



Práctico 2: Git y GitHub

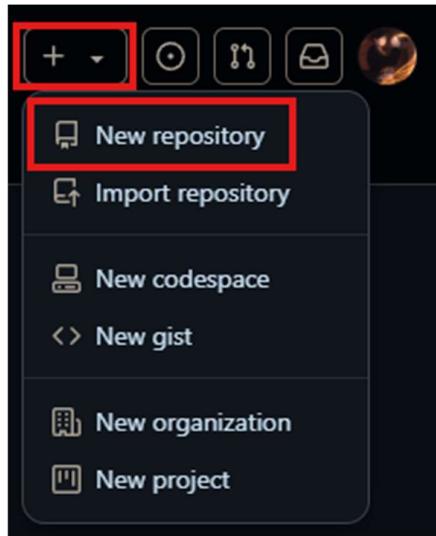
Estudiante: Avilas Martín Jonás

Actividades

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

- ¿Qué es GitHub?
GitHub es una plataforma digital que facilita el almacenamiento y administración de proyectos de código mediante Git. Opera como un backup colaborativo y remoto en el que los programadores tienen la posibilidad de almacenar versiones de su código (mediante Git), colaborar con otros de manera ordenada y compartir sus proyectos de forma pública (código abierto) o privada.
- ¿Cómo crear un repositorio en GitHub?

1. Entrá a GitHub.
2. Arriba a la derecha hay un botón con un “+”.
3. Hacé clic ahí y elegí “**New repository**”.



- ¿Cómo crear una rama en Git?

```
git Branch nombre-rama
```

- ¿Cómo cambiar a una rama en Git?

```
git checkout rama
```

- ¿Cómo fusionar ramas en Git?

```
git merge rama
```

- ¿Cómo crear un commit en Git?

```
git commit -m "mensaje"
```

- ¿Cómo enviar un commit a GitHub?

```
git push
```

- ¿Qué es un repositorio remoto?

Es una copia del proyecto que está alojada en Internet, normalmente en plataformas como GitHub o GitLab.

- ¿Cómo agregar un repositorio remoto a Git?

Cuando tenés tu proyecto local (una carpeta) y querés vincularlo con GitHub:

```
git remote add origin https://github.com/tu-usuario/tu-repo.git
```

origin es el nombre que le das al remoto (la mayoría de las veces, se llama así)
Ese link es el del repositorio en GitHub

- ¿Cómo empujar cambios a un repositorio remoto?

```
git push origin main
```

- ¿Cómo tirar de cambios de un repositorio remoto?

```
git pull origin main
```

- ¿Qué es un fork de repositorio?

Un fork es como hacer una copia de un proyecto de otra persona a tu cuenta de GitHub, para poder modificarlo sin afectar el original.

- ¿Cómo crear un fork de un repositorio?

1. Entrás al repositorio que querés forkar.
2. Arriba a la derecha hay un botón que dice "Fork".
3. Hacés clic, y GitHub crea una copia en tu cuenta.

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

1. Hacés un fork del repo y clonás tu copia.
2. Creamos una rama nueva, hacemos cambios y los subimos (git push origin rama-nueva)
3. Entrás al repo en GitHub y le das a "Compare & pull request"
4. Completás título, descripción y le das a "Create pull request"

- ¿Cómo aceptar una solicitud de extracción?

En la pestaña "Pull Requests":

1. Revisar los cambios propuestos
2. Hacer clic en "Merge pull request"
3. Confirmar con "Confirm merge"

- ¿Qué es una etiqueta en Git?

Una etiqueta es como un marcador en el tiempo. Se usa para marcar versiones importantes del proyecto, por ej: "v1.0".

- ¿Cómo crear una etiqueta en Git?

```
git tag v1.0 (manera simple)
```

```
git tag -a v1.0 -m "Versión 1.0 estable" (manera más completa)
```

- ¿Cómo enviar una etiqueta a GitHub?

```
git push origin v1.0 (si sólo tenés una para enviar)  
git push origin --tags (para subir directamente todas)
```

- ¿Qué es un historial de Git?

Es la lista de todos los commits hechos en un repositorio. Podés ver qué cambios se hicieron, cuándo y por quién.

- ¿Cómo ver el historial de Git?

```
git log
```

- ¿Cómo buscar en el historial de Git?

```
git log --grep="palabra clave" (por mensaje)  
git log nombre-de-archivo.txt (Por archivo)
```

- ¿Cómo borrar el historial de Git?

```
rm -rf .git
```

- ¿Qué es un repositorio privado en GitHub?

Es un repo que solo vos y las personas que invites pueden ver o modificar. No aparece en las búsquedas públicas.

- ¿Cómo crear un repositorio privado en GitHub?

1. Entrás a GitHub y hacés clic en "New repository"
2. Completás nombre, descripción y tildás la opción "Private"
3. Crear

- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
 1. Entrás a tu repo
 2. Vas a Settings > Collaborators
 3. Ponés el nombre de usuario y hacés clic en "Add collaborator"
- ¿Qué es un repositorio público en GitHub?

Es un repo que cualquiera puede ver, descargar o clonar. Ideal para proyectos de código abierto o para compartir en un portfolio.

- ¿Cómo crear un repositorio público en GitHub?
 1. En GitHub > New repository
 2. Completás los datos y tildás "Public"
 3. Crear
- ¿Cómo compartir un repositorio público en GitHub?

Copiás el link del repo y se lo pasás a quien quieras.

2) *Realizar la siguiente actividad:*

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - 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).

Creando Branchs

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

<https://github.com/314-thon/Tp2-Git-GitHub>

3) *Realizar la siguiente actividad:*

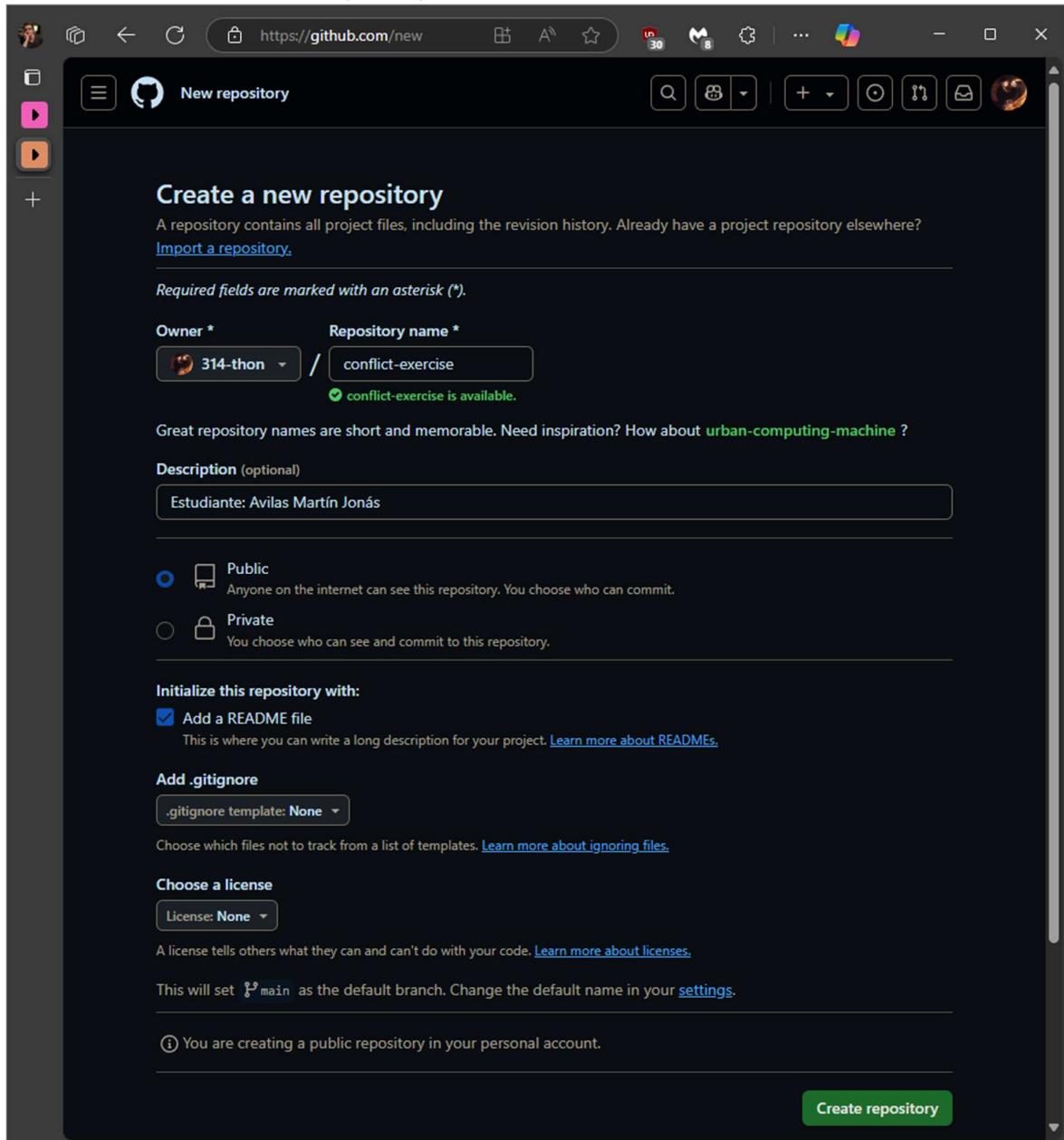
Paso 1: Crear un repositorio en GitHub

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/tuusuari/conflict-exercise.git>).
- • Abre la terminal o línea de comandos en tu máquina.
- • Clona el repositorio usando el comando:

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

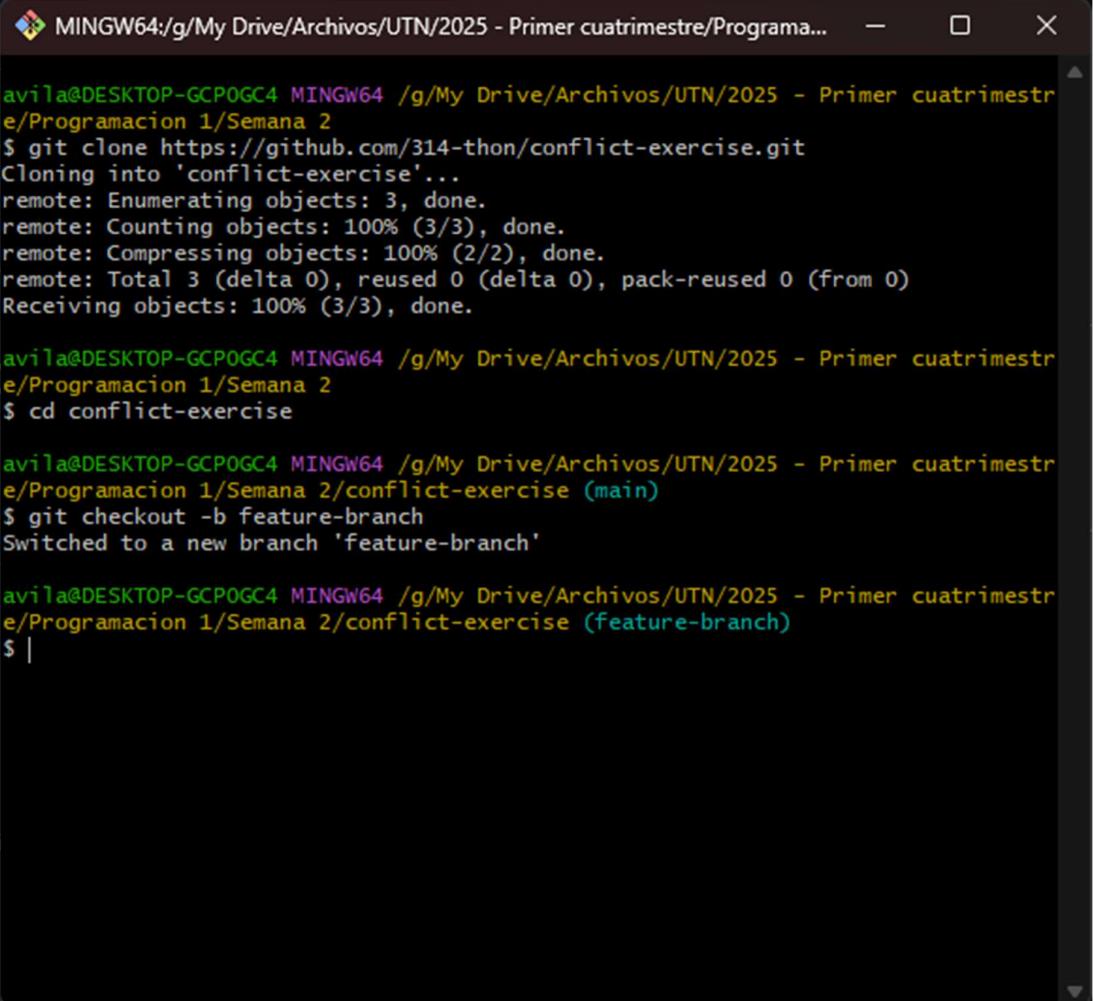
- Entra en el directorio del repositorio:

```
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
```



```
avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2
$ git clone https://github.com/314-thon/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.

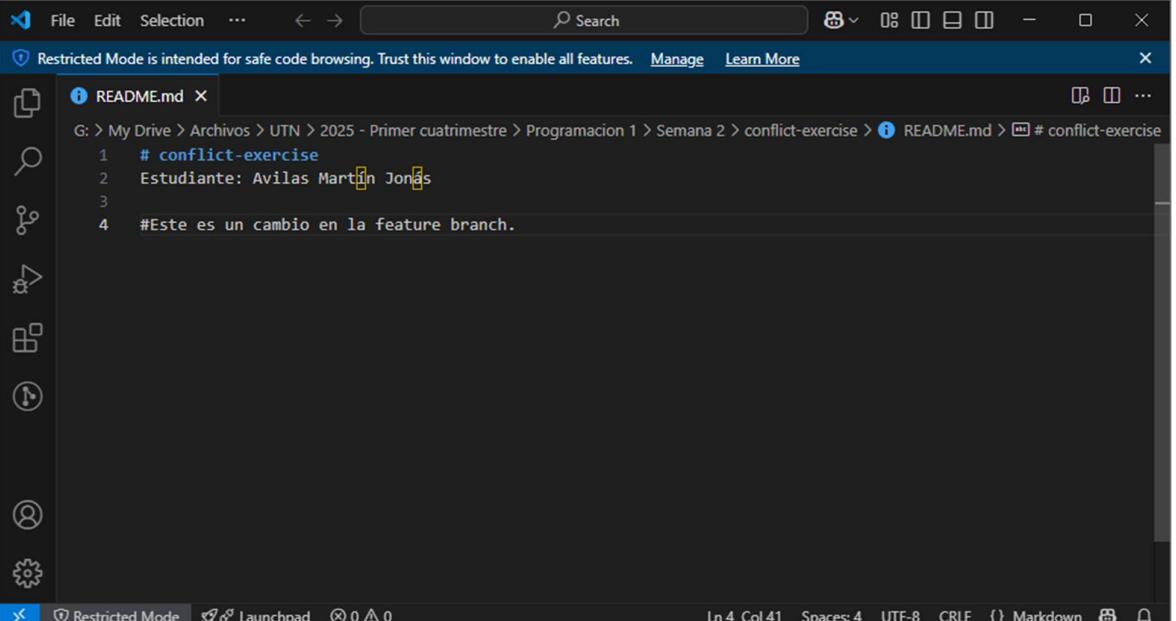
avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2
$ cd conflict-exercise

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (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.



```
File Edit Selection ... ← → ⌂ Search ⓘ README.md ×
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
G: > My Drive > Archivos > UTN > 2025 - Primer cuatrimestre > Programacion 1 > Semana 2 > conflict-exercise > ⓘ README.md > ⓘ # conflict-exercise
1 # conflict-exercise
2 Estudiante: Avilas Martín Jonás
3
4 #Este es un cambio en la feature branch.
```

- Guarda los cambios y haz un commit:

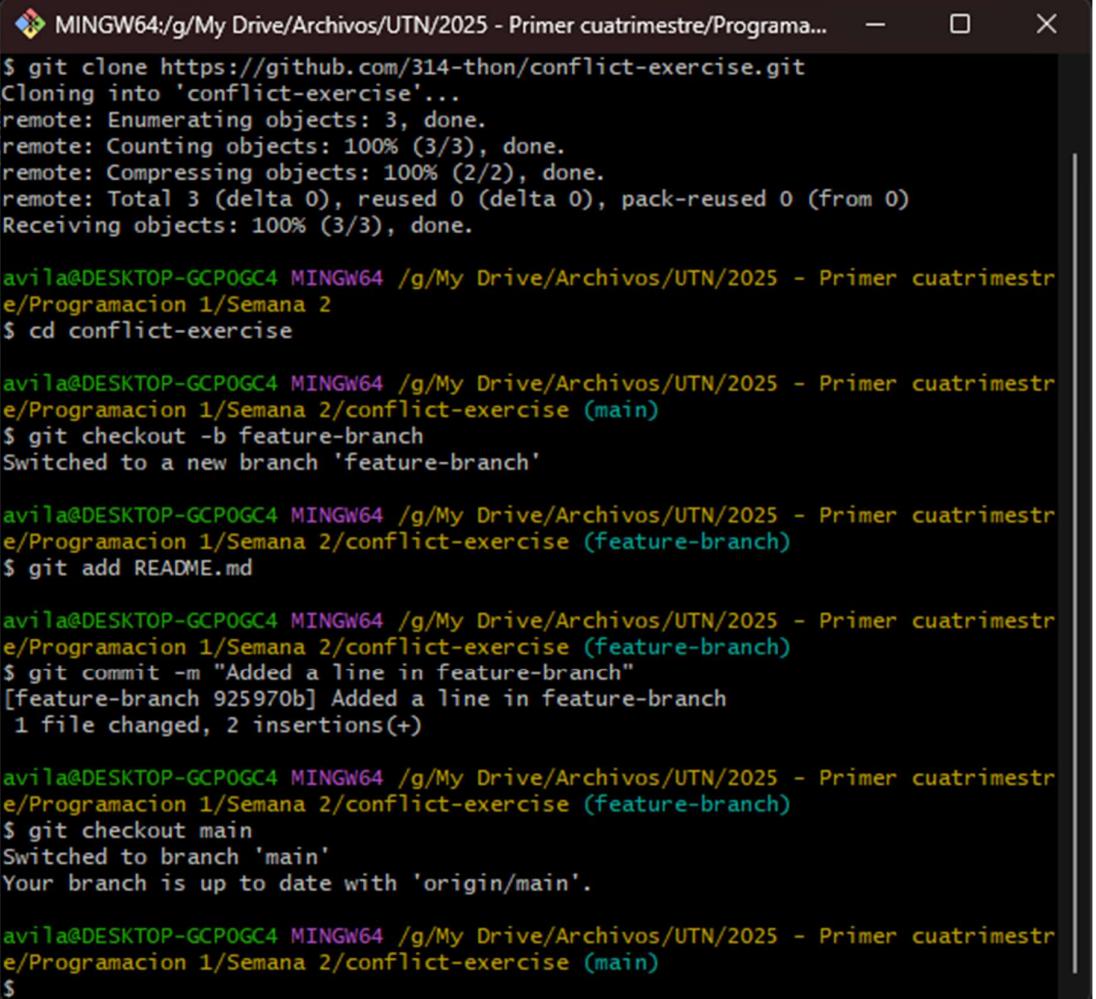
```
git add README.md
```

```
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
```



```
$ git clone https://github.com/314-thon/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.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2
$ cd conflict-exercise

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (feature-branch)
$ git add README.md

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 925970b] Added a line in feature-branch
 1 file changed, 2 insertions(+)

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$
```

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

Este es un cambio en la main branch.

The screenshot shows a GitHub Codespace interface. The top bar includes 'File', 'Edit', 'Selection', a search bar, and window control buttons. A message at the top says 'Restricted Mode is intended for safe code browsing. Trust this window to enable all features.' with links to 'Manage' and 'Learn More'. The left sidebar has icons for file operations like copy, paste, move, delete, and refresh. The main area shows a file tree: G: > My Drive > Archivos > UTN > 2025 - Primer cuatrimestre > Programacion 1 > Semana 2 > conflict-exercise > README.md. The content of README.md is as follows:

```
G: > My Drive > Archivos > UTN > 2025 - Primer cuatrimestre > Programacion 1 > Semana 2 > conflict-exercise > README.md # conflict-exercise
1 # conflict-exercise
2 Estudiante: Avilas Martin Jonas
3
4 #Segundo edit del archivo de texto
```

- 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.

```
MINGW64:/g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programa...
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git add README.md

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git commit -m "Added a linea in main branch"
[main 6f9e385] Added a linea in main branch
1 file changed, 1 insertion(+), 1 deletion(-)

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ 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.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main|MERGING)
$
```

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
```

The screenshot shows a terminal window with the following content:

```
File Edit Selection ... ← → ⌘ Search
i README.md ! x
G: > My Drive > Archivos > UTN > 2025 - Primer cuatrimestre > Programacion 1 > Semana 2 > conflict-exercise > i README.md > # conflict-exercise
You, 21 seconds ago | 1 author (You)
1 # conflict-exercise You, 2 hours ago • Initial commit ...
2 Estudiante: Avilas Martin Jonás
3
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<< HEAD (Current Change)
5 Realizamos cambio en el txt en la línea main para forzar el error
6 =====
7 Este es un cambio en la feature branch.
8 Este es un cambio en la feature branch.
9 >>>> feature-branch (Incoming Change)
10
```

At the bottom right of the terminal window, there is a blue button labeled "Resolve in Merge Editor".

- 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 estos 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
```

```
MINGW64:/g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programa... — □ ×
[feature-branch ac2b124] Added a linea y feature-branch
 1 file changed, 1 insertion(+)

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (feature-branch)
$ git checkout main
git: 'checkout' is not a git command. See 'git --help'.

The most similar command is
  checkout

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git add README.md

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git commit -m "Added a linea in main branch"
[main 6f9e385] Added a linea in main branch
 1 file changed, 1 insertion(+), 1 deletion(-)

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ 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.

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main|MERGING)
$ git add README.md

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main 292ba09] Resolved merge conflict

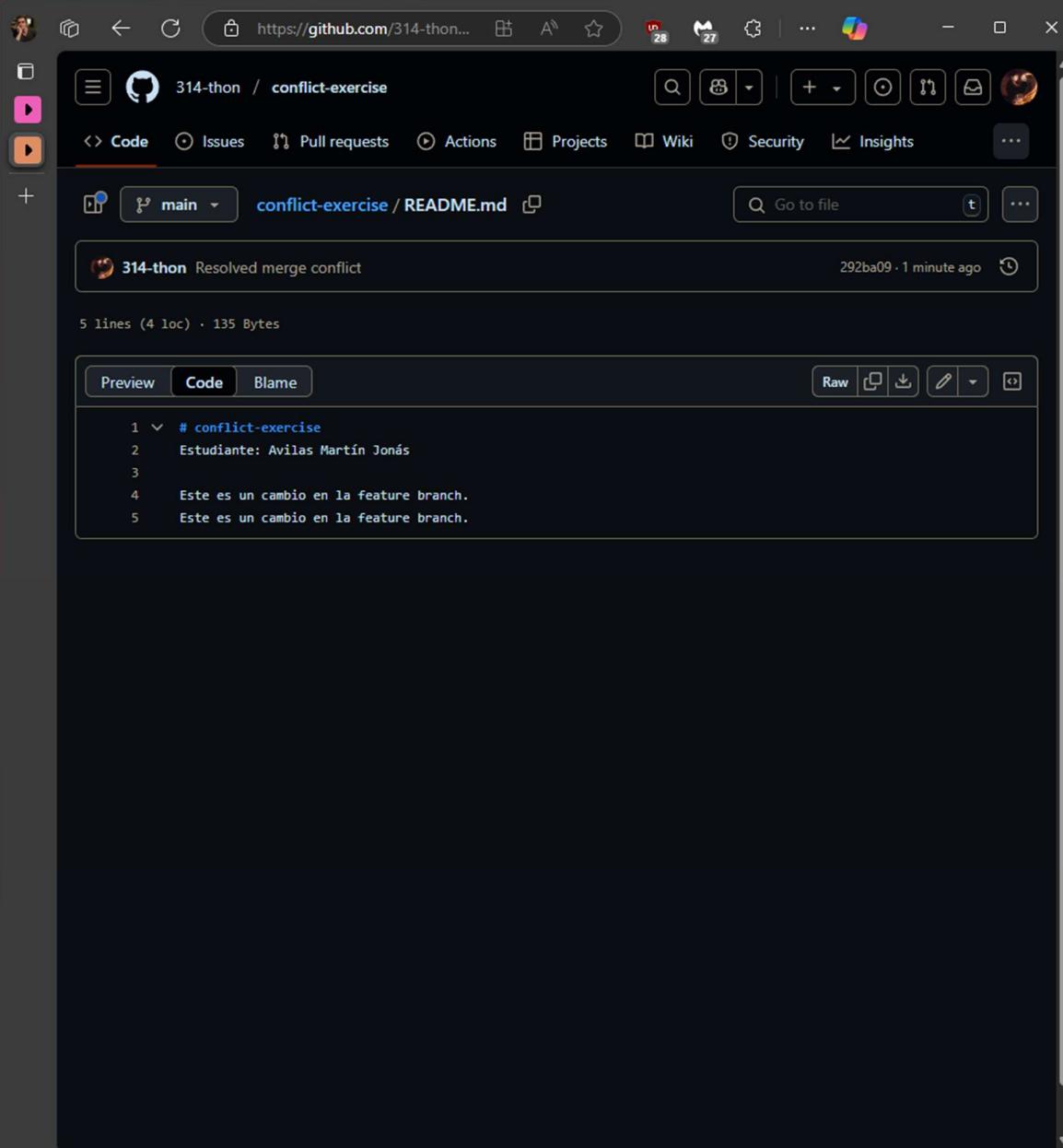
avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 1.29 KiB | 219.00 KiB/s, done.
Total 15 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/314-thon/conflict-exercise.git
  5465adf..292ba09  main -> main

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (main)
$ 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/314-thon/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/314-thon/conflict-exercise.git
 * [new branch]      feature-branch -> feature-branch

avila@DESKTOP-GCP0GC4 MINGW64 /g/My Drive/Archivos/UTN/2025 - Primer cuatrimestre/Programacion 1/Semana 2/conflict-exercise (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



The screenshot shows a GitHub repository interface. The repository name is "314-thon / conflict-exercise". The "Code" tab is selected, showing the file "conflict-exercise / README.md". A commit message from "314-thon" is visible, stating "Resolved merge conflict" with a timestamp of "292ba09 · 1 minute ago". The code editor displays the following content:

```
1 ✓ # conflict-exercise
2   Estudiante: Avilas Martín Jonás
3
4   Este es un cambio en la feature branch.
5   Este es un cambio en la feature branch.
```

<https://github.com/314-thon/conflict-exercise>