

## Git: Branches, conflictos, merge.

### Branches

Las branches, son copias del código fuente (main) que le permiten a los distintos miembros de un equipo de desarrollo, trabajar simultáneamente en diferentes características y versiones sin llegar a afectar a la rama principal. Una vez que termina la fase de testing, se fusionan las ramas con el main. Esto permite una mejor integración y mantenimiento del código.

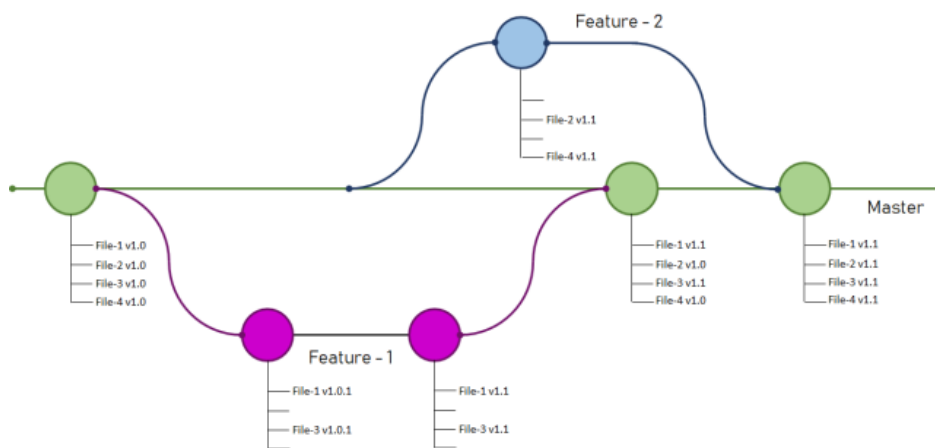


Fig. 1 GIT Branching and Merging

Pasos para crear una nueva rama en Git:

1. Ubicarse en la rama principal del proyecto llamada "main", con el comando `git checkout main`.
2. Crea una nueva rama con el comando `git branch` y el nombre de la nueva rama. Por ejemplo, para crear una nueva rama llamada "Dev-Sanchez", debes ejecutar `git branch Dev-Sanchez`.
3. Cambiar a la nueva rama que se acaba de crear usando el comando `git checkout Dev-Sanchez`.
4. Ahora se puede trabajar en la nueva rama como si fuera la rama principal.

## Conflictos

Los conflictos en Git son comunes durante el trabajo en equipo en un proyecto, especialmente cuando varios desarrolladores trabajan en diferentes ramas.

Son situaciones en las que el sistema no puede fusionar automáticamente dos ramas de desarrollo ya que ambas ramas han modificado el mismo archivo o la misma sección del código.

Cuando se produce un conflicto, Git indica que hay un problema y marca los archivos que han sufrido conflictos. El desarrollador debe solucionar manualmente el o los conflictos que haya surgido, es decir, debe elegir cuál de las dos versiones del archivo se va a mantener y cómo se deben combinar los cambios de ambas versiones.

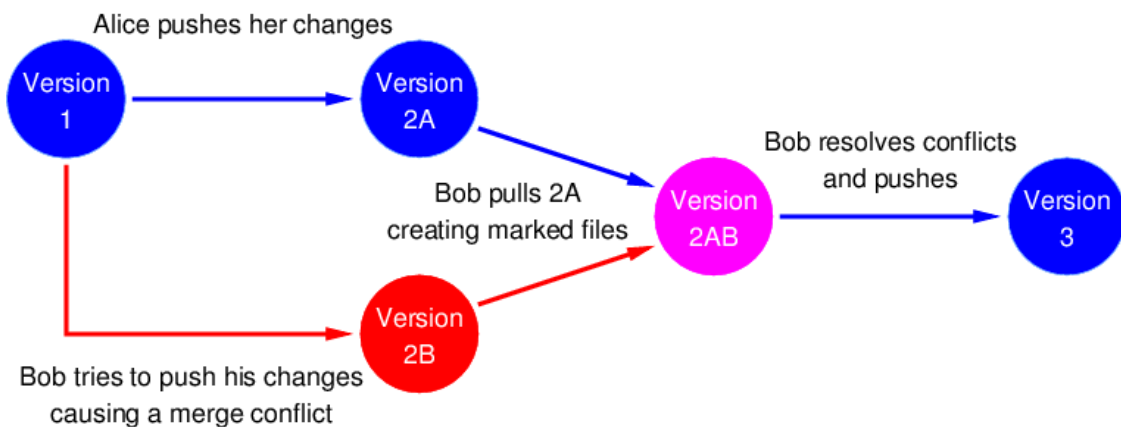


Fig. 2 GIT Conflicts

## Merge

Es una función que permite combinar y fusionar los cambios de dos o más ramas de desarrollo.

Durante el proceso de merge, se selecciona la rama que se quiere fusionar con la rama actual. Por ejemplo, si estás en la rama "main" y quieres fusionar la rama "Dev-Sanchez". Luego, Git intentará combinar automáticamente los cambios de ambas ramas. Si Git encuentra algún conflicto, detendrá el proceso e indicará qué archivos o líneas de código presentan conflictos. El desarrollador debe resolver manualmente estos conflictos antes de continuar con el merge. Una vez que se han resuelto los conflictos, se realiza un commit para confirmar.

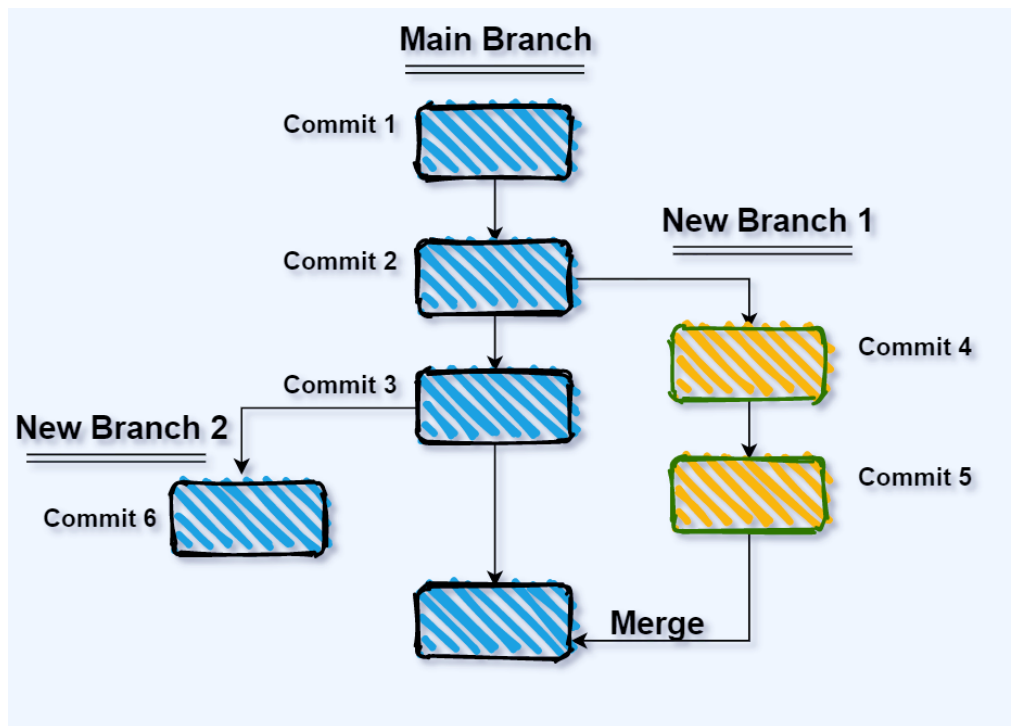


Fig. 3 GIT Merge

Referencias:

Git-it-guide, GitHub

Disponible en: <https://jlord.us/git-it/index.html>

Prabhu Vignesh Kumar Rajagopal, GIT Branch And Its Operations.

Disponible en: <https://digitalvarys.com/git-branch-and-its-operations/>

Resolving Merge Conflicts

Disponible en: <http://www.cs.utsa.edu/~cs3443/git/merge-conflicts.html>

Git Merge

Disponible en: <https://devdojo.com/guide/git/git-merge>