



Nombre:

Juan Carlos Díaz Vásquez

Matricula:

2022-1946

Carrera:

Desarrollo de Software

Docente:

Kelyn Tejada

Materia:

Programación III

Tema:

“Tarea GIT”

Tabla de contenido

Introducción	3
Desarrollo del Cuestionario.....	3
¿Qué es Git?	3
¿Cuál es el propósito del comando git init en Git?	3
¿Qué representa una rama en Git y cómo se utiliza?	3
¿Cómo puedo determinar en qué rama estoy actualmente en Git?	4
¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?	4
¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?	4
¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?	5
Conclusión	5

Introducción

El uso de sistemas de control de versiones es esencial en el desarrollo de software moderno. Git, uno de los sistemas de control de versiones más populares y poderosos, ofrece a los desarrolladores una manera eficaz de gestionar el historial de cambios, colaborar en proyectos y mantener la integridad del código fuente. Este cuestionario se centra en los aspectos fundamentales de Git, proporcionando una comprensión clara de sus funciones básicas, comandos esenciales y su importancia en la industria del software. A lo largo de este documento, exploraremos qué es Git, cómo inicializar un repositorio, la gestión de ramas y otros comandos cruciales que facilitan el desarrollo colaborativo y eficiente.

Desarrollo del Cuestionario

¿Qué es Git?

Git es un sistema de control de versiones distribuido, diseñado para manejar todo tipo de proyectos con rapidez y eficiencia. Esta herramienta permite a múltiples desarrolladores trabajar en paralelo, gestionando los cambios de manera coherente y asegurando que el historial de versiones se mantenga íntegro y accesible.

¿Cuál es el propósito del comando `git init` en Git?

El comando `git init` en Git tiene como objetivo inicializar un nuevo repositorio de Git. Este comando crea un nuevo subdirectorio llamado `.git` en el directorio de trabajo actual, donde se almacenarán todos los metadatos y objetos necesarios para el control de versiones del proyecto. En esencia, `git init` convierte un directorio ordinario en un repositorio de Git, listo para seguir y gestionar cambios en el código.

¿Qué representa una rama en Git y cómo se utiliza?

Una rama en Git representa una línea independiente de desarrollo dentro del repositorio. Las ramas permiten a los desarrolladores trabajar en características, correcciones de errores o experimentos de manera aislada del código principal. Se utilizan mediante la

creación de nuevas ramas con el comando `git branch` y el cambio entre ellas con `git checkout` o `git switch`. De esta manera, las ramas facilitan el desarrollo concurrente y la integración controlada de nuevas funcionalidades.

¿Cómo puedo determinar en qué rama estoy actualmente en Git?

Para determinar en qué rama se encuentra actualmente en Git, se puede utilizar el comando `git status`, que muestra el estado del repositorio incluyendo la rama activa. Alternativamente, el comando `git branch` también lista todas las ramas locales y destaca la rama en la que se encuentra el usuario con un asterisco (*) antes del nombre de la rama.

¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?

Git fue creado por Linus Torvalds, el mismo creador del sistema operativo Linux. Su desarrollo comenzó en abril de 2005 como una respuesta a la necesidad de una herramienta de control de versiones rápida y eficiente, después de la discontinuación del uso de BitKeeper por la comunidad del núcleo de Linux.

¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?

Entre los comandos esenciales de Git se encuentran:

- **git clone**: utilizado para copiar un repositorio existente.
- **git add**: agrega cambios en el directorio de trabajo al área de preparación (staging area).
- **git commit**: guarda los cambios preparados en el historial del repositorio.
- **git push**: envía los commits locales a un repositorio remoto.
- **git pull**: actualiza el repositorio local con cambios desde el repositorio remoto.

- **git merge:** integra cambios de una rama en otra.

Estos comandos son fundamentales para la gestión diaria de los repositorios y el flujo de trabajo colaborativo.

¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?

- **Linux Kernel:** el repositorio oficial del núcleo de Linux, el proyecto para el cual Git fue originalmente desarrollado.
- **TensorFlow:** un repositorio popular para la biblioteca de aprendizaje automático desarrollada por Google.
- **React:** el repositorio del conocido framework de JavaScript para la construcción de interfaces de usuario, mantenido por Facebook.
- **Vue.js:** otro framework de JavaScript, ampliamente utilizado para la creación de interfaces de usuario interactivas.
- **Node.js:** un entorno de ejecución de JavaScript que permite ejecutar código JavaScript en el lado del servidor.

Conclusión

En conclusión, Git se ha consolidado como una herramienta indispensable en el desarrollo de software gracias a su capacidad para gestionar proyectos de manera eficiente y facilitar la colaboración entre desarrolladores. A través de este cuestionario, hemos revisado desde conceptos básicos como la inicialización de un repositorio hasta la utilización de comandos esenciales y la gestión de ramas. Comprender y dominar estas herramientas permite a los desarrolladores trabajar de manera más organizada, mantener un historial de versiones detallado y colaborar eficazmente en proyectos de cualquier magnitud. Con Git, el desarrollo de software se vuelve más estructurado, seguro y adaptado a las necesidades dinámicas de la industria.