ESTUDIA EN EL INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS (ITLA)

PRESENTACIÓN

ASIGNATURA:

Programación 3

ESTUDIANTE:

Luis Elier Pujols Rosa

MATRÍCULA:

2023-1667

DOCENTE:

Kelyn Tejada Belliard

RECINTO:

Itla Santo Domingo Norte, Mamá Tingo

Indice

¿Qué es Git?

Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan la copia del repositorio con la del servidor. Este paradigma es distinto del control de versiones centralizado, donde los clientes deben sincronizar el código con un servidor antes de crear nuevas versiones.

¿Qué es Git?

El comando git init crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git, o para inicializar un nuevo repositorio vacío. La mayoría de los demás comandos de Git no se encuentran disponibles fuera de un repositorio inicializado, por lo que este suele ser el primer comando que se ejecuta en un proyecto nuevo.

Al ejecutar git init, se crea un subdirectorio de .git en el directorio de trabajo actual, que contiene todos los metadatos de Git necesarios para el nuevo repositorio. Estos metadatos incluyen subdirectorios de objetos, referencias y archivos de plantilla. También se genera un archivo HEAD que apunta a la confirmación actualmente extraída.

¿Qué es una rama en Git?

Una rama Git es simplemente un apuntador móvil apuntando a una de esas confirmaciones. La rama por defecto de Git es la rama master/main. Con la primera confirmación de cambios que realicemos, se creará esta rama principal master/main apuntando a dicha confirmación.

¿Cómo saber en que rama estoy trabajando?

Comando git Branch: Este comando muestra una lista de las ramas en tu repositorio local. La rama actual estará marcada con un asterisco (*).

¿Quién creo Git?

Git fue creado por Linus Torvalds en 2005. Torvalds es más conocido por ser el creador del kernel de Linux.

¿Cuáles son los comandos esenciales de Git?

- **git add**: Mueve los cambios del directorio de trabajo al área de ensayo
- **git branch**: Lista, crea o elimina ramas
- **git checkout**: Cambiar entre ramas o restaurar archivos
- **git clone**: Descarga el código fuente existente desde un repositorio remoto
- **git diff**: Muestra las diferencias entre las confirmaciones
- **git fetch**: Obtiene cambios de un repositorio remoto sin fusionar
- **git merge**: Combina ramas para crear un historial único
- **git pull**: Obtiene y fusiona cambios de un repositorio remoto
- **git rm**: Elimina archivos del área de ensayo y el directorio de trabajo
- **git tag**: Enumera, crea o muestra información sobre etiquetas del repositorio local

¿Qué es GitFlow?

GitFlow es un modelo de flujo de trabajo de Git que define un conjunto estricto de reglas para la gestión de ramas en un repositorio Git. Su objetivo principal es facilitar el desarrollo colaborativo y la gestión de versiones en proyectos de software.

¿ Qué es el desarrollo basado en trunk (Trunk Based Development)?

El desarrollo basado en troncos es una práctica de control de versiones en la que los desarrolladores fusionan pequeñas actualizaciones de forma frecuente en un "tronco" o rama principal. Dado que esta práctica simplifica las fases de fusión e integración, ayuda a lograr la CI y la CD y, al mismo tiempo, aumenta la entrega de software y el rendimiento de la organización.

En los primeros tiempos del desarrollo de software, los programadores no disfrutaban del lujo de los sistemas de control de versiones modernos. Más bien, desarrollaban dos versiones del software a la vez para poder realizar un seguimiento de los cambios y revertirlos si era necesario. Con el tiempo, este proceso resultó ser muy laborioso, costoso e ineficiente.

Bibliografía

https://www.atlassian.com/

https://git-scm.com/doc

https://learn.microsoft.com/es-es/devops