

TRABAJO PRÁCTICO Nº 2

Git y GitHub

Consigna 1

1. Github es una plataforma en línea que permite almacenar proyectos que usan el sistema de control de versiones Git. Es el servicio más popular del mundo para compartir y colaborar en proyectos de software, tanto de código abierto como privados. Almacena los repositorios en la nube, lo que facilita el trabajo colaborativo entre desarrolladores ubicados en diferentes partes del mundo.
2. Para crear un repositorio en GitHub, inicia sesión en tu cuenta, ve a la página principal y haz clic en "Nuevo repositorio". Asigna un nombre, descripción y elige si será público o privado. Puedes optar por inicializar el repositorio con un archivo README.
3. Para crear una rama en Git, puedes utilizar el comando `git branch <nombre>`.
4. Para cambiar a una rama en Git, puedes utilizar el comando `git checkout <nombre>`.
5. Para fusionar ramas en Git, puedes utilizar el comando `git merge <rama>`.
6. Para crear un commit en Git, primero debes usar el comando `git add` para añadir los archivos que quieres incluir en el commit, y luego usar el comando `git commit` para crear la instantánea y registrar los cambios. El comando `git commit` requiere un mensaje descriptivo para explicar los cambios realizados.
7. Un repositorio remoto es un repositorio alojado generalmente en plataformas como GitHub o GitLab. Permite sincronizar el trabajo con otros desarrolladores.
8. Para agregar un repositorio remoto nuevo, use el comando `git remote add` en el terminal, dentro del directorio donde está almacenado su repositorio. El comando `git remote add` toma dos argumentos: Un nombre remoto, por ejemplo, origin.
9. Para "empujar" (o subir) cambios a un repositorio remoto en Git, se usa el comando `git push`. Primero, debes asegurarte de que todos los cambios locales estén confirmados (commit) y luego usar `git push <nombre_del_repositorio_remoto> <nombre_de_la_rama>`. Por ejemplo, `git push origin main` subirá los cambios de la rama main local al repositorio remoto llamado origin.
10. Para "tirar" (o push) cambios a un repositorio remoto en Git, se usa el comando `git push`.

- 11.** Un "fork" de repositorio, en el contexto de Git y plataformas como GitHub, significa crear una copia independiente de un repositorio original en tu propia cuenta. Esto te permite tener tu propia versión del proyecto, realizar cambios, y, si lo deseas, sugerir estos cambios al proyecto original. Es como una bifurcación, donde ambos repositorios (el original y el fork) pueden evolucionar independientemente, o el fork puede ser una base para contribuir al original.
- 12.** Para crear un fork de un repositorio, primero debes navegar a la página del repositorio original. Luego, busca el botón "Fork" (o "Bifurcar" en algunos casos) en la parte superior derecha de la página y haz clic en él. Se te pedirá que selecciones el usuario o la organización en la que quieres crear el fork. El fork se crea como una copia independiente de tu repositorio original, que podrás usar para realizar cambios sin afectar el repositorio original.
- 13.** Para enviar una solicitud de extracción (Pull Request) en GitHub, primero debes tener una bifurcación (fork) del repositorio que quieras modificar. Luego, crea una rama local con tus cambios, realiza los cambios, sube la rama a tu bifurcación y finalmente, crea una solicitud de extracción en la bifurcación original.
- 14.** Para aceptar una solicitud de extracción (Pull Request) en GitHub, primero debes revisar los cambios propuestos, asegurándote de que cumplan con los estándares de calidad y que no tengan errores. Luego, si estás de acuerdo con los cambios, puedes aprobar la solicitud y fusionarla con el repositorio principal.
- 15.** En Git, una etiqueta (tag) es una referencia que apunta a un punto específico en el historial de versiones de un proyecto. Generalmente, se utilizan para marcar lanzamientos, hitos o versiones importantes del software.
- 16.** Para crear una etiqueta en git asegúrate de tener un repositorio Git existente usando el comando `git init` en el directorio raíz de tu proyecto. Luego, usa el comando `git tag -a <tagname>` para crear una nueva etiqueta. Esto abrirá el editor de texto para que puedas escribir un mensaje asociado a esta versión. Guárdalo cuando termines.
- 17.** Para enviar una etiqueta de Git a un repositorio remoto, primero debes crear la etiqueta localmente y luego usar el comando `git push` con la opción `--tags` para enviarla al servidor remoto.
- 18.** En Git, el historial (history) es un registro de todas las modificaciones realizadas en un proyecto a lo largo del tiempo. Este registro se compone de una serie de

confirmaciones (commits), que son instantáneas del proyecto en un momento específico. Cada commit contiene información sobre los cambios realizados, el autor, la fecha y un mensaje descriptivo.

19. Para ver el historial de Git, puedes usar el comando `git log`. Este comando muestra las confirmaciones en orden cronológico, mostrando el ID del commit, el autor, la fecha y el mensaje de cada confirmación. También puedes filtrar el historial para ver solo los cambios en un archivo específico, o por autor, fecha, o incluso por una línea de código específica.
20. Para buscar en el historial de Git, puedes usar el comando `git log` con diferentes opciones para filtrar la información y encontrar los commits que necesitas.
21. Para eliminar una confirmación específica de tu historial, puedes usar `git rebase`. Este método es preferible para las confirmaciones que no están al principio de la rama. Reemplaza `<commit-hash>` con el hash de la confirmación justo antes de la que deseas eliminar.
22. Un repositorio privado en GitHub es un espacio para almacenar código, archivos y revisiones de forma que solo tú y las personas que le des acceso puedan verlo.
23. Para invitar a alguien a un repositorio privado en GitHub, debes acceder a la configuración del repositorio, luego ir a "Colaboradores y equipos" y finalmente agregar a la persona como colaborador, especificando el rol que tendrá (lectura, escritura o administración).
24. Un repositorio público en GitHub es un almacén de código, archivos y su historial de versiones, accesible para cualquier persona en internet. Esto significa que cualquier usuario, incluso sin una cuenta en GitHub, puede ver, descargar y clonar el repositorio. Sin embargo, para realizar cambios o contribuir al código, se requiere ser el dueño del repositorio o tener permisos de colaborador.
25. Para crear un repositorio público en GitHub:
 1. Crear un nuevo repositorio: En GitHub, puedes crear un nuevo repositorio.
 2. Seleccionar "Público" en la visibilidad: Cuando creas el repositorio, elige la opción "Público" para que sea accesible para todos.
 3. Generar una licencia (opcional pero recomendable): Para facilitar el uso del código, es recomendable generar una licencia que defina los términos de uso, modificación y distribución del software.
 4. Compartir la URL: Una vez creado, puedes compartir la URL del repositorio con otros para que puedan acceder a él.

26. Para compartir un repositorio público en GitHub tenes que: en el nombre del repositorio, haz clic en Configuración. Si no puedes ver la pestaña "Configuración", selecciona el menú desplegable y, a continuación, haz clic en Configuración. En la sección "Acceso" de la barra lateral, haz clic en Colaboradores. Haz clic en Agregar personas.

Consigna 2

<https://github.com/LuMendez7/Actividad-2>

Consigna 3

https://github.com/LuMendez7/Actividad_3