

# Guía Básica de Git

Git es un sistema de control de versiones distribuido que permite a los desarrolladores rastrear cambios en el código fuente a lo largo del tiempo y colaborar en proyectos de software.

## Conceptos básicos de Git

**Repositorio (repo):** Espacio de almacenamiento donde se guarda el código fuente y su historial de cambios. Puede ser local o remoto.

**Commit:** Instantánea del estado del código en un momento dado. Incluye un mensaje descriptivo.

**Branch:** Línea de desarrollo independiente dentro de un repositorio.

**Merge:** Proceso de combinar dos ramas diferentes en una sola.

**Clone:** Crea una copia local de un repositorio remoto.

**Push:** Envía commits locales a un repositorio remoto.

**Pull:** Descarga cambios del repositorio remoto al local.

## Acciones básicas de Git

Inicializar un repositorio	git init nombre_del_repositorio
Navegar al directorio	cd nombre_del_repositorio
Crear archivo README.md	echo "# Mi Proyecto" > README.md
Agregar archivo al staging	git add README.md
Realizar commit	git commit -m "Primer commit"
Agregar repositorio remoto	git remote add origin URL_DEL_REPOSITORIO
Hacer push inicial	git push -u origin main
Clonar repositorio	git clone URL_DEL_REPOSITORIO
Crear rama	git branch nombre_de_la_rama
Cambiar a una rama	git checkout nombre_de_la_rama
Fusionar rama	git merge nombre_de_la_rama
Hacer pull	git pull origin main
Hacer push	git push origin main

## Ejercicios propuestos

- Práctica 1: Crear un nuevo repositorio local y realizar un commit inicial.
- Práctica 2: Crear un repositorio remoto en GitHub y hacer un push de tu commit inicial.
- Práctica 3: Clonar un repositorio remoto y realizar cambios en el código.
- Práctica 4: Crear una nueva rama, realizar cambios y hacer un merge con la rama principal.

## Conclusión

Git es una herramienta poderosa para el control de versiones y la colaboración. Con estos conceptos básicos puedes comenzar a trabajar con Git y colaborar en proyectos. A medida que adquieras experiencia, podrás explorar comandos más avanzados como rebase, cherry-pick y resolución de conflictos.

## Recursos adicionales

- Documentación oficial de Git
- GitHub Guides
- Pro Git Book
- Git Cheat Sheet
- Git Immersion
- Learn Git Branching