Carlos Salazar | 2021-1932

ITLA  Programación III

Tarea #3

**INSTITUTO TECNOLOGICO DE LAS AMERICAS (ITLA)**

Logotipo

Descripción generada automáticamente

**Materia**

Programación III

**Maestro**

Kelyn Tejeda Belliard

**Integrantes del Grupo #1**

Carlos Rafael Salazar Delgado | 2021-1932

**Fecha de Entrega**

03 de Abril

**Tema**

Tarea 3

Desarrolla el siguiente Cuestionario

1. **¿Qué es Git?**

Git es un sistema de control de versiones distribuido, diseñado para rastrear cambios en archivos y coordinar el trabajo de múltiples personas en proyectos de desarrollo de software. Fue creado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux y desde entonces ha ganado una amplia adopción en la industria del software debido a su eficiencia y flexibilidad.

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

El comando git init en Git se utiliza para iniciar un nuevo repositorio Git en un directorio existente o vacío. Su propósito principal es establecer un nuevo repositorio Git en el directorio actual, lo que permite a Git comenzar a rastrear cambios en los archivos dentro de ese directorio.

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

En Git, una rama (branch) es una línea independiente de desarrollo que se deriva del estado actual del repositorio. Cada repositorio Git comienza con una rama principal predeterminada, generalmente llamada "master" (aunque ahora es comúnmente referida como "main" debido a razones de inclusión y diversidad). Las ramas se utilizan para trabajar en nuevas características, solucionar problemas (bugs), realizar experimentos u otras tareas de desarrollo sin afectar directamente la rama principal.

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

Para determinar en qué rama estás actualmente en Git, se utiliza el siguiente comando:

*git branch*

Este comando mostrará todas las ramas en el repositorio local y resaltará la rama actual con un asterisco (\*). La rama actual es aquella en la que estás trabajando actualmente.

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

Git fue creado por Linus Torvalds, el mismo desarrollador que creó el kernel del sistema operativo Linux. Torvalds comenzó a trabajar en Git en 2005 como una respuesta a las deficiencias que percibía en otros sistemas de control de versiones disponibles en ese momento. La primera versión estable de Git fue lanzada en abril de 2005. Desde entonces, Git ha experimentado un crecimiento significativo en popularidad y se ha convertido en uno de los sistemas de control de versiones más ampliamente utilizados en la industria del desarrollo de software.

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

**git init:** Se utiliza para iniciar un nuevo repositorio Git en un directorio existente o vacío.

**git clone:** Se utiliza para clonar un repositorio Git existente desde una ubicación remota (como GitHub, GitLab, etc.) hacia tu máquina local.

**git add:** Agrega cambios en los archivos al área de preparación (staging area) para ser incluidos en el próximo commit.

**git commit:** Registra los cambios realizados en los archivos en el repositorio. Se puede acompañar de un mensaje descriptivo que explique los cambios realizados.

**git push:** Sube los cambios locales confirmados (commits) al repositorio remoto.

**git pull:** Obtiene los cambios desde el repositorio remoto y los fusiona con tu rama local.

**git status:** Muestra el estado actual del repositorio, incluyendo los archivos modificados, agregados al área de preparación, y los pendientes de confirmar.

**git log:** Muestra el historial de commits del repositorio, incluyendo información sobre quién realizó cada commit, cuándo se realizó y el mensaje asociado.

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

**git checkout:** Se utiliza para cambiar entre ramas o para restaurar archivos a una versión específica del repositorio.

**git merge:** Fusiona cambios de una rama en otra. Se utiliza para incorporar el trabajo realizado en una rama a otra, como fusionar una rama de desarrollo con una rama principal.

**git stash:** Guarda temporalmente los cambios en un lugar llamado "stash" para poder trabajar en otra cosa antes de volver a aplicar esos cambios más tarde.

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

**GitHub:** GitHub es posiblemente el repositorio de Git más popular y ampliamente utilizado en el mundo. Ofrece alojamiento de repositorios remotos gratuitos y de pago, así como características de colaboración y seguimiento de problemas. Es utilizado por millones de desarrolladores y empresas en todo el mundo.

**GitLab:** GitLab es una plataforma de desarrollo de software integral que incluye repositorios de Git, seguimiento de problemas, integración continua y mucho más. Ofrece tanto una versión en la nube como una versión de autohospedaje, lo que lo convierte en una opción popular para empresas que desean mantener el control total sobre su infraestructura de desarrollo.

**Bitbucket:** Bitbucket es un servicio de alojamiento de repositorios de Git y Mercurial propiedad de Atlassian. Ofrece alojamiento gratuito y de pago para repositorios privados y públicos, y está integrado con otras herramientas de desarrollo de Atlassian como Jira y Bamboo.

**GitKraken:** GitKraken es un cliente de Git con una interfaz de usuario visual y fácil de usar. Además de permitirte interactuar con tus repositorios de Git, también ofrece características como integración con GitHub y Bitbucket, y la capacidad de resolver conflictos de fusión visualmente.

**SourceForge:** SourceForge ha sido un popular repositorio de código abierto durante muchos años. Aunque ha perdido algo de popularidad en comparación con GitHub y otros, todavía alberga una gran cantidad de proyectos de software de código abierto.