

# Trabajo Práctico 2: Git y GitHub

---

Tecnicatura Universitaria en Programación - Programación I

Alumno: Jose Ricardo Miranda

Fecha: 10/05/2025

## Parte 1 - Preguntas teóricas

### ¿Qué es GitHub?

GitHub es una plataforma en la nube que permite almacenar proyectos que usan Git como sistema de control de versiones. Facilita el trabajo colaborativo al permitir subir código, revisarlo, discutir cambios mediante 'pull requests' y trabajar con ramas de forma organizada.

### ¿Cómo crear un repositorio en GitHub?

Iniciás sesión en GitHub, hacés clic en "New repository", completás nombre y opciones (público/privado, README), y hacés clic en 'Create repository'.

### ¿Cómo crear una rama en Git?

Usás el comando `git branch nombre_de_la_rama` o `git checkout -b nombre_de_la_rama` para crearla y cambiarte a ella.

### ¿Cómo cambiar a una rama en Git?

Con `git checkout nombre_de_la_rama`.

### ¿Cómo fusionar ramas en Git?

Desde la rama de destino (ej. main), usás `git merge nombre_de_la_otra_rama`.

### ¿Cómo crear un commit en Git?

Primero se hace `git add archivo.txt` y luego `git commit -m "mensaje"`.

### ¿Cómo enviar un commit a GitHub?

Después del commit local, se usa `git push origin nombre_rama`.

### ¿Qué es un repositorio remoto?

Es una copia del repositorio local alojada en un servidor como GitHub, que permite compartir y colaborar.

### ¿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 origin nombre_rama`

### ¿Cómo tirar de cambios de un repositorio remoto?

`git pull origin nombre_rama`

### ¿Qué es un fork de repositorio?

Una copia completa de un repositorio GitHub que se guarda en tu cuenta. Te permite experimentar o contribuir sin afectar el original.

### ¿Cómo crear un fork de un repositorio?

Ingresas al repo original, hacés clic en 'Fork' y seleccionás tu cuenta como destino.

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

Desde tu fork o rama, hacés clic en 'Compare & pull request', escribís un mensaje y lo enviás.

### ¿Cómo aceptar una solicitud de extracción?

Solo el dueño del repo puede aceptarla desde la pestaña 'Pull requests', revisando y clic en 'Merge pull request'.

### ¿Qué es un etiqueta en Git?

Una etiqueta es un marcador que señala un punto importante del historial, como una versión o release.

### ¿Cómo crear una etiqueta en Git?

`git tag v1.0` o `git tag -a v1.0 -m "mensaje".`

### ¿Cómo enviar una etiqueta a GitHub?

`git push origin v1.0` o `git push origin --tags`

### ¿Qué es un historial de Git?

El registro completo de todos los commits hechos en el proyecto.

### ¿Cómo ver el historial de Git?

`git log` o `git log --oneline`

### ¿Cómo buscar en el historial de Git?

`git log --grep="palabra"`

### ¿Cómo borrar el historial de Git?

No se recomienda, pero se puede hacer con `git rebase` o reiniciando el repo.

### ¿Qué es un repositorio privado en GitHub?

Un repo que no es visible públicamente, sólo accesible a usuarios invitados.

### ¿Cómo crear un repositorio privado en GitHub?

Durante la creación, elegís la opción 'Private'.

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

En 'Settings' > 'Manage access', clic en 'Invite a collaborator'.

### ¿Qué es un repositorio público en GitHub?

Un repositorio accesible públicamente que cualquiera puede ver.

### ¿Cómo crear un repositorio público en GitHub?

Durante la creación, seleccionás 'Public'.

### ¿Cómo compartir un repositorio público en GitHub?

Simplemente copiás y compartís la URL del repo.

## Parte 2 y 3 - Actividades prácticas

A continuación se describen los pasos realizados en las actividades prácticas del trabajo práctico.

- 1. Crear el repositorio en GitHub llamado 'tp2-git', configurado como público e inicializado con un README.
- 2. Clonar el repositorio localmente con: `git clone URL_DEL_REPOSITORIO`
- 3. Crear un archivo llamado `mi-archivo.txt` con contenido simple.
- 4. Agregar el archivo con: `git add`.
- 5. Confirmar los cambios con: `git commit -m 'Agregando mi-archivo.txt'`
- 6. Subir los cambios con: `git push origin main`
- 7. Crear una nueva rama con: `git checkout -b nueva-rama`
- 8. Modificar o agregar un archivo, hacer commit y subirlo con `git push origin nueva-rama`
- 9. Crear nuevo repositorio llamado 'conflict-exercise' con README inicial.
- 10. Clonarlo localmente, crear la rama `feature-branch`, modificar `README.md` y hacer commit.
- 11. Volver a la rama `main`, modificar la misma línea del `README.md` y hacer commit.
- 12. Hacer merge con `git merge feature-branch` en `main`, y resolver manualmente el conflicto en `README.md`.
- 13. Confirmar resolución del conflicto con `git add` y `git commit`.
- 14. Subir los cambios al repositorio remoto con `git push origin main`.

<https://github.com/conibre/tp2-git.git>

conibre / conflict-exercise

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Commits

main

All usersAll time

Commits on May 10, 2025

Resuelvo conflicto despues del pull	6686222		
conibre committed 3 hours ago			
Resuelve el conflicto entre main y feature-branch	9e3a053		
conibre committed 3 hours ago			
agrega linea en main	43594f5		
conibre committed 3 hours ago			
Agrega linea en feature-branch	c8beeb7		
conibre committed 3 hours ago			
agrega linea en main	e87ee04		
conibre committed 3 hours ago			
Agrega linea en main	b711441		
conibre committed 3 hours ago			
Agrega linea en feature-branch	5e94c7f		
conibre committed 3 hours ago			
Update README.md	871768c	Verified	
conibre authored 3 hours ago			
Update README.md	9ff8c68	Verified	
conibre authored 3 hours ago			
Initial commit	e18197a	Verified	
conibre authored 3 hours ago			