

Desarrolla el siguiente cuestionario.

1. ¿Qué es Git?

Git es un sistema de control de versiones que permite a desarrolladores gestionar y realizar un seguimiento de los cambios en su código a lo largo del tiempo.

2. ¿Para qué funciona el comando **Git init**?

El comando 'git init' es utilizado para crear un nuevo repositorio Git. Cuando se ejecuta este comando en un directorio, Git inicializa un nuevo repositorio en ese directorio y crea un subdirectorio '.git', que contiene todos los archivos necesarios y la metadata para el repositorio.

3. ¿Qué es una rama?

Una rama es una línea separada de desarrollo que diverge de la línea principal de desarrollo, usualmente referida como la rama "maestra". Una rama permite a los desarrolladores trabajar en funciones, corrección de errores, o experimentos sin afectar la base del código principal hasta que los cambios estén listos para fusionarse con la rama maestra.

4. ¿Cómo saber en qué rama estoy?

Se utiliza el comando 'git branch' para mostrar una lista de todas las ramas en tu repositorio de Git. La rama en la que se está actualmente tendrá un asterisco a su lado para mostrar que es la que está actualmente activa.

5. ¿Quién creó git?

Fue creado por Linus Torvalds en el año 2005, y desde entonces se ha convertido en uno de los sistemas de control de versión más utilizados en la industria del desarrollo de software.

6. ¿Cuáles son los comandos más esenciales de Git?

- **'git init'**: Este comando inicializa un nuevo repositorio de Git en el directorio actual, creando un subdirectorio '.git' y permitiendo un control de versión de los archivos del directorio.
- **'git add'**: Este comando agrega uno o más archivos al área de preparación de Git, preparándonos para el siguiente commit. Los archivos se pueden agregar individualmente o usar comodines para agregar todos los archivos en un directorio.
- **'git commit'**: Este comando crea una nueva confirmación con los cambios en el área de preparación de Git y agrega un mensaje de confirmación para describir los cambios. Las confirmaciones representan instantáneas del código en un momento específico y son esenciales para el control de versiones.
- **'git branch'**: Este comando crea, enlista, o borra ramas del repositorio de Git. Las ramas usualmente son utilizadas para aislar cambios y experimentar con nuevos cambios sin afectar el código principal.

- **'git merge'**: Este comando fusiona los cambios de una rama en otra, combinando los cambios de ambas ramas y creando una nueva versión del código que incluye estos cambios. Esta es una parte esencial para integrar nuevas funciones y corregir errores del código principal.

7. ¿Qué es Git flow?

Git flow es un modelo de ramificación y un conjunto de flujos de trabajo para Git que proporciona un enfoque estructurado para administrar sucursales y lanzamientos en un proyecto de desarrollo de software.

El modelo Git flow define dos ramas principales: 'maestro' y 'desarrollo'. La rama 'maestro' es la rama estable que representa el estado actual listo para la producción del código. La rama 'desarrollo' es la rama principal para desarrollo en proceso y representa el último estado del código base.

8. ¿Qué es trunk-based development?

Es un enfoque de desarrollo de software en el que los desarrolladores envían cambios directamente a la rama principal (a menudo llamada "tronco"), en lugar de utilizar ramas de características de larga duración. En el trunk-based development, la rama principal siempre se mantiene en un estado liberable, con cambios pequeños y frecuentes fusionados rápidamente.