

Trabajo Práctico 2: Git y GitHub

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

¿Qué es GitHub?

Es una comunidad donde nosotros podemos compartir nuestros repositorios de forma pública o privada. En GitHub podemos crear un perfil, seguir a diferentes usuarios, ver sus proyectos y proponer modificaciones a los mismos, también tiene varias funcionalidades más.

¿Cómo crear un repositorio en GitHub?

Desde nuestro perfil de GitHub podemos crear un repositorio, vamos a la parte superior derecha en el símbolo "+", en "New Repository", nos abre un menú para ingresar el nombre del repositorio y diferentes opciones, entre ellas si queremos que el mismo sea público o privado, también tenemos la opción de agregar un README donde podemos describir nuestro proyecto.

¿Cómo crear una rama en Git?

Para crear una rama en Git, vamos a utilizar el comando **git branch <nombre de la rama>**

¿Cómo cambiar a una rama en Git?

Para movernos a una rama, vamos a utilizar el comando **git switch <nombre a cambiar>**

¿Cómo fusionar ramas en Git?

Para fusionar una rama con otra, vamos a utilizar el comando **git merge <nombre de la rama desde donde traigo los cambios>**

¿Cómo crear un commit en Git?

Para crear un commit, vamos a utilizar el comando **git commit -m <"ingresamos una descripción">**

¿Cómo enviar un commit a GitHub?

Para enviar un commit a nuestro repositorio remoto de GitHub, lo hacemos mediante el siguiente comando: **git push -u <dirección del repositorio remoto>**

¿Qué es un repositorio remoto?

Un repositorio remoto es un lugar donde nosotros guardamos nuestro proyecto en la nube, que es este caso es GitHub.

¿Cómo agregar un repositorio remoto a Git?

Lo que tenemos que hacer es mediante la dirección que encontraremos en GitHub, vamos a usar el comando **git remote add origin <dirección del repositorio>**

¿Cómo empujar cambios a un repositorio remoto?

Para empujar datos a un repositorio remoto, lo hacemos mediante el comando **git push**

¿Cómo tirar de cambios de un repositorio remoto?

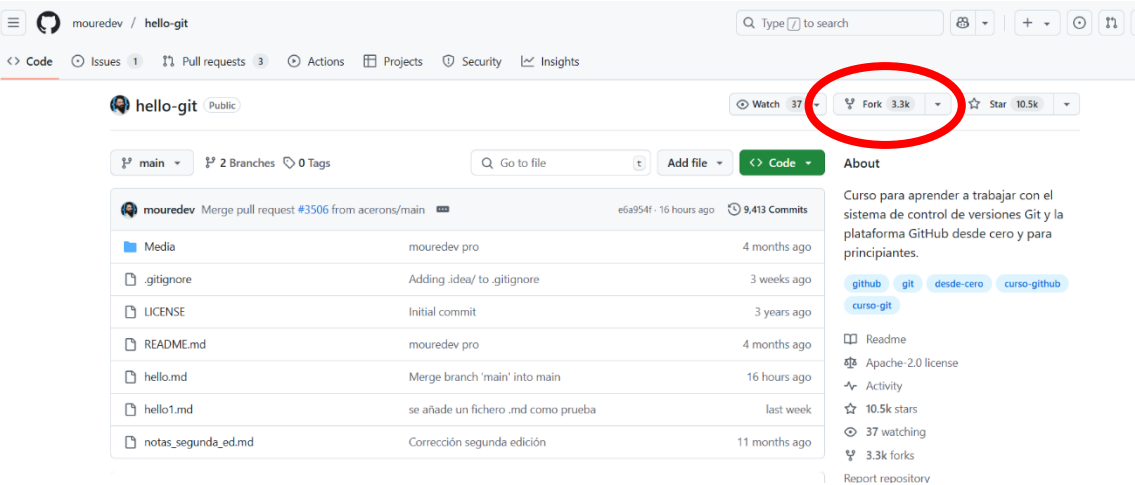
Para tirar de cambios de un repositorio remoto lo hacemos con el comando **git pull**

¿Qué es un fork de repositorio?

Es una copia de un repositorio de un usuario diferente, que nos permite realizar cambios sin afectar el repositorio original. Dicha copia se realiza en nuestra cuenta.

¿Cómo crear un fork de un repositorio?

Ingresando al repositorio del usuario que queremos copiar, en la parte superior derecha



Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner *

JMALV22

Repository name *

hello-git

hello-git is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Curso para aprender a trabajar con el sistema de control de versiones Git y la plataforma GitHub desde cero y

☒ Copy the main branch only

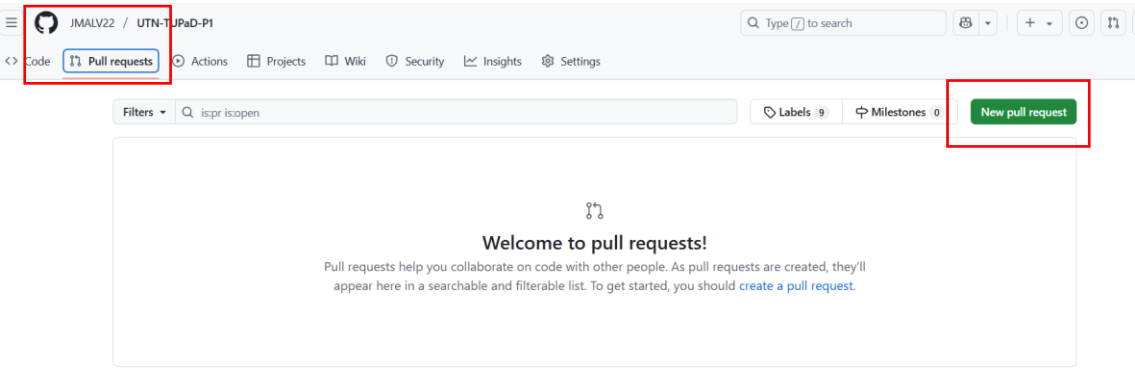
Contribute back to mouredev/hello-git by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

En este sector vamos a elegir un nombre con el que aparecerá en nuestra cuenta, podemos modificar la descripción y elegir la rama si es necesario, ponemos “Create fork” y listo. Hicimos un Fork de este repositorio del usuario elegido.

¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
El pull request nos ayuda a poder contribuir en un repositorio de otro usuario, para este aporte, se ingresa a GitHub, desde el repositorio el cual hicimos un fork, vamos a Pull request, presionamos en “New pull request” / “Create pull request”



En esta sección nos aparece desde donde se realiza el pull request, también se puede agregar un título y una descripción. Debajo se encuentran los commit con las modificaciones realizadas al proyecto.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base repository: sbruselar/UTN-TUPaD-P1

base: main

head repository: JMALV22/UTN-TUPaD-P1

compare: main

✓ Able to merge. These branches can be automatically merged.

Add a title

Title

Add a description

Write

Preview

H B I i e < > @ ↩ ↲

Add your description here...

Markdown is supported Paste, drop, or click to add files

Allow edits by maintainers

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

4 commits

3 files changed

1 contributor

Commits on Mar 24, 2025

Add files via upload

JMALV22 authored 3 days ago

Verified 4ee5e76

Se aareea archivo con TP-01

Verified b4b4a1e

Una vez enviado el pull request, queda sujeto a la aprobación y revisión del usuario que recibe el mismo.

¿Cómo aceptar una solicitud de extracción?

Las solicitudes de extracción nos aparecen en nuestra cuenta de GitHub, en la pestaña de “pull request”, donde podemos ver que se desea cambiar en el proyecto o el aporte que se realiza, nosotros podemos rechazar, aprobar o pedir que se realice alguna modificación al aporte.

¿Qué es una etiqueta en Git?

Es utilizado para etiquetar los commits, se suelen usar para referenciar puntos importantes que queremos guardar. Se puede ver cuando utilizamos por ejemplo el git log.

¿Cómo crear una etiqueta en Git?

Las etiquetas las creamos por medio del comando:

git tag <nombre del tag> <hash o referencia del commit>

si no le pones el hash o referencia del commit, el tag se inserta sobre la HEAD. Como buena práctica los tags se suelen poner en minúsculas.

¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta especifica mediante el comando **git push <etiqueta>** o podemos agregar todas las etiquetas con el comando **git push --tags**

¿Qué es un historial de Git?

El historial de Git es donde nos aparece todos los commits que venimos realizando, podemos ver como información el commit, autor y fecha. Git nos muestra en orden cronológico de forma decreciente.

¿Cómo ver el historial de Git?

Podemos ver el historial de git con el comando:

git log / este nos muestra el historial de commit de las ramas actuales.

git log --all / sumando el --all, poder ver el historial completo de todos los commit.

```
commit c10ad679cb76db9a8ce390d56de950d6ceefc2f5 (HEAD -> main, origin/main, origin/HEAD)
Author: JMALV22 <jm.alvarez@outlook.com.ar>
Date: Thu Mar 27 19:40:47 2025 -0300

    mi primer commit en repo clonado

commit 67c5bb1638dc6250be8306e001e452763808e807
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:22:04 2025 -0300

    Update README.MD

commit bdbdalc03d1eac806b95c5a4a816e20f9efad18c
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:19:38 2025 -0300

    Se agrega archivo con TP-01

commit 4ee5e763caa104565cce8728b460fae3fa11f6ea
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:17:30 2025 -0300

...skipping...
commit c10ad679cb76db9a8ce390d56de950d6ceefc2f5 (HEAD -> main, origin/main, origin/HEAD)
Author: JMALV22 <jm.alvarez@outlook.com.ar>
Date: Thu Mar 27 19:40:47 2025 -0300

    mi primer commit en repo clonado

commit 67c5bb1638dc6250be8306e001e452763808e807
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:22:04 2025 -0300

    Update README.MD

commit bdbdalc03d1eac806b95c5a4a816e20f9efad18c
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:19:38 2025 -0300

    Se agrega archivo con TP-01

commit 4ee5e763caa104565cce8728b460fae3fa11f6ea
Author: Jose Matias Alvarez <jm.alvarez@outlook.com.ar>
Date: Mon Mar 24 14:17:30 2025 -0300

    Add files via upload

commit c08d4db5d1200368ed91685f7bfb8a7f5ba28e96
Author: sbruselario <sbruselario@gmail.com>
Date: Mon Mar 17 22:06:33 2025 -0300

    ¡Bienvenid@ a Programación 1!
```

¿Cómo buscar en el historial de Git?

Las búsquedas en el historial las puedo hacer por:

git log - -grep = "palabra clave" / Palabra clave en el mensaje del commit.

git log - -author = "Nombre o email" / Autor específico

git show <hash_del_commit> / Un hash en específico

También, git nos permite buscar mediante muchas más opciones como, por ejemplo, fechas específicas, archivos, etc.

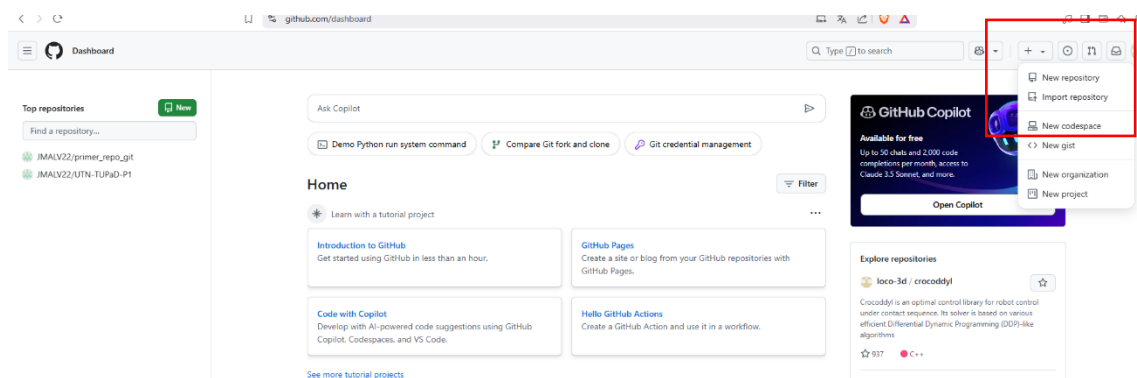
¿Cómo borrar el historial de Git?

La única forma de borrar el historial es eliminando la carpeta **.git**

¿Qué es un repositorio privado en GitHub?

Un repositorio privado es aquel que solo yo y las personas que autorice pueden ver el contenido de este.

¿Cómo crear un repositorio privado en GitHub?



Desde nuestro perfil, vamos al símbolo "+", en la parte superior derecha y vamos a nuevo repositorio.

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 *

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

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.](#)

☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

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

Nombre y descripción de repositorio

Aquí podemos asignar si es público o privado

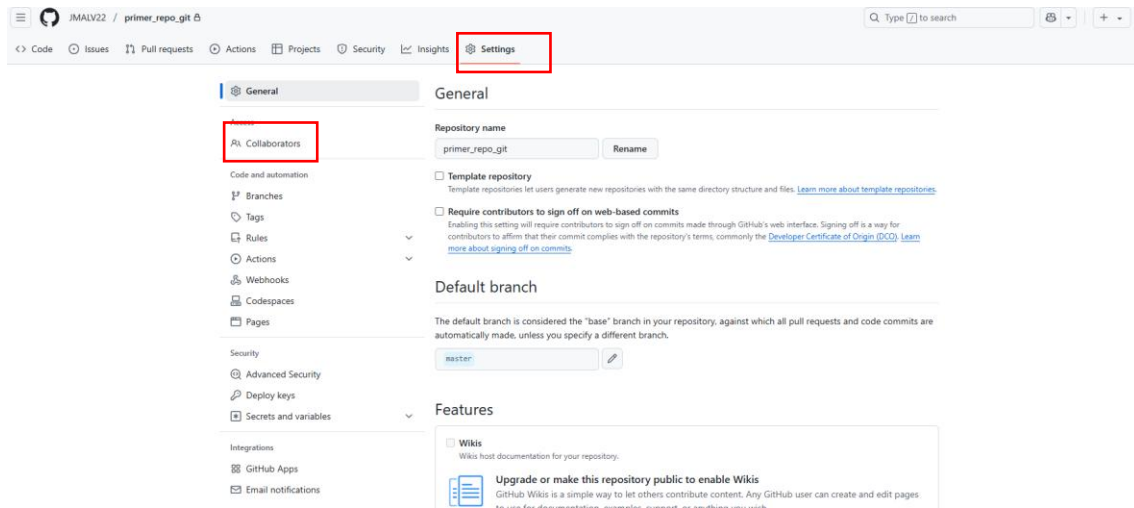
Nos da la opción de crear el repositorio con un README

Podemos agregar un .gitignore para descartar algunos archivos y elegir tipo de licencia

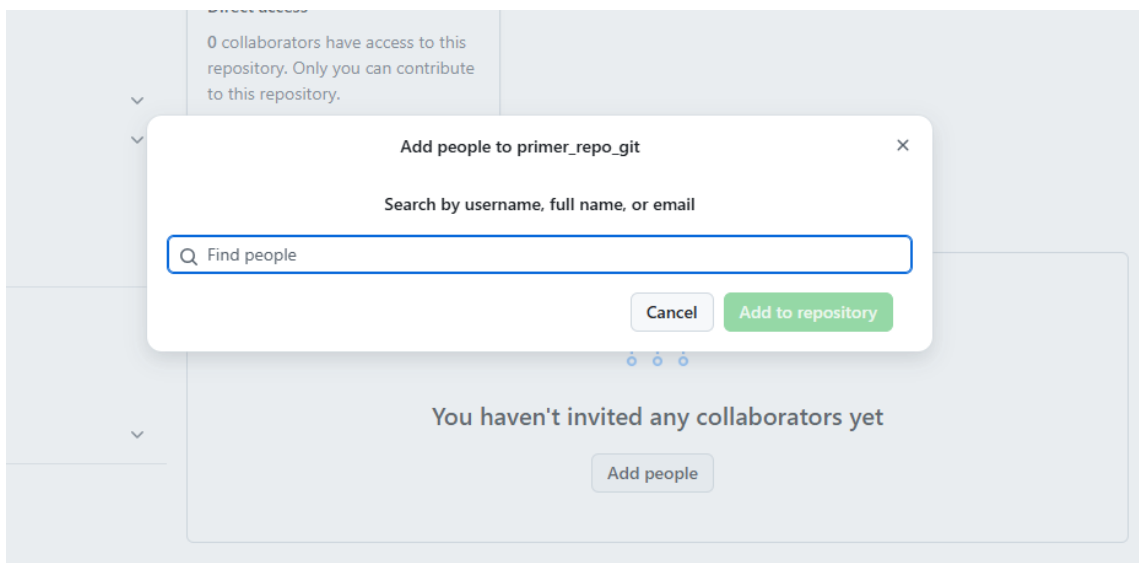
Le damos a "create Repository"

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

1. Se pide el nombre del usuario a la persona que vamos a invitar como colaboradora.
2. En el repositorio que vamos a necesitar colaboradores, vamos a “Settings”



3. En el menú de la izquierda, haz clic en Colaboradores. Nos va a pedir nuestra clave como seguridad.



4. Haz clic en Agregar personas.
5. Ingresamos el nombre de la persona.
6. Haga clic en agregar persona.
7. El usuario recibirá un correo electrónico invitándolo al repositorio. Una vez que acepte la invitación, tendrá acceso de colaborador a tu repositorio.

¿Qué es un repositorio público en GitHub?

Un repositorio público es aquel donde todo el mundo puede tener acceso.

¿Cómo crear un repositorio público en GitHub?

Un repositorio publico se crea de la misma forma en la se describe el privado, con la diferencia que seleccionamos la opción de público.

¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio publico solo tengo que enviar el link de este.

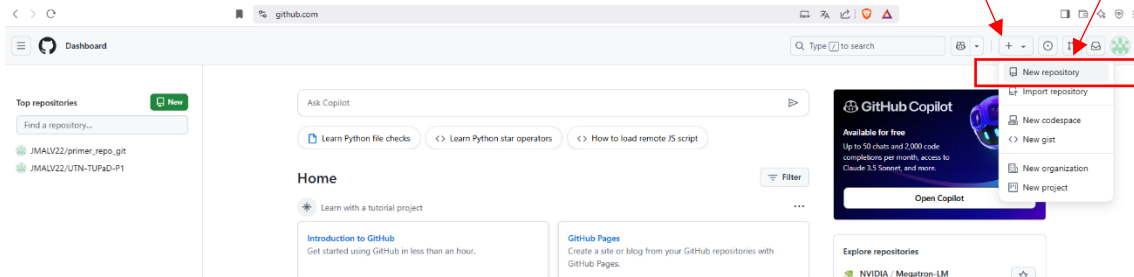
2) Realizar la siguiente actividad:

Crear un repositorio.

- o Dale un nombre al repositorio.
- o Elije el repositorio sea público.
- o Inicializa el repositorio con un archivo.

Presiono “+”

luego vamos a “New 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 *
<input type="text" value="JMALV22"/>	<input type="text" value="repo_para_tp2"/>

repo_para_tp2 is available.

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

Description (optional)

<input checked="" type="radio"/>	<input type="checkbox"/>	Public
		Anyone on the internet can see this repository. You choose who can commit.
<input type="radio"/>	<input type="checkbox"/>	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.](#)

Add .gitignore

.gitignore template:

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

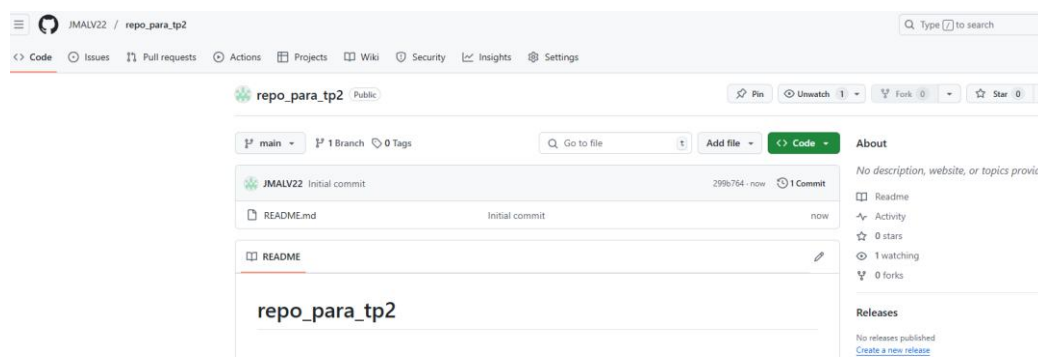
License:

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](#).

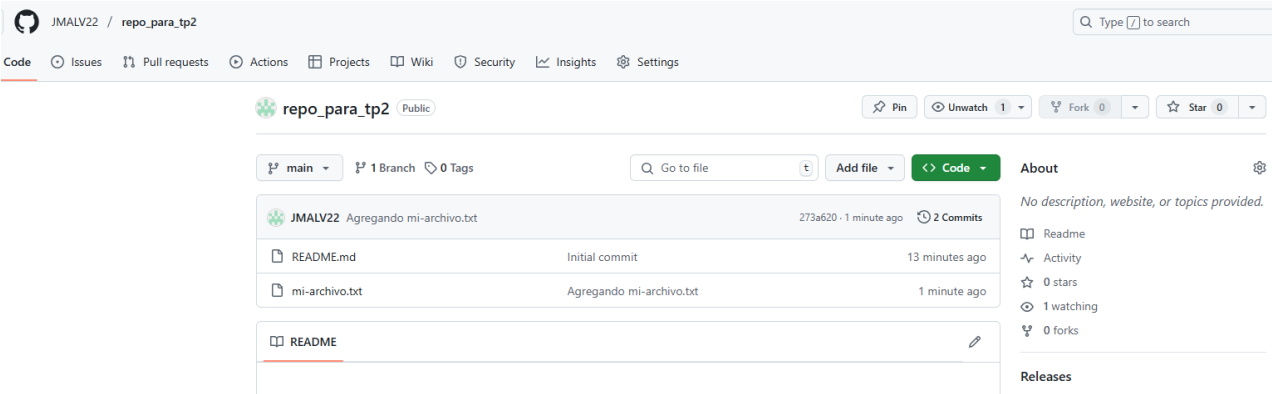
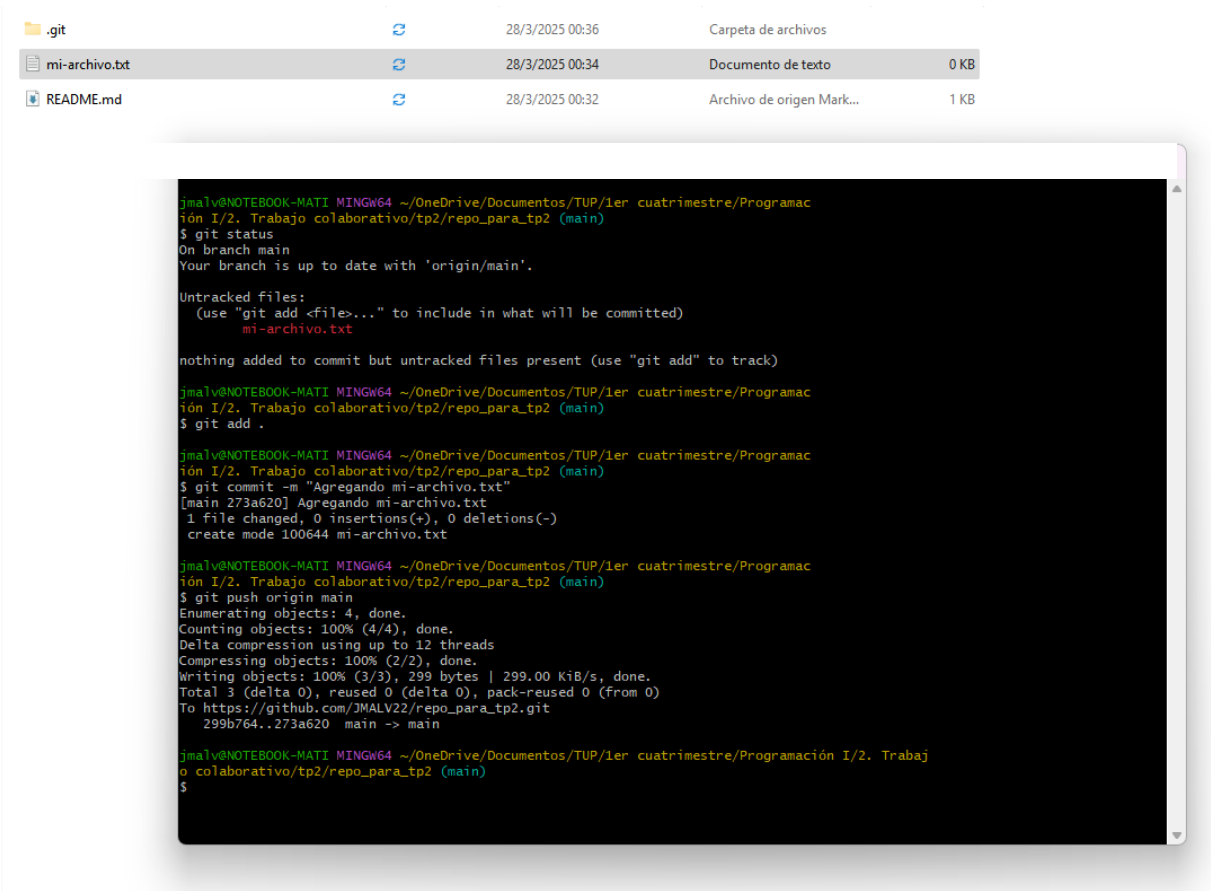
① You are creating a public repository in your personal account.

Create repository

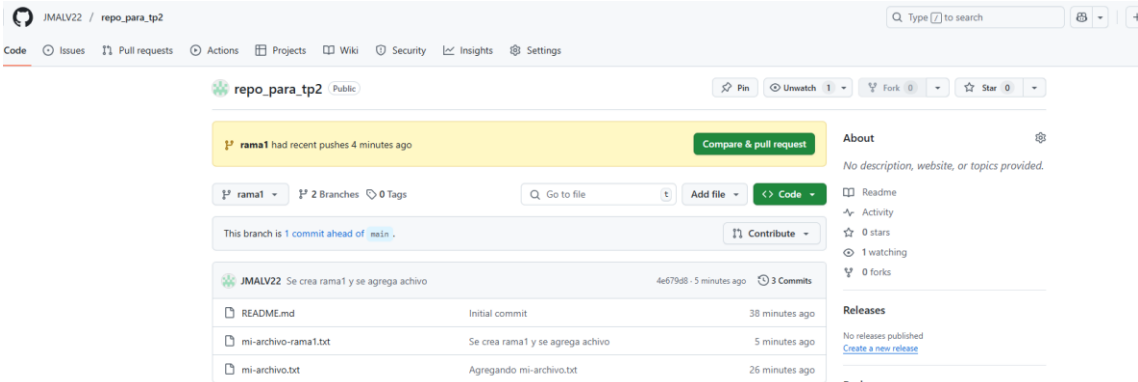


Agregando un Archivo

- 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).



o Realizar cambios o agregar un archivo



1) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

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 *

JMALV22

Repository name *

conflict-exercise

conflict-exercise is available.

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

Description (optional)

Repositorio para ejercicio numero 3 del TP2 de Programación I - TPU

☒ 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.](#)

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

JMALV22 / conflict-exercise

codeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

conflict-exercisePublic

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to file

Add fileCode

About

JMALV22 Initial commit1256d87 · now1 Commit

README.mdInitial commitnow

README

conflict-exercise

Repositorio para ejercicio numero 3 del TP2 de Programación I - TPU

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Paso 2: Clonar el repositorio a tu máquina local

JMALV22 / conflict-exercise

IssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

conflict-exercisePublic

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to file

Add fileCode

About

JMALV22 Initial commit

README.mdInitial commit

README

conflict-exercise

Repositorio para ejercicio numero 3 del TP2 de Programación I - TPU

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Local

Codespaces

Clone

HTTPSSSHGitHub CLICopy url to clipboard

https://github.com/JMALV22/conflict-exercise.git

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

Copiamos link para realizar el clonado desde la terminal

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
conflict-exercise		28/3/2025 01:12	Carpeta de archivos	
repo_para_tp2		28/3/2025 01:00	Carpeta de archivos	

ntos

5

Personal

35

)

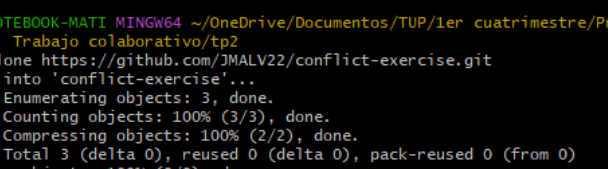
are

ip

Local (C:)


as

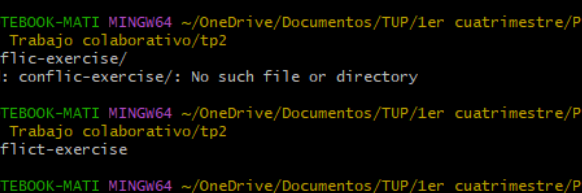
control



The screenshot shows a Windows terminal window with a dark background. The title bar at the top reads "MINGW64; c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Pro...". The terminal content shows the user cloning a repository from GitHub. The prompt is "jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2". The command entered is "git clone https://github.com/3MALV22/conflict-exercise.git". The output shows the cloning process: "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)", and "Receiving objects: 100% (3/3), done.". The prompt returns to "jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2" and the user has entered a dollar sign "\$" followed by a cursor.

```
MINGW64; c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Pro...  
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2  
$ git clone https://github.com/3MALV22/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.  
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2  
$ |
```

conflict-exercise		28/3/2025 01:12	Carpeta de archivos
repo_para_tp2		28/3/2025 01:00	Carpeta de archivos



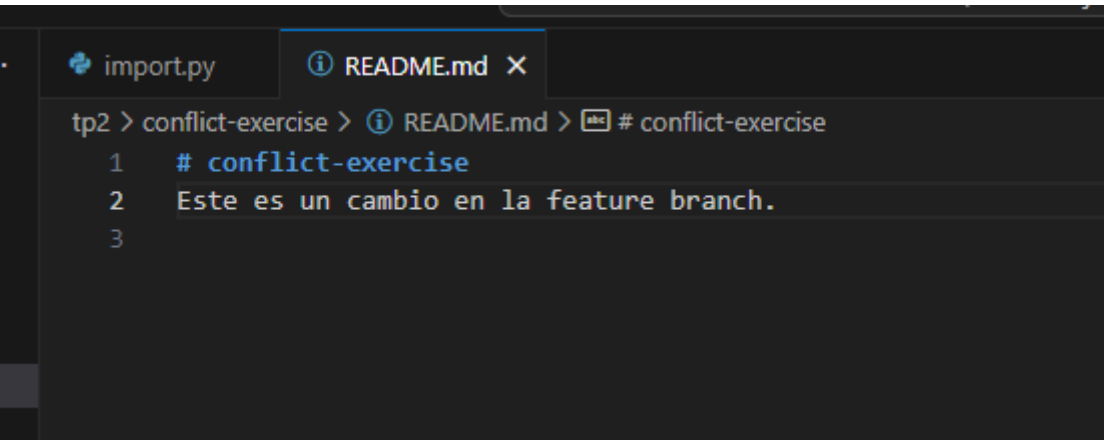
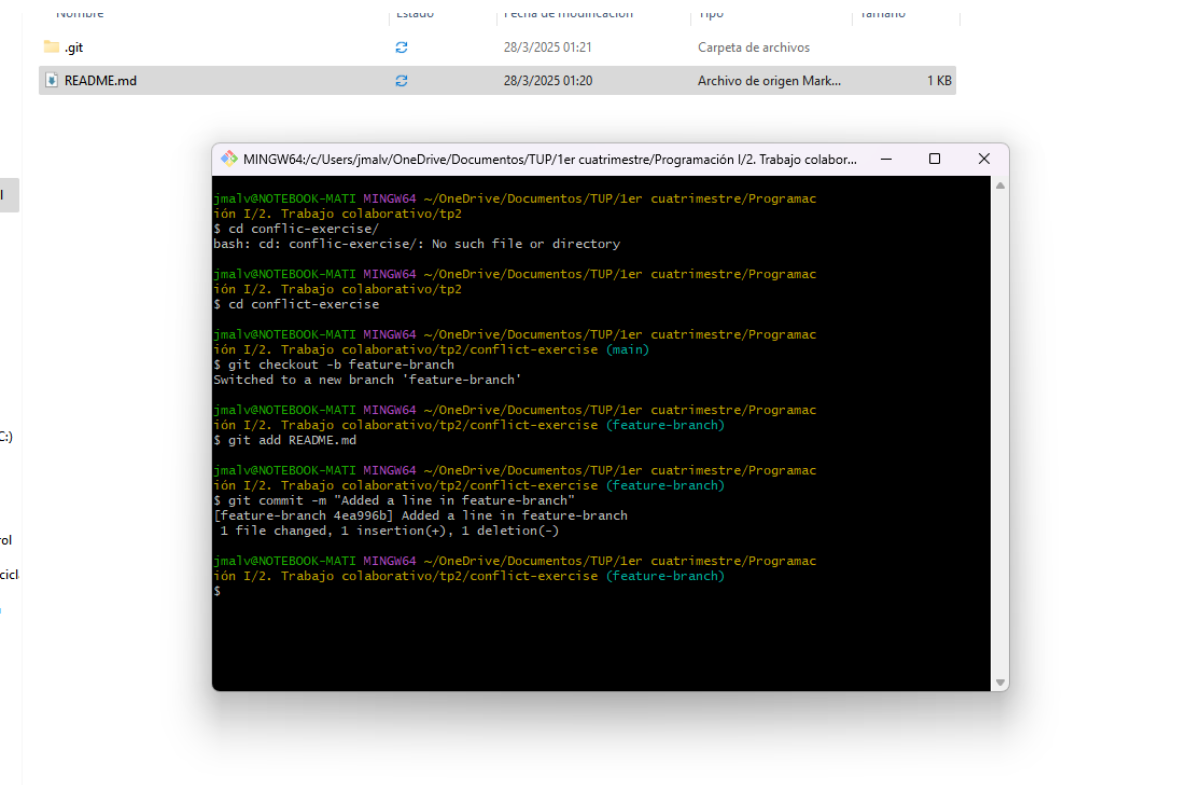
The screenshot shows a Windows terminal window with the title bar "MINGW64; c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Pro...". The terminal displays the following commands and output:

```
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programaci3n I/2. Trabajo colaborativo/tp2
$ cd conflict-exercise/
bash: cd: conflict-exercise/: No such file or directory

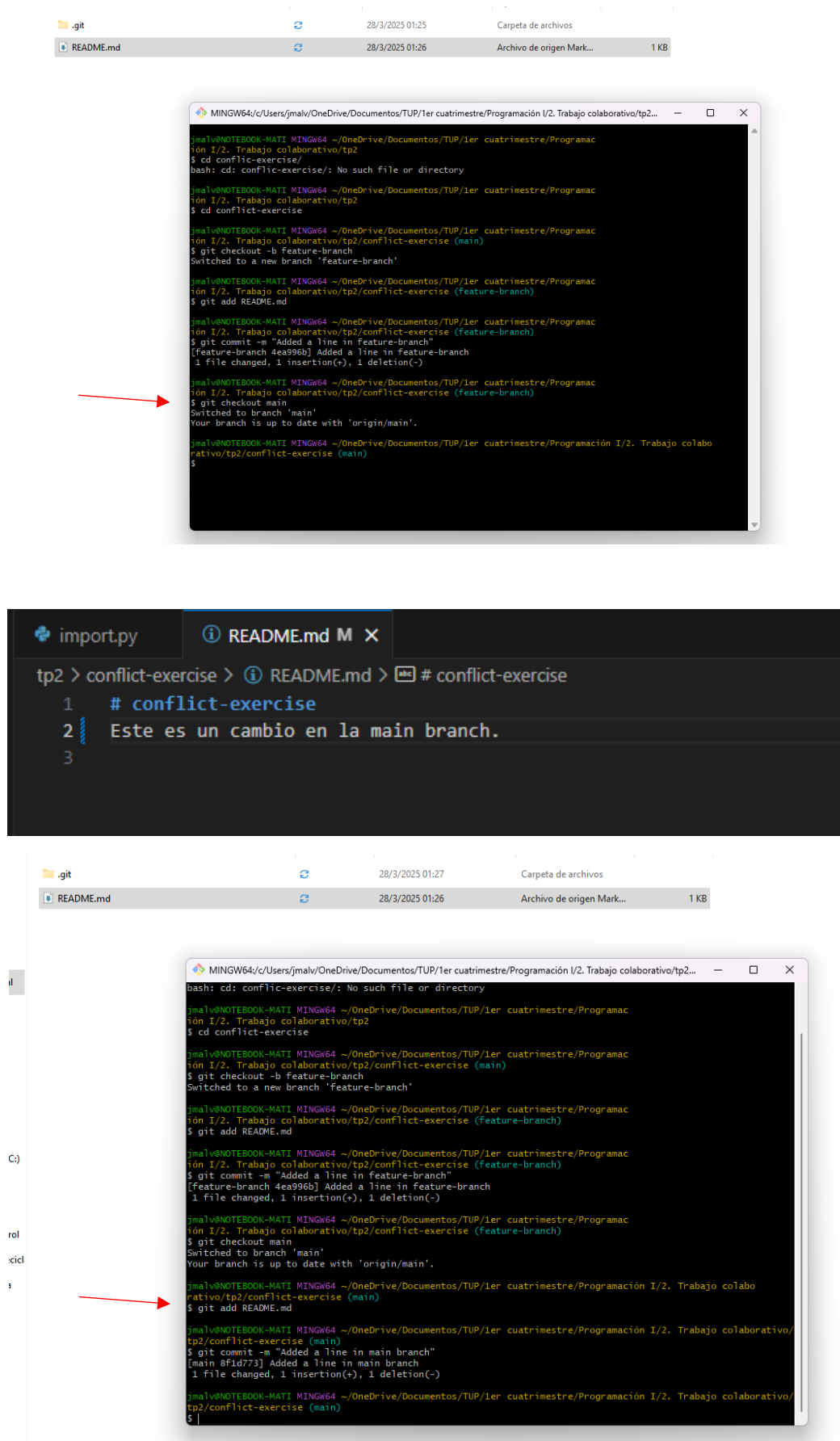
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programaci3n I/2. Trabajo colaborativo/tp2
$ cd conflict-exercise

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programaci3n I/2. Trabajo colaborativo/tp2/conflict-exercise (main)
$
```

Paso 3: Crear una nueva rama y editar un archivo



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



Paso 5: Hacer un merge y generar un conflicto

```
MINGW64/c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2...
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/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.

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main|MERGING)
$
```

Paso 6: Resolver el conflicto

```
tp2 > conflict-exercise > README.md > # <<<<<< HEAD Este es un cambio en la main branch.
1 # conflict-exercise
2 Acceptar cambio actual | Acceptar cambio entrante | Comparar cambios
3 <<<<<< HEAD (Cambio actual)
4 Este es un cambio en la main branch.
5 =====
6 Este es un cambio en la feature branch.
7 >>>>>> feature-branch (Cambio entrante)
```

Se decide aceptar cambio entrante.

```
tp2 > conflict-exercise > README.md > # conflict-exercise
1 # conflict-exercise
2 Este es un cambio en la feature branch.
3
```

```
MINGW64/c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2...
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/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.

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main|MERGING)
$ git add README.md

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main 5e5eab7] Resolved merge conflict

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main)
$
```

Paso 7: Subir los cambios a GitHub

```
MINGW64/c:/Users/jmalv/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2...
jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main)
$ 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), 740 bytes | 246.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/JMALV22/conflict-exercise.git
 f256fd7..5e5eab7 main -> main

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/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/JMALV22/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/JMALV22/conflict-exercise.git
 * [new branch]      feature-branch -> feature-branch

jmalv@NOTEBOOK-MATI MINGW64 ~/OneDrive/Documentos/TUP/1er cuatrimestre/Programación I/2. Trabajo colaborativo/tp2/conflict-exercise (main)
$
```

Paso 8: Verificar en GitHub

conflict-exercise / README.md

JMALV22 Added a line in feature-branch

Preview

Code

Blame

2 lines (2 loc) · 60 Bytes

Code 55% faster with GitHub Copilot

conflict-exercise

Este es un cambio en la feature branch.

Commits

main

All users

All time

Commits on Mar 28, 2025

Resolved merge conflict

JMALV22 committed 3 minutes ago

5e5eab7

Added a line in main branch

JMALV22 committed 9 minutes ago

8f1d773

Added a line in feature-branch

JMALV22 committed 16 minutes ago

4ea996b

Initial commit

JMALV22 authored 28 minutes ago

Verified f256fd7