INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS

Las Americas Institute of Technology

PRESENTACION

NOMBRE:

Arianny Mabel

APELLIDO:

Carvajal Rojas

MATRICULA:

2021-2172

ASIGNATURA:

Programacion 3

MAESTRO:

Kelyn Tejada

1-Que 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 su copia del repositorio con la copia en el 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 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.

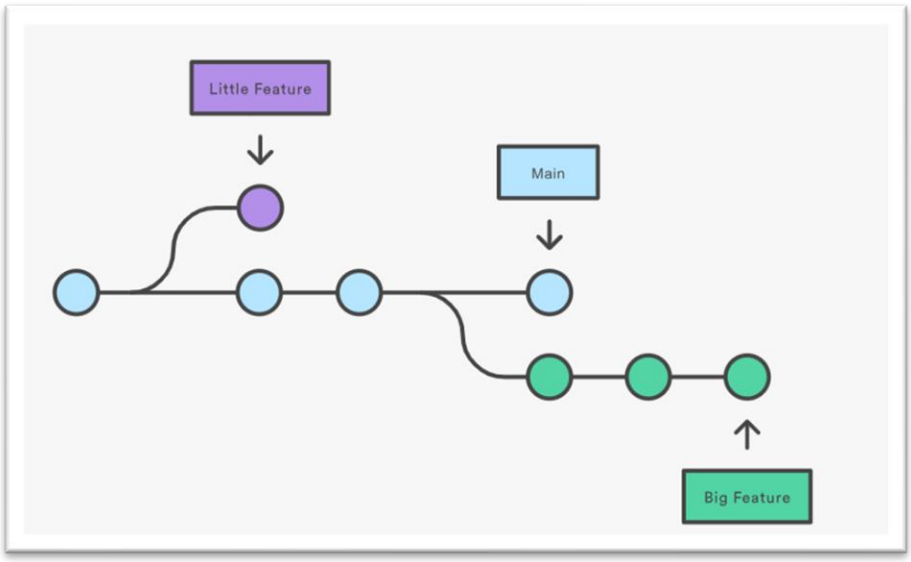
La mayoría de los demás comandos de Git no se encuentran disponibles fuera de un repositorio inicializado, por lo que este suele ser el primer comando que se ejecuta en un proyecto nuevo.

Al ejecutar git init, se crea un subdirectorio de .git en el directorio de trabajo actual, que contiene todos los metadatos de Git necesarios para el nuevo repositorio. Estos metadatos incluyen subdirectorios de objetos, referencias y archivos de plantilla. También se genera un archivo HEAD que apunta a la confirmación actualmente extraída.

4-Que es una rama?

En Git, las ramas son parte del proceso de desarrollo diario. Las ramas de Git son un puntero eficaz para las instantáneas de tus cambios. Cuando quieres añadir una nueva función o solucionar un error, independientemente de su tamaño, generas una nueva rama para alojar estos cambios.

Esto hace que resulte más complicado que el código inestable se fusione con el código base principal, y te da la oportunidad de limpiar tu historial futuro antes de fusionarlo con la rama principal.



El diagrama anterior representa un repositorio con dos líneas de desarrollo aisladas, una para una función pequeña y otra para una función más extensa. Al desarrollarlas en ramas, no solo es posible trabajar con las dos de forma paralela, sino que también se evita que el código dudoso se fusione con la rama main.

3-Como saber es que rama estoy?

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

```
$> git branch
main
another_branch
feature_inprogress_branch
$> git checkout feature_inprogress_branch
```

En el ejemplo anterior, se muestra cómo ver una lista de ramas disponibles ejecutando el comando git branch y cómo cambiar a una rama específica, en este caso, la rama feature_inprogress_branch.

5-Quien creo git?

hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. 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. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto.

Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).

6-Cuales son los comandos más esenciales de Git?

- git add
- Git clone
- Git Branch
- Git checkout
- Git status
- Git commit
- Git push
- Git pull
- Git revert
- Git merge

7-Que 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. Fue Vincent Driessen en nvie quien lo publicó por primera vez y quien lo popularizó. En comparación con el desarrollo basado en troncos, Gitflow tiene diversas ramas de más duración y mayores confirmaciones. Según este modelo, los desarrolladores crean una rama de función y retrasan su fusión con la rama principal del tronco hasta que la función está completa. Estas ramas de función de larga duración requieren más colaboración para la fusión y tienen mayor riesgo de desviarse de la rama troncal. También pueden introducir actualizaciones conflictivas.

8-Que es trunk-based development?

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).
Es una metodología de desarrollo de software que se centra en mantener una única línea principal o "trunk" (rama principal) en el repositorio de código fuente. En lugar de utilizar múltiples ramas durante el desarrollo y la implementación, como suele hacerse en otros enfoques como el Gitflow, el TBD se basa en trabajar directamente sobre la rama principal en todo momento.