

## TP 2 Mauro Cuquejo (Git-Github)

1)

### ¿Qué es github?

Es una plataforma que permite a desarrolladores trabajar con repositorios de git de forma colaborativa. Permite conectarse entre desarrolladores, armar repositorios y forks para trabajar localmente y subir cambios a través de pull requests.

|

### ¿Cómo crear un repositorio en Github?

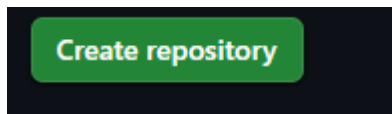
Se debe iniciar sesión.

Desde la pestaña Repositories, seleccionar el botón New:



Se define un nombre de repositorio, si es público o privado, si se quiere agregar un readme y el .gitignore y, por último, el tipo de licencia que se desee utilizar.

Luego hay que darle al botón Create repository.



### ¿Cómo crear una rama en git?

Se debe ejecutar el comando `git Branch nombre_de_la_rama`

Donde `nombre_de_la_rama` representa el nombre que se le dará al nuevo Branch.

### ¿Cómo cambiar a una rama en Git?

Se debe ejecutar el comando `git checkout nombre_de_la_rama`

Donde `nombre_de_la_rama` representa el nombre de la rama a la que se quiere cambiar.

### Como fusionar ramas en git?

A través del comando `git merge nombre_de_la_rama`

Donde `nombre_de_la_rama` representa el Branch que uno quiere mergear sobre el Branch en donde se encuentre posicionado.

### ¿Cómo crear un commit en Git?

Suponiendo que los archivos ya se encuentran en el staging área, simplemente se debe ejecutar `git commit -m "mensaje del commit"`

### ¿Cómo enviar un commit a github?

Se debe ejecutar el comando `git commit -m "mensaje"`

Si es la primera vez, se debe ejecutar la configuración de mail y nombre de usuario a través de los comandos

`Git config --global user.email mi@email.com`

`Git config --global user.name "mauro Cuquejo"`

`Git remote add origin https://github.com/nombre-del-repo.git`

Y luego se suben los cambios a través del comando:

`git push -u origin nombre_de_la_rama`

Donde `nombre_de_la_rama` es el Branch hacia donde se quieren subir los cambios de forma remota. Luego de la primera vez que se suben los cambios, se puede ejecutar simplemente

`git push.`

### ¿Qué es un repositorio remoto?

Es un espacio en la herramienta de versionado que se utilice, por ejemplo github, en el que quedan guardados los cambios de un repositorio de git, para poder utilizarse desde cualquier lugar, simplemente descargándolo de forma local.

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

A través del comando `git remote add origin https://github.com/nombre-del-repo.git`

Donde `nombre-del-repo` es el nombre que se le asignó al repositorio remoto.

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

Entiendo que la pregunta es similar a la de enviar un commit a github:

`git push -u origin nombre_de_la_rama`

Donde nombre de la rama es el Branch hacia donde se quieren subir los cambios de forma remota. Luego de la primera vez que se suben los cambios, se puede ejecutar simplemente

`git push.`

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

Ejecutando el comando `git pull origin nombre-de-la-rama`

Si no es la primera vez, basta con ejecutar `git pull`

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

Es una copia del proyecto original. A diferencia de lo que ocurre al conectarse directamente al repositorio remoto original, esta es una copia que se crea dentro del repositorio remoto propio, de forma independiente. Luego es necesario solicitar un pull request si es que se quisieran agregar los cambios sobre el repositorio original.

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



Desde github, parado sobre un proyecto al que se tenga acceso, se debe presionar el botón Fork

Se puede modificar el nombre del repositorio y la descripción.

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

Una vez commiteados y pusheados los cambios en el fork, se debe presionar el botón Compare and Pull Request desde github. Luego de agregar un nombre al pr, se agrega una breve descripción de los cambios y se presiona "Create pull request"

### ¿Cómo aceptar una solicitud de extracción?

Primero se debe presionar en Review changes, desde la solapa de Pull Request del repositorio remoto. Acá se pueden agregar comentarios a modo de feedback o aprobar el PR.

Una vez aprobado, el usuario puede mergear el Branch del fork al remoto.

### ¿Qué es una etiqueta en git?

Una etiqueta, o tag, es simplemente una herramienta que permite versionar una entrega o release. No se pueden agregar commits a una rama de un tag.

### ¿Cómo crear una etiqueta en Git?

Simplemente ejecutando el comando `git tag nombre-del-tag`

### ¿Cómo enviar una etiqueta a GitHub?

Ejecutando el comando `git push origin nombre-del-tag`

### ¿Qué es un historial de Git?

Según el archivo Git Cheatsheet, se habla del historial del HEAD (reflog), que permite visualizar todos los cambios en branches, commits y eliminaciones de los mismos.

También está el historial de commits (log), en donde se pueden ver todos los cambios subidos desde el staging área al History.

### ¿Cómo ver el historial de git?

Si la pregunta anterior se refiere al reflog, ejecutando el comando `git reflog`

```
PS E:\Users\mauro\OneDrive\Escritorio\cursos\utn\programacion-1\UTN-TUPaD-P1> git reflog
d9cde6d (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: commit: version preliminar del tp 02
c08d4db HEAD@{1}: clone: from https://github.com/mauro-cuquejo/UTN-TUPaD-P1.git
```

Sino, sería ejecutando el comando `git log`.

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

Si se quiere buscar por el historial de un archivo, se debe ejecutar

`git log --follow nombre-del-archivo`

### ¿Cómo borrar el historial de Git?

- Con el comando `git reset --modo HEAD^` si se quiere eliminar el ultimo commit  
Donde *modo* puede ser *soft*, *mixed* o *hard*, dependiendo de si se quiere modificar o no el staging área.
- También se puede reemplazar `HEAD^` por `HEAD^numero` donde *numero* es la cantidad de commits anteriores que se quieren borrar
- Con `git reset --modo hash-commit`, se puede volver a un commit específico, reemplazando *hash-commit* por el hash del commit al que se quiere regresar
- Y por último, se puede ejecutar `git revert hash-commit` que además de volver al commit especificado, se genera un commit nuevo con el estado de dicho commit.

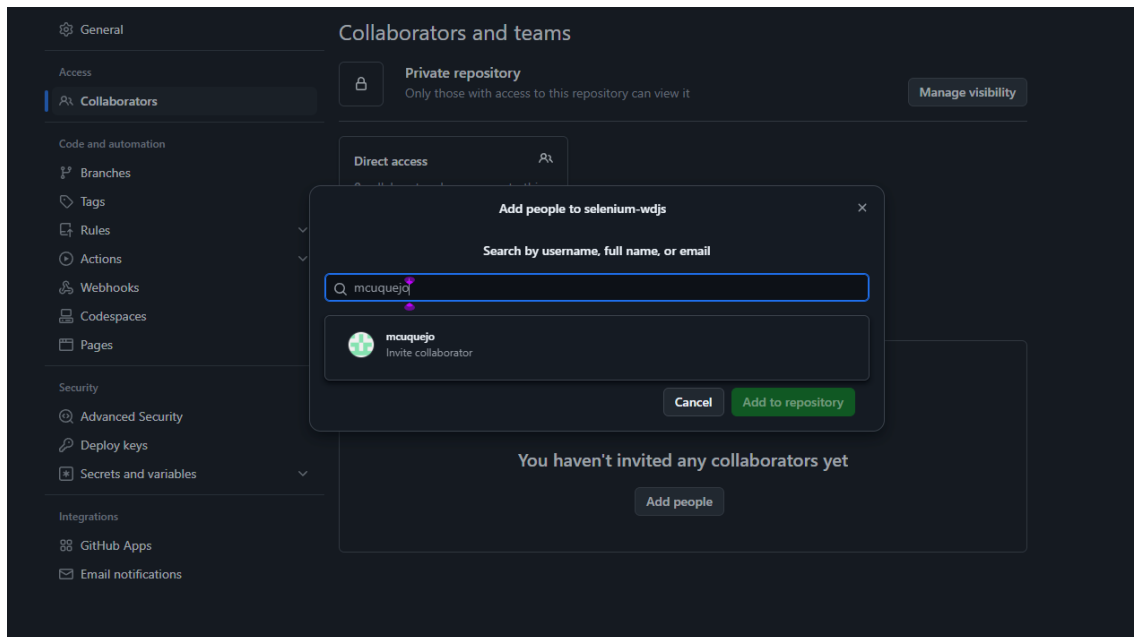
### ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que no se puede acceder abiertamente. Es necesario tener los permisos para poder visualizarlo.

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

De la misma forma en que creamos un repositorio, en las configuraciones, se puede definir como privado.

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



Desde el repositorio privado, se debe ingresar a la solapa settings, Collaborators, si se tiene un factor de autenticación, se debe ingresar el código, y luego Add people.

Simplemente buscando el nombre de la cuenta a la que le queremos dar acceso, podremos seleccionar al usuario y darle permisos, presionando el botón Add to repository.

## ¿Qué es un repositorio público en GitHub?

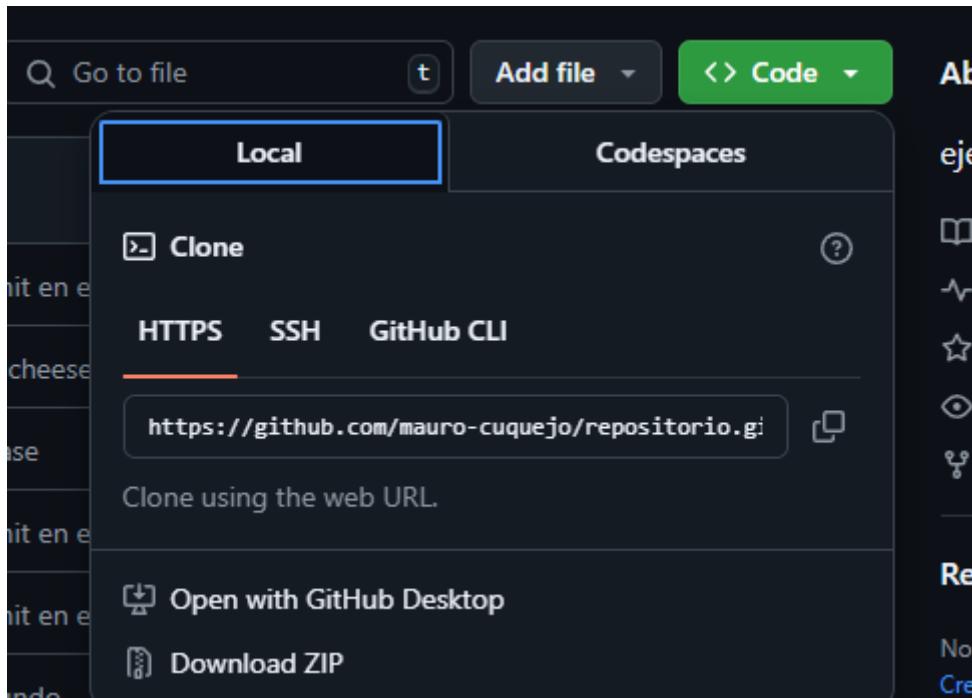
Es un repositorio al que se puede acceder simplemente a través de su url. A diferencia de los privados, Github mantiene los proyectos públicos dentro de un programa “github archive program”, que permiten la preservación de dicho repo.

## ¿Cómo crear un repositorio público en GitHub?

Desde la configuración al crear el repositorio, se selecciona la opción Publico.

## ¿Cómo compartir un repositorio público en github?

La forma más sencilla es simplemente compartiendo la URL, ya sea http o SSH, si se solicitan certificados de seguridad



2) <https://github.com/mauro-cuquejo/repositorio-para-tp-2>

```
PS E:\Users\mauro\OneDrive\Escritorio\cursos\utn\programacion-1\repositorio-prueba> git log
commit 95d5690e44f71f3635b93371a5cc7c9368623296 (HEAD -> branch-de-prueba, origin/branch-de-prueba)
Author: Mauro Cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 21:34:30 2025 -0300

    Modificando mi branch de prueba

commit d8fbb8f8143da2df93bf5734bea84deffb6efb8b (origin/main, main)
Author: Mauro Cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 21:32:54 2025 -0300
```

3) <https://github.com/mauro-cuquejo/conflict-exercise>

```
PS E:\Users\mauro\OneDrive\Escritorio\cursos\utn\programacion-1\conflict-exercise\conflict-exercise> git log
commit 4eb09ff3a42169da54f0ac91b031d0798398dfeb (HEAD -> main, origin/main, origin/HEAD)
Merge: bc86343 2d3a4f7
Author: Mauro Cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 23:00:13 2025 -0300

    Resolved merge conflict

commit bc8634343c1067029c739a77effc2131152c8584
Author: Mauro Cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 22:59:03 2025 -0300

    Added a line in main branch

commit 2d3a4f79d6dcaa295309f786113eb9e3e9868505 (origin/feature-branch, feature-branch)
Author: Mauro Cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 22:56:16 2025 -0300

    Added a line in feature-branch

commit af2db349f03cc34b9fbf0c59280e30edc6db37
Author: mauro-cuquejo <mauro.cuquejo@gmail.com>
Date: Sun Mar 30 22:53:44 2025 -0300

    Initial commit
PS E:\Users\mauro\OneDrive\Escritorio\cursos\utn\programacion-1\conflict-exercise\conflict-exercise>
```

