

# PROGRAMACION I

## Trabajo Práctico N.º 2: GIT y GitHub

Mairone Juan Ignacio

**Actividad 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 proyectos con **control de versiones** usando Git. Permite:

- **Colaboración:** Varios desarrolladores trabajan en el mismo proyecto.
- **Gestionar código:** *Forks, pull requests.*
- **Despliegue continuo:** Integración con herramientas como GitHub Actions.

#### **Ejemplo:**

Un equipo usa GitHub para desarrollar una app web. Cada miembro clona el repositorio, crea ramas y envía cambios mediante *pull requests*.

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

Para crear un repositorio se tienen que realizar los siguientes pasos:

1. Inicia sesión en GitHub.
2. Haz clic en "**New**" (botón verde).
3. Completa:
  - o **Nombre:** mi-proyecto.
  - o **Visibilidad:** Público. Esto significa que va a ser visible para todos, por ende, es ideal para proyectos open-source (que significa de código abierto).

También tenemos la visibilidad Privada, la cual hace referencia a que solo va a ser accesible para colaboradores invitados.

Para invitar a un colaborador a un repositorio privado, se realizan los siguientes pasos:

1. En GitHub, dirigirse a **Settings > Collaborators**.
  2. Ingresa el nombre de usuario o email del colaborador.
  3. Haz clic en **"Add collaborator"**.
- o **Opcional:** Marca **"Add a README file"**.
4. Clic en **"Create repository"**.

### • ¿Cómo crear una rama en Git?

Para crear una rama en Git se utiliza el siguiente comando: **git branch <nombre>**

Un ejemplo de esto sería de la siguiente manera: **git branch feature-login**

### • ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en Git se utiliza el siguiente comando: **git checkout <nombre>**

Un ejemplo de esto sería de la siguiente manera: **git checkout main**

### • ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git se realizan las siguientes acciones:

- 1) Cambiar o moverse a la rama destino, que sería la rama donde nosotros queremos que quede todo fusionado. Para realizar esto se utiliza el siguiente comando de Git: **git checkout main**
- 2) Una vez parados en dicha rama, fusionamos, utilizando el siguiente comando de Git: **git merge feature-login**
- 3) En algunos casos al fusionar ramas pueden aparecer conflictos entre dichas ramas. Si ocurre esto, hay que resolver dichos conflictos, haciendo un seguimiento detallista de línea por línea de los posibles conflictos que se hayan generado, eligiendo que líneas dejar y cuáles no. Esto es una muy buena práctica para ver si realmente entendemos el código que nosotros mismos hemos realizado, o también, el código que algún compañero haya realizado.

### • ¿Cómo crear un commit en Git?

Para crear un nuevo commit en Git se realizan las siguientes acciones:

- 1) Introducimos el siguiente comando de Git: **git add** . Lo que realiza este comando es que añade todos los cambios. (el comando es con él **"."** al final, no se tiene que olvidar.).

- 2) Luego de introducir el comando anterior, introducimos el siguiente comando: **git commit -m "Mensaje descriptivo de los cambios que se realizaron"**

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

Para enviar un commit a GitHub tenemos que hacer uso del siguiente comando: **git push origin main**. Lo que realiza este comando es subir todos los cambios que hayamos efectuado en los commits anteriores a la rama main.

- **¿Qué es un repositorio remoto?**

Un repositorio remoto es una copia del repositorio en un servidor (por ejemplo, GitHub).

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

Para agregar un repositorio remoto a Git utilizamos el siguiente comando:

**git remote add origin https://github.com/tuusuario/repo.git**

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

Para empujar cambios a un repositorio remoto hacemos uso del siguiente comando:

**git push -u origin main**

Se aclara que “-u” sirve para guardar la rama por defecto, por ende, se puede utilizar como no.

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

Para tirar de cambios de un repositorio remoto se utiliza el siguiente comando:

**git pull origin main**. Al efectuar este comando se actualiza tu repo local.

- **¿Qué es un fork de repositorio?**

Un fork sirve para copiar un repositorio ajeno a tu cuenta. En GitHub lo encontramos con forma de un botón que dice “**Fork**”.

Realizar un fork permite lo siguiente:

- Modificar el código sin afectar el original.
- Contribuir al proyecto original mediante pull request.

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

Para crear un fork de un repositorio se tienen que realizar los siguientes pasos:

- 1) Ir al repositorio de GitHub al cual le quieras realizar fork.
- 2) Haz clic en fork. El mismo es un botón situado en la esquina superior derecha.
- 3) Elegí TU CUENTA como destino.

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

Para enviar una solicitud de extracción (pull request) se tienen que realizar los siguientes pasos:

- 1) Hacer cambios en tu fork y subirlos. Un ejemplo de esto utilizando comandos seria de la siguiente manera:

```
git checkout -b mi-fix
```

```
"Nuevo cambio" >> README.md
```

```
git add . && git commit -m "Fix: error en documentación"
```

```
git push origin mi-fix
```

- 2) En GitHub, movete a tu fork > Pull Request > New Pull Request.
- 3) Compara las ramas. Por ejemplo: **main** del origin vs **mi-fix** de tu fork
- 4) Hacer clic en Create new Pull Request.

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

Para aceptar una solicitud de extracción (suponiendo o teniendo en cuenta que sos el que mantiene el repo o sos el admin del repo o que mínimamente tenes los permisos o autorizaciones pertinentes para esto) se deben realizar los siguientes pasos:

- 1) Te dirigís a pull request.
- 2) Revisas los cambios y haces clic en Merge Pull Request.
- 3) Confirmas con Confirm Merge.

Un ejemplo de esto podría ser que aceptas un PR que corrige un error en el **README.md**

- **¿Qué es una etiqueta en Git?**

Una etiqueta o Tags en Git sirve para marcar un punto específico o importante en el historial.

Por ejemplo: versionado/versiones **v1.0.0**

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

Para crear una etiqueta en Git se hace uso del siguiente comando:

```
- git tag -a v1.0.0 -m "Versión estable 1.0.0"
```

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

Teniendo en cuenta el ejemplo del comando anterior (git tag -a v1.0.0 -m "Versión estable 1.0.0")

Para enviar dicha etiqueta a GitHub se utiliza el siguiente comando: git push origin v1.0.0

- **¿Qué es un historial de Git?**

El historial de Git es el registro de todos los commits, ramas y cambios en el proyecto.

- **¿Cómo ver el historial de Git?**

Para ver el historial de Git se hace uso de los siguientes comandos:

`git log --oneline` Muestra commits resumidos.

`git log -p` Muestra cambios en cada commit.

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

Para buscar en el historial de Git se puede utilizar el siguiente comando:

`git log --grep="bug"` Busca commits con "bug" en el mensaje.

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

Un ejemplo de cómo borrar el historial de Git puede ser el siguiente:

**Crear un nuevo commit sin historial:**

`git checkout --orphan nueva-rama`

`git add . && git commit -m "Nuevo inicio"`

`git branch -D main` Borra la rama main

`git branch -m main` Renombra nueva-rama a main

`git push -f origin main` Fuerza el push (solo para repos personales)

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

Un repositorio privado en GitHub hace referencia a que solo va a ser accesible para colaboradores invitados.

Para invitar a un colaborador a un repositorio privado, se realizan los siguientes pasos:

1. En GitHub, dirigirse a **Settings > Collaborators**.
2. Ingresa el nombre de usuario o email del colaborador.
3. Haz clic en **"Add collaborator"**.

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

Para crear un repositorio privado en GitHub se deben realizar los siguientes pasos:

- Ir a [github.com](https://github.com) y acceder con tu usuario.
- Una vez hayas ingresado a GitHub hacer clic New repository (es un botón verde que está situado en la esquina superior derecha).
- Elegir **“Private”** en la opción de **visibilidad** y completa el resto de datos que solicita.

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

Para invitar a alguien a un repositorio privado en GitHub se deben realizar los siguientes pasos:

- Dirigirse a **Settings > Collaborators**.
- Ingresa el usuario o email del colaborador.
- Hacer clic en **“Add collaborator”**.

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

Un repositorio público en GitHub significa que va a ser visible para todos, por ende, es ideal para proyectos open-source (que significa de código abierto).

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

Para crear un repositorio público se tienen que realizar los siguientes pasos:

- Ir a [github.com](https://github.com) y acceder con tu usuario.
- Una vez hayas ingresado a GitHub hacer clic New repository (es un botón verde que está situado en la esquina superior derecha).
- Elegir **“Publico”** en la opción de **visibilidad** y completa el resto de datos que solicita.

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

Para compartir un repositorio público se tienen que realizar los siguientes pasos

- Compartir el link o URL de tu repositorio. Ejemplo: <https://github.com/tu-usuario/repo>.
- Dar clic en **“Share”** (share significa compartir. Este es un botón que está en la página del repositorio)

**Actividad 2:** Se realiza todo lo que solicita el ejercicio 2 del archivo pdf correspondiente.

La URL del repositorio donde está plasmado todo lo que solicita el PDF es la siguiente:

<https://github.com/JuanIgnacioMairone/MaironeJuanUTN>

**Actividad 3:** Se realiza todo lo que solicita el ejercicio 3 del archivo pdf correspondiente.

La URL del repositorio donde está plasmado todo lo que solicita el PDF es la siguiente:

<https://github.com/JuanMaironeUTN/JuanMaironeConflict-exercise>