Trabajo Practico N°2 Git y GitHub

Que es GitHub?

Es una plataforma en línea que permite a los desarrolladores y equipos de desarrollo almacenar, gestionar y compartir código fuente. Es una herramienta basada en Git, un sistema de control de versiones distribuidas, que ayuda a seguir el historial de los cambios realizados en el código y a colaborar con otros programadores.

Cómo crear un repositorio en GitHub?

Entrar a la Página de GitHub, Iniciar sesión (Si no tienes una cuenta) registrarte, Una vez dentro, Hacer clic en el icono de New Repositorio. Escribir un nombre al repositorio, Escribir una Descripción y modificar si hacerlo Público o Privado

Cómo crear una rama en Git?

Una rama en Git es un proceso fundamental dentro del control de versiones, permite desarrollar nuevas característica y corregir errores sin afectar la rama principal. La ventaja es que permite trabajar en múltiples desarrollos de manera simultánea sin interferir con el código

La rama principal en la mayoría de los repositos Git es main o master, pero Git permite la creación de múltiples ramas para distintas finalidades

¿Cómo cambiar a una rama en Git?

**Moverte de un entorno de trabajo a otro**, cada uno con su propio conjunto de cambios, archivos y versiones del proyecto. Al cambiar de rama, Git cambia el contexto del repositorio para que el directorio de trabajo se corresponda con los archivos y el historial de la rama seleccionada. Esto permite trabajar de manera aislada en distintas tareas sin que los cambios en una rama afecten a otras.

Cómo fusionar ramas en Git?

**Git** se utiliza no solo para gestionar y versionar código, sino también para facilitar el trabajo colaborativo y la organización de proyectos complejos mediante el uso de **ramas**. La posibilidad de **fusionar ramas** es uno de los aspectos más poderosos de Git, ya que permite a los desarrolladores combinar cambios realizados en diferentes ramas sin perder el trabajo de ninguno de ellos. Esta fusión de ramas es una operación común en el flujo de trabajo de desarrollo colaborativo.

¿Cómo crear un commit en Git?

Un **commit** es una unidad de trabajo en Git que guarda el estado de los archivos en el repositorio en un momento específico. Cada commit es independiente, pero a la vez está vinculado al commit anterior, formando así una cadena que constituye el **historial del proyecto**. Esta historia se convierte en un registro importante que permite a los desarrolladores realizar un seguimiento de los cambios, revertir a versiones anteriores o colaborar con otros miembros del equipo sin perder el contexto de los cambios.

Cómo enviar un commit a GitHub?

Una de las principales ventajas de usar **Git** junto con plataformas de hospedaje de código como **GitHub** es la posibilidad de **trabajar de manera colaborativa** y mantener el código centralizado en un repositorio remoto. Después de hacer cambios en tu repositorio local, el siguiente paso es enviar esos cambios (commits) al repositorio en GitHub para que estén accesibles para ti y para los demás miembros del equipo. Este proceso implica el uso de **push** para enviar los commits desde tu repositorio local al remoto.

Qué es un repositorio remoto?

Un **repositorio remoto** es un repositorio de código fuente que se encuentra alojado en un servidor o en la nube, accesible a través de internet. A diferencia de un **repositorio local**, que está en el equipo del desarrollador, el repositorio remoto permite que varios usuarios o desarrolladores puedan colaborar de manera simultánea en el mismo proyecto, compartiendo y actualizando el código desde diferentes ubicaciones.

Cómo agregar un repositorio remoto a Git?

Para agregar un **repositorio remoto** a Git, se debe vincular tu repositorio local con uno que esté alojado en un servidor o en una plataforma de repositorios en la nube como GitHub. Este proceso se hace mediante el comando git remote add.

¿Cómo empujar cambios a un repositorio remoto?

Para **realizar cambios a un repositorio remoto** en Git, es decir, para hacer modificaciones en un repositorio alojado en un servidor se utiliza el comando git push. Este comando sincroniza tu repositorio local con el repositorio remoto, permitiendo que otros colaboradores puedan ver y trabajar en tus cambios.

Cómo tirar de cambios de un repositorio remoto?

Para **tirar de cambios** de un repositorio remoto en Git, lo que se busca es **descargar** los últimos cambios que otros colaboradores hayan realizado y empujado al repositorio remoto, y traerlos a tu repositorio local. El proceso para hacer esto se realiza mediante el comando git pull.

Qué es un fork de repositorio?

Un **fork** de repositorio es una **copia** completa de un repositorio de código fuente que se crea bajo tu propia cuenta de usuario en una plataforma de alojamiento de código. Al hacer un fork de un repositorio, se crea una nueva versión del mismo repositorio, con la que puedes trabajar de manera independiente, pero manteniendo el historial y la estructura del proyecto original.

¿Cómo crear un fork de un repositorio?

Crear un **fork** de un repositorio es un proceso relativamente sencillo, y la mayoría de las plataformas de alojamiento de código lo hacen muy accesible a través de sus interfaces gráficas. En teoría, el proceso implica hacer una **copia** de un repositorio existente bajo tu propia cuenta de usuario, lo que te permitirá modificarlo independientemente del repositorio original.

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

Enviar una **solicitud de extracción** o **pull request** es el proceso mediante el cual propones cambios desde tu **fork** o rama local de un repositorio hacia el repositorio original o hacia una rama de otro colaborador en un proyecto de código compartido. En teoría, esto permite que los mantenedores del proyecto revisen tus cambios antes de integrarlos al repositorio principal.

¿Cómo aceptar una solicitud de extracción?

Aceptar una **solicitud de extracción** o **pull request,** es el proceso mediante el cual un mantenedor o colaborador con permisos en un repositorio revisa y fusiona los cambios propuestos por otro colaborador en un proyecto. Este proceso implica revisar los cambios, asegurarse de que sean correctos y compatibles con el repositorio, y luego integrarlos a la rama principal o a la rama correspondiente del repositorio.

Qué es un etiqueta en Git?

Es una referencia que señala un punto específico en el historial de un repositorio, normalmente usada para marcar versiones importantes de un proyecto. Las etiquetas son útiles para señalar puntos de referencia que se consideran estables o relevantes

¿Cómo crear una etiqueta en Git?

Una **etiqueta** es un proceso sencillo que te permite marcar un commit específico en el historial del repositorio. Las etiquetas son útiles para señalar puntos importantes en el tiempo, como **lanzamientos de versiones** o **hitos** en el desarrollo del proyecto.

Cómo enviar una etiqueta a GitHub?

Es necesario **empujar la etiqueta** desde tu repositorio local al repositorio remoto. Este proceso asegura que las etiquetas creadas localmente estén disponibles para otros colaboradores y que puedan ser utilizadas, por ejemplo, para marcar versiones de un proyecto.

Qué es un historial de Git?

Es una **lista cronológica** de los **cambios** realizados en un repositorio, que incluye todas las modificaciones hechas a los archivos del proyecto a lo largo del tiempo. Este historial está compuesto por **commits**, que son registros de las alteraciones en el código, los cuales pueden incluir información adicional como el autor, la fecha y el mensaje que describe el cambio.

¿Cómo ver el historial de Git?

Puedes ver el historial de cambios realizados en un repositorio utilizando el comando git log, que te permite explorar los **commits** realizados a lo largo del tiempo en el proyecto. Este historial te da información detallada sobre los cambios realizados, los autores de los cambios, los mensajes descriptivos, las fechas de los commits, entre otros.

Cómo buscar en el historial de Git?

En **Git**, puedes **buscar en el historial** de commits para encontrar información específica, como ciertos cambios, mensajes, autores o fechas. Git ofrece varias formas de realizar búsquedas en el historial de commits utilizando comandos como git log con diferentes opciones de filtrado y búsqueda.

Cómo borrar el historial de Git?

Puedes eliminar o reescribir los registros de los commits de un repositorio. Esto puede ser útil si necesitas limpiar un repositorio, quitar información sensible o simplemente reescribir la historia del proyecto. Sin embargo, es importante tener en cuenta que manipular el historial de Git puede afectar a otros colaboradores si ya han clonado el repositorio, ya que cambiar el historial puede hacer que haya diferencias irreconciliables.

Qué es un repositorio privado en GitHub?

Es un repositorio de código fuente que está restringido a un grupo específico de usuarios y no es accesible al público en general. A diferencia de los **repositorios públicos**, en los que cualquier persona puede ver, clonar y colaborar en el código, los repositorios privados solo pueden ser accedidos por las personas a las que el propietario del repositorio ha concedido permisos explícitos.