

# Conflictos entre ramas en Git

Los conflictos entre ramas surgen cuando dos desarrolladores tratan de editar el mismo contenido, estos conflictos pueden producir costos y nos hacen perder el tiempo por lo que para evitar estos conflictos los desarrolladores deben trabajar en ramas aisladas independientes.

La función principal del comando **git merge** es combinar ramas independientes y resolver los cambios en conflicto.

En la mayoría de los casos git se encarga de integrar los cambios automáticamente, pero qué pasa si no funciona?

Normalmente los conflictos surgen cuando dos personas han cambiado las mismas líneas de un archivo o si un desarrollador ha eliminado un archivo mientras otro lo estaba modificando. En estos casos, Git no puede determinar automáticamente qué es correcto. Los conflictos sólo afectan al desarrollador que realiza la fusión, el resto del equipo no se entera del conflicto. Git marcará el archivo como que tiene un conflicto y detendrá el proceso de fusión. Entonces el desarrollador es el responsable de resolver el conflicto.

## Tipos de conflictos:

- **Git no inicia la fusión:**

Git no inicia la fusión porque las **confirmaciones que se están fusionando podrían sobrescribir estos cambios pendientes**.

Cuando esto sucede, no se debe a conflictos con otros desarrolladores, sino con cambios locales pendientes. El estado local tendrá que estabilizarse mediante git stash, git checkout, git commit o git reset.

- **Git falla durante la fusión:**

Un fallo **DURANTE** una fusión indica un conflicto entre la rama local actual y la rama que se está fusionando. Esto indica un conflicto con el código de otro desarrollador. Git hará lo posible para fusionar los archivos, pero te dejará cosas para que se resuelvan manualmente en los archivos con conflictos.

## Identificar conflictos de fusión

Git generará un resultado descriptivo para indicarnos que se ha producido un conflicto (CONFLICT). Para obtener más información, podemos ejecutar el comando **git status**.

El resultado de git status indica si hay y cuales son las rutas sin fusionar debido a un conflicto.

## Resolver conflictos mediante la línea de comandos:

La forma más sencilla para resolver conflictos es editar el archivo conflictivo y usar el comando **git add** para preparar el nuevo contenido fusionado, finalmente realizar un **commit** y git nos confirmará que se ha fusionado.

### Comandos que nos ayudan a resolver conflictos

- git status: El comando status se utiliza frecuentemente cuando se trabaja con Git y durante una fusión **ayudará a identificar los archivos con conflictos**.
- git log--merge: Al pasar el argumento --merge al comando git log, se creará un registro con una **lista de confirmaciones que entran en conflicto entre las ramas que se van a fusionar**.
- git diff: ayuda a **encontrar diferencias entre los estados de un repositorio/unos archivos**. Esto es útil para predecir y evitar conflictos de fusión.

## Herramientas si no se puede realizar una fusión

- git checkout: para deshacer cambios en los archivos o para cambiar ramas.
- git resset --mixed: para deshacer cambios en el directorio de trabajo y el entorno de ensayo.

## Herramientas para cuando surgen conflictos de git durante una fusión

- git merge --abort: se saldrá del proceso de fusión y **volverá a poner la rama en el estado que tenía antes de que empezara la fusión**.
- git reset: para restablecer los archivos conflictivos a un estado que se sabe que es adecuado.

