

## Trabajo Practico N° 2: Git y GitHub

Nombre: Ruff Luca Genaro

DNI: 43028752

Comisión: 21

### Actividad 1:

- **¿Qué es GitHub?:** GitHub es una plataforma de desarrollo colaborativo para proyectos de software. Ofrece herramientas para gestionar el código, realizar seguimiento de cambios y colaborar con otros desarrolladores en tiempo real.
- **¿Cómo crear un repositorio en GitHub?:**
  1. **Crear una cuenta en GitHub:** Si no tienes una cuenta en GitHub, debes crear una antes de empezar.
  2. **Iniciar sesión:** Inicia sesión en tu cuenta de GitHub.
  3. **Acceder al panel de control:** Una vez que hayas iniciado sesión, haz clic en el botón "+" en la esquina superior derecha de la pantalla y selecciona "Nuevo repositorio"
  4. **Elegir el tipo de repositorio:** Selecciona el tipo de repositorio que deseas crear: público, privado o interno
  5. **Ingrese la información del repositorio:**
  6. **Nombre del repositorio:** Ingresa un nombre único y descriptivo para tu repositorio.
  7. **Descripción:** Ingresa una breve descripción del repositorio.
  8. **Inicializar este repositorio con un archivo README:** Selecciona esta opción si deseas crear un archivo.
  9. **Agregar una licencia:** Puedes agregar una licencia a tu repositorio seleccionando una de las opciones disponibles.
  10. **Crear repositorio:** Haz clic en el botón "Crear repositorio" para crear tu nuevo repositorio.  
**Configuración adicional:**
  11. Agregar un archivo README: Si no has creado un archivo durante el proceso de creación, puedes agregar uno ahora.
  12. Agregar archivos al repositorio: Puedes agregar archivos al repositorio utilizando la interfaz web de GitHub o mediante la línea de comandos con Git.
  13. Configurar colaboradores: Puedes agregar colaboradores al repositorio para que puedan contribuir al proyecto.
- **¿Cómo crear una rama en Git?:**
  1. Abrir la terminal: Abre la terminal en tu computadora y navega hasta el directorio donde se encuentra tu repositorio Git.
  2. Verificar el estado del repositorio: Ejecuta el comando "git status" para verificar que no haya cambios sin committear.

3. Crear la rama: Ejecuta el comando "git branch (nombre-rama)" para crear una nueva rama. Reemplaza (nombre-rama) con el nombre que desees darle a la rama.
- **¿Cómo cambiar de rama en Git?:** Ejecuta el comando "git checkout -b (nombre-rama)" para cambiar a la rama deseada y crearla si no existe.
  - **¿Cómo fusionar ramas en Git?:**
    1. Ejecuta el comando "git checkout (nombre-rama-principal)" para cambiar a la rama principal (por lo general, master).
    2. Ejecuta el comando "git merge (nombre-rama-secundaria)" para fusionar la rama secundaria con la rama principal.
  - **¿Cómo crear un commit en Git?:**
    1. Ejecuta el comando "git add ." para agregar los cambios realizados en el archivo al área de preparación.
    2. Ejecuta el comando "git commit -m (mensaje-del-commit)" para crear el commit.
  - **¿Cómo enviar un commit a GitHub?:** Después de realizar todos los pasos para crear el commit, ejecuta el comando "git push -u origin (nombre-rama)" para enviar el commit a GitHub.
  - **¿Qué es un repositorio remoto?:** Un repositorio remoto en GitHub es una copia de tu repositorio de código que se almacena en los servidores de GitHub. Esto permite que otros desarrolladores puedan acceder y contribuir al código, y también te permite mantener una copia de seguridad de tu trabajo.
  - **¿Cómo agregar un repositorio remoto a Git?:** Ejecuta el comando "git remote add (nombre-remoto) (url-del-repositori)" para agregar el repositorio remoto. Para verificar si se agregó correctamente ejecutamos "git remote -v"
  - **¿Cómo empujar cambios a un repositorio remoto?:**
    1. Agregar los cambios al área de preparación: Ejecuta el comando "git add ." para agregar los cambios realizados en el archivo al área de preparación.
    2. Crear un commit: Ejecuta el comando "git commit -m ("mensaje-del-commit")" para crear un commit.
    3. Empujar los cambios al repositorio remoto: Ejecuta el comando "git push (nombre-remoto) (nombre-rama)" para empujar los cambios al repositorio remoto.
  - **¿Cómo tirar de cambios de un repositorio remoto?:** Ejecuta el comando "git pull (nombre-remoto) (nombre-rama)" para tirar de cambios del repositorio remoto.
  - **¿Qué es un fork de repositorio?:** Un fork es una copia independiente del repositorio original, lo que significa que los cambios realizados en el fork no afectan directamente al repositorio original.
  - **¿Cómo crear un fork de un repositorio?:**
    1. Inicia sesión en GitHub: Inicia sesión en tu cuenta de GitHub.
    2. Navega al repositorio: Navega al repositorio que deseas forkear.
    3. Haz clic en el botón "Fork": En la esquina superior derecha de la página del repositorio, haz clic en el botón "Fork"

- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?:**
  1. Inicia sesión en GitHub: Inicia sesión en tu cuenta de GitHub.
  2. Navega al repositorio: Navega al repositorio donde deseas enviar la solicitud de extracción.
  3. Haz clic en "Pull requests": En la pestaña "Pull requests" del repositorio, haz clic en el botón "New pull request".
  4. Selecciona la rama: Selecciona la rama que deseas comparar con la rama principal del repositorio.
  5. Escribe un título y una descripción: Escribe un título y una descripción para la solicitud de extracción.
  6. Haz clic en "Create pull request": Haz clic en el botón "Create pull request" para crear la solicitud de extracción.
- **¿Cómo aceptar una solicitud de extracción?:**
  1. Haz clic en "Merge pull request": Haz clic en el botón "Merge pull request" para aprobar la solicitud de extracción.
  2. Escribe un comentario: Escribe un comentario para explicar por qué se aprobó la solicitud de extracción.
  3. Haz clic en "Confirm merge": Haz clic en el botón "Confirm merge" para fusionar los cambios con la rama principal del repositorio.
- **¿Qué es una etiqueta en Git?:** Las etiquetas en Git son referencias a commits específicos en un repositorio. Permiten identificar y marcar puntos importantes en la historia del proyecto, como versiones estables o lanzamientos.
- **¿Cómo crear una etiqueta en Git?:** Existen 2 tipos de etiqueta, la etiqueta ligera que es una simple referencia a un commit, usamos "git tag (nombre-etiqueta)" para crearla y la etiqueta anotada que contiene una información adicional, usamos git tag -a (nombre-etiqueta) -m ("mensaje")
- **¿Cómo enviar una etiqueta a GitHub?:** Ejecuta el comando "git push origin (nombre-etiqueta)" para enviar la etiqueta a GitHub y para mandar todas las etiquetas se ejecuta "git push origin --tags".
- **¿Qué es un historial de Git?:** Un historial de Git es un registro cronológico de todos los cambios realizados en un repositorio de Git. Es una herramienta fundamental para entender la evolución de un proyecto y para gestionar los cambios realizados por los desarrolladores.
- **¿Cómo ver el historial de Git?:** Para ver el historial de Git, se ejecuta el comando "git log".
- **¿Cómo buscar en el historial de Git?:** Para buscar en el historial de commits por mensaje de commit utilizando la opción -S o --grep con el comando git log. Ej: git log -S "mensaje de commit".  
 Para buscar en el historial de commits por autor utilizando la opción --author con el comando git log. Ej: git log --author="Nombre del Autor".  
 Para buscar en el historial de commits por fecha utilizando las opciones --since y --until con el comando git log. Ej: git log --since="1 semana", git log --until="1 semana".  
 Para buscar en el historial de commits por contenido utilizando los siguientes comandos git log -G "palabra o frase"; git grep "palabra o frase".

Puedes utilizar el comando “gitk” para visualizar el historial de commits y buscar en él.

- **¿Cómo borrar el historial de Git?:** Para borrar el historial de Git utilizamos los siguientes comando:  
“git reset”: elimina el stage de todos los archivos y carpetas del proyecto  
“git reset (nombre-archivo)”: elimina el stage de archivo declarado  
“git reset (nombre-carpeta/)”: elimina todos los stage que se encuentran en esa carpeta  
“git reset (nombre-carpeta/\*.extensión)”: elimina todos los archivos con la condición declarada dentro de la carpeta del stage
- **¿Qué es un repositorio privado en GitHub?:** Un repositorio privado en GitHub es un tipo de repositorio que solo es accesible para los usuarios que han sido invitados a colaborar en él. A diferencia de los repositorios públicos, que son accesibles para cualquier persona, los repositorios privados están protegidos por permisos de acceso y solo pueden ser vistos y modificados por los usuarios autorizados.
- **¿Cómo crear un repositorio privado en GitHub?:** Para crear un repositorio privado se debe hacer lo siguientes pasos:
  1. Iniciar Sesión en GitHub.
  2. Crear un Nuevo Repositorio (Haz clic en el botón "+" en la esquina superior derecha de la pantalla y selecciona “nuevo repositorio”)
  3. Configurar el Repositorio: (Ingresa el nombre del repositorio en el campo "Nombre del repositorio, si quieres puedes agregarle una descripción en el campo “Descripción”, selecciona “Privado” como tipo de repositorio.
  4. Configurar los Permisos de Acceso: Puedes agregar colaboradores al repositorio seleccionando "Agregar colaboradores" y ingresar sus nombres de usuario de GitHub, también puedes asignarles permisos de accesos
- **¿Cómo invitar a alguien a un repositorio privado en GitHub?:** Para invitar a alguien a un repositorio privado debemos hacer los siguientes pasos:
  1. Iniciar Sesión en GitHub.
  2. haz clic en la pestaña "Settings" del repositorio
  3. Selecciona "Collaborators" en el menú de la izquierda.
  4. haz clic en el botón "Add people" e ingresa el nombre de usuario de GitHub de la persona que deseas invitar. (Selecciona el nivel de acceso que deseas otorgar: Read, Triage, Write, Maintain, o Admin.)
  5. Haz clic en el botón "Add" para enviar la invitación.
- **¿Qué es un repositorio público en GitHub?:** Un repositorio público en GitHub es un tipo de repositorio que es accesible para cualquier persona en Internet. Cualquiera puede ver, clonar y forkear un repositorio público, lo que lo hace ideal para proyectos de código abierto y colaboración en comunidad.
- **¿Cómo crear un repositorio público en GitHub?:** Para crear un repositorio público se debe hacer lo siguientes pasos:
  1. Iniciar Sesión en GitHub.
  2. Crear un Nuevo Repositorio (Haz clic en el botón "+" en la esquina superior derecha de la pantalla y selecciona “nuevo repositorio”)
  3. Configurar el Repositorio: (Ingresa el nombre del repositorio en el campo "Nombre del repositorio, si quieres puedes agregarle una descripción en el campo “Descripción”, selecciona “Público” como tipo de repositorio.

- **¿Cómo compartir un repositorio público en GitHub?:** Para compartir un repositorio público en GitHub se debe hacer lo siguiente:
  1. Iniciar Sesión en GitHub.
  2. Haz clic en el botón "Repositorios" en la barra de navegación superior.
  3. Selecciona el repositorio público que deseas compartir.
  4. Haz clic en el botón "Compartir" en la parte superior derecha de la página del repositorio.
  5. Selecciona la opción de compartir que deseas:
  6. **Enlace:** Comparte el enlace del repositorio.  
**Redes sociales:** Comparte el repositorio en redes sociales como Twitter, Facebook o LinkedIn.  
**Correo electrónico:** Comparte el repositorio por correo electrónico.

## Actividad 2:

LINK: <https://github.com/LucaR0801/TP-2-GITHUB.git>

## Actividad 2:

LINK: <https://github.com/LucaR0801/conflict-exercise.git>

