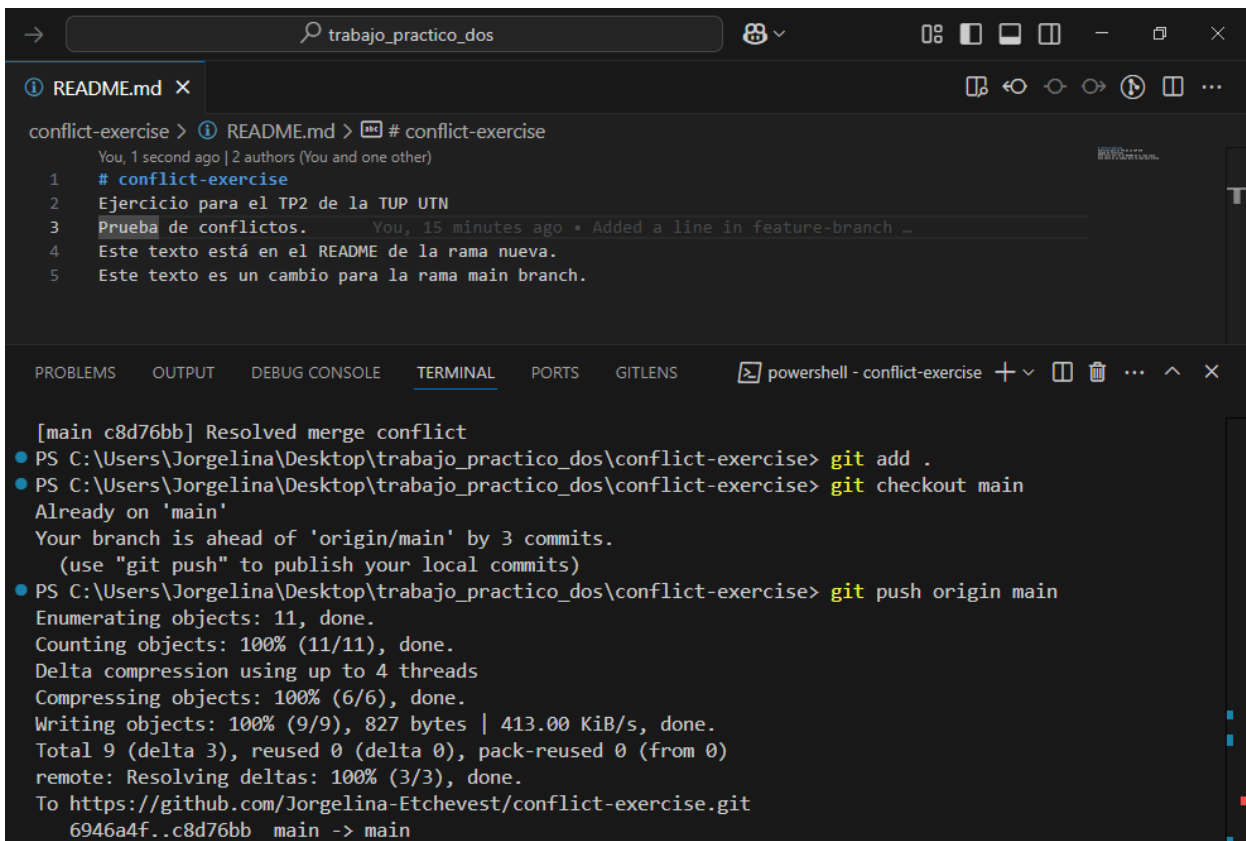
### Paso 7: Subir los cambios a GitHub

- ✓ 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
```



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

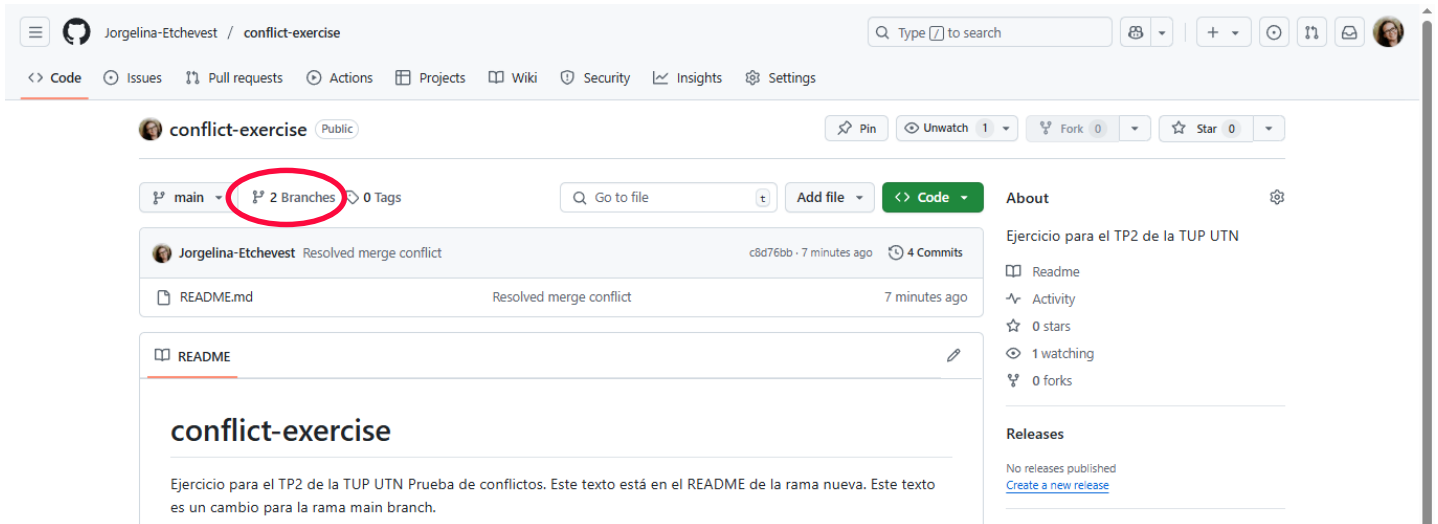
```
conflict-exercise > README.md > # conflict-exercise
You, 1 second ago | 2 authors (You and one other)
1 # conflict-exercise
2 Ejercicio para el TP2 de la TUP UTN
3 Prueba de conflictos.
4 Este texto está en el README de la rama nueva.
5 Este texto es un cambio para la rama main branch.
```

Below the editor, the TERMINAL panel is active, showing the output of several Git commands in a PowerShell session:

```
[main c8d76bb] Resolved merge conflict
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git add .
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)
PS C:\Users\Jorgelina\Desktop\trabajo_practico_dos\conflict-exercise> git push 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), 827 bytes | 413.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Jorgelina-Etchevest/conflict-exercise.git
6946a4f..c8d76bb main -> main
```

## 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.



conflict-exercise (Public)

main 2 Branches 0 Tags

Jorgelina-Etchevest Resolved merge conflict c8d76bb · 7 minutes ago 4 Commits

README.md Resolved merge conflict 7 minutes ago

### conflict-exercise

Ejercicio para el TP2 de la TUP UTN Prueba de conflictos. Este texto está en el README de la rama nueva. Este texto es un cambio para la rama main branch.

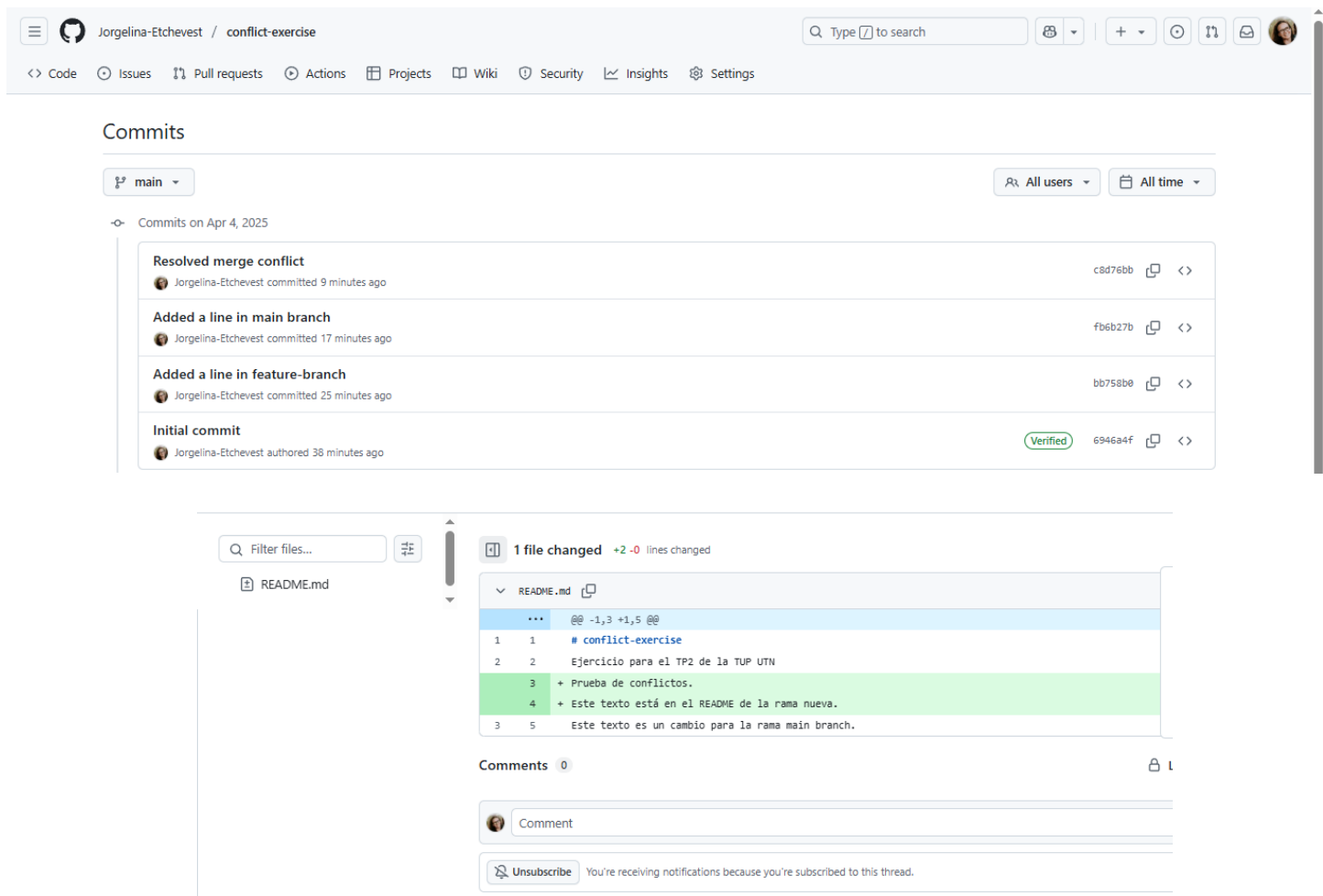
About

Ejercicio para el TP2 de la TUP UTN

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published  
[Create a new release](#)



### Commits

main

Commits on Apr 4, 2025

- Resolved merge conflict**  
Jorgelina-Etchevest committed 9 minutes ago c8d76bb
- Added a line in main branch**  
Jorgelina-Etchevest committed 17 minutes ago fb6b27b
- Added a line in feature-branch**  
Jorgelina-Etchevest committed 25 minutes ago bb758b8
- Initial commit**  
Jorgelina-Etchevest authored 38 minutes ago 6946a4f (Verified)

1 file changed +2 -0 lines changed

README.md

```
@@ -1,3 +1,5 @@
1 1 # conflict-exercise
2 2 Ejercicio para el TP2 de la TUP UTN
3 3 + Prueba de conflictos.
4 4 + Este texto está en el README de la rama nueva.
3 5 Este texto es un cambio para la rama main branch.
```

Comments 0

Comment

Unsubscribe You're receiving notifications because you're subscribed to this thread.