

1)

### ¿Qué es GitHub?

Es una plataforma online donde distintos usuarios pueden cooperar para la creación y modificación de código de manera colaborativa.

### ¿Cómo crear un repositorio en GitHub?

Desde nuestro perfil vamos a Repositories > New. Le ponemos nombre y descripción y lo seteamos como público o privado.

### ¿Cómo crear una rama en Git?

Usando el siguiente comando en la terminal: `git branch nombre-de-la-rama`

Esto crea una nueva rama a partir de la rama actual, sin cambiarte a ella.

### ¿Cómo cambiar a una rama en Git?

Por medio del comando `git checkout nombre-de-la-rama`

### ¿Cómo fusionar ramas en Git?

Con el comando `git merge nombre-de-la-rama`

### ¿Cómo crear un commit en Git?

Con `git commit -m "Mensaje del commit"`, donde vamos a hacer un comentario de referencia sobre esta modificación.

Previamente deberemos haber hecho un `git add .` para agregar a los archivos sobre los cuales hacemos el commit.

### ¿Cómo enviar un commit a GitHub?

Después de hacer un commit, debes enviar (push) los cambios al repositorio remoto:  
`git push origin nombre-de-la-rama`

### ¿Qué es un repositorio remoto?

Es una versión del repositorio que está alojada en un servidor externo, como GitHub, y permite colaborar con otros desarrolladores de forma remota.

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

Usando el comando:

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

Esto conecta tu repositorio local con el repositorio remoto en GitHub.

¿Cómo empujar cambios a un repositorio remoto?

Con **git push origin master**

¿Cómo tirar de cambios de un repositorio remoto?

Con **git pull origin master**

¿Qué es un fork de repositorio?

Es una copia de un repositorio que se crea en tu cuenta de GitHub. Permite modificar el proyecto sin afectar el repositorio original, ideal para contribuir a otros proyectos.

¿Cómo crear un fork de un repositorio?

Desde el repositorio de interés en GitHub, haciendo click en el botón Fork.

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

Desde GitHub, haciendo click en el botón Contribute > Open pull request. Además deberemos indicar en qué es que mejora nuestro código al original.

¿Cómo aceptar una solicitud de extracción?

Abriendo la solicitud de extracción, y seleccionando Aprobar, para que los cambios se realicen.

¿Qué es una etiqueta en Git?

Una etiqueta (tag) es una referencia que se utiliza para marcar puntos específicos en la historia del repositorio, comúnmente usada para versiones (por ejemplo: v1.0.0).

¿Cómo crear una etiqueta en Git?

Con **git tag nombre-de-la-etiqueta**

¿Cómo enviar una etiqueta a GitHub?

Con **git push -tags** para todas las etiquetas. También se puede enviar una en particular con el comando **git push nombre-de-la-etiqueta**

¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un repositorio, incluyendo los cambios, fechas, autores, etc.

¿Cómo ver el historial de Git?

Con **git log** vemos todos los commit (en orden descendente).

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

Con **git log --grep**, para filtrar una búsqueda específica entre todos los commits. También se puede usar: **git log --nombre\_archivo** si necesitamos ese en particular, **git log --since="yyyy-mm-dd" --until="yyyy-mm-dd"** para buscar en un rango de fechas, o **git log --author="nombre\_autor"** para buscar por autor.

### ¿Cómo borrar el historial de Git?

Localmente, usamos **git reset** para borrar todo, **git reset nombre\_archivo** para borrar uno específico.

Remotamente, con el comando **git rebase**, para borrar commits consecutivos.

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

Es un repositorio que solo puede ser visto por ti y por las personas que tú autorices. Ideal para proyectos confidenciales o en desarrollo.

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

Al momento de crear el repositorio en GitHub seleccionar la opción Privado.

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

Yendo a Colaboradores > Agregar personas. Se debe indicar por el mail del usuario, establecer su rol y elegir el repositorio al cual será invitado.

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

Es un repositorio visible para cualquier persona en internet.

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

Al momento de crear el repositorio en GitHub seleccionar la opción Público.

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

En el repositorio de interés, desde Configuración > Acceso > Agregar personas.

## 2) Actividad GitHub N°1

- Creo repositorio


### 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 \*

 Nikivz


 / 

UTN-Programacion1-TP2


 **UTN-Programacion1-TP2 is available.**

Great repository names are short and memorable. Need inspiration? How about **literate-guide** ?

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.

- Creo archivo, lo añado, hago commit y pusheo

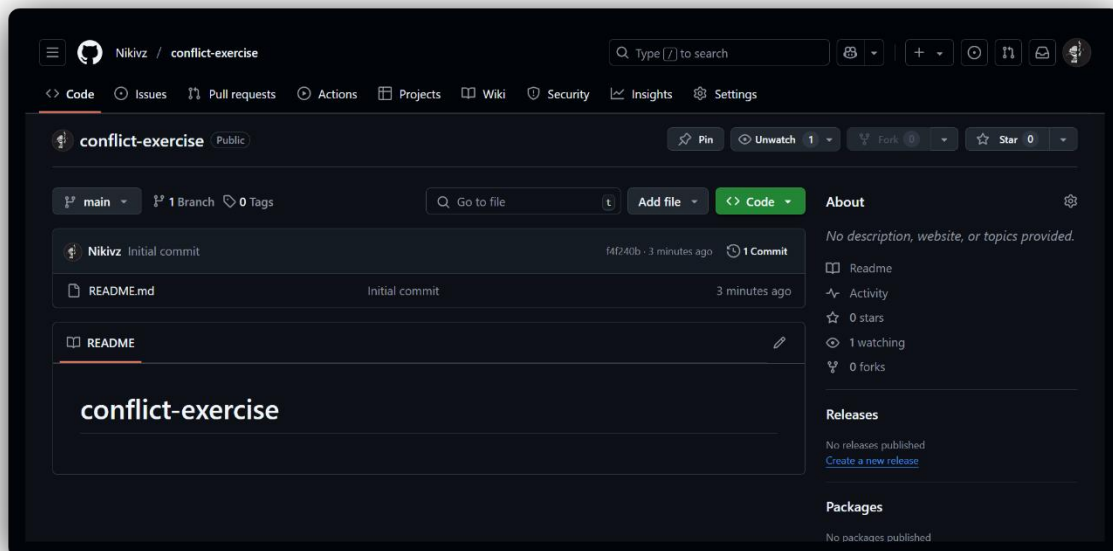
```
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 git init
Initialized empty Git repository in C:/Users/Niki/Desktop/FACU/Programacion 1/TP 2/Actividad Github 1/.git/
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 %master 21 git add .
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 %master 21 git commit -m "Agregando mi-archivo.txt"
[master (root-commit) d8938ab] Agregando mi-archivo.txt
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo.txt
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 %master git branch -M main
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 %main git remote add origin https://github.com/Nikivz/UTN-Programacion1-TP2.git
~\Desktop\FACU\Programacion 1\TP 2\Actividad Github 1 %main git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 242.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Nikivz/UTN-Programacion1-TP2.git
* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
```

- Creo nueva rama y me muevo a ella, agrego un archivo y lo subo a ella

```
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 1 > |main git branch rama
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 1 > |main git checkout rama
Switched to branch 'rama'
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 1 > |rama echo "Nueva rama creada" > mi-archivo2.txt
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 1 > |rama git commit -m "Nuevo archivo en rama"
[rama 671e545] Nuevo archivo en rama
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo2.txt
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 1 > |rama git push -u origin rama
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'rama' on GitHub by visiting:
remote:   https://github.com/Nikivz/UTN-Programacion1-TP2/pull/new/rama
remote:
To https://github.com/Nikivz/UTN-Programacion1-TP2.git
 * [new branch]      rama -> rama
branch 'rama' set up to track 'origin/rama'.
```

## 2) Actividad GitHub N°2

- Creo repositorio



- Clono repositorio a máquina local

```
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2 git clone https://github.com/Nikivz/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.
```

- Creo nueva rama, me voy a ella, modifico archivo, lo agrego y hago commit

```
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /main git checkout -b feature-branch
Switched to a new branch 'feature-branch'
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /feature-branch echo "\nEste es un cambio en la feature branch" > README.md
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /feature-branch git add README.md
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /feature-branch git commit -m "Added a line in feature-branch"
[feature-branch d0a4a1f] Added a line in feature-branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

- Vuelvo a la rama main, edito archivo, guardo cambio y hago commit.  
Hago merge, generando el conflicto y lo resuelvo conservando los dos contenidos

```
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /main git add README.md
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /main git commit -m "Added a line in main branch"
[main 7c587c7] Added a line in main branch
1 file changed, 1 insertion(+), 1 deletion(-)
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\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.
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /feature-branch git add README.md
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /feature-branch git commit -m "Resolved merge conflict"
[main e496f4e] Resolved merge conflict
~\Desktop\FACU\Programacion I\TP 2\Actividad Github 2\conflict-exercise /main git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 806 bytes | 806.00 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/Nikivz/conflict-exercise.git
 f4f248b..e496f4e main -> main
```

