

## Práctico 2: Git y GitHub

Alumno: Santarelli Gustavo Yoel

¿Qué es GitHub?

Es una plataforma en la nube que permite almacenar proyectos y así poder colaborar con otras personas mediante repositorios remotos.

¿Cómo crear un repositorio en GitHub?

Suponiendo que ya tenés creada tu cuenta de Github e iniciaste sesión en la misma, vamos a ir a la parte superior en donde está la imagen de tu usuario, le damos click y se habrá una menú desplegable, ahí buscaremos la opción “Your repositories”, una vez que entramos, nos va a aparecer un botón verde que dentro dice “New”, le damos click y nos llevará a donde crearemos nuestro nuevo repositorio, tendremos que asignarle un nombre al repositorio, decidir si queremos que sea público o privado y darle al botón verde que dice “Create repository” y listo, ahí tendremos creado nuestro nuevo repositorio.

¿Cómo crear una rama en Git?

Para crear una rama en Git tenemos que poner en la consola `git branch nombre-de-la-rama` y listo ahí está creada tu nueva rama.

¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en Git tendremos que poner en la consola `git checkout nombre-de-la-rama` y nos llevará a la rama que queremos.

¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero tendremos que estar en la rama a donde queremos fusionar los cambios, si estuviere en la rama `nombre-de-la-rama` y queremos llevar esos cambios a la rama principal que supongamos que se llama `main`, deberíamos de hacer un `git checkout main`, y luego para traer los datos de la rama tendríamos que poner un `git merge nombre-de-la-rama` y ahí traemos los cambios de rama a `main`.

¿Cómo crear un commit en Git?

Para crear un commit en Git primero tendríamos que agregar todos los cambios recientes que realizamos en el archivo, poniendo git add. Eso significa que agregamos todo lo que tenemos en la carpeta, después si podríamos realizar el commit y para hacer eso tendríamos que poner git commit -m "Mensaje de lo que realizaste" y ahí tendríamos creado tu commit.

¿Cómo enviar un commit a GitHub?

Suponiendo que ya creamos un repositorio en remoto, para enviar ese commit a Github tenemos que poner git push origin nombre-de-la-rama y así enviaríamos el archivo al repositorio remoto.

¿Qué es un repositorio remoto?

Es una copia del repositorio alojada en un servidor y te permite compartir tu código.

¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git tenemos que utilizar git remote add origin <https://github.com/usuario/repositorio.git>

¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios tendremos que utilizar git push origin nombre-de-la-rama.

¿Cómo tirar de cambios de un repositorio remoto?

Para tirar de cambios de un repositorio remoto utilizaremos el git pull origin nombre-de-la-rama.

¿Qué es un fork de repositorio?

Fork es una copia de un repositorio a tu cuenta de Github que te permite modificarlo sin afectar al original.

¿Cómo crear un fork de un repositorio?

para crear un fork de un repositorio tendremos que ir al repositorio original en Github, luego a arriba a la derecha hay un botón que dice

forks le das click y ahí podrás modificar el nombre del repositorio o la descripción, le das al botón verde de “Create fork” y listo ya tendrás tu fork creado

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

Para realizar una solicitud de pull request a un repositorio tenemos que haber hecho un cambio antes en una rama o fork, después tienes que ir a la pestaña que dice pull request luego darle al botón verde que dice pull request, seleccionar tu rama y enviar la solicitud

¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de pull requests tendremos que ir al repositorio en Github, abrir la pestaña de pull requests, ahí te saldrán los cambios que habrán, una vez que verifiques que todo está bien, podrás aceptar el “Merge pull request”

¿Qué es una etiqueta en Git?

Una etiqueta en Git es un punto específico en el historial, comúnmente usada para versiones.

¿Cómo crear una etiqueta en Git?

para crear una etiqueta en Git tendremos que utilizar `git tag -a v1.0 -m “Versión 1.0”`

¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a Github primero tendremos que utilizar `git push origin v1.0` y para enviarlas a todas `git push origin --tags`

¿Qué es un historial de Git?

El historial en Git es un registro donde están guardados todos los commits realizados en un repositorio.

¿Cómo ver el historial de Git?

Para ver el historial de Git tenemos que usar `git log`

¿Cómo buscar en el historial de Git?

Para buscar en el historial tenemos que usar git log  
-grep="palabra clave"

¿Cómo borrar el historial de Git?

Para borrar el historial tenemos que utilizar  
rm -rf .git  
git init

Esto elimina todo por lo tanto no es recomendable hacerlo.

¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo quien lo crea y las personas a que invites pueden ver o modificar el repositorio.

¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio en privado, antes de crearlo tienes que seleccionar la opción "private"

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

Para invitar a alguien hay que entrar al repositorio, ir a "settings" después a "Collaborators" e invita ingresando el nombre de usuario.

¿Qué es un repositorio público en GitHub?

Es un repositorio el cual es visible para todos en GitHub.

¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio en privado, antes de crearlo tienes que seleccionar la opción "public"

¿Cómo compartir un repositorio público en GitHub?

Para compartir el repositorio tenés q copiar y enviar la URL del repositorio


## Actividad 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 \*

 GYoelSantarelli ▾

Repository name \*

/

✔ mi-repositorio is available.

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

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.](#)



You are creating a public repository in your personal account.

Create repository

```
MINGW64:/c/Users/Yoel/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2
$ git init
Initialized empty Git repository in c:/Users/Yoel/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2/.git/

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (master)
$ git remote add origin https://github.com/GYoelsantarelli/mi-repositorio.git

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (master)
$ git branch -M main
```

```
MINGW64:/c/Users/Yoel/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (main)
$ git add .

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (main)
$ git commit -m "Agregando mi-archivo.txt"
[main (root-commit) cf3b411] Agregando mi-archivo.txt
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo.txt.txt

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 260 bytes | 260.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/GYoelsantarelli/mi-repositorio.git
* [new branch]      main -> main
```

```
Yoe1@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (main)
$ git branch "tp-2"

Yoe1@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (main)
$ git checkout tp-2
Switched to branch 'tp-2'

Yoe1@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (tp-2)
$ git add .

Yoe1@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (tp-2)
$ git commit -m "Agregando rama-2.txt"
[tp-2 52c5ca5] Agregando rama-2.txt
1 file changed, 1 insertion(+)
create mode 100644 rama2.txt.txt

Yoe1@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 2 (tp-2)
$ git push origin tp-2
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), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'tp-2' on GitHub by visiting:
remote:   https://github.com/GYoelSantarelli/mi-repositorio/pull/new/tp-2
remote:
To https://github.com/GYoelSantarelli/mi-repositorio.git
 * [new branch]      tp-2 -> tp-2
```


## Actividad 3

# 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 \*

 GYoelSantarelli ▾

Repository name \*

/ conflict-exercise

✔ conflict-exercise is available.

Great repository names are short and memorable. Need inspiration? How about **shiny-eureka** ?

Description (optional)

Mi repositorio de Programación 1.



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](#).



You are creating a public repository in your personal account.


Create repository



```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3
$ git clone https://github.com/GYoelsantarelli/conflict-exercise.git
Cloning into 'conflict-exercise'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
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.

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3
$ cd conflict-exercise/
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

 README: Bloc de notas

Archivo Edición Formato Ver Ayuda


```
# conflict-exercise
Mi repositorio de Programación 1.
Este es un cambio en la feature branch|
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (feature-branch)
$ git add .

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch bba1567] Added a line in feature-branch
1 file changed, 1 insertion(+)

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (feature-branch)
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

 \*README: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# conflict-exercise
Mi repositorio de Programación 1.
Este es un cambio en la main branch|
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (main)
$ git add README.md
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (main)
$ git commit -m "Added a line in main branch"
[main 08e355a] Added a line in main branch
1 file changed, 1 insertion(+)
```

```
Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/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.
```

README.md ! Merging: README.md ! X

C: > Users > Yoel > Desktop > Uni > Programacion I > TP 2 GIT - GITHUB > Actividad 3 > conflict-exercise > README.md > # conflict-exercise

Incoming	Current
1 # conflict-exercise	1 # conflict-exercise
2 Mi repositorio de Programación 1.	2 Mi repositorio de Programación 1.
3 Este es un cambio en la feature branch	3 Este es un cambio en la main branch

Result C:\Users\Yoel\Desktop\Uni\Programacion I\TP 2 GIT - GITHUB\Actividad 3\conflict-exercise\README.md 0 Conflicts Remaining

```
1 # conflict-exercise
2 Mi repositorio de Programación 1.
3 Este es un cambio en la feature branch
4 Este es un cambio en la main branch
```

```

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/conflict-exercise (main|MERGING)
$ git commit -m "Resolved merge conflict"
[main bdf2bda] Resolved merge conflict

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/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), 773 bytes | 773.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/GYoelsantarelli/conflict-exercise.git
    f5fe151..bdf2bda  main -> main

Yoel@DESKTOP-II36KTA MINGW64 ~/Desktop/Uni/Programacion I/TP 2 GIT - GITHUB/Actividad 3/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/GYoelsantarelli/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/GYoelsantarelli/conflict-exercise.git
 * [new branch]      feature-branch -> feature-branch

```


**conflict-exercise**
Public
Pin
Unwatch

main
2 Branches
0 Tags

Add file
Code


**GYoelSantarelli**
Resolved merge conflict
bdf2bda · 2 minutes ago
4 Commits


**README.md**
Resolved merge conflict
2 minutes ago


**README**


# conflict-exercise

Mi repositorio de Programación 1. Este es un cambio en la feature branch Este es un cambio en la main branch