

- **¿Qué es GitHub?**

Es una comunidad donde podemos compartir nuestros repositorios. Es similar a una red social, podemos cargar repositorios, seguir otros usuarios, clonar repositorios, modificarlos, etc

- **¿Cómo crear un repositorio en GitHub?**

Hacer una cuenta en GitHub, ir a “New Repository” y configurarlo de la manera que uno quiera. También podemos hacer un Push de nuestro repositorio local hacia GitHub

- **¿Cómo crear una rama en Git?**

Usamos el comando “git Branch” seguido del nombre que le queremos asignar a esa rama.
Ejm.: git branch *nombre_rama*

- **¿Cómo cambiar a una rama en Git?**

Usamos el comando “git checkout” seguido del nombre de la rama a la que queremos cambiar.

- **¿Cómo fusionar ramas en Git?**

Usamos el comando “merge”

- **¿Cómo crear un commit en Git?**

Para crear una instantánea del proyecto hacemos un commit , de la siguiente manera:

Git add . (*ósea todo lo que haya dentro de la carpeta*)

Git commit -m “el nombre de las modificaciones que hicimos”

- **¿Cómo enviar un commit a GitHub?**

Se envía haciendo un “push” de la siguiente manera:

git push origin (*nuestro repositorio remoto*) main (*la rama que queremos pushear*).

- **¿Qué es un repositorio remoto?**

Es un clon de un proyecto que se encuentra en un servidor remoto.

- **¿Cómo agregar un repositorio remoto a Git?**

Haciendo un git clone <url del repo> Crea una copia local de un repositorio remoto.

- **¿Cómo empujar cambios a un repositorio remoto?**

Usando el comando git push.

- **¿Cómo tirar de cambios de un repositorio remoto?**

Con el comando git pull, este descarga y fusiona los cambios del repositorio remoto en el repositorio local.

- **¿Qué es un fork de repositorio?**

Es una copia de un repositorio creada en una cuenta diferente permitiendo desarrollar cambios sin afectar el original.

- **¿Cómo crear un fork de un repositorio?**

Ingresamos con nuestra cuenta de GitHub, presionamos el botón “fork”, agregamos descripción y le damos al botón “create fork”

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

Luego de hacer el “fork”, clonamos el proyecto en nuestro repositorio local, realizamos los cambios, creamos una rama , hacemos el commit y lo pusheamos, vamos a GitHub y tocamos el botón “compare & pull request”, nos pregunta si lo queremos hacer en el repositorio base, confirmamos tocando el botón “create pull request”.

- **¿Cómo aceptar una solicitud de extracción?**

Confirmando el merge.

- **¿Qué es una etiqueta en Git?**

Son un tipo de puntero para marcar versiones específicas en un repositorio.

Se asocian a un punto específico en nuestro histórico, existen dos tipos:

Anotadas: tienen un msj asociado y se almacenan como objetos por separado.

Simples: son simplemente un puntero de referencia.

- **¿Cómo crear una etiqueta en Git?**

Con el comando “git tag” seguido del nombre.

- **¿Cómo enviar una etiqueta a GitHub?**

Usando el comando “git push”.

- **¿Qué es un historial de Git?**

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

- **¿Cómo ver el historial de Git?**

Usando el comando “git log”.

- **¿Cómo buscar en el historial de Git?**

Usando el comando “git log” con la opción “-L”.

- **¿Cómo borrar el historial de Git?**

Podemos usar el comando “git reset” seguido el nombre del commit hasta el cual queremos conservar, el nombre lo sabremos después de utilizar el comando “git log”.

- **¿Qué es un repositorio privado en GitHub?**

Es un repositorio al cual solo puede acceder su autor o personas autorizadas por el.

- **¿Cómo crear un repositorio privado en GitHub?**

A la hora de crearlo en “new repository” está la opción para configurarlo como “privado”.

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

En la ventana de configuración del repositorio, ingresar a “colaboradores”, tocamos “agregar personas” , buscamos el usuario que deseamos agregar y el rol que queremos darle.

- **¿Qué es un repositorio público en GitHub?**

Un repositorio al que puede acceder cualquier persona con cuenta en GitHub.

- **¿Cómo crear un repositorio público en GitHub?**

De la misma forma que el privado que expliqué mas arriba, pero tocando la opción “publico”.

- **¿Cómo compartir un repositorio público en GitHub?**

Compartiendo la URL del repositorio.

URL:

<https://github.com/GasTonPerez17/UTN-TUP-P1.git>

gitbash

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise
$ git clone https://github.com/GasTonPerez17/conflict-exercise
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.
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise
$ cd conflict-exercise
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (main)
$ git branch feature-branch
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (main)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (main)
$ git branch
  feature-branch
* main
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (feature-branch)
$ git add README.md
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 533e054] Added a line in feature-branch
1 file changed, 1 insertion(+)
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-exercise/conflict-exercise (feature-branch)
$ git branch main
fatal: a branch named 'main' already exists
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (feature-branch)  
$ git branch  
* feature-branch  
main
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (feature-branch)  
$ git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main)  
$ git add README.md
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main)  
$ git commit -m "Added a line in main branch"  
[main 19ebf9c] Added a line in main branch  
1 file changed, 1 insertion(+)
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/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.
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main|MERGING)  
$  
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main|MERGING)  
$ git add README.md
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main|MERGING)  
$ git commit -m "Resolved merge conflict"  
[main 5e30ea3] Resolved merge conflict
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/conflict-exercise (main)  
$ git push origin 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), 840 bytes | 60.00 KiB/s, done.  
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), done.  
To https://github.com/GasTonPerez17/conflict-exercise  
298c09f..5e30ea3 main -> main
```

```
Usuario@DESKTOP-60NSQCD MINGW64 ~/Desktop/conflict-  
exercise/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/GasTonPerez17/conflict-  
exercise/pull/new/feature-branch  
remote:  
To https://github.com/GasTonPerez17/conflict-exercise  
* [new branch]      feature-branch -> feature-branch
```

Vs conflict

```
# conflict-exercise  
tp2.3  
<<<<<< HEAD  
Este es un cambio en la main branch  
=====  
Este es un cambio en la feature branch.  
>>>>>> feature-branch
```