Nombres:

Marcos Jose Montero

Henríquez

Matricula:

2022-0105

Materia:

Programación III

Sección:

#10

Profesor:

Kelyn Tejada

Belliard

Fecha de

entrega:

Desarrolla el siguiente Cuestionario

1. ¿Qué es Git?

Es un sistema de control de versiones distribuido creado para manejar y controlar lo que pasa en proyectos de programación de manera eficiente. La función principal es permitir que varios desarrolladores puedan trabajar de manera simultánea en el mismo proyecto sin que sus cambios choquen entre si. Con Git, se puede tener un seguimiento de cada cambio realizado en el código, revertir a versiones anteriores si hace falta y gestionar distintas ramas del desarrollo. Esto no solamente sirve para mejorar la colaboración sino también para mejor la productividad del equipo y también tener el historial completo de las modificaciones realizadas para su revisión.

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

El git init lo utilizamos para crear un nuevo repositorio de Git en el directorio donde nos encontramos. Al ejecutar el git init se inicializa un repositorio vacio y se crea un carpeta oculta en el directorio llamado .git, la misma contiene todos los archivos que necesitaremos para llevar el seguimiento de las versiones. Ese seria el primer paso para convertir un directorio de algun proyecto estándar en un repositorio gestionado por Git, de esta manera permitir empezar a rastrear los cambios y configurar el control de versiones.

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

Hablando de git una rama es una version paralela de proyecto, esto permite a los desarrolladores el poder trabajar en funcionalidades o correcciones de

errores sin dañar o afectar la versión principal u original del mismo proyecto. Por defecto, todos los repositorios de git empiezan con una rama llamada main (anteriormente conocida como master). Las ramas se utilizan para desarrollar nuevas caracteristicas de forma independiente y una vez completadas y testeadas, se pueden fusionar con la rama principal. Todo esto nos permite mantener el código base funcionando mientras se realizan cambios experimentales o importantes en otras ramas.

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

Para determinar en que rama te encuentras, puedes utilizar el comando git branch. El comando muestra todas las ramas existentes asociadas con el repositorio y etiqueta con un asterisco (*) la rama activa. De la misma manera, puedes obtener información sobre la rama actual si usas el comando git status que le permite ver el estado del repositorio en general y confirma la rama.

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

Linus Torvalds, el mismo desarrollador que desarrolló el núcleo principal del sistema operativo Linux, también fue el responsable de crear Git en 2005. Torvalds se enfrentó a limitaciones y problemas con el sistema de control de versiones que se usaba en ese momento para desarrollar el kernel de Linux. Torvalds creó Git con un enfoque en la velocidad, la eficiencia y el trabajo distribuido.

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

git clone: Se utiliza para clonar un repositorio existente en un nuevo directorio local.

git add: Añade archivos al área de preparación (staging area) en preparación para

un commit.
git commit: Registra los cambios en el repositorio local. Cada commit tiene un mensaje asociado que describe los cambios realizados.
git push: Envía los commits locales a un repositorio remoto.
git pull: Actualiza el repositorio local con los cambios del repositorio remoto.
git merge: Combina cambios de diferentes ramas en una sola.
git status: Muestra el estado actual del repositorio, incluyendo cambios no confirmados y la rama activa.

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

- **Linux Kernel:** El repositorio del núcleo de Linux es uno de los más conocidos y utilizados, mantenido por miles de desarrolladores en todo el mundo.
- **Homebrew:** Un popular gestor de paquetes para macOS que permite instalar software usando comandos de terminal.
- **Bootstrap:** Un framework de código abierto para desarrollar con HTML, CSS y JavaScript.
- **TensorFlow:** Una biblioteca de código abierto para el aprendizaje automático y la inteligencia artificial, desarrollada por Google.
- **React:** Una biblioteca de JavaScript para construir interfaces de usuario, mantenida por Facebook.

8. **Link de repositorio**

https://github.com/MarcoJzz/ProjectGIT20220105