



Las Americas Institute of Technology

Docente:

Kelyn Tejada Belliard

Asignatura:

Programación 3

Alumno y matrícula:

Manuel de Jesús de la Cruz Solano
(2021-1967)

Sección:

Grupo 2 (sábados de 09:00 a 12:59)

Desarrolla el siguiente Cuestionario

1- ¿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.

2- ¿Para que funciona el comando Git init?

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.

3- ¿Qué es una rama?

Una rama es una versión del código del proyecto sobre el que estás trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura.

4- ¿Como saber en qué rama estoy?

Para saber en qué rama te encuentras en Git, puedes ejecutar el siguiente comando en la línea de comandos dentro del directorio de tu repositorio: Git Branch.

Esto mostrará una lista de todas las ramas locales en tu repositorio, y la rama actual estará resaltada con un asterisco. Por ejemplo, si estás en la rama "main", la salida sería algo así:

En este ejemplo, la rama actual es "main".

```
* main
feature-branch
another-feature-branch
```

5- ¿Quién creo git?

Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005.

6- ¿Cuáles son los comandos más esenciales de Git?

- **git init:** Inicializa un nuevo repositorio de Git en un directorio vacío o existente.
- **git add:** Agrega archivos al área de preparación (staging area) para ser incluidos en el próximo commit.
- **git commit:** Crea un nuevo commit con los archivos en el área de preparación y agrega un mensaje de confirmación (commit message) que describe los cambios realizados.
- **git status:** Muestra el estado actual del repositorio, incluyendo los archivos modificados, los archivos en el área de preparación y los archivos sin seguimiento.
- **git log:** Muestra un historial de todos los commits realizados en el repositorio, con detalles como el autor, la fecha y el mensaje de confirmación.
- **git branch:** Muestra una lista de todas las ramas (branch) en el repositorio y te permite crear, cambiar o eliminar ramas.
- **git merge:** Combina dos o más ramas de un repositorio.
- **git pull:** Obtiene (fetch) y fusiona (merge) los cambios desde un repositorio remoto en tu repositorio local.
- **git push:** Sube tus cambios locales a un repositorio remoto.

Estos son solo algunos de los comandos más esenciales de Git, pero hay muchos otros comandos y opciones disponibles para realizar tareas más avanzadas y específicas.

7- ¿Qué es git Flow?

Gitflow es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales.

Inicializa el espacio de trabajo. De forma automática, crea las ramas que necesitamos y permite configurar el nombre de las mismas. Podemos ver que por defecto (usando `intro` en vez de escribir nada) pone nombres por defecto a cada rama.

8- ¿Que es trunk-based development?

Trunk-based development es una práctica de gestión de control de versiones en la que los desarrolladores fusionan pequeñas actualizaciones de forma frecuente en un “tronco” o rama principal (main).