



*Las Americas Institute of Technology*



**NOMBRE**

Joel de Jesús

**APELLIDO**

Oseliz Reynoso

**MATRÍCULA**

20231132

**CARRERA**

Desarrollo de Software

**MATERIA**

Programación III

**NOMBRE DEL DOCENTE**

Kelyn Tejada Belliard

**TEMA**

Módulo 5 – Herramientas de Administración de Fuentes – Tarea 3

**FECHA**

21/11/2024

## DESARROLLO DE LOS CONCEPTOS

### 1- Que es Git?

Git es un sistema de control de versiones de software de código abierto que permite a los desarrolladores colaborar en proyectos de software de manera eficiente. Los desarrolladores pueden utilizar Git para realizar un seguimiento de los cambios en el código fuente de un proyecto, así como para colaborar en el desarrollo de nuevas funcionalidades y corregir errores.

Fuente bibliográfica: <https://gamco.es/glosario/git/>

### 2- Para que funciona el comando git init?

El comando **git init** sirve para crear un repositorio nuevo, es decir, desde cero, a diferencia de **git clone**, que es para un repositorio ya existente.

Fuente bibliográfica: <https://ed.team/blog/que-es-git-y-como-funciona>

### 3-Que es una rama?

Las ramas son nuevos caminos, bifurcaciones, ramificaciones que toma el proyecto para no comprometer a la rama principal. De este modo, cada desarrollador puede crear diferentes ramas para hacer pruebas, corregir errores, agregar nuevas características. Luego estas ramas deben integrarse con la rama principal. Salen, toman su camino y luego se integran, esa es una fusión.

**También podríamos decir que:**

La rama principal se llama máster y es donde está el proyecto que sale a producción. Cada vez que se saca una nueva característica o que se quiere corregir algo se saca una rama, de tal manera que se pueda trabajar en un ambiente aislado. Es una copia exacta del proyecto, pero está separada. De este modo, si algo se rompe en ese proyecto, no comprometes al proyecto original. Si todo va bien, puedes unificar esa rama con el proyecto principal, si va mal, puedes eliminarla sin ningún problema.

Fuente bibliográfica: <https://ed.team/blog/que-es-git-y-como-funciona>

### 4- ¿Cómo saber en cual rama estoy?

Bueno, para poder tener conocimiento de la rama actual en la que nos encontramos, solo hace falta hacer uso del comando **git status**.

Fuente bibliográfica: <https://es.stackoverflow.com/questions/514716/como-saber-en-que-rama-est%C3%A1s-en-git-sin-consultarlo-continuamente>

## 5- ¿Quién creo Git?

El desarrollo de Git fue iniciado por Linus Torvalds (luego por la comunidad de desarrollo de Linux) en abril de 2005 cuando el sistema propietario de control de versiones usado en el desarrollo del kernel de Linux desde 2002, BitKeeper revoco la licencia gratuita usada para el desarrollo del Linux.

**Fuente bibliográfica:**

<https://es.wikipedia.org/wiki/Git>

<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Una-breve-historia-de-Git>

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

- **git init:** Inicia un nuevo repositorio Git.
- **git add:** Agrega cambios a los archivos al índice para incluirlos en el próximo commit.
- **git commit:** Registra los cambios en el repositorio.
- **git status:** Muestra el estado actual del repositorio, incluyendo archivos modificados, agregados y no rastreados.
- **git log:** Muestra el historial de commits.
- **git push:** Envía los cambios a un repositorio remoto.
- **git pull:** Obtiene cambios del repositorio remoto al repositorio local.
- **git branch:** Crea, lista o elimina ramas.
- **git merge:** Combina cambios de diferentes ramas.

**Fuente bibliográfica:** <https://www.hostgator.mx/blog/que-es-git/>

## 7- Que es Git Flow?

Creado en 2010 por Vincent Driessen. Es el flujo de trabajo más conocido. Está pensado para aquellos proyectos que tienen entregables y ciclos de desarrollo bien definidos. Está basado en dos grandes ramas con infinito tiempo de vida (ramas master y develop) y varias ramas de apoyo, unas orientadas al desarrollo de nuevas funcionalidades (ramas feature-\*), otras al arreglo de errores (hotfix-\*) y otras orientadas a la preparación de nuevas versiones de producción (ramas release-\*). La herramienta gitflow facilita la automatización de las tareas implicadas en este flujo de trabajo

**Fuente bibliográfica:** <https://es.wikipedia.org/wiki/Git>

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

Es una estrategia de Git donde existe un trunk(un branch principal, usualmente llamado master/main) en el cual todo el equipo colabora he integra directamente (hace push)

Dicho de otra manera:

**El desarrollo basado en tronco** 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.

**Fuente bibliográfica:**

<https://dev.to/marianocodes/por-que-trunk-based-development-i5n>

<https://openwebinars.net/blog/trunk-based-development-vs-git-flow-cual-elegir/#qu%C3%A9-es-trunk-based-development>