

# Trabajo Práctico N.º 2: Git y Github

- **Ejercicio 1:**

1. GitHub es una plataforma basada en la web que permite almacenar y gestionar proyectos que utilizan sistemas de control de versiones. Git facilita la comunicación entre los desarrolladores, permitiendo subir modificar y compartir código y así como trabajar en equipo mediante ramas, issues
2. Inicia sesión en Github, luego hace click en el botón "+" (arriba a la derecha) y selecciona New repository. Completa el nombre del repositorio, una descripción general de lo que tratara el proyecto (opcional) y elegi si será publico o privado. Opcionalmente puedes agregarle un README y por ultimo haces click en "Create repository"
3. Para poder crear una rama en git se usa el comando:  
***git branch nombre\_rama***  
crea una rama pero no te cambia a ella automáticamente.
4. Para poder cambiarme de rama en git se usa el comando:  
***git checkout nombre\_rama***
5. Para poder fusionar 2 ramas en git realiza los siguientes pasos:
  - a. Cambiar a la rama que quieres mantener actualizada : ***git checkout main***
  - b. Fusiona los cambios desde otra rama que ya tengas creada: ***git merge nombre\_rama***.
6. Para crear un commit en Git, luego de hacer los cambios que quieres ingresar, hay que seguir una serie de pasos:
  - a. Agregar archivos al área de preparación (stage): ***git add .***
  - b. Crea un commit : ***git commit -m "Mi primer commit"***.
7. Para enviar un commit a Github primero hay que estar seguros de que hicimos un commit. Luego usar el comando : ***git push origin nombre\_rama***.

8. Un repositorio remoto es una versión del repo que esta en el servidor como github . Permite que varias personas trabajen en el mismo proyecto y desde diferentes lugares.

9. Para agregar un repositorio remoto a git usar el comando:  
***git remote add origin <https://github.com/usuario/repositorio.git>***

10. Para empujar cambios a un repositorio remoto usar el comando:

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

11. Para traer cambios a un repositorio remoto se usa el comando:

***git pull origin nombre-de-la-rama***

Este comando trae los últimos cambios del repositorio remoto y los combina con tu rama actual.

12. Un fork es una copia de un repositorio en tu cuenta GitHub. Te permite experimentar o modificar un proyecto sin afectar el original.

13. Para crear un fork en un repositorio sigue estos pasos:

- a. Ingresa al repositorio original en GitHub.
- b. Hace click en el botón "Fork".
- c. Selecciona la cuenta como destino para crear la copia.

14. Para enviar una solicitud de extracción (pull request) segui estos pasos:

- a. Subi los cambios a tu repositorio (fork)
- b. En GitHub, hace click en "**Compare & pull request**".
- c. Revisa los cambios, escribir un mensaje explicando la modificación y luego click en "**Create pull request**".

15. Para aceptar los cambios de una solicitud de extracción, el líder del proyecto debe ir a la pestaña de "Pull requests". Elegir la solicitud, revisar los cambios y si esta todo correcto hacer click en "**Merge pull request**" y luego "**Confirm Merge**".

16. Una *etiqueta* (tag) marca puntos específicos en el historial del repositorio, como versiones estables. Por ejemplo, `v1.0`.
17. Para crear una etiqueta se usa el comando : ***git tag nombre-de-la-etiqueta*** , como ejemplo se le podría poner: ***git tag v1.0***
18. Para enviar una etiqueta a GitHub se usa el comando : ***git push origin nombre-de-la-etiqueta***.
19. El historial de de Git es el registro completo de todos los cambios y versiones del repositorio, incluyendo los commits, ramas, merges, etc.
20. Para ver el historial de Git: ***git log*** o mas especifico ***git log --oneline***
21. Podes buscar por mensaje de commit con el comando: ***git log --grep="palabra clave"***
22. No se recomienda borrar el historial de Github, pero si es necesario se puede hacerlo con el comando : ***git rebase -i HEAD~n***  
  
O directamente eliminar el repositorio y crear uno nuevo. **¡Precaución!** Esta acción puede ser destructiva.
23. Un repositorio privado es un repositorio visible solo para vos y las personas que invites. Ideal para proyectos internos o personales.
24. Para crear un repositorio privado lo haces cuando creás un repositorio nuevo, selecciona la opción "**Private**" antes de confirmar.
25. Para agregar colaboradores a tu proyecto sigue estos pasos:  
  
A Entrá al repositorio.  
  
B Hací clic en "**Settings**" > "**Collaborators**".  
  
C Ingresá el nombre del usuario y hacé clic en "**Add collaborator**".

26. Un repositorio publico es un repositorio visible para cualquier persona en Internet. Se puede clonar, ver y colaborar libremente (según permisos).
27. Para crear un repositorio publico es lo mismo que en el privado, pero seleccionando **"Public"** en la configuración inicial.
28. Para compartir un repositorio solo copió la URL del repositorio desde la barra de direcciones o haciendo clic en **"Code"** > **"HTTPS"** y compártala.

-----//-----

- **Ejercicio 2:** Adjunto el link con el repositorio subido a GitHub y realizando la tarea que solicitaron.

<https://github.com/NicolasViruel/PracticoN-2.git>