

TRABAJO PRÁCTICO Nº 2

Alumna: Santarelli Rocío.

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

- **¿Qué es GitHub?**

GitHub es una plataforma en línea que permite almacenar proyectos que usan el sistema de control de versiones Git. Es el servicio más popular del mundo para compartir y colaborar en proyectos de software, tanto de código abierto como privados. Almacena los repositorios en la nube, lo que facilita el trabajo colaborativo entre desarrolladores ubicados en diferentes partes del mundo. Además de servir como un "espacio remoto" donde subir tu código, GitHub incluye herramientas para organizar, discutir, revisar y aprobar cambios, lo que lo convierte en una solución integral para el desarrollo de software moderno.

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

Para esto primero debemos crearnos una cuenta en GitHub, recomendamos ir a GitHub (<https://github.com>) y crearse una cuenta.

Para subir tu proyecto a GitHub, deberás crear un repositorio donde alojarlo. Para crear un repositorio deberemos:

1º: En la esquina superior derecha de cualquier página, utiliza el menú desplegable + y selecciona Repositorio Nuevo

2º: Escribe un nombre corto y fácil de recordar para tu repositorio. Por ejemplo: "mi-proyecto".

3º: También puedes agregar una descripción de tu repositorio. Por ejemplo, "Mi primer repositorio en GitHub".

4º: Elige la visibilidad del repositorio. Puedes restringir quién tiene acceso a un repositorio eligiendo la visibilidad de un repositorio: público o privado.

5º: Podemos crear el repositorio con un ReadMe.

6º: Por último se da click en Crear repositorio.

Se puede copiar la URL del repositorio (por ejemplo:

<https://github.com/usuario/mi-proyecto.git>)

- **¿Cómo crear una rama en Git?**

git branch <nombre>: Crea una nueva rama.

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

git checkout <nombre>: Cambia a otra rama.

- **¿Cómo fusionar ramas en Git?**

git merge <rama>: Fusiona una rama con la actual.

- **¿Cómo crear un commit en Git?**

Cada vez que hacés git commit, los cambios se almacenan aquí.

`git commit -m "mensaje"`: Guarda los archivos preparados como una nueva versión del proyecto. Anterior a ello debes hacer un `git add`.

- **¿Cómo enviar un commit a GitHub?**

Primero debemos clonar el repositorio de GitHub a nuestra máquina local:

```
git clone https://github.com/usuario/repositorio.git
```

```
cd repositorio
```

Haz cambios en los archivos del repositorio y agregalos.

Una vez que hayas hecho cambios y los hayas confirmado (`git add + git commit`), podés subirlos con: `git push origin -nombre de la rama-` y luego enviar un pull request desde GitHub para que los responsables del proyecto original revisen y, si lo desean, integren tus cambios.

- **¿Qué es un repositorio remoto?**

Un repositorio remoto es una versión de tu repositorio que generalmente se encuentra alojado en plataformas como GitHub o GitLab. Permite sincronizar el trabajo con otros desarrolladores.

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

Usás el comando:

```
git remote add origin <URL-del-repositorio>
```

Donde origin es un nombre que se le da al remoto pero se le puede poner otro nombre si se quiere, y la URL es la dirección del repositorio remoto.

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

Con `git push` se suben los commits al repositorio remoto:

```
git push origin nombre_de_la_rama
```

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

Para tirar los cambios del repositorio remoto usa el comando `git pull` para descargar y fusionar los cambios del repositorio remoto con tu rama local:

```
git pull origin nombre_de_la_rama
```

Por ejemplo, si estás trabajando en la rama main o master:

```
git pull origin main/master.
```

- **¿Qué es un fork de repositorio?**

Un fork es una copia completa de un repositorio de GitHub que se crea dentro de tu cuenta.

Te permite:

- * Probar cambios sin afectar el proyecto original.
- * Proponer mejoras enviando los cambios mediante un pull request.
- * Trabajar con repositorios a los que no tenés permisos directos.

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

1. Ingresá a un repositorio público de GitHub (por ejemplo, <https://github.com/otro-usuario/proyecto>).

2. Hacé clic en el botón Fork (esquina superior derecha).

3. Elegí tu cuenta para crear una copia del proyecto en tu propio GitHub.

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

Para crear la Pull Request desde GitHub sigue los siguientes pasos:

1. Andá a tu fork en GitHub.

2. GitHub te va a sugerir hacer un "Compare & Pull Request". Hacé clic ahí.

3. Asegurate de que la base (base repository) sea el original y la comparación (compare) sea tu rama.

4. Agregá una descripción clara de lo que hiciste.

5. Hacé clic en "Create Pull Request".

• ¿Cómo aceptar una solicitud de extracción?

1. Entrá al repositorio en GitHub.

2. Andá a la pestaña "**Pull requests**" (arriba, cerca de "Code").

3. Vas a ver una lista de PRs abiertas. Hacé clic en la que querés revisar.

4. Revisá los cambios:

- Podés ver los archivos modificados en la pestaña "**Files changed**".
- También podés dejar comentarios si querés sugerir algo.

5. Si todo está bien, hacé clic en el botón verde que dice "**Merge pull request**".

6. Confirmá haciendo clic en "**Confirm merge**".

7. (Opcional) Podés borrar la rama del colaborador haciendo clic en "**Delete branch**" (esto no afecta su fork, solo limpia tu repo un poco).

• ¿Qué es una etiqueta en Git?

Las etiquetas (tags) en Git son como marcadores que se usan para etiquetar puntos específicos como importantes en la historia de tu proyecto, como versiones (v1.0, v2.1.3, etc). Es una forma de identificar un commit específico.

• ¿Cómo crear una etiqueta en Git?

Git utiliza dos tipos principales de etiquetas: ligeras y anotadas.

La etiqueta ligera (solo marca el commit): `git tag v1.0`

La etiqueta anotada (más completa, recomendada): `git tag -a v1.0 -m "Versión 1.0: primera versión estable"`

• ¿Cómo enviar una etiqueta a GitHub?

Para subir una etiqueta: `git push origin v1.0`

Para subir todas las etiquetas de golpe: `git push origin --tags`

• ¿Qué es un historial de Git?

El historial de Git es una secuencia de todos los cambios realizados en un repositorio de Git. Cada cambio en el repositorio se guarda como un commit, y cada commit contiene información sobre el estado del proyecto en un momento específico.

- **¿Cómo ver el historial de Git?**

El comando básico para ver el historial es `git log`.

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

Se puede buscar de diferentes maneras:

1. Buscar por mensaje de commit: **`git log --grep="palabra clave"`** (Te muestra todos los commits cuyo mensaje contiene esa palabra clave.
2. Buscar por autor: **`git log --author="Rocio"`** (por ej)
3. Buscar cambios en un archivo específico: **`git log archivo.txt`**
4. Buscar un commit que modificó cierta línea: **`git blame archivo.txt`**

Y después podés usar el hash que te da para ver el commit completo: **`git show <hash>`**

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

Una forma es borrando el historial manteniendo los archivos actuales:

```
rm -rf .git
```

```
git init
```

```
git add .
```

```
git commit -m "Nuevo inicio del historial"
```

Y otra forma es con el `git reset`, el cual quita del stage todos los archivos y carpetas del proyecto.

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

Un repositorio privado solo puede ser visto por vos y por las personas a las que vos invites.

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

1. Entrá a <https://github.com/new>

2. Completá:

- Nombre del repositorio
- Descripción (Opcional)

3. Elegí la opción "Private"

4. Hacé clic en "Create repository"

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

1. Entrá al repositorio.

2. Hacé clic en "Settings" (pestaña arriba del todo).

3. En el menú lateral, andá a "Collaborators" (puede estar dentro de "Access").

4. Hacé clic en "Add people"

5. Escribí su usuario de GitHub y elegí "Read", "Write", o "Admin" según lo que necesiten.

6. GitHub les va a mandar una invitación que tienen que aceptar.

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

Un repositorio público es visible para todo el mundo. Cualquiera puede ver tu código, y si querés, pueden hacer forks o pull requests.

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

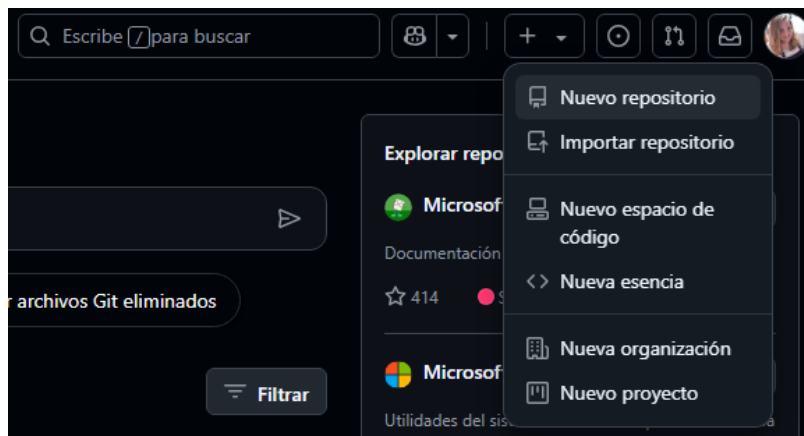
1. Entrá a <https://github.com/new>
2. Completá el nombre, descripción, etc.
3. Elegí la opción **"Public"**
4. Hacé clic en **"Create repository"**

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

Para compartir un repositorio solo hay que copiar la URL del repositorio y proporcionar el enlace directo.

2) Crear un repositorio:

Una vez en mi cuenta de GitHub selecciono el + que está arriba a la derecha y elijo la opción "Nuevo repositorio":



Luego completo con el nombre del repositorio que deseo y selecciono la opción "Público":

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importa un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (*).

Dueño *



santarellirocio ▾

Nombre del repositorio *

/ mi_repositorio

✓ mi_repositorio está disponible.

Los buenos nombres de repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué te parece? **carnaval de peluche ?**

Descripción (opcional)

Mi Repositorio Programación 1



Público

Cualquier persona en internet puede ver este repositorio. Tú decides quién puede contribuir.



Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Por último seleccione “Crear repositorio”:



Público

Cualquier persona en internet puede ver este repositorio. Tú decides quién puede contribuir.



Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:



Agregar un archivo README

Aquí puedes escribir una descripción detallada de tu proyecto. [Obtén más información sobre los archivos README.](#)

Agregar .gitignore

Plantilla .gitignore : Ninguna ▾

Seleccione los archivos que no desea rastrear de una lista de plantillas. [Obtenga más información sobre cómo ignorar archivos.](#)

Elija una licencia

Licencia : Ninguna ▾

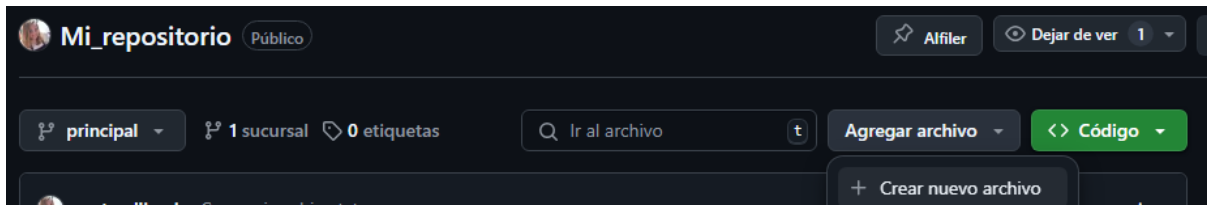
Una licencia indica a otros qué pueden y no pueden hacer con tu código. [Obtén más información sobre las licencias.](#)



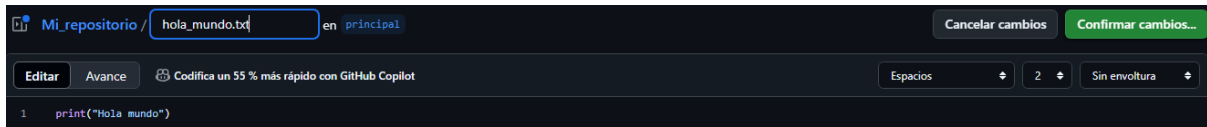
Estás creando un repositorio público en tu cuenta personal.

Crear repositorio

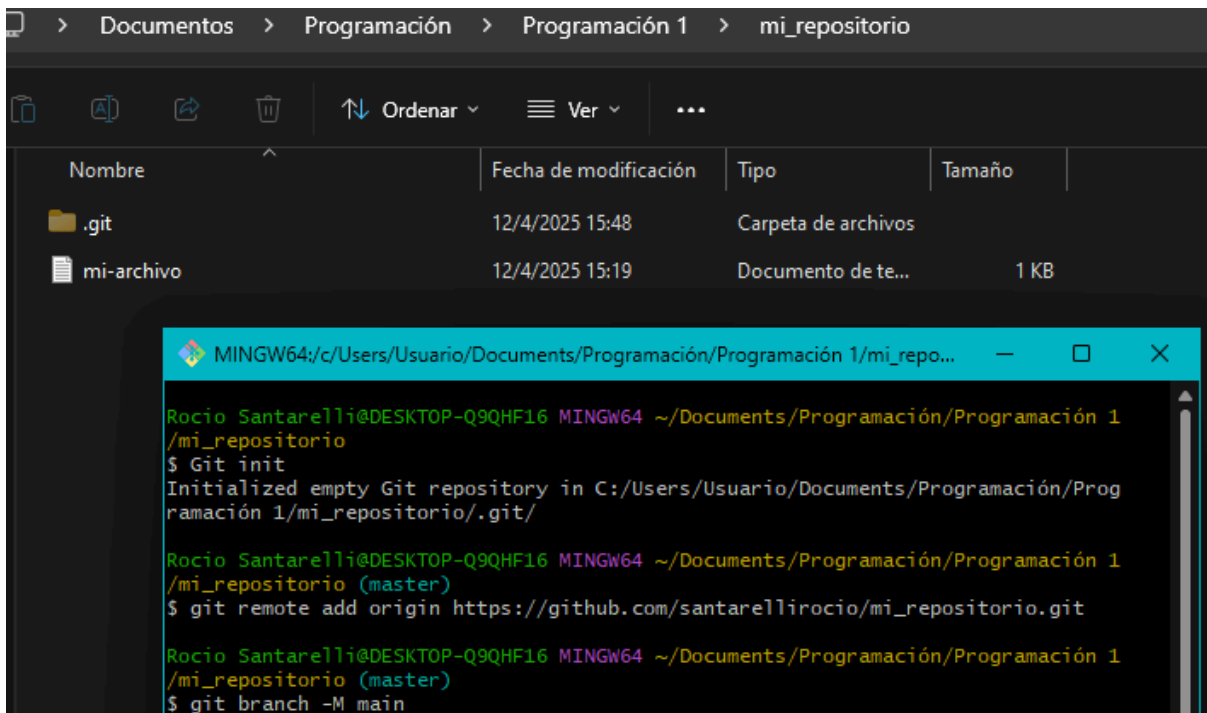
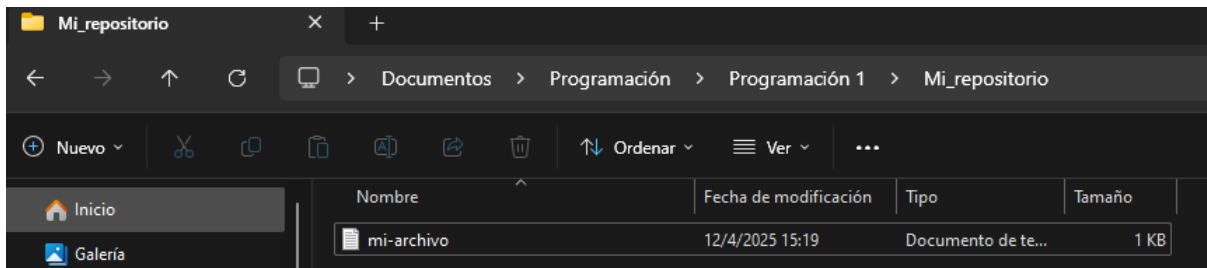
Dentro de mi repositorio selecciono “Agregar archivo” y pongo “Crear nuevo archivo”:



Elijo un nombre para mi archivo y doy a “Confirmar cambios”:



Agregando un Archivo:



Documentos > Programación > Programación 1 > mi_repositorio

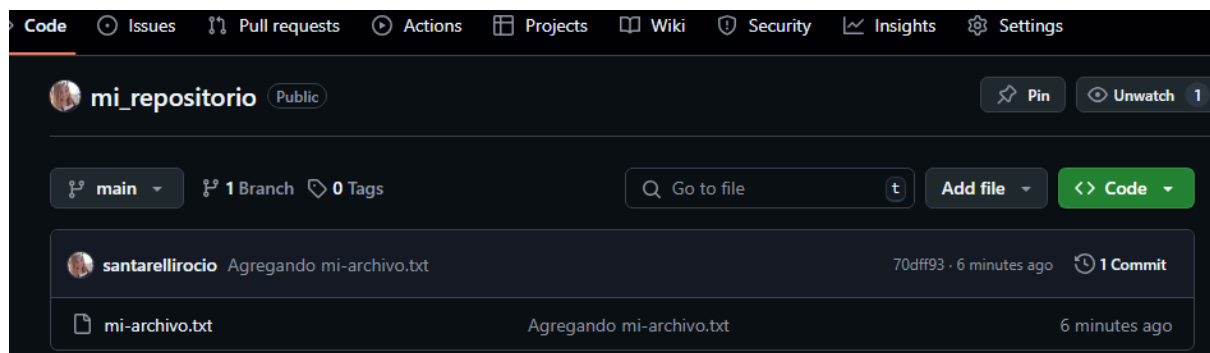
Nombre	Fecha de modificación	Tipo	Tamaño
.git	12/4/2025 16:04	Carpeta de archivos	
mi-archivo	12/4/2025 15:19	Documento de te...	1 KB

```
MINGW64:/c:/Users/Usuario/Documents/Programación/Programación 1/mi_repositorio
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (main)
$ git add .

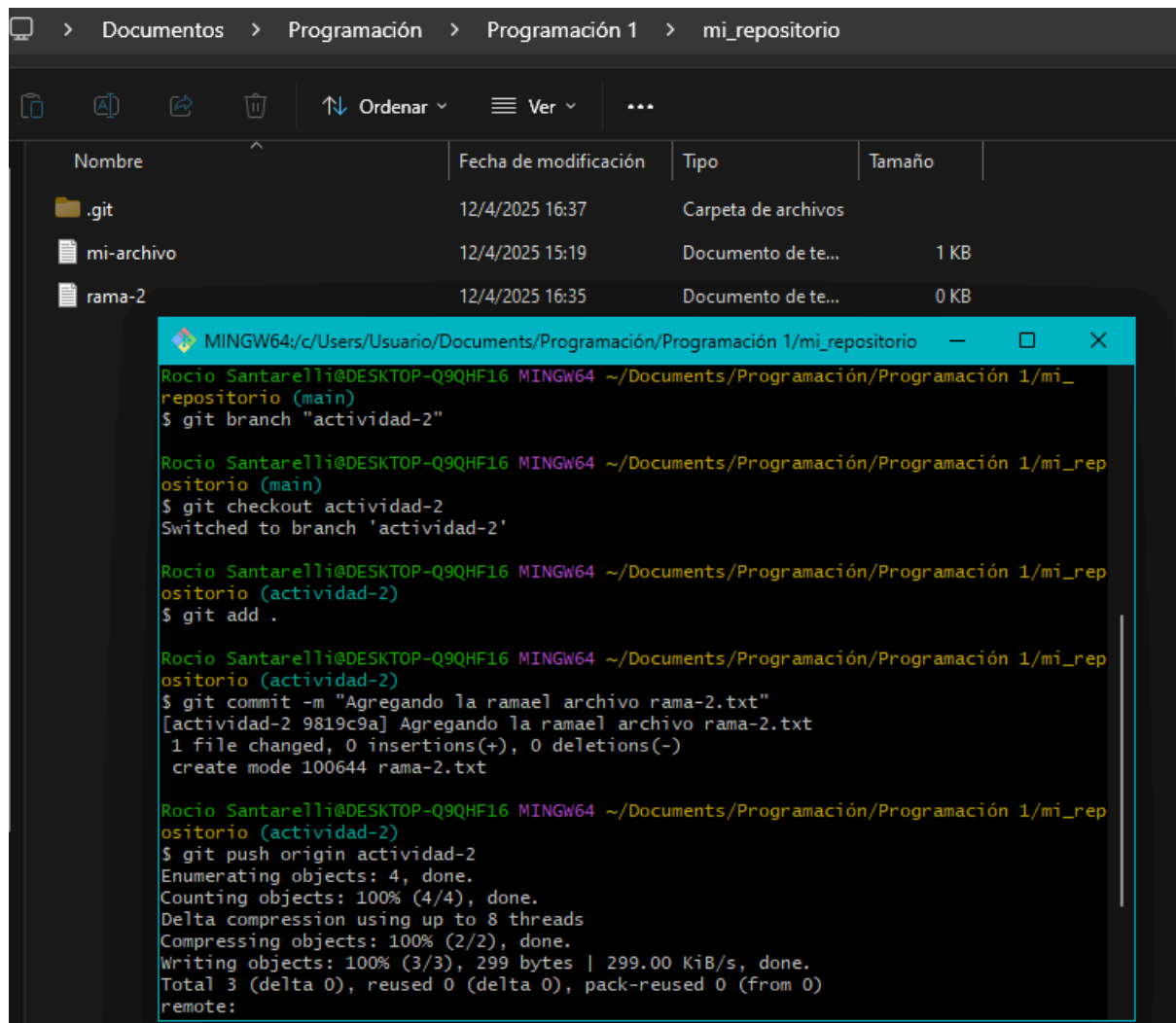
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (main)
$ git commit -m "Agregando mi-archivo.txt"
[main (root-commit) 70dff93] Agregando mi-archivo.txt
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo.txt

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 256 bytes | 128.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/santarellirocio/mi_repositorio.git
 * [new branch]      main -> main

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (main)
$
```



Creando Branchs:



```

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (actividad-2)
$ git push origin actividad-2
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), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'actividad-2' on GitHub by visiting:
remote:   https://github.com/santarellirocio/mi_repositorio/pull/new/actividad-2
remote:
To https://github.com/santarellirocio/mi_repositorio.git
* [new branch]   actividad-2 -> actividad-2

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/mi_repositorio (actividad-2)
$

```

3. Paso 1: Crear un repositorio en GitHub

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyecto en otro lugar? [Importa un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (*).

Dueño *



santarellirocio

Nombre del repositorio *

conflict-exercise

✓ El ejercicio de conflicto está disponible.

Los buenos nombres de repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué te parece? **cuscus automático** ?

Descripción (opcional)

A partir de un conflicto generado, resolverlo



Público

Cualquier persona en internet puede ver este repositorio. Tú decides quién puede contribuir.



Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:



Agregar un archivo README

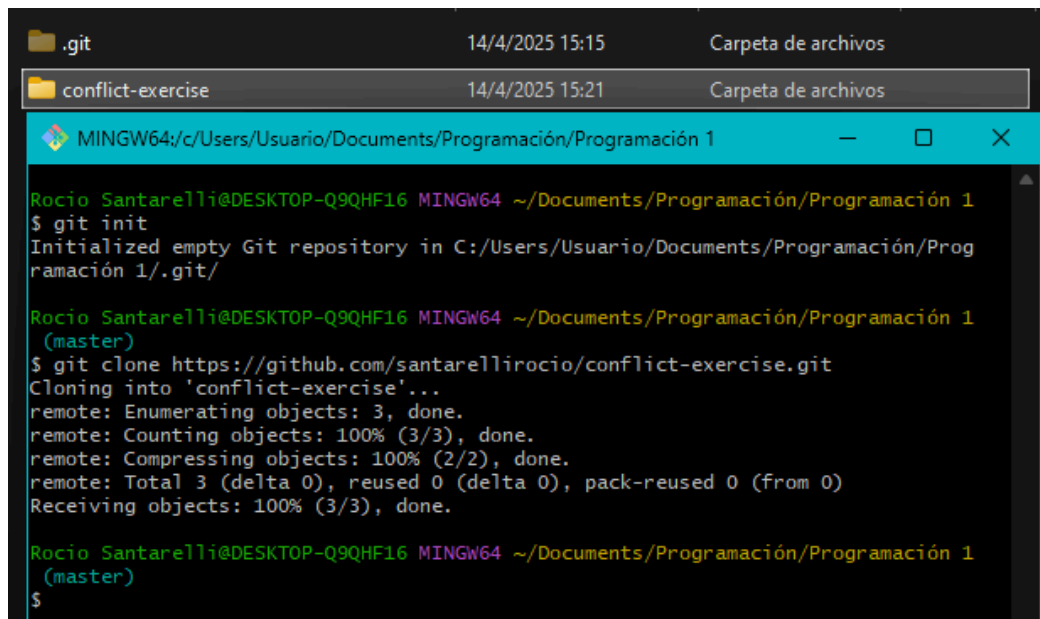
Aquí puedes escribir una descripción detallada de tu proyecto. [Obtén más información sobre los archivos README.](#)

Agregar .gitignore

Plantilla .gitignore : Ninguna

The screenshot shows the GitHub repository page for 'conflict-exercise' by 'santarellirocio'. The repository is public and has 1 commit. The main branch is 'main'. The repository contains a README.md file. The repository description is 'A partir de un conflicto generado, resolverlo'. The repository has 0 stars, 1 watching, and 0 forks. The repository is pinned to the user's profile.

Paso 2: Clonar el repositorio a tu máquina local

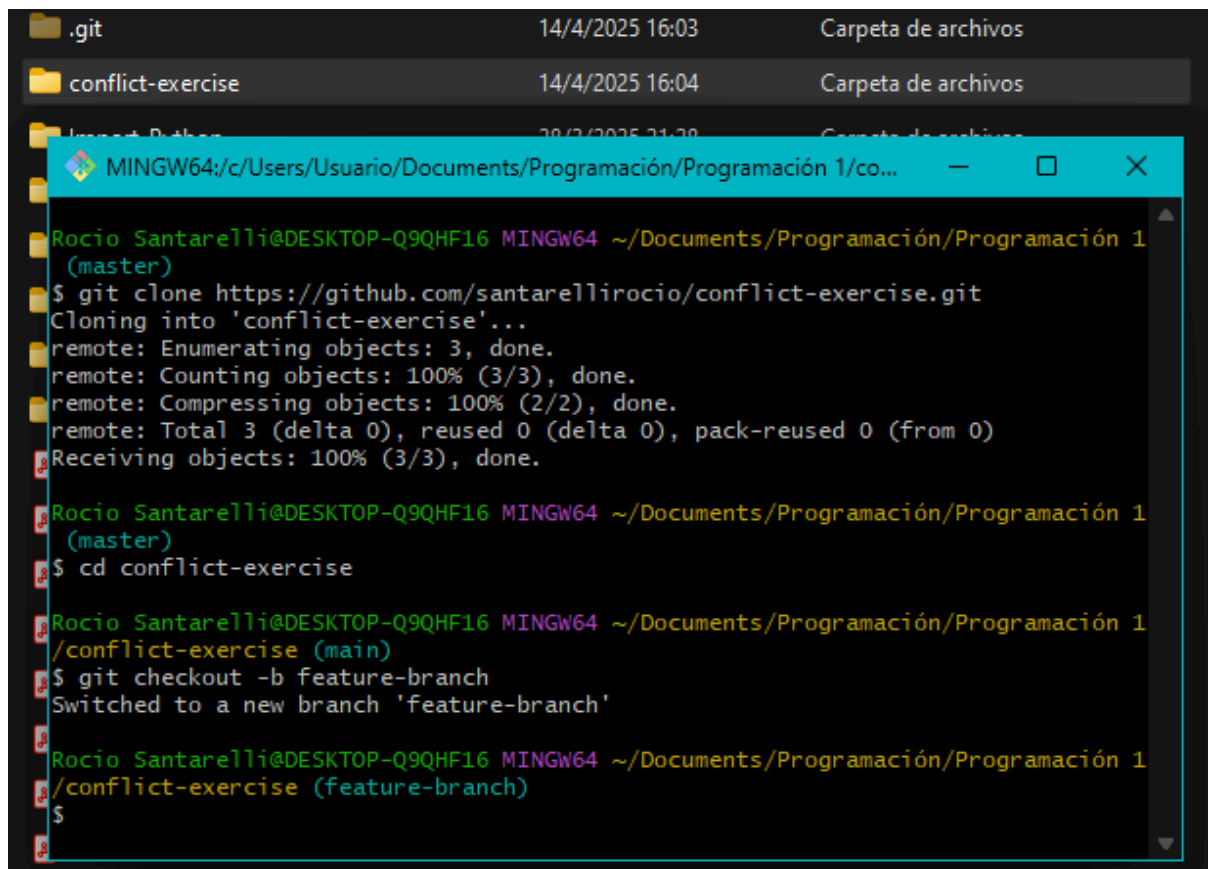


```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
$ git init
Initialized empty Git repository in C:/Users/Usuario/Documents/Programación/Programación 1/.git/

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
(master)
$ git clone https://github.com/santarellirocio/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
(master)
$
```

Paso 3: Crear una nueva rama y editar un archivo



```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
(master)
$ git clone https://github.com/santarellirocio/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
(master)
$ cd conflict-exercise

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (feature-branch)
$
```

```
① README.md X
C: > Users > Usuario > Documents > Programación > Programación 1 > conflict-exercise > ① README.md
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la feature branch.
```

```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (feature-branch)
$ git add README.md

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 5f89058] Added a line in feature-branch
1 file changed, 1 insertion(+)

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (feature-branch)
$ |
```

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

```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (main)
$ |
```

```
① README.md X
C: > Users > Usuario > Documents > Programación > Programación 1 > conflict-exercise > ① README.md > # conflict-exercise
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la main branch.
```

Antes de hacer lo que está aquí abajo (el git add README.md) la imagen de arriba aparecía con una M por la modificación realizada

```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1
/conflict-exercise (main)
$ git add README.md

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main e3bb4ff] Added a line in main branch
1 file changed, 1 insertion(+)

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$
```

Paso 5: Hacer un merge y generar un conflicto

```
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$ git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main|MERGING)
$ |
```

Paso 6: Resolver el conflicto

```
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 <<<<<< HEAD (Current Change)
4 Este es un cambio en la main branch.
5 =====
6 Este es un cambio en la feature branch.
7 >>>>>> feature-branch (Incoming Change)
```

Seleccioné ambos cambios poniendo las tildes y luego complete Merge para resolver el conflicto, mantuve ambos cambios pero ya no están en la misma línea:

```
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la feature branch.

1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la main branch.

Result C:\Users\Usuario\Documents\Programación\Programación 1\conflict-exercise\README.md 0 Conflicts Remaining

1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la feature branch.
4 Este es un cambio en la main branch.
```

```
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la feature branch.
4 Este es un cambio en la main branch.
```

```
① README.md X
C: > Users > Usuario > Documents > Programación > Programación 1 > conflict-exercise > ① README.md > # conflict-exercise
1 # conflict-exercise
2 A partir de un conflicto generado, resolverlo
3 Este es un cambio en la feature branch.
4 Este es un cambio en la main branch.

MINGW64/c/Users/Usuario/Documents/Programación/Programación 1/conf...
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main|MERGING)
$ git add README.md

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main caec124] Resolved merge conflict

Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$
```

Paso 7: Subir los cambios a GitHub


```
MINGW64/c/Users/Usuario/Documents/Programación/Programación 1/conf...
Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$ 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), 763 bytes | 381.00 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/santarellirocio/conflict-exercise.git
6b3a327..caec124 main -> main


Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote: https://github.com/santarellirocio/conflict-exercise/pull/new/featur
e-branch
remote:
To https://github.com/santarellirocio/conflict-exercise.git
* [new branch] feature-branch -> feature-branch


Rocio Santarelli@DESKTOP-Q9QHF16 MINGW64 ~/Documents/Programación/Programación 1/
conflict-exercise (main)
$
```

Paso 8: Verificar en GitHub

conflict-exercise / README.md


 **santarellirocio** Resolved merge conflict



caec124 · 5 minutes ago  History


Preview Code Blame 4 lines (4 loc) · 142 Bytes  Code 55% faster with GitHub Copilot

Raw Copy Download Edit

```
1  # conflict-exercise
2  A partir de un conflicto generado, resolverlo
3  Este es un cambio en la feature branch.
4  Este es un cambio en la main branch.
```


 **conflict-exercise** Public


 Pin  Unwatch 1


 main 2 Branches 0 Tags


Go to file

Add file

 Code


 **santarellirocio** Resolved merge conflict


caec124 · 8 minutes ago  1 Commit

 README.md

Resolved merge conflict

8 minutes ago

 README



conflict-exercise