



Nombre

Leonardo Felipe

Apellido

Pérez Batista

Matrícula

2022-0910

Materia

Programación III

Asignación

Tarea 3 del módulo 5

A) Desarrolla el siguiente cuestionario

1- ¿Que es Git?

Git es un sistema de control de versiones distribuido que permite a los desarrolladores monitorear los cambios en sus proyectos de software. Es una herramienta que ayuda a los equipos de desarrollo a trabajar juntos en el mismo proyecto, fusionar sus cambios y resolver conflictos de manera eficiente. Git también es una herramienta vital para el desarrollo de software porque permite revertir cambios no deseados y explorar versiones anteriores del código.

2- ¿Para que funciona el comando Git init?

Para inicializar un nuevo repositorio de Git en un directorio específico, se utiliza el comando "Git init". Git crea una carpeta oculta llamada ".git" en el directorio del proyecto cuando ejecuta este comando en un proyecto. Esta carpeta contiene toda la información necesaria para que Git controle el historial de versiones del proyecto y rastree los cambios en los archivos. Después de ejecutar "Git init", el directorio se transforma en un repositorio de Git, lo que le permite comenzar a monitorear los cambios en los archivos y realizar operaciones de control de versiones.

3- ¿Que es una rama?

Una rama es una línea de desarrollo independiente que se origina a partir de una versión principal (normalmente la rama "master") en el contexto de Git. Las ramas permiten a los desarrolladores desarrollar nuevas características, resolver problemas o realizar experimentos sin afectar directamente la rama principal del proyecto. Cada rama tiene su propio conjunto de cambios; cuando se completa el desarrollo o cuando los cambios se consideran estables, se puede fusionar con otras ramas. Las ramas son una parte importante de la metodología de control de versiones y facilitan el desarrollo colaborativo y paralelo en un proyecto.

4- ¿Como saber es que rama estoy?

Puede usar el comando "git branch" o "git status" en su terminal o línea de comandos para saber en qué rama te encuentras en Git. El comando "git branch" muestra todas las ramas en el repositorio, con la rama actual destacada con un asterisco (*). El comando "git status", por otro lado, te proporcionará más información sobre el estado actual de tu repositorio, incluida su rama actual y si hay cambios pendientes de confirmación.

5- ¿Quién creo git?

En 2005, Linus Torvalds creó Git. Linus es un programador finlandés conocido principalmente por ser el creador y coordinador principal del kernel de Linux, un sistema operativo de código abierto muy popular en todo el mundo. La creación de Git fue una solución a las limitaciones de otros sistemas de control de versiones existentes en ese momento y se convirtió en una herramienta muy popular y utilizada por la comunidad de desarrollo de software.

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

"git clone": se usa para copiar o clonar un repositorio actual en su máquina local.

"git add": para agregar los cambios que hiciste en tus archivos al área de preparación.

"git commit": crea una nueva confirmación o versión del proyecto con los cambios que has agregado al área de preparación.

"git push": actualiza la versión del proyecto en el servidor enviando confirmaciones locales al repositorio remoto.

"git pull": descarga y fusiona los cambios del repositorio remoto en su rama local.

"git branch": muestra las ramas existentes y destaca la rama actual.

"git merge": fusiona una rama con la rama actual.

"git status": muestra el estado actual del repositorio, incluidos los cambios sin confirmar y las ramas activas.

7- ¿Que es git Flow?

Es un flujo de trabajo que se basa en la creación de ramas específicas para diferentes etapas del desarrollo de un proyecto y se utiliza para el control de versiones con Git. Fue propuesto por Vincent Driessen y se ha convertido en una técnica popular para el desarrollo de software en equipos grandes y complejos. Las dos ramas principales de Git Flow son "master" y "develop". Se crean ramas adicionales, como "feature" (para agregar nuevas características), "release" (para preparar versiones estables) y "hotfix" (para solucionar problemas en producción). Cada sección tiene una función específica y el flujo de trabajo está diseñado para facilitar la colaboración y el lanzamiento ordenado del software.

8- ¿Que es trunk based development ?

Es un enfoque de desarrollo de software que fomenta el trabajo en equipo continuo y la implementación rápida de cambios. En este modelo, los desarrolladores no crean ramas separadas para cada característica o corrección; en cambio, envían sus cambios directamente a la rama principal del repositorio, que generalmente se conoce como "tronco" o "maestro". El objetivo es reducir la duración de las ramas y las diferencias entre las características en desarrollo y el código principal. Esto facilita la detección temprana de conflictos e problemas de integración, lo que resulta en una entrega más rápida y un proyecto más estable.

Bibliografía/Fuentes

<https://www.atlassian.com/es/git/tutorials/what-is-git>

<https://platzi.com/clases/1557-git-github/20215-que-es-git/>

[https://openwebinars.net/blog/trunk-based-development-vs-git-flow-cual-elegir/#:~:text=comandos%20en%20Git.-,Qu%C3%A9%20es%20Trunk%20Based%20Development,o%20rama%20principal%20\(main\).](https://openwebinars.net/blog/trunk-based-development-vs-git-flow-cual-elegir/#:~:text=comandos%20en%20Git.-,Qu%C3%A9%20es%20Trunk%20Based%20Development,o%20rama%20principal%20(main).)

<https://dev.to/marianocodes/por-que-trunk-based-development-i5n>