

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Práctico 2: Git y GitHub *Actividades*

Actividad 1:

1. ¿Qué es GitHub?

GitHub es una plataforma que permite a los desarrolladores almacenar, gestionar y compartir código utilizando Git, un sistema de control de versiones. Facilitando la colaboración en proyectos y el seguimiento de cambios.

2. ¿Cómo crear un repositorio en GitHub?

1. Primero inicia sesión en GitHub, luego haz clic en el botón "+" en la esquina superior derecha y selecciona "Nuevo repositorio". Decide un nombre para el repositorio, y elige que características deseas que tenga, puedes seleccionar si quieres que el repositorio sea público o privado y también puedes agregar opcionalmente un README. Finalmente haz clic en "Crear repositorio".

3. ¿Cómo crear una rama en Git?

Desde la terminal, se usa el siguiente comando: `git branch nombre_de_la_rama`

4. ¿Cómo cambiar a una rama en Git?

Se usa el siguiente comando:

```
git checkout nombre_de_la_rama
```

5. ¿Cómo fusionar ramas en Git?

Primero, cambia a la rama donde quieres hacer la fusión y luego se usa: `git merge nombre-de-la-rama`

¿Cómo crear un commit en Git?

Primero asegúrate de haber agregado los archivos con “`git add`” y luego se usa el siguiente comando para crear un commit con mensaje:

```
git commit -m "Mensaje descriptivo"
```

7. ¿Cómo enviar un commit a GitHub?

Después de realizar un commit, se usa el siguiente comando para enviar los cambios al repositorio remoto:

```
git push origin nombre_de_la_rama
```

8. ¿Qué es un repositorio remoto?

Es un repositorio que está alojado en una plataforma como GitHub o GitLab y permite compartir el código con otros colaboradores.

9. ¿Cómo agregar un repositorio remoto a Git?

Se usa el siguiente comando:

```
git remote add origin URL_DEL_REPOSITORIO
```

10. ¿Cómo empujar cambios a un repositorio remoto?

Se usa el siguiente comando: `git push origin nombre_de_la_rama`

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

Se usa el siguiente comando: `git pull origin nombre_de_la_rama`

12. ¿Qué es un fork de repositorio?

Es una copia de un repositorio ajeno que se aloja en tu cuenta de GitHub, permitiéndote modificar el código sin afectar el original.

13. ¿Cómo crear un fork de un repositorio?

Primero ve al repositorio en GitHub y luego haz clic en el botón "Fork" en la esquina superior derecha.

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

Ve a tu fork en GitHub, cambia algo y envíalo ("push"). Luego en GitHub, selecciona "Pull Request" y sigue los pasos para enviar la solicitud.

15. ¿Cómo aceptar una solicitud de extracción?

Ve al repositorio original y en la pestaña "Pull Requests" selecciona la solicitud, revisa los cambios y haz clic en donde dice "Merge pull request".

16. ¿Qué es una etiqueta en Git?

Una etiqueta es una marca para señalar puntos importantes en la historia de tu proyecto, como una nueva versión.

17. ¿Cómo crear una etiqueta en Git? Se usa el siguiente comando: `git tag nombre_de_la_etiqueta`

18. ¿Cómo enviar una etiqueta a GitHub?

`git push origin nombre_de_la_etiqueta`

19. ¿Qué es un historial de Git?

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

20. ¿Cómo ver el historial de Git?

Se usa el siguiente comando:

`git log`

21. ¿Cómo buscar en el historial de Git?

Se usa el siguiente comando: `git log --grep="palabra_clave"`

22. ¿Cómo borrar el historial de Git?

Si deseas eliminar completamente el historial de Git, puedes borrar el repositorio con `rm -rf .git`, aunque esto eliminará todo el progreso. Si solo quieres modificar el historial sin eliminar el repositorio, puedes sobrescribir commits con el comando `"git rebase"`.

23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio accesible solo para personas autorizadas.

24. ¿Cómo crear un repositorio privado en GitHub?

Al crear un repositorio, selecciona la opción `Privado`.

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

Primero ve a "Configuración", luego a "Colaboradores". Agrega al usuario y envía la invitación.

26. ¿Qué es un repositorio público en GitHub?

Es un repositorio accesible para cualquier usuario.

27. ¿Cómo crear un repositorio público en GitHub?

Selecciona "Público" al crear el repositorio.

28. ¿Cómo compartir un repositorio público en GitHub?

Envía la URL del repositorio a quien lo necesite.

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 * KEY1A / Repository name * TP2_Programacion
TP2_Programacion is available.

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

Description (optional)

Trabajo Practico N2 de Programacion UTN

- ☒  **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.

Windows PowerShell

```
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git add .
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git commit -m "Agregando mi_archivo.txt"
[main ef8cf05] Agregando mi_archivo.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mi_archivo.txt
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git push origin main
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), 297 bytes | 297.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/KEY1A/TP2_Programacion.git
 25eac1d..ef8cf05  main -> main
PS C:\Users\keyla\utn\tp2\TP2_Programacion> |
```

```
Windows PowerShell
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git branch test
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git checkout test
Switched to branch 'test'
PS C:\Users\keyla\utn\tp2\TP2_Programacion> ni archivo2.txt

Directory: C:\Users\keyla\utn\tp2\TP2_Programacion

Mode                LastWriteTime         Length Name
----                -
-a-----          4/2/2025   4:52 PM             0 archivo2.txt

PS C:\Users\keyla\utn\tp2\TP2_Programacion> git add .
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git commit -m "Agregando archivo2.txt"
[test 08e2e13] Agregando archivo2.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo2.txt
PS C:\Users\keyla\utn\tp2\TP2_Programacion> git push -u origin test
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 294 bytes | 294.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'test' on GitHub by visiting:
remote:   https://github.com/KEY1A/TP2_Programacion/pull/new/test
remote:
To https://github.com/KEY1A/TP2_Programacion.git
 * [new branch]      test -> test
branch 'test' set up to track 'origin/test'.
PS C:\Users\keyla\utn\tp2\TP2_Programacion> |
```

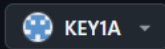
Actividad 3:

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 *



Repository name *

/ conflict-exercise

✔ conflict-exercise is available.

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

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.](#)

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.

```
PS C:\Users\keyla\utn> mkdir conflict_ejercicio
```

```
Directory: C:\Users\keyla\utn
```

Mode	LastWriteTime	Length	Name
d-----	4/2/2025 4:57 PM		conflict_ejercicio

```
PS C:\Users\keyla\utn> cd .\conflict_ejercicio\  
PS C:\Users\keyla\utn\conflict_ejercicio> git clone https://github.com/KEY1A/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.  
PS C:\Users\keyla\utn\conflict_ejercicio> cd conflict-exercise  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> |
```

```
Windows PowerShell X + v  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git checkout feature-branch  
Switched to branch 'feature-branch'  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> notepad README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git add README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git commit -m "Added a line in feature-branch"  
[feature-branch bb97cc3] Added a line in feature-branch  
1 file changed, 2 insertions(+), 1 deletion(-)  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> notepad README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git add README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git commit -m "Added a line in main branch"  
[main 8118d87] Added a line in main branch  
1 file changed, 2 insertions(+), 1 deletion(-)  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> 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.  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> notepad README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git add README.md  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git commit -m "Resolved merge conflict"  
[main dbb7798] Resolved merge conflict  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> git push origin main  
Enumerating objects: 11, done.  
Counting objects: 100% (11/11), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (9/9), 861 bytes | 861.00 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/KEY1A/conflict-exercise.git  
5bd27ad..dbb7798 main -> main  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> 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/KEY1A/conflict-exercise/pull/new/feature-branch  
remote:  
To https://github.com/KEY1A/conflict-exercise.git  
* [new branch] feature-branch -> feature-branch  
PS C:\Users\keyla\utn\conflict_ejercicio\conflict-exercise> |
```