

## **ALTERIO MICAELA TP2 GIT Y GITHUB**

### **Actividades:**

Contestar las siguientes preguntas utilizando las guías y documentación proporcionada:

#### **1 ¿Qué es GitHub?**

Es una plataforma de almacenamiento de repositorios de control de versiones que permite la colaboración y el trabajo en equipo en proyectos de desarrollo de software.

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

Una vez iniciada sesión, se hace click en el botón "New repository". Se asigna un nombre y se elige si será público o privado. Luego hacer click en "Create repository".

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

Se ejecuta en la consola el comando "git branch nombre-de-rama"

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

Se ejecuta en la consola el comando "git checkout nombre-de-rama"

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

Se ejecuta en la consola el comando "git merge nombre-de-rama"

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

Se ejecuta en la consola el comando "git add ." y luego "git commit -m "mensaje aclaratorio""

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

Se ejecuta en la consola el comando "git push origin nombre-de-rama"

#### **8 ¿Qué es un repositorio remoto?**

Un repositorio remoto es una versión almacenada en la plataforma de GitHub que permite el acceso a varios colaboradores.

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

Se ejecuta en la consola el comando "git remote add origin URL-del-repositorio"

#### **10 ¿Cómo empujar cambios a un repositorio remoto?**

Se ejecuta en la consola el comando "git push origin nombre-de-rama"

#### **11 ¿Cómo tirar de cambios de un repositorio remoto?**

Se ejecuta en la consola el comando "git pull origin nombre-de-rama"

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

Un fork es una copia de un repositorio en una cuenta de GitHub que permite modificarlo sin afectar el original.

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

Dirigirse al repositorio en GitHub y hacer click en el botón "Fork". Luego GitHub creará una copia en tu cuenta.

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

Subir los cambios a al fork. Luego en el repositorio original, hacer click en "Pull Requests" y luego en "New Pull Request". Seleccionar la rama y hacer click en "Create Pull Request"

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

Dentro del repositorio ir a "Pull Requests". Seleccionar la solicitud. Hacer click en "Merge Pull Request" y confirmar la fusión.

## **16 ¿Qué es un etiqueta en Git?**

Las etiquetas marcan versiones específicas en la historia del proyecto.

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

Se ejecuta el comando "git tag -a v1.0 -m "Versión 1.0""

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

Se ejecuta el comando "git push origin v1.0"

## **19 ¿Qué es un historial de Git?**

El historial de Git registra todos los commits realizados en el repositorio.

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

Se ejecuta el comando "git log --oneline --graph --decorate --all"

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

Se ejecuta el comando "git log --grep="término-de-búsqueda""

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

El comando "git reset" se usa para borrar cosas.

"git reset" Quita todos los archivos y carpetas del proyecto.

"git reset nombreArchivo" Quita un archivo específico.

"git reset nombreCarpeta/" Quita todos los archivos de esa carpeta.

"git reset nombreCarpeta/nombreArchivo" Quita ese archivo dentro de esa carpeta.

"git reset nombreCarpeta/\*.extensión" Quita todos los archivos que cumplan con la condición indicada dentro de esa carpeta.

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

Un repositorio privado es un repositorio solo accesible para su dueño y colaboradores autorizados.

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

Al crear un repositorio en GitHub, se selecciona "Private" en la configuración.

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

Ingresa a "Settings" y luego "Manage Access" en el repositorio que se quiere compartir. A continuación se hace click en "Invite a collaborator". Se ingresa el usuario que se quiere invitar y se envía la invitación.

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

Un repositorio público que es accesible por cualquier persona en GitHub.

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

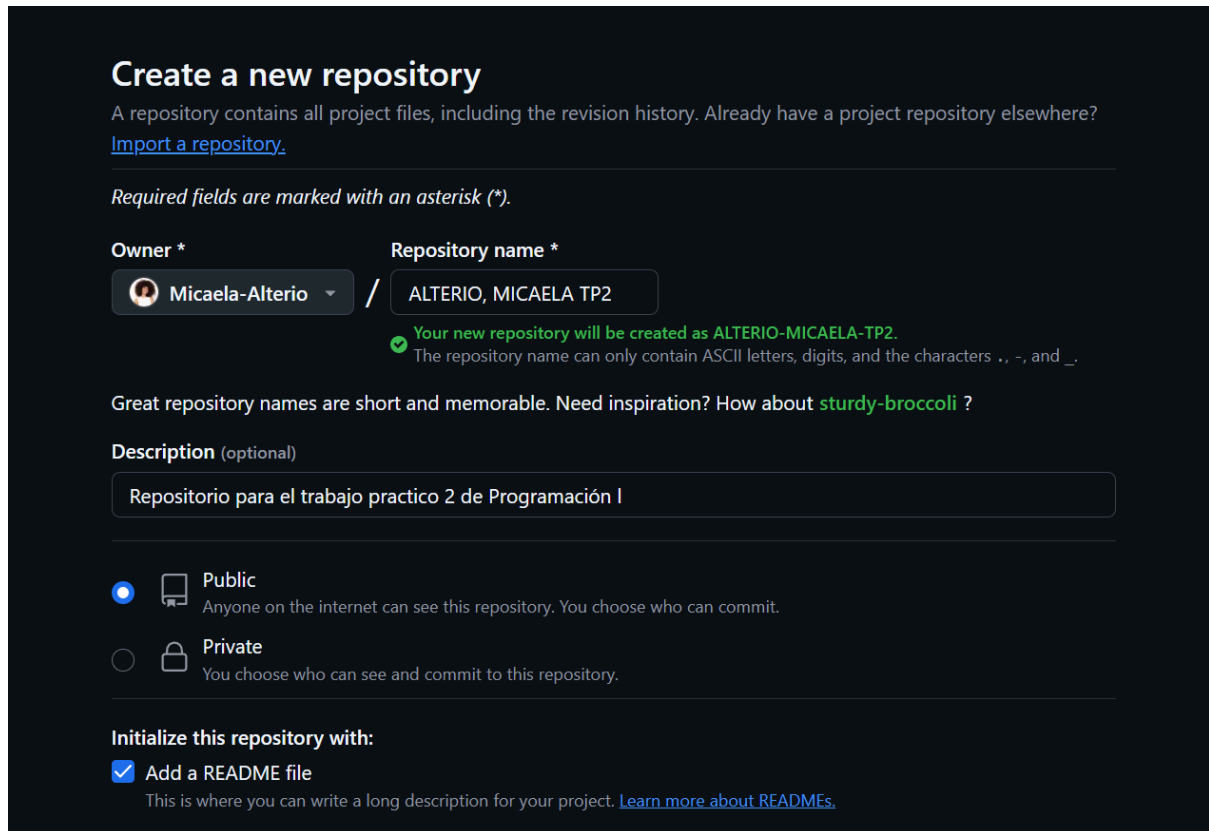
Al crear un repositorio en GitHub, se selecciona "Public" en la configuración.

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

Se comparte la URL del repositorio, copiándolo de la barra del navegador o en la opción "Code" del repositorio.

Realizar la siguiente actividad:

**1.1 Crear un repositorio.** Dale un nombre al repositorio. Elige que 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 \*** Micaela-Alterio / **Repository name \*** ALTERIO, MICAELA TP2

✓ Your new repository will be created as ALTERIO-MICAELA-TP2.  
The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.

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

**Description** (optional)

Repositorio para el trabajo practico 2 de Programación I

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

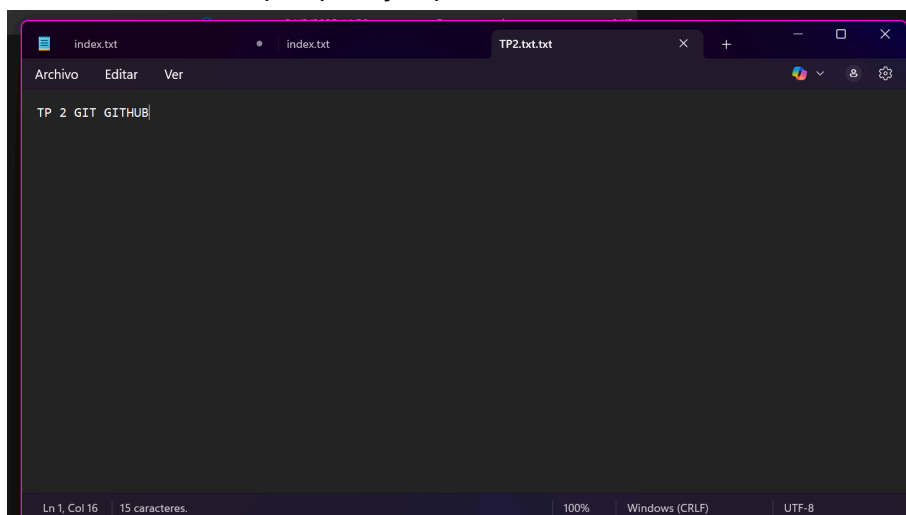
☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

## 1.2 Agregando un Archivo

a) Crea un archivo simple, por ejemplo, "mi-archivo.txt".



- b) Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.

```
MINGW64/c:/Users/mikaa/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git init
Initialized empty Git repository in C:/Users/mikaa/OneDrive/Documentos/TUP/Prime
r Cuatrimestre/PROGRAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub/.git/

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git add .

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git commit -m "Agregando archivo TP2.txt"
[master (root-commit) b87e8f2] Agregando archivo TP2.txt
1 file changed, 1 insertion(+)
create mode 100644 TP2.txt.txt

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ |
```

- c) Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

```
mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 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/Micaela-Alterio/ALTERIO-MICAELA-TP2/pull/new/mas
ter
remote:
To https://github.com/Micaela-Alterio/ALTERIO-MICAELA-TP2.git
 * [new branch]      master -> master
```

### 1.3 Creando Branchs

- a) Crear una Branch

```
mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git branch nueva-rama

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git branch
* master
  nueva-rama

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROG
RAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (master)
$ git checkout nueva-rama
D      TP2.txt.txt
Switched to branch 'nueva-rama'
```

- b) Realizar cambios o agregar un archivo

```
mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROGRAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (nueva-rama)
$ git add .

mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROGRAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (nueva-rama)
$ git commit -m "modificando nueva rama"
[nueva-rama fd4dd81] modificando nueva rama
2 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 TP2.txt
delete mode 100644 TP2.txt.txt
```

- c) Subir la Branch

```
mikaa@DESKTOP-A0080JL MINGW64 ~/OneDrive/Documentos/TUP/Primer Cuatrimestre/PROGRAMACIÓN 1/TP/ALTERIO MICAELA TP2 Git GitHub (nueva-rama)
$ git push origin nueva-rama
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 488 bytes | 244.00 KiB/s, done.
Total 6 (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/Micaela-Alterio/ALTERIO-MICAELA-TP2/pull/new/nueva-rama
remote:
To https://github.com/Micaela-Alterio/ALTERIO-MICAELA-TP2.git
 * [new branch]      nueva-rama -> nueva-rama
```

## 2.1 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".


### 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 \*

 Micaela-Alterio

/

TP2 conflict-exercise

 Your new repository will be created as TP2-conflict-exercise.

The repository name can only contain ASCII letters, digits, and the characters `.`, `-`, and `_`.

Great repository names are short and memorable. Need inspiration? How about **psychic-robot** ?

Description (optional)

Repositorio para el ejercicio de resolucion de conflictos

☒ 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.](#)

## 2.2 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`

```
mikaa@DESKTOP-A0080JL MINGW64 ~ (master)
$ git clone https://github.com/Micaela-Alterio/TP2-conflict-exersice.git
Cloning into 'TP2-conflict-exersice'...
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.

mikaa@DESKTOP-A0080JL MINGW64 ~ (master)
$ cd TP2-conflict-exersice
```

## 2.3 Crear una nueva rama y editar un archivo.

- a) Crea una nueva rama llamada feature-branch: `git checkout -b feature-branch`.

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

- b) 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.

```
❖ README.md ●
C: > Users > mikaa > TP2-conflict-exersice > ⓘ README.md > [abc] # TP2-conflict-exersice
1   # TP2-conflict-exersice
2   Repositorio para el ejercicio de resolucion de conflictos
3   Este es un cambio en la feature-brand
```

- c) Guarda los cambios y haz un commit: `git add README.md` `git commit -m "Added a line in feature-branch"`.

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (feature-branch)
$ git add README.md

mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (feature-branch)
$ git commit -m "Agregando una linea a feature-branch"
[feature-branch 61225b0] Agregando una linea a feature-branch
1 file changed, 1 insertion(+)
```

## 2.4 Volver a la rama principal y editar el mismo archivo

- a) Cambia de vuelta a la rama principal (main): `git checkout main`

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

- b) Edita el archivo README.md de nuevo, añadiendo una línea diferente: Este es un cambio en la main branch.

```
① README.md ●
C: > Users > mikaa > TP2-conflict-exersice > ① README.md > [abc] # TP2-conflict-exersice
1 # TP2-conflict-exersice
2 Repositorio para el ejercicio de resolucion de conflictos
3 Este es un cambio en la main branch
```

- c) Guarda los cambios y haz un commit: git add README.md git commit -m "Added a line in main branch"

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main)
$ git add README.md

mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main)
$ git commit -m "Agregando linea en la rama main"
[main 6010434] Agregando linea en la rama main
1 file changed, 1 insertion(+)
```

**2.5 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.

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (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.
```

## 2.6 Resolver el conflicto

- a) 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

```
① README.md X
C: > Users > mikaa > TP2-conflict-exersice > ① README.md > [abc] # TP2-conflict-exersice
1 | # TP2-conflict-exersice
2 | Repositorio para el ejercicio de resolucion de conflictos
3 | Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 | <<<<<<< HEAD (Current Change)
5 | Este es un cambio en la main branch
6 | =====
7 | Este es un cambio en la feature-brand
8 | >>>>>>> feature-branch (Incoming Change)
```



- b) 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).

```
① README.md X
C: > Users > mikaa > TP2-conflict-exersice > ① README.md > # TP2-conflict-exersice
1 # TP2-conflict-exersice
2 Repositorio para el ejercicio de resolucion de conflictos
3 Este es un cambio en la main branch
4 Este es un cambio en la feature-brand
5
```

- c) Añade el archivo resuelto y completa el merge: `git add README.md git commit -m "Resolved merge conflict"`

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main|MERGING)
$ git add README.md

mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main|MERGING)
$ git commit -m "Conflicto resuelto"
[main 9a4f1e6] Conflicto resuelto
```


**2.7 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`.

```
mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 799 bytes | 266.00 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/Micaela-Alterio/TP2-conflict-exersice.git
  2686c2d..9a4f1e6  main -> main

mikaa@DESKTOP-A0080JL MINGW64 ~/TP2-conflict-exersice (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/Micaela-Alterio/TP2-conflict-exersice/pull/new/f
remote:
remote:
To https://github.com/Micaela-Alterio/TP2-conflict-exersice.git
 * [new branch]      feature-branch -> feature-branch
```

## 2.8 Verificar en GitHub

- a) Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.

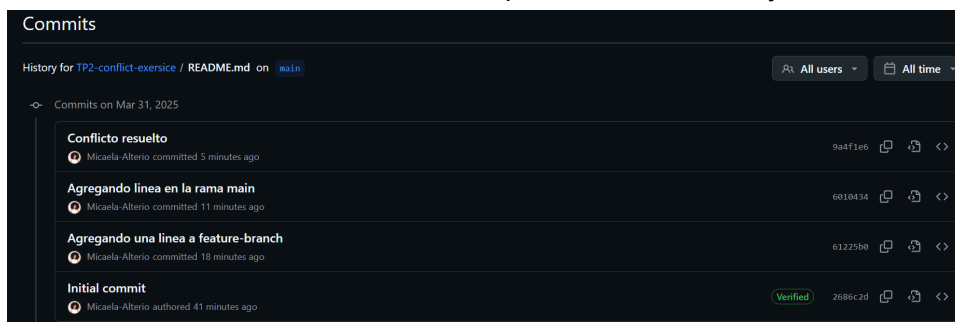


The screenshot shows the GitHub interface for a file named `README.md`. At the top, there are tabs for `Preview`, `Code`, and `Blame`. The `Code` tab is selected. The file content is displayed in a dark-themed editor with line numbers 1 through 4. The text of the file is as follows:

```
1  # TP2-conflict-exersice
2  Repositorio para el ejercicio de resolucion de conflictos
3  Este es un cambio en la main branch
4  Este es un cambio en la feature-brand
```

At the top right of the code area, it says "4 lines (4 loc) · 156 Bytes" and "Code 55% faster with GitHub Copilot".

- b) Puedes revisar el historial de commits para ver el conflicto y su resolución.



The screenshot shows the "Commits" page for the file `README.md` on the `main` branch. The page title is "Commits". Below the title, it says "History for TP2-conflict-exersice / README.md on main". There are filters for "All users" and "All time". The commit history is listed below, showing four commits:

Commit Message	Author	Time	SHA-1	Actions
Conflicto resuelto	Micaela Alterio	committed 5 minutes ago	9a4f1e6	Copy, Download, Compare
Agregando linea en la rama main	Micaela Alterio	committed 11 minutes ago	6010434	Copy, Download, Compare
Agregando una linea a feature-branch	Micaela Alterio	committed 18 minutes ago	61225b0	Copy, Download, Compare
Initial commit	Micaela Alterio	authored 41 minutes ago	2686c2d	Copy, Download, Compare

The "Initial commit" is marked as "Verified".