



PRESENTACIÓN



Nombre: Diego Olivero Matos

Matrícula: 2021-1803

Carrera: Desarrollo de Software

Asignatura: Programación III

Profesor/a: Kelyn Tejada

Tipo: Asignación

Tema: Herramientas de administración de fuentes

Fecha: 31/03/2023

- DESARROLLA EL SIGUIENTE CUESTIONARIO

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido y gratuito, diseñado para ayudar en el seguimiento y control de cambios en los archivos de un proyecto. Fue creado por Linus Torvalds en el año 2005 y se ha convertido en uno de los sistemas de control de versiones más populares en la actualidad.

2. ¿Para qué funciona el comando "Git init"?

El comando "git init" se utiliza para inicializar un nuevo repositorio de Git en un directorio vacío o en uno existente que no sea un repositorio de Git.

Cuando se ejecuta "git init", Git crea un nuevo subdirectorio ".git" en el directorio actual. Este subdirectorio contiene todos los archivos necesarios para el control de versiones, como los metadatos del repositorio y la base de datos de objetos Git.

Una vez que se ha inicializado el repositorio, puedes empezar a agregar archivos al repositorio, hacer confirmaciones (commits), ramificar (branch) el código y trabajar en diferentes características o experimentos en paralelo, etc.

3. ¿Qué es una rama?

En Git, una rama es una línea de desarrollo independiente que permite a los desarrolladores trabajar en diferentes versiones del mismo código al mismo tiempo. Cada rama puede contener un conjunto diferente de cambios y se puede utilizar para experimentar con nuevas características, corregir errores o implementar mejoras en el código existente.

4. ¿Cómo saber es que rama estoy?

Para saber en qué rama estás actualmente, puedes utilizar el comando "git branch" en la línea de comandos.

Al escribir "git branch" en la terminal, Git mostrará una lista de todas las ramas en el repositorio y resaltará la rama actual con un asterisco "*".

5. ¿Quién creo Git?

Git fue creado por el desarrollador de software finlandés Linus Torvalds en 2005. Torvalds es conocido principalmente por ser el creador del kernel de Linux, el cual es un sistema operativo de código abierto muy popular.

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

Algunos de los comandos más esenciales de Git son:

1. **git init:** Inicializa un nuevo repositorio de Git en un directorio vacío o en uno existente que no sea un repositorio de Git.
2. **git clone:** Crea una copia local de un repositorio remoto en tu máquina.
3. **git add:** Agrega archivos al área de preparación (staging) para la confirmación (commit).
4. **git commit:** Confirma los cambios realizados en el área de preparación y crea un nuevo commit con un mensaje descriptivo.
5. **git status:** Muestra el estado actual del repositorio, incluyendo los cambios pendientes, los archivos no seguidos, los archivos modificados y los archivos en el área de preparación.
6. **git push:** Envía los cambios confirmados a un repositorio remoto.
7. **git pull:** Descarga los cambios realizados en un repositorio remoto y los fusiona con tu rama local.
8. **git branch:** Muestra una lista de todas las ramas en el repositorio y resalta la rama actual.
9. **git checkout:** Cambia a una rama diferente o a una versión anterior de un archivo.
10. **git merge:** Fusiona los cambios de una rama a otra.

7. ¿Qué es Git Flow?

GitFlow es un modelo de flujo de trabajo para Git, que se basa en una serie de prácticas recomendadas para trabajar con ramas en un repositorio de Git. Fue desarrollado por Vincent Driessen en 2010 y se ha convertido en una metodología popular entre los desarrolladores de software.

GitFlow proporciona un conjunto estructurado de ramas y una metodología clara para su uso, lo que ayuda a simplificar y organizar el proceso de desarrollo de software. El modelo utiliza dos ramas principales: "master" y "develop", que se utilizan para controlar la versión estable del software y el desarrollo activo del mismo, respectivamente.

8. ¿Qué es Trunk Based Development?

Trunk Based Development (TBD) es una metodología de desarrollo de software que se basa en trabajar directamente sobre la rama principal del repositorio de código (llamada "trunk" en inglés). A diferencia de otros modelos de flujo de trabajo, como GitFlow, que utilizan ramas separadas para cada característica o cambio, TBD promueve el trabajo en una sola rama, lo que hace que la integración sea más frecuente y los cambios se incorporen más rápidamente al producto final.