

**TRABAJO PRÁCTICO NRO 2**  
**PROGRAMACIÓN 1**  
**AZCUY NICOLÁS - DNI: 33.368.267**

DIRECCIÓN DEL REPOSITORIO:

<https://github.com/nazcuy/UTN-TUPaD-P1/tree/main/02%20Trabajo%20Colaborativ>  
[o](#)

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

- **¿Qué es GitHub?**

GitHub es una plataforma en línea para alojar repositorios Git. Permite alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código. Facilita la colaboración en proyectos, el control de versiones y cambios, seguimiento de errores, etc.

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

- Iniciar sesión en GitHub.
- Hacer clic en + (en la esquina superior derecha), y luego en New repository.
- Completar los campos: Repository name: Nombre del repositorio; Description: Descripción opcional; Definir la visibilidad: si es Public o Private; Definir si se agrega un txt de presentación opcional: Marca Add a README file.
- Hacer clic en Create repository.

- **¿Cómo crear una rama en Git?**

`git branch nombre-de-la-rama`

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

`git checkout nombre-de-la-rama`

- **¿Cómo fusionar ramas en Git?**

`git merge nombre-de-la-rama`

- **¿Cómo crear un commit en Git?**

Guarda todos los archivos modificados:

git add .

Crea el comit:

git commit -m "Descripción del cambio"

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

git push origin nombre-de-la-rama

- **¿Qué es un repositorio remoto?**

Es una copia de un repositorio Git alojado en un servidor externo (ej: GitHub). Permite la colaboración y la sincronización de cambios entre equipos de trabajadores.

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

git remote add origin https://github.com/usuario/repositorio.git

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

git push -u origin nombre-de-la-rama

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

git pull origin nombre-de-la-rama

- **¿Qué es un fork de repositorio?**

Una copia independiente de un repositorio en la cuenta propia de GitHub. Se usa para proponer cambios a proyectos ajenos sin que se modifique el original.

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

- Trasladarse hasta el repositorio en GitHub que se quiere realizar el fork.
- Hacer clic en Fork (esquina superior derecha).
- Elegir la cuenta propia como destino.

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

- Realizar los cambios en el fork propio.
- Ir a la página del repositorio original en GitHub.
- Hacer clic en New pull request.
- Seleccionar las ramas.
- Describir los cambios y enviar la solicitud.

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

- En GitHub, trasladarse hasta el pull request.
- Revisar los cambios en la pestaña Files changed.
- Hacer clic en Merge pull request.

- **¿Qué es una etiqueta en Git?**

Es una referencia estática a un commit específico en el historial del repositorio. Se usa para marcar momentos importantes del desarrollo como puede ser una versión nueva del software. A diferencia de las ramas, las etiquetas no cambian, registran estados históricos inmutables.

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

`git tag -a nombre-de-la-etiqueta -m "mensaje-adjunto"`

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

`git push origin nombre-de-la-etiqueta.`

- **¿Qué es un historial de Git?**

Registro cronológico de todos los commits en un repositorio, incluyendo autor, fecha y cambios.

- **¿Cómo ver el historial de Git?**

Muestra historial del repositorio:

`git reflog`

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

Muestra los commit línea por línea:

`git log --oneline`

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

`git rm archivo-o-carpeta`

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

Es un repositorio visible solo para colaboradores explícitamente invitados. Requiere suscripción pagada para equipos (gratuito para cuentas personales).

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

- Mismos pasos para crear un repositorio.
- Seleccionar Private en la sección de visibilidad.

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

- Ir al repositorio en GitHub.
- Settings > Collaborators > Add people.
- Ingresa el nombre de usuario o email del colaborador.

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

Repositorio visible para cualquier usuario de internet. No requiere invitación para ver o hacer fork.

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

- Mismos pasos para crear un repositorio.
- Seleccionar Public en la sección de visibilidad.

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

Se comparte el enlace del repositorio (ej: <https://github.com/usuario/repositorio>).

---

## **2) Realizar la siguiente actividad:**

- **Crear un repositorio: Dale un nombre al repositorio / Elige el repositorio sea público / Inicializa el repositorio con un archivo.**

- **Agregando un Archivo: Crea un archivo simple, por ejemplo, mi-archivo.txt" / Realiza los comandos `git add .` y `git commit -m "Agregando mi-archivo.txt"` en la línea de comandos / Sube los cambios al repositorio en GitHub con `git push origin main` (o el nombre de la rama correspondiente).**

- **Creando Branchs: Crear una Branch / Realizar cambios o agregar un archivo / Subir la Branch.**

Dirección del repositorio: <https://github.com/nazcuy/repositorio-prog1.git>

MINGW64/c/Users/Tania/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1

```
Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git init
Initialized empty Git repository in C:/Users/Tania/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1/.git/

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git add .

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git commit -m "Iniciando el repositorio con un archivo"
[master (root-commit) 773f2d5] Iniciando el repositorio con un archivo
1 file changed, 86 insertions(+)
create mode 100644 index.html

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git remote add origin https://github.com/nazcuy/repositorio-prog1.git

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.12 KiB | 1.12 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/nazcuy/repositorio-prog1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git branch
* master
```

```
Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git branch trabajo-practico2

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git branch
* master
  trabajo-practico2

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (master)
$ git checkout trabajo-practico2
Switched to branch 'trabajo-practico2'

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/repositorio-prog1 (trabajo-practico2)
$ git branch
  master
* trabajo-practico2
```

```
Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/
repositorio-prog1 (trabajo-practico2)
$ git add .

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/
repositorio-prog1 (trabajo-practico2)
$ git status
On branch trabajo-practico2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   comandos-tp-2.txt

Tania@LAPTOP-LEQKVA26 MINGW64 ~/Desktop/NICO/Tecnicatura en Programación/Programación 1/U2 - Git/
repositorio-prog1 (trabajo-practico2)
$ git commit -m "Se agrega archivo .txt nuevo con los comandos hechos hasta el momento"
[trabajo-practico2 b569fb0] Se agrega archivo .txt nuevo con los comandos hechos hasta el momento
 1 file changed, 54 insertions(+)
 create mode 100644 comandos-tp-2.txt
```

### 3) Realizar la siguiente actividad:

**Paso 1: Crear un repositorio en GitHub.**

**Paso 2: Clonar el repositorio a tu máquina local.**

**Paso 3: Crear una nueva rama y editar un archivo.**

**Paso 4: Volver a la rama principal y editar el mismo archivo.**

**Paso 5: Hacer un merge y generar un conflicto.**

**Paso 6: Resolver el conflicto.**

**Paso 7: Subir los cambios a GitHub.**

**Paso 8: Verificar en GitHub.**

Dirección del repositorio en GitHub: <https://github.com/nazcuy/conflict-exercise.git>

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git>git clone https://github.com/nazcuy/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.

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git>cd conflict-exercise

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git checkout -b feature-branch
Switched to a new branch 'feature-branch'

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git add README.md

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git commit -m "Added a line in feature-branch"
[feature-branch 6bd2772] Added a line in feature-branch
 1 file changed, 3 insertions(+)

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git add README.md

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git commit -m "Added a line in main branch"
[main bb26e46] Added a line in main branch
 1 file changed, 2 insertions(+)

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\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.
```

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - i

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herrar

README.md

```
1 # conflict-exercise
2 Ejercicio del tp2
3
4 <<<<<< HEAD
5 Este es un cambio en la main branch.
6 =====
7
8 Este es un cambio en la feature branch.
9 >>>>>> feature-branch
10
```

C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\co



Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramienta


README.md



```
1 # conflict-exercise
2 Ejercicio del tp2
3
4 Esta es la corrección del archivo README.md
5
```




```
C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git add README.md
C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>git commit -m "Resolved merge conflict"
[main fc4ee6a] Resolved merge conflict
C:\Users\Tania\Desktop\NICO\Tecnicatura en Programación\Programación 1\U2 - Git\conflict-exercise>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), 908 bytes | 454.00 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/nazcuy/conflict-exercise.git
  339c770..fc4ee6a  main -> main
```



 main  1 Branch  0 Tags

 Go to file  t

Add file 

 Code 

 <b>nazcuy</b> Resolved merge conflict	fc4ee6a · 1 minute ago	 4 Commits
 README.md	Resolved merge conflict	1 minute ago

 README 

---

## conflict-exercise

Ejercicio del tp2

Esta es la corrección del archivo README.md 