



Las Americas Institute of Technology

Nombres:

Misael Ramírez 2020-9910

Período:

Septiembre-diciembre

Materia:

Programación III

Profesor:

Kelyn Tejada

Práctica:

Tarea #3

Desarrollo el siguiente cuestionario

1. ¿Qué es Git?

Git es un sistema distribuido para gestionar versiones, diseñado para llevar un registro de los cambios realizados en proyectos de desarrollo de software. Facilita la colaboración entre equipos y permite organizar el trabajo de manera eficiente mediante un historial de revisiones.

2. ¿Para qué funciona el comando Git init?

El comando git init sirve para iniciar un repositorio en un directorio. Este proceso crea una carpeta llamada .git que contiene toda la información necesaria para que Git comience a rastrear los cambios del proyecto.

3. ¿Qué es una rama?

Una rama es una vía separada para el desarrollo de un proyecto. Su propósito es permitir que los programadores trabajen en tareas específicas como nuevas funciones o correcciones sin interferir con la rama principal.

4. ¿Cómo saber en cuál rama estoy?

Para identificar la rama activa, se puede usar el comando git branch, que muestra todas las ramas y marca la actual con un asterisco. También, el comando git status proporciona información sobre la rama en uso.

5. ¿Quién creó git?

Git fue desarrollado en 2005 por Linus Torvalds, el mismo creador de Linux, como una herramienta para organizar el trabajo colaborativo en el desarrollo del núcleo de este sistema operativo.

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

Algunos de los comandos esenciales son:

- **git init:** Crea un nuevo repositorio.
- **git clone:** Duplica un repositorio existente.
- **git add:** Agrega cambios al área de preparación.
- **git commit:** Guarda cambios en el historial del repositorio.
- **git push:** Sube cambios al repositorio remoto.
- **git pull:** Descarga y combina cambios del remoto.
- **git branch:** Administra ramas.
- **git checkout:** Cambia de una rama a otra.
- **git merge:** Combina ramas.

7. ¿Qué es git Flow?

Git Flow es un enfoque estructurado para gestionar ramas en proyectos de Git. Establece un flujo de trabajo con ramas específicas para distintas tareas como desarrollo, correcciones, preparación de lanzamientos y mantenimiento, facilitando la colaboración y organización del equipo.

8. ¿Qué es un trunk based development?

Trunk-Based Development es una estrategia en la que todos los desarrolladores trabajan directamente en una sola rama principal, llamada "trunk". Los cambios realizados son pequeños y se integran rápidamente, minimizando conflictos y permitiendo lanzamientos más continuos.

Link Repositorio **GitHub**: <https://github.com/Rubirosa/Tarea-3/tree/develop>