

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

- ¿Qué es GitHub?

Es un sitio web o una comunidad que almacena, gestiona y muestran la actividad de repositorios, a grandes rasgos, sólo son copias de nuestros repositorios en otra computadora.

Que podemos compartir de manera pública o privada

Propiedades importantes: funciona como un backup y más importante, sociabiliza la programación. Y cuya herramienta subyacente es git.

- ¿Cómo crear un repositorio en GitHub?

Crear una cuenta en github, luego en el perfil hacer click en el botón "+" en el extremo superior derecho y seleccionar "New repository".

Luego en la página que se carga elegir un nombre para el repositorio, seleccionar si es público o privado, y click a botón "Create repository"

- ¿Cómo crear una rama en Git?

```
git branch nuevarama
```

- ¿Cómo cambiar a una rama en Git?

cambiar a una rama existente

```
git checkout nuevarama
```

crear y cambiar a una rama

```
git checkout -b nuevarama
```

- ¿Cómo fusionar ramas en Git?

parado en nuevarama funcionar con viejarama

```
git merge viejarama
```

- ¿Cómo crear un commit en Git?

Guardar los cambios y commitarlos

```
git add .
```

```
git commit -m "Cmabios"
```

- ¿Cómo enviar un commit a GitHub?

una vez que el cambio ya está commiteado se puede subir a nuevarama donde estabamos trabajando

```
git push origin nuevarama
```

- ¿Qué es un repositorio remoto?

Es una copia de nuestros repositorios en otra computadora que podemos mantener sincronizada con nuestro repositorio local

- ¿Cómo agregar un repositorio remoto a Git?

```
git remote add origin https://github.com/usuario/repositorio.git
```

- ¿Cómo empujar cambios a un repositorio remoto?

agregarlos, commitarlos y pusharlos

```
git add .  
git commit -m "Cambios"  
git push origin nuevarama
```

- ¿Cómo tirar de cambios de un repositorio remoto?

Usando el comando pull

```
git pull origin reporemoto
```

- ¿Qué es un fork de repositorio?

Es una copia de un repositorio desde el commit en el que estoy parado en nuestra cuenta de github

- ¿Cómo crear un fork de un repositorio?

Desde github parado en un repositorio hacer click en el botón fork, y luego seleccionar nombre y descripción

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

Primero hay que hacer una push en el repositorio de los cambios en el branch que estuve trabajando

```
git push origin nuevarama
```

Luego desde el portal de github parado en el repositorio al cual quiero enviar la solicitud, luego click en “pull requests”, y “New pull request” y agregar título, descripción y a quien está asignado

- ¿Cómo aceptar una solicitud de extracción?

Desde Github, en la pestaña “pull request” del repositorio, seleccionar el pull request que se desea aceptar, luego revisar los cambios y dar click al botón de “merge and pull request” y luego a “confirm merge”

- ¿Qué es una etiqueta en Git?

es una referencia a un punto específico de la historia en el repositorio, no se pueden hacer cambios en las etiquetas.

- ¿Cómo crear una etiqueta en Git?

simplemente parado en el repositorio donde se quiere crear el tag ejecutar

```
git tag nuevotag
```

- ¿Cómo enviar una etiqueta a GitHub?

Al igual que con un branch usando un push

```
git push origin nuevotag
```

- ¿Qué es un historial de Git?

El listado de commits desde que se creó el proyecto hasta el momento en que se consulta el historial

- ¿Cómo ver el historial de Git?

Se puede ver ejecutando

```
git log
```

- ¿Cómo buscar en el historial de Git?

Se puede buscar usando el parámetro grep junto con el comando git log

```
git log --grep="la busqueda"
```

buscar por autor

```
git log --author="autor"
```

O por fecha

```
git log --since="2025-03-31" --until="2025-04-06"
```

- ¿Cómo borrar el historial de Git?

Se puede usar un git reset --mixed para borrar el historial, no se borra el primer commit y tampoco los cambios sobre los archivos

```
git reset --mixed HEAD^
```

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio que tiene el acceso restringido a los usuarios que tienen autorización en el repositorio, por lo general el usuario que los creó, y los usuarios invitados a ver el repo.

- ¿Cómo crear un repositorio privado en GitHub?

Al momento de la creación tildar la opción de repositorio privado, o convertir en privado desde

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

En github parado en el repositorio ir a la pestaña "settings", luego a "Access" y "Collaborators". Luego click en el botón "add people" y escribir en nombre del usuario o el email, seleccionarlo y clic en el botón "Add <nombreDelUsuario>"

- ¿Qué es un repositorio público en GitHub?

Al ser público significa que el repo es accesible por cualquier usuario y que puede clonarlo o hacer un fork. También puede colaborar sin que tenga que ser invitado por el dueño del repositorio (mediante pull requests).

- ¿Cómo crear un repositorio público en GitHub?

En github hacer click en el botón “+” en el extremo superior derecho y seleccionar "New repository".

Luego en la página que se carga elegir un nombre para el repositorio, seleccionar que sea público, y click a botón "Create repository"

- ¿Cómo compartir un repositorio público en GitHub?

Compartiendo la URL, por ejemplo se puede copiar directamente de la url del navegador o en github haciendo clic en el botón “code”, y en HTTPS copiar la url y compartirla.

2) Realizar la siguiente actividad:

Crear un repositorio.

<https://github.com/raulduarte85utn/actividad2-2>


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 *

 raulduarte85utn


 /

actividad2-2


 actividad2-2 is available.

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

Description (optional)

☒  **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](#).

Agregando un Archivo

```

[] > [] ~/Doc/UTN/Programacion I/actividad2-2 > on [] [] main touch mi-archivo.txt
[] > [] ~/Doc/U/Programacion I/actividad2-2 > on [] [] main ?1 git add .
[] > [] ~/Doc/U/Programacion I/actividad2-2 > on [] [] main +1 git commit -m "agregando mi-archivo.txt"
[main ad57089] agregando mi-archivo.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo.txt
[] > [] ~/Doc/U/Programacion I/actividad2-2 > on [] [] main +1 git push origin main
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), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:raulduarte85utn/actividad2-2.git
c5c2fc1..ad57089 main -> main
[] > [] ~/Doc/UTN/Programacion I/actividad2-2 > on [] [] main

```

Creando Branchs

```

[] > [] ~/Doc/UTN/Programacion I/actividad2-2 > on [] [] main git branch newbranch
[] > [] ~/Doc/UTN/Programacion I/actividad2-2 > on [] [] main git checkout newbranch
Switched to branch 'newbranch'
[] > [] ~/Doc/U/Pro/actividad2-2 > on [] [] newbranch touch mi-archivo2.txt
[] > [] ~/Doc/U/Pro/actividad2-2 > on [] [] newbranch ?1 git add .
[] > [] ~/Doc/U/Pro/actividad2-2 > on [] [] newbranch +1 git commit -m "agregando mi-archivo2.txt"
[newbranch b449f2a] agregando mi-archivo2.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 mi-archivo2.txt
[] > [] ~/Doc/U/Pro/actividad2-2 > on [] [] newbranch git push origin newbranch
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 287 bytes | 287.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'newbranch' on GitHub by visiting:
remote: https://github.com/raulduarte85utn/actividad2-2/pull/new/newbranch
remote:
To github.com:raulduarte85utn/actividad2-2.git
* [new branch] newbranch -> newbranch
[] > [] ~/Doc/U/Pro/actividad2-2 > on [] [] newbranch

```

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 *

 raulduarte85utn

Repository name *

conflict-exercise

 conflict-exercise is available.

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

Description (optional)

☒ 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](#).

Paso 2: Clonar el repositorio a tu máquina local

```
Apple > ~/Documents/UTN/Programacion I git clone git@github.com:raulduarte85utn/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
Apple > ~/Documents/UTN/Programacion I cd conflict-exercise
Apple > ~/Documents/UTN/Programacion I/conflict-exercise > on main
```

Paso 3: Crear una nueva rama y editar un archivo

```
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main git checkout -b feature-branch
Switched to a new branch 'feature-branch'
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 feature-branch vim README.md
Apple> ~/Doc/U/Pro/conflict-exercise > on 🍏 📁 feature-branch !1 cat README.md
# conflict-exercise
Este es un cambio en la feature branch.
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 feature-branch !1 git add README.md
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 feature-branch +1 git commit -m "Added a line in feature-branch"
[feature-branch 92400cb] Added a line in feature-branch
1 file changed, 2 insertions(+), 1 deletion(-)
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 feature-branch
```

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

```
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 feature-branch git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main vim README.md
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 main !1 cat README.md
# conflict-exercise
Este es un cambio en la main branch.
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main !1 git add README.md
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1 git commit -m "Added a line in main branch"
[main b0b3e83] Added a line in main branch
1 file changed, 2 insertions(+), 1 deletion(-)
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1
```

Paso 5: Hacer un merge y generar un conflicto

```
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1 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.
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1 merge ~1
```

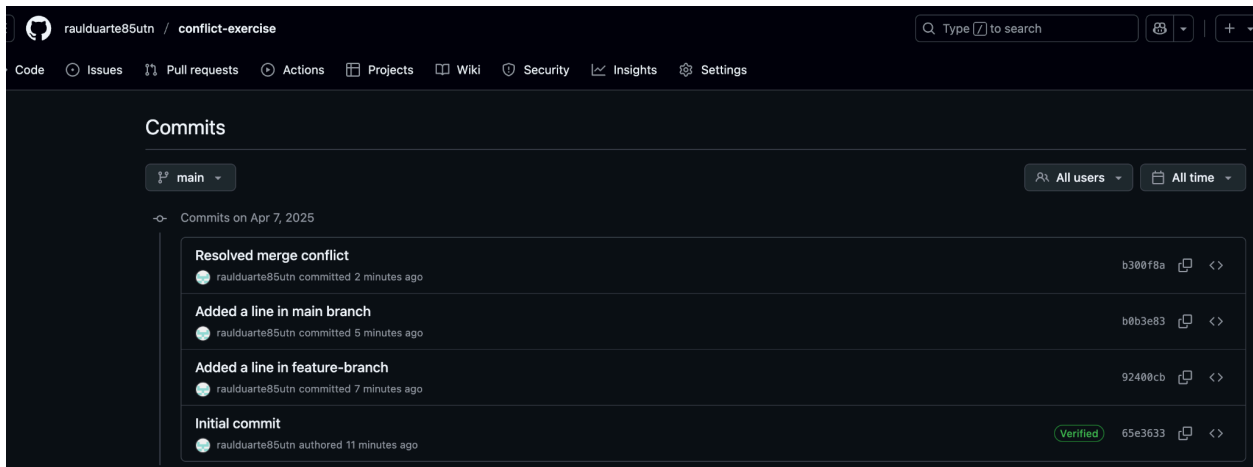
Paso 6: Resolver el conflicto

```
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1 merge ~1 vim README.md
Apple> ~/Doc/U/Pro/conflict-exercise > on 🍏 📁 main +1 merge ~1 git add README.md
Apple> ~/Doc/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +1 merge +1 git commit -m "Resolved merge conflict"
[main b300f8a] Resolved merge conflict
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +3 cat README.md
# conflict-exercise
Este es un cambio en la main branch.
Este es un cambio en la feature branch.
Apple> ~/Documents/UTN/Programacion I/conflict-exercise > on 🍏 📁 main +3
```

Paso 7: Subir los cambios a GitHub

```
Apple > ~/Documents/UTN/Programacion I/conflict-exercise > on 🐙 main +3 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), 796 bytes | 796.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:raulduarte85utn/conflict-exercise.git
 65e3633..b300f8a  main -> main
Apple > ~/Documents/UTN/Programacion I/conflict-exercise > on 🐙 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/raulduarte85utn/conflict-exercise/pull/new/feature-branch
remote:
To github.com:raulduarte85utn/conflict-exercise.git
* [new branch]   feature-branch -> feature-branch
Apple > ~/Documents/UTN/Programacion I/conflict-exercise > on 🐙 main
```

Paso 8: Verificar en GitHub



The screenshot shows the GitHub repository page for 'raulduarte85utn / conflict-exercise'. The 'Commits' tab is selected, showing a list of commits on the 'main' branch. The commits are ordered from most recent to oldest. The most recent commit is 'Resolved merge conflict' by raulduarte85utn, committed 2 minutes ago, with hash b300f8a. Below it is 'Added a line in main branch' by raulduarte85utn, committed 5 minutes ago, with hash b0b3e83. Then 'Added a line in feature-branch' by raulduarte85utn, committed 7 minutes ago, with hash 92400cb. The oldest commit is 'Initial commit' by raulduarte85utn, authored 11 minutes ago, with hash 65e3633, which is marked as 'Verified'.

Commit Message	Author	Committed	Hash	Verified
Resolved merge conflict	raulduarte85utn	2 minutes ago	b300f8a	
Added a line in main branch	raulduarte85utn	5 minutes ago	b0b3e83	
Added a line in feature-branch	raulduarte85utn	7 minutes ago	92400cb	
Initial commit	raulduarte85utn	11 minutes ago	65e3633	Verified