Tarea No. 1

**¿Qué es Git?**

Git es un proyecto de código abierto desarrollado por Linus Torvalds, es el principal control de versiones a diferencia de los controles de versión anteriores como CVS o subversión que almacenaban el historial de versiones en único espacio del software, en Git cada desarrollador cuenta con su propio espacio de trabajo el cual genera un historial de versiones independiente, el cual es llamado repositorio.

Con Git es posible saber en que momento se realizo el cambio y quien realizo el cambio y en caso dar algún error no impacta el trabajo de los demás desarrolladores, siendo un software que brinda Rendimiento, Seguridad y Flexibilidad.

Git es una herramienta que facilita el trabajo en equipo entre desarrolladores.

**¿Control de Versiones de Git?**

El control de versiones es un sistema que registra los cambios realizados en un archivo o un conjunto de archivos a lo largo del tiempo, de modo que las versiones anteriores pueden ser recuperadas un tiempo después.

La recuperación de las versiones anteriores son de gran beneficio porque le da trazabilidad a los cambios que se ejecutaron (Quien, Cuando y en donde) dando el alcance al usuario de poder comparar las versiones anteriores con las actuales, poder experimentar con el código sin temor a que la versión que funcionaba se pueda arruinar, detectar en que momento se trabajo un código el cual está dando error y está afectando.

**Sistemas de Control de Versiones Locales**

La metodología es muy sencilla y comúnmente lo que los usuarios ejecutan es copiar los archivos a otra carpeta o directorio, lo malo de este método es que es muy propenso a errores y perder la trazabilidad, los errores comunes son:

* Olvidar en cual archivo se encuentra.
* Guardar accidentalmente en el archivo equivocado.
* Sobre escribir en archivos que no era necesario.

La herramienta más utilizada para el control de versiones fue RCS el cual funciona guardando parches en un formato especial en el disco y es capaz de recrear un archivo en cualquier momento.

**Sistema de Control de Versiones Centralizados**

Los CVCS fueron creados para los usuarios que necesitan trabajar en conjunto con otros usuarios en otros sistemas y su metodología es que tienen un único servidor el cual contiene todos los archivos versionados y varios clientes que descargan los archivos desde ese lugar central. Frente al control de versiones locales.

Este sistema ofrece ventajas en su configuración como: los administradores cuentan con un mayor alcance de saber en que está trabajando los otros desarrolladores, los administradores cuentan con un control detallado sobre que puede hacer cada usuario y es más fácil de administrar.

Sin embargo, este sistema también cuenta con desventajas como: no cuenta con un soporte en caso el servidor centralizado falle, es decir que si el servidor falla toda la labor de los desarrolladores y usuarios se detiene hasta que el problema sea resuelto, si el disco duro se arruina y el responsable no ha realizado una copia de seguridad, la información se perderá. Es decir, en este estilo de sistema es tener toda la información en un mismo lugar.

**Sistema de Control de Versiones Distribuidos**

Es un sistema el cual ofrece soluciones para los problemas que se han mencionado de los sistemas anteriores. Es un DVCS entre los sistemas más importantes se encuentra Git, los clientes no solo descargan la última copia instantáneamente si no que réplica y maneja el repositorio completo. Es decir que si un servidor deja de funcionar y estos sistemas estaban colaborando a través del, cualquier repositorio disponible en los clientes puede ser copia y restaurado, este sistema brinda mejora para que diferentes grupos de trabajo se integren en un mismo proyecto e implementar flujos de trabajo como son los modelos jerárquicos.

**¿Cuáles son los estados del archivo Git?**

* Archivos Untracked: Son documentos fuera de Git (en un disco duro), los cuales nunca se han visto afectados por un comando Git.
* Archivos Unstaged: Son documentos dentro de Git los cuales nunca han sido afectados por un comando Git a lo cual Git reconoce que está desactualizado y sus ultimas versiones fueron guardadas en un disco duro
* Archivos Staged: son documentos que Git reconoce, han sido afectados por Git add, pero aún no se ha guardado en un repositorio con el comando Git commit
* Archivos Traked: Archivos dentro de Git, no tienen cambios pendientes y se encuentran guardados por medio de comandos git add y git comment

**¿Cómo se configura un Repositorio?**