

## Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

GitHub es un espacio en línea donde puedes guardar tus proyectos de programación, como si tuvieras un disco duro en la nube diseñado especialmente para código. Permite que varias personas trabajen juntas en el mismo proyecto, compartiendo enlaces y viendo los cambios de cada uno, lo que facilita la colaboración a distancia

- ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, el primer paso es acceder a tu cuenta. Si aún no tienes una, el proceso de creación es sencillo: dirígete a la esquina superior derecha de la página y haz clic en "Registrarse". Sigue las instrucciones, ingresando tu nombre de usuario, correo electrónico y contraseña. GitHub te enviará un código de verificación para confirmar tu cuenta.

Una vez que tengas tu cuenta, inicia sesión y busca el botón "Crear repositorio", generalmente ubicado en la parte superior izquierda de la página. Al hacer clic, se abrirá una nueva página donde podrás configurar tu repositorio. Aquí, debes elegir un nombre para tu repositorio, que será la identificación principal de tu proyecto. También puedes agregar una descripción opcional, que ayudará a otros a entender de qué trata tu proyecto.

Un paso importante es elegir la visibilidad de tu repositorio: público o privado. Si eliges público, cualquiera podrá ver tu código; si eliges privado, solo las personas que invitas podrán acceder a él. Además, tienes la opción de inicializar tu repositorio con un archivo README, que es un archivo de texto que suele usarse para proporcionar información sobre el proyecto.

Finalmente, haz clic en el botón "Crear repositorio" para finalizar el proceso. ¡Y listo! Tu repositorio estará creado y listo para que comiences a subir tu código.

- ¿Cómo crear una rama en Git?

Para crear una rama en Git, debes abrir la terminal en el directorio de tu repositorio. Luego, utiliza el comando `git branch <nombre_de_la_rama>`. Por ejemplo, para crear una rama llamada "nueva-funcionalidad", escribirías `git branch nueva-funcionalidad`.

Es importante recordar que este comando solo crea la rama, pero no te cambia a ella.

- ¿Cómo cambiar a una rama en Git?

Para comenzar a trabajar en la nueva rama, necesitas usar el comando `git checkout <nombre_de_la_rama>`. Siguiendo el ejemplo anterior, escribirías `git checkout nueva-funcionalidad`.

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero debes cambiar a la rama en la que deseas incorporar los cambios de otra rama. Esto se hace con el comando `git checkout <rama_destino>`. Por

ejemplo, si quieres fusionar una rama llamada "nueva-funcionalidad" en la rama "master", primero debes cambiar a la rama "master" con `git checkout master`. Luego, utiliza el comando `git merge <rama_origen>`. En el ejemplo anterior, escribirías `git merge nueva-funcionalidad`. Git intentará combinar los cambios automáticamente.

- ¿Cómo crear un commit en Git?

Para crear un commit en Git, el primer paso es revisar los cambios que se han realizado. Puedes hacerlo con el comando `git status` en la terminal, dentro del directorio de tu repositorio. Este comando te mostrará los archivos que han sido modificados o agregados.

Una vez que estés seguro de los cambios que quieres guardar, debes agregarlos al área de preparación (stage) con el comando `git add`. Si quieres agregar un archivo específico, usa `git add <nombre_del_archivo>`. Para agregar todos los cambios en el directorio actual, usa `git add ..`

Después de agregar los cambios al área de preparación, puedes crear el commit con el comando `git commit -m "Mensaje del commit"`. El mensaje del commit debe ser claro y descriptivo, explicando qué cambios se han realizado. Por ejemplo, `git commit -m "Corregidos errores de diseño en la página de inicio"`.

Si olvidas la bandera "-m" seguida del mensaje, Git abrirá un editor de texto en la terminal para que escriba el mensaje del commit.

- ¿Cómo enviar un commit a GitHub?

Después de crear un commit en tu repositorio local con el comando `git commit`, los cambios se guardan solo en tu computadora. Para enviar esos cambios a GitHub, necesitas usar el comando `git push`.

Primero, asegúrese de que su repositorio local esté conectado a su repositorio remoto en GitHub. Esto se hace con el comando `git remote add origin <URL_del_repositorio>`. Reemplaza <URL\_del\_repositorio> con la URL de tu repositorio en GitHub.

1

Este paso generalmente solo es necesario la primera vez que conectes tu repositorio local con el remoto.

Luego, para enviar el commit a GitHub, utiliza el comando `git push origin <nombre_de_la_rama>`. Reemplaza <nombre\_de\_la\_rama> con el nombre de la rama que quieres enviar. Por ejemplo, si estás en la rama "main", escribirías `git push origin main`.

Este comando envía los commits de tu rama local a la rama correspondiente en GitHub. Si es la primera vez que envías una rama nueva, GitHub creará la rama automáticamente en el repositorio remoto.

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu proyecto que se encuentra alojada en un servidor en línea, como si fuera un disco duro en la nube, como GitHub o GitLab. A diferencia de un repositorio local, que reside en tu computadora, un repositorio remoto permite la colaboración entre múltiples desarrolladores y sirve como una ubicación centralizada para almacenar y compartir el código.

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git, el comando fundamental es `git remote add`. Este comando establece una conexión entre tu repositorio local y un repositorio remoto, permitiéndote enviar y recibir cambios.

- ¿Cómo empujar cambios a un repositorio remoto?

Para enviar los commits que has creado localmente a un repositorio remoto en GitHub, debes usar el comando `git push`. Este comando sube los cambios de tu repositorio local al repositorio remoto.

- ¿Cómo tirar de cambios de un repositorio remoto?

Para obtener los cambios más recientes de un repositorio remoto y fusionarlos con tu rama local, utiliza el comando `git pull origin <nombre_de_la_rama>`. Reemplaza `<nombre_de_la_rama>` con el nombre de la rama que quieres actualizar. Por ejemplo, `git pull origin master`. Este comando descarga los cambios del repositorio remoto y los fusiona automáticamente con tu rama local.

- ¿Qué es un fork de repositorio?

Un "fork" de un repositorio es una copia personal de un repositorio que pertenece a otra persona. Te permite realizar cambios en el código sin afectar el repositorio original. Los forks son comunes en proyectos de código abierto, donde los desarrolladores contribuyen con mejoras y correcciones.

- ¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio, ve a la página del repositorio en GitHub y haz clic en el botón "Fork". GitHub creará una copia del repositorio en tu cuenta.

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

Después de realizar cambios en tu fork, puedes enviar una solicitud de extracción (pull request) al repositorio original. Esto le notifica al propietario del repositorio que tienes cambios que te gustaría que considerara.

- ¿Cómo aceptar una solicitud de extracción?

El propietario del repositorio original puede revisar tus cambios y, si está satisfecho, puede aceptar la solicitud de extracción. Esto fusiona tus cambios con el repositorio original.

- ¿Qué es un etiqueta en Git?

Una etiqueta (tag) en Git es una instantánea estática de un punto específico en la historia de tu repositorio. Se utiliza comúnmente para marcar versiones de lanzamiento (por ejemplo, "v1.0").

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta, usa el comando `git tag <nombre_de_la_etiqueta>`. Para crear una etiqueta con un mensaje, usa `git tag -a <nombre_de_la_etiqueta> -m "Mensaje de la etiqueta"`.

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub, usa el comando ``git push origin <nombre_de_la_etiqueta>``. Para enviar todas las etiquetas, usa ``git push origin --tags``.

- ¿Qué es un historial de Git?

El historial de Git es un registro de todos los commits realizados en un repositorio. Te permite ver quién hizo qué cambios y cuándo.

- ¿Cómo ver el historial de Git?

Para ver el historial de Git, usa el comando ``git log``. Este comando muestra una lista de commits, junto con sus mensajes, autores y fechas.

- ¿Cómo buscar en el historial de Git?

Puedes buscar en el historial de Git usando el comando ``git log --grep="palabra clave"``. Esto mostrará los commits que contienen la palabra clave especificada en sus mensajes.

- ¿Cómo borrar el historial de Git?

Borrar el historial de Git puede ser peligroso y generalmente no se recomienda. Sin embargo, si necesitas hacerlo, puedes usar el comando ``git reset``. Ten mucho cuidado con este comando, ya que puede provocar la pérdida de datos.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio que solo pueden ver las personas que invitas.

- ¿Cómo crear un repositorio privado en GitHub?

Al crear un nuevo repositorio en GitHub, puedes elegir la opción "Private".

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

En la página de configuración del repositorio, puedes invitar a otros usuarios a colaborar.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio que cualquiera puede ver.

- ¿Cómo crear un repositorio público en GitHub?

Al crear un nuevo repositorio en GitHub, puedes elegir la opción "Public".

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público, simplemente comparte la URL del repositorio. Cualquier persona puede acceder a él.