

TUPaD - UTN

31/03/25

PROGRAMACIÓN 1 TRABÁJO PRÁCTICO 2 – TEMA DEL TRABÁJO

PROFESOR: Ariel Enferrer

ESTUDIANTE: Federico Garcia Bengolea

COMISION: M2025-14

Ejercicios:

 Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

• ¿Qué es GitHub?

- GitHub es una plataforma online que se utiliza para almacenar proyectos de diferentes indoles y, más sobre todo, el control de versionado de los mismos.

¿Cómo crear un repositorio en GitHub?

 Se pueden crear dentro de la plataforma de GitHub, se debe hacer click en el nombre del usuario o logo, luego buscar la sección de "Tus Repositorios" y tocando el botón "Nuevo".
 Allí mismo solicita datos para la creación siendo el nombre, una descripción y el alcance de privacidad (privado o público).

• ¿Cómo crear una rama en Git?

- Dentro de la consola (Bash o CMD) y estando posicionado en la carpeta main, se debe tipear git branch nombre-de-rama para que se cree una rama.

• ¿Cómo cambiar a una rama en Git?

- Para cambiar de una rama a otra se debe utilizar el comando git checkout nombre-de-rama.

• ¿Cómo fusionar ramas en Git?

- Para fusionar dos ramas en Git se utiliza el comando *git merge* dev siendo dev el repositorio el cual se fusionará con el actual en uso.

• ¿Cómo crear un commit en Git?

- Para crear un commit previamente se debe tener ya agregados los archivos del proyecto, luego ejecutar el comando git commit -m "Mensaje del commit".

• ¿Cómo enviar un commit a GitHub?

- El commit se envía a GitHub utilizando el comando git push origin main, siendo origin el repositorio local y main el repositorio de GitHub

• ¿Qué es un repositorio remoto?

 Un repositorio remoto es un lugar donde poder almacenar los proyectos el cual se puede acceder desde cualquier parte del mundo

¿Cómo agregar un repositorio remoto a Git?

 Para crear un repositorio remoto en Git se debe crear primero el repositorio en GitHub, luego hay dos opciones, la primera te lo brinda la plataforma indicando los comandos que se debe utilizar para unirlos. La otra opción es conocer los comandos y ejecutarlos manualmente desde el Símbolo del Sistema (CMD – BASH – Powershell VSC)

¿Cómo empujar cambios a un repositorio remoto?

- Para empujar cambios a un repositorio remoto se debe ejecutar el comando *git push origin main*, el cual previamente debe estar sincronizado a repositorio.

· ¿Cómo tirar de cambios de un repositorio remoto?

- Para tirar cambios se debe ejecutar el comando *git pull origin* main, el mismo se encarga de integrar los cambios desde otros repositorios o ramas locales

• ¿Qué es un fork de repositorio?

 Un FORK de un repositorio es la bifurcación del proyecto en GitHub, el cual permite "descargar" o "copiar" el proyecto para realizar modificaciones o actualizaciones sin perjudicar el original.
 Se utiliza mucho en proyectos de código abierto.

• ¿Cómo crear un fork de un repositorio?

 Se debe crear yendo al repositorio del proyecto, seleccionar el botón llamado "FORK" y luego completar la creación del repositorio.

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

 La solicitud de extracción se realiza desde GitHub solicitando permisos al propietario del repositorio, para poder trabajar con el mismo código y poner unirlo en caso de que la solicitud sea aceptada.

• ¿Cómo aceptar una solicitud de extracción?

 Desde la plataforma llega la notificación sobre la recepción de una extracción, en la cual se validan los cambios actualizados con el repo original, siempre y cuando el autor lo acepte, se puede unir o no dicha actualización.

¿Qué es un etiqueta en Git?

- Una etiqueta o "tag" en Git se realiza para tener un punto de control bajo el historial de los commits realizados, es como un commits "más importante" y a nivel general del proyecto.

¿Cómo crear una etiqueta en Git?

- Para crear una etiqueta se utiliza el comando git tag -m "mensaje del tag"

¿Cómo enviar una etiqueta a GitHub?

- Se realiza con el comando *git pull origin tag* siendo *tag* la etiqueta a subir, también es posible enviar todas las etiquetas con el *git pull* origin --tags.

• ¿Qué es un historial de Git?

 El historial de Git hace referencia a todos los cambios que se llevaron acabo sobre un proyecto, es el registro de todos los commits y tags que tuvo el repositorio en donde queda almacenado todas las actualizaciones y/o cambios que se realizaron.

¿Cómo ver el historial de Git?

- Para ver el historial de Git se debe utilizar el comando git log o también git log -online para un resumen de todos.

• ¿Cómo buscar en el historial de Git?

 Para buscar dentro del historial hace falta el ID del commits/tags especifico o buscar con el anterior comando para ver todo el historial.

¿Cómo borrar el historial de Git?

 No es posible borrar el historial de Git, más que poder forzar a su escritura desde un punto (commit o tag) especifico.
 Aunque si se borrar el archivo git se eliminarían todos los registros de las versiones.

• ¿Qué es un repositorio privado en GitHub?

- Un repositorio privado significa que es un lugar para almacenar un proyecto en el cual solo tiene acceso el dueño y sus colaboradores, nadie más puede visualizarlo o buscarlo.

¿Cómo crear un repositorio privado en GitHub?

- De la misma forma en la que se crear un repositorio común, solo que en el momento de cargar los datos se debe especificar que alance tiene la privacidad, ahí se debe elegir "privado".

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

- Para invitar a un repo privado se debe agregar a los usuarios desde la configuración del proyecto.

• ¿Qué es un repositorio público en GitHub?

- Un repositorio publico significa que cualquier persona podrá ver el código del proyecto, como también realizar forks y solicitar pull-requests para fusionar los cambios con el código original.

· ¿Cómo crear un repositorio público en GitHub?

 De la misma forma en la cual se crea un nuevo repositorio, se debe aclarar el alcance del mismo seleccionado la opción "publico".

• ¿Cómo compartir un repositorio público en GitHub?

- Se puede compartir usando el link del repositorio desde la barra de búsqueda o desde el botón "Code" dentro del repo, allí figura también el link. Otra opción sería realizar una bifurcación.

2) Actividad de Branch

https://github.com/FeddericoGarcia/utn-conflict-exercise

```
Switched to branch 'feature-branch'
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat

On branch feature-branch
Untracked files:
    (use "git add <file>..." to include in what will be committed)
        mi-archivo.txt

Your branch is up to date with 'origin/main'.
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
On branch main

Untracked files:
    (use "git add <file>..." to include in what will be committed)
        mi-archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm
1 file changed, 3 insertions(+)
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 376 bytes | 376.00 KiB/s, done.
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise>
git chec
h
```

```
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote: https://github.com/FeddericoGarcia/utn-conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/FeddericoGarcia/utn-conflict-exercise.git
* [new branch] feature-branch -> feature-branch
branch 'feature-branch' set up to track 'origin/feature-branch'.
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git status

On branch feature-branch
Your branch is up to date with 'origin/feature-branch'.

nothing to commit, working tree clean
```

3) Actividad de conflict-exercise

https://github.com/FeddericoGarcia/utn-conflict-exercise

```
:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos>cd conflict-exercise
C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise>git clone https://github.com/FeddericoGarcia/utn-conflict-exercise.git
Cloning into 'utn-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.
 C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise>_
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> ^C
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git conf
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git chec
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm
     Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)
     no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm
  1 file changed, 4 insertions(+), 1 deletion(-)

• PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git chec
     Switched to branch 'main'
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add

    PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm

     Your branch is up to date with 'origin/main'.
     nothing to commit, working tree clean
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git merg
Updating 942d2e1..719b7ad
      Updating 942d2e1..719b7ad
       README.md | 5 ++++
       1 file changed, 4 insertions(+), 1 deletion(-)
     On branch main
      On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
     nothing to commit, working tree clean
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
     Vour branch main

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

PS :\User\Garberio\(Garberio\)\Garberio\(Garberio\)\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise\ git stat

(use "git push" to publish your local commits)
        (use "git add <file>..." to update what will be committed)
        (use "git restore <file>...'
modified: README.md
                                                  to discard changes in working directory)
     no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git bran
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git chec
     PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
On branch feature-branch
```

```
On branch feature-branch
 Changes not staged for commit:
 Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git bran
 PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add
 On branch feature-branch
 Changes to be committed:

(use "git restore --staged <file>..." to unstage)
 PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm
 [feature-branch ce06a1c] Resolved merge conflict
1 file changed, 2 insertions(+), 3 deletions(-)
 On branch feature-branch
nothing to commit, working tree clean
 PS C:\Users\Garberio\OnePrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git chec
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.

PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
 On branch main
 Your branch is ahead of 'origin/main' by 1 commit.
 nothing to commit, working tree clean
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
 On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
  S C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git pull
Already up to date
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean feature-branch
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
On branch main
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git add PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-l\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git comm
 1 file changed, 1 insertion(+), 1 deletion(-)
  feature-branch
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git merg
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git bran
   feature-branch
Switched to branch 'feature-branch'
PS C:\Users\Garberio\OneDrive\Escritorio\Files\TUP-UTN\Programacion-1\TrabajosPracticos\conflict-exercise\utn-conflict-exercise> git stat
On branch feature-branch
```

8