

## Programación 1

### Práctico 2: Git y GitHub

Nombre: Mauro Gaspar

Comisión: M2025-14

#### Actividades

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

- ¿Qué es GitHub?  
*Es una plataforma de alojamiento de código y colaboración para desarrolladores.*
- ¿Cómo crear un repositorio en GitHub?  
*Una vez iniciado en tu cuenta, hacer click en el botón "+", nuevo repositorio, seleccionar si será público o privado, crear repositorio.*
- ¿Cómo crear una rama en Git?  
*git branch <nombre-rama>*
- ¿Cómo cambiar a una rama en Git?  
*git checkout <rama>*
- ¿Cómo fusionar ramas en Git?  
*git merge <rama>*
- ¿Cómo crear un commit en Git?  
*git commit -m "mensaje"*
- ¿Cómo enviar un commit a GitHub?  
*git commit -m "cambios"*  
*git remote add origin <url>*  
*git push origin main*
- ¿Qué es un repositorio remoto?  
*Es una copia de un repositorio de Git almacenada en un servidor remoto, como GitHub.*
- ¿Cómo agregar un repositorio remoto a Git?  
*git remote add origin <url>*
- ¿Cómo empujar cambios a un repositorio remoto?  
*git push -u origin master*
- ¿Cómo tirar de cambios de un repositorio remoto?  
*git pull origin master*
- ¿Qué es un fork de repositorio?  
*Es una copia independiente de un repositorio existente en GitHub.*

- ¿Cómo crear un fork de un repositorio?  
*Click en el botón “Fork” en la esquina superior derecha de la página del repositorio.*
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?  
*Click en el botón “Nueva solicitud de extracción”*
- ¿Cómo aceptar una solicitud de extracción?  
*Si estás de acuerdo con los cambios, click en el botón “Merge pull request”*
- ¿Qué es un etiqueta en Git?  
*(Tag) Es un marcador que se asigna a un commit específico.*
- ¿Cómo crear una etiqueta en Git?  
*git tag <nombre>*
- ¿Cómo enviar una etiqueta a GitHub?  
*git push --tags*
- ¿Qué es un historial de Git?  
*Es un registro de todos los cambios realizados en un repositorio Git. Es una cronología de todos los commits.*
- ¿Cómo ver el historial de Git?  
*git log*  
*git log --all (muestra el historial de commits de todas las ramas)*
- ¿Cómo buscar en el historial de Git?  
*git log --decorate --all --graph --oneline (ver commits graficado)*
- ¿Cómo borrar el historial de Git?  
*git reset <nombre de archivo>*  
*git branch -d <rama> (Borra rama)*
- ¿Qué es un repositorio privado en GitHub?  
*Es un proyecto que solo tú y las personas que autorices pueden ver y acceder.*
- ¿Cómo crear un repositorio privado en GitHub?  
*Una vez iniciado en tu cuenta, hacer click en el botón “+”, nuevo repositorio, seleccionar que será privado, crear repositorio.*
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?  
*Click en la pestaña “Configuración”, selecciona “Colaboradores”, “Agregar Colaboradores”, ingresa el nombre de usuario y enviar la invitación.*
- ¿Qué es un repositorio público en GitHub?  
*Es un proyecto que cualquiera puede ver, aunque no tenga cuenta en GitHub.*
- ¿Cómo crear un repositorio público en GitHub?

*Una vez iniciado en tu cuenta, hacer click en el botón “+”, nuevo repositorio, seleccionar que será público, crear repositorio.*

- ¿Cómo compartir un repositorio público en GitHub?  
*Abre tu repositorio en GitHub, copia la URL del navegador.*

2) Realizar la siguiente actividad:

- Crear un repositorio.
  - o Dale un nombre al repositorio.
  - o Elije el repositorio sea público.
  - o Inicializa el repositorio con un archivo.
- Agregando un Archivo
  - o Crea un archivo simple, por ejemplo, "mi-archivo.txt".
  - o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
  - o Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).
- Creando Branchs
  - o Crear una Branch
  - o Realizar cambios o agregar un archivo
  - o Subir la Branch

*Ejercicio resuelto:*

<https://github.com/27mau/primer-repo/tree/main>

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS

PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I> git init
Initialized empty Git repository in D:/Mauro/Documents/Mauro/Proyectos/TUP/1. Programaci|n I/.git/
./git/
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I> cd primer-repo
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git init
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git add .
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git commit -m "Agregando mi-archivo.txt"
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git remote add origin https://github.com/27mau/primer-
repo
error: remote origin already exists.
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git remote -v
origin https://github.com/27mau/primer-repo (fetch)
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git push origin main
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Writing objects: 100% (3/3), 291 bytes | 29.00 KiB/s, done.
* main
nueva-rama
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git checkout nueva-rama
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo>
> echo "hola, archivo nuevo" > otro-archivo.txt
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo>
git add .
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git commit -m "Agregando archivo en nueva rama"
[nueva-rama 06e9a71] Agregando archivo en nueva rama
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 otro-archivo.txt
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git push origin nueva-rama
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 358 bytes | 89.00 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/27mau/primer-repo/pull/new/nueva-rama
remote:
To https://github.com/27mau/primer-repo
* [new branch] nueva-rama -> nueva-rama
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> ]
```

### 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.

*Ejercicio resuelto:*

<https://github.com/27mau/conflicto-ejercicio>

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I> git init
Initialized empty Git repository in D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programaci
n I
/.git/
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I> cd primer-repo
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git init
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git add .
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git commit -m "Agregando mi-archivo.txt"
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git remote add origin https://github.com/27mau/primer-
repo
error: remote origin already exists.
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git remote -v
origin https://github.com/27mau/primer-repo (fetch)
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git push origin main
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Writing objects: 100% (3/3), 291 bytes | 29.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/27mau/primer-repo
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git branch
* main
* main
nueva-rama
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo> git checkout nueva-rama
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo>
> echo "hola, archivo nuevo" > otro-archivo.txt
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\primer-repo>
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS

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.
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git commit -
m "Added a line in feature-branch"
    (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
-b feature-branch
ME.md
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git commit -
1 file changed, 2 insertions(+)
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git checkout
main
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git commit -
1 file changed, 1 insertion(+)
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git merge fe
ature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git add READ
ME.md
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git commit -
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> git push ori
gin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 870 bytes | 58.00 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/27mau/conflicto-ejercicio
4060621..22dea3c  main -> main
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> 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/27mau/conflicto-ejercicio/pull/new/feature-branch
remote:
To https://github.com/27mau/conflicto-ejercicio
* [new branch]   feature-branch -> feature-branch
PS D:\Mauro\Documents\Mauro\Proyectos\TUP\1. Programación I\conflicto-ejercicio> 
```