

Trabajo practico n2 git - github

¿Qué es GitHub?

GitHub es una plataforma web utilizada para alojar y gestionar proyectos de software que emplean Git como sistema de control de versiones. Permite a los desarrolladores colaborar en proyectos de manera eficiente, facilitando el seguimiento de cambios y la integración de contribuciones. Además, GitHub proporciona herramientas como issues, pull requests y acciones automáticas para mejorar el flujo de trabajo de los equipos de desarrollo.

¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, primero inicia sesión en tu cuenta y haz clic en el botón "New" en la parte superior de tu perfil. Luego, deberás proporcionar un nombre para el repositorio y seleccionar si será público o privado. Después de completar la configuración, haz clic en "Create repository" para crear el repositorio y comenzar a usarlo.

¿Cómo crear una rama en Git?

Puedes crear una nueva rama en Git utilizando el comando `git branch nombre-de-la-rama`, donde "nombre-de-la-rama" es el nombre que deseas asignar. Las ramas permiten trabajar de manera aislada en diferentes características o funcionalidades sin afectar la rama principal. Esta es una técnica común para desarrollar características sin interrumpir el flujo de trabajo principal del proyecto.

¿Cómo cambiar a una rama en Git?

Para cambiar a una rama existente en Git, utiliza el comando `git checkout nombre-de-la-rama`. Esto moverá tu área de trabajo a la rama seleccionada y comenzará a reflejar los cambios y commits de esa rama. Asegúrate de haber hecho un commit de tus cambios actuales antes de cambiar de rama para evitar perder trabajo no guardado.

¿Cómo fusionar ramas en Git?

Para fusionar una rama en Git, primero asegúrate de estar en la rama a la que deseas fusionar los cambios, generalmente la rama principal (main). Usa el comando `git merge nombre-de-la-rama` para combinar los cambios de la rama seleccionada con la rama activa. Si hay conflictos entre las ramas, Git te pedirá que los resuelvas manualmente antes de completar la fusión.

¿Cómo crear un commit en Git?

Para crear un commit en Git, primero añade los cambios al área de preparación con `git add nombre-del-archivo`. Luego, utiliza el comando `git commit -m "mensaje del commit"` para registrar esos cambios en el historial de Git, describiendo lo que has modificado. Los commits son fundamentales para seguir el progreso y el historial de tu proyecto de manera organizada.

¿Cómo enviar un commit a GitHub?

Para enviar tus cambios locales a GitHub, utiliza el comando `git push origin nombre-de-la-rama`. Esto empujará tus commits desde tu repositorio local al repositorio remoto alojado en GitHub. Asegúrate de haber configurado previamente el repositorio remoto y de tener acceso a él con las credenciales adecuadas.

¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de tu repositorio alojada en un servidor externo, como GitHub, GitLab o Bitbucket. Los repositorios remotos permiten a los desarrolladores colaborar y compartir el código de manera centralizada, facilitando el acceso a todos los miembros del equipo. También sirven como respaldo seguro de tu trabajo.

¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a tu repositorio local, utiliza el comando `git remote add origin URL-del-repositorio`. Esto vincula tu repositorio local con el remoto, lo que te permite enviar y recibir cambios entre ambos. Asegúrate de usar la URL correcta del repositorio remoto para poder interactuar con él.

¿Cómo empujar cambios a un repositorio remoto?

Para empujar tus cambios al repositorio remoto, utiliza el comando `git push origin nombre-de-la-rama`. Esto actualizará el repositorio en GitHub (o cualquier otro servidor remoto) con los commits que hayas realizado localmente. El comando también asegura que el repositorio remoto se mantenga sincronizado con el trabajo que estás desarrollando localmente.

¿Cómo tirar de cambios de un repositorio remoto?

Si deseas obtener cambios de un repositorio remoto, usa el comando `git pull origin nombre-de-la-rama`. Este comando descargará y fusionará automáticamente los cambios realizados en la rama remota a tu repositorio local. Es útil para mantener tu proyecto actualizado con los últimos cambios hechos por otros colaboradores.

¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio de otro usuario que te permite hacer cambios de manera independiente sin afectar el repositorio original. Los forks son ideales cuando deseas contribuir a un proyecto pero no tienes permisos para modificar directamente el repositorio original. Además, puedes proponer cambios mediante un pull request una vez que hayas terminado de trabajar en tu fork.

¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio en GitHub, ve al repositorio original y haz clic en el botón "Fork" en la esquina superior derecha. GitHub creará una copia del repositorio en tu cuenta, permitiéndote modificarlo sin afectar al repositorio original. Desde allí, puedes trabajar en tus propios cambios y realizar pull requests para contribuir al proyecto original.

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

Para enviar una pull request (PR), primero realiza tus cambios en una rama de tu fork. Luego, ve al repositorio original y haz clic en "New pull request". Selecciona las ramas de tu fork y del repositorio original, proporciona una descripción clara de los cambios y haz clic en "Create pull request" para enviarla a los mantenedores del repositorio.

¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción (PR), ve a la página de la PR en GitHub y revisa los cambios propuestos. Si todo está correcto, haz clic en el botón "Merge pull request" para fusionar los cambios en tu rama principal. Este proceso asegura que los contribuyentes puedan integrar sus modificaciones en el proyecto sin errores.

¿Qué es una etiqueta en Git?

Una etiqueta en Git es una referencia a un punto específico en el historial de commits,

generalmente utilizada para marcar versiones importantes del proyecto, como lanzamientos o hitos. Las etiquetas no se modifican con el tiempo, lo que las convierte en un marcador permanente e inmutable en el historial del proyecto. Son útiles para hacer un seguimiento de versiones específicas.

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git, utiliza el comando `git tag nombre-de-la-etiqueta`. Las etiquetas pueden aplicarse en cualquier commit, y puedes usarlas para marcar versiones como v1.0, v2.0, etc. Es recomendable crear etiquetas después de cada lanzamiento importante del proyecto para facilitar la identificación de versiones.

¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub, usa el comando `git push origin nombre-de-la-etiqueta`. Esto sube la etiqueta creada a tu repositorio remoto en GitHub, donde quedará registrada y accesible para todos los colaboradores. Las etiquetas también se pueden listar en GitHub en la pestaña de Releases.

¿Qué es un historial de Git?

El historial de Git es el registro completo de todos los commits realizados en un repositorio. Este historial permite a los desarrolladores ver el progreso del proyecto a lo largo del tiempo, así como identificar quién hizo cada cambio y por qué. El historial es crucial para el seguimiento y la resolución de problemas en el código.

¿Cómo ver el historial de Git?

Para ver el historial de commits en un repositorio, utiliza el comando `git log`. Este comando muestra una lista detallada de todos los commits, incluyendo el autor, la fecha y el mensaje de cada uno. Puedes usar opciones adicionales para personalizar la salida, como `git log --oneline` para una vista más compacta.

¿Cómo buscar en el historial de Git?

Si necesitas buscar en el historial de commits, puedes usar el comando `git log --grep="palabra-clave"`. Esto te permitirá encontrar commits específicos que contienen la palabra clave que hayas especificado en el mensaje del commit. Es útil cuando buscas un

cambio particular o un commit relacionado con un problema específico.

¿Cómo borrar el historial de Git?

Para borrar o modificar el historial de Git, puedes usar comandos como `git rebase` o `git reset`. El comando `git rebase` te permite reescribir el historial de commits, mientras que `git reset` te permite deshacer cambios realizados en el historial local. Es importante tener cuidado al realizar estas acciones, ya que pueden afectar la integridad del historial del proyecto.

¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio cuyo acceso está restringido solo a usuarios autorizados. Esto significa que solo los colaboradores que hayan sido invitados pueden ver o modificar el código. Los repositorios privados son útiles para proyectos en desarrollo que aún no están listos para ser compartidos públicamente.

¿Cómo crear un repositorio privado en GitHub?

Al crear un repositorio en GitHub, puedes seleccionar la opción "Private" en la sección de visibilidad. Esto hará que solo los usuarios que hayas invitado puedan acceder al repositorio y colaborar en él. Los repositorios privados son ideales para mantener la privacidad de proyectos sensibles o en desarrollo.

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

Para invitar a alguien a tu repositorio privado en GitHub, ve a `Settings > Collaborators` y agrega el nombre de usuario de la persona que deseas invitar. La persona recibirá una invitación para colaborar en el repositorio. Una vez que acepte la invitación, tendrá acceso completo o limitado dependiendo de los permisos que hayas configurado.

¿Qué es un repositorio público en GitHub?

Un repositorio público es un repositorio accesible por cualquier persona en la web. Cualquier usuario puede ver, clonar o contribuir al proyecto, lo que lo hace ideal para proyectos de código abierto. Los repositorios públicos permiten compartir y colaborar en proyectos con la comunidad global.

¿Cómo crear un repositorio público en GitHub?

Cuando creas un nuevo repositorio en GitHub, puedes seleccionar "Public" en la opción de visibilidad. Esto hace que el repositorio sea accesible por cualquier persona y visible públicamente en la web. Los repositorios públicos son ideales para proyectos de código abierto o cuando deseas que otros contribuyan a tu proyecto.

¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub, simplemente copia la URL del repositorio y compártela con otras personas. Cualquiera que tenga acceso a esta URL podrá ver el código, clonar el repositorio o hacer contribuciones a través de pull requests. Los repositorios públicos son una excelente manera de fomentar la colaboración en proyectos abiertos.