

# TRABAJO PRÁCTICO N°2

Luciano Martín Rottoli

## Actividad 1

### ¿Qué es GitHub?

GitHub es una plataforma basada en la nube que permite almacenar, gestionar y colaborar en proyectos de desarrollo utilizando Git. Proporciona herramientas para el control de versiones, gestión de código, revisión colaborativa y automatización de procesos.

### ¿Cómo crear un repositorio en GitHub?

1. Inicia sesión en [GitHub](https://github.com).
2. Haz clic en el ícono + en la esquina superior derecha y selecciona **"New repository"**.
3. Ingresa un nombre para el repositorio y selecciona su visibilidad (público o privado).
4. Opcionalmente, puedes añadir una descripción, un archivo README.md, una licencia y un .gitignore.
5. Haz clic en **"Create repository"**.

### ¿Cómo crear una rama en Git?

Para crear una nueva rama en Git, usa el siguiente comando: `git branch nombre-de-la-rama` en donde creará una nueva rama basada en la actual.

### ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama existente, usa el comando: `git checkout nombre-de-la-rama` o en versiones mas recientes `git switch nombre-de-la-rama`.

### ¿Cómo fusionar ramas en Git?

Para fusionar una rama en otra (por ejemplo, feature en main):

1. Cambia a la rama main: `git checkout main`
2. Fusiona la rama feature en main: `git merge feature`

Si hay conflictos, Git te pedirá que los resuelvas antes de completar la fusión.

### ¿Cómo crear un commit en Git?

1. Agrega los archivos al área de preparación: `git add .`

2. Crea el commit con un mensaje descriptivo: `git commit -m "Descripción del cambio"`

### ¿Cómo enviar un commit a GitHub?

Para enviar un commit al repositorio remoto en GitHub: `git push origin nombre-de-la-rama`

Ejemplo si estás en main: `git push origin main`

### ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión almacenada en la nube de un repositorio local. Permite que varios desarrolladores colaboren en el mismo proyecto desde diferentes ubicaciones.

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

Para vincular un repositorio remoto a tu proyecto local:

`git remote add origin URL-del-repositorio`

Ejemplo:

`git remote add origin https://github.com/usuario/repositorio.git`

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

Para enviar los cambios al repositorio remoto: `git push origin nombre-de-la-rama`

Si es la primera vez: `git push -u origin nombre-de-la-rama`

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

Para obtener los últimos cambios desde el repositorio remoto: `git pull origin nombre-de-la-rama`

Esto sincroniza tu repositorio local con el remoto.

### ¿Qué es un fork de repositorio?

Un **fork** es una copia de un repositorio que se almacena en tu cuenta de GitHub. Permite modificar el código sin afectar el repositorio original y contribuir con cambios mediante pull requests.

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

1. Ve al repositorio en GitHub que deseas forquear.

2. Haz clic en el botón **"Fork"** en la parte superior derecha.
3. Espera a que GitHub cree la copia en tu cuenta.

Ahora puedes clonar y trabajar en tu propio repositorio sin afectar el original.

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

1. Asegúrate de haber subido los cambios a una rama en tu repositorio: `git push origin nombre-de-la-rama`
2. Ve al repositorio en GitHub.
3. Dirígete a la pestaña **"Pull requests"** y haz clic en **"New pull request"**.
4. Selecciona la rama con los cambios y compárala con la rama principal del proyecto.
5. Escribe un título y descripción del cambio.
6. Haz clic en **"Create pull request"**.

### ¿Cómo aceptar una solicitud de extracción?

1. Ve a la pestaña **"Pull requests"** en GitHub.
2. Abre la solicitud de extracción.
3. Revisa los cambios y, si todo está correcto, haz clic en **"Merge pull request"**.
4. Confirma la fusión con **"Confirm merge"**.
5. Opcionalmente, elimina la rama con **"Delete branch"**.

### ¿Qué es una etiqueta en Git?

Una **etiqueta** (tag) en Git es un marcador que se usa para identificar versiones importantes en la historia del repositorio, como versiones de lanzamiento (v1.0, v2.0).

### ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta anotada con un mensaje: `git tag -a v1.0 -m "Versión 1.0"`

Para una etiqueta ligera sin mensaje: `git tag v1.0`

### ¿Cómo enviar una etiqueta a GitHub?

Para subir una etiqueta específica: `git push origin v1.0`

Para subir todas las etiquetas: `git push -tags`

### ¿Qué es un historial de Git?

El historial de Git es un registro de todos los commits realizados en un proyecto, incluyendo detalles como el autor, la fecha y el mensaje de cada cambio.

### ¿Cómo ver el historial de Git?

Para ver el historial de commits: `git log`

Para un resumen más compacto: `git log --oneline --graph`

Para ver los cambios detallados en cada commit: `git log -p`

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

Para buscar un commit por una palabra clave en el mensaje: `git log --grep="texto"`

Para buscar cambios en un archivo específico: `git log -- archivo.txt`

Para buscar commits de un autor específico: `git log --author="nombre"`

### ¿Cómo borrar el historial de Git?

No se puede borrar el historial sin eliminar el repositorio, pero se puede sobrescribir con:

```
git checkout --orphan nueva-rama
```

```
git add .
```

```
git commit -m "Nuevo inicio"
```

```
git branch -D main
```

```
git branch -m main
```

```
git push -f origin main
```

 Esto eliminará permanentemente el historial anterior.

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

Un **repositorio privado** es un repositorio en GitHub que solo pueden ver y acceder las personas autorizadas.

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

1. Ve a [GitHub](#) y haz clic en + > **"New repository"**.
2. Escribe un nombre y selecciona **"Private"**.
3. Configura opciones adicionales si es necesario.
4. Haz clic en **"Create repository"**.

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

1. Ve al repositorio en GitHub.
2. Haz clic en **"Settings"** > **"Manage access"**.
3. Haz clic en **"Invite a collaborator"**.
4. Escribe el nombre de usuario o correo de la persona.
5. Envía la invitación y espera que la acepte.

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

Un **repositorio público** es un repositorio en GitHub accesible para cualquier persona. Cualquier usuario puede ver el código, pero solo los colaboradores pueden modificarlo.

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

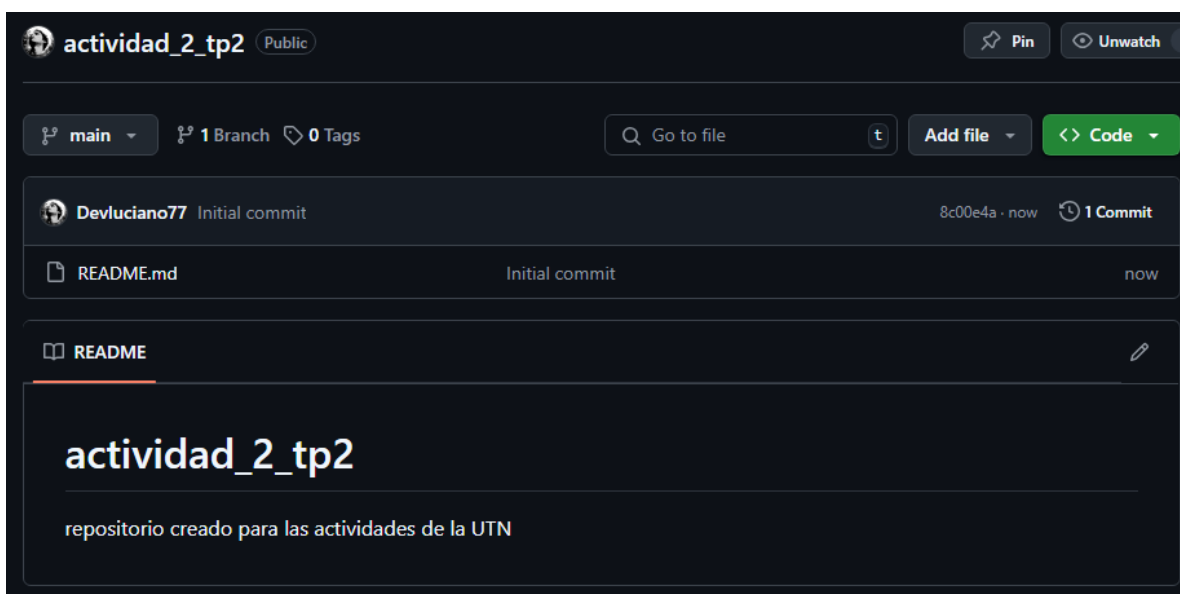
1. Ve a GitHub y haz clic en + > **"New repository"**.
2. Escribe un nombre y selecciona **"Public"**.
3. Configura las opciones y haz clic en **"Create repository"**.

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

1. Copia la URL del repositorio desde la barra de direcciones o usa: `git remote -v`
2. Comparte la URL con otros para que puedan verlo o clonarlo: `git clone https://github.com/usuario/repositorio.git`

## Actividad 2

Crear un repositorio.



## Agregado de un archivo

```
Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2
actividad 2
$ cd actividad_2_tp2

Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2
actividad 2/actividad_2_tp2 (main)
$ git add .

Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2
actividad 2/actividad_2_tp2 (main)
$ git commit -m "Agregando archivo_simple.txt"
[main 19b3593] Agregando archivo_simple.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo_simple.txt

Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2
actividad 2/actividad_2_tp2 (main)
$ git push origin main
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), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Devluciano77/actividad_2_tp2.git
8c00e4a..19b3593 main -> main
```

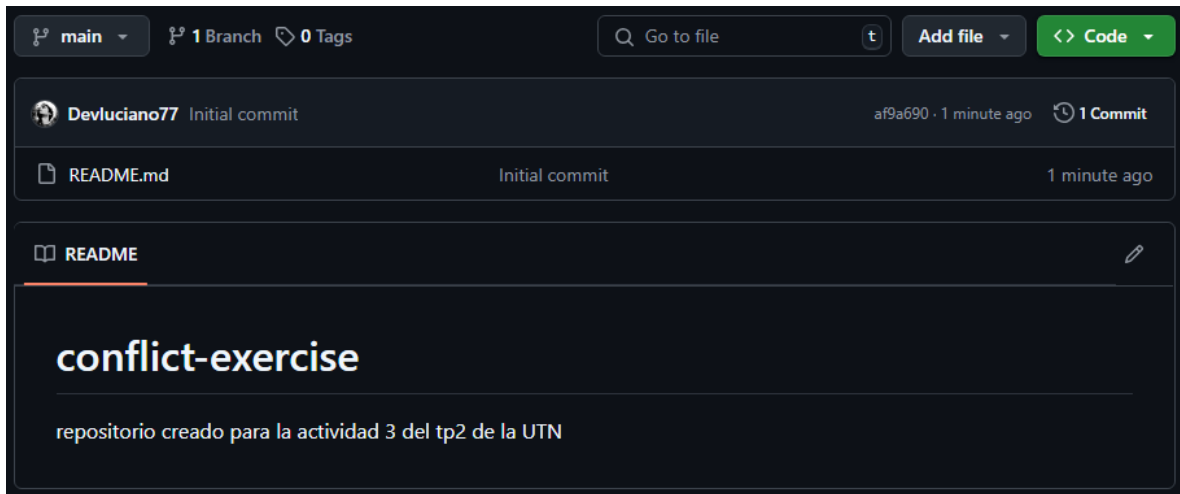
## Creando branches

The screenshot shows the GitHub interface for the repository 'actividad\_2\_tp2'. At the top, it indicates the repository is 'Public' and has 1 watcher. A notification from 'rama2' states 'had recent pushes 58 seconds ago' with a 'Compare & pull request' button. Below this, the repository is owned by 'rama2' and has '2 Branches' and '0 Tags'. A search bar and buttons for 'Add file' and 'Code' are present. A status bar shows 'This branch is 1 commit ahead of main'. The commit history section, titled 'DevLuciano77', lists three commits: 'Initial commit' (13 minutes ago), 'Agregando archivo\_simple.txt' (7 minutes ago), and 'agregado segundo archivo' (1 minute ago). The commit hash '474cac7' and '3 Commits' are also shown.

Commit Message	Author	Time Ago
Initial commit	DevLuciano77	13 minutes ago
Agregando archivo_simple.txt	DevLuciano77	7 minutes ago
agregado segundo archivo	DevLuciano77	1 minute ago

### Actividad 3

Paso 1: Crear un repositorio en GitHub



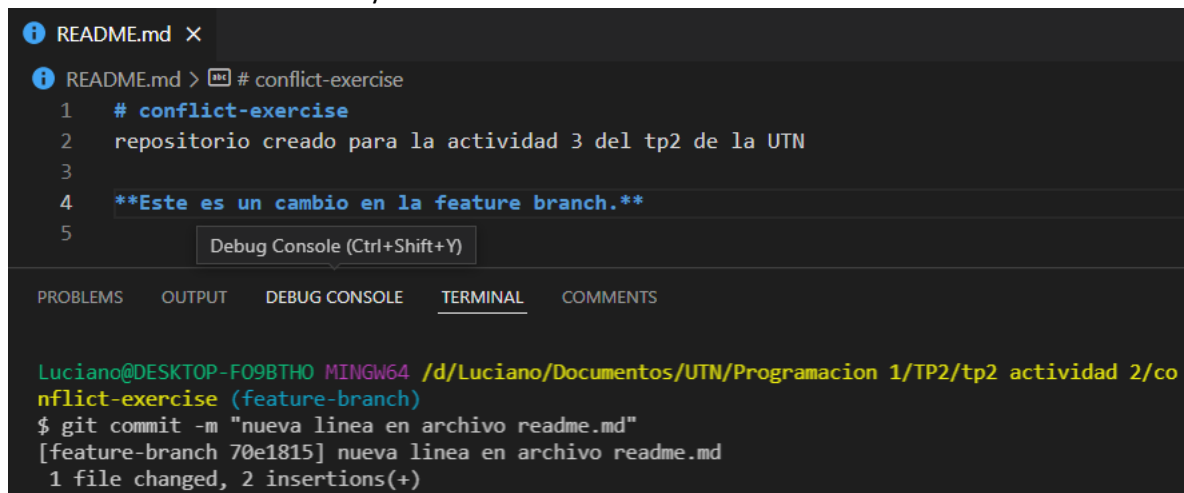
Paso 2: Clonar el repositorio a tu máquina local

```
Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp
actividad 2
$ git clone https://github.com/Devluciano77/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.

Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp
actividad 2
$ cd conflict-exercise

Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp
actividad 2/conflict-exercise (main)
$ |
```

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



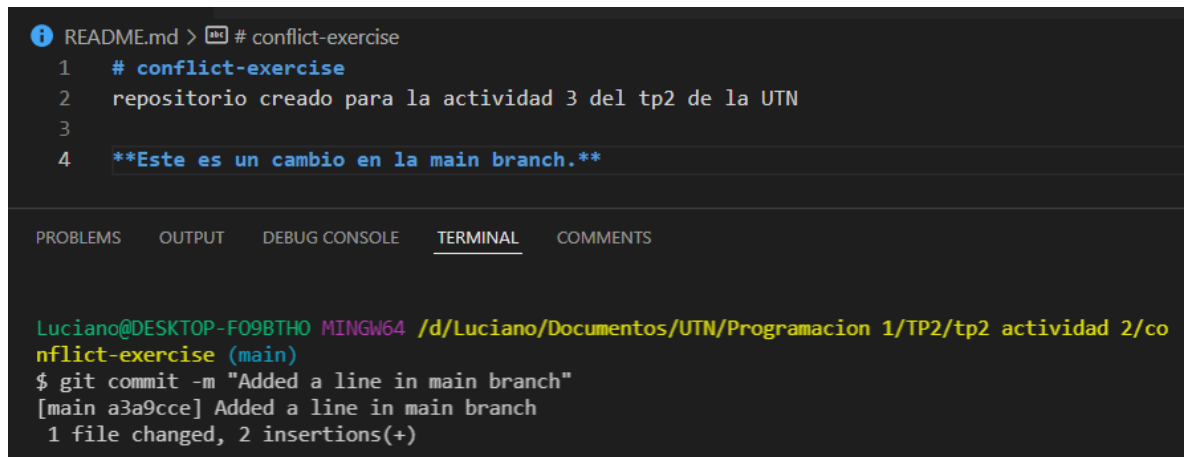
```
README.md > # conflict-exercise
1 # conflict-exercise
2 repositorio creado para la actividad 3 del tp2 de la UTN
3
4 **Este es un cambio en la feature branch.**
5
```

Debug Console (Ctrl+Shift+Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
Luciano@DESKTOP-F09BTH0 MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2 actividad 2/co
nflict-exercise (feature-branch)
$ git commit -m "nueva linea en archivo readme.md"
[feature-branch 70e1815] nueva linea en archivo readme.md
1 file changed, 2 insertions(+)
```

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

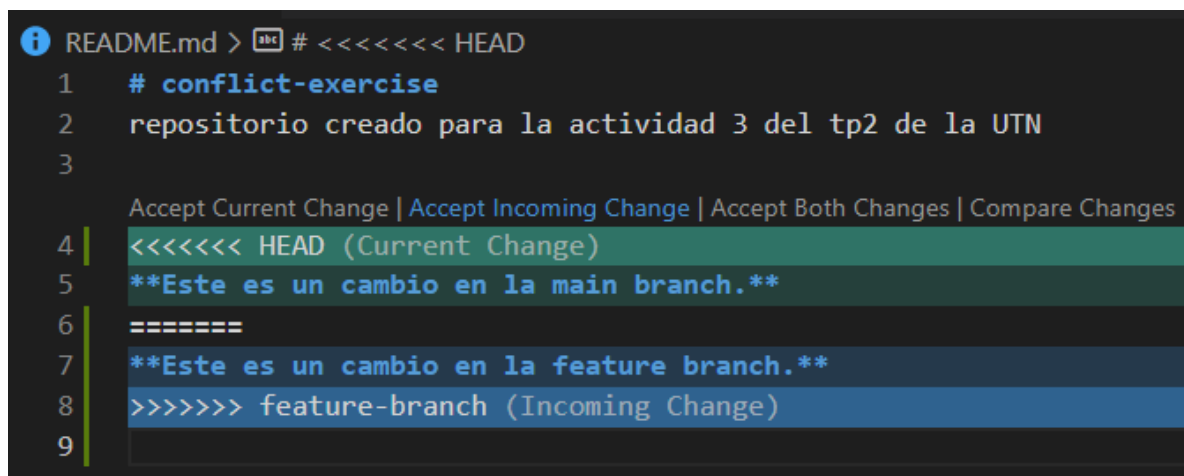


```
README.md > # conflict-exercise
1 # conflict-exercise
2 repositorio creado para la actividad 3 del tp2 de la UTN
3
4 **Este es un cambio en la main branch.**
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
Luciano@DESKTOP-F09BTH0 MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2 actividad 2/co
nflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main a3a9cce] Added a line in main branch
1 file changed, 2 insertions(+)
```

### Paso 5: Hacer un merge y generar un conflicto



```
README.md > # <<<<<<< HEAD
1 # conflict-exercise
2 repositorio creado para la actividad 3 del tp2 de la UTN
3
4 <<<<<<< HEAD (Current Change)
5 **Este es un cambio en la main branch.**
6 =====
7 **Este es un cambio en la feature branch.**
8 >>>>>> feature-branch (Incoming Change)
9
```

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes



## Paso 6: Resolver el conflicto

```
i README.md > # conflict-exercise
1  # conflict-exercise
2  repositorio creado para la actividad 3 del tp2 de la UTN
3
4  **Este es un cambio en la main branch.**
5  **Este es un cambio en la feature branch.**
6
```

## Paso 7: Subir los cambios a GitHub

```
Luciano@DESKTOP-F09BTHO MINGW64 /d/Luciano/Documentos/UTN/Programacion 1/TP2/tp2 actividad 2/co
nflict-exercise (main)
$ 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), 819 bytes | 819.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Devluciano77/conflict-exercise.git
   af9a690..ed3f442  main -> main
```

## Paso 8: Verificar en GitHub

