



Nombre: Edward Paul Nuñez Ventura

Matricula: 2022-0310

Asignatura: Programación III

Maestro: Kelyn Tejada Belliard

Asignación: Tarea 3

Fecha: 3/4/2024

Desarrolla el siguiente Cuestionario

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido, el cual permite registrar los cambios que el usuario realiza en sus archivos y también le permite al usuario volver a versiones anteriores por si algo sale mal. Este sistema fue diseñado por Linus Torvalds con el objetivo de garantizar la eficiencia y confiabilidad del mantenimiento de versiones de aplicaciones que tienen un gran número de archivos de código fuente.

Las bases de este sistema son el rendimiento, seguridad y flexibilidad. El sistema de trabajo con el que cuenta git son las ramas, que estas últimas la función que desempeñan es poder realizar cambios o actualizaciones, esto ocurre mediante la creación de líneas de desarrollo independientes las cuales provienen de la línea principal, esto permite que si en las actualizaciones que realicen los desarrolladores ocurre algún error, esto no afecte al proyecto principal.

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

El comando de git init se utiliza para crear un nuevo repositorio en un directorio existente o se crea uno nuevo si no existe. Cuando el comando git init se ejecuta, git automáticamente crea una subcarpeta oculta llamada .git en el directorio principal del proyecto. En esta carpeta se encuentran todos los archivos necesarios para el funcionamiento interno de git.

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

Las ramas en git representan líneas de desarrollo independiente, cuando se crea un repositorio por primera vez, se crea una rama predeterminada llamada main o master. Esta rama representa la línea principal de desarrollo, de esta rama principal se pueden crear otras ramas, en las cuales se pueden realizar cambios o correcciones de errores de forma aislada sin afectar a la rama principal, y después de que se hagan algunas pruebas se pueden fusionar los cambios.

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

Esto se puede determinar utilizando el comando '**git branch**'. Este comando lo que hará es mostrar todas las ramas locales y resaltará con un asterisco (*) la rama en la que nos encontramos actualmente.

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

Git fue creado por Linus Torvalds, que este es el creador del kernel de Linux. Linus desarrolló git en 2005 esto fue con el objetivo de crear una solución para gestionar el desarrollo de Linux de manera más eficiente. De hecho esta herramienta solucionó el inconveniente principal que tenían las herramientas de control de

versiones de ese entonces que se encontraban muy limitadas, a diferencia de git que permitia un desarrollo mas eficiente y sin las limitaciones que tenían las herramientas de antes.

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

Entre los comandos esenciales de git se encuentran:

git init: Se utiliza para inicializar un nuevo repositorio Git en un directorio existente.

git clone: Se utiliza para clonar un repositorio existente desde una ubicación remota (por ejemplo, un repositorio en GitHub) hacia tu sistema local.

git add: Agrega cambios en el directorio de trabajo al área de preparación (staging area) para que puedan ser incluidos en el próximo commit.

git commit: Registra los cambios confirmados en el repositorio. Es necesario proporcionar un mensaje descriptivo para cada commit.

git status: Muestra el estado actual del directorio de trabajo y del área de preparación, incluyendo qué archivos han sido modificados, agregados o eliminados.

git push: Sube los commits locales a un repositorio remoto, actualizando así la rama remota con los cambios locales.

git pull: Descarga los cambios desde un repositorio remoto y los integra automáticamente en tu rama local.

git branch: Lista todas las ramas presentes en el repositorio y te permite crear, eliminar o cambiar entre ramas.

git checkout: Se utiliza para cambiar entre diferentes ramas o para restaurar archivos a un estado específico de confirmación (commit).

git merge: Fusiona los cambios de una rama a otra. Por ejemplo, puedes fusionar una rama de desarrollo en la rama principal.

git log: Muestra el historial de commits del repositorio, incluyendo detalles como el autor, la fecha y el mensaje del commit.

git remote: Gestiona las conexiones remotas con otros repositorios. Puedes agregar, eliminar o listar conexiones remotas.

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

Algunos de los repositorios más de git más utilizados actualmente son:

GitHub: Este repositorio es de los mas utilizados por los desarrolladores, debido a la capacidad de poder almacenar diferentes versiones de proyectos de una manera simple y eficaz. Esto es una de las principales características que posee GitHub además de contar con una navegación cómoda, permitiendo poder localizar fácilmente las líneas de código deseadas independientemente del repositorio en el que estemos buscando el problema.

GitLab: Teniendo algunas similitudes con GitHub, GitLab es una plataforma de gestión de repositorios de git que ofrece alojamiento de proyectos, seguimiento de problemas, integración continua, etc. Algo que destaca en GitLab es que ofrece la posibilidad instalar y ejecutar repositorios en servidores en la nube, esto permitiendo al usuario no tener que ocupar espacio en su equipo de trabajo con la instalación de ningún software.

Bitbucket: Es otra plataforma de alojamiento de repositorios de Git que ofrece tanto repositorios públicos como privados. También este repositorio al ser propiedad de Atlassian permite la integración de otras herramientas de desarrollo como Jira y Trello. Este repositorio se destaca principalmente por la facilidad de integración con diferentes tipos de repositorios.