

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.

Actividades

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

• ¿Qué es GitHub?

GitHub es una plataforma con almacenamiento basado en la nube que nos permite almacenar nuestros archivos, compartirlos, trabajar con otros usuarios y nos permite tener un seguimiento de los cambios que hemos realizados en dichos archivos, es decir nos permite tener un control de las versiones de nuestros archivos.

• ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio desde la consola de GitHub primero tenemos que crear una cuenta e iniciar sesión. Una vez allí vamos a Repositorio y hacemos click donde dice new, elegimos un nombre, agregamos una descripción y creamos el repositorio.

Para terminar de crear el repositorio es necesario crear al menos un archivo, esto se puede hacer mediante una terminal (es necesario tener Git instalado en la PC) o usando el Quick Setup que nos permite crear un nuevo archivo o subir un archivo desde nuestra computadora.

```
...or create a new repository on the command line

echo "# ddfsf sdf" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Laura-Ambrosio/ddfsf sdf.git
git push -u origin main
```



| | |
|--|--|
| Crear una rama en Git | git branch <nombre de la rama> |
| Cambiar a una rama en Git | git checkout <nombre de la rama> |
| Fusionar ramas (Ej: fusionar el contenido de la rama test con main) | 1° voy a la rama a la que le agregaré los cambios git checkout main 2° Fusiono la información con merge. git merge test |
| Crear un commit en Git | git commit -m "descripción del commit" |
| Enviar un commit a GitHub | git push origin <nombre_de la rama_a_enviar_el_commit> |

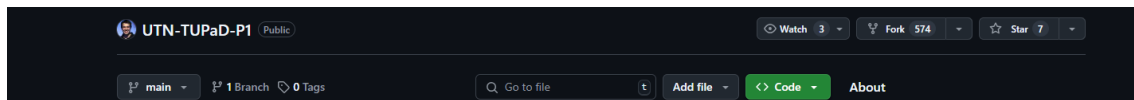
- **¿Qué es un repositorio remoto?**

Un repositorio remoto es el que está alojado en la nube, a diferencia del local que se encuentra en nuestra PC. En este caso el repositorio creado en GitHub es el repositorio remoto.

| | |
|---|--|
| Agregar un repositorio remoto a Git | 1° Si el repositorio ya está creado en GitHub git clone <url del repositorio> |
| Empujar cambios a un repositorio remoto | git push origin <nombre_de_la_rama> |
| Tirar de cambios de un repositorio remoto | git pull origin <nombre_de_la_rama> |

- **¿Qué es un fork de repositorio y cómo se crea?**

Un fork es un repositorio que se crea a partir de el repositorio de otro usuario. En GitHub podemos ver los repositorios públicos de otro usuario y crear una copia de ese.



Para crear un fork se presiona el botón que dice fork, se elige un nombre y se agrega una descripción. Una vez creado podremos modificar el repositorio como si fuera nuestro, y los cambios realizados solo se aplicarán a nuestra copia del repositorio, no al original.

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

Para enviar una solicitud, en la consola escribo:

```
git push origin <repository_name>
```

Luego en GitHub el repositorio al que le envíe el pull request va a aparecer un botón verde que dice Compare and pull request. Luego selecciono el Branch y agrego una descripción.

Presiono Create pull request y espero a que revisen los cambios que quiero agregar y si los aceptan o no. Si lo aceptan se hace un merge pull request (botón en GitHub).

- **¿Qué es un etiqueta en Git y cómo crearla?**

En Git las tags o etiquetas sirven para identificar puntos específicos en el historial de un repositorio.

Para crear una annotated tag:

```
git tag -a <nombre_tag> -m "comentario"
```

Esta etiqueta contiene información como el nombre de quien la creo, email, fecha, el mensaje, etc.

Para crear una lightweight tag:

```
git tag <nombre_tag>
```

Solo contiene el checksum del commit.

Un checksum es: “ la ejecución de un algoritmo dentro de un archivo que permite verificar y garantizar la integridad y seguridad de la información cuando es almacenada o compartida con otros usuarios en la red.”

Ambrosio Laura

- ¿Cómo enviar una etiqueta a GitHub?

Git no envía las tags de manera predeterminada cuando usamos el comando push, tiene que ser explicitado.

Para esto se usa el comando:

```
git push origin <tagname>
```

Para ver una lista de las tags en nuestro repositorio usamos:

```
git tag
```

- ¿Qué es un historial de Git? ¿Cómo podemos visualizarlo y buscar en el historial de Git? ¿Cómo se borra?

El historial de git almacena información sobre los commits que realizamos. Puede **visualizarse** usando el comando “git log”.

Para **buscar** es necesario agregar flags al comando git log. Por ejemplo:

```
git log --author= 'laura'
```

Es posible **borrar el historial** de git usando el comando git reset. Por ejemplo:

```
git reset
```

Saca del stage archivos y carpetas

```
git reset --hard HEAD~<numero>
```

Borra tantos commits como los indicados por el número que elegimos, contando desde el último commit realizado.

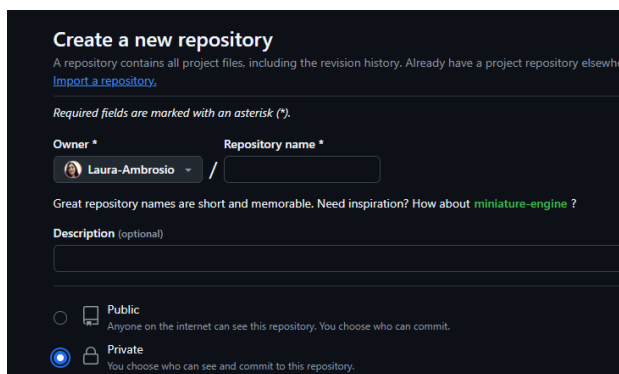
- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio que solo puede visualizar quien lo creo o aquellos usuarios autorizados.

- ¿Cómo crear un repositorio privado en GitHub?

Luego de iniciar sesión, en la parte derecha de la pantalla vemos un signo +, al hacer click allí encontramos la opción “New Repository”, allí podemos elegir si queremos un repositorio público o privado.

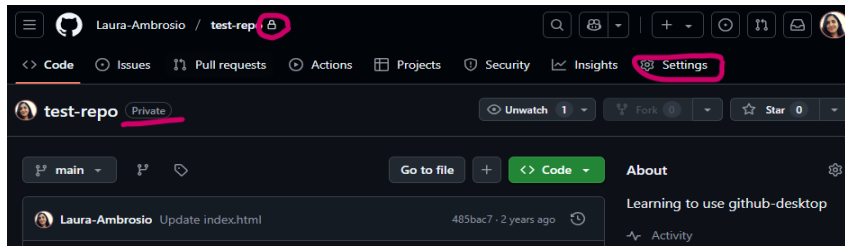
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?



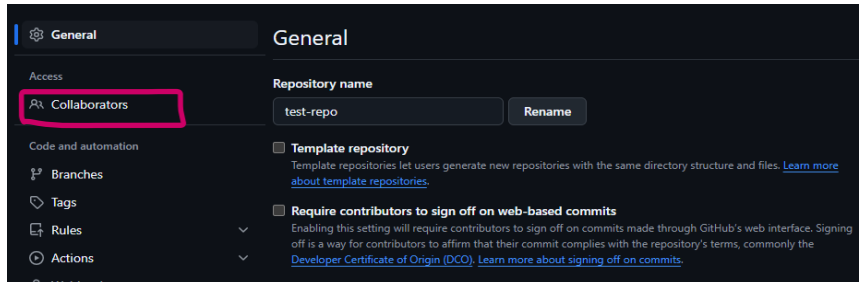
The screenshot shows the GitHub interface for creating a new repository. At the top, it says 'Create a new repository' and explains that a repository contains all project files, including the revision history. Below this, there's a link to 'Import a repository'. A note states 'Required fields are marked with an asterisk (*)'. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'Laura-Ambrosio'. The 'Repository name' field is empty. Below these, there's a suggestion: 'Great repository names are short and memorable. Need inspiration? How about [miniature-engine](#) ?'. There's a 'Description (optional)' text area. At the bottom, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.'

Ambrosio Laura

Para invitar a alguien primero abrimos nuestro repositorio privado y luego presionamos settings



Seleccionamos “Colaboratos”



Nos va a pedir que iniciemos sesion nuevamente y entonces nos mostrará las opciones de la pestaña Colaboradores. Hacemos click en “Add people” y nos aparecerá una barra de búsqueda, allí podemos ingresar el email, nombre completo o nombre de usuario de la cuenta de GitHub a la que le queremos dar permiso. Una vez que la encontramos hacemos click en “Add Repository”.

- ¿Qué es un repositorio público en GitHub?

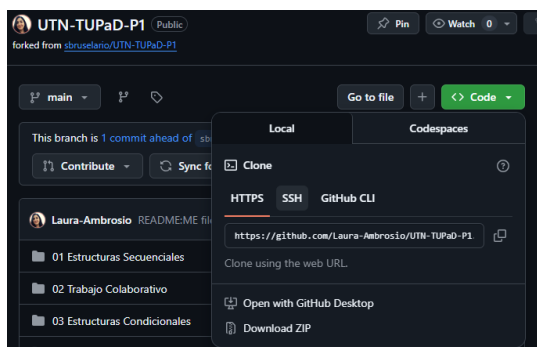
Un repositorio público contiene un proyecto que esta accesible para cualquier usuario. Otros usuarios pueden verlo y copiarlo y solicitar hacer cambios.

- ¿Cómo crear un repositorio público en GitHub?

Se siguen los mismos pasos que para crear un repositorio privado pero en la pantalla de creación, en la que colocamos el nombre del repositorio y la descripción, seleccionamos “Público”. Luego seguimos los pasos explicados en la pregunta “•¿Cómo crear un repositorio en GitHub?”.

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público ingresamos al repositorio presionamos el botón Code y allí encontramos la formas en las que podemos compartir el repositorio.

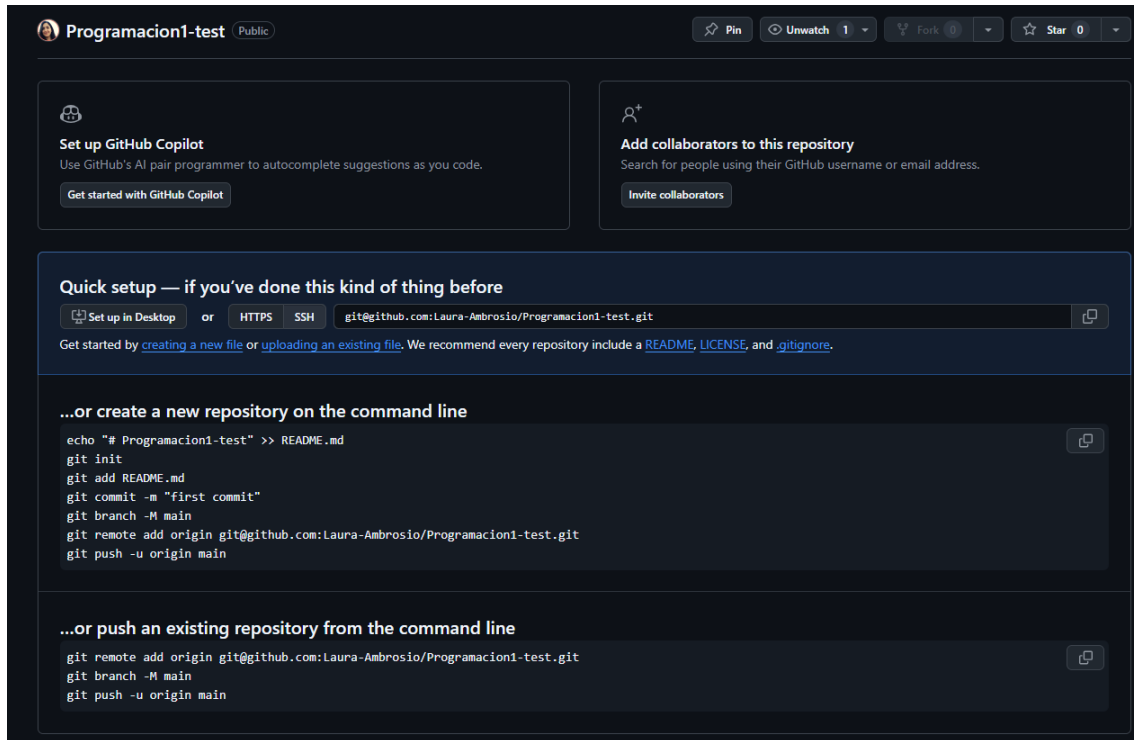


La opción HTTPS nos permite compartir la URL del repositorio, de esta manera otro usuario puede clonarlo.

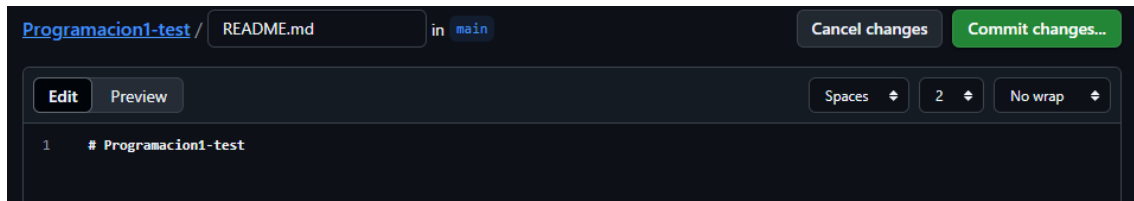
Ambrosio Laura

2) Realizar la siguiente actividad:

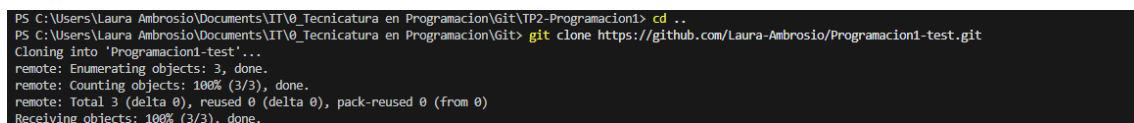
- Crear un repositorio.
- Dale un nombre al repositorio.
- Elije el repositorio sea público.



- Inicializa el repositorio con un archivo.

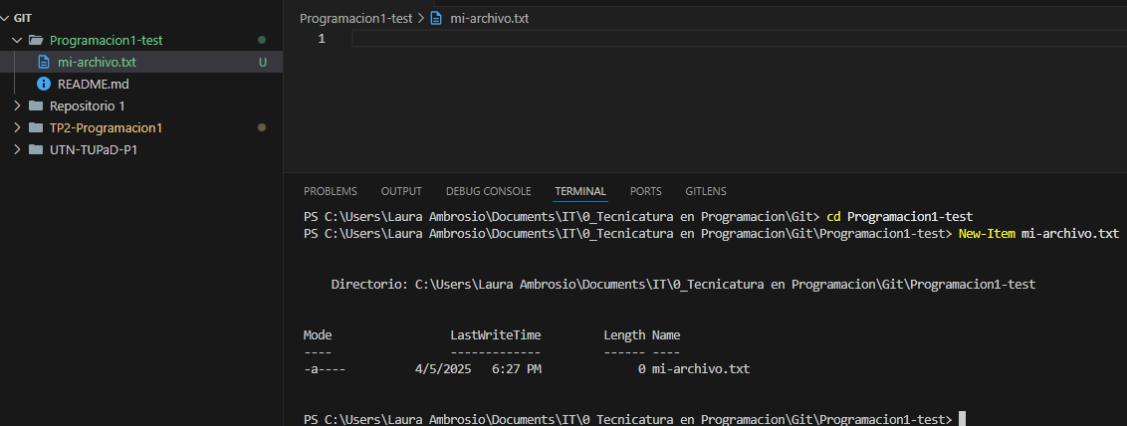


- Agregando un Archivo



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

Ambrosio Laura



Programacion1-test > mi-archivo.txt

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

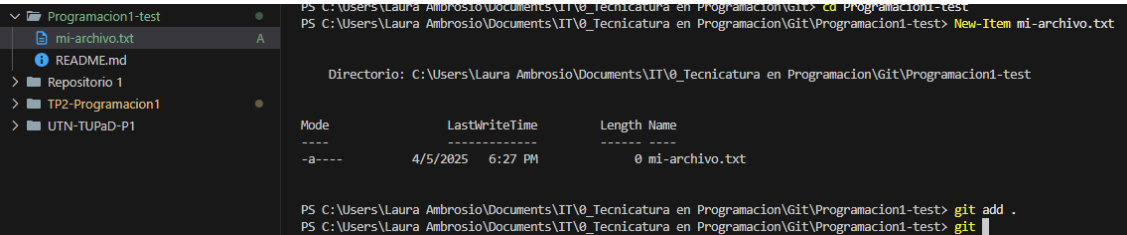
```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git> cd Programacion1-test
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> New-Item mi-archivo.txt
```

Directorio: C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test

| Mode | LastWriteTime | Length | Name |
|--------|------------------|--------|----------------|
| ---- | ----- | ----- | ---- |
| -a---- | 4/5/2025 6:27 PM | 0 | mi-archivo.txt |

PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test>

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



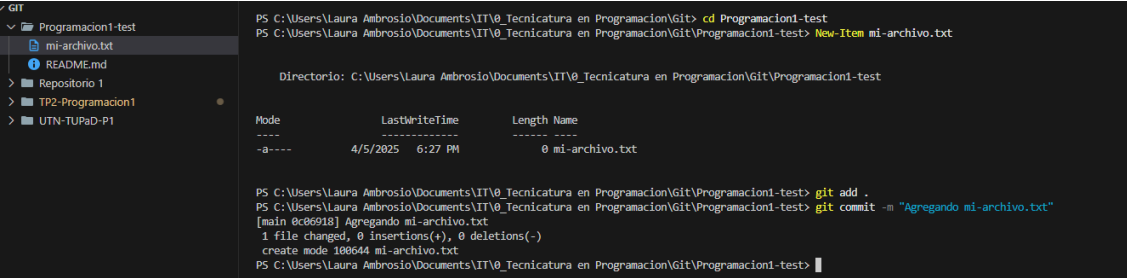
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git> cd Programacion1-test

PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> New-Item mi-archivo.txt

Directorio: C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test

| Mode | LastWriteTime | Length | Name |
|--------|------------------|--------|----------------|
| ---- | ----- | ----- | ---- |
| -a---- | 4/5/2025 6:27 PM | 0 | mi-archivo.txt |

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git add .
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git
```



PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git> cd Programacion1-test

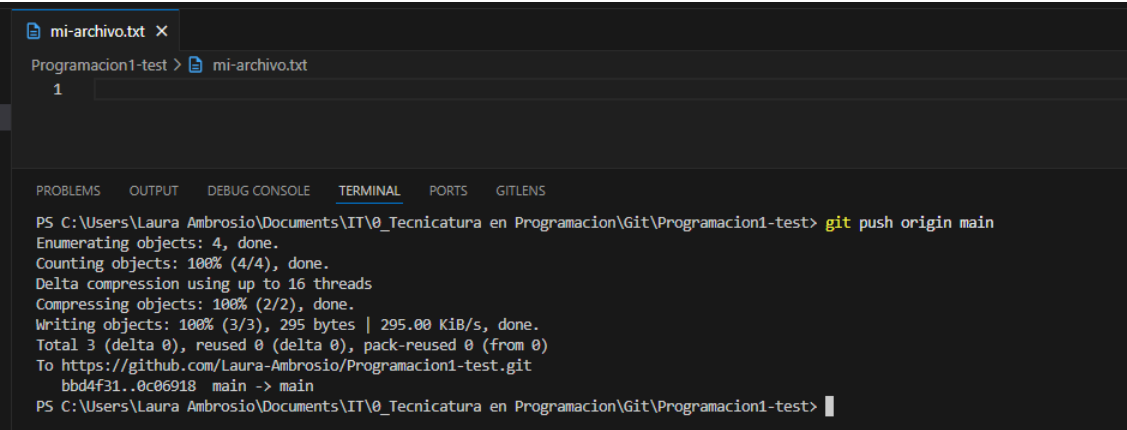
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> New-Item mi-archivo.txt

Directorio: C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test

| Mode | LastWriteTime | Length | Name |
|--------|------------------|--------|----------------|
| ---- | ----- | ----- | ---- |
| -a---- | 4/5/2025 6:27 PM | 0 | mi-archivo.txt |

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git add .
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git commit -m "Agregando mi-archivo.txt"
[main 0c06918] Agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test>
```

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



mi-archivo.txt X

Programacion1-test > mi-archivo.txt

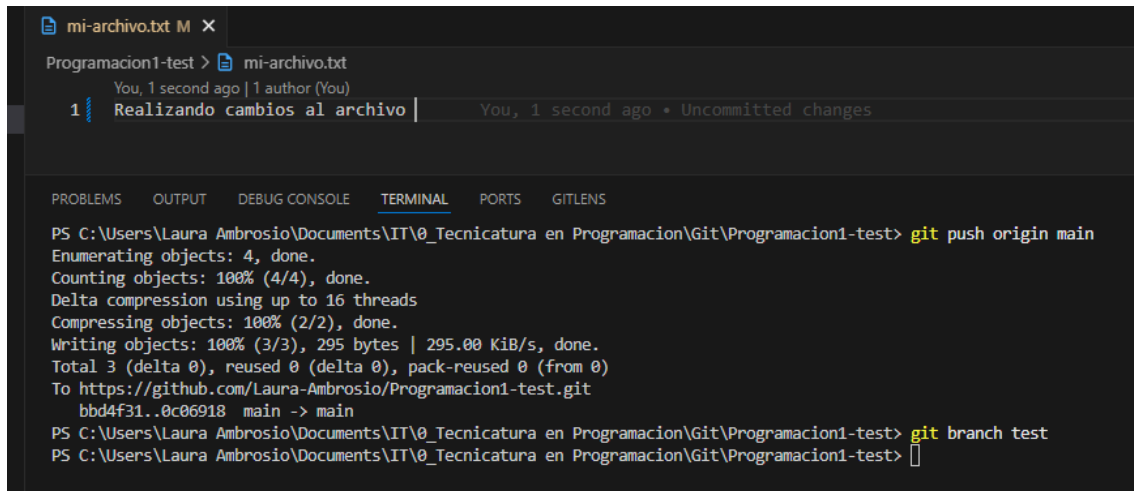
1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Laura-Ambrosio/Programacion1-test.git
 bbd4f31..0c06918 main -> main
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test>
```

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

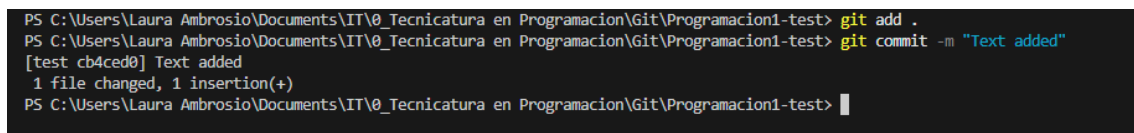
Ambrosio Laura



The screenshot shows a VS Code editor with a file named `mi-archivo.txt` open. The file content is `Realizando cambios al archivo`. The terminal window at the bottom shows the following commands and output:

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Laura-Ambrosio/Programacion1-test.git
 bbd4f31..0c06918  main -> main
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git branch test
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test>
```

○ Subir la Branch



The screenshot shows a VS Code editor with a terminal window displaying the following commands and output:

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git add .
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test> git commit -m "Text added"
[test cb4ced0] Text added
1 file changed, 1 insertion(+)
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\Programacion1-test>
```

Ambrosio Laura

3) 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 *

Laura-Ambrosio

Repository name *

TP2-Programacion1

TP2-Programacion1 is available.

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

Description (optional)

Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN

☒ 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

TP2-Programacion1

Public

Pin

Unwatch 1

Fork 0

Star 0

main

Go to file

Code

Laura-Ambrosio

Initial commit

1f2e33d · now

README.md

Initial commit

now

README

TP2-Programacion1

Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN

About

Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN

Readme

Activity

0 stars

1 watching

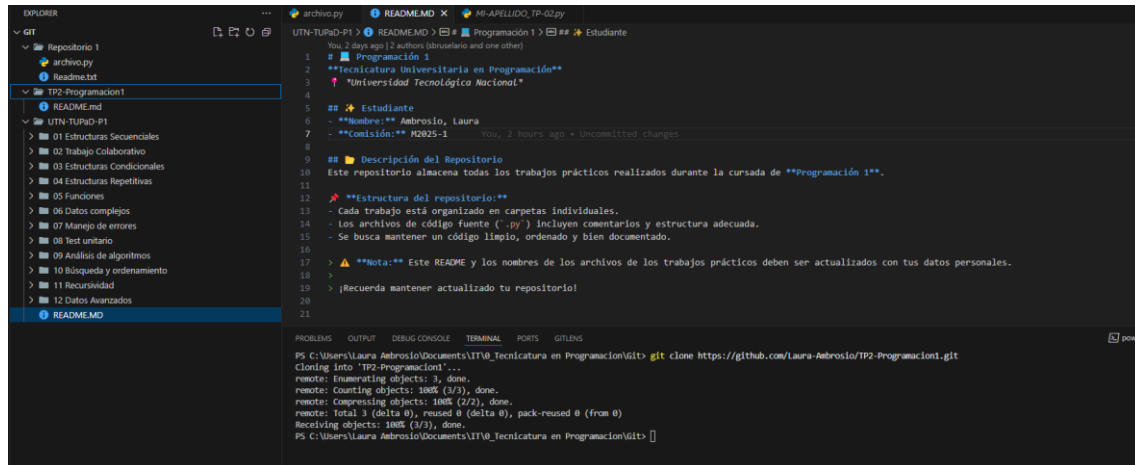
0 forks

Releases

No releases published

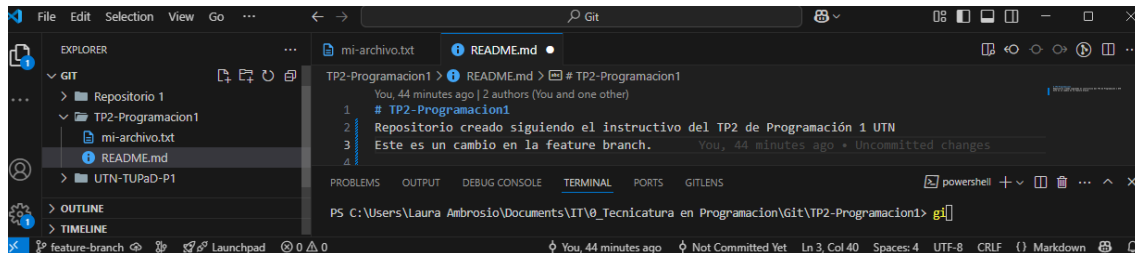
Create a new release

Paso 2: Clonar el repositorio a tu máquina local

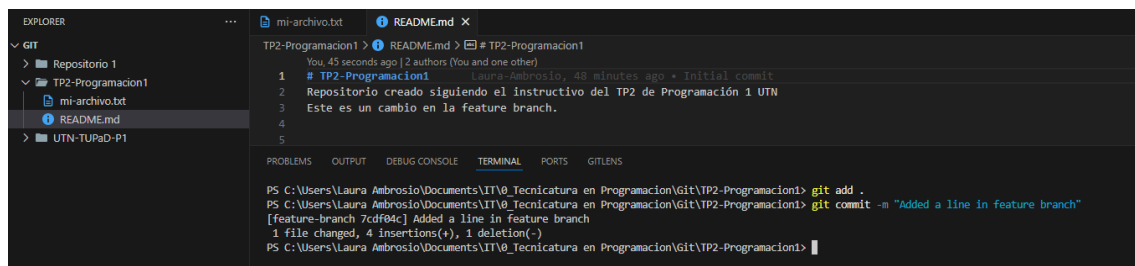


Paso 3: Crear una nueva rama y editar un archivo

- 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,



- Guarda los cambios y haz un commit:



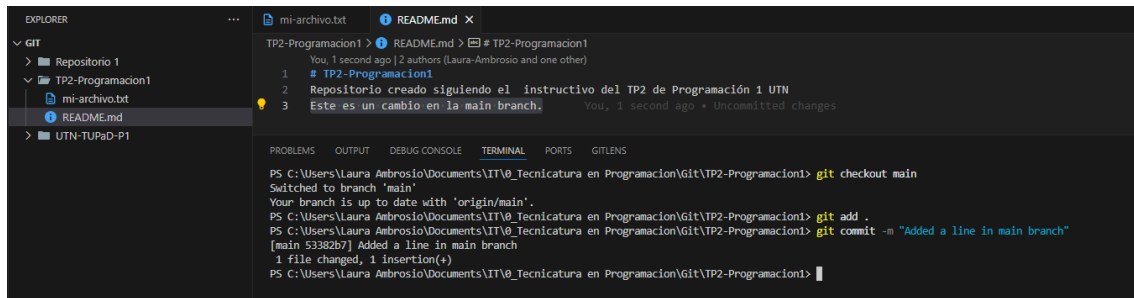
Paso 4: Volver a la rama principal y editar el mismo archivo

- Cambia de vuelta a la rama principal (main)
- Edita el archivo README.md de nuevo, añadiendo una línea diferente:
- Guarda los cambios y haz un commit:

git add README.md

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

Ambrosio Laura



```
EXPLORER
  GIT
  Repositorio 1
  TP2-Programacion1
    mi-archivo.txt
    README.md
  UTN-TUPaD-P1

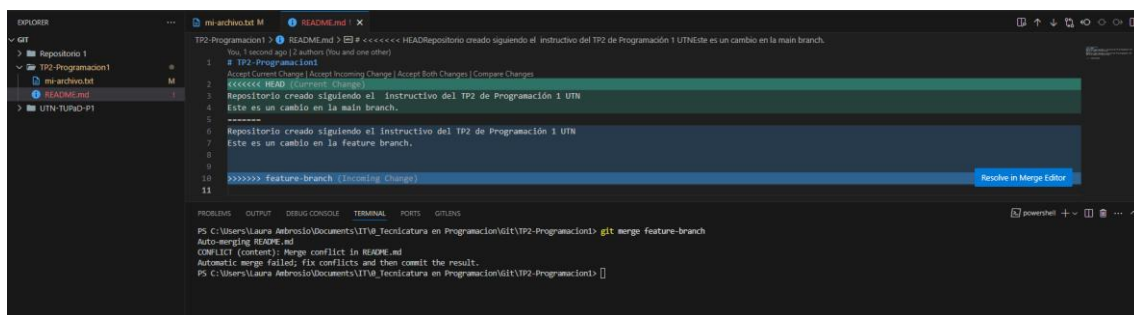
TP2-Programacion1 > README.md > # TP2-Programacion1
You, 1 second ago | 2 authors (Laura-Ambrosio and one other)
1 # TP2-Programacion1
2
3 Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN
4 Este es un cambio en la main branch. You, 1 second ago • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> git add .
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> git commit -m "Added a line in main branch"
[main 53382b7] Added a line in main branch
1 file changed, 1 insertion(+)
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1>
```

Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de a feature-branch en la rama main:
- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.



```
EXPLORER
  GIT
  Repositorio 1
  TP2-Programacion1
    mi-archivo.txt
    README.md
  UTN-TUPaD-P1

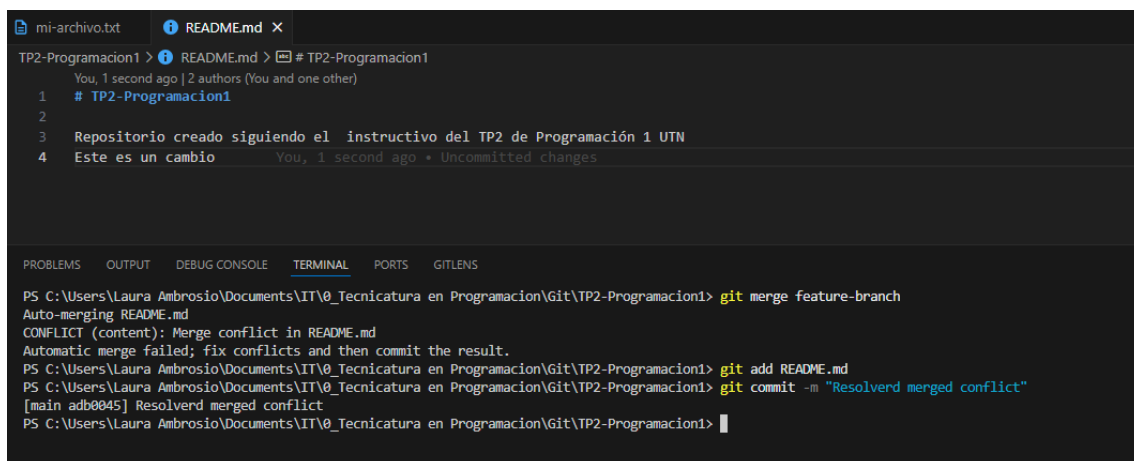
TP2-Programacion1 > README.md > # <<<<<< HEADRepositorio creado siguiendo el instructivo del TP2 de Programación 1 UTNEste es un cambio en la main branch.
You, 1 second ago | 2 authors (You and one other)
1 # TP2-Programacion1
2
3 Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN
4 Este es un cambio en la main branch.
5
6 Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN
7 Este es un cambio en la feature branch.
8
9
10 >>>>>> feature-branch (Incoming Changes)
11
Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> 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\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1>
```

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto.
- Decide cómo resolver el conflicto.
- Añade el archivo resuelto y completa el merge



```
mi-archivo.txt README.md X

TP2-Programacion1 > README.md > # TP2-Programacion1
You, 1 second ago | 2 authors (You and one other)
1 # TP2-Programacion1
2
3 Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN
4 Este es un cambio You, 1 second ago • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> 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\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> git add README.md
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1> git commit -m "Resolved merged conflict"
[main adb0045] Resolved merged conflict
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnatura en Programacion\Git\TP2-Programacion1>
```

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

Ambrosio Laura

```
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnica en Programacion\Git\TP2-Programacion1> git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.20 KiB | 1.20 MiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Laura-Ambrosio/TP2-Programacion1.git
0f4c463..adb0045  main -> main
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnica en Programacion\Git\TP2-Programacion1> 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/Laura-Ambrosio/TP2-Programacion1/pull/new/feature-branch
remote:
To https://github.com/Laura-Ambrosio/TP2-Programacion1.git
* [new branch]      feature-branch -> feature-branch
PS C:\Users\Laura Ambrosio\Documents\IT\0_Tecnica en Programacion\Git\TP2-Programacion1> |
```

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.

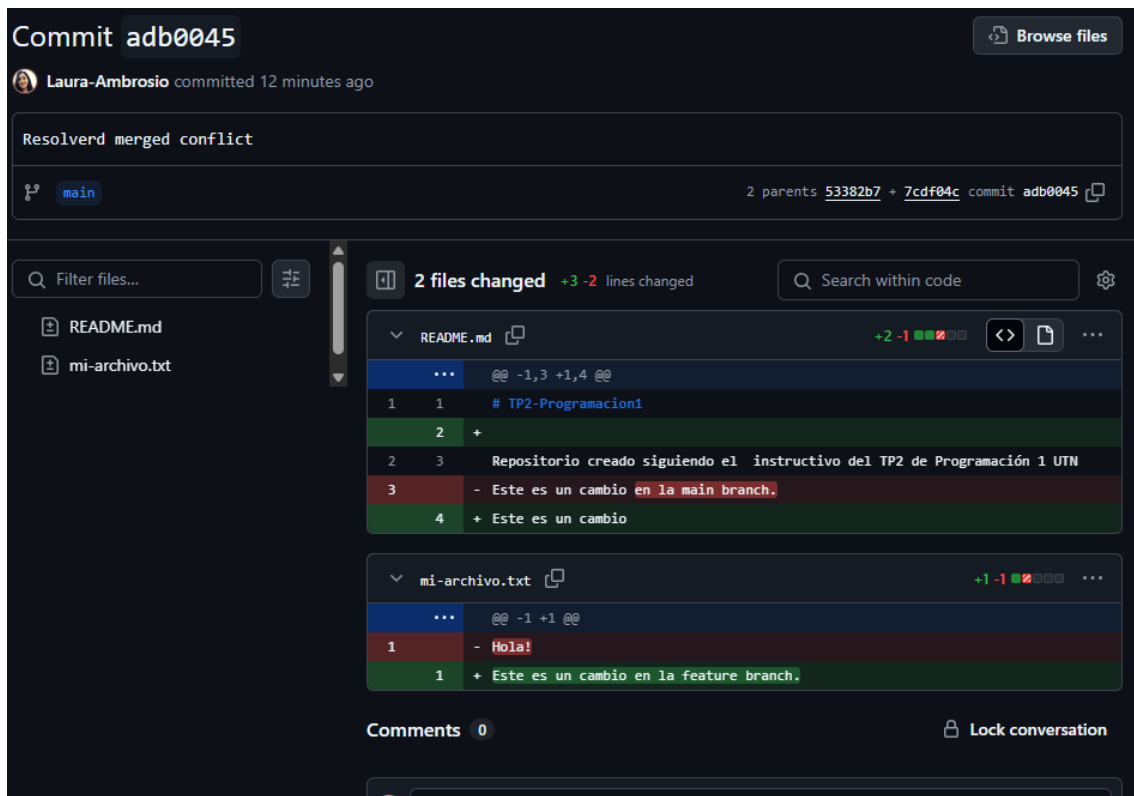


The screenshot shows the GitHub interface for the repository 'TP2-Programacion1'. The file 'README.md' is selected, showing a preview of the content. The commit history indicates a merge of 'adb0045' 11 minutes ago. The README content is as follows:

```
TP2-Programacion1

Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN Este es un cambio
```

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



The screenshot shows the GitHub commit page for commit 'adb0045' by Laura-Ambrosio, committed 12 minutes ago. The commit message is 'Resolverd merged conflict'. The diff shows changes to two files:

- README.md**: Changes include adding a new line (line 2) and modifying line 3. The diff shows:

```

... @@ -1,3 +1,4 @@
1 1 # TP2-Programacion1
2 +
2 3 Repositorio creado siguiendo el instructivo del TP2 de Programación 1 UTN
3 - Este es un cambio en la main branch.
4 + Este es un cambio

```
- mi-archivo.txt**: Changes include removing a line (line 1) and adding a new line (line 1). The diff shows:

```

... @@ -1 +1 @@
1 - Hola!
1 + Este es un cambio en la feature branch.

```

The commit page also shows a list of files (README.md, mi-archivo.txt) and a 'Comments' section with 0 comments.