



*Las Americas Institute of Technology*

NOMBRE

Cristopher

APELLIDO

Santana

MATRÍCULA

2023-1193

CARRERA

Desarrollo de software

## INDICE

<b>Cuestionario.....</b>	<b>3</b>
1. ¿Qué es Git?.....	3
2. ¿Para qué sirve el comando git init?.....	3
3. ¿Qué es una rama en Git?.....	3
4. ¿Cómo saber en cuál rama estoy trabajando?.....	4
5. ¿Quién creó Git?.....	4
6. ¿Cuáles son los comandos esenciales de Git?.....	4
7. ¿Qué es Git Flow?.....	5
Es una estrategia popular de ramificación de Git que simplifica la gestión de lanzamientos. Fue introducida por el desarrollador de software Vincent Driessen en 2010. Básicamente, Git flow consiste en aislar el trabajo en diferentes tipos de ramas de Git.....	5
8. ¿Qué es el desarrollo basado en trunk (Trunk Based Development)?.....	6
<b>Bibliografía.....</b>	<b>7</b>

# 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. 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.

## 2. ¿Para qué sirve 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.

## 3. ¿Qué es una rama en Git?

Una rama o branch 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.

En otras palabras, un brunch o rama en Git es una rama que proviene de otra. Imagina un árbol, que tiene una rama gruesa, y otra más fina, en la rama más gruesa tenemos los commits principales y en la rama fina tenemos otros commits que pueden ser de hotfix, development entre otros.

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

Para saber qué ramas están disponibles y cuál es el nombre de la rama actual, ejecuta `git branch`

#### **5. ¿Quién creó 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 esenciales de Git?**

- **Git clone:** Git clone es un comando para descargar el código fuente existente desde un repositorio remoto.
- **Git branch:** Las ramas (branch) son altamente importantes en el mundo de Git. Usando ramas, varios desarrolladores pueden trabajar en paralelo en el mismo proyecto simultáneamente.

- Git add: Cuando creamos, modificamos o eliminamos un archivo, estos cambios suceden en local y no se incluirán en el siguiente commit (a menos que cambiemos la configuración).

Necesitamos usar el comando git add para incluir los cambios del o de los archivos en tu siguiente commit.

- Git commit: Git commit es como establecer un punto de control en el proceso de desarrollo al cual puedes volver más tarde si es necesario.

También necesitamos escribir un mensaje corto para explicar qué hemos desarrollado o modificado en el código fuente.

- Git push: Después de haber confirmado tus cambios, el siguiente paso que quieres dar es enviar tus cambios al servidor remoto. Git push envía tus commits al repositorio remoto.
- Git merge: Cuando ya hayas completado el desarrollo de tu proyecto en tu rama y todo funcione correctamente, el último paso es fusionar la rama con su rama padre (dev o master).
- Git init: Se usa para inicializar un nuevo repositorio Git en un directorio

## 7. ¿Qué es Git Flow?

Es una estrategia popular de ramificación de Git que simplifica la gestión de lanzamientos. Fue introducida por el desarrollador de software

Vincent Driessen en 2010. Básicamente, Git flow consiste en aislar el trabajo en diferentes tipos de ramas de Git.

## **8. ¿Qué es el desarrollo basado en trunk (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).

Dado que esta práctica simplifica las fases de fusión e integración, ayuda a lograr la CI/CD (Continuous Integration / Continuous Deployment, siendo estos la Integración Continua y Despliegue Continuo) y, al mismo tiempo, aumenta la entrega de software y el rendimiento de la organización.

Muestra y organiza los productos disponibles para la venta. Incluye descripciones, imágenes de varios ángulos, precios, descuento si hay alguna y opciones de personalización.

## Bibliografía

freeCodeCamp. 5 ago 2022. Cómo usar Git en un proyecto real. freeCodeCamp.

<https://www.freecodecamp.org/espanol/news/comandos-basicos-de-git-como-usar-git-en-un-proyecto-real/>

Jerson Martínez. 16 de septiembre de 2022. Trunk Based Development vs Git Flow, cuál elegir. openwebinars.

<https://openwebinars.net/blog/trunk-based-development-vs-git-flow-cual-elegir/>