

## TP 2 GITHUB

### 1. ¿Qué es GitHub?

GitHub es una plataforma basada en la web que permite alojar repositorios de código utilizando Git. Facilita la colaboración entre desarrolladores y ofrece herramientas para gestionar versiones, realizar seguimiento de cambios y trabajar en equipo.

### 2. ¿Cómo crear un repositorio en GitHub?

1. Inicia sesión en [GitHub](#).
2. Haz clic en el ícono "+" en la esquina superior derecha y selecciona **"New repository"**.
3. Ingresa un nombre para el repositorio y una descripción (opcional).
4. Elige si será público o privado.
5. (Opcional) Agrega un archivo README, una licencia o un .gitignore.
6. Haz clic en **"Create repository"**.

### 3. ¿Cómo crear una rama en Git?

```
CopiarEditar
git branch nombre-de-la-rama
```

O si deseas crear y cambiar a la nueva rama al mismo tiempo:

```
CopiarEditar
git checkout -b nombre-de-la-rama
```

### 4. ¿Cómo cambiar a una rama en Git?

```
CopiarEditar
git checkout nombre-de-la-rama
```

O con Git moderno:

```
CopiarEditar
git switch nombre-de-la-rama
```

### 5. ¿Cómo fusionar ramas en Git?

1. Cambia a la rama principal (por ejemplo, main o master):

```
CopiarEditar
git checkout main
```

2. Fusiona la rama deseada:

```
CopiarEditar
git merge nombre-de-la-rama
```

## 6. ¿Cómo crear un commit en Git?

```
CopiarEditar
git add .
git commit -m "Mensaje del commit"
```

## 7. ¿Cómo enviar un commit a GitHub?

```
CopiarEditar
git push origin nombre-de-la-rama
```

## 8. ¿Qué es un repositorio remoto?

Es una versión almacenada en un servidor (como GitHub, GitLab o Bitbucket) que se sincroniza con tu repositorio local.

## 9. ¿Cómo agregar un repositorio remoto a Git?

```
CopiarEditar
git remote add origin URL-del-repositorio
```

Ejemplo:

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

## 10. ¿Cómo empujar cambios a un repositorio remoto?

```
CopiarEditar
git push origin nombre-de-la-rama
```

## 11. ¿Cómo tirar de cambios de un repositorio remoto?

```
CopiarEditar
git pull origin nombre-de-la-rama
```

## 12. ¿Qué es un fork de repositorio?

Un **fork** es una copia de un repositorio de otra cuenta en GitHub. Permite trabajar sobre el código de otro sin afectar el original.

## 13

### . ¿Cómo crear un fork de un repositorio?

1. Ve al repositorio en GitHub.
2. Haz clic en "**Fork**" en la parte superior derecha.

3. Espera a que GitHub copie el repositorio a tu cuenta.

## 14. ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

1. Sube los cambios a tu repositorio en GitHub.
2. Ve al repositorio original.
3. Haz clic en "**Pull Requests**" y luego en "**New Pull Request**".
4. Selecciona la rama a fusionar y agrega un mensaje explicativo.
5. Haz clic en "**Create Pull Request**".

## 15. ¿Cómo aceptar una solicitud de extracción?

1. Ve a la sección **Pull Requests** del repositorio en GitHub.
2. Selecciona la solicitud de extracción.
3. Revisa los cambios y haz clic en "**Merge Pull Request**".
4. Confirma con "**Confirm Merge**".

## 16. ¿Qué es una etiqueta en Git?

Una etiqueta (**tag**) es una referencia a un punto específico del historial de Git, generalmente usada para marcar versiones de un proyecto.

## 17. ¿Cómo crear una etiqueta en Git?

```
CopiarEditar
git tag -a v1.0 -m "Versión 1.0"
```

## 18. ¿Cómo enviar una etiqueta a GitHub?

```
CopiarEditar
git push origin --tags
```

## 19. ¿Qué es un historial de Git?

Es el registro de todos los cambios realizados en el repositorio.

## 20. ¿Cómo ver el historial de Git?

```
CopiarEditar
git log
```

O de manera simplificada:

```
CopiarEditar
git log --oneline --graph --all
```

## 21. ¿Cómo buscar en el historial de Git?

CopiarEditar  
git log --grep="palabra clave"

## 22. ¿Cómo borrar el historial de Git?

Si deseas eliminar solo los commits locales, puedes hacer:

CopiarEditar  
git reset --hard HEAD~n

Si quieres eliminar completamente el historial en GitHub, tendrías que eliminar y recrear el repositorio.

## 23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo pueden ver los colaboradores autorizados.

## 24. ¿Cómo crear un repositorio privado en GitHub?

1. En GitHub, crea un nuevo repositorio.
2. Marca la opción "**Private**" antes de crearlo.

## 25. ¿Cómo invitar a alguien a un repositorio privado en GitHub?

1. Ve a la configuración del repositorio.
2. Selecciona "**Manage Access**".
3. Haz clic en "**Invite a collaborator**" y escribe el nombre de usuario.

## 26. ¿Qué es un repositorio público en GitHub?

Es un repositorio visible para todos en GitHub.

## 27. ¿Cómo crear un repositorio público en GitHub?

1. En GitHub, crea un nuevo repositorio.
2. Marca la opción "**Public**" antes de crearlo.

## 28. ¿Cómo compartir un repositorio público en GitHub?

Simplemente comparte la URL del repositorio, por ejemplo:

CopiarEditar  
<https://github.com/usuario/repositorio>

2)

```
$ git add .

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (main|MERG
ING)
$ git commit -m "Agrego tp2"
[main 9c2f32d] Agrego tp2

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 523 bytes | 523.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Gabiitow1/UTN-TUPaD-P1
   af1c9f0..9c2f32d  main -> main

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ |
```

```
gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (main)
$ git checkout -b nuevaRama
Switched to a new branch 'nuevaRama'

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ git branch
  main
* nuevaRama

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ echo "Modificacion hecha desde nuevaRama" > tp2

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ git add .
warning: in the working copy of '02 Trabajo Colaborativo/tp2', LF will be replac
ed by CRLF the next time Git touches it
```

```

$ git status
On branch nuevaRama
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   tp2

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ git commit -m "Agregando modificacion desde NuevaRama"
[nuevaRama 9023fe9] Agregando modificacion desde NuevaRama
 1 file changed, 1 insertion(+)
 create mode 100644 02 Trabajo Colaborativo/tp2

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ git push origin nuevaRama
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 413 bytes | 413.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'nuevaRama' on GitHub by visiting:
remote:   https://github.com/Gabiitow1/UTN-TUPaD-P1/pull/new/nuevaRama
remote:
To https://github.com/Gabiitow1/UTN-TUPaD-P1
 * [new branch]   nuevaRama -> nuevaRama

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/p
rimer-repo-git/conflict-exercise/UTN-TUPaD-P1/02 Trabajo Colaborativo (nuevaRama
)
$ |

```

3)

```

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1 (
master)
$ git clone https://github.com/Gabiitow1/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 12 (delta 1), reused 9 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1 (
master)
$ cd conflict-exercise

gabriell@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/c
onflict-exercise (main)
$ git checkout -b feature-branch

```

```
gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1 (main)
$ cd conflict-exercise

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (feature-branch)
$ git add README.md

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch a9d3908] Added a line in feature-branch
1 file changed, 1 insertion(+)

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (feature-branch)
$ git checkout main
```

```
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (main)
$ git add README.md

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (main)
$ git commit -m "added a line in main branch"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (main)
$ git merge feature-branch
Updating e0383ee..a9d3908
Fast-forward
 README.md | 1 +
1 file changed, 1 insertion(+)

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/conflict-exercise (main)
```

```
$ git add README.md

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/c
onflict-exercise (main)
$ git commit -m "Resolved merge conflict"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/c
onflict-exercise (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 328 bytes | 328.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gabiitow1/conflict-exercise.git
   e0383ee..a9d3908  main -> main
```

```
gabriel@DESKTOP-MCIK1PK MINGW64 ~/Desktop/UTN Programacion/2025 PROGRAMACION 1/c
onflict-exercise (main)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gabiitow1/conflict-exercise.git
   b82631c..a9d3908  feature-branch -> feature-branch
```