



Las Americas Institute of Technology

Nombre: Jeremy Gabriel García Méndez

Matrícula: 2021-0055

Materia: Programación III

Profesor: Kelyn Tejada Belliard

Tema: Tarea III (Individual)

Desarrolla el siguiente Cuestionario

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido que proporciona una plataforma robusta para rastrear y gestionar cambios en el código fuente a lo largo del tiempo. A diferencia de los sistemas de control de versiones centralizados, Git no depende de un servidor central y permite que cada usuario tenga una copia completa del historial del proyecto en su máquina local. Esto facilita la colaboración en equipos distribuidos y permite a los desarrolladores trabajar de manera independiente en sus propias ramas antes de fusionar sus cambios en la rama principal. Git utiliza una estructura de datos eficiente y un algoritmo de hash para garantizar la integridad de la información y proporcionar un historial de cambios sólido y confiable.

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

El comando git init desencadena el proceso de creación de un nuevo repositorio Git en un directorio local. Al ejecutar este comando, Git inicializa la carpeta oculta “.git” en el directorio, que contiene toda la información necesaria para el control de versiones, como la base de datos de objetos, las configuraciones y las referencias a las diferentes ramas. Este proceso establece el contexto de seguimiento de versiones en el directorio seleccionado, permitiendo a los desarrolladores comenzar a rastrear cambios, crear ramas y realizar commits. El comando git init es esencial al iniciar un nuevo proyecto o al convertir un directorio existente en un repositorio Git.

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

En el contexto de Git, una rama es esencialmente una línea de desarrollo independiente que permite a los desarrolladores trabajar en características específicas o solucionar problemas sin interferir directamente con el código en la rama principal del proyecto. Las ramas en Git son livianas y fáciles de crear, lo que facilita la implementación de nuevas funcionalidades de manera aislada. Una vez que se completa el trabajo en una rama, los cambios pueden fusionarse de nuevo en la rama principal. Este enfoque facilita el desarrollo en paralelo, ya que varios equipos o desarrolladores individuales pueden trabajar en diferentes aspectos del proyecto al mismo tiempo, sin afectar la estabilidad de la rama principal.

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

Para determinar en qué rama te encuentras, puedes usar el comando `git branch`, que lista todas las ramas presentes en tu repositorio local. La rama actual estará resaltada con un asterisco (). Alternativamente, puedes emplear el comando `git status`, que no solo muestra la rama actual, sino también el estado de tus archivos y la relación con el repositorio remoto. Ambos comandos proporcionan información valiosa para entender la ubicación y el estado de tu trabajo dentro del contexto de versionado de Git.*

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

Git fue creado por Linus Torvalds en el año 2005, el mismo desarrollador que creó el kernel de Linux. Torvalds desarrolló Git para gestionar el proyecto del kernel de Linux y para abordar las limitaciones que experimentó con otros sistemas de control de versiones existentes. Su objetivo era crear un sistema eficiente, rápido y robusto que permitiera el desarrollo colaborativo en un entorno distribuido.

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

Algunos comandos esenciales de Git incluyen:

- *git init: Inicializa un nuevo repositorio.*
- *git clone: Clona un repositorio existente.*
- *git add: Agrega cambios al área de preparación.*
- *git commit: Guarda los cambios en el repositorio.*
- *git pull: Obtiene cambios del repositorio remoto.*
- *git push: Envía cambios al repositorio remoto.*

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

Algunos de los repositorios más usados y útiles de GitHub son:

-Construya su propio X

<https://github.com/danistefanovic/build-your-own-x>

-Algoritmos JavaScript

<https://github.com/trekhleb/javascript-algorithms>

-Ossu

<https://github.com/ossu/computer-science>

-30 Segundos de Código

<https://github.com/30-seconds/30-seconds-of-code>

LINK A REPOSITORIO GITHUB:

https://github.com/Gabriel0825/tarea03_prog3