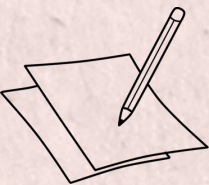




Algunos conceptos para comenzar a entender

GIT



¿Qué es un commit?

Con el comando "git commit" git toma una "captura" de los cambios preparados (staged) en el momento ejecutarlo dentro de un proyecto.

Cuando se hace un commit se guarda un "objeto captura" que contiene lo siguiente:

- Nombre e email del autor
- El mensaje
- Una referencia (pointer) al o los commits directamente anteriores a este

Comando para crear una nueva rama:
git branch [nombreDeLaNuevaBranch]

¿Qué es una branch?

- Una branch es una referencia a un commit (usando el ID del commit). Esta referencia es actualizada cada que se hace un commit nuevo.
- Una branch siempre apunta al último commit.



Por ejemplo si tienes una branch llamada Master, cada que se hace un commit el apuntador de master se mueve automáticamente.

HEAD

.Siempre necesitamos saber en qué branch estamos trabajando. Existe un apuntador llamado HEAD que indica cuál branch es la que está activa. Así cuando creas una branch sólo creas un nuevo apuntador pero no mueves el apuntador HEAD.



HEAD

Master

Commit 1

Commit 2

Commit 3

Dev

Comando para cambiar de branch:
git checkout [nombre_de_la_branch]

git checkout Dev

¿Qué es un merge?

Supongamos en nuestro proyecto decidimos crear una nueva branch para atender una funcionalidad nueva y mantener en la rama Master lo anterior a esto.

Master

Commit 1

Commit 2

Commit 3

Dev



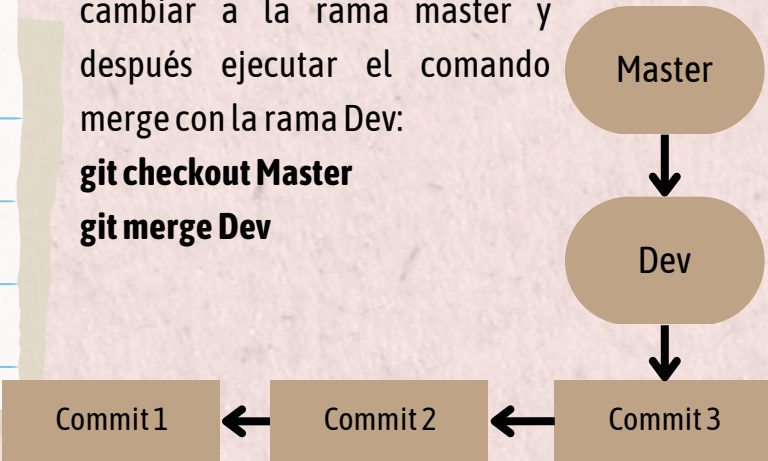
GIT

Cuando termines de implementar la funcionalidad nueva seguro querras integrarla con tu rama anterior (Master)

Para esto nos ayudan los merge, para integrar diferentes ramas.

Para esto primero debemos cambiar a la rama master y después ejecutar el comando merge con la rama Dev:

```
git checkout Master
git merge Dev
```

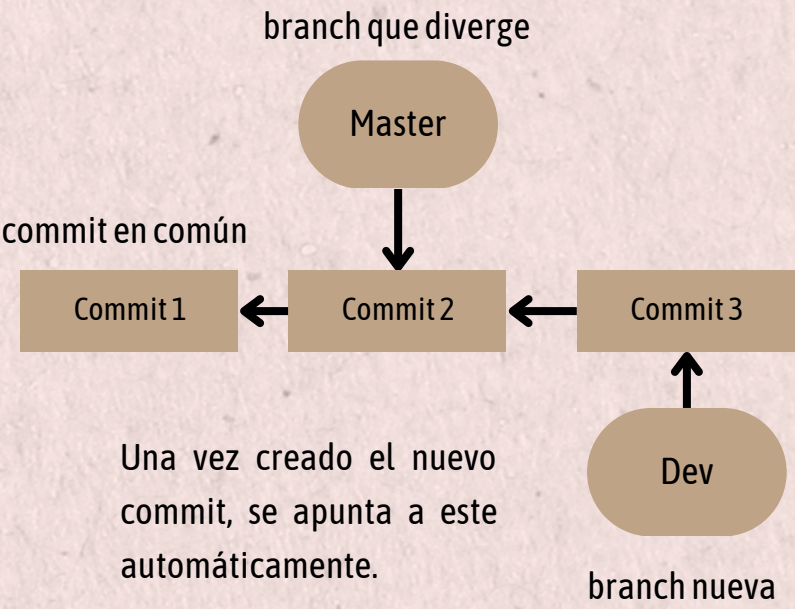


ESA FUE LA FORMA MÁS SENCILLA DE HACER UN MERGE... CUANDO NO EXISTEN DIVERGENCIAS ENTRE LAS RAMAS ...



¿QUÉ PASA CUANDO SI EXISTEN DIVERGENCIAS AL HACER UN MERGE?

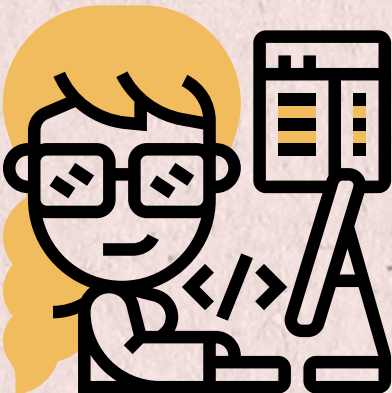
Git creara un nuevo commit conocido como **merge commit** a partir de tres cosas: el commit al que apunta la branch que diverge, el commit de la branch que se quiere integrar y el commit de dónde divergen ambas.



¿Conflictos?

No siempre se pueden integrar las branches divergentes automáticamente. Cuando esto sucede se genera un **conflict** y este debe ser solucionado manualmente

Git señalará cuáles son los archivos y los cambios que no puede integrar.



Se deben revisar estos archivos y corregirlos. Se puede utilizar alguna herramienta o "merge tool" para integrar los archivos.

Una vez resueltos, con el comando **git commit** se indica a git proseguir con el commit.