

Práctico 2: Git y GitHub

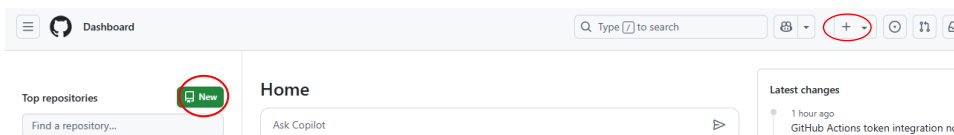
Objetivo:

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.

Resultados de aprendizaje:

1. Comprender los conceptos básicos de Git y GitHub: Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.
2. Manejar comandos esenciales de Git: Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.
3. Aplicar técnicas de colaboración en GitHub: Configurar y utilizar repositorios remotos, realizar forks, y gestionar pull requests para facilitar el trabajo colaborativo.
4. Resolver conflictos en un entorno de control de versiones: Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.

Actividades

- 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):
 - ¿Qué es GitHub?
GitHub es una plataforma en línea que permite almacenar proyectos que usan Git. Sirve 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. Además, incluye herramientas para organizar, discutir, revisar y aprobar cambios.
 - ¿Cómo crear un repositorio en GitHub?
 - 1) Iniciar sesión en GitHub.
 - 2) Hacer clic en el botón verde “New” o “+” → “New repository”.
 - 3) Poner un nombre, descripción (opcional) y elegir si será público o privado.
 - 4) Hacer clic en “Create repository”.
- 
- ¿Cómo crear una rama en Git?
Para crear una rama en Git se utiliza el comando `git branch nombreRama`, donde

nombreRama es el nombre que se le quiere dar a la rama.

- ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en específico se utiliza el comando `git checkout nombreRama`, donde nombre rama es el nombre de la rama a la que quiero cambiar.

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas se utiliza el comando `git merge`.

Para hacer un merge correctamente, hay que estar posicionado en la rama donde se quiere integrar los cambios.

- ¿Cómo crear un commit en Git?

Primero debemos agregar los cambios con el comando `git add`.

Luego usando el comando `git commit -m "Descripcion del cambio realizado"`

- ¿Cómo enviar un commit a GitHub?

Para enviar un commit a Github debemos usar el comando `git push -u origin nombreDeLaRama`

- ¿Qué es un repositorio remoto?

Es una versión del proyecto alojada en GitHub. Se sincroniza con el repositorio local.

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git se utiliza el comando `git remote add origin https://github.com/"usuario"/"mi-proyecto".git`

Esto le dice a Git que "origin" será el repositorio remoto.

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios a un repositorio remoto se utiliza el comando `git push -u origin nombreDeLaRama`

- ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar o traer cambios de un repositorio remoto se utiliza el comando `git pull origin nombreDeLaRama`

- ¿Qué es un fork de repositorio?

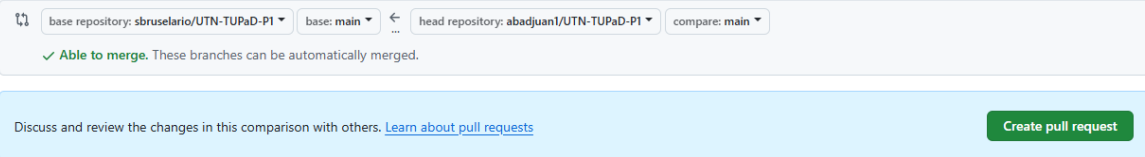
Un fork es una copia completa de un repositorio de GitHub que se crea dentro de tu cuenta.

- ¿Cómo crear un fork de un repositorio?

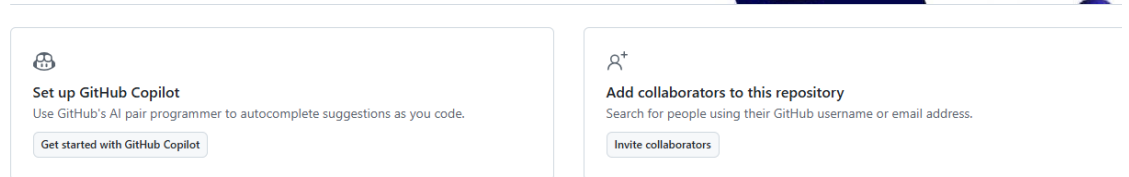
- 1) Ingresar a un repositorio público de GitHub
- 2) Hacer clic en el botón Fork (esquina superior derecha).
- 3) Elegir tu cuenta para crear una copia del proyecto en tu propio GitHub

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
Una solicitud de extracción se envía desde Github. En tu fork clonado aparece la opción **Create pull request**
Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



- ¿Cómo aceptar una solicitud de extracción?
El autor del repositorio debe dirigirse a sus pull request para ver los cambios que hemos sugerido y decidir si los acepta.
- ¿Qué es un etiqueta en Git?
Es una marca que señala un punto específico en la historia del repositorio, comúnmente usada para versiones.
- ¿Cómo crear una etiqueta en Git?
Para crear una etiqueta se utiliza el comando `git tag nombre`
- ¿Cómo enviar una etiqueta a GitHub?
Para enviar una etiqueta a Github se utiliza el comando `git push --tags`
- ¿Qué es un historial de Git?
El historial de Git es el registro completo de todos los commits realizados en un proyecto.
- ¿Cómo ver el historial de Git?
Para ver el historial de Git se utiliza el comando `git log`
- ¿Cómo buscar en el historial de Git?
Para buscar en el historia se utiliza el comando `git log --grep="palabra_clave"`
 - ¿Cómo borrar el historial de Git?
Con el comando `rm -rf .git`
Esto elimina todo el historial.
- ¿Qué es un repositorio privado en GitHub?
Es un repositorio el cual solo el autor y las personas que autorice pueden ver.
- ¿Cómo crear un repositorio privado en GitHub?
 1. Crear un nuevo repositorio
 2. Elegir la opción "Private"
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
Una vez creado el repositorio dirigirse a la opción **Invite Collaborators**



- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio cuyo contenido es accesible a cualquier persona en Internet.

- ¿Cómo crear un repositorio público en GitHub?
 - . Crear un nuevo repositorio
 - . Elegir la opción "Public"
- ¿Cómo compartir un repositorio público en GitHub?

Compartiendo la dirección URL del repositorio.


2) Realizar la siguiente actividad:

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

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 *
 abadjuan1	/ Repo1

✓ Repo1 is available.

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

Description (optional)

Este es mi primer repositorio

- ☒  **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.](#)

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

- Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
- Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion
$ git clone https://github.com/abadjuan1/Repo1.git
Cloning into 'Repo1'...
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.

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion
$ cd Repo1

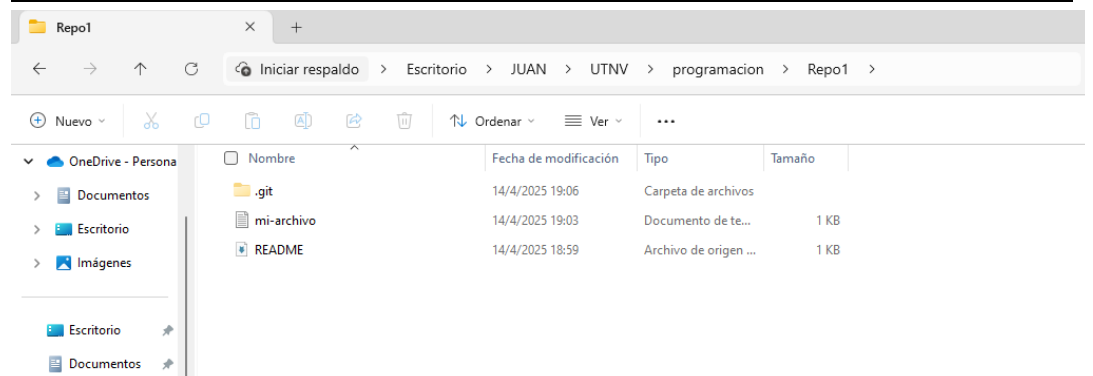
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ echo "Este es mi texto" > mi-archivo.txt

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git add .
warning: in the working copy of 'mi-archivo.txt', LF will be replaced by CRLF the ne

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git commit -m "Agregando mi archivo de texto"
[main de02586] Agregando mi archivo de texto
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo.txt

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ |

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git push -u 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), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/abadjuan1/Repo1.git
484132d..de02586 main -> main
branch 'main' set up to track 'origin/main'.
```



Repo1 Public

Pin

Unwatch 1

main

1 Branch 0 Tags

Go to file

t

Add file

<> Code

abadjuan1 Agregando mi archivo de texto

de02586 · 4 minutes ago 2 Commits

README.md	Initial commit	14 minutes ago
mi-archivo.txt	Agregando mi archivo de texto	4 minutes ago

README

Repo1

Este es mi primer repositorio

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

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git branch
* main

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git branch nueva-rama

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (main)
$ git checkout nueva-rama
Switched to branch 'nueva-rama'

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (nueva-rama)
$ git branch
main
* nueva-rama
```

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (nueva-rama)
$ echo "Esta es la nueva rama" > mi-archivo.txt

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (nueva-rama)
$ git add .
warning: in the working copy of 'mi-archivo.txt', LF will be replaced by CRLF the next time Git touches it

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (nueva-rama)
$ git commit -m "Agregando desde nueva rama"
[nueva-rama 9f968e4] Agregando desde nueva rama
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/Repo1 (nueva-rama)
$ git push -u origin nueva-rama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nueva-rama' on GitHub by visiting:
remote:   https://github.com/abadjuan1/Repo1/pull/new/nueva-rama
remote:
To https://github.com/abadjuan1/Repo1.git
 * [new branch]   nueva-rama -> nueva-rama
branch 'nueva-rama' set up to track 'origin/nueva-rama'.
```

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".

Paso 2: Clonar el repositorio a tu máquina local

- Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuario/conflict-exercise.git>).

- Abre la terminal o línea de comandos en tu máquina.
- Clona el repositorio usando el comando:

```
git clone https://github.com/tuusuario/conflict-exercise.git
```

- Entra en el directorio del repositorio:

```
cd conflict-exercise
```

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, por ejemplo:

Este es un cambio en la feature branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

```
git commit -m "Added a line in feature-branch"
```

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

- Cambia de vuelta a la rama principal (main):

```
git checkout main
```

- Edita el archivo README.md de nuevo, añadiendo una línea diferente:

Este es un cambio en la main branch.

- Guarda los cambios y haz un commit:

```
git add README.md
```

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

Paso 5: Hacer un merge y generar un conflicto

- Intenta hacer un merge de la feature-branch en la rama main:

```
git merge feature-branch
```

- Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

- Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

```
<<<<<<< HEAD
```

Este es un cambio en la main branch.

```
=====
```

Este es un cambio en la feature branch.

```
>>>>>>> feature-branch
```

- Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.
- Edita el archivo para resolver el conflicto y guarda los cambios (Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).
- Añade el archivo resuelto y completa el merge:

```
git add README.md
```

```
git commit -m "Resolved merge conflict"
```

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
```

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

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 *  abadjuan1 / Repository name *

✓ conflict-exercise is available.

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

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.](#)

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion
$ git clone https://github.com/abadjuan1/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.

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion
$ cd conflict-exercise

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main)
$ git branch feature-branch

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'

① README.md ✕

C: > Users > abadc > Desktop > JUAN > UTNV > programacion > conflict-exercise > ① README.md > [ms] # conflict-exercise
1  # conflict-exercise
2
3  Este es un cambio en la feature-branch
4
```

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (feature-branch)
$ git add README.md

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 340ba17] Added a line in feature-branch
1 file changed, 3 insertions(+), 1 deletion(-)
```

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

① README.md X

```
C: > Users > abadc > Desktop > JUAN > UTV > programacion > conflict-exercise > ① README.md > # conflict-exercise
1 # conflict-exercise
2 Esto es un cambio en la main branch |
```

```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main)
$ git add README.md

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main 70772c6] Added a line in main branch
1 file changed, 3 insertions(+), 1 deletion(-)

abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/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.
```

① README.md X

```
C: > Users > abadc > Desktop > JUAN > UTV > programacion > conflict-exercise > ① README.md > # conflict-exercise
1 # conflict-exercise
2
3 <<<<<< HEAD (Current Change)
4 Esto es un cambio en la main branch
5 =====
6 Este es un cambio en la feature-branch
7 >>>>>> feature-branch (Incoming Change)
8
```

① README.md X


```
C: > Users > abadc > Desktop > JUAN > UTV > programacion > conflict-exercise > ① README.md > # conflict-exercise
1 # conflict-exercise
2 |
3 Esto es un cambio en la main branch
4 Este es un cambio en la feature-branch
5
```




```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main|MERGING)
$ git add README.md


abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
```


```
abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/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), 835 bytes | 417.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/abadjuan1/conflict-exercise.git
   0b0e58d..9f31c26  main -> main



abadc@DESKTOP-4T39JRB MINGW64 ~/Desktop/JUAN/UTNV/programacion/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/abadjuan1/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/abadjuan1/conflict-exercise.git
 * [new branch]      feature-branch -> feature-branch
```

 **conflict-exercise** Public Pin Unwatch 1

 main  2 Branches  0 Tags t Add file <> Code

 **abadjuan1** Resolved merge conflict 9f31c26 · 2 minutes ago 4 Commits

 README.md Resolved merge conflict 2 minutes ago

 **README** 

conflict-exercise

Esto es un cambio en la main branch Este es un cambio en la feature-branch

