

# PROGRAMACION III

Angel Alberto Ortiz Fernandez  
INSTITUTO TECNOLOGICO DE LAS AMERICAS



*Las Americas Institute of Technology*

Asignatura:

Programación III

Maestro:

Kelyn Tejada

Estudiante:

Angel Alberto Ortiz Fernandez

Matricula:

2020-9569

Sección: #2

## **Asignación individual**

**Valor 5 Puntos**

**Trabajar de forma individual**

**Desarrolla el siguiente Cuestionario**

### **1-Que 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 un proyecto de desarrollo de software. Permite mantener un historial completo de cambios, fusionar contribuciones de diferentes desarrolladores y facilitar la colaboración en equipo.

### **2-Para que funciona el comando Git init?**

El comando "git init" se utiliza para inicializar un nuevo repositorio de Git en un directorio local. Cuando ejecutas este comando en una carpeta, Git crea una estructura de datos necesaria para el control de versiones y comienza a rastrear los cambios en los archivos presentes en ese directorio.

### **3-Que es una rama?**

Una rama en Git es una línea de desarrollo independiente que permite a los desarrolladores trabajar en nuevas funcionalidades o solucionar problemas sin afectar la rama principal (comúnmente llamada "master" o "main"). Las ramas se utilizan para aislar el trabajo y, posteriormente, pueden fusionarse con la rama principal una vez que se han probado y revisado los cambios.

### **4-Como saber es que rama estoy?**

Para conocer la rama en la que te encuentras actualmente, puedes utilizar el comando "git branch" o "git status". Ambos comandos te mostrarán la rama actualmente activa y destacarán las diferencias entre esa rama y otras existentes.

## 5-Quien creo git?

Git fue creado por Linus Torvalds, el mismo desarrollador que creó el kernel de Linux. Lo creó en 2005 como respuesta a la necesidad de un sistema de control de versiones eficiente y robusto para el desarrollo del kernel de Linux y otros proyectos.

## 6-Cuales son los comandos más esenciales de Git?

- "git init": Inicializa un nuevo repositorio local.
- "git add": Agrega cambios de archivos al área de preparación (staging).
- "git commit": Crea un nuevo commit con los cambios preparados.
- "git push": Envía los cambios locales a un repositorio remoto.
- "git pull": Obtiene los cambios del repositorio remoto y los fusiona con la rama actual.
- "git branch": Muestra las ramas disponibles y destaca la rama actual.
- "git merge": Fusiona una rama con la rama actual.
- "git clone": Clona un repositorio remoto en un nuevo directorio local.
- "git status": Muestra el estado actual del repositorio y los archivos modificados.
- "git init": Inicializa un nuevo repositorio local.
- "git add": Agrega cambios de archivos al área de preparación (staging).
- "git commit": Crea un nuevo commit con los cambios preparados.
- "git push": Envía los cambios locales a un repositorio remoto.

- "git pull": Obtiene los cambios del repositorio remoto y los fusiona con la rama actual.
- "git branch": Muestra las ramas disponibles y destaca la rama actual.
- "git merge": Fusiona una rama con la rama actual.
- "git clone": Clona un repositorio remoto en un nuevo directorio local.
- "git status": Muestra el estado actual del repositorio y los archivos modificados.
- "Git checkout": Usamos git checkout principalmente para cambiar de una rama a otra.

## **7-Que es git Flow?**

es un conjunto de reglas y convenciones para el uso de Git que ayuda a gestionar y organizar el flujo de trabajo en proyectos de desarrollo de software. Define una estructura de ramas específica y define cómo se deben combinar y desplegar las diferentes partes del proyecto.

## **8-Que es trunk based development?**

Trunk Based Development (Desarrollo Basado en Trunk) es una metodología de desarrollo de software que promueve la idea de que todas las ramas de desarrollo deben estar vinculadas a una única rama principal (trunk). En lugar de crear numerosas ramas para funcionalidades aisladas, los desarrolladores envían sus cambios directamente al tronco principal varias veces al día. Esto fomenta la colaboración continua y la entrega frecuente de pequeños cambios.