

Práctico 2: Git y GitHub

Objetivo:

El estudiante desarrollará competencias para trabajar con Git y GitHub, aplicando conceptos fundamentales de control de versiones, colaboración en proyectos y resolución de conflictos, en un entorno simulado y guiado.

Resultados de aprendizaje:

1. Comprender los conceptos básicos de Git y GitHub: Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.
2. Manejar comandos esenciales de Git: Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.
3. Aplicar técnicas de colaboración en GitHub: Configurar y utilizar repositorios remotos, realizar forks, y gestionar pull requests para facilitar el trabajo colaborativo.
4. Resolver conflictos en un entorno de control de versiones: Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.

Actividades

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

1) ¿Qué es GitHub?

GitHub es una plataforma en línea diseñada para alojar, gestionar y compartir código de aplicaciones, herramientas y software. Esta herramienta permite a los desarrolladores subir proyectos, colaborar con otros programadores, descargar código de la comunidad y acceder a perfiles de diferentes usuarios. Gracias a su integración con Git, ofrece control de versiones y facilita la colaboración en proyectos de desarrollo de software.

2) ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, se debe iniciar sesión en la plataforma accediendo a <https://github.com> y usando una cuenta. Luego, el usuario debe hacer clic en el ícono "+" ubicado en la esquina superior derecha y seleccionar "New repository". A continuación, se ingresa un nombre para el repositorio, una descripción opcional y se elige si será público o privado. Finalmente, para completar esta acción, se debe hacer clic en el botón "Create repository".

3) ¿Cómo crear una rama en Git?

Para crear una nueva rama, el usuario debe abrir la terminal y ubicarse en el repositorio correspondiente. Para crear la nueva rama, vamos a usar el siguiente comando, ***git branch nombre-de-la-rama***.

4) ¿Cómo cambiar a una rama en Git?

Para listar las ramas disponibles en el repositorio y verificar cuál es la actual, se utiliza el siguiente comando, ***git branch***. Si el usuario desea cambiarse a una nueva rama, para realizar esta acción, vamos a usar el siguiente comando, ***git checkout nombre-de-la-rama***.

5) ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero es necesario ubicarse en la rama destino. Para realizar esta acción, vamos a usar el siguiente comando, ***git checkout nombre-de-la-rama-destino***. Una vez posicionado, se pueden integrar los cambios de otra rama utilizando el siguiente comando, ***git merge nombre-de-la-rama-a-fusionar***. En caso de que surjan conflictos, el usuario debe abrir los archivos afectados, resolver los conflictos manualmente y guardar los cambios. Para realizar esta acción, vamos a usar el siguiente comando, ***git add***. seguido de ***git commit -m "Resolviendo conflictos de merge"***. Finalmente, para verificar que la fusión se haya realizado correctamente, se utiliza el siguiente comando, ***git log --oneline --graph***.

6) ¿Cómo crear un commit en Git?

Para crear un commit en Git, primero se debe navegar al directorio del repositorio con la terminal y verificar el estado de los archivos pendientes. Para realizar esta acción, vamos a usar el siguiente comando, ***git***

status. Luego, para agregar todos los archivos modificados al área de preparación, se utiliza el siguiente comando, **git add ..** Si solo se desea agregar un archivo específico, se usa el siguiente comando, **git add nombre-del-archivo**. Para guardar los cambios en el repositorio, se debe crear un commit usando el siguiente comando, **git commit -m "Mensaje breve y descriptivo"**. Por último, para confirmar que el commit fue registrado, se puede consultar el historial de cambios utilizando el siguiente comando, **git log --oneline**.

7) ¿Cómo enviar un commit a GitHub?

Para enviar un commit a GitHub, primero se debe asegurar que los cambios hayan sido guardados en el repositorio local. Luego, para realizar esta acción, vamos a usar el siguiente comando, **git push origin nombre-de-la-rama**.

8) ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de un repositorio alojada en un servidor externo, como GitHub, que facilita la colaboración entre desarrolladores al permitir compartir código y cambios realizados en el proyecto.

9) ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git, el usuario debe vincular la URL del repositorio remoto con el proyecto local. Para realizar esta acción, vamos a usar el siguiente comando, **git remote add origin URL-del-repositorio**.

10) ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios desde el repositorio local al remoto, primero se deben confirmar los cambios realizados. Para realizar esta acción, vamos a usar el siguiente comando, **git push origin nombre-de-la-rama**.

11) ¿Cómo tirar de cambios de un repositorio remoto?

Para traer cambios del repositorio remoto al local, se debe estar ubicado en la rama correspondiente. Para realizar esta acción, vamos a usar el siguiente comando, **git pull origin nombre-de-la-rama**.

12) ¿Qué es un fork de repositorio?

Un fork es una copia independiente de un repositorio en GitHub que permite modificar un proyecto sin afectar al original, facilitando la colaboración en proyectos abiertos.

13) ¿Cómo crear un fork de un repositorio?

Para crear un fork, el usuario debe acceder al repositorio en GitHub y hacer clic en el botón "Fork" ubicado en la parte superior derecha. Finalmente, para completar esta acción, debe esperar a que el fork se cree en la cuenta del usuario.

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

Para enviar una solicitud de extracción, primero se deben subir los cambios al repositorio. Luego, en la pestaña "Pull requests", se hace clic en "New pull request". Finalmente, para completar esta acción, el usuario debe elegir las ramas a comparar y presionar "Create pull request" si todo es correcto.

15) ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción, el usuario debe acceder a la sección "Pull requests" y revisar los cambios realizados. Luego, para completar esta acción, debe hacer clic en "Merge pull request" y confirmar la fusión.

16) ¿Qué es una etiqueta en Git?

Una etiqueta es un marcador utilizado para señalar versiones específicas del código, como lanzamientos.

17) ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git, el usuario debe ejecutar el siguiente comando, **`git tag -a v1.0 -m "Versión 1.0"`**.

18) ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta al repositorio remoto, el usuario debe ejecutar el siguiente comando, **`git push origin v1.0`**.

19) ¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un repositorio.

20) ¿Cómo ver el historial de Git?

Para ver el historial de Git, el usuario debe ejecutar el siguiente comando, **`git log`**.

21) ¿Cómo buscar en el historial de Git?

Para buscar información en el historial de Git, el usuario debe ejecutar el siguiente comando, **`git log --grep "palabra clave"`**.

22) ¿Cómo borrar el historial de Git?

Para eliminar commits recientes, el usuario debe ejecutar el siguiente comando, **`git reset --hard <commit_id>`**.

23) ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo puede ser accedido por usuarios autorizados.

24) ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub, el usuario debe seleccionar la opción "Private" durante la configuración del repositorio.

25) ¿Cómo invitar a alguien a un repositorio privado?

Para invitar a alguien a un repositorio privado, el usuario debe acceder a la sección "Settings", dirigirse a "Manage access" y presionar "Invite a collaborator". Finalmente, para completar esta acción, debe ingresar el nombre de usuario y enviar la invitación.

26) ¿Qué es un repositorio público en GitHub?

Un repositorio público es visible y accesible para cualquier usuario en GitHub.

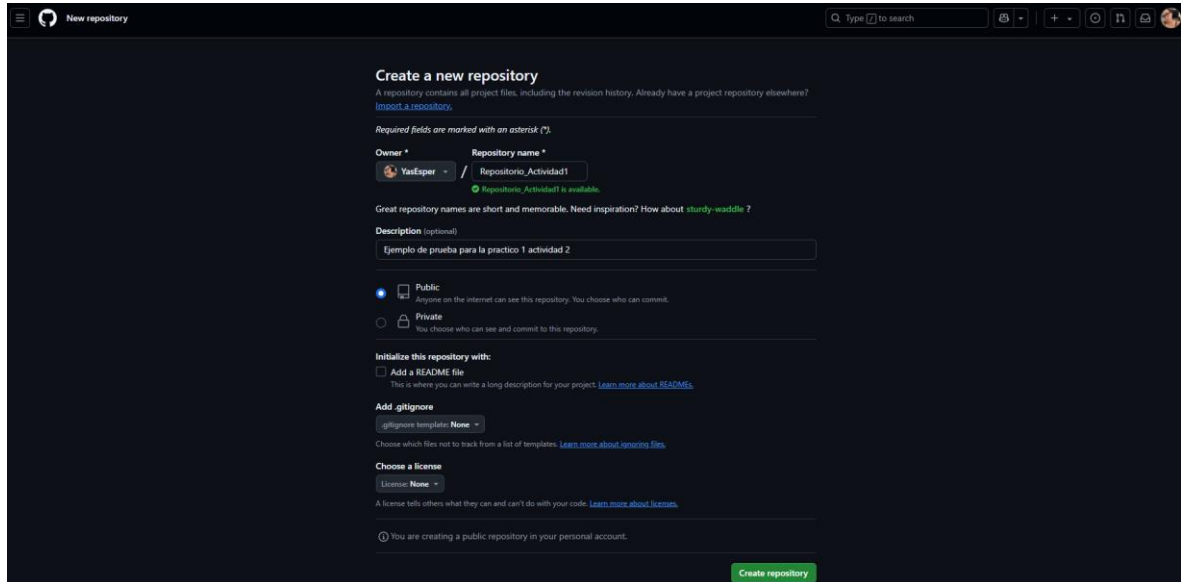
27) ¿Cómo compartir un repositorio público?

Para compartir un repositorio público, el usuario puede proporcionar el enlace directo o indicar que puede ser clonado usando el siguiente comando, **`git clone URL-del-repositorio`**.

A. Realizar la siguiente actividad:

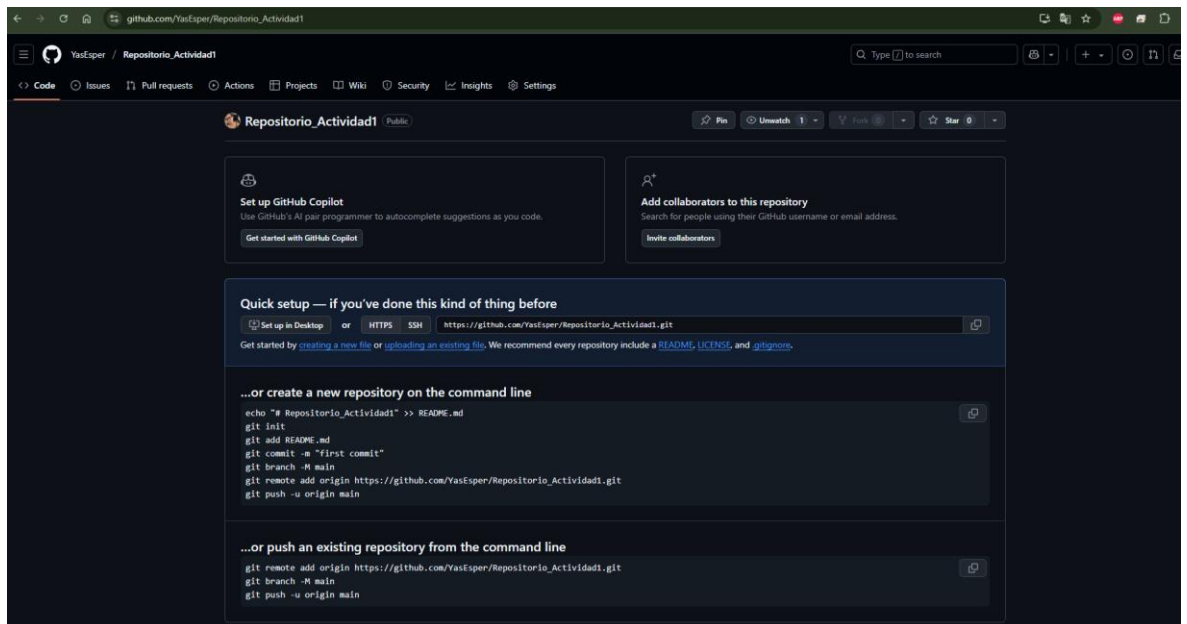
Crear un repositorio.

- Dale un nombre al repositorio.



The screenshot shows the 'Create a new repository' page on GitHub. The page is in dark mode. At the top, there's a search bar and navigation icons. The main heading is 'Create a new repository'. Below it, a note says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note below that says 'Required fields are marked with an asterisk (*)'. The 'Owner' field is set to 'YasEsper'. The 'Repository name' field is 'Repositorio_Actividad1', with a green checkmark indicating it's available. Below this is a 'Description (optional)' field with the text 'Ejemplo de prueba para la practico 1 actividad 2'. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there's a checkbox for 'Add a README file'. Below that is a section for 'Add gitignore' with a dropdown set to 'None'. There's also a 'Choose a license' section with a dropdown set to 'None'. At the bottom, a green button says 'Create repository'.

- Elije el repositorio sea público.



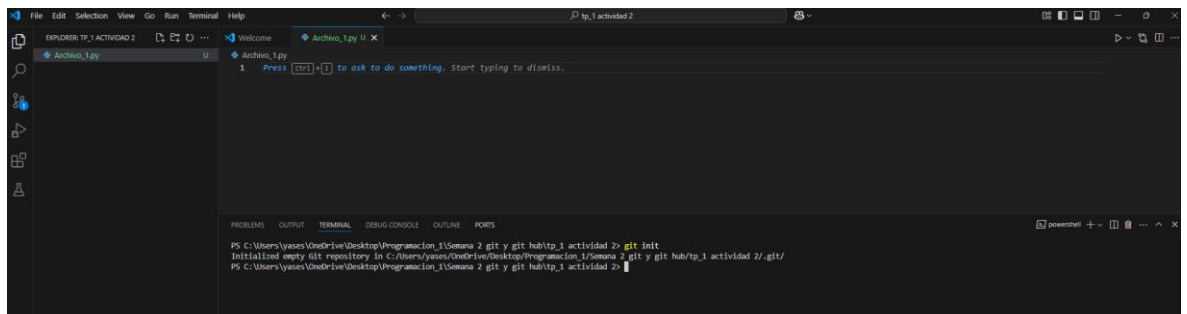
The screenshot shows the GitHub repository page for 'Repositorio_Actividad1'. The page is in dark mode. At the top, there's a search bar and navigation icons. The repository name 'Repositorio_Actividad1' is displayed with a 'Public' badge. Below the name, there are buttons for 'Pin', 'Unwatch', 'Fork', and 'Star'. There are two main cards: 'Set up GitHub Copilot' and 'Add collaborators to this repository'. Below these is a 'Quick setup' section with instructions for setting up the repository on the command line. The instructions include creating a new repository and pushing an existing repository. The code for creating a new repository is:

```
echo "# Repositorio_Actividad1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/YasEsper/Repositorio_Actividad1.git
git push -u origin main
```

 The code for pushing an existing repository is:

```
git remote add origin https://github.com/YasEsper/Repositorio_Actividad1.git
git branch -M main
git push -u origin main
```

- Inicializa el repositorio con un archivo.

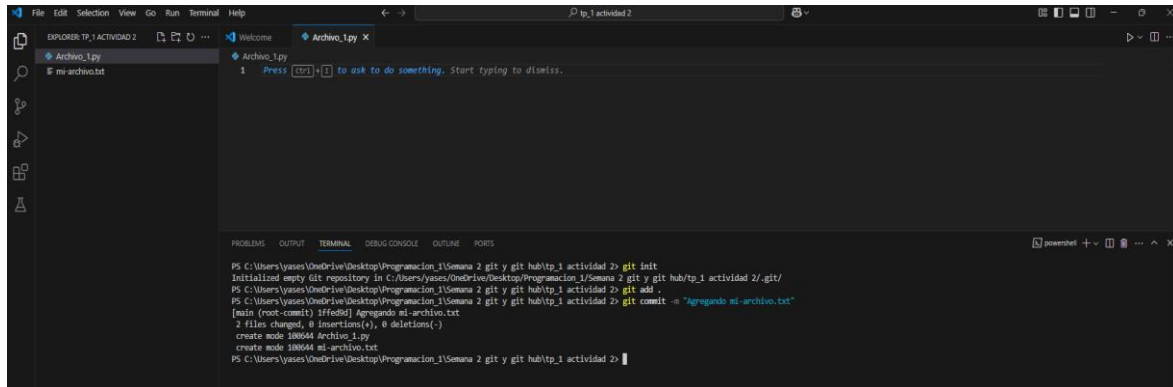


The screenshot shows a terminal window with the following commands and output:

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2> git y git hub tp_1 actividad 2> git init
Initialized empty Git repository in C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2> git y git hub tp_1 actividad 2> git/
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2> git y git hub tp_1 actividad 2>
```

Agregando un Archivo.

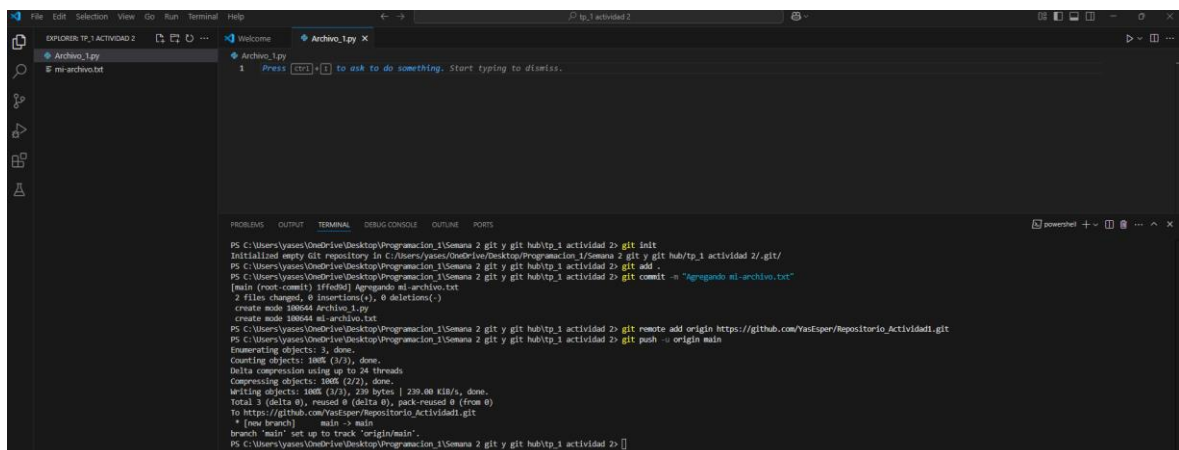
- Crea un archivo simple, por ejemplo, "mi-archivo.txt".
- Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

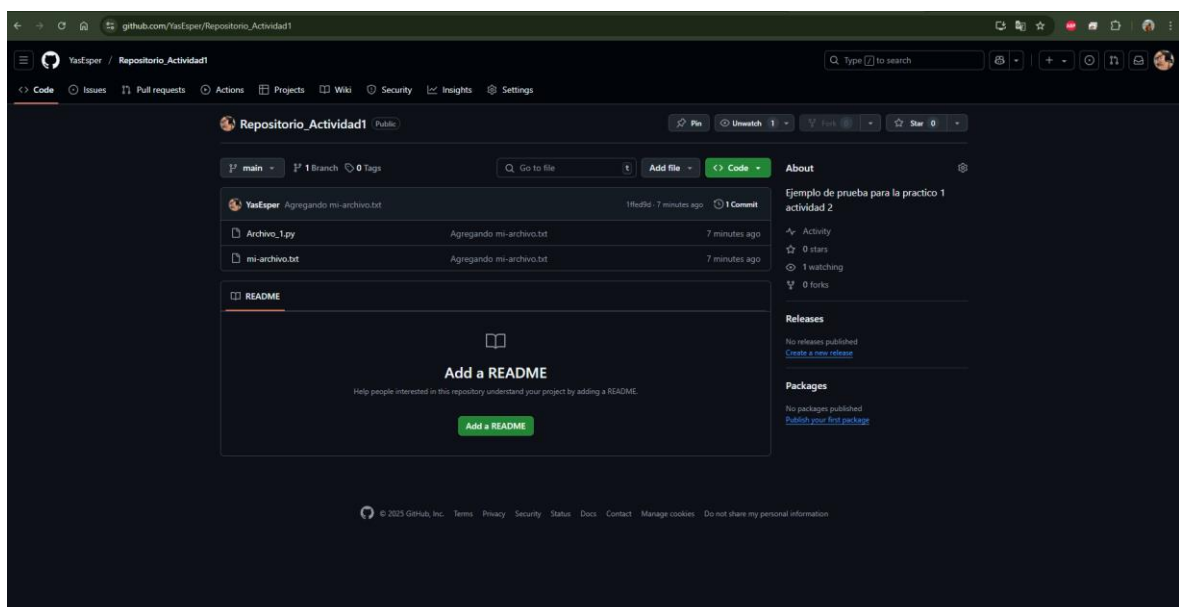
```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git init
Initialized empty Git repository in C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2\.git\
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git add .
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git commit -m "Agregando mi-archivo.txt"
[main (root-commit) 1f6fed6] Agregando mi-archivo.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Archivo_1.py
create mode 100644 mi-archivo.txt
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2>
```

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



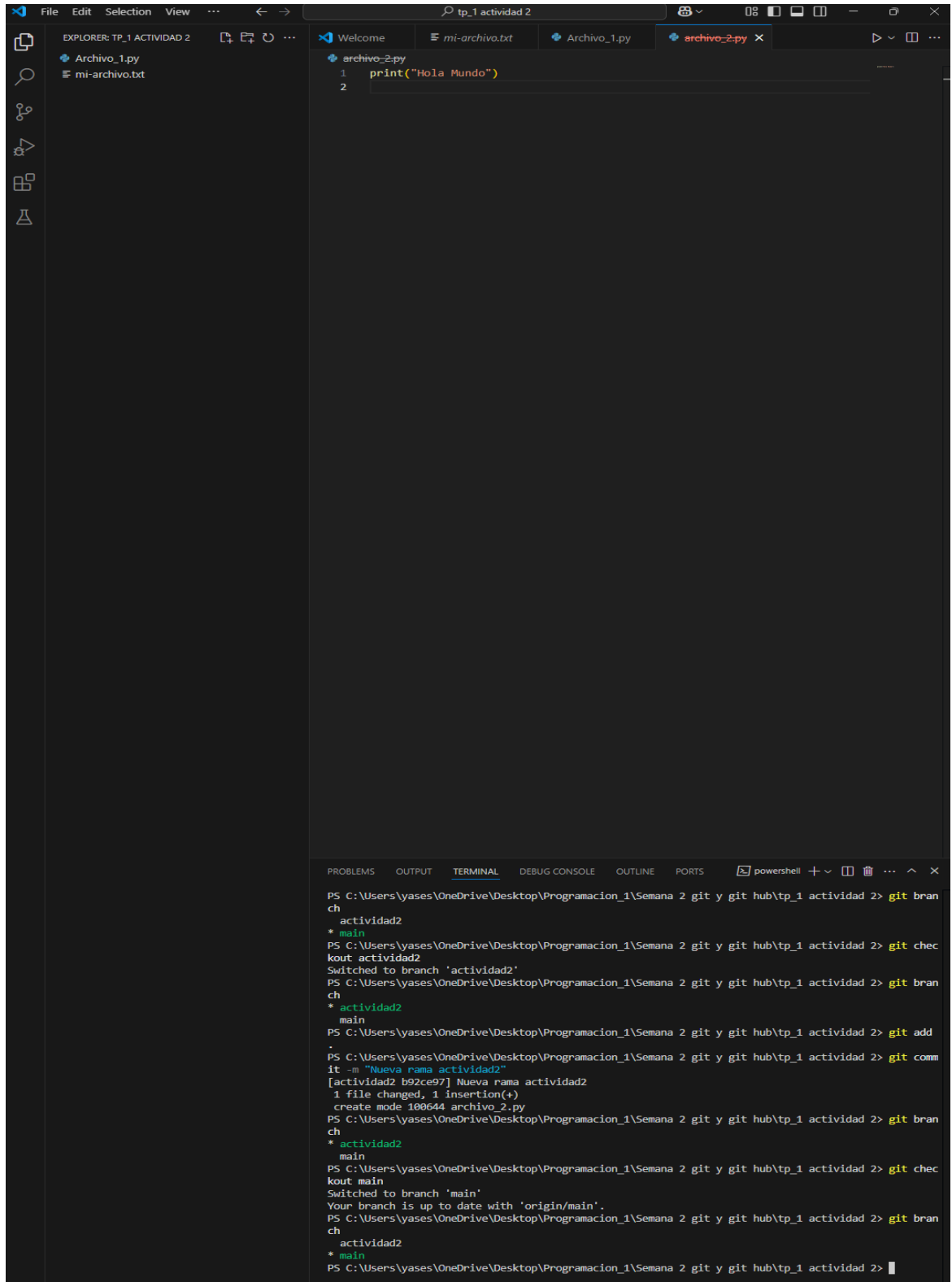
The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git init
Initialized empty Git repository in C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2\.git\
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git add .
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git commit -m "Agregando mi-archivo.txt"
[main (root-commit) 1f6fed6] Agregando mi-archivo.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Archivo_1.py
create mode 100644 mi-archivo.txt
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git remote add origin https://github.com/YasEsper/Repositorio_Actividad1.git
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 239 bytes | 239.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YasEsper/Repositorio_Actividad1.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2>
```



Creando Branchs.

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



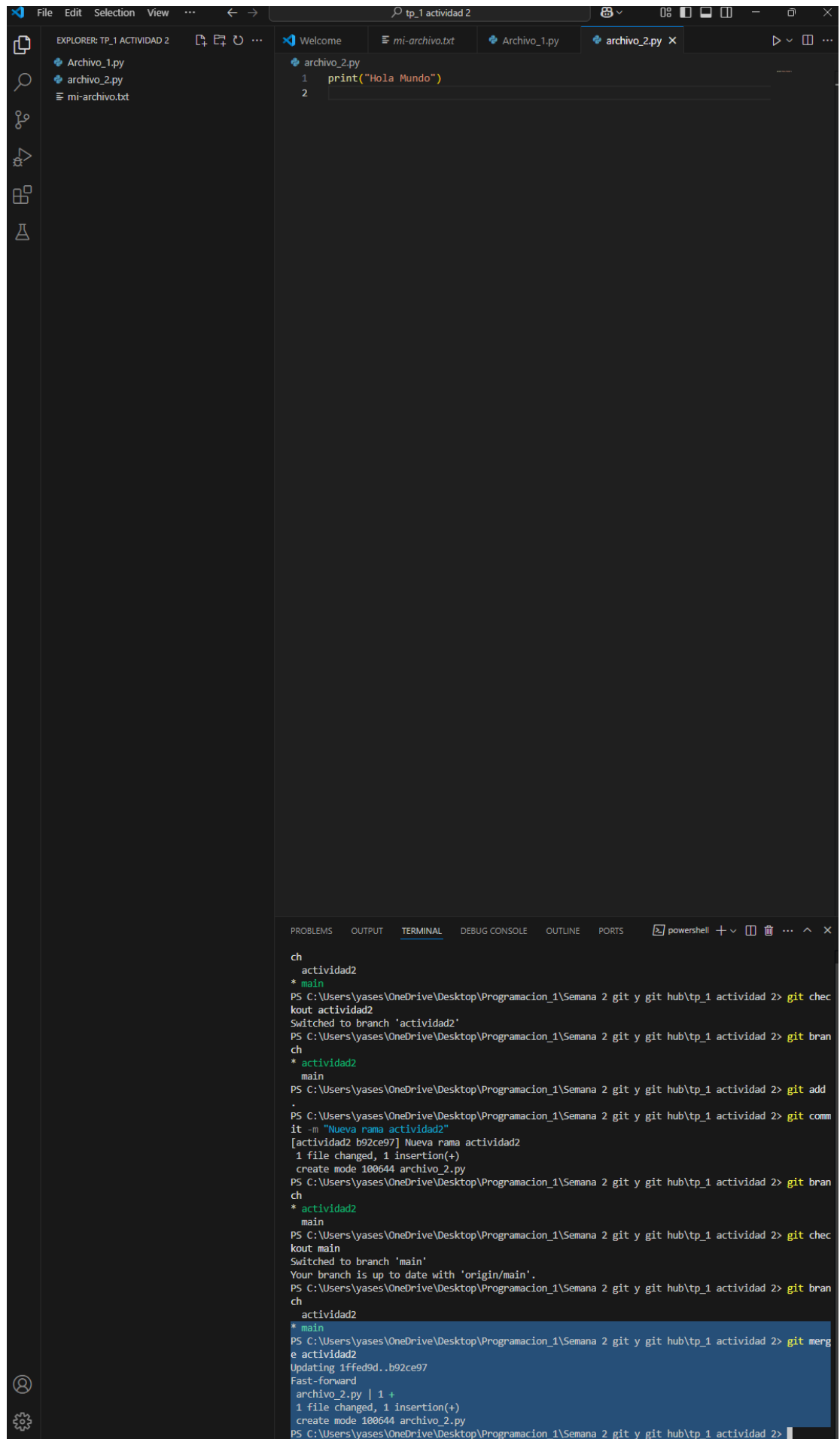
The screenshot shows the Visual Studio Code interface. The Explorer on the left shows a project named 'tp_1 actividad 2' with files 'Archivo_1.py' and 'mi-archivo.txt'. The main editor shows a file named 'archivo-2.py' with the following content:

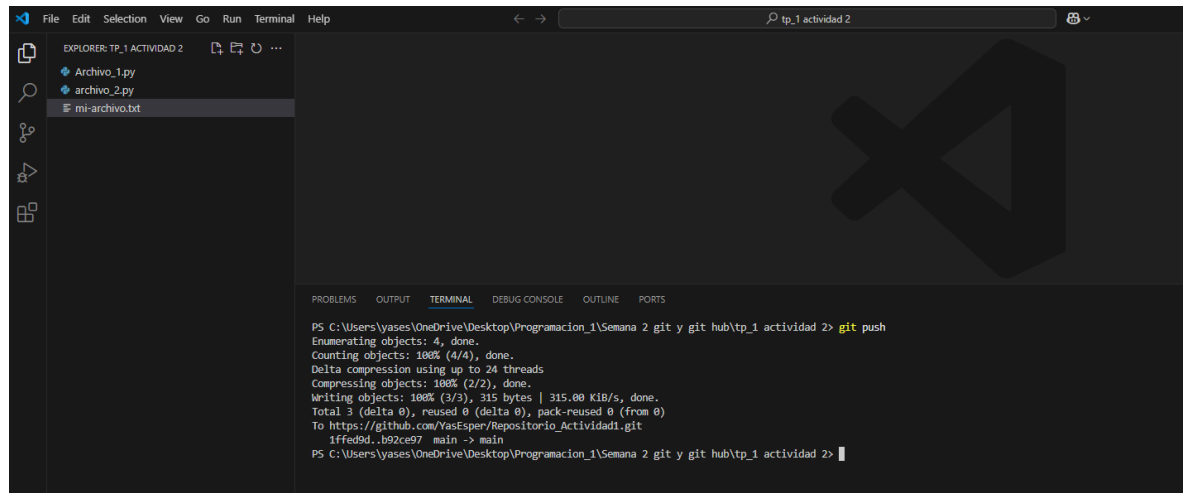
```
1 print("Hola Mundo")
2
```

The bottom panel shows the TERMINAL with the following output:

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git bran
ch
* main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git chec
kout actividad2
Switched to branch 'actividad2'
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git bran
ch
* actividad2
main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git add
.
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git comm
it -m "Nueva rama actividad2"
[actividad2 b92ce97] Nueva rama actividad2
1 file changed, 1 insertion(+)
create mode 100644 archivo_2.py
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git bran
ch
* actividad2
main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git chec
kout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git bran
ch
actividad2
* main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2>
```

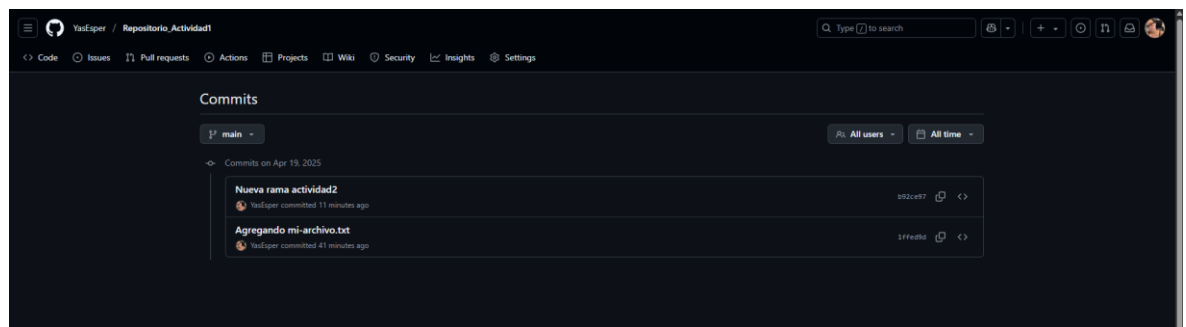
unificación de Ramas creadas





The screenshot shows the Visual Studio Code editor with a file explorer on the left containing 'Archivo_1.py', 'archivo_2.py', and 'mi-archivo.txt'. The terminal window at the bottom displays the output of a 'git push' command, showing object enumeration, compression, and writing progress. The command prompt shows the user is in the directory 'C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2'.

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 215 bytes | 315.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YasEper/Repositorio_Actividad1.git
1ffed9d..b92ce97 main -> main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Semana 2 git y git hub\tp_1 actividad 2>
```



B. Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub.

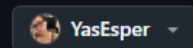
- Ve a GitHub e inicia sesión en tu cuenta.
- Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
- Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
- Opcionalmente, añade una descripción.
- Marca la opción "Initialize this repository with a README".
- Haz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

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 *

conflict-exercise

conflict-exercise is available.

Great repository names are short and memorable. Need inspiration? How about **potential-fishstick** ?

Description (optional)

Repositorio Tp 1 actividad 3

☒ 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.

YasEsper / conflict-exercise

Q Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

conflict-exercise (Public)

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file

Add file Code

YasEsper Initial commit 7691c · now 1 Commit

README.md Initial commit now

README

conflict-exercise

Repositorio Tp 1 actividad 3

About

Repositorio Tp 1 actividad 3

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

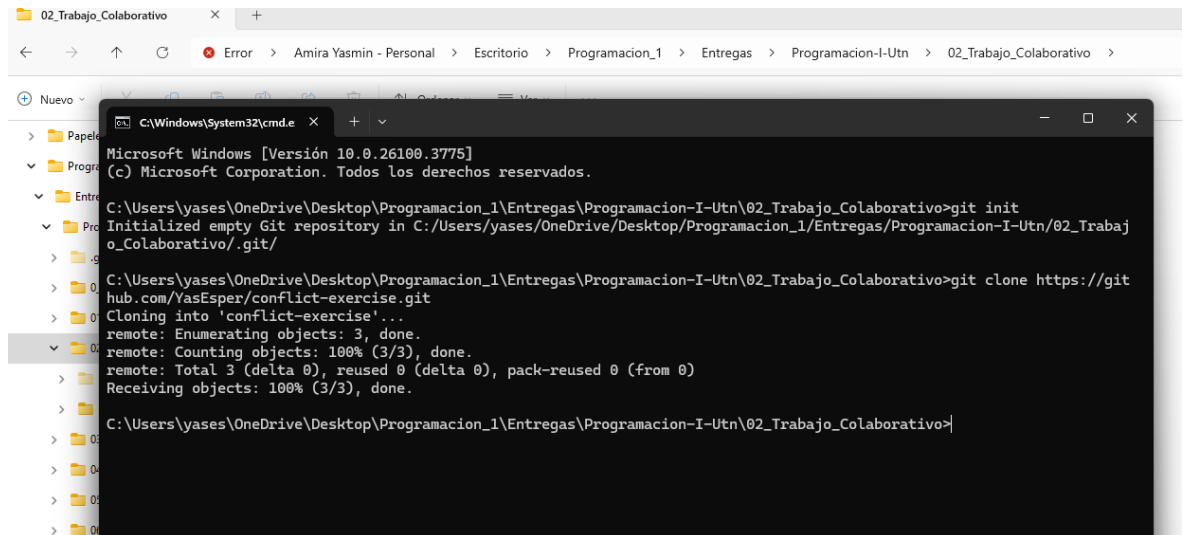
No packages published Publish your first package

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Paso 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>



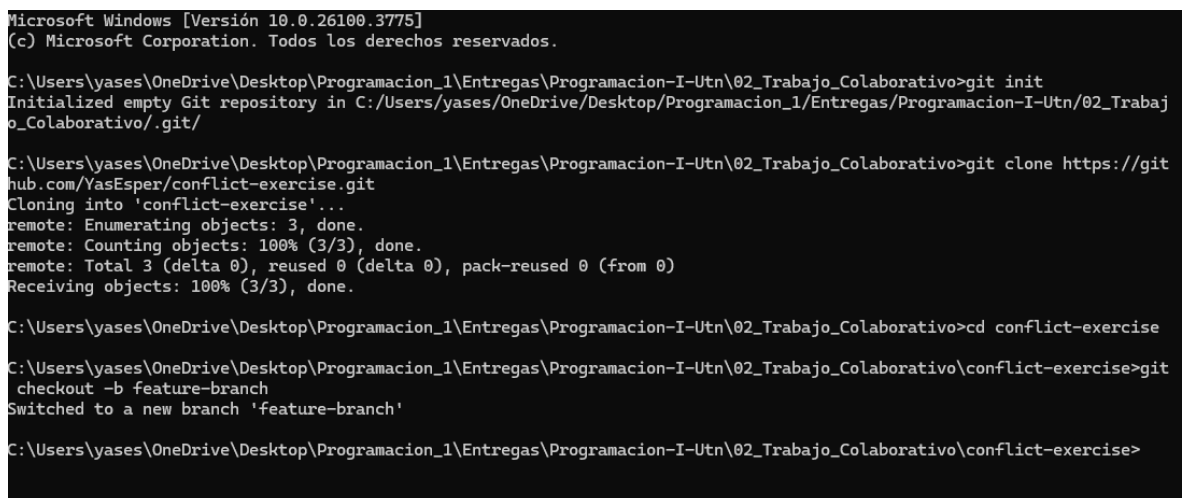
- Entra en el directorio del repositorio:

cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

- Crea una nueva rama llamada feature-branch:

git checkout -b feature-branch



- Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:

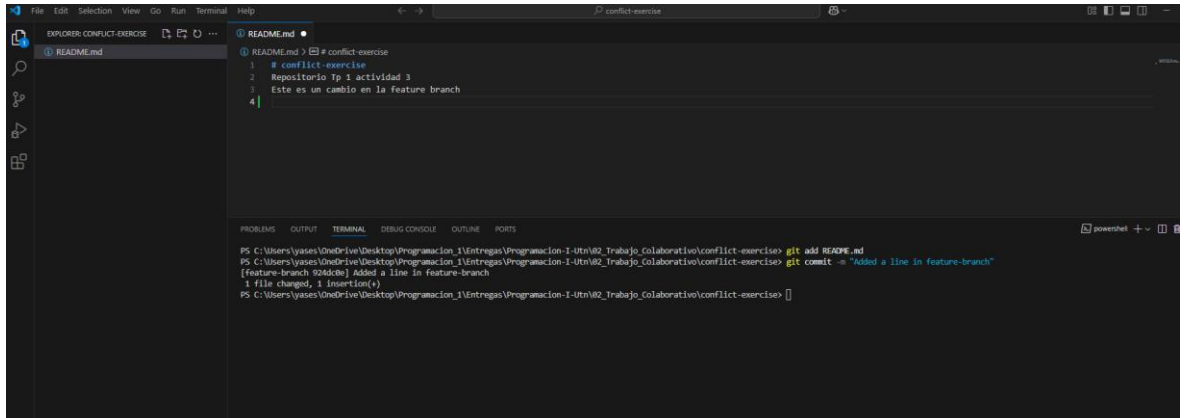
Este es un cambio en la feature branch.

git

- Guarda los cambios y haz un commit:

`git add README.md`

`git commit -m "Added a line in feature-branch"`



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a file named 'README.md'. The main editor area displays the content of 'README.md':

```
# conflict-exercise
1 # conflict-exercise
2 Repositorio Tp 1 actividad 3
3 Este es un cambio en la feature branch
4
```

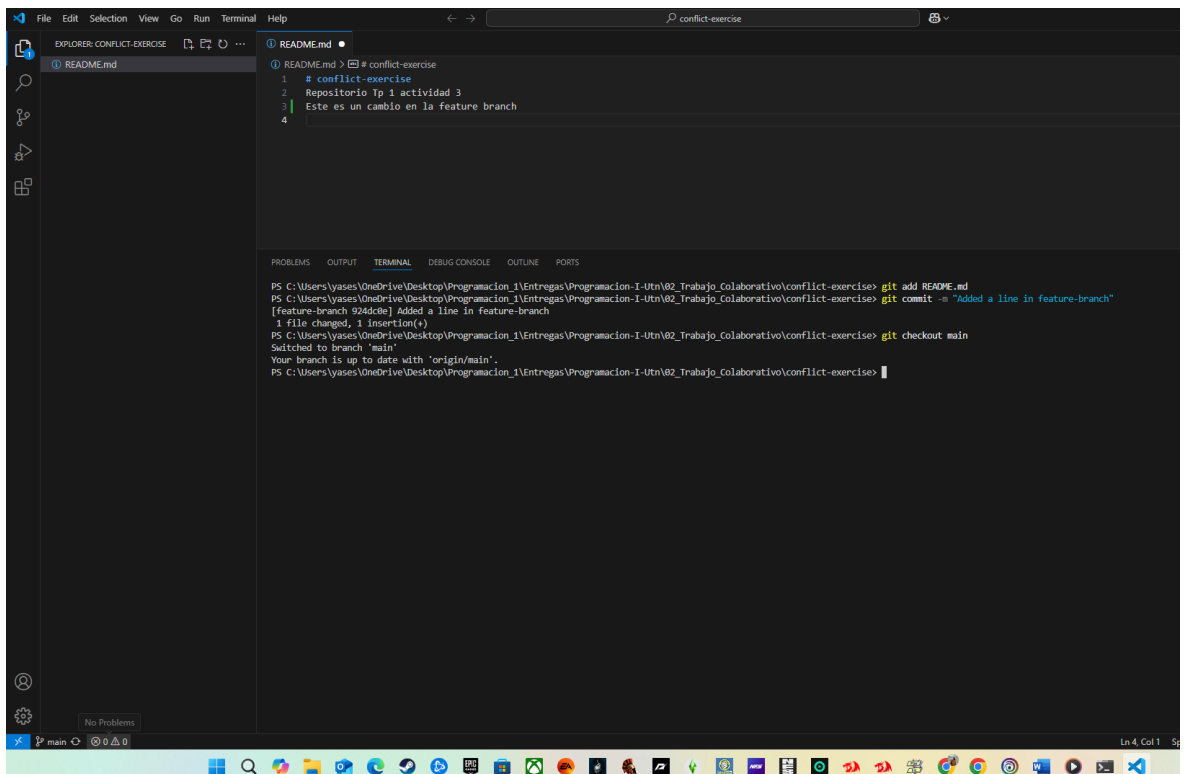
The terminal at the bottom shows the following commands and output:

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborativo\conflict-exercise> git add README.md
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborativo\conflict-exercise> git commit -m "Added a line in feature-branch"
[feature-branch 924dcbe] Added a line in feature-branch
1 file changed, 1 insertion(+)
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborativo\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`



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a file named 'README.md'. The main editor area displays the content of 'README.md':

```
# conflict-exercise
1 # conflict-exercise
2 Repositorio Tp 1 actividad 3
3 Este es un cambio en la feature branch
4
```

The terminal at the bottom shows the following commands and output:

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborativo\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborativo\conflict-exercise>
```

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

- Guarda los cambios y haz un commit:

`git add README.md`

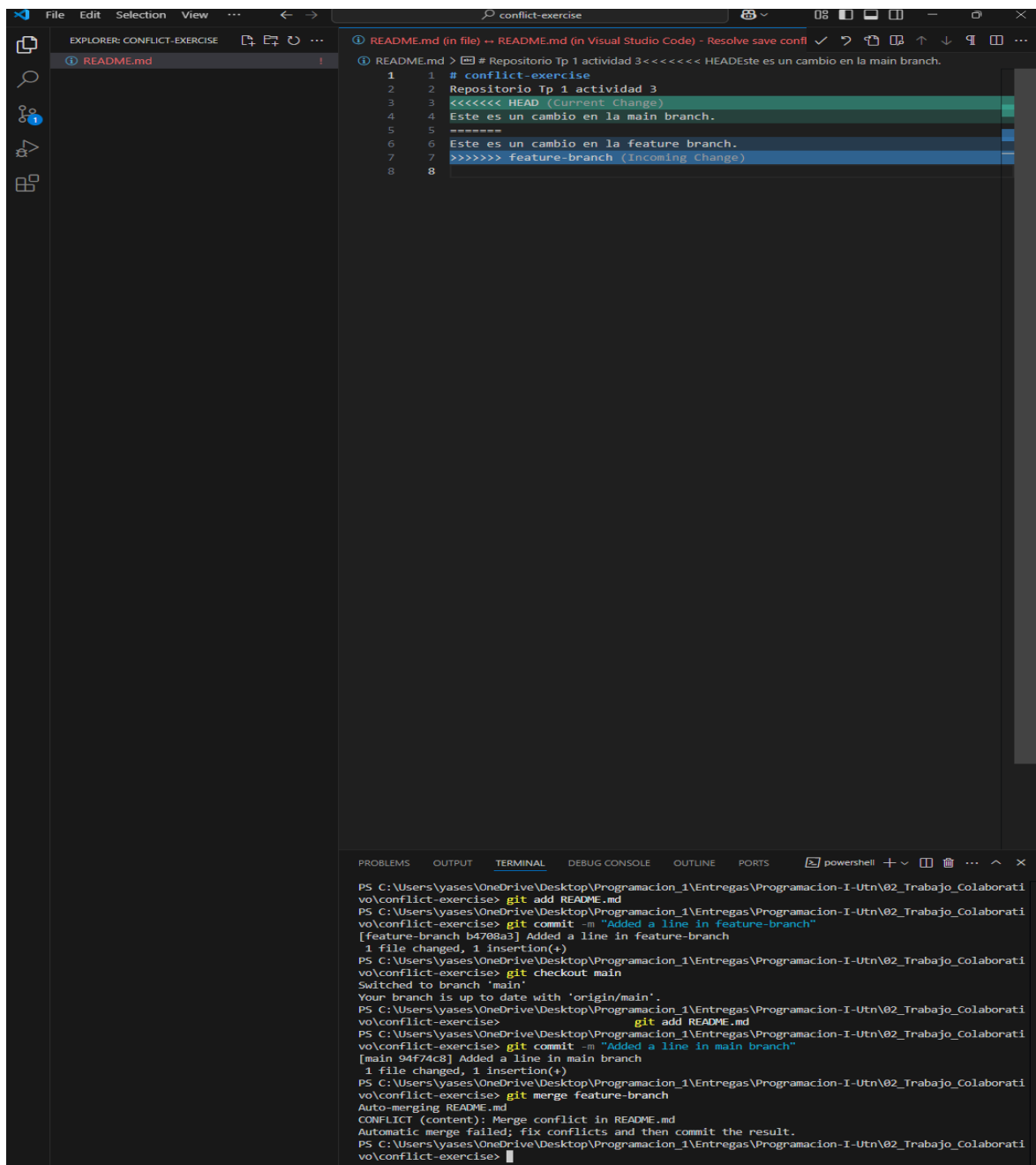
`git commit -m "Added a line in main branch"`

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.



```
1 1 # conflict-exercise
2 2 Repositorio Tp 1 actividad 3
3 3 <<<<<< HEAD (Current Change)
4 4 Este es un cambio en la main branch.
5 5 =====
6 6 Este es un cambio en la feature branch.
7 7 >>>>>> feature-branch (Incoming Change)
8 8
```

```
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git add README.md
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git commit -m "Added a line in feature-branch"
[feature-branch b4708a3] Added a line in feature-branch
1 file changed, 1 insertion(+)
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git add README.md
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git commit -m "Added a line in main branch"
[main 94f74cb] Added a line in main branch
1 file changed, 1 insertion(+)
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\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 C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\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
```

- 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).
- 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
```

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.

File Edit Selection View ... ← → conflict-exercise

SOURCE CONTROL

CHANGES

Message (Ctrl+Enter to commit on "main")

Commit

GRAPH

Resolved merge conflict Vases... main

Merge branch 'feature-branch' Vasesper

Added a line in feature-branch Vasesper

Added a line in main branch Vasesper

Initial commit Vasesper

README.md

```
1 # conflict-exercise
2 Repositorio Tp 1 actividad 3
3 Este es un cambio en la main branch.
4
```

Output (Ctrl+Shift+U)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE OUTLINE PORTS powershell

```
add : El término 'add' no se reconoce como nombre de un cmdlet, función, archivo de script o
programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de
acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 1
+ add README.md
+ ~~~~
+ CategoryInfo          : ObjectNotFound: (add:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git add README.md
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git commit -m "Resolved merge conflict"
[main 2eb4163] Resolved merge conflict
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 24 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 930 bytes | 930.00 KiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/YasEsper/conflict-exercise.git
 9ab5caf..2eb4163  main -> main
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\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/YasEsper/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/YasEsper/conflict-exercise.git
 * [new branch]   feature-branch -> feature-branch
PS C:\Users\yases\OneDrive\Desktop\Programacion_1\Entregas\Programacion-I-Utn\02_Trabajo_Colaborati
vo\conflict-exercise>
```

github.com/YasEper/conflict-exercise

YasEper / conflict-exercise

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

conflict-exercisePublic

PinUnwatch1ForkStar0

main2 Branches0 Tags

Go to fileAdd fileCode

YasEperResolved merge conflict2eb4163 · 2 minutes ago3 Commits

README.mdResolved merge conflict2 minutes ago

README

conflict-exercise

Repositorio Tp 1 actividad 3 Este es un cambio en la main branch.

Repositorio Tp 1 actividad 3

ReadmeActivity0 stars1 watching0 forks

No releases publishedCreate a new release

No packages publishedPublish your first package

© 2025 GitHub, Inc. TermsPrivacySecurityStatusDocsContactManage cookiesDo not share my personal information

YasEper / conflict-exercise

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Commits

main

All usersAll time

Commits on Apr 20, 2025

Resolved merge conflict

YasEper committed 2 minutes ago2eb4163

Merge branch 'feature-branch'

YasEper committed 4 minutes ago5afe9a0

Added a line in main branch

YasEper committed 11 minutes ago94f74c8

Added a line in feature-branch

YasEper committed 12 minutes agob4798a3

Commits on Apr 19, 2025

Initial commit

YasEper authored 18 minutes agoVerified9a85caf