

UNIDAD 2 - TRABAJO COLABORATIVO

Práctico 2: Git y GitHub

Actividades

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

- **¿Qué es GitHub?**

GitHub es una comunidad (sitio web), en donde los usuarios pueden subir sus repositorios, ya sea de forma pública o privado, y compartirlos con otros usuarios. De esta forma, pusheamos nuestro repositorio local a un repositorio remoto, que quedará almacenado en GitHub.

- **¿Cómo crear un repositorio en GitHub?**

Para crear un repositorio en GitHub, tenemos que:

- 1-crearnos un usuario en el sitio <https://github.com/>.
- 2-Una vez que se crea nuestro usuario, hacemos click en el botón New.
- 3-Cuando aparezca esta pantalla, ponemos un nombre en el campo "Repository name":
- 4-Podemos insertar una descripción en el campo de descripción.
- 5-Luego seleccionamos la opción Public o Private, según si queremos que nuestro repositorio pueda ser visto y consultado por otros usuarios (Public), o si deseamos limitado su acceso sólo a quién nosotros elijamos (Private) .

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 *

Floresia-study / repo_TP2

repo_TP2 is available.

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

Description (optional)

Repo creado para el TP 2

☒ 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:

6-Click en Create Repository.

https://github.com/new

Repo creado para el TP 2

☒ 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.](#)

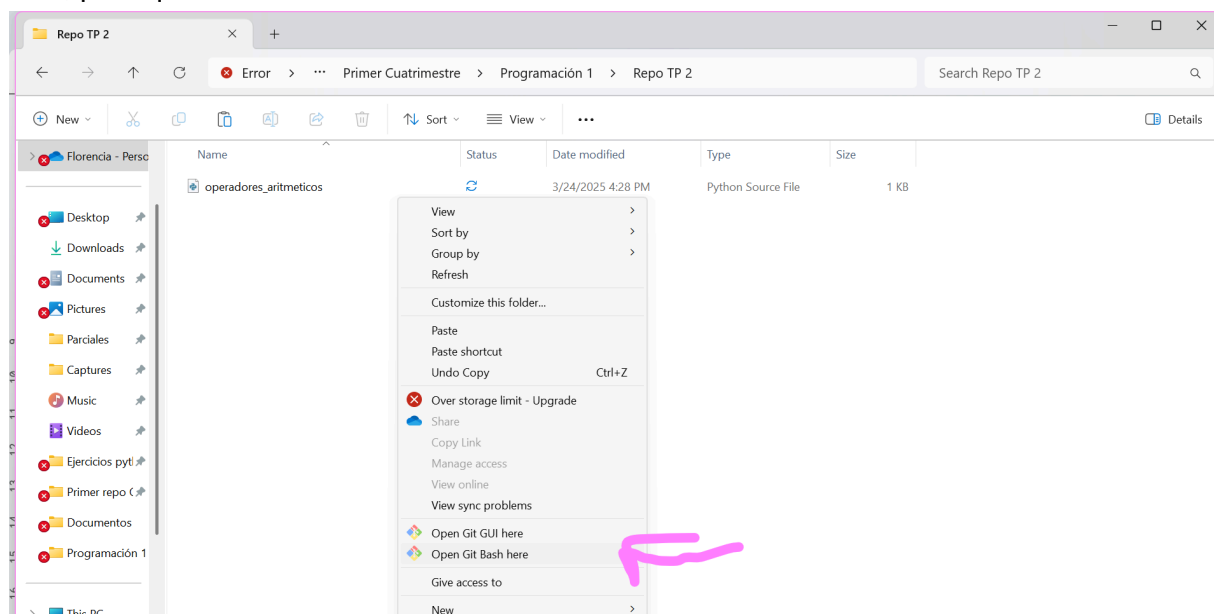
You are creating a public repository in your personal account.

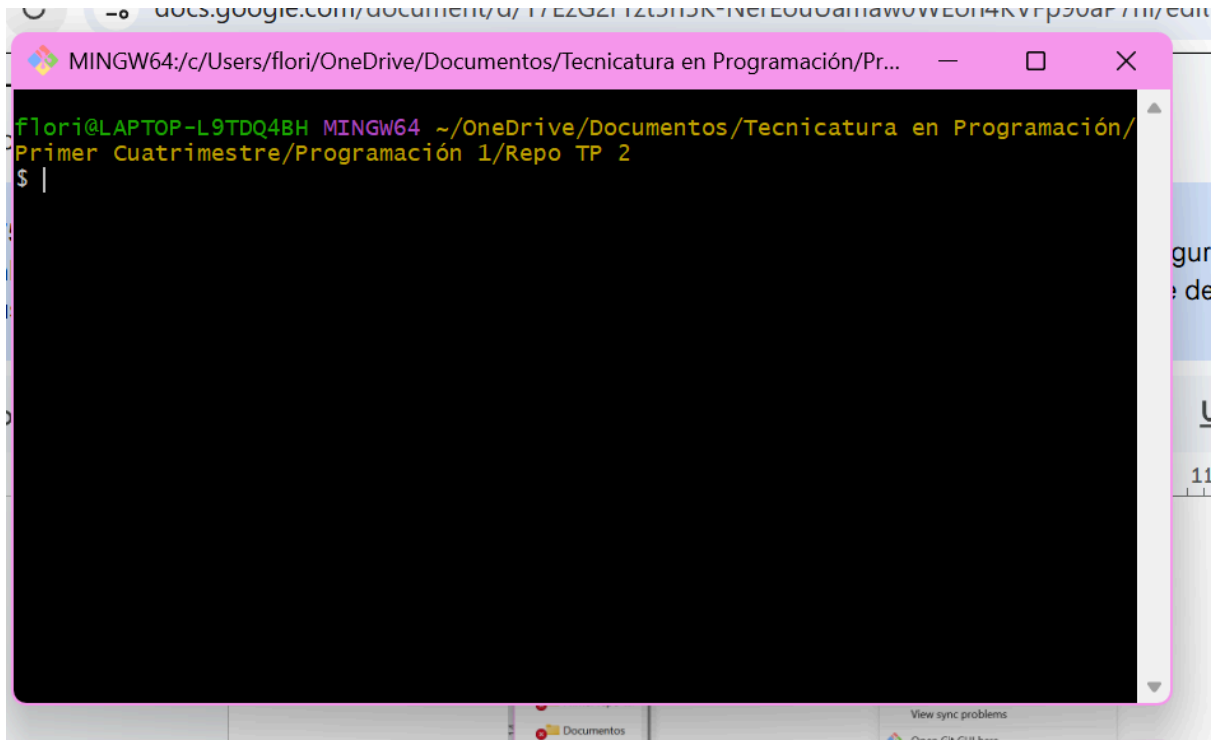
[Create repository](#)

• ¿Cómo crear una rama en Git?

Para crear una rama en Git, debemos:

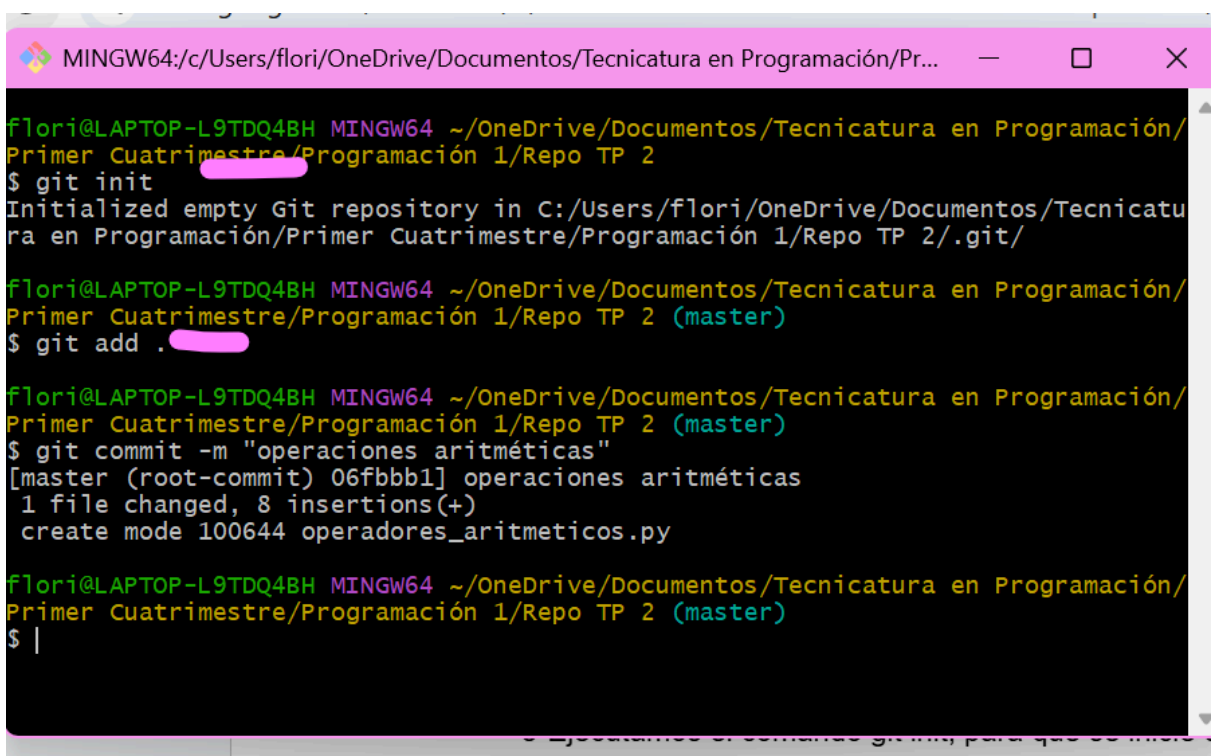
- 1-Crear/Localizar la carpeta guardada en nuestra PC, ,en donde se encuentran los proyectos que queremos incorporar en nuestra rama.
- 2-Hacer click con boton derecho dentro de la carpeta, y seleccionar Open Git Bash here, para que se abra la consola de Git.





A screenshot of a terminal window titled "MINGW64: c:/Users/flori/OneDrive/Documentos/Tecnicatura en Programación/Pr...". The prompt is "flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2". The cursor is on a new line after the "\$" prompt.

3-Ejecutamos el comando git init, para que se inicie el repositorio, y ya vemos que por defecto se inicializa la rama (master) .
Luego tendremos que usar el comando git add . para que quede guardado en este repositorio el o los archivos que tenemos en nuestra carpeta.



A screenshot of a terminal window showing the execution of Git commands. The prompt is "flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2". The commands and their outputs are as follows:

```
$ git init
Initialized empty Git repository in C:/Users/flori/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2/.git/

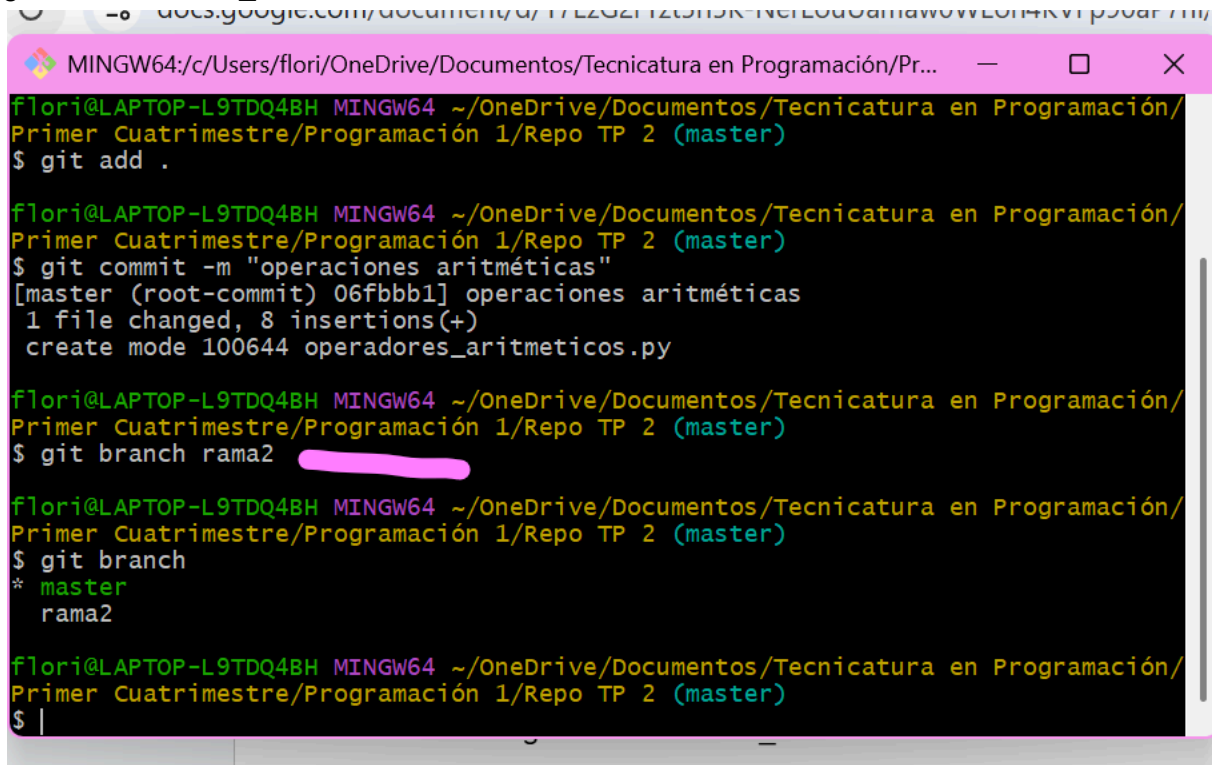
flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2 (master)
$ git add .

flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2 (master)
$ git commit -m "operaciones aritméticas"
[master (root-commit) 06fbbb1] operaciones aritméticas
1 file changed, 8 insertions(+)
create mode 100644 operadores_aritmeticos.py

flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2 (master)
$
```

NOTA: Esto también puede hacerse desde el vscode.

-Si queremos crear un nombre de rama nuevo, diferente de la rama "master", usamos el comando
git branch nombre_ramanueva



```
flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documents/Tecnatura en Programaci3n/Primer Cuatrimestre/Programaci3n 1/Repo TP 2 (master)
$ git add .

flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documents/Tecnatura en Programaci3n/Primer Cuatrimestre/Programaci3n 1/Repo TP 2 (master)
$ git commit -m "operaciones aritm3ticas"
[master (root-commit) 06fbbb1] operaciones aritm3ticas
1 file changed, 8 insertions(+)
create mode 100644 operadores_aritmeticos.py

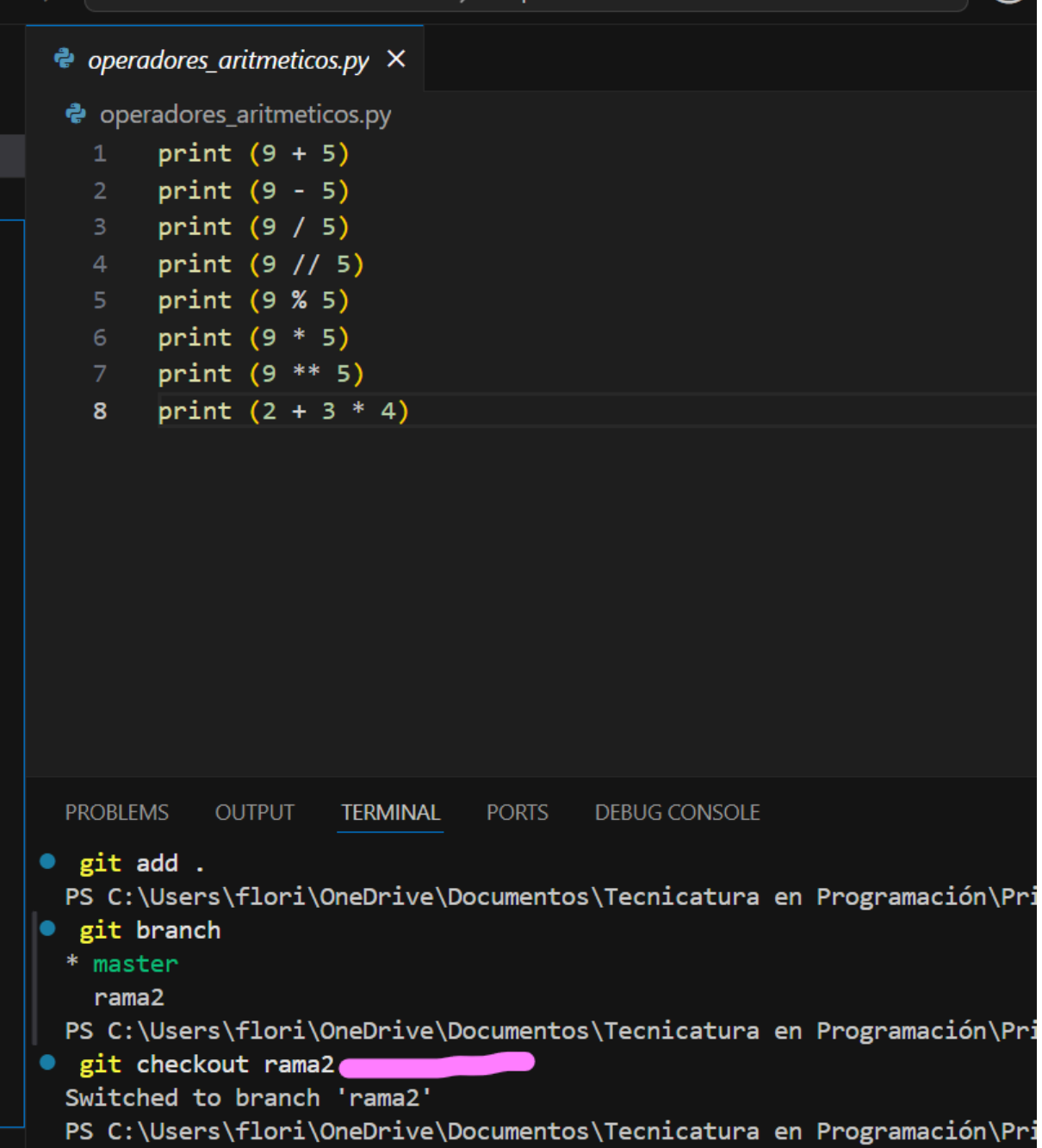
flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documents/Tecnatura en Programaci3n/Primer Cuatrimestre/Programaci3n 1/Repo TP 2 (master)
$ git branch rama2

flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documents/Tecnatura en Programaci3n/Primer Cuatrimestre/Programaci3n 1/Repo TP 2 (master)
$ git branch
* master
  rama2

flori@LAPTOP-L9TDQ4BH MINGW64 ~/OneDrive/Documents/Tecnatura en Programaci3n/Primer Cuatrimestre/Programaci3n 1/Repo TP 2 (master)
$ |
```

- **¿C3mo cambiar a una rama en Git?**

Para cambiar a otra rama en Git, usamos el comando git checkout nombre_rama a la que quiero cambiarme



The image shows a code editor with a file named `operadores_aritmeticos.py` open. The file contains eight lines of Python code that demonstrate various arithmetic operations. Below the code editor, there is a terminal window with tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, and DEBUG CONSOLE. The terminal shows a series of Git commands being executed in a PowerShell prompt, including `git add .`, `git branch`, and `git checkout rama2`. The `git checkout rama2` command is highlighted with a pink brush.

```
operadores_aritmeticos.py X
operadores_aritmeticos.py
1  print (9 + 5)
2  print (9 - 5)
3  print (9 / 5)
4  print (9 // 5)
5  print (9 % 5)
6  print (9 * 5)
7  print (9 ** 5)
8  print (2 + 3 * 4)

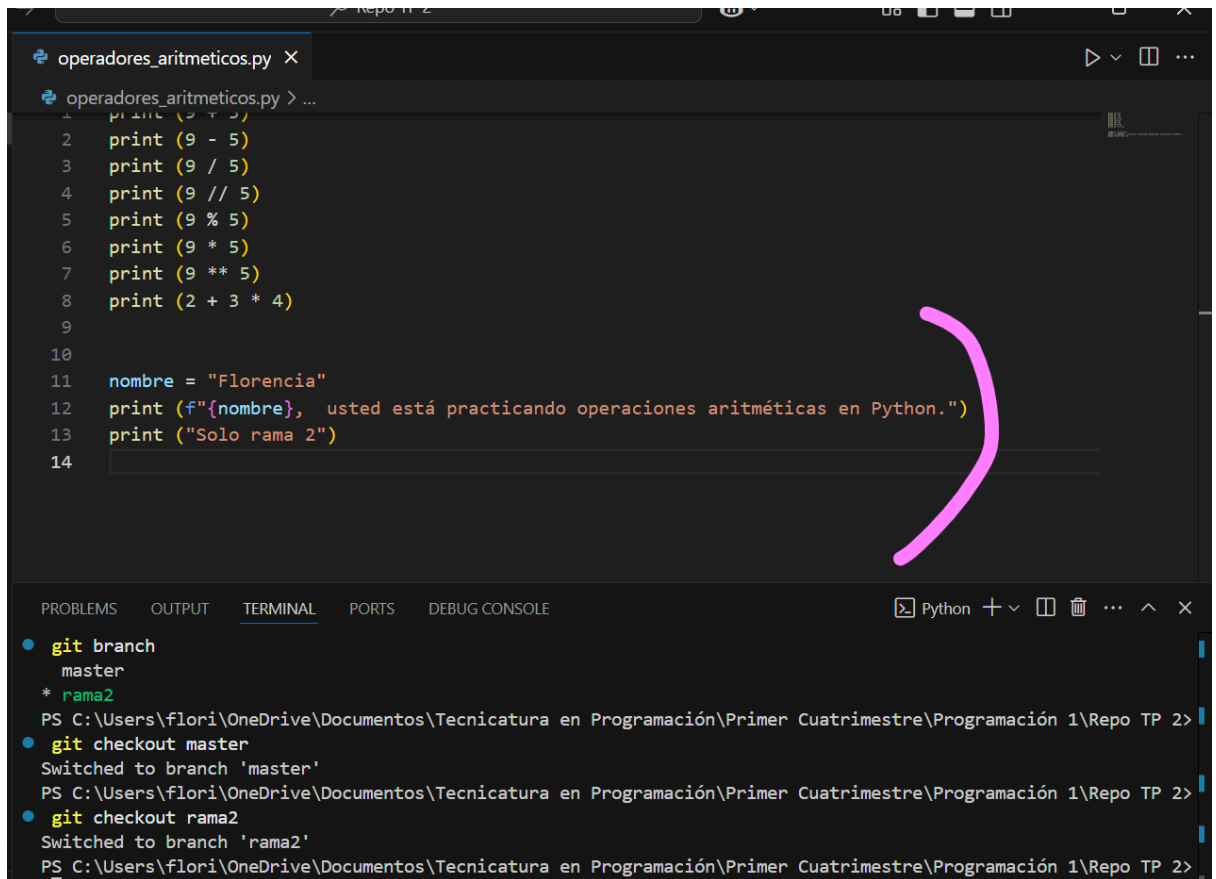
PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
• git add .
  PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Pri
• git branch
  * master
    rama2
  PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Pri
• git checkout rama2
  Switched to branch 'rama2'
  PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Pri
```

- **¿Cómo fusionar ramas en Git?**

Para fusionar ramas en git, debo seguir los siguientes pasos:

1- Uso el comando `git checkout nombre_rama` e introduzco el nombre de la rama a la que quiero llevar los nuevos cambios. Por ejemplo, si quiero llevar mis cambios de la rama2 a la rama master, debo usar `git checkout master`

rama 2:



The screenshot shows the Visual Studio Code editor with a file named `operadores_aritmeticos.py` open. The file contains the following Python code:

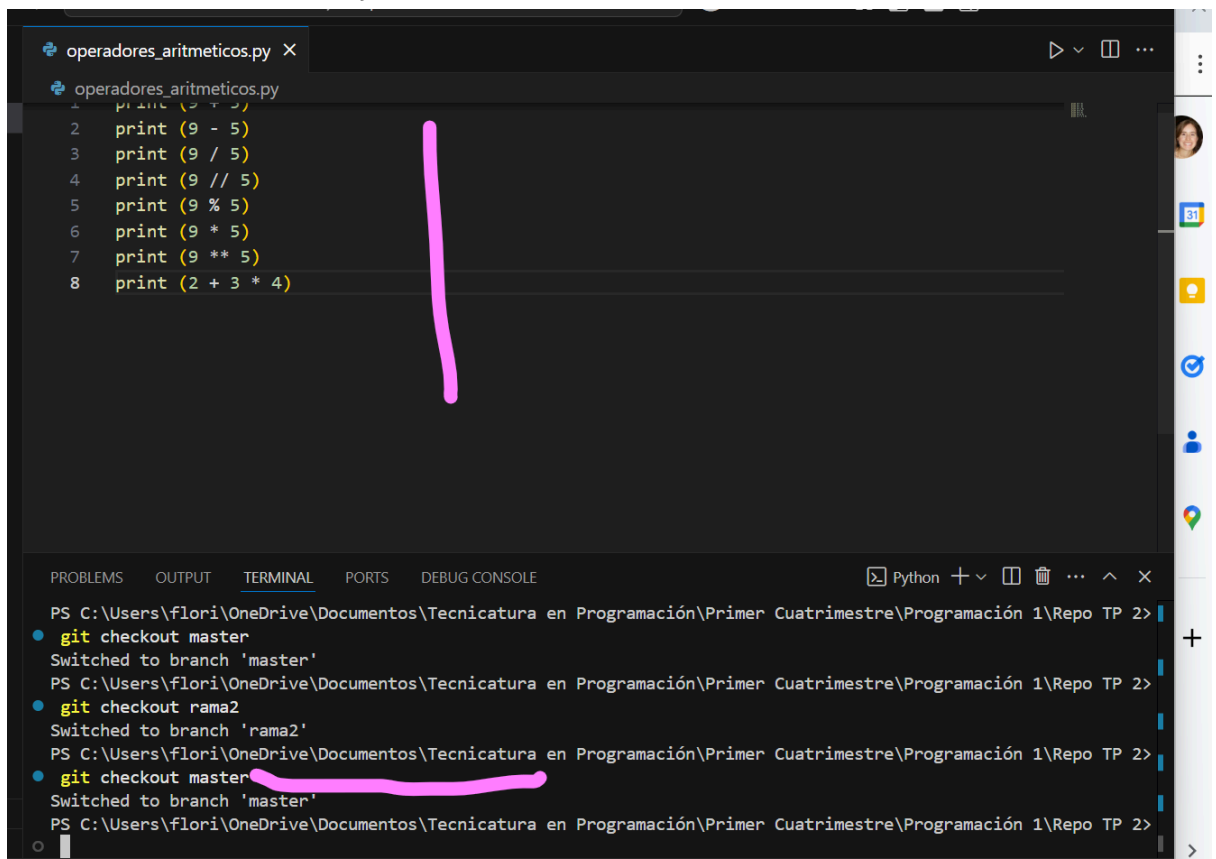
```
1 print(9 + 5)
2 print(9 - 5)
3 print(9 / 5)
4 print(9 // 5)
5 print(9 % 5)
6 print(9 * 5)
7 print(9 ** 5)
8 print(2 + 3 * 4)
9
10
11 nombre = "Flores"
12 print(f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print("Solo rama 2")
14
```

A large pink bracket is drawn on the right side of the code, spanning from line 11 to line 14.

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

```
git branch
master
* rama2
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
git checkout master
Switched to branch 'master'
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
git checkout rama2
Switched to branch 'rama2'
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
```

me paso a la rama master: }



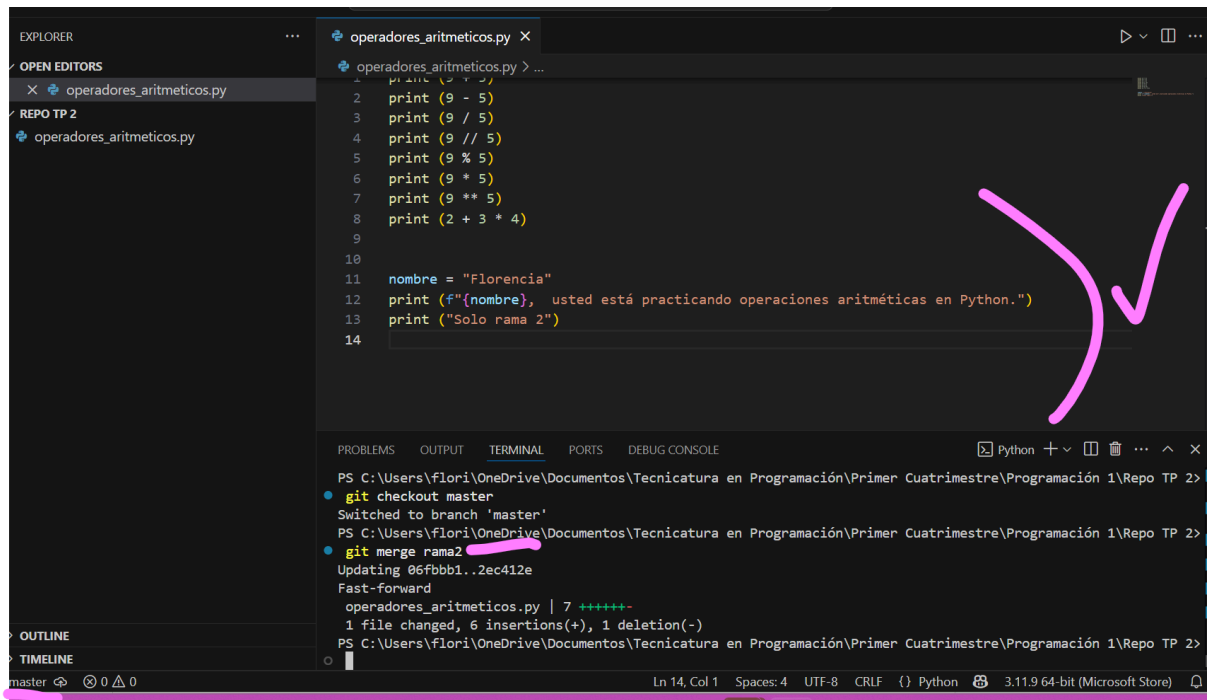
The screenshot shows the Visual Studio Code editor with the same file `operadores_aritmeticos.py` open. The code is the same as in the previous screenshot.

A large pink bracket is drawn on the right side of the code, spanning from line 11 to line 14.

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

```
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
git checkout master
Switched to branch 'master'
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
git checkout rama2
Switched to branch 'rama2'
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
git checkout master
Switched to branch 'master'
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
```

2-Introduzco el comando git merge rama2 para pasar los cambios de mi rama 2 a la rama master:



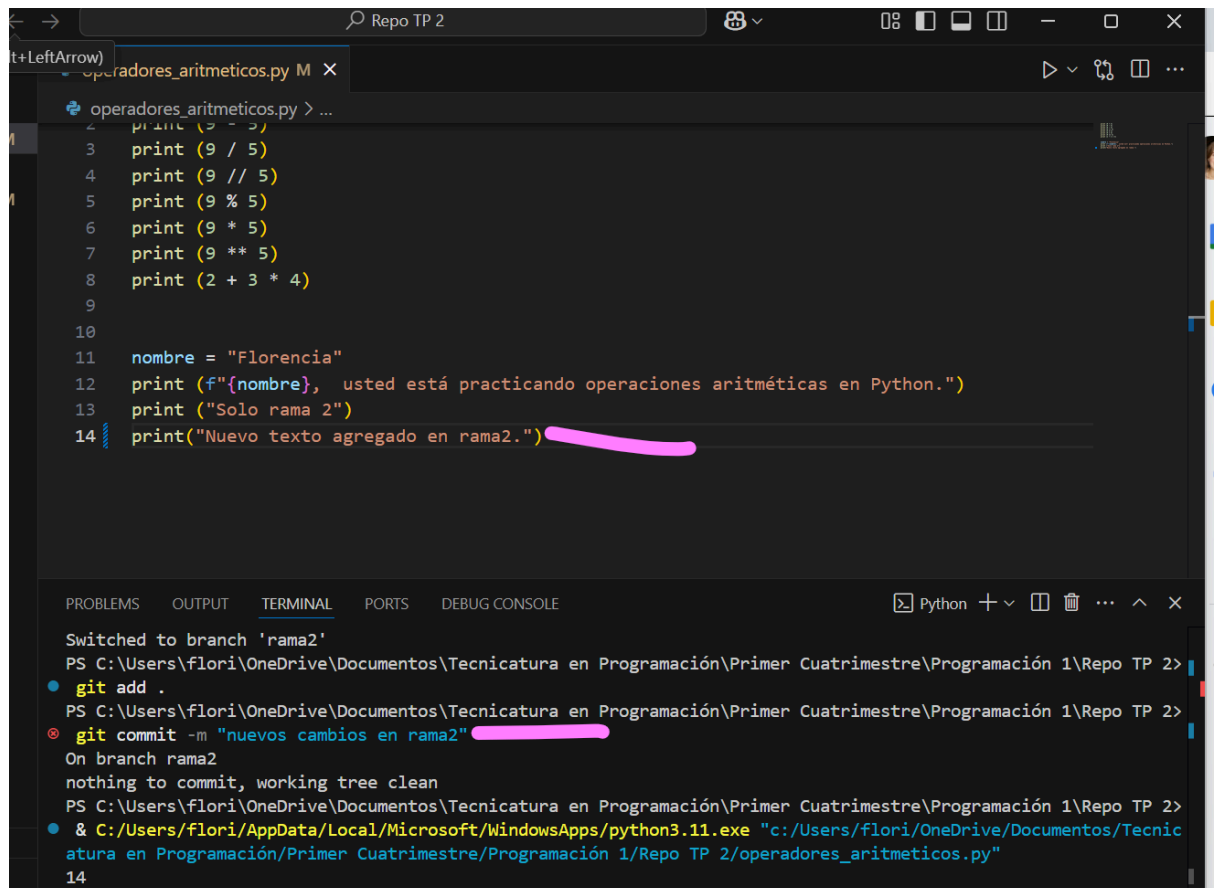
```
operadores_aritmeticos.py
1 print(9 + 5)
2 print(9 - 5)
3 print(9 / 5)
4 print(9 // 5)
5 print(9 % 5)
6 print(9 * 5)
7 print(9 ** 5)
8 print(2 + 3 * 4)
9
10
11 nombre = "Florencia"
12 print(f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print("Solo rama 2")
14
```

```
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
• git checkout master
Switched to branch 'master'
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
• git merge rama2
Updating 06fbb1..2ec412e
Fast-forward
 operadores_aritmeticos.py | 7 +++++-
 1 file changed, 6 insertions(+), 1 deletion(-)
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
```

• ¿Cómo crear un commit en Git?

Usar el comando git commit -m “escribo descripción del cambio”

y de esta forma podrán quedar guardados los cambios en mi repositorio local



```
operadores_aritmeticos.py
1 print(9 + 5)
2 print(9 - 5)
3 print(9 / 5)
4 print(9 // 5)
5 print(9 % 5)
6 print(9 * 5)
7 print(9 ** 5)
8 print(2 + 3 * 4)
9
10
11 nombre = "Florencia"
12 print(f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print("Solo rama 2")
14 print("Nuevo texto agregado en rama2.")
```

```
Switched to branch 'rama2'
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
• git add .
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
• git commit -m "nuevos cambios en rama2"
On branch rama2
nothing to commit, working tree clean
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2>
• & C:/Users/flori/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/flori/OneDrive/Documentos/Tecnicatura en Programación/Primer Cuatrimestre/Programación 1/Repo TP 2/operadores_aritmeticos.py"
14
```

• ¿Cómo enviar un commit a GitHub?

Para enviar un commit a un repositorio remoto en github, una vez que ya creé mi repositorio, debo:

1-Ejecutar el comando `git add .` para guardar mis cambios en mi repo local.

2-Ejecutar `git commit -m ""` para indicar qué cambio estoy añadiendo a mi repositorio.

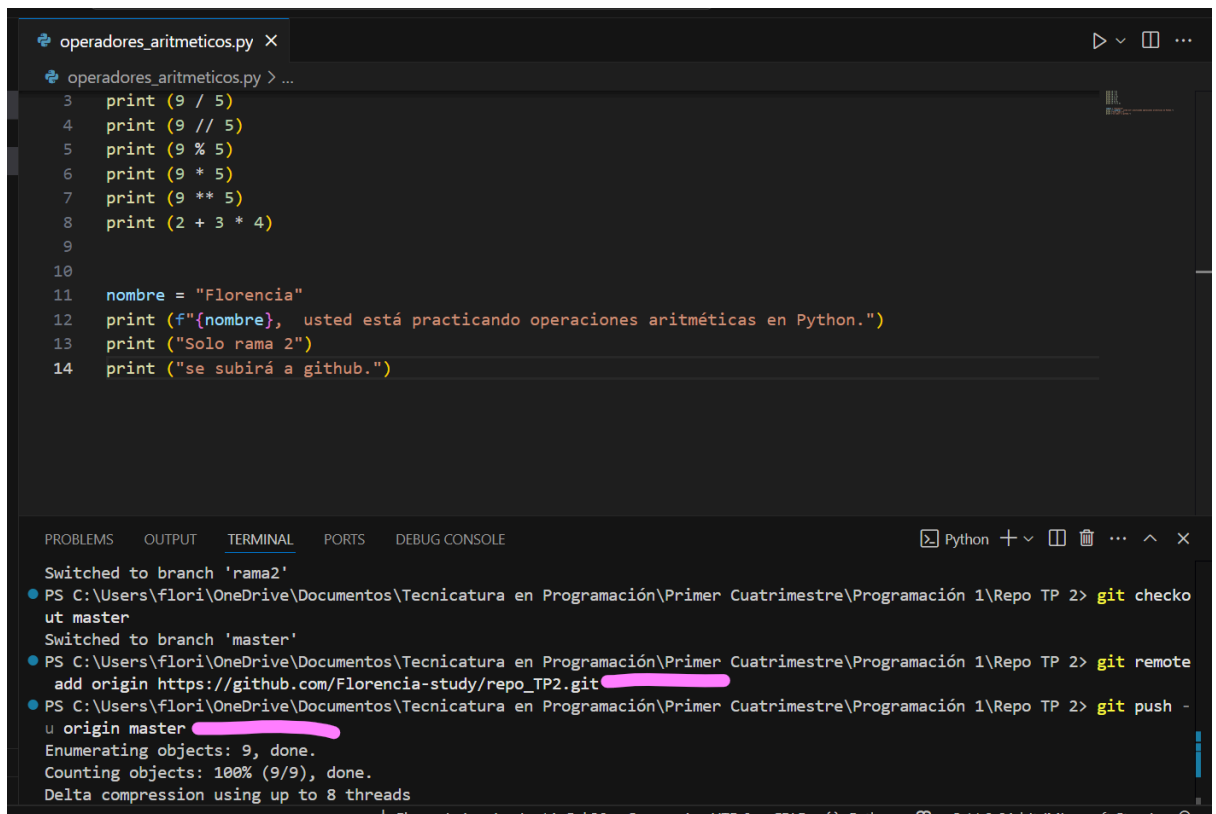
3-Ejecutar el comando `git remote add origin`

`https://github.com/Florencia-study/repo_TP2.git`

(reemplazar por el url de nuestro repositorio remoto) para subir nuestro commit a github.

4- Luego , ejecutar este comando para finalmente pushear nuestro código a ese repositorio remoto que indicamos anteriormente:

`git push -u origin master`



The screenshot shows a Visual Studio Code editor with a file named `operadores_aritmeticos.py` open. The file contains the following Python code:

```
3 print (9 / 5)
4 print (9 // 5)
5 print (9 % 5)
6 print (9 * 5)
7 print (9 ** 5)
8 print (2 + 3 * 4)
9
10
11 nombre = "Florencia"
12 print (f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print ("Solo rama 2")
14 print ("se subirá a github.")
```

Below the editor, the TERMINAL panel is active, showing the following commands and output:

```
Switched to branch 'rama2'
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2> git checkout master
Switched to branch 'master'
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2> git remote add origin https://github.com/Florencia-study/repo_TP2.git
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2> git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
```

Y ahí queda nuestro código subido:

Floresia-study / repo_TP2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

repo_TP2 Public

master 1 Branch 0 Tags

Go to file Add file Code

Floresia-study guardar github 6aeaa84 · 10 minutes ago 3 Commits

operadores_aritmeticos.py guardar github 10 minutes ago

README

repo_TP2 / operadores_aritmeticos.py

Floresia-study guardar github 6aeaa84 · 10 minutes ago History

Code Blame 14 lines (12 loc) · 276 Bytes Code 55% faster with GitHub Copilot Raw Copy Download Edit View

```
1 print (9 + 5)
2 print (9 - 5)
3 print (9 / 5)
4 print (9 // 5)
5 print (9 % 5)
6 print (9 * 5)
7 print (9 ** 5)
8 print (2 + 3 * 4)
9
10
11 nombre = "Floresia"
12 print (f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print ("Solo rama 2")
14 print ("se subirá a github.")
```

- **¿Qué es un repositorio remoto?**

Comunidad o sitio web en donde se guarda el código de un usuario (distinto de su local), y en donde podrá compartir ese repo con otros usuarios, ya sea de forma pública o privada (limitado a algunos usuarios solamente).

- **¿Cómo agregar un repositorio remoto a Git?**

Debemos usar el comando `git remote add origin URL` del repositorio remoto creado.

- **¿Cómo empujar cambios a un repositorio remoto?**

Para empujar cambios a un repositorio remoto, hay que usar el comando `git push -u origin master` (reemplazar master por el nombre de la rama a la que quiero pushear mis cambios)

- **¿Cómo tirar de cambios de un repositorio remoto?**

Podemos usar el comando `git pull origin master` para descargar cambios del repositorio remoto y bajarlos a mi local, para que puedan fusionarse en mi rama local.

- **¿Qué es un fork de repositorio?**

Un fork es una copia del repositorio de otro usuario. Esta copia de su repositorio se crea en mi cuenta de usuario, y yo podré trabajar en ella y realizar mis cambios en ese repo forkeado, y los cambios entonces quedarán grabados allí, en ese fork (y no en el repo original del usuario de quién creé la copia).

- **¿Cómo crear un fork de un repositorio?**

- 1-Debo iniciar sesión en github.com
- 2-Luego, navego hacia la URL del repositorio que quiero forkear.
- 3-En ese repositorio, hago click en el botón Fork.
- 4-Allí me aseguro de que la cuenta a la que se copiará el fork sea la mía, puedo elegir un nombre al repo forkeado, y cambiar la descripción.
- 5-Me aseguro de que se esté forkeando la rama main.
- 6-Click en Create fork.

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

Para solicitar que mis cambios se mergeen a la rama principal compartida con otros usuarios (hacer un pull request), debo:

- navegar a la rama principal del repositorio.
- Seleccionar la rama que contiene mis propios commits, es decir mi rama.
- Click en "Compare and Pull request".
- Cuando ya tengo mis cambios listos, click en "Create Pull Request".

- **¿Cómo aceptar una solicitud de extracción?**

- Hacer clic en Solicitudes de incorporación de cambios en el nombre del repositorio
- Seleccionar la solicitud que voy a revisar.
- Revisar los cambios
- Hacer clic en Revisar cambios sobre el código modificado
- Se puede escribir un comentario sobre los cambios
- Seleccionar Aprobar
- Hacer clic en Enviar revisión

- **¿Qué es un etiqueta en Git?**

Son rótulos que permiten al usuario categorizar y documentar de modo más visible los tópicos, por ej: si es un bug, una solicitud de cambio, duplicado, mejora, revisado, etc.

- **¿Cómo crear una etiqueta en Git?**

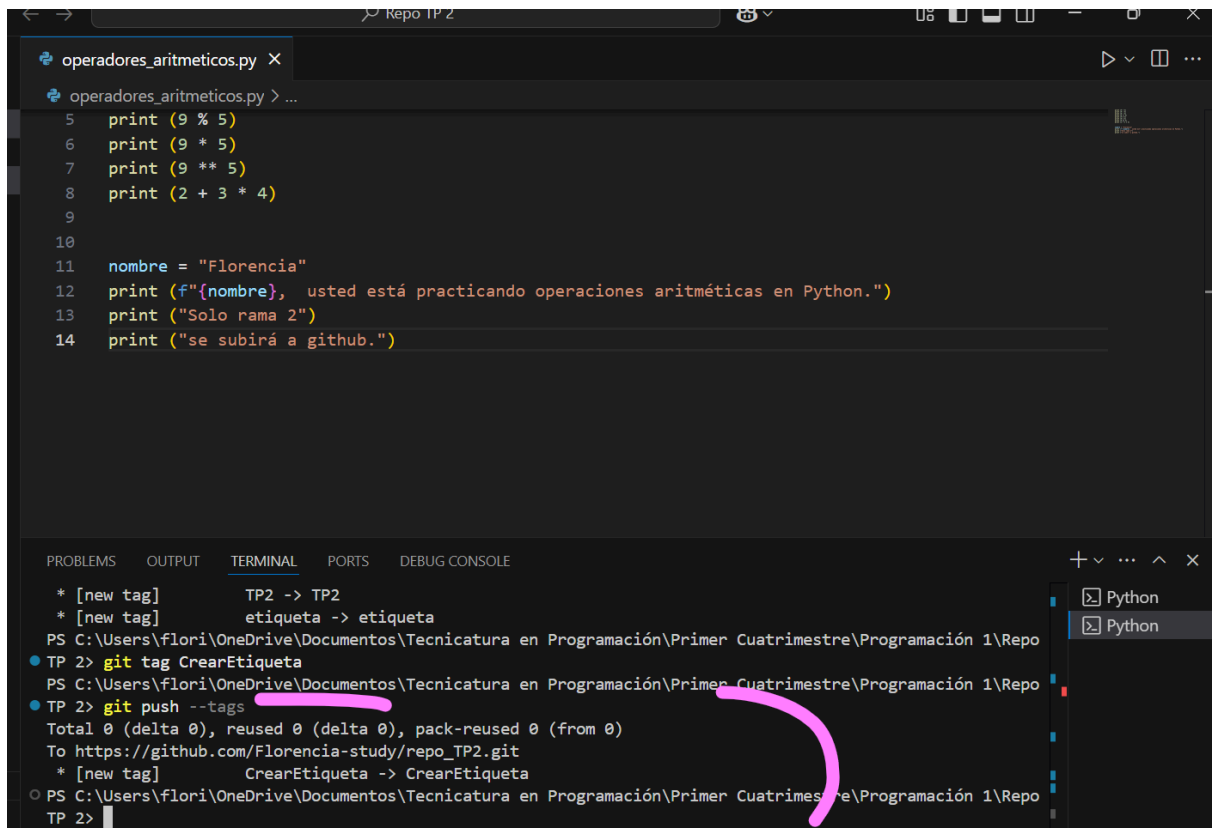
Para Git, podemos usar este comando: `git tag TAG_NAME`

Para GitHub, se puede seguir estos pasos:

- 1-Navegar hasta mi repo.
- 2-Hacer click en Issues o Pull requests.
- 3-Hacer click en Labels – New labels.
- 4-Ponerle un nombre, y luego click en Create Label.

- **¿Cómo enviar una etiqueta a GitHub?**

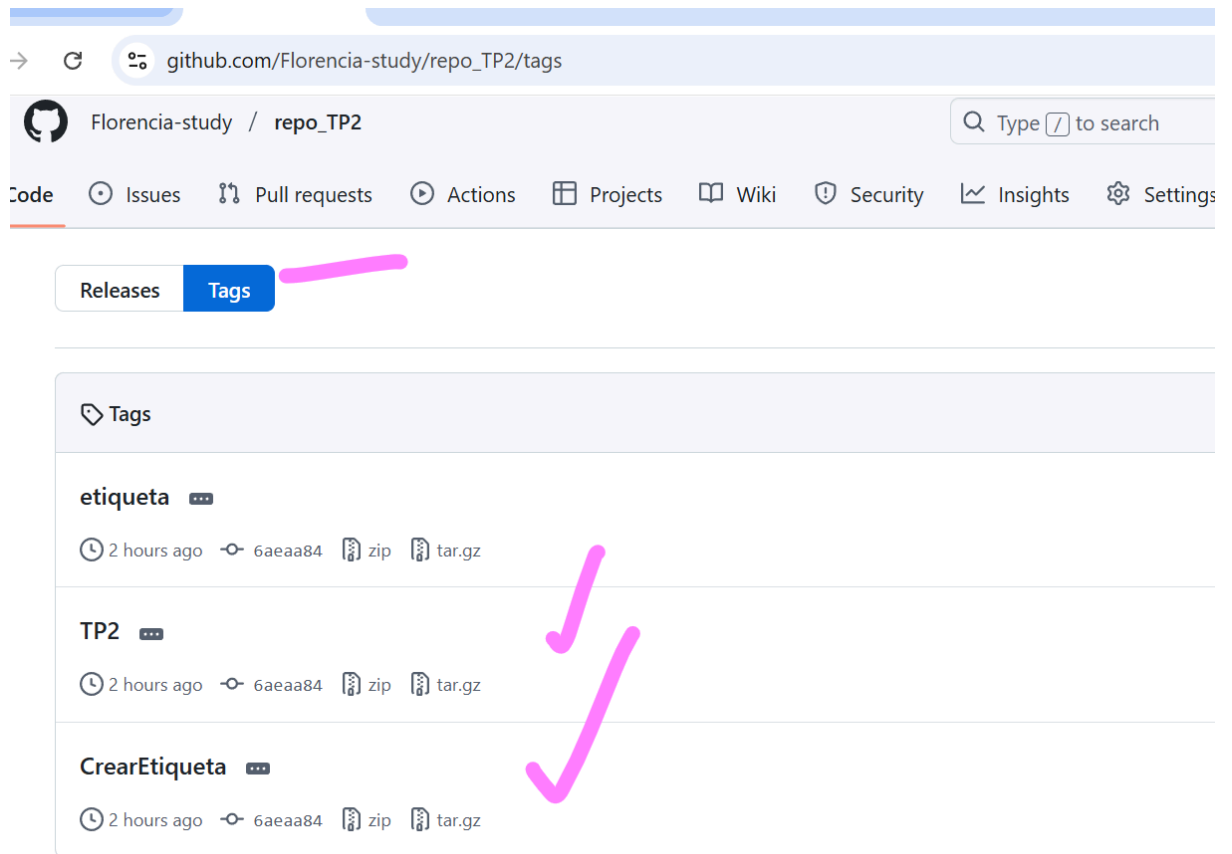
Hay que usar el comando `git push --tags`, para que las tags que hemos creado se suban también a nuestro repo en línea:



```
operadores_aritmeticos.py X
operadores_aritmeticos.py > ...
5 print (9 % 5)
6 print (9 * 5)
7 print (9 ** 5)
8 print (2 + 3 * 4)
9
10
11 nombre = "Florencia"
12 print (f"{nombre}, usted está practicando operaciones aritméticas en Python.")
13 print ("Solo rama 2")
14 print ("se subirá a github.")

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
* [new tag] TP2 -> TP2
* [new tag] etiqueta -> etiqueta
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git tag CrearEtiqueta
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Florencia-study/repo_TP2.git
* [new tag] CrearEtiqueta -> CrearEtiqueta
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2>
```

Luego reviso en git hub, me meto en mi repo, click en Tags, y puedo ver que mi etiqueta aparece ahora también en mi repo en línea:



- **¿Qué es un historial de Git?**

Es un registro de los cambios realizados en el repo en cuestión.

- **¿Cómo ver el historial de Git?**

Debo usar el comando git log

- **¿Cómo buscar en el historial de Git?**

En Internet y en el sitio oficial de Git, encontré los siguientes comandos que pueden ayudar a realizar una búsqueda más eficiente por el historial:

git grep y el nombre de una palabra o numero que quiero ver

git grep -n para ver los números de línea donde se encontraron coincidencias

git grep --count para ver cuántas coincidencias hay en cada archivo

git grep -p para ver en qué método o función se encontró una coincidencia

git log

git log -L para ver el historial de una función o línea de código

git log --pickaxe para buscar confirmaciones que agreguen o eliminen una cadena específica

git --grep=<pattern> para buscar un patrón en los mensajes de confirmación

git blame - Muestra el último commit que modificó cada línea de un archivo

gitk - Muestra la gráfica de commits y los archivos de los árboles de cada revisión

Para filtrar los resultados de las búsquedas: git log --since="6 days" para mostrar solo los commits realizados en los últimos 6 días.

- **¿Cómo borrar el historial de Git?**

Usando el comando git clean

- **¿Qué es un repositorio privado en GitHub?**

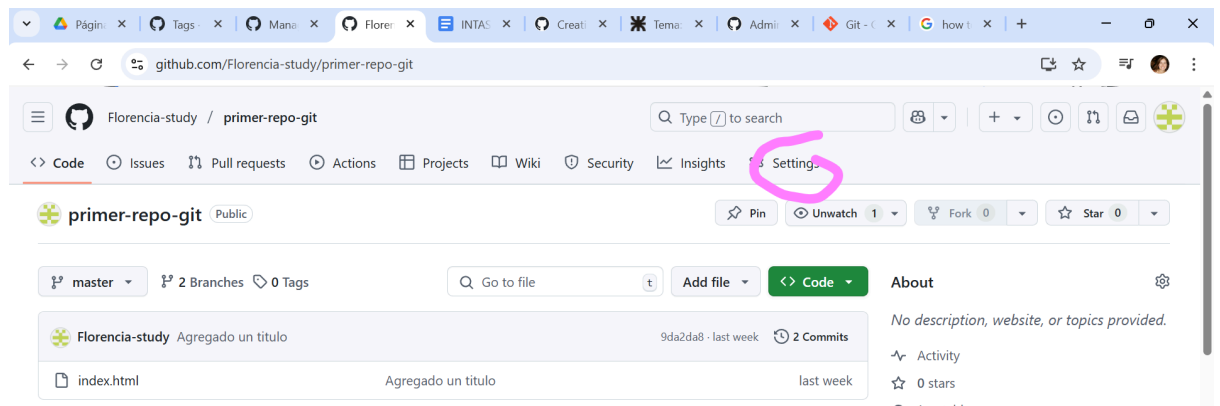
Es un repositorio que creamos en nuestra cuenta de GitHub, pero que no queremos que otros usuarios vean. En él podemos seleccionar o restringir a qué usuarios se le dará acceso y visibilidad a nuestro repositorio, y qué usuarios podrán enviar un commit a nuestro repo.

- **¿Cómo crear un repositorio privado en GitHub?**

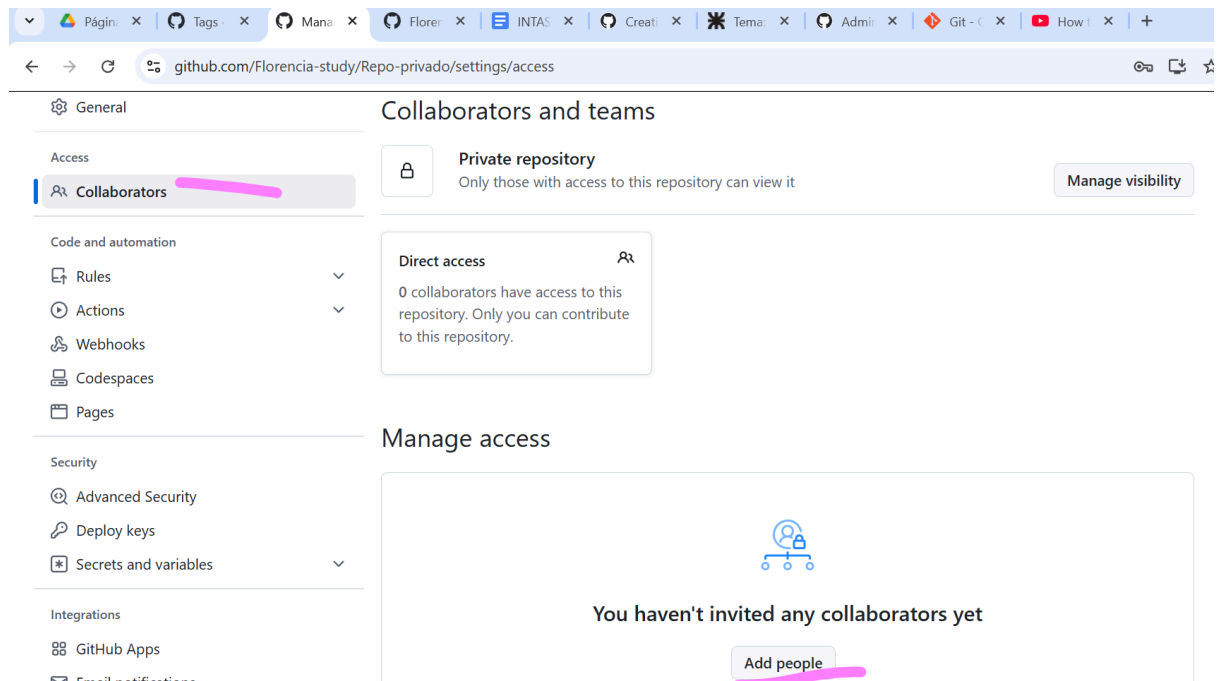
Cuando seleccionamos la opción “New”, debemos seleccionar la opción Privado.

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

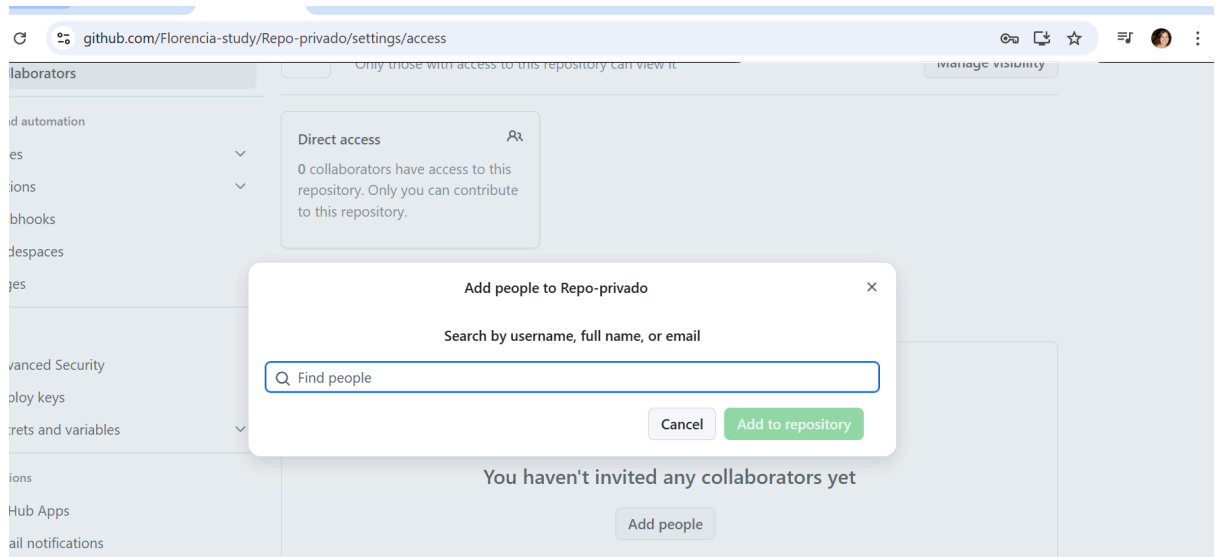
Dentro de mi repositorio privado, debo hacer click en Settings:



-Luego click en Access –Collaborators , poner nuevamente mi contraseña, y luego click en Add people



-Allí insertamos el nombre de usuario o cuenta de e-mail de la persona con quién quiero compartir mi repo, y click en Add to repository.



- **¿Qué es un repositorio público en GitHub?**

Un repositorio público en GitHub es aquel que está visible para otros usuarios, con lo que podrán por ej forkear o clonar ese repo.

- **¿Cómo crear un repositorio público en GitHub?**

Cuando seleccionamos la opción “New”, debemos seleccionar la opción Public.

- **¿Cómo compartir un repositorio público en GitHub?**

-Dentro de mi repositorio privado, debo hacer click en Settings:

-Luego click en Access –Collaborators ,, y luego click en Add people

-Allí insertamos el nombre de usuario o cuenta de e-mail de la persona con quién quiero compartir mi repo, y click en Add to repository.

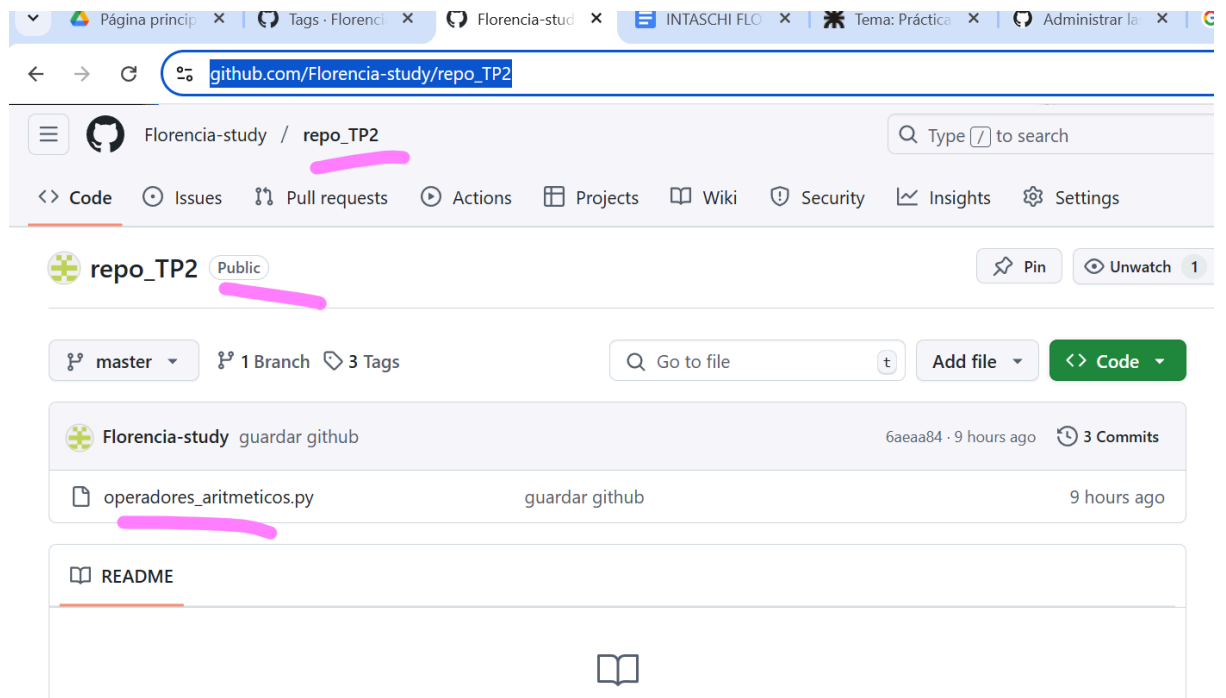
*Otra opción: puedo compartir el URL de mi repositorio con el usuario.

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.

- HECHO :

Link del repositorio creado:

https://github.com/Florencia-study/repo_TP2



- 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).

HECHO:

```
operadores_aritmeticos.py x mi-archivo.txt cls
mi-archivo.txt
1 Este es un archivo que estoy realizando para incluir en el Repo TP2 creado para el TP 2 de Github
2 Alumna: Florencia Intaschi.
3

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git add .
warning: in the working copy of 'cls', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git commit -m "Creando mi-archivo.txt"
[master eed915a] Creando mi-archivo.txt
2 files changed, 17 insertions(+)
create mode 100644 cls
create mode 100644 mi-archivo.txt
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> con git push origin master
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF P
```



```
operadores_aritmeticos.py  mi-archivo.txt  cls

mi-archivo.txt
1 Este es un archivo que estoy realizando para incluir en el Repo TP2 creado para el TP 2 de Github.
2 Alumna: Florencia Intaschi.
3

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 685 bytes | 342.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Florencia-study/repo_TP2.git
```

Los cambios y el archivo se subieron correctamente a mi repositorio en GitHub:

github.com/Florencia-study/repo_TP2

Florencia-study / repo_TP2

repo_TP2 Public

master 1 Branch 3 Tags

Go to file Add file Code

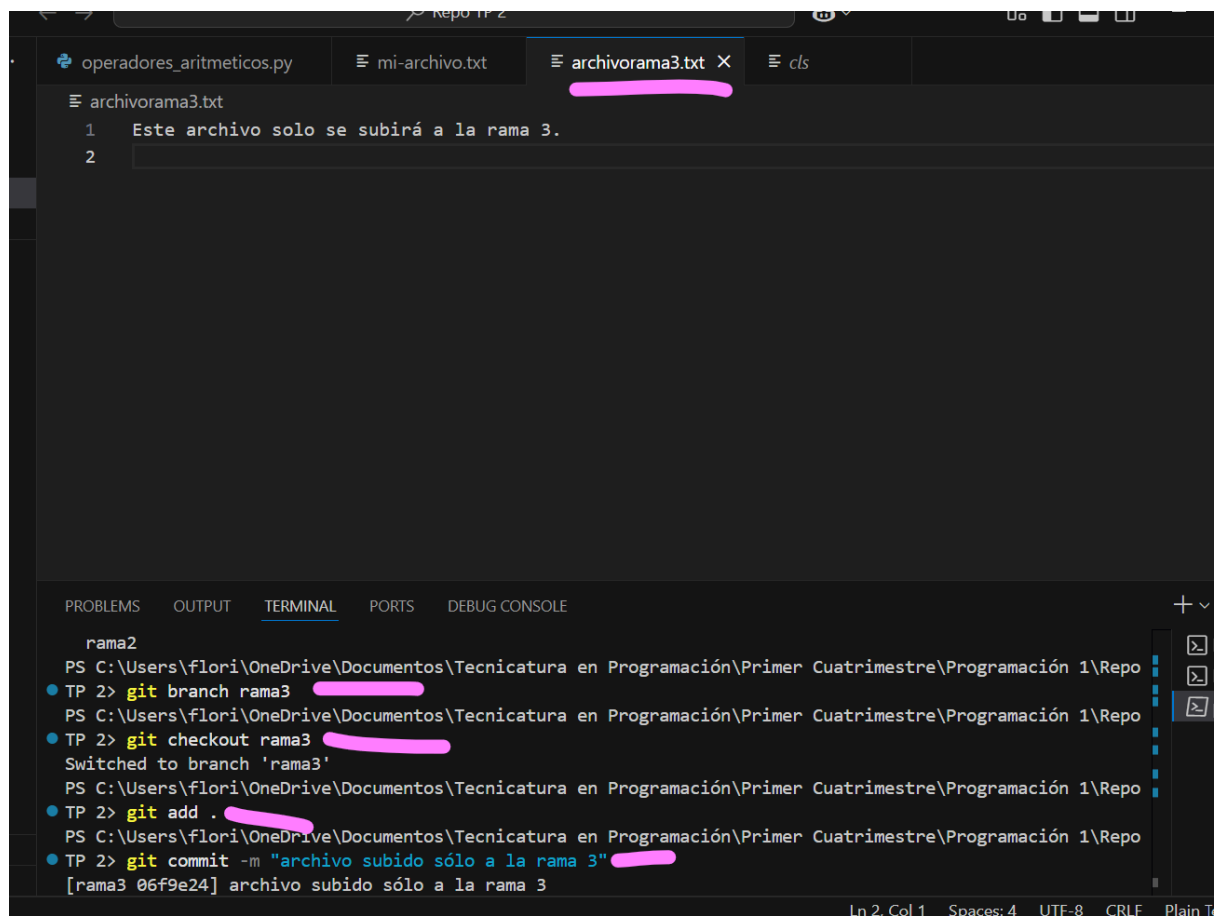
File	Commit Message	Time
cls	Creando mi-archivo.txt	4 minutes ago
mi-archivo.txt	Creando mi-archivo.txt	4 minutes ago
operadores_aritmeticos.py	guardar github	10 hours ago

README



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

HECHO:



```
operadores_aritmeticos.py  mi-archivo.txt  archivorama3.txt  cls

archivorama3.txt
1 Este archivo solo se subirá a la rama 3.
2

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

PS C:\Users\flori\OneDrive\Documents\Tecnología en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2> git push -u origin rama3
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 341 bytes | 341.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'rama3' on GitHub by visiting:
```

← → ↻ [github.com/Florencia-study/repo_TP2/tree/rama3](#)

repo_TP2 Public Pin Unwatch 1

rama3 had recent pushes 32 seconds ago Compare & pull request

rama3 2 Branches 3 Tags Go to file Add file Code

This branch is 1 commit ahead of **master** Contribute

Florencia-study archivo subido sólo a la rama 3 06f9e24 · 2 minutes ago 6 Commits

archivorama3.txt	archivo subido sólo a la rama 3	2 minutes ago
cls	Creando mi-archivo.txt	17 minutes ago
mi-archivo.txt	nuevos cambios al txt	7 minutes ago
operadores_aritmeticos.py	guardar github	10 hours ago

README

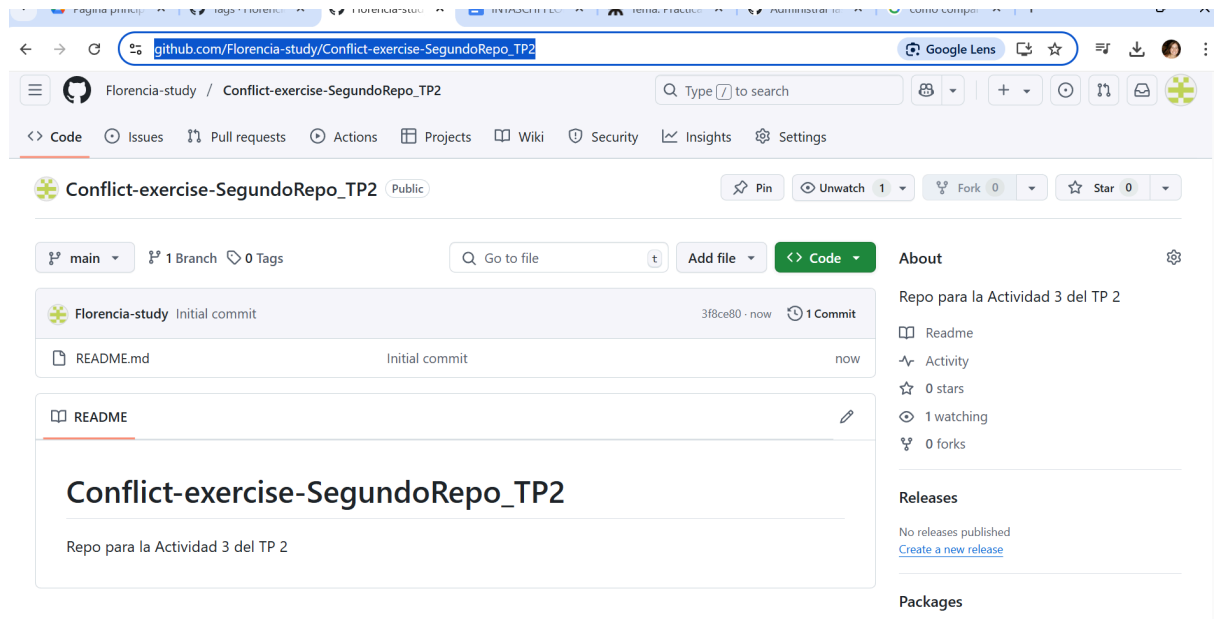
3) Realizar la siguiente actividad:

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".

HECHO:

Link del nuevo repositorio:

https://github.com/Florencia-study/Conflict-exercise-SegundoRepo_TP2



Paso 2: Clonar el repositorio a tu máquina local

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the file structure of a repository named 'REPO TP 2', which includes files like 'Conflict-exercise-SegundoRepo_TP2', 'archivorama3.txt', 'cls', 'mi-archivo.txt', and 'operadores_aritmeticos.py'. The main editor area shows the 'archivorama3.txt' file with two lines of text: 'Este archivo solo se subirá a la rama 3.' and an empty line. The bottom panel shows the 'TERMINAL' tab with a PowerShell session. The command executed is 'git clone https://github.com/Florencia-study/Conflict-exercise-SegundoRepo_TP2', which has successfully cloned the repository into the current directory. The status bar at the bottom indicates the current file is 'archivorama3.txt'.

This screenshot shows the same VS Code environment. The Explorer sidebar remains the same. The main editor area is now empty. The 'TERMINAL' tab shows a new PowerShell session where the command 'cd Conflict-exercise-SegundoRepo_TP2' has been entered. Below the command, a detailed error message is displayed: 'ObjectNotFound: (C:\Users\flori\...\n\nflict-exercise:String) [Set-Location], ItemNotFoundException'. This indicates that the directory path specified in the command does not exist. The status bar at the bottom shows the current file is 'rama3'.

Paso 3: Crear una nueva rama y editar un archivo

```
operadores_aritmeticos.py  mi-archivo.txt  README.md  archivorama3.txt
Conflict-exercise-SegundoRepo_TP2 > README.md > # Conflict-exercise-SegundoRepo_TP2
1  # Conflict-exercise-SegundoRepo_TP2
2  Repo para la Actividad 3 del TP 2
3

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git branch feature-branch
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git checkout feature-branch
Switched to branch 'feature-branch'
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git branch
* feature-branch
  main
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2>

Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  {}  Markdo
```

The image shows a Visual Studio Code editor window with a dark theme. The editor has several tabs open at the top: 'operadores_aritmeticos.py', 'mi-archivo.txt', 'README.md' (which is the active tab), and 'archivorama3.txt'. The 'README.md' file contains the following text:

```
Conflict-exercise-SegundoRepo_TP2 > # Conflict-exercise-SegundoRepo_TP2
1  # Conflict-exercise-SegundoRepo_TP2
2  Repo para la Actividad 3 del TP 2
3
4  Este es un cambio en la feature branch
5
```

Below the editor, there is a panel with tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', 'PORTS', and 'DEBUG CONSOLE'. The 'TERMINAL' tab is active, showing a series of git commands and their outputs:

```
TP 2\Conflict-exercise-SegundoRepo_TP2> git branch
* feature-branch
  main
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git add README.md
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git commit -m "Added a line in feature-branch"
[feature-branch 27563c5] Added a line in feature-branch
 1 file changed, 2 insertions(+)
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2>
```

At the bottom of the terminal panel, there is a status bar showing 'Flores (now)', 'Ln 4, Col 39', 'Spaces: 4', 'UTF-8', 'CRLF', and '{ } Mar'.

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

Conflict-exercise-SegundoRepo_TP2 > README.md > # Conflict-exercise-SegundoRepo_TP2

```
1 # Conflict-exercise-SegundoRepo_TP2
2 Repo para la Actividad 3 del TP 2
3
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

- TP 2\Conflict-exercise-SegundoRepo_TP2> git add README.md
- PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
- TP 2\Conflict-exercise-SegundoRepo_TP2> git commit -m "Added a line in feature-branch"
- [feature-branch 27563c5] Added a line in feature-branch
- 1 file changed, 2 insertions(+)
- PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
- TP 2\Conflict-exercise-SegundoRepo_TP2> git checkout main
- Switched to branch 'main'
- Your branch is up to date with 'origin/main'.
- PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
- TP 2\Conflict-exercise-SegundoRepo_TP2>

Ln 3, Col 1 Spaces: 4 UTF-8 CRLF {} Markdown

The image shows a VS Code editor with a file named `README.md` open. The file content is as follows:

```
1 # Conflict-exercise-SegundoRepo_TP2
2 Repo para la Actividad 3 del TP 2
3
4 Linea diferente: Este es un cambio en la main branch.
```

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

```
TP 2\Conflict-exercise-SegundoRepo_TP2> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git add README.md
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git commit -m "Added a line in main branch"
[main 914ded1] Added a line in main branch
1 file changed, 2 insertions(+)
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2>
```

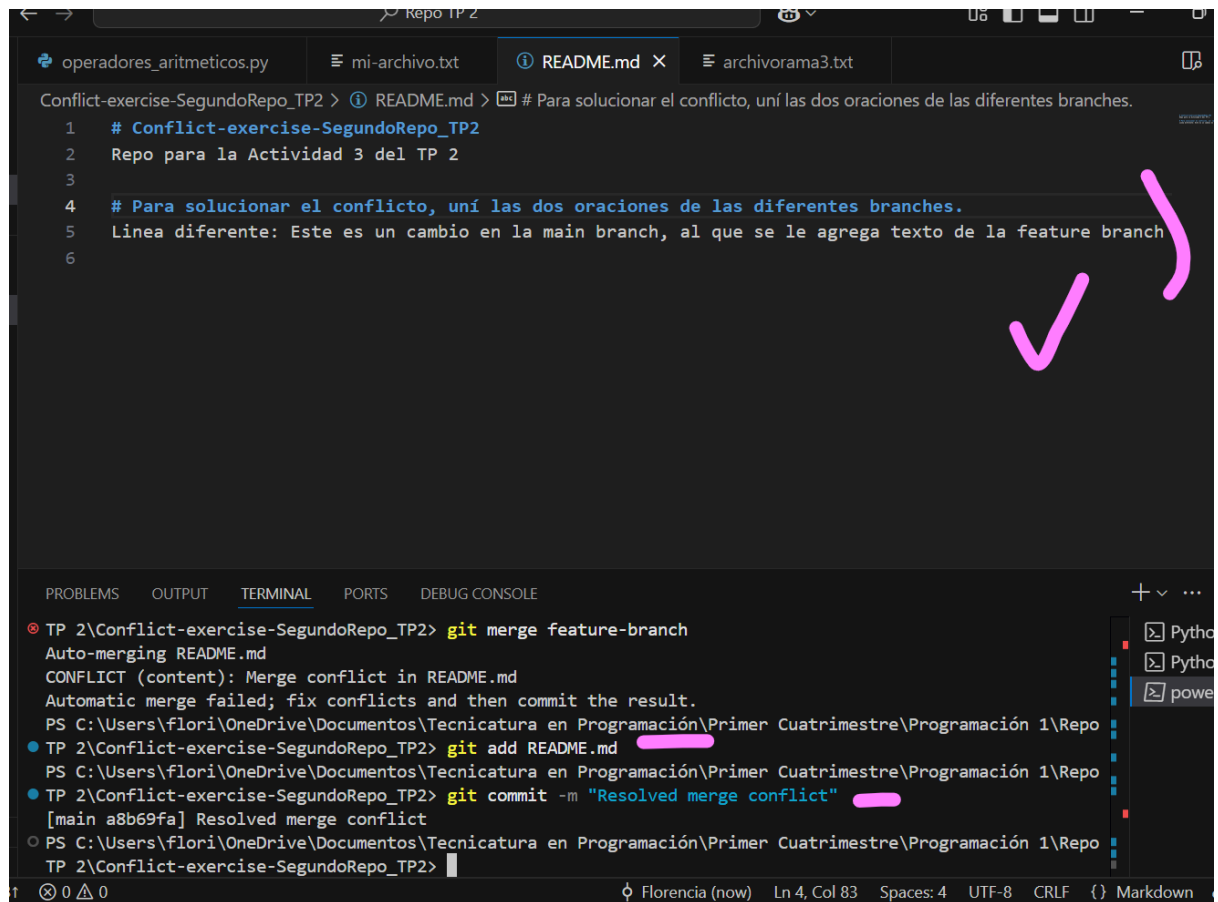
Paso 5: Hacer un merge y generar un conflicto

The image shows a VS Code editor with a file named `README.md` open, displaying a merge conflict. The conflict is highlighted with a pink bracket on the right side of the editor. The conflict is between the current HEAD (Current Change) and the feature-branch (Incoming Change). The conflict is resolved in the Merge Editor.

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

```
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git commit -m "Added a line in main branch"
[main 914ded1] Added a line in main branch
1 file changed, 2 insertions(+)
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> 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\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2>
```

Paso 6: Resolver el conflicto



The screenshot shows the Visual Studio Code editor with a file named `README.md` open. The file content is as follows:

```
1 # Conflict-exercise-SegundoRepo_TP2
2 Repo para la Actividad 3 del TP 2
3
4 # Para solucionar el conflicto, uní las dos oraciones de las diferentes branches.
5 Línea diferente: Este es un cambio en la main branch, al que se le agrega texto de la feature branch
6
```

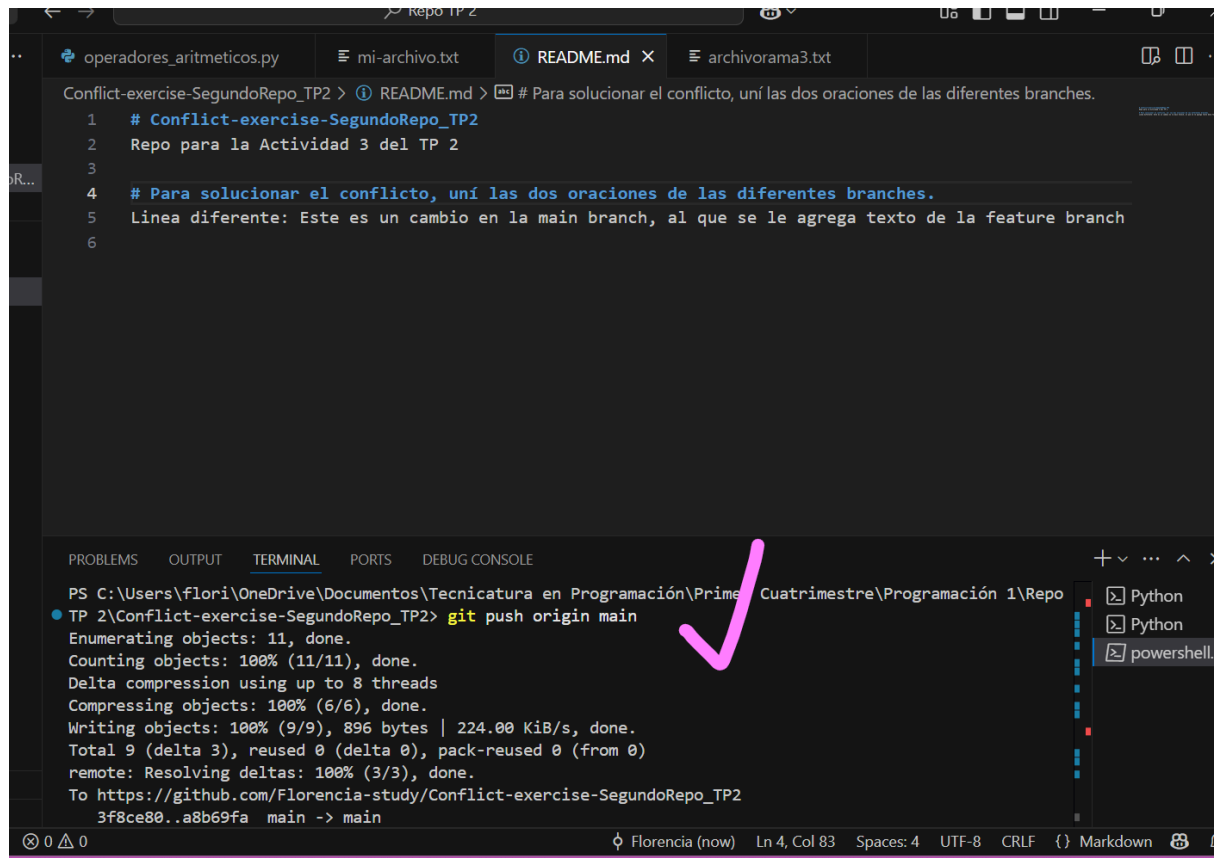
Hand-drawn pink annotations include a checkmark and a bracket on the right side of the editor, highlighting the conflict resolution instructions.

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

```
TP 2\Conflict-exercise-SegundoRepo_TP2> 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\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git add README.md
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> git commit -m "Resolved merge conflict"
[main a8b69fa] Resolved merge conflict
TP 2\Conflict-exercise-SegundoRepo_TP2>
```

Paso 7: Subir los cambios a GitHub

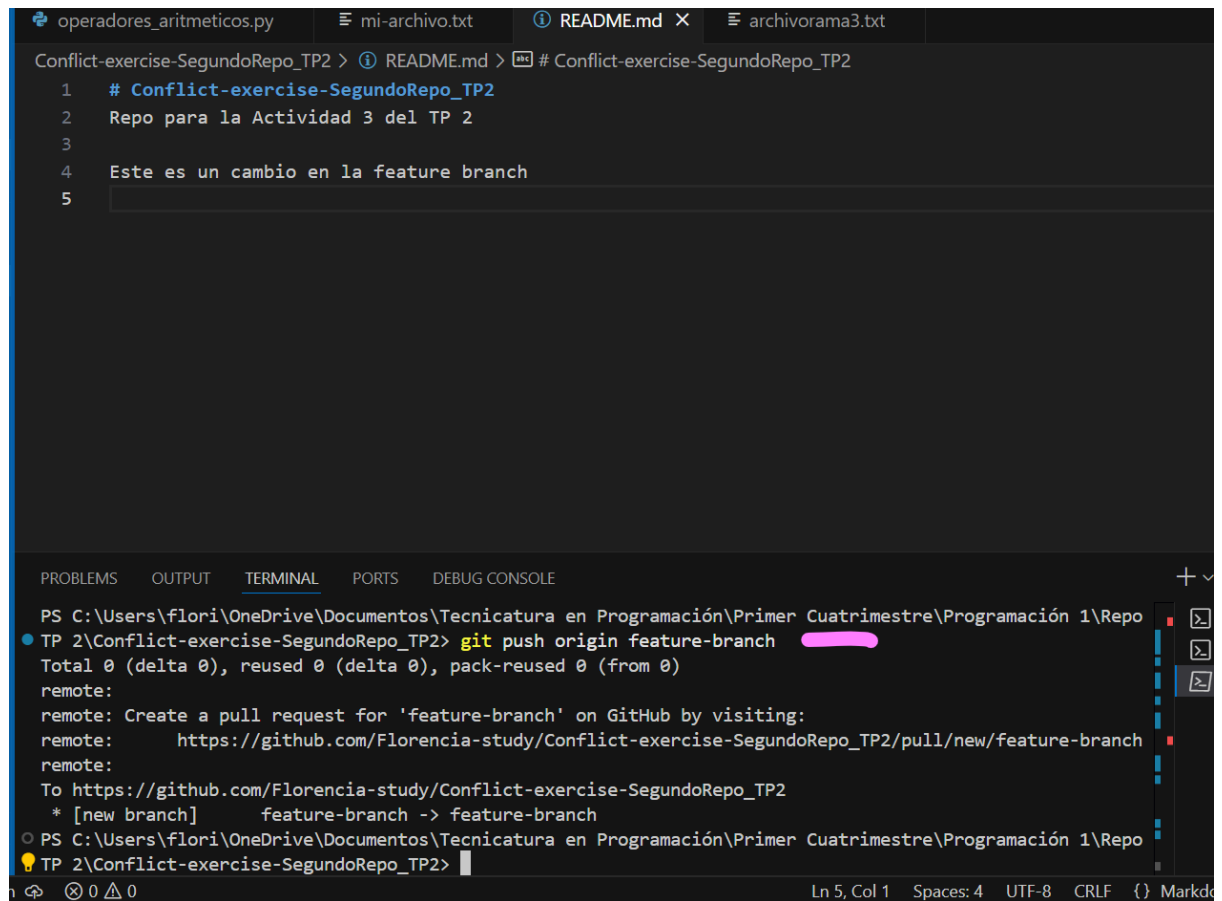
Subir main branch a github:



```
Conflict-exercise-SegundoRepo_TP2 > README.md > # Para solucionar el conflicto, uní las dos oraciones de las diferentes branches.
1 # Conflict-exercise-SegundoRepo_TP2
2 Repo para la Actividad 3 del TP 2
3
4 # Para solucionar el conflicto, uní las dos oraciones de las diferentes branches.
5 Línea diferente: Este es un cambio en la main branch, al que se le agrega texto de la feature branch
6
```

```
PS C:\Users\flori\OneDrive\Documentos\Tecnatura en Programación\Primer Cuatrimestre\Programación 1\Repo TP 2\Conflict-exercise-SegundoRepo_TP2> git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 896 bytes | 224.00 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/Florencia-study/Conflict-exercise-SegundoRepo_TP2
3f8ce80..a8b69fa main -> main
```

Subir feature-branch:



The image shows a Visual Studio Code editor window with four tabs: `operadores_aritmeticos.py`, `mi-archivo.txt`, `README.md` (active), and `archivorama3.txt`. The `README.md` file contains the following text:

```
1 # Conflict-exercise-SegundoRepo_TP2
2 Repo para la Actividad 3 del TP 2
3
4 Este es un cambio en la feature branch
5
```

Below the editor, the **TERMINAL** panel is open, showing the output of a `git push` command. The terminal text is as follows:

```
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2> 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/Florencia-study/Conflict-exercise-SegundoRepo_TP2/pull/new/feature-branch
remote:
To https://github.com/Florencia-study/Conflict-exercise-SegundoRepo_TP2
 * [new branch]      feature-branch -> feature-branch
PS C:\Users\flori\OneDrive\Documentos\Tecnicatura en Programación\Primer Cuatrimestre\Programación 1\Repo
TP 2\Conflict-exercise-SegundoRepo_TP2>
```

The status bar at the bottom indicates the cursor is at line 5, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and a Markdown file type.

Paso 8: Verificar en GitHub

github.com/Florencia-study/Conflict-exercise-SegundoRepo_TP2

Conflict-exercise-SegundoRepo_TP2 Public

main 2 Branches 0 Tags

Go to file Add file Code

About

Repo para la Actividad 3 del TP 2

Resolved merge conflict 4 minutes ago 4 Commits

README.md

README

Conflict-exercise-SegundoRepo_TP2

Repo para la Actividad 3 del TP 2

Para solucionar el conflicto, uní las dos oraciones de las diferentes branches.

Linea diferente: Este es un cambio en la main branch, al que se le agrega texto de la feature branch

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

github.com/Florencia-study/Conflict-exercise-SegundoRepo_TP2/tree/feature-branch

feature-branch 2 Branches 0 Tags

Go to file Add file Code

About

Repo para la Actividad 3 del TP 2

This branch is 2 commits behind main

Contribute

Florencia-study Added a line in feature-branch 13 minutes ago 2 Commits

README.md

Added a line in feature-branch

13 minutes ago

README

Conflict-exercise-SegundoRepo_TP2

Repo para la Actividad 3 del TP 2

Este es un cambio en la feature branch

Releases

No releases published


Create a new release


Packages


No packages published


Publish your first package

Commits


 main

 All users



 All time

 Commits on Apr 6, 2025


Resolved merge conflict

 Florence-study committed 11 minutes ago


a8b69fa


Added a line in main branch

 Florence-study committed 16 minutes ago

Added a line in feature-branch

 Florence-study committed 19 minutes ago

Initial commit

 Florence-study authored 34 minutes ago

Verified

3f8ce80

