César Eduardo del Valle Ramírez 201610993 3002080570101

¿Qué es GIT?

Git es un sistema de control de versiones distribuido, que se utiliza para administrar y realizar el seguimiento de los cambios en el código fuente y otros archivos de un proyecto.

Control de versiones con GIT

El control de versiones con Git se refiere al proceso de rastrear y administrar los cambios realizados en un proyecto de software a lo largo del tiempo. Con Git, los desarrolladores pueden realizar un seguimiento de cada cambio que se realiza en el código, ya sea una adición, modificación o eliminación, y pueden revertir a una versión anterior en caso de errores o problemas.

Estados de un archivo en GIT

En Git, los archivos de un proyecto pueden tener varios estados diferentes a medida que se modifican y se agregan al repositorio. Estos estados son los siguientes:

- Untracked: los archivos que no están siendo rastreados por Git y no están incluidos en el historial de cambios del repositorio. Estos archivos no se incluirán en el siguiente commit.
- Unmodified: los archivos que están siendo rastreados por Git pero no han sido modificados desde la última confirmación.
- Modified: los archivos que han sido modificados desde la última confirmación y aún no se han confirmado. Estos archivos se han cambiado en el directorio de trabajo pero no han sido agregados al área de preparación.
- Staged: los archivos que han sido modificados y agregados al área de preparación (staging area) para ser incluidos en el siguiente commit.
- Committed: los archivos que han sido confirmados en el repositorio. Estos archivos están guardados en la base de datos de Git y forman parte del historial de cambios.

Como se configura un repositorio

- Crear un directorio para el repositorio: crea un nuevo directorio en tu ordenador donde almacenarás el repositorio. Puedes hacer esto a través de la línea de comandos utilizando el comando "mkdir".
- Inicializar el repositorio: una vez que tengas el directorio creado, ve a la línea de comandos y navega hasta el directorio. Luego, utiliza el comando "git init" para inicializar el repositorio. Esto creará una carpeta oculta ".git" dentro del directorio que contiene todos los archivos necesarios para el control de versiones.

- Agregar archivos al repositorio: ahora que el repositorio está configurado, puedes agregar archivos al mismo utilizando el comando "git add". Puedes agregar archivos individuales o todo el directorio utilizando "git add ." para agregar todos los archivos en el directorio.

- Confirmar los cambios: después de agregar los archivos al repositorio, utiliza el comando "git commit" para confirmar los cambios. Puedes añadir un mensaje descriptivo utilizando el argumento "-m" seguido del mensaje.

- Configurar un repositorio remoto: si quieres compartir tu repositorio con otros desarrolladores, debes configurar un repositorio remoto. Puedes hacer esto utilizando servicios como GitHub, GitLab o Bitbucket. Una vez que hayas creado un repositorio remoto, utiliza el comando "git remote add" para agregar la URL del repositorio remoto a tu repositorio local.

- Subir cambios al repositorio remoto: finalmente, para subir los cambios a tu repositorio remoto, utiliza el comando "git push". Esto enviará los cambios confirmados en tu repositorio local al repositorio remoto.

Comandos en GIT

- git init: inicializa un repositorio de Git vacío en un directorio.
- git clone: crea una copia de un repositorio existente en un nuevo directorio.
- git status: muestra el estado actual del repositorio, incluyendo archivos modificados, agregados y eliminados.
- git add: agrega cambios en archivos específicos o en todos los archivos del directorio actual al área de preparación (staging area).
- git commit: confirma los cambios agregados al área de preparación y los guarda en el historial de cambios del repositorio.
- git push: envía los cambios confirmados en el repositorio local al repositorio remoto.
- git pull: obtiene los cambios del repositorio remoto y los fusiona con los cambios locales.
- git branch: muestra una lista de ramas en el repositorio y permite crear nuevas ramas.
- git checkout: cambia la rama actual o la versión de un archivo específico.
- git merge: fusiona los cambios de una rama a otra.