Informe: Comandos Básicos de Git

¿Qué es Git?  
  
Git es un sistema de control de versiones distribuido que permite a múltiples personas trabajar en el mismo proyecto sin sobrescribir los cambios de otros. Es como una especie de “máquina del tiempo” que guarda el historial de tus archivos y te permite volver a versiones anteriores o colaborar de forma segura.

## git init

Este comando se usa para iniciar un nuevo repositorio Git en una carpeta.  
Ejemplo:  
 git init  
Esto crea una carpeta oculta llamada .git, que contiene toda la información necesaria para el seguimiento del proyecto.

## git clone

Sirve para copiar un repositorio que ya existe (por ejemplo, desde GitHub) a tu computadora.  
Ejemplo:  
 git clone https://github.com/usuario/repositorio.git

## git status

Muestra el estado actual de los archivos: cuáles están modificados, listos para guardar o aún no seguidos.  
Ejemplo:  
 git status

## git add

Agrega archivos al área de preparación (staging).  
Ejemplo:  
 git add archivo.txt  
 git add .

## git commit

Guarda los cambios agregados con add en el historial del proyecto.  
Ejemplo:  
 git commit -m "Mensaje que describe lo que hiciste"

## git add y git commit siempre van en pareja

## Primero usas git add para decir qué archivos quieres guardar, y luego git commit para guardar esos cambios con un mensaje.

## git log

Muestra un historial de los commits realizados en el repositorio.  
Ejemplo:  
 git log

## git push

Envía los commits realizados en tu máquina al repositorio remoto.  
Ejemplo:  
 git push origin main

## git pull

Actualiza tu repositorio local con los últimos cambios del remoto.  
Ejemplo:  
 git pull origin main

### git pull antes de git push

Antes de subir tus cambios, haz un git pull para asegurarte de tener lo último del equipo. Así evitas conflictos.

## git branch

Sirve para ver, crear o cambiar ramas.  
Ejemplo:  
 git branch  
 git branch nueva-rama  
 git checkout nueva-rama

## git merge

Combina los cambios de una rama con otra.  
Ejemplo:  
 git merge nueva-rama

# Conclusión

Estos son los comandos esenciales de Git que te permitirán comenzar a trabajar en proyectos de desarrollo de manera organizada. A medida que avances, irás aprendiendo más herramientas y conceptos como conflictos de merge, rebase, stash, entre otros. Por ahora, con esto tienes una excelente base para comenzar.