Comandos básicos de Git

Guía rápida para principiantes

# Introducción

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Permite gestionar el historial de cambios en los archivos de un proyecto, facilitando la colaboración entre varios desarrolladores.

## Comandos esenciales

* git init: Inicializa un nuevo repositorio Git en el directorio actual.
* git clone : Clona un repositorio remoto en tu ordenador.
* git status: Muestra el estado de los archivos en el directorio de trabajo y el área de preparación (staging area).
* git add : Añade archivos al área de preparación para el próximo commit.
* git commit -m "mensaje": Guarda los cambios añadidos al historial con un mensaje descriptivo.
* git log: Muestra el historial de commits del repositorio.
* git diff: Muestra las diferencias entre archivos modificados y el último commit.
* git branch: Lista, crea o elimina ramas.
* git checkout : Cambia a la rama especificada.
* git merge : Fusiona la rama indicada con la rama actual.
* git pull: Descarga y fusiona los cambios del repositorio remoto en la rama actual.
* git push: Sube tus commits locales al repositorio remoto.

## Comandos útiles adicionales

* git remote -v: Muestra las URLs de los repositorios remotos configurados.
* git rm : Elimina archivos del área de trabajo y del historial.
* git reset : Elimina archivos del área de preparación (staging).
* git stash: Guarda temporalmente los cambios no confirmados.
* git fetch: Descarga los últimos cambios del repositorio remoto, pero no los fusiona.

## Consejos prácticos

* Usa git status frecuentemente para saber en qué estado están tus archivos.
* Escribe mensajes de commit claros y descriptivos.
* Crea ramas para trabajar en nuevas funcionalidades o correcciones sin afectar la rama principal (main o master).
* Sincroniza tu rama con git pull antes de hacer git push para evitar conflictos.

## Conclusión

Dominar estos comandos básicos de Git te permitirá gestionar eficazmente tus proyectos y colaborar de forma eficiente con otros desarrolladores. ¡No dudes en consultar la documentación oficial de Git para profundizar más!