

Universidad Tecnológica Nacional



Programación 1

Alumno: Mateo Joaquín Serafini
Dni: 45089673

Trabajo Práctico N.º 2: Git y GitHub

Objetivo

El estudiante desarrollará competencias para trabajar con Git y GitHub, aplicando conceptos fundamentales de control de versiones, colaboración en proyectos y resolución de conflictos, en un entorno simulado y guiado.

Resultados de Aprendizaje

1. **Comprender los conceptos básicos de Git y GitHub:** Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.
 2. **Manejar comandos esenciales de Git:** Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.
 3. **Aplicar técnicas de colaboración en GitHub:** Configurar y utilizar repositorios remotos, realizar forks y gestionar pull requests para facilitar el trabajo colaborativo.
 4. **Resolver conflictos en un entorno de control de versiones:** Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.
-

Actividades

1) Cuestionario de Conceptos

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

- ¿Qué es GitHub?
- ¿Cómo crear un repositorio en GitHub?
- ¿Cómo crear una rama en Git?
- ¿Cómo cambiar a una rama en Git?
- ¿Cómo fusionar ramas en Git?
- ¿Cómo crear un commit en Git?
- ¿Cómo enviar un commit a GitHub?
- ¿Qué es un repositorio remoto?
- ¿Cómo agregar un repositorio remoto a Git?
- ¿Cómo empujar cambios a un repositorio remoto?
- ¿Cómo tirar de cambios de un repositorio remoto?
- ¿Qué es un fork de repositorio?
- ¿Cómo crear un fork de un repositorio?
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
- ¿Cómo aceptar una solicitud de extracción?
- ¿Qué es una etiqueta en Git?
- ¿Cómo crear una etiqueta en Git?

- ¿Cómo enviar una etiqueta a GitHub?
 - ¿Qué es un historial de Git?
 - ¿Cómo ver el historial de Git?
 - ¿Cómo buscar en el historial de Git?
 - ¿Cómo borrar el historial de Git?
 - ¿Qué es un repositorio privado en GitHub?
 - ¿Cómo crear un repositorio privado en GitHub?
 - ¿Cómo invitar a alguien a un repositorio privado en GitHub?
 - ¿Qué es un repositorio público en GitHub?
 - ¿Cómo crear un repositorio público en GitHub?
 - ¿Cómo compartir un repositorio público en GitHub?
-

2) Creación y Manipulación de un Repositorio

- **Crear un repositorio**
 - Asignar un nombre al repositorio.
 - Seleccionar que sea público.
 - Inicializar el repositorio con un archivo.
- **Agregando un archivo**
 - Crear un archivo simple, por ejemplo, "mi-archivo.txt".

Ejecutar los comandos:

`git add .`

- `git commit -m "Agregando mi-archivo.txt"`
 - Subir los cambios al repositorio en GitHub:
`git push origin main`
 - **Creando Branchs**
 - Crear una branch.
 - Realizar cambios o agregar un archivo.
 - Subir la branch.
-

3) Resolución de Conflictos en Git

Paso 1: Crear un repositorio en GitHub

- Iniciar sesión en GitHub y crear un nuevo repositorio.
- Asignar un nombre al repositorio, por ejemplo, `conflict-exercise`.
- Inicializar con un archivo README.

Paso 2: Clonar el repositorio a la máquina local

- Copiar la URL del repositorio.

Ejecutar en la terminal:

git clone https://github.com/tuusuario/conflict-exercise.git

- cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

- Crear una nueva rama:
git checkout -b feature-branch

Editar **README.md**, agregar una nueva línea y hacer commit:

git add README.md

- git commit -m "Added a line in feature-branch"

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

Cambiar a la rama **main** y hacer otro cambio en **README.md**:

git checkout main

git add README.md

- git commit -m "Added a line in main branch"

Paso 5: Hacer un merge y generar un conflicto

- Intentar hacer un merge:
git merge feature-branch
- Se generará un conflicto.

Paso 6: Resolver el conflicto

- Editar **README.md**, decidir qué cambios mantener y guardar.

Completar el merge:

git add README.md

- git commit -m "Resolved merge conflict"

Paso 7: Subir los cambios a GitHub

Subir los cambios:

git push origin main

- git push origin feature-branch

Paso 8: Verificar en GitHub

- Revisar el archivo **README.md** y el historial de commits.

1) Cuestionario de Conceptos

¿Qué es GitHub?:

GitHub es una plataforma de alojamiento/nube de guardado de repositorios Git, que permiten la gestión de proyectos, colaboración entre desarrolladores y control de versiones.

¿Cómo crear un repositorio en GitHub?

Los pasos serían los suponiendo que ya tienes la cuenta creada.

- Iniciar sesión en GitHub.
- Hacer clic en "New Repository".
- Introducir un nombre y descripción.
- Elegir visibilidad (público o privado).
- Hacer clic en "Create repository".

¿Cómo crear una rama en Git?

En la consola de git se utilizaría el siguiente comando (`git branch nombre-de-la-rama`).

¿Cómo cambiar a una rama en Git?

En la consola de git se utilizaría el siguiente comando (`git checkout nombre-de-la-rama`) o en otras versiones también es utilizable el siguiente comando (`git switch nombre-de-la-rama`).

¿Cómo fusionar ramas en Git?

En la consola de git se utilizaría el siguiente comando (`git merge nombre-de-la-rama`).

¿Cómo crear un commit en Git?

En la consola de git se utilizaría el siguiente comando (`git add .`) para agregar todos los archivos, y (`git commit -m "Mensaje del commit"`) para crear el commit con un mensaje que dice algo de información acerca de lo subido.

¿Cómo enviar un commit a GitHub?

En la consola de git se utilizaría el siguiente comando (`git push nombre-del-proyecto`).

¿Qué es un repositorio remoto?

Un repositorio remoto es un repositorio almacenado remotamente (ej. un servidor), que permite la colaboración y sincronización entre desarrolladores.

¿Cómo agregar un repositorio remoto a Git?

En la consola de git se utilizaría el siguiente comando (`git remote add origin URL_DEL_REPOSITORIO`).

¿Qué es un fork de repositorio?

Una forma de copiar un repositorio en tu cuenta de GitHub, que nos permite hacer cambios sin afectar el original.

¿Cómo crear un fork de un repositorio?

Ir al repositorio en GitHub que queremos copiar.

Hacer clic en "Fork".

Elegir tu cuenta como destino.

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

Subir cambios a una rama.

Ir al repositorio en GitHub.

Hacer clic en "New pull request".

Comparar cambios y enviar la solicitud.

¿Cómo aceptar una solicitud de extracción?

Ir a la pestaña "Pull requests".

Seleccionar la solicitud.

Revisar los cambios y hacer clic en "Merge pull request".

¿Qué es una etiqueta en Git?

Un marca en el historial del repositorio, usado para identificar versiones específicas y/o importantes.

¿Cómo crear una etiqueta en Git?

```
git tag -a e1.0 -m "Etiqueta 1.0"
```

¿Cómo enviar una etiqueta a GitHub?

```
git push origin e1.0
```

¿Qué es un historial de Git?

El registro de todos los cambios y modificaciones realizados en un repositorio.

¿Cómo ver el historial de Git?

En la consola de git se utilizaria el sig comando (git log).

¿Cómo buscar en el historial de Git?

En la consola de git se utilizaria el sig comando (git log --grep="palabra clave").

¿Cómo borrar el historial de Git?

No se puede borrar completamente. Se puede hacer algo similar que es resetearlo con el siguiente comando (resetear git reset --hard HEAD~n)

¿Qué es un repositorio privado en GitHub?

Un repositorio accesible sólo por usuarios autorizados.

¿Cómo crear un repositorio privado en GitHub?

Crear un nuevo repositorio.

Seleccionar "Private".

Crear el repositorio.

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

Ir a "Settings" del repositorio.

Ir a "Manage access" y agregar colaboradores.

¿Qué es un repositorio público en GitHub?

Un repositorio accesible por cualquier usuario de GitHub.

¿Cómo crear un repositorio público en GitHub?

Crear un nuevo repositorio.

Seleccionar "Public".

¿Cómo compartir un repositorio público en GitHub?

Copiar la URL del repositorio y compartirla.

2) Creación y Manipulación de un Repositorio

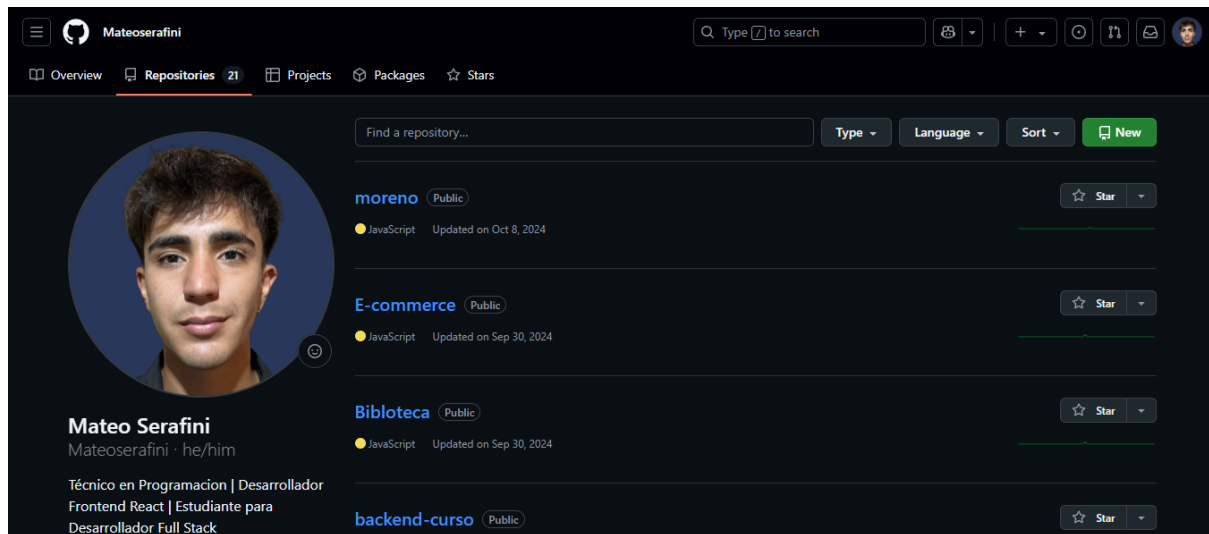
- **Crear un repositorio**
 - Asignar un nombre al repositorio.
 - Seleccionar que sea público.
 - Inicializar el repositorio con un archivo.
- **Agregando un archivo**
 - Crear un archivo simple, por ejemplo, "mi-archivo.txt".

Ejecutar los comandos:

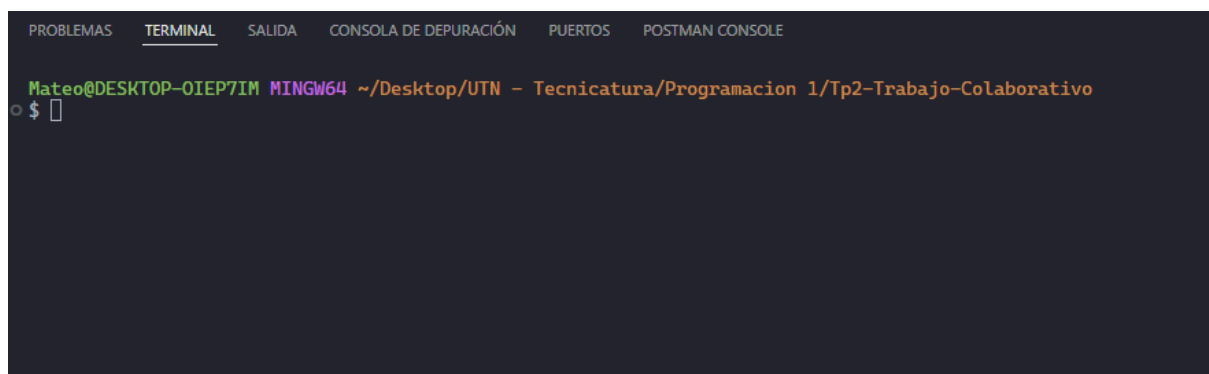
git add .

- git commit -m "Agregando mi-archivo.txt"
 - Subir los cambios al repositorio en GitHub:
git push origin main
- **Creando Branchs**
 - Crear una branch.
 - Realizar cambios o agregar un archivo.
 - Subir la branch.

Captura del repositorio antes de comenzar.



Consola Git Bash en la carpeta específica del ejercicio.



Inicializando el repositorio.

```
Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo
$ git init
Initialized empty Git repository in C:/Users/Mateo/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo/.git/
```

Subiendo el repositorio local al repositorio remoto.

```
Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (master)
$ git add .

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (master)
$ git commit -m "Primer Commit"
[master (root-commit) db66f58] Primer Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Readme.md
```



```
Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (master)
$ git branch -M main

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (main)
$ git remote add origin https://github.com/Mateoserafini/Tp2-Trabajo-Colaborativo.git

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 213 bytes | 213.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Mateoserafini/Tp2-Trabajo-Colaborativo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Creando la rama y subiendo al repositorio remoto.

```
Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (main)
$ git branch primera-rama

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (main)
$ git checkout primera-rama
Switched to branch 'primera-rama'

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (primera-rama)
$ git add .

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (primera-rama)
$ git commit -m "desde la rama"
[primera-rama 2795ed0] desde la rama
1 file changed, 1 insertion(+)

Mateo@DESKTOP-OIEP7IM MINGW64 ~/Desktop/UTN - Tecnicatura/Programacion 1/Tp2-Trabajo-Colaborativo (primera-rama)
$ git push -u origin primera-rama
Enumerating objects: 5, done.
Counting objects: 100% (5/5), 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)
remote:
remote: Create a pull request for 'primera-rama' on GitHub by visiting:
remote:      https://github.com/Mateoserafini/Tp2-Trabajo-Colaborativo/pull/new/primera-rama
remote:
To https://github.com/Mateoserafini/Tp2-Trabajo-Colaborativo.git
 * [new branch]      primera-rama -> primera-rama
branch 'primera-rama' set up to track 'origin/primera-rama'.
```

3) Resolución de Conflictos en Git

Paso 1: Crear un repositorio en GitHub

- Iniciar sesión en GitHub y crear un nuevo repositorio.
- Asignar un nombre al repositorio, por ejemplo, conflict-exercise.
- Inicializar con un archivo README.

Paso 2: Clonar el repositorio a la máquina local

- Copiar la URL del repositorio.

Ejecutar en la terminal:

```
git clone https://github.com/tuusuario/conflict-exercise.git
```

- `cd conflict-exercise`

Paso 3: Crear una nueva rama y editar un archivo

- Crear una nueva rama:
`git checkout -b feature-branch`

Editar `README.md`, agregar una nueva línea y hacer commit:
`git add README.md`

- `git commit -m "Added a line in feature-branch"`

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

Cambiar a la rama `main` y hacer otro cambio en `README.md`:
`git checkout main`
`git add README.md`

- `git commit -m "Added a line in main branch"`

Paso 5: Hacer un merge y generar un conflicto

- Intentar hacer un merge:
`git merge feature-branch`
- Se generará un conflicto.

Paso 6: Resolver el conflicto

- Editar `README.md`, decidir qué cambios mantener y guardar.

Completar el merge:
`git add README.md`

- `git commit -m "Resolved merge conflict"`

Paso 7: Subir los cambios a GitHub

Subir los cambios:
`git push origin main`

- `git push origin feature-branch`

Paso 8: Verificar en GitHub

- Revisar el archivo `README.md` y el historial de commits.

Creando el repositorio.

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 * Mateoserafini / Repository name * conflict-exercise

✔ conflict-exercise is available.

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

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.](#)

Clonando repositorio

```
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1> git clone https://github.com/Mateoserafini/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\Mateo\Desktop\UTN - Tecnicatura\Programacion 1>
```

Creando rama

```
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1> cd conflict-exercise
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise> git checkout -b feature-branch
Switched to a new branch 'feature-branch'
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise>
```

```
1 # conflict-exercise
2 | Nueva linea desde la rama en el README
```

Modificando README en la línea principal

```
1 # conflict-exercise
2 Nueva linea desde el main en el readme
```

```
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise> git add README.md
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise> git commit -m "Added a line in main branch"
[main dbaa033] Added a line in main branch
1 file changed, 2 insertions(+), 1 deletion(-)
```

Generando conflicto

```
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\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.
```

Resolviendo conflicto

```
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise> git add README.md
>>
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\conflict-exercise> git commit -m "Resolved merge conflict"
>>
[main 12cf6c3] Resolved merge conflict
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\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), 806 bytes | 806.00 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/Mateoserafini/conflict-exercise.git
9c7d59b..12cf6c3 main -> main
PS C:\Users\Mateo\Desktop\UTN - Tecnicatura\Programacion 1\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/Mateoserafini/conflict-exercise/pull/new/feature-branch
remote:
To https://github.com/Mateoserafini/conflict-exercise.git
* [new branch] feature-branch -> feature-branch
```

```
1 # conflict-exercise
2 Nueva linea desde el main en el readme
3 Nueva linea desde la rama en el README
4
```