

# **Presentación**

## **MATERIA**

Programación 3

## **MAESTRO**

Kelyn Tejada

## **Nombre**

Alvaro Minyety

## **MATRICULA**

2021-0194

¿Qué es Git?

Git es como un súper gestor de archivos para desarrolladores. Te ayuda a llevar un control de todas las versiones de tu código. Piensa en él como un historial detallado que te permite ver quién hizo qué cambios y cuándo. Es genial para colaborar con otros, porque cada uno puede trabajar en diferentes partes del proyecto sin pisarse los talones.

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

Cuando usas `git init`, le estás diciendo a Git que empiece a rastrear los cambios en tu proyecto. Básicamente, este comando convierte tu carpeta en un repositorio de Git, creando una especie de "caja fuerte" donde se guarda toda la información sobre los cambios que hagas en el código.

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

Imagina que estás trabajando en una nueva característica para tu proyecto. En lugar de hacer todos esos cambios directamente en la versión principal del proyecto, creas una rama. Es como una copia del proyecto donde puedes hacer experimentos sin afectar el trabajo principal. Cuando terminas y todo está bien, puedes combinar esa rama con el proyecto principal para integrar los cambios.

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

Si quieres saber en qué rama estás, simplemente ejecuta `git branch`. Esto te mostrará todas las ramas disponibles y marcará con un asterisco (\*) la rama en la que estás trabajando en ese momento.

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

Git fue creado por Linus Torvalds en 2005. Sí, ¡el mismo tipo que hizo el sistema operativo Linux! Lo creó porque necesitaba una herramienta para gestionar el desarrollo del kernel de Linux de manera más eficiente.

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

Aquí tienes algunos comandos básicos que te serán muy útiles:

`git add`: Añade los archivos que has modificado a una "lista de espera" para que luego puedas confirmar esos cambios.

`git commit`: Guarda esos cambios en el historial del proyecto con un mensaje que describe qué hiciste.

`git push`: Envía esos cambios a un repositorio remoto para que otros puedan verlos.

`git pull`: Trae los cambios más recientes del repositorio remoto y los combina con tu trabajo local.

`git status`: Muestra qué archivos has cambiado y en qué estado está tu repositorio.

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

Claro, aquí tienes algunos de los más populares:

**GitHub**: Es súper conocido y muy utilizado, especialmente para proyectos de código abierto.

**GitLab**: Además de alojar repositorios, ofrece herramientas para integración continua y gestión de proyectos.

**Bitbucket**: También ofrece alojamiento de repositorios y se integra bien con otras herramientas de Atlassian.