**Comandos Esenciales de Git y su Contexto de Uso**

## git init

Inicializa un nuevo repositorio en el directorio actual. Úsalo cuando quieras empezar un proyecto con control de versiones.

## git clone <url>

Clona un repositorio remoto en tu máquina local. Úsalo para obtener una copia de un proyecto existente.

## git status

Muestra el estado actual del repositorio: archivos modificados, nuevos o en staging. Úsalo para verificar antes de hacer commits.

## git add <archivo>

Agrega cambios al área de preparación (staging). Úsalo cuando quieras incluir archivos específicos en el próximo commit.

## git add .

Agrega todos los cambios en el directorio actual al staging. Útil cuando quieres preparar todo de una sola vez.

## git commit -m "mensaje"

Guarda los cambios en el historial con un mensaje descriptivo. Úsalo para documentar cambios importantes.

## git log

Muestra el historial de commits. Úsalo para revisar cambios anteriores o encontrar un commit específico.

## git diff

Muestra las diferencias entre archivos modificados y el último commit. Útil antes de hacer un commit para verificar cambios.

## git branch

Lista todas las ramas en el repositorio. Úsalo para ver en qué rama estás trabajando.

## git branch <nombre>

Crea una nueva rama. Úsalo para desarrollar nuevas características sin afectar la rama principal.

## git checkout <rama>

Cambia a una rama específica. Úsalo para trabajar en diferentes ramas del proyecto.

## git merge <rama>

Combina cambios de otra rama con la actual. Úsalo cuando hayas terminado una característica y quieras integrarla.

## git pull

Actualiza el repositorio local con los últimos cambios del remoto. Úsalo antes de empezar a trabajar para evitar conflictos.

## git push

Envía tus commits locales al repositorio remoto. Úsalo para compartir tus cambios con el equipo.

## git remote -v

Muestra las URLs de los repositorios remotos configurados. Útil para verificar a dónde estás haciendo push/pull.

## git reset --hard <commit>

Reinicia el repositorio a un commit específico, eliminando cambios posteriores. Úsalo con precaución para deshacer errores.

## git stash

Guarda cambios no confirmados temporalmente sin hacer commit. Úsalo cuando necesites cambiar de rama rápidamente.

## git stash pop

Recupera los cambios guardados en stash. Úsalo cuando quieras volver a aplicar tus modificaciones.

## git revert <commit>

Crea un nuevo commit que deshace los cambios de un commit específico. Úsalo para revertir errores sin modificar el historial.