Presentacion
Nombre:
Franklin Javier Carpio Pepen
Matricula:
2023-0948
Materia:
Programacion 3
Profesor:
Kelyn Tejada Belliard
Fecha:
4/4/2025

Cuestionario sobre Git y Git Flow

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite que la administración y los archivos eficientes se rastreen. Está diseñado para facilitar a los colegas en los desarrolladores para que puedan trabajar en diferentes partes del proyecto sin interrupción. Su estructura de distribución significa que cada usuario tiene una copia completa del historial del proyecto, garantiza más seguridad y flexibilidad.

2. ¿Para qué sirve el comando git init?

El comando git init se utiliza para iniciar un nuevo repositorio de Git en una carpeta. Al aplicarlo, el GIT crea un directorio oculto. GIT, almacenará la información toda la información suficiente para monitorear la versión del proyecto. Este es el primer paso para trabajar con un git en un nuevo proyecto.

3. ¿Qué es una rama en Git?

Git es una versión paralela del código en el repositorio de ramas. Esto permite la versión principal de impactar la versión principal del proyecto sin desarrollar nuevas características, sin resolver o experimentar. Las ramas más comunes son las ramas primarias o primarias, porque se pueden hacer muchas ramas según sea necesario para ajustar el flujo de trabajo.

4. ¿Cómo saber en cuál rama estoy trabajando?

Para determinar qué trabajo de rama está trabajando, puede usar el siguiente comando:

- Git Branch = Star (*) muestra una lista de ramas y marcadores con ramas.
- Git Status = Ver información sobre la rama actual y la ubicación del archivo en el repositorio.
- Git Parridge-Abbabe-Reef Head = Solo el nombre actual de la rama (para secuencias de comandos o automatización).

5. ¿Quién creó Git?

Git fue creado por Linus Torvalds en 2005, el mismo desarrollador que inició el proyecto del sistema operativo Linux. La razón principal para su desarrollo fue la necesidad de un sistema de control de versiones eficiente y distribuido para gestionar el código fuente del kernel de Linux.

6. ¿Cuáles son los comandos esenciales de Git?

Algunos de los comandos esenciales de Git son:

- git init = Inicializa un repositorio.
- git clone [URL] = Clona un repositorio remoto en la máquina local.
- git status = Muestra el estado actual del repositorio.
- git add [archivo] = Añade un archivo al área de preparación (staging).
- git commit -m "mensaje" = Guarda los cambios en el historial del repositorio.
- git branch = Muestra las ramas disponibles.
- git checkout [rama] = Cambia a otra rama.
- git merge [rama] = Fusiona una rama con la actual.
- git push = Sube los cambios al repositorio remoto.
- git pull = Descarga y aplica cambios desde el repositorio remoto.

7. ¿Qué es Git Flow?

Git Flow es un modelo de flujo de trabajo basado en ramas que facilita la gestión del desarrollo en proyectos de software. Define reglas y convenciones sobre cómo deben usarse las ramas en un proyecto. Las principales ramas en Git Flow son:

- main: Contiene el código estable listo para producción.
- develop: Se usa para el desarrollo y contiene los cambios en progreso.
- feature: Se crean a partir de develop para desarrollar nuevas funciones.
- release: Se usan para preparar una nueva versión antes de integrarla en main.
- hotfix: Se crean desde main para corregir errores urgentes y luego se fusionan en develop y main.

Git Flow ayuda a mantener un flujo de trabajo estructurado y organizado, especialmente en equipos grandes.

8. ¿Qué es el desarrollo basado en trunk (Trunk Based Development)?

El desarrollo basado en trunk (Trunk Based Development) es una estrategia en la que los desarrolladores trabajan directamente en la rama principal (trunk o main) en lugar de utilizar múltiples ramas a largo plazo. En este modelo, las ramas son de corta duración y se fusionan rápidamente en la rama principal para evitar problemas de integración.

Este enfoque favorece la entrega continua y la integración frecuente del código, reduciendo la acumulación de cambios grandes y minimizando los conflictos. Es utilizado en entornos donde se busca rapidez y estabilidad en la entrega de software.