

# Instituto Tecnológico de Las Américas



*– Cuestionario sobre Git.*

<b>Docente:</b>	Kelyn Tejada
<b>Asignatura:</b>	Programación 3
<b>Alumna:</b>	Yerelin Vanessa Rosario Taveras
<b>Matricula:</b>	2023-1751

muy usado en trabajos en equipo para la programación porque podemos evitar sobrescribir el trabajo de los demás.

## **2. ¿Para qué sirve el comando git init?**

Se usa para inicializar un nuevo repositorio Git en una carpeta, después de ejecutarlo este empieza a detectar los cambios en los archivos de esa carpeta y crea una carpeta oculta llamada .git, donde se almacenan los datos en el repositorio.

## **3. ¿Qué es una rama en Git?**

Una rama en Git es una línea de desarrollo independiente dentro del mismo proyecto que permite trabajar nuevas versiones sin afectar la versión principal.

## **4. ¿Cómo saber en cuál rama estoy trabajando?**

En Visual Studio Code se puede visualizar el nombre de la rama en la parte inferior del archivo, también se puede usar git status, que este te dice en que rama estas.

## **5. ¿Quién creó Git?**

Lo creó Linus Torvads en 2005, el mismo que creó el sistema operativo Linux.

## **6. ¿Cuáles son los comandos esenciales de Git?**

Git init, inicia el repositorio.

Git add, agrega archivos al área de preparación.

Git status, muestra el estado actual del repositorio.

Git branch, gestiona las ramas.

Git pull, descarga y fusiona cambios del repositorio remoto.

Git push, sube los cambios al repositorio remoto.

Git merge, fusiona las ramas

Git checkout, cambia de rama

Git commit guarda los cambios en el historial.

## **7. ¿Qué es Git Flow?**

Esta es una forma organizada de trabajar con Git, esta usa varias ramas con propósitos específicos, por ejemplo:

“Main” contiene la versión estable del proyecto

“develop” es donde se trabaja en nuevas funciones antes de pasarlas a main

Feature/\* es para desarrollar nuevas características. Etc.

#### **8. ¿Qué es el desarrollo basado en trunk (Trunk Based Development)?**

Es otra forma de trabajar con git per en lugar de trabajar con muchas ramas, casi todo el trabajo ocurre en la principal “main”. Esto es útil para equipos grandes porque evita problemas de integración con muchas ramas.