

## Trabajo practico GIT/GITUP

Alumno: Leonel Jesus Aballay

Comisión: 6

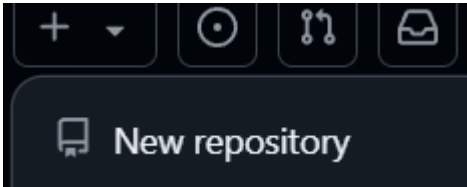
### Actividades:

1. ¿Qué es GitHub?
2. ¿Cómo crear un repositorio en GitHub?
3. ¿Cómo crear una rama en Git?
4. ¿Cómo cambiar a una rama en Git?
5. ¿Cómo fusionar ramas en Git?
6. ¿Cómo crear un commit en Git?
7. ¿Cómo enviar un commit a GitHub?
8. ¿Qué es un repositorio remoto?
9. ¿Cómo agregar un repositorio remoto a Git?
10. ¿Cómo empujar cambios a un repositorio remoto?
11. ¿Cómo tirar de cambios de un repositorio remoto?
12. ¿Qué es un fork de repositorio?
13. ¿Cómo crear un fork de un repositorio?
14. ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
15. ¿Cómo aceptar una solicitud de extracción?
16. ¿Qué es un etiqueta en Git?
17. ¿Cómo crear una etiqueta en Git?
18. ¿Cómo enviar una etiqueta a GitHub?
19. ¿Qué es un historial de Git?
20. ¿Cómo ver el historial de Git?
21. ¿Cómo buscar en el historial de Git?
22. ¿Cómo borrar el historial de Git?
23. ¿Qué es un repositorio privado en GitHub?
24. ¿Cómo crear un repositorio privado en GitHub?
25. ¿Cómo invitar a alguien a un repositorio privado en GitHub?
26. ¿Qué es un repositorio público en GitHub?
27. ¿Cómo crear un repositorio público en GitHub?
28. ¿Cómo compartir un repositorio público en GitHub?

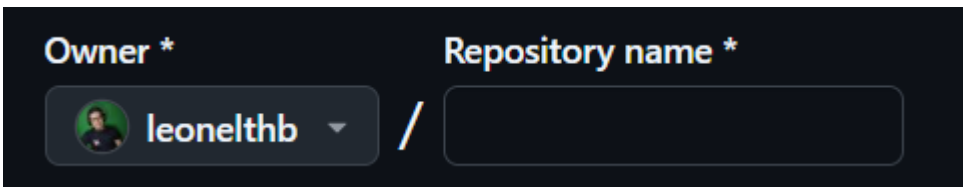
1). GitHub es una plataforma de desarrollo colaborativo basada en la web que utiliza Git como sistema de control de versiones. Permite a los desarrolladores almacenar, compartir y gestionar proyectos de código fuente.

2). Crear un perfil/ cuenta en <https://github.com>

Arriba a la derecha darle en el icono + y nuevo repositorio



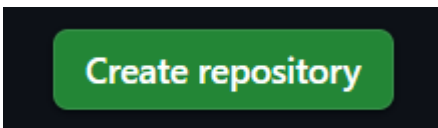
Ingresar un nombre para el repositorio

A screenshot of the 'New repository' form on GitHub. It features two main input fields. The first is labeled 'Owner \*' and contains a dropdown menu with a profile picture and the username 'leonelthb'. The second is labeled 'Repository name \*' and is an empty text input field. A slash '/' is positioned between the two fields.

Una descripción “opcional”

Determinar si será de uso Publico o privado

Y para finalizar “Crear Repositorio”



3). Dentro de la terminal de GIT usar el comando

`git branch “name”`

`git commit` para guardar y para separar las ramas.

4). Para cambiar de una rama a otra usamos el comando

`git checkout “Nombre de la rama”`

5). Para fucionar ramas en GIT debemos usar el comando

Debemos asegurar de estar en la rama secundaria utilizando el “**git checkout(nombre de la rama)**”

Luego usamos el “**git merge main**” para unir las ramas

6). Agregar cambios en el archivo que estemos trabajando

Git add .

En caso de querer agregar un archivo especifico

**Git add Archivo\_especifico**

Y para realizar un commit con un mensaje informando los cambio s

**Git commit -m** (permite incluir el mensaje directamente) “**mensaje del cambio**”

7). Para enviar un repositorio local a remoto debemos hacer un PUSH

Siempre y cuando tengamos el repositorio creado, lo que debemos hacer es ir a nuestro perfil copiar el link del repositorio remoto y pegar en nuestra consola

**Git remote add origin** <https://github.com/leonelthb/conflict-exercise.git>

Luego dar la instruccion push

**Git push -u origin master**

8). Un **repositorio remoto** es una versión de tu proyecto que se almacena en un servidor en línea, fuera de tu máquina local, lo que te permite acceder y compartir tu código con otros usuarios de manera colaborativa.

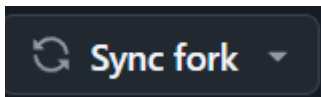
9). Para agregar un repositorio remoto a Git, puedes usar el comando git remote add. Esto vincula tu repositorio local con un repositorio remote

**Git remote add origin** <https://github.com/leonelthb/conflict-exercise.git>

10). Con el comando EJ: `Git push -u origin master`

11). Con el comando: `git pull origin (Nombre de la rama)`

12). Crear un **fork** de un repositorio en GitHub es un proceso sencillo que te permite crear una copia del repositorio original en tu propia cuenta de GitHub. Esto es útil cuando deseas contribuir a un proyecto, pero no tienes permisos para modificar directamente el repositorio original.



13). Acceder al repositorio original en GitHub

En la parte superior, buscar el botón fork

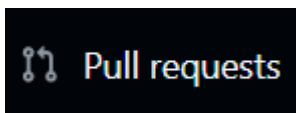
Para crear una copia y para finalizar colar a tu maquina local

Git clone (Direccion URL)

14). Hacer cambios en una rama y realizar un commit

Subir los cambios a tu repositorio en GitHub, en el repositorio hacer click en "Compare" o crear un nuevo pull request en la pestaña "Pull requests", escribir un titulo y dejar una explicación para explicar los cambios y para finalizar "Create Pull request"

15). En la pestaña "Pull requests" del repositorio



Revisar "Pull requests"

Buscar "Merge pull request" y confirmar el merge "Confirm Merge"

16). Son etiquetas que hacen referencias que apuntan a commits específicos. Pueden ser ligeras (simples punteros) o anotadas (con metadatos adicionales). Se utilizan para marcar versiones o puntos importantes en el desarrollo del proyecto.

17). Usar el comando git tag, que te permite marcar un punto específico en el historial de commits, hay dos tipos. Etiqueta ligera y etiquetas anotadas

Git tag nombre-etiqueta

Git tag v1.0

Git tag -a nombre-etiqueta -m "mensaje de la etiqueta"

18). Crear una etiqueta en tu repositorio local

**Etiqueta ligera:** Git tag nombre-etiqueta

**Etiqueta anotada:** git tag -a nombre-etiqueta -m "mensaje de la etiqueta"

Verificar que la etiqueta fue creada

Enviar una etiqueta a github

**Git push origin nombre-etiqueta**

Enviar todas las etiquetas al repositorio remoto

Para finalizar ir a la página de tu repositorio github y en la pestaña "releases o tags" del repositorio, podrás ver la etiqueta que has subido, esto especialmente útil cuando marcas versiones o lanzamiento de tu proyecto.

19). Es una representación completa de todos los cambios realizados en un repositorio. Se basa en los commits y contiene información sobre la autoría, fecha, mensajes y como las ramas se han desarrollado a lo largo del tiempo. El historial permite a los desarrolladores ver como ha evolucionado un proyecto, colaborar de manera efectiva y realizar un seguimiento o revisión de cambios si es necesario.

20). Para ver el historial de git, usar el comando `git log`, muestra los commits realizados en el repositorio, mostrando los detalles del autor, fecha y mensaje de los commits.

21). Hay varios comandos que permiten filtrar los commits,

`--grep` para una frase específica

`--author` para buscar el autor,

`--since` y `--until` para buscar entre dos fechas

`-s` para buscar contenido de archivos

`--follow` ver el historial de un archivo

`Git show` por si conoces los hash de un commit específico

`--graph` ver historial con gráficos de ramas

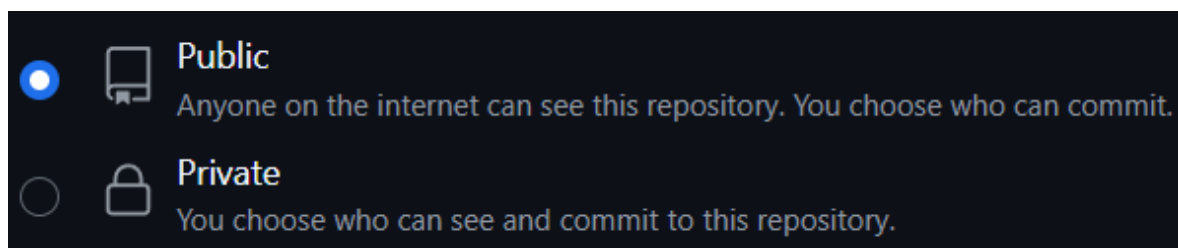
`-n` limitar con el número de commits

22). Para eliminar todo el historial del repositorio y empezar de nuevo (No recomendable) es eliminar el repositorio local y crear uno nuevo.

23). Un repositorio privado, en el que solo las personas que tienen acceso explícito pueden ver, colaborar y realizar cambios en el código., es ideal cuando necesitas mantener tu trabajo o código en desarrollo confidencial y controlado.

Tiene acceso restringido, seguridad, control de acceso, colaboración privada, beneficios en entornos de trabajo.

24). Al iniciar sesión en github, cuando creas un nuevo repositorio, luego de poner el nombre al mismo, puede elegir entre publico o privado.



25). Para invitar a alguien a un repositorio privado en GitHub, necesitas ser el propietario o tener permisos de administrador en ese repositorio, en configuración, gestionar acceso, invitar a un colaborador y elegir los permisos.

26). Un repositorio publico en GitHub es un tipo de repositorio cuyo contenido es accesible para cualquier persona. Acceso a ver, clonar, fork, hacer comentarios y contribuir en el repositorio si tienen los permisos adecuados. Estos repositorios son ideales para proyectos de código abierto.

27). Para crear un repositorio público, se necesita iniciar sesión, crear un nuevo repositorio, configurarlo como publico y luego agregar archivos o colaborar con otros usuarios.

28). Comparte el enlace directo del repositorio en tu navegador

Usar la opción share en github para compartir en redes sociales

Invita a otros a Fork o Clone en tu repositorio

Agrega enlaces dentro de tu README.md

Usar QR para compartir el enlace de manera visual.