

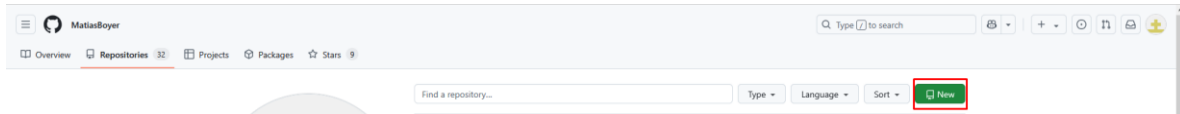
1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas)

- ¿Qué es GitHub?

Es un sitio web (plataforma), donde se pueden alojar repositorios utilizando GIT

- ¿Cómo crear un repositorio en GitHub?

Desde GitHub, podemos navegar a nuestro perfil -> repositorios -> NEW.



Luego, rellenamos los datos correspondientes:

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 * MatiasBoyer / Repository name * tupad_repo_ej2 ← nombre repo
✓ tupad_repo_ej2 is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-memory](#)?

Description (optional)

☒ Public ← repo público
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 ← lo inicializo con un archivo README
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](#).

① You are creating a public repository in your personal account.

[Create repository](#)

Matías Ruben Boyer Rodríguez
DNI 43.901.799
matiasboyer7@gmail.com

- ¿Cómo crear una rama en Git?

Podemos utilizar el comando '**git branch <NOMBRE>**', donde <NOMBRE> es el nombre de la rama nueva a crear.

- ¿Cómo cambiar a una rama en Git?

Podemos utilizar el comando '**git checkout <NOMBRE>**', donde <NOMBRE> es el nombre de la rama a cambiar.

- ¿Cómo fusionar ramas en Git?

Nos ubicamos en la rama donde queremos integrar los cambios utilizando **GIT CHECKOUT <NOMBRE_A>**. Luego, utilizamos '**git merge <NOMBRE_B>**', para integrar los cambios de la branch **NOMBRE_B** a la branch **NOMBRE_A**

- ¿Cómo crear un commit en Git?

Primero necesitamos agregar los archivos utilizando '**git add .**' (o indicando el nombre del archivo en la posición del punto).

Luego, realizamos un commit de esos archivos utilizando '**git commit -m "<MENSAJE>"**', donde <MENSAJE> es un texto corto indicando los cambios realizados.

- ¿Cómo enviar un commit a GitHub?

Una vez el commit está realizado, debemos utilizar el comando '**git push origin <BRANCH>**', donde <BRANCH> es la rama que queremos enviar a GitHub.

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una ubicación web donde podemos subir todos nuestros cambios, para no mantenerlos de forma local.

- ¿Cómo agregar un repositorio remoto a Git?

Utilizando el comando '**git remote add origin <URL>.git**', donde <URL>.git es la url del repositorio (generalmente GitHub) con el archivo .git

- ¿Cómo empujar cambios a un repositorio remoto?

Utilizando el mismo comando para enviar un commit a GitHub, '**git push origin <BRANCH>**'.

- ¿Cómo tirar de cambios de un repositorio remoto?

Podemos utilizar el comando '**git pull origin <BRANCH>**', donde <BRANCH> es la rama de donde queremos pullear.

- ¿Qué es un fork de repositorio?

Es una copia exacta del repositorio, en nuestro usuario, para poder realizar cambios sin afectar al repositorio original.

- ¿Cómo crear un fork de un repositorio?

Utilizando GitHub, podemos ir al repositorio y presionar el botón 'FORK'.

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

Podemos crear desde GitHub, sección 'Pull Request', una nueva solicitud, indicando la rama origen y la rama destino.

- ¿Cómo aceptar una solicitud de extracción?

Desde GitHub, sección 'Pull Request', abrimos un PR creado, y revisamos los cambios. En caso de que todo esté OK, se puede aprobar.

- ¿Qué es un etiqueta en Git?

Es un tipo de marcador para poder indicar commits especiales (por ej, versiones del proyecto)

- ¿Cómo crear una etiqueta en Git?

Utilizando '**git tag <NOMBRE>**', creamos una etiqueta al commit actual.

- ¿Cómo enviar una etiqueta a GitHub?

Podemos utilizar:

'git push origin <ETIQUETA>' para enviar una etiqueta particular

'git push --tags' para enviar todas las etiquetas actuales

- ¿Qué es un historial de Git?

El historial de git, es utilizado para revisar todos los commits hechos a lo largo de la historia del repositorio.

- ¿Cómo ver el historial de Git?

Podemos utilizar el comando '**git log**'

- ¿Cómo buscar en el historial de Git?

Podemos filtrar utilizando '**git log**', agregando al final alguno de los siguientes parámetros:

--author

Para buscar por autor de commit

-- archivo

Para buscar los cambios realizados a un archivo

- ¿Cómo borrar el historial de Git?

NO se puede borrar el historial, sin borrar commits o repositorios.

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que solamente tiene acceso el usuario que lo creó, o los usuarios a los que se les permitió acceso

- ¿Cómo crear un repositorio privado en GitHub?

De la misma forma que se crea un repositorio en GitHub, indicamos la opción '**Private**'.

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

En nuestro repo, desde la tab '**Settings**', navegamos a '**Collaborators**'.

Presionamos '**Add people**' e invitamos a la persona(s) que queremos que pueda ver nuestro repositorio.

- ¿Qué es un repositorio público en GitHub?

Es un repositorio el cuál todo el mundo puede ver, incluso sin una cuenta de GitHub

- ¿Cómo crear un repositorio público en GitHub?

De la misma forma que se crea un repositorio en GitHub, indicamos la opción '**Public**'.

- ¿Cómo compartir un repositorio público en GitHub?

Podemos compartir la URL del mismo.

Matías Ruben Boyer Rodríguez

DNI 43.901.799

matiasboyer7@gmail.com

2) Realizar la siguiente actividad:

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Elije 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 * MatiasBoyer / Repository name * tupad_repo_ej2 nombre repo

tupad_repo_ej2 is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-memory](#)?

Description (optional)

☐ Public repo público
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 lo inicializo con un archivo README

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

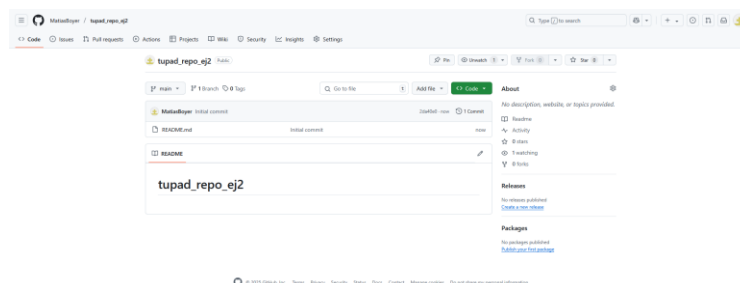
License: None

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](#)



Para poder seguir con los ejercicios, agrego el origen remoto:

```
Windows PowerShell
PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion> mkdir tupad_repo_ej2

Directorio: G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion

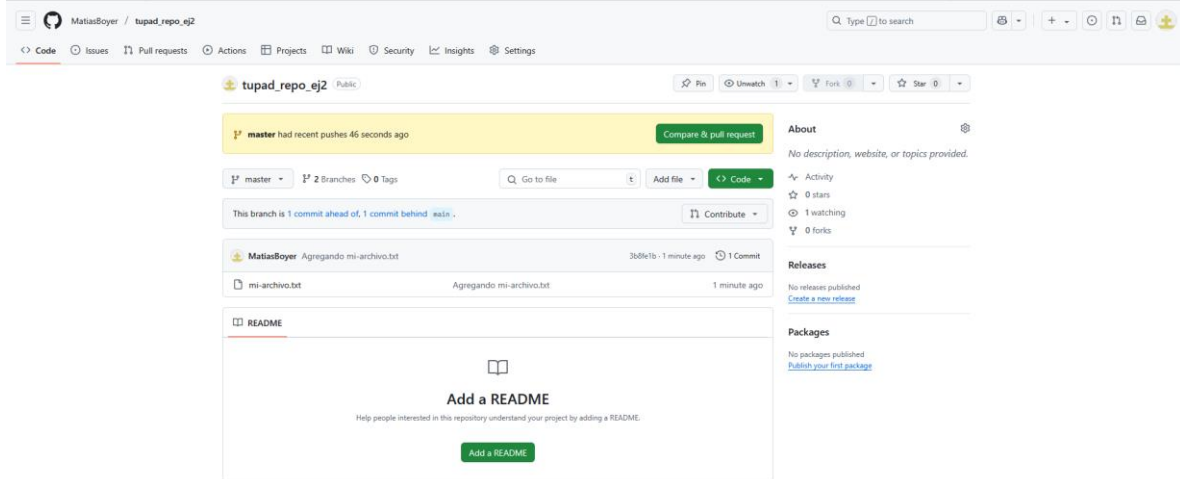
Mode                LastWriteTime         Length Name
----                -
d-----          24/3/2025   21:37         tupad_repo_ej2

PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion> cd .\tupad_repo_ej2\
PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git remote add origin https://github.com/MatiasBoyer/tupad_repo_ej2.git
fatal: not a git repository (or any of the parent directories): .git
PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git init
Initialized empty Git repository in G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2/.git/
PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git remote add origin https://github.com/MatiasBoyer/tupad_repo_ej2.git
PS G:\Mi unidad\MATIAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> |
```

Matías Ruben Boyer Rodríguez
DNI 43.901.799
matiasboyer7@gmail.com

- 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).

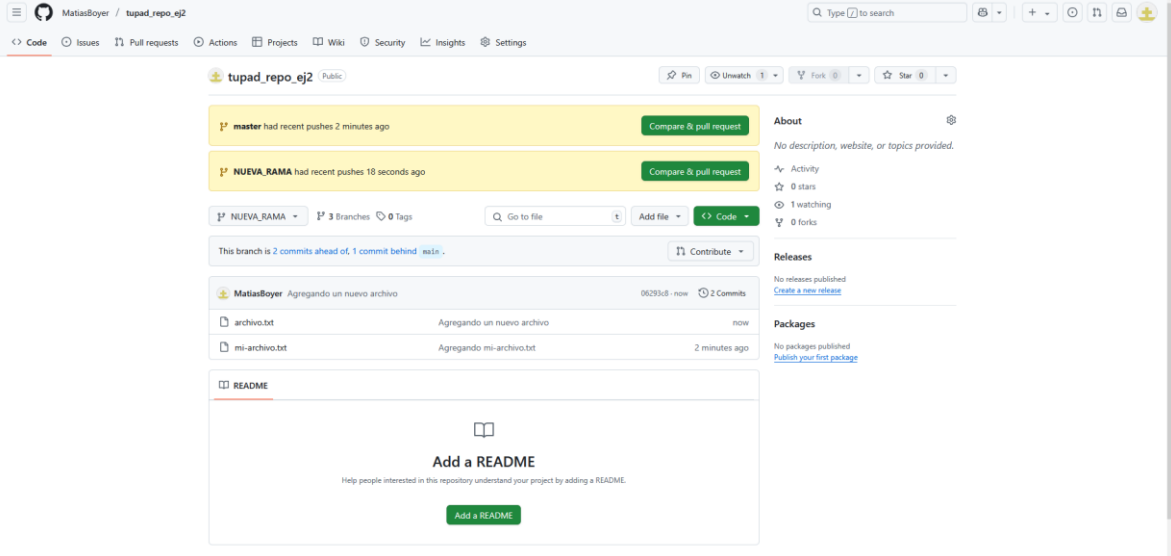
```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> echo "ARCHIVO TXT ejercicio 2" > mi-archivo.txt
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git add .
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git commit -m "Agregando mi-archivo.txt"
[master (root-commit) 3b8fe1b] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git status
On branch master
nothing to commit, working tree clean
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 55.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/MatiasBoyer/tupad_repo_ej2/pull/new/master
remote:
To https://github.com/MatiasBoyer/tupad_repo_ej2.git
 * [new branch]      master -> master
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2>
```



Matías Ruben Boyer Rodríguez
 DNI 43.901.799
 matiasboyer7@gmail.com

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

```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git branch NUEVA_RAMa
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git checkout NUEVA_RAMa
Switched to branch 'NUEVA_RAMa'
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> echo "NUEVO ARCHIVO!!!!" > archivo.txt
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git add .
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git commit -m "Agregando un nuevo archivo"
[NUEVA_RAMa 06293c8] Agregando un nuevo archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo.txt
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> git push origin NUEVA_RAMa
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 64.00 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/MatiasBoyer/tupad_repo_ej2/pull/new/NUEVA_RAMa
remote:
To https://github.com/MatiasBoyer/tupad_repo_ej2.git
 * [new branch]      NUEVA_RAMa -> NUEVA_RAMa
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\tupad_repo_ej2> |
```

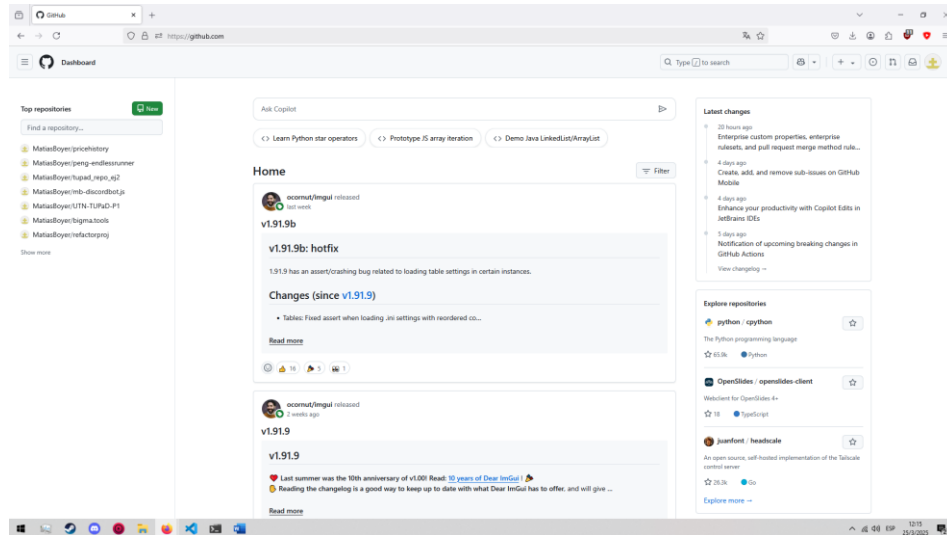


Matías Ruben Boyer Rodríguez
 DNI 43.901.799
 matiasboyer7@gmail.com

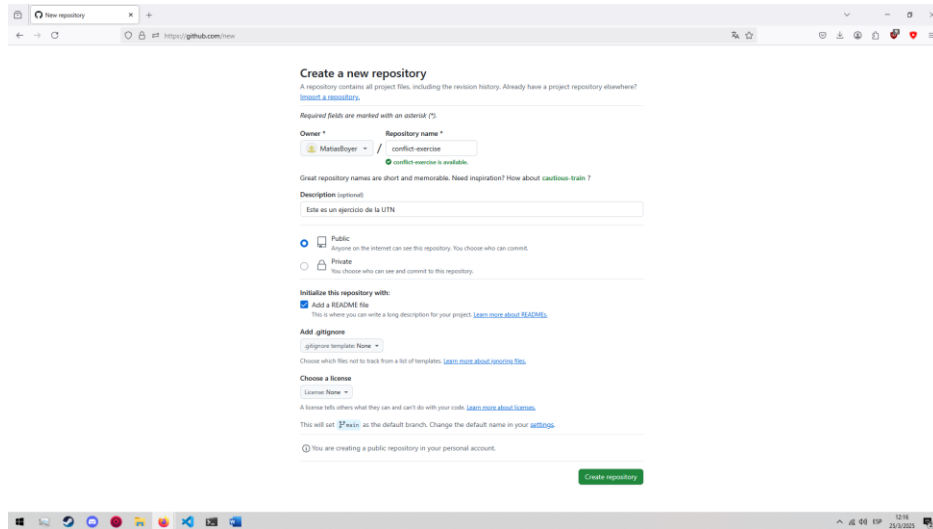
3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

- Ve a GitHub e inicia sesión en tu cuenta.

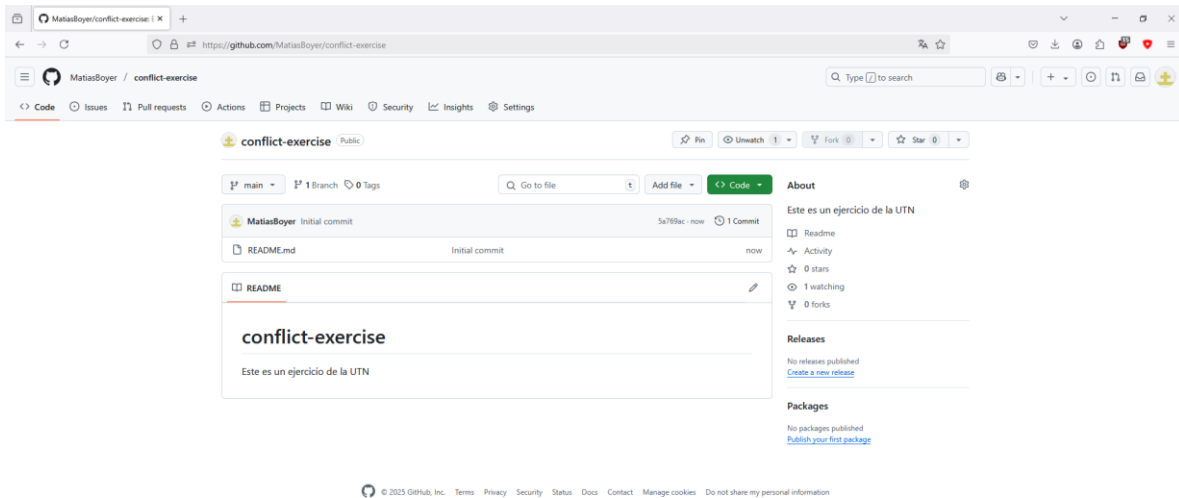


- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".



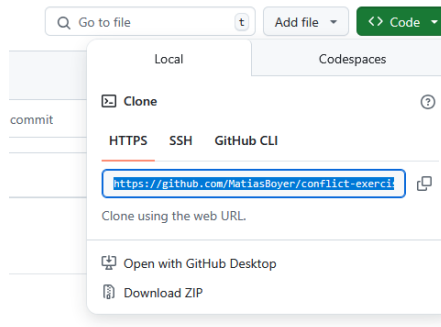
Matías Ruben Boyer Rodríguez
DNI 43.901.799
matiasboyer7@gmail.com

- Haz clic en "Create repository".



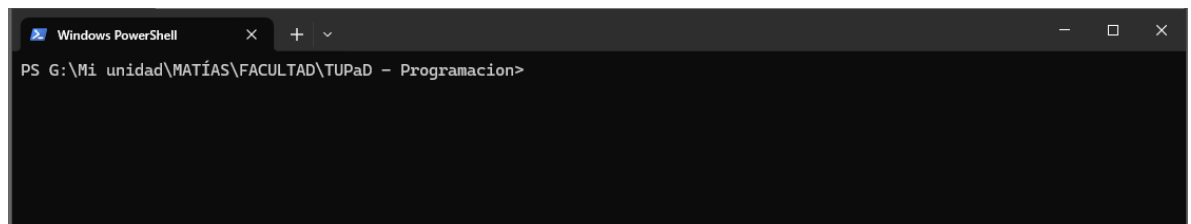
Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuario/conflict-exercise.git>).



<https://github.com/MatiasBoyer/conflict-exercise.git>

- Abre la terminal o línea de comandos en tu máquina.



Matías Ruben Boyer Rodríguez
DNI 43.901.799
matiasboyer7@gmail.com

- Clona el repositorio usando el comando:

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

- Entra en el directorio del repositorio:

`cd conflict-exercise`

```
Windows PowerShell
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion> git clone https://github.com/MatiasBoyer/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 G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion> cd .\conflict-exercise\
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise>
```

Paso 3: Crear una nueva rama y editar un archivo

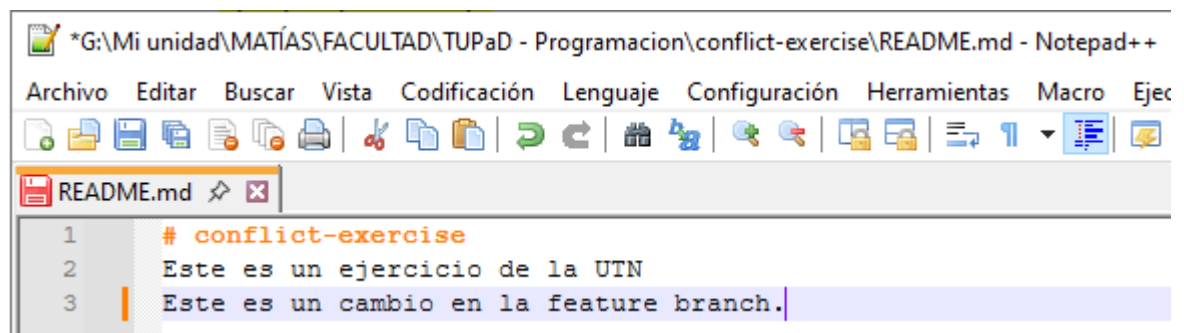
- Crea una nueva rama llamada feature-branch:

`git checkout -b feature-branch`

```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git checkout -b feature-branch
Switched to a new branch '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"`

```
Windows PowerShell
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git add README.md
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git commit -m "Added a line in feature-branch"
[feature-branch d042d8e] Added a line in a feature-branch
1 file changed, 1 insertion(+)
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\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`

```
Windows PowerShell
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> |
```

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

Este es un cambio en la main branch.

```
G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise\README.md - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Complementos  Pestañas ?
1  # conflict-exercise
2  Este es un ejercicio de la UTN
3  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"`

```
Windows PowerShell
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git add .\README.md
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git commit -m "Added a line in main branch"
[main b12b0dc] Added a line in main branch
1 file changed, 1 insertion(+)
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\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.

```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\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 G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise>
```

Paso 6: Resolver el conflicto

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

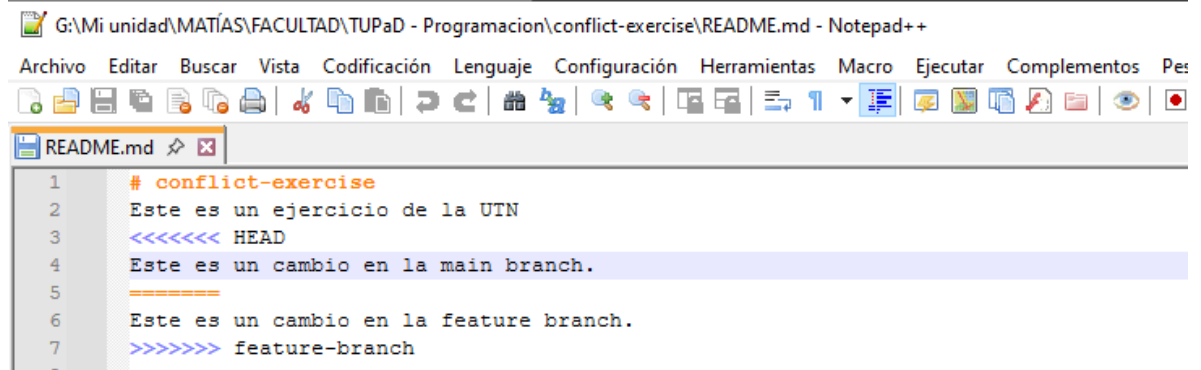
<<<<<< HEAD

Este es un cambio en la main branch.

=====

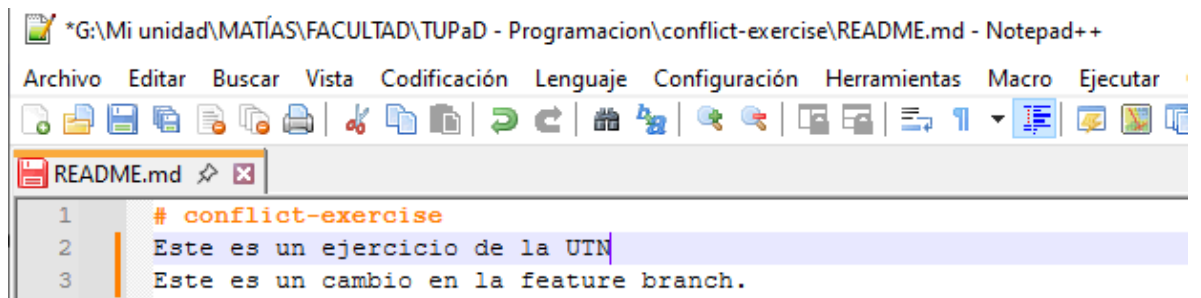
Este es un cambio en la feature branch.

>>>>>> feature-branch



```
G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise\README.md - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Complementos  Pe:
README.md
1  # conflict-exercise
2  Este es un ejercicio de la UTN
3  <<<<<< HEAD
4  Este es un cambio en la main branch.
5  =====
6  Este es un cambio en la feature branch.
7  >>>>>> feature-branch
8
```

- 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).

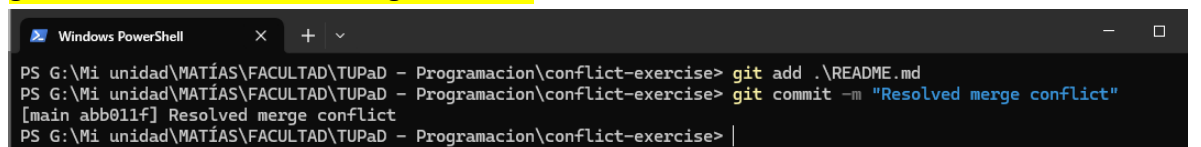


```
*G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise\README.md - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar
README.md
1  # conflict-exercise
2  Este es un ejercicio de la UTN
3  Este es un cambio en la feature branch.
```

- Añade el archivo resuelto y completa el merge:

git add README.md

git commit -m "Resolved merge conflict"



```
Windows PowerShell
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git add .\README.md
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git commit -m "Resolved merge conflict"
[main abb011f] Resolved merge conflict
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\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

```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 636 bytes | 17.00 KiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/MatiasBoyer/conflict-exercise.git
 5a769ac..abb011f main -> main
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> |
```

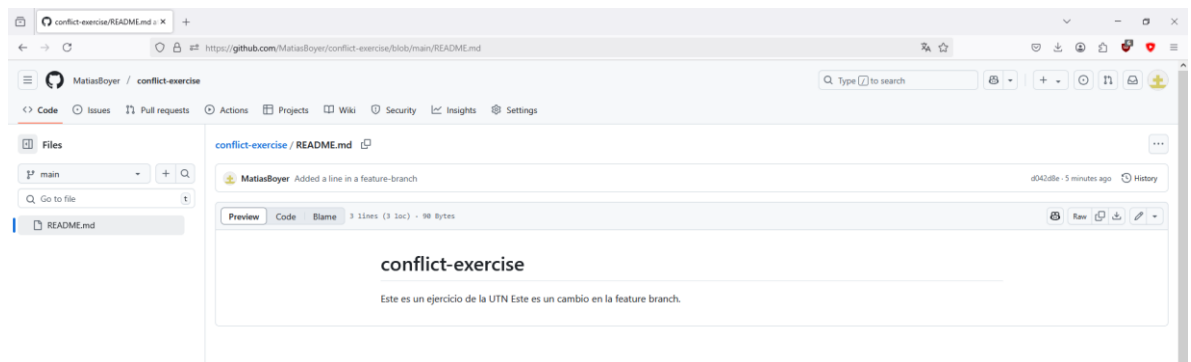
- También sube la feature-branch si deseas:

git push origin feature-branch

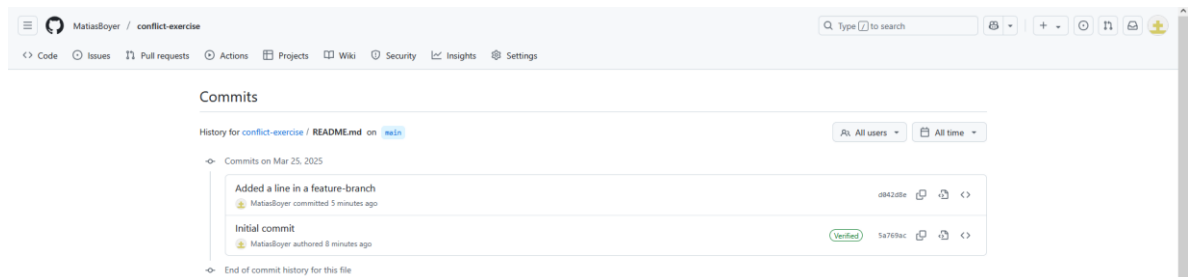
```
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\conflict-exercise> git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/MatiasBoyer/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/MatiasBoyer/conflict-exercise.git
 * [new branch]      feature-branch -> feature-branch
PS G:\Mi unidad\MATÍAS\FACULTAD\TUPaD - Programacion\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.



Matías Ruben Boyer Rodríguez
DNI 43.901.799
matiasboyer7@gmail.com