

## Trabajo Práctico 2: Git y GitHub

### Actividades

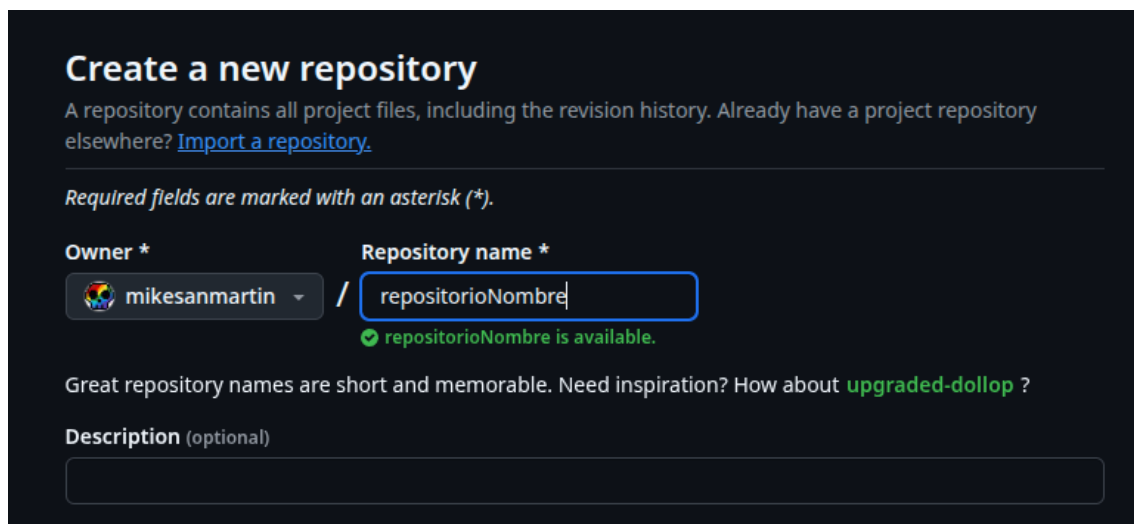
1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada

(Desarrollar las respuestas) :

- ¿Qué es GitHub?

Plataforma donde se alojan y comparten repositorios.

- ¿Cómo crear un repositorio en GitHub?

The image shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' in bold. Below that, a paragraph explains that a repository contains all project files and revision history, with a link to 'Import a repository' if one already exists. A note states 'Required fields are marked with an asterisk (\*)'. The 'Owner' field is set to 'mikesanmartin' with a dropdown arrow. The 'Repository name' field contains 'repositorioNombre' and is highlighted with a blue border. Below the name field, a green checkmark indicates 'repositorioNombre is available.' A suggestion for 'upgraded-dollop' is shown. The 'Description (optional)' field is empty.

(de forma local) ejecutando el comando "git init" en la terminal desde la ubicación del proyecto. En GitHub ingresando a <https://github.com/new> e indicando los datos de un nuevo repo.

- ¿Cómo crear una rama en Git?

En la terminal, posicionado sobre el repositorio del proyecto, utilizar el comando: "git branch nombreBranchNuevo", para generar una nueva rama.

También es posible utilizar el comando: "git checkout -b nombreBranchNuevo", esto no solo genera la rama nueva sino que nos posiciona sobre la misma.

- ¿Cómo cambiar a una rama en Git?

Se utiliza en terminal el comando: "git checkout branch", donde branch es la rama a la cual me quiero dirigir.

- ¿Cómo fusionar ramas en Git?

En la terminal, desde la una rama, debo ingresar: "git merge branch", donde branch es la rama que deseo fusionar con mi rama actual seleccionada.

- ¿Cómo crear un commit en Git?

Inicialmente agregar los cambios realizados en el repositorio mediante: "git add .", y luego ingresar el comando: "git commit -m 'mensaje'", donde el mensaje puede ser alguna descripción breve del commit.

- ¿Cómo enviar un commit a GitHub?

Una vez realizado el commit, se puede enviar mediante el comando: "git push -u origin master", acá se informo master como rama genérica, cuando en realidad puede ser la rama en la cual estemos trabajando y tenga otro nombre.

- ¿Qué es un repositorio remoto?

Un repositorio que se encuentra alojado en GitHub (o GitLab o soluciones de terceros al manejo del versionado).

- ¿Cómo agregar un repositorio remoto a Git?

Utilizando el comando: "git remote add origin url" siendo la URL la del repositorio alojado en Github

- ¿Cómo empujar cambios a un repositorio remoto?

Una vez informado el origen remoto, se debe utilizar "git push"

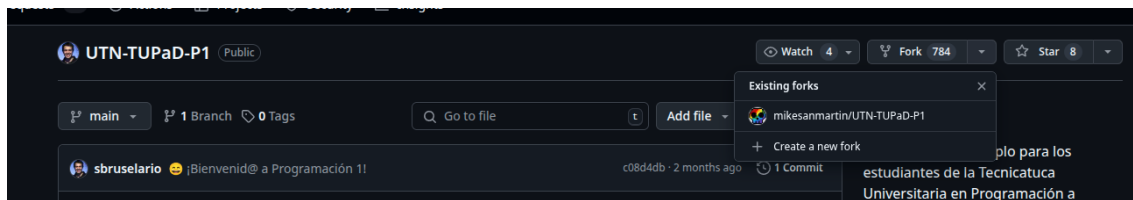
- ¿Cómo tirar de cambios de un repositorio remoto?

Se debe utilizar el comando: "git pull origin branch", informando la rama de la cual traer los datos.

- ¿Qué es un fork de repositorio?

Es una copia independiente del repositorio, distinta al branch/rama.

- ¿Cómo crear un fork de un repositorio?



Desde GitHub, seleccionado la opción de crear un nuevo fork, para traer dicho fork al perfil propio.

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

Una vez realizado el push de cambios, se genera una PR (pull request) en la página de GitHub, donde se incluye una breve descripción de los cambios propuestos y evidencia de los mismos, adicionalmente GitHub muestra los archivos que han sido modificados dentro de esa rama.

- ¿Cómo aceptar una solicitud de extracción?

En GitHub, aceptados los cambios de la pull request se procede a mergear los cambios de los mismos integrándolos en la rama indicada (que puede o no ser la rama principal del proyecto).

- ¿Qué es un etiqueta en Git?

Una etiqueta es un marcador que apunta a un commit en específico.

- ¿Cómo crear una etiqueta en Git?

Utilizando el comando: "git tag nombreEtiqueta", opcionalmente se puede incluir "-m" para incluir un mensaje referencial.

- ¿Cómo enviar una etiqueta a GitHub?

Utilizando el comando: "git push origin nombreDeLaEtiqueta"

- ¿Qué es un historial de Git?

Es el registro de todos los commits/cambios realizados en el proyecto.

- ¿Cómo ver el historial de Git?

Utilizando el comando: "git log", pero también puede utilizarse "git reflog" para obtener mayores detalles.

- ¿Cómo buscar en el historial de Git?

Utilizando el comando "git log"

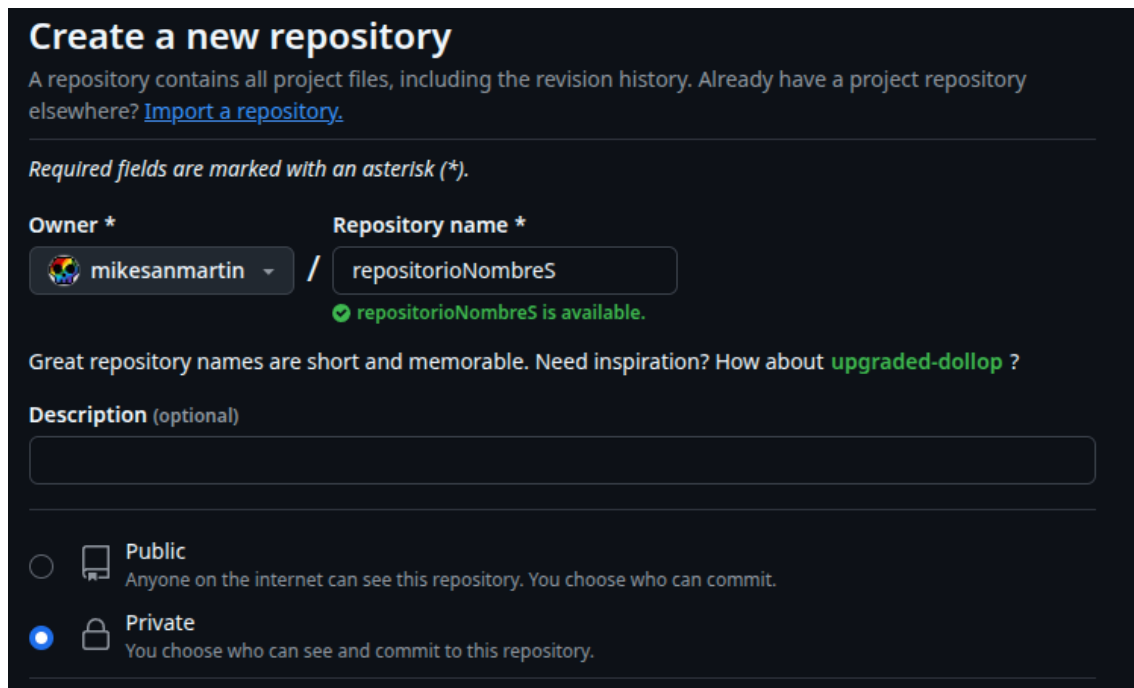
- ¿Cómo borrar el historial de Git?

Utilizando el comando: "git reset --hard HEAD~N", donde N es el numero de commits a eliminar.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio no accesible a usuarios que no esten previamente autorizados.

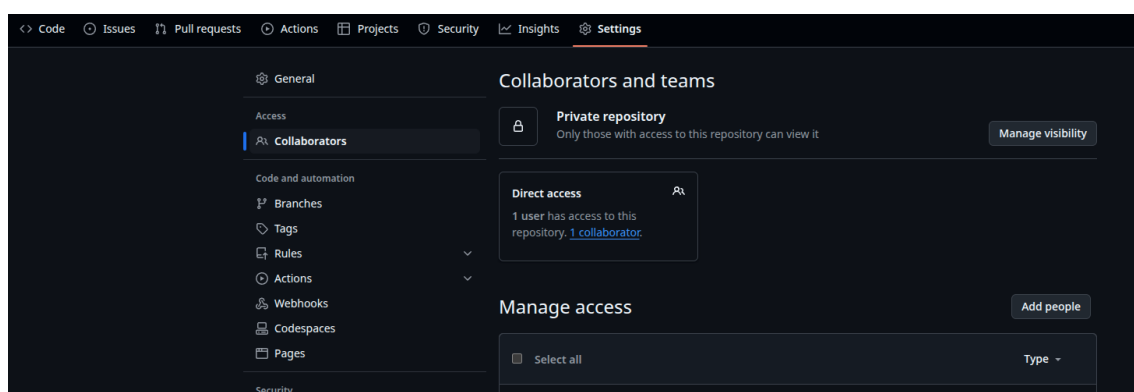
- ¿Cómo crear un repositorio privado en GitHub?



The screenshot shows the 'Create a new repository' form on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, it asks if the user already has a project repository elsewhere and provides 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' section shows a dropdown menu with 'mikesanmartin' selected. The 'Repository name' section has a text input field containing 'repositorioNombreS', which is marked as available with a green checkmark. Below these fields, there is a suggestion for a great repository name: 'upgraded-dollop'. A 'Description (optional)' text area is provided. At the bottom, there are two radio buttons for 'Public' and 'Private'. The 'Private' option is selected, indicating that only those with access can view the repository.

En la creacion del repositorio se puede informar si sera publico o privado.

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



The screenshot shows the 'Settings' page for a repository on GitHub. The left sidebar contains a list of settings categories: General, Access, Collaborators, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Codespaces, Pages, and Security. The 'Collaborators' section is currently selected. The main content area is titled 'Collaborators and teams' and shows that the repository is a 'Private repository'. It indicates that only those with access to the repository can view it. Below this, there is a section for 'Direct access' which shows that 1 user has access to the repository. At the bottom, there is a 'Manage access' section with a button to 'Add people' and a table to manage access.

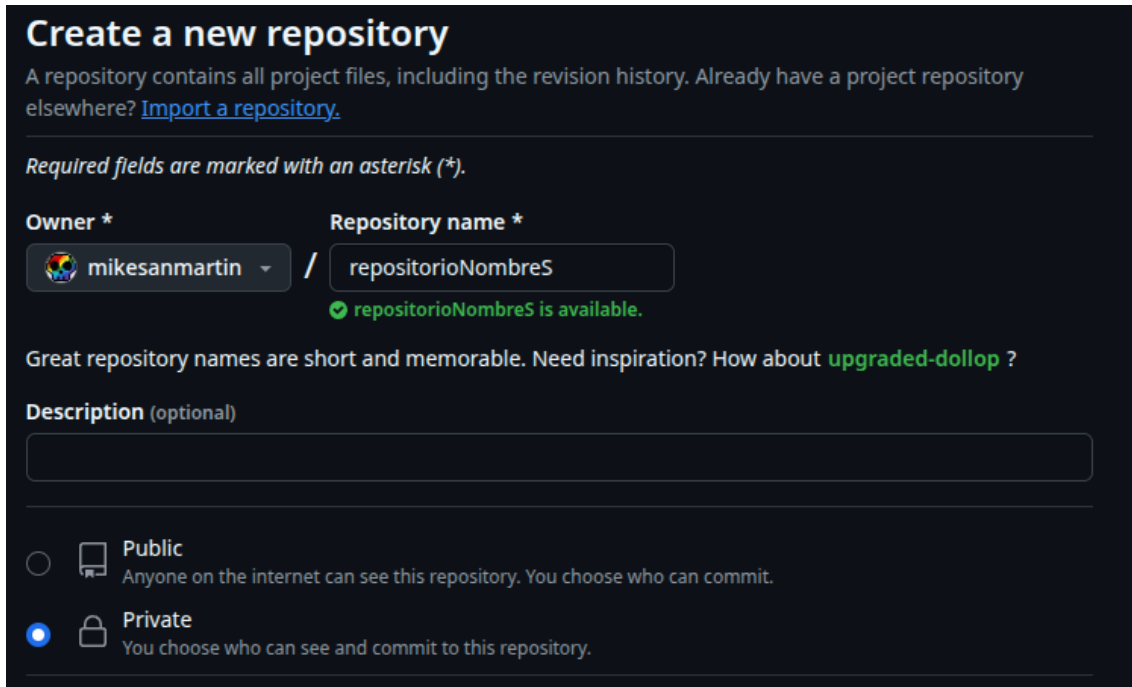
Dentro de los Settings del proyecto en GitHub se debe informar el nombre de usuario el cual sera invitado como colaborador, el mismo usuario debera aceptar tambien dicha solicitud.

- ¿Qué es un repositorio público en GitHub?

Un repositorio que, a diferencia del privado, permite la colaboración directa sin necesidad de autorizaciones y permite acceder al código libremente.

- ¿Cómo crear un repositorio público en GitHub?

Al crear un repositorio, se debe informar si el mismo será público o privado



The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' in bold. Below that, a note explains that a repository contains all project files and revision history, and offers a link to 'Import a repository' if the user already has one elsewhere. A line of text states 'Required fields are marked with an asterisk (\*)'. The form has two main sections: 'Owner \*' and 'Repository name \*'. The 'Owner' field shows a dropdown menu with 'mikesanmartin' selected. The 'Repository name' field contains 'repositorioNombreS' and has a green checkmark below it indicating 'repositorioNombreS is available'. Below these fields, a tip suggests that great repository names are short and memorable, and offers 'upgraded-dollop' as an example. There is 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.'

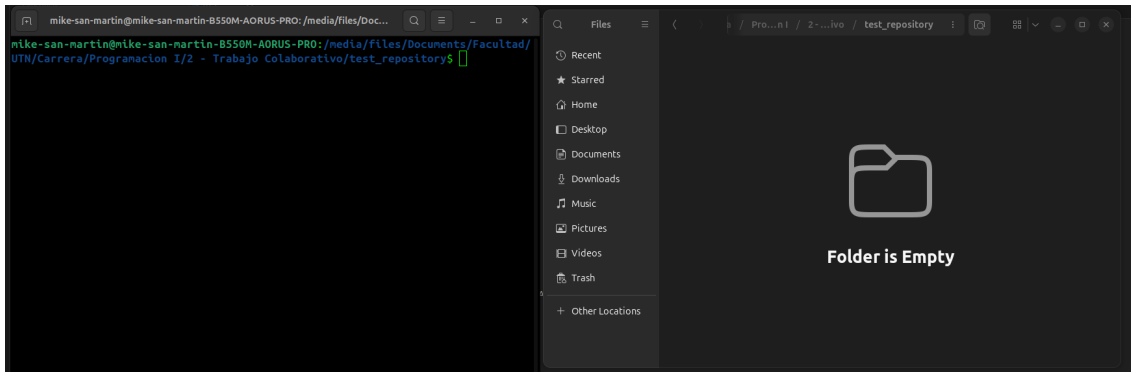
- ¿Cómo compartir un repositorio público en GitHub?

Mediante la URL del mismo.

2) Realizar la siguiente actividad:

Crear un repositorio:

Creo la carpeta test\_repository a nivel local y accedo a ella en la terminal:



Corremos el comando git init:

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO: /media/files/Doc...
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git init
Reinitialized existing Git repository in /media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository/.git/
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Podemos verificar en terminal que se generaron los archivos:

```
Reinitialized existing Git repository in /media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository/.git/
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ ls -la
total 4
drwxrwxrwx 1 mike-san-martin mike-san-martin 144 May  3 14:57 .
drwxrwxrwx 1 mike-san-martin mike-san-martin 4096 May  3 14:56 ..
drwxrwxrwx 1 mike-san-martin mike-san-martin 448 May  3 15:03 .git
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Genero archivo mi-archivo.txt

```
drwxrwxrwx 1 mike-san-martin mike-san-martin 144 May  3 14:57 .
drwxrwxrwx 1 mike-san-martin mike-san-martin 4096 May  3 14:56 ..
drwxrwxrwx 1 mike-san-martin mike-san-martin 448 May  3 15:03 .git
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ touch mi-archivo.txt
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

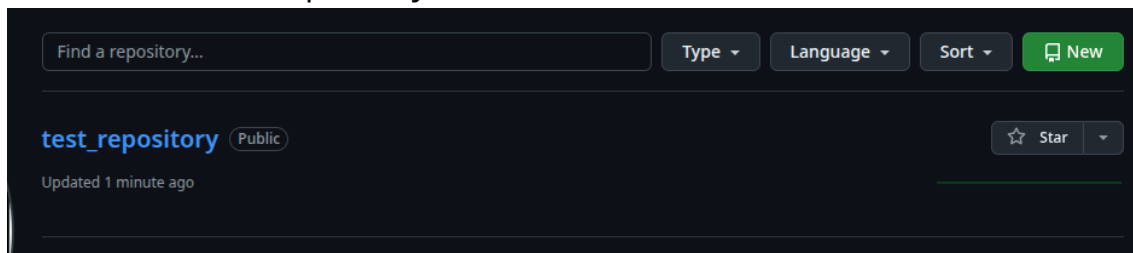
See agrego mediante commit el archivo:

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git add .
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git commit -m "agregando mi-archivo.txt"
[master (root-commit) 529418f] agregando mi-archivo.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mi-archivo.txt
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Se realizo el push:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git push
Everything up-to-date
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Se verifica el test\_repository en GitHub



Se crea nuevo branch y se switchea a la misma:

```
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git checkout -b "new-branch"
Switched to a new branch 'new-branch'
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Se creo nuevo archivo y se commiteo en la branch:

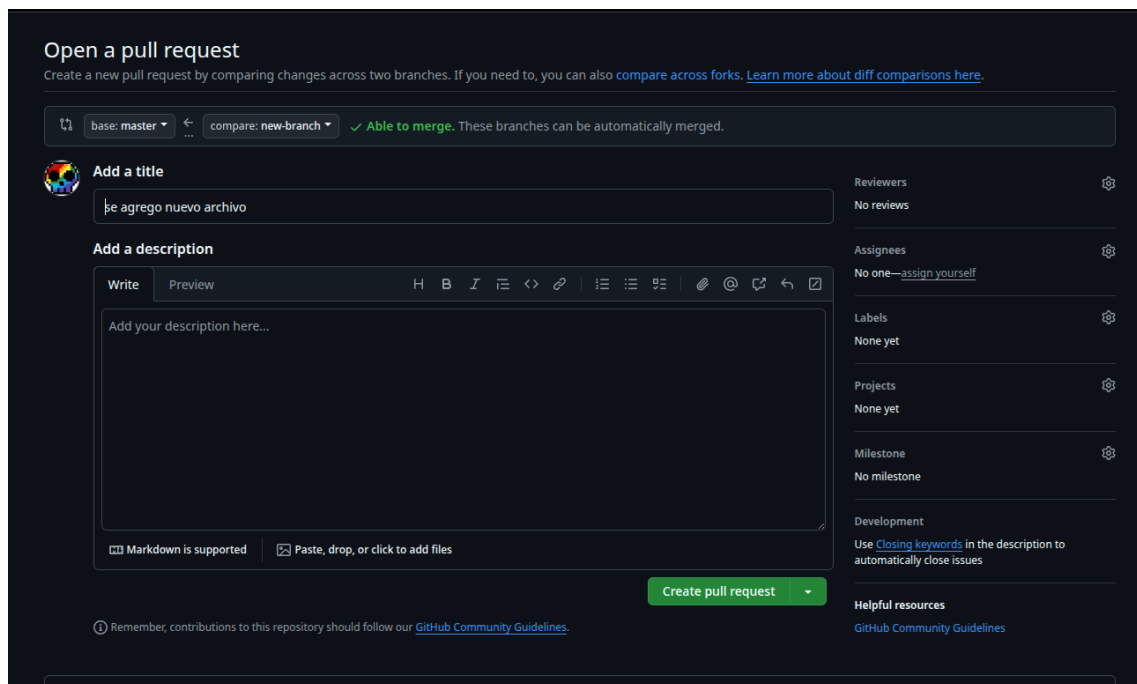
```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ touch mi-nuevo-archivo.txt
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git status
On branch new-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    mi-nuevo-archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git add .
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git commit -m "se a grego nuevo archivo"
[new-branch 963cf73] se agrego nuevo archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-nuevo-archivo.txt
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
```

Se pusheo cambio en new-branch

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$ git push --set-upst
ream origin new-branch
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 264 bytes | 264.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'new-branch' on GitHub by visiting:
remote:   https://github.com/mikesanmartin/test_repository/pull/new/new-branch
remote:
To https://github.com/mikesanmartin/test_repository.git
* [new branch]      new-branch -> new-branch
branch 'new-branch' set up to track 'origin/new-branch'.
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository$
```

Verificando la URL informada por la terminal, nos dirige directo a generar una nueva PR:



3) Actividad:

Paso 1:

Creo el nuevo 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 \*

 mikesanmartin ▾

Repository name \*

/ test\_repository\_2

✔ test\_repository\_2 is available.

Great repository names are short and memorable. Need inspiration? How about **expert-potato** ?

Description (optional)

Ejercicio 3, Trabajo Practico 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](#).

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

Paso 2:

se clona localmente el repositorio:

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
● Carrera/Programacion I/2 - Trabajo Colaborativo$ git clone https://github.com/mikesa
nmartin/test_repository_2.git
Cloning into 'test_repository_2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
○ Carrera/Programacion I/2 - Trabajo Colaborativo$
```

Ingreso al repositorio clonado

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
● Carrera/Programacion I/2 - Trabajo Colaborativo$ cd test_repository_2
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
○ Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

Ingreso a la nueva branch y cambio el archiv README.md

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
● Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git status
On branch feature-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
● Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git add README.md
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
● Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git commit -m "ad
ded line in feature-branch"
[feature-branch e31de73] added line in feature-branch
1 file changed, 2 insertions(+)
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
○ Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

Se cambia a branch main y se realiza modificacion sobre README.md

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git add README.md
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git commit -m "ad
ded line in main branch"
[main bab1d7a] added line in main branch
1 file changed, 2 insertions(+)
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

Ejecuto git merge feature-branch

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ 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.
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
• Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

```
① README.md ! ✕
① README.md > # Este cambio fue realizado en feature-branch
1 # test_repository_2
2 Ejercicio 3, Trabajo Practico 2
3
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<<<< HEAD (Current Change)
5 # Línea agregada en main branch
6 =====
7 # Este cambio fue realizado en feature-branch
8 >>>>>> feature-branch (Incoming Change)
9
```

Resuelvo conflicto:

```
Result: README.md
1 # test_repository_2
2 Ejercicio 3, Trabajo Practico 2
3
Incoming | Remove Incoming
4 # Este cambio fue realizado en feature-branch

Complete Merge
```

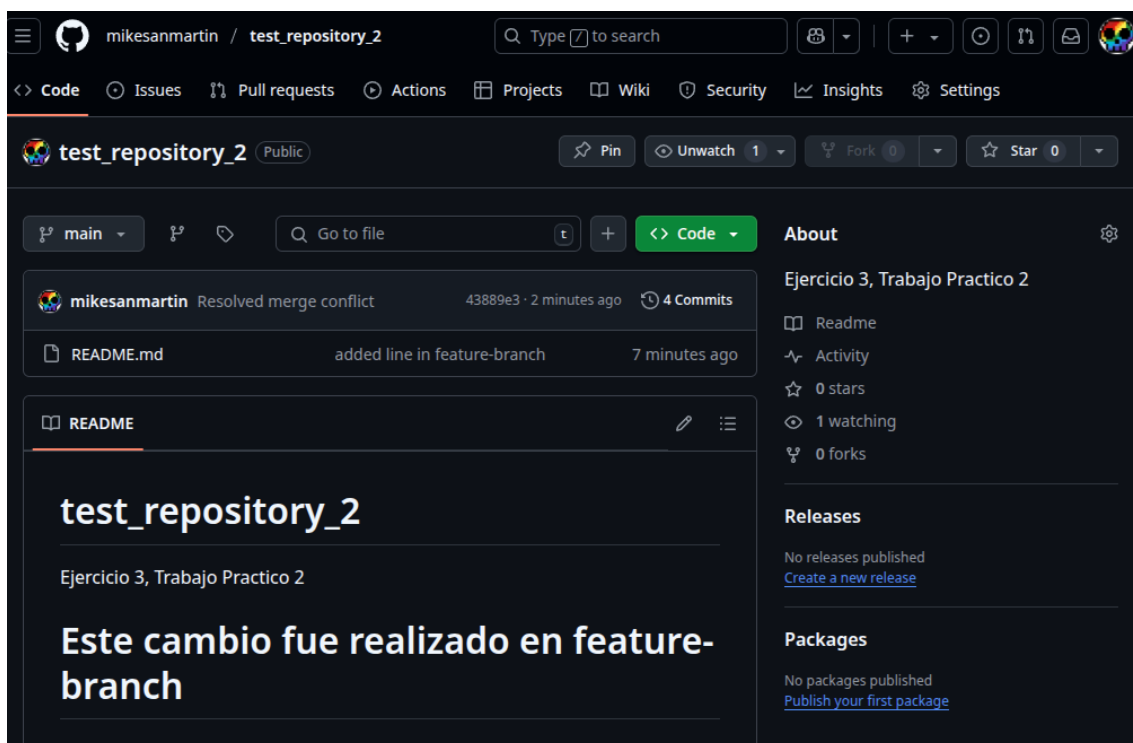
Se genero commit nuevo por la resolucion del conflicto en merge

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git add README.m
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git commit -m "R
solved merge conflict"
[main 43889e3] Resolved merge conflict
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

Pusheo los cambios:

```
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$ git push origin m
ain
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 722 bytes | 722.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/mikesanmartin/test_repository_2.git
bddfbc4..43889e3 main -> main
mike-san-martin@mike-san-martin-B550M-AORUS-PRO:/media/files/Documents/Facultad/UTN/
Carrera/Programacion I/2 - Trabajo Colaborativo/test_repository_2$
```

Se verifica cambio en GitHub



Verifico el historial

```
commit 43889e31075862d7e0e2f014f288fd4b9b044272 (HEAD -> main, origin/main, origin/H  
EAD)  
Merge: bab1d7a e31de73  
Author: Michel San Martin <mikeemmanuel91@gmail.com>  
Date: Sat May 3 15:42:00 2025 -0300  
  
Resolved merge conflict  
  
commit bab1d7a6fbee1834e4e38676e1bcf77e64ffa5  
Author: Michel San Martin <mikeemmanuel91@gmail.com>  
Date: Sat May 3 15:38:15 2025 -0300  
  
added line in main branch  
  
commit e31de731f1cec280004e762fe8a3e2f4396d4dd2 (feature-branch)  
Author: Michel San Martin <mikeemmanuel91@gmail.com>  
Date: Sat May 3 15:36:25 2025 -0300  
  
added line in feature-branch  
:[]
```