

**Nombre:** Christopher Montero D'oleo  
**Matricula:** 2021-0622  
**Materia:** Programacion 3  
**Tema:** Herramienta de administracion de fuentes



## Desarrolla el siguiente Cuestionario

- **Que es 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).

Git, que presenta una arquitectura distribuida, es un ejemplo de DVCS (sistema de control de versiones distribuido, por sus siglas en inglés). En lugar de tener un único espacio para todo el historial de versiones del software, como sucede de manera habitual en los sistemas de control de versiones antaño populares, como CVS o Subversion (también conocido como SVN), en Git, la copia de trabajo del código de cada desarrollador es también un repositorio que puede albergar el historial completo de todos los cambios.

Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

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

El comando "\$ git init" se utiliza para iniciar un repositorio local en git.

- **Que es una rama?**

Una rama representa una línea independiente de desarrollo. Las ramas sirven como una abstracción de los procesos de cambio, preparación y confirmación. Puedes concebirlas como una forma de solicitar un nuevo directorio de trabajo, un nuevo entorno de ensayo o un nuevo historial de proyecto. Las nuevas confirmaciones se registran en el historial de la rama actual, lo que crea una bifurcación en el historial del proyecto.

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

Para saber en que rama estoy ubicado estas se puede usar el comando "\$ git branch".

- **Quien creo git?**

Git es un software de control de versiones diseñado por **Linus Torvalds** en 2007.

- **Cuales son los comandos más esenciales de Git?**

- git init
- git add
- git commit
- git status
- git log
- git branch

- git checkout
- git merge
- **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 2010 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.

Gitflow puede utilizarse en proyectos que tienen un ciclo de publicación programado, así como para la práctica recomendada de DevOps de entrega continua. Este flujo de trabajo no añade ningún concepto o comando nuevo, aparte de los que se necesitan para el flujo de trabajo de ramas de función. Lo que hace es asignar funciones muy específicas a las distintas ramas y definir cómo y cuándo deben estas interactuar. Además de las ramas de función, utiliza ramas individuales para preparar, mantener y registrar publicaciones. Por supuesto, también puedes aprovechar todas las ventajas que aporta el flujo de trabajo de ramas de función: solicitudes de incorporación de cambios, experimentos aislados y una colaboración más eficaz.

- **Que es trunk based development ?**

El desarrollo basado en troncos es una práctica de control de versiones en la que los desarrolladores fusionan pequeñas actualizaciones de forma frecuente en un "tronco" o rama principal. Dado que esta práctica simplifica las fases de fusión e integración, ayuda a lograr la CI y la CD y, al mismo tiempo, aumenta la entrega de software y el rendimiento de la organización.

En los primeros tiempos del desarrollo de software, los programadores no disfrutaban del lujo de los sistemas de control de versiones modernos. Más bien, desarrollaban dos versiones del software a la vez para poder realizar un seguimiento de los cambios y revertirlos si era necesario. Con el tiempo, este proceso resultó ser muy laborioso, costoso e ineficiente.