

Cuestionario sobre Git

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido utilizado para gestionar el historial de cambios en proyectos de software. Permite a los desarrolladores trabajar en paralelo sin interferir con el trabajo de otros, asegurando que los cambios se registren de manera eficiente y se puedan revertir o fusionar cuando sea necesario.

2. ¿Para qué funciona el comando git init?

El comando git init se utiliza para inicializar un nuevo repositorio Git en un directorio. Cuando se ejecuta, crea una carpeta oculta llamada .git que contiene todos los archivos necesarios para llevar el control de versiones del proyecto. Es el primer paso para comenzar a usar Git en un proyecto local.

3. ¿Qué es una rama?

Una rama en Git es una versión paralela de un proyecto. Se utiliza para trabajar en nuevas características o realizar cambios sin afectar el código principal. Las ramas permiten el desarrollo independiente y la posibilidad de fusionarlas más tarde con la rama principal, típicamente llamada main o master.

4. ¿Cómo saber en qué rama estoy?

Para saber en qué rama te encuentras en Git, puedes usar el comando git branch. Este comando mostrará todas las ramas del repositorio, y la rama actual estará indicada con un asterisco (*) al lado de su nombre.

5. ¿Quién creó Git?

Git fue creado por Linus Torvalds en 2005, principalmente para gestionar el código fuente del sistema operativo Linux. Fue desarrollado como una respuesta a la insatisfacción con los sistemas de control de versiones existentes en ese momento.

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

Los comandos más esenciales de Git son:

- git init: Inicializa un repositorio.
- git clone: Crea una copia local de un repositorio remoto.
- git add: Agrega cambios al área de preparación para ser confirmados.
- git commit: Guarda los cambios en el historial del repositorio.
- git status: Muestra el estado actual de los archivos y las modificaciones.
- git push: Envía los cambios confirmados a un repositorio remoto.
- git pull: Descarga los cambios de un repositorio remoto.
- git branch: Muestra y gestiona las ramas del repositorio.

7. ¿Qué es Git Flow?

Git Flow es una estrategia de ramificación que define cómo se deben crear y gestionar las ramas en un repositorio Git. Se basa en un flujo estructurado que incluye ramas principales como master y develop, y ramas de soporte como feature, release y hotfix, lo que permite gestionar el ciclo de vida del desarrollo de software de manera ordenada y eficiente.

8. ¿Qué es Trunk-Based Development?

Trunk-Based Development (TBD) es una práctica en la que todos los desarrolladores integran su código de manera continua en una única rama principal, también conocida como "trunk" o "main". En este enfoque, las ramas temporales son muy breves y se integran rápidamente al trunk, fomentando entregas frecuentes y la reducción de conflictos de integración.