

# Manejo de Ramas en Git

Git es un sistema de control de versiones distribuido que permite trabajar con diferentes versiones del código fuente simultáneamente. Uno de los conceptos más poderosos y útiles de Git es el manejo de ramas (branches), que facilita el desarrollo paralelo, la experimentación y la colaboración.

## ¿Qué es una rama?

Una rama en Git es una línea independiente de desarrollo. Permite que los desarrolladores trabajen en nuevas características, correcciones o experimentos sin afectar la rama principal (generalmente llamada main o master).

## Comandos básicos para manejar ramas

- **Crear una nueva rama:** `git branch nombre_rama`
- **Cambiar de rama:** `git checkout nombre_rama` o `git switch nombre_rama`
- **Crear y cambiar a una nueva rama:** `git checkout -b nombre_rama`
- **Listar ramas existentes:** `git branch`
- **Eliminar una rama:** `git branch -d nombre_rama`
- **Fusionar ramas:** `git merge nombre_rama` (desde la rama donde se quiere integrar los cambios)

## Buenas prácticas al trabajar con ramas

- Usar nombres descriptivos para las ramas, como `feature/login` o `bugfix/corrigir-error-404`.
- Mantener la rama principal (main/master) siempre estable y funcional.
- Fusionar frecuentemente para evitar conflictos grandes.
- Eliminar las ramas que ya no se utilicen para mantener el repositorio limpio.
- Utilizar `pull requests` o `merge requests` para revisar el código antes de fusionarlo.

## Flujo de trabajo común con ramas

1. Crear una rama desde main para desarrollar una nueva funcionalidad. 2. Realizar commits frecuentes en la nueva rama. 3. Fusionar los cambios en main cuando la funcionalidad esté completa. 4. Resolver conflictos si los hubiera. 5. Eliminar la rama luego de la fusión.

## Conclusión

El uso correcto de las ramas en Git mejora la organización, la colaboración y la calidad del desarrollo de software. Dominar el flujo de trabajo con ramas es fundamental para cualquier equipo de desarrollo moderno.