



# Introducción a los Repositorios de Código Distribuido

*Manejo de ramas*

Diego Madariaga

# Contenidos de la clase

1. Ramas
2. Creación y merge de ramas

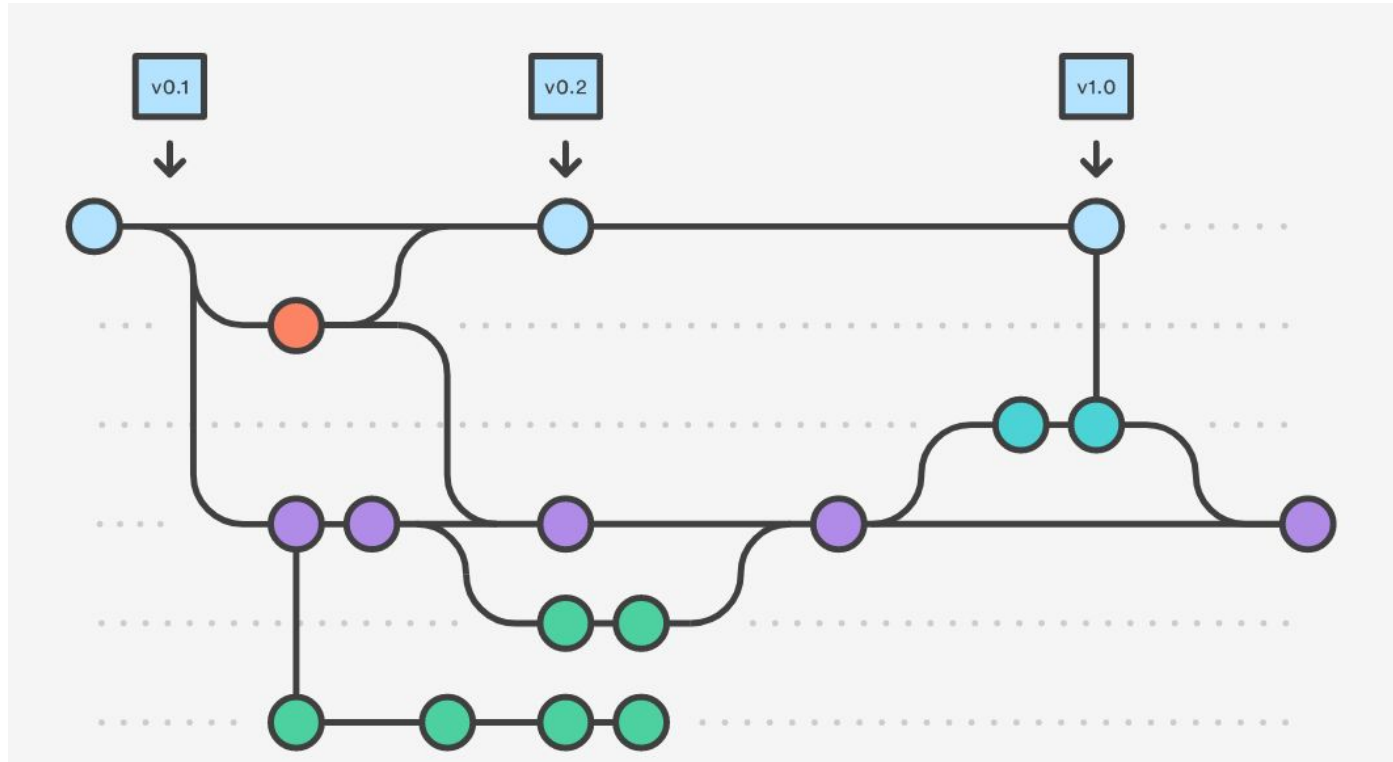
1.

Ramas

# Ramas

- ▷ Método fundamental para lanzar una nueva línea de desarrollo dentro de un proyecto
- ▷ Permite continuar en múltiples direcciones de forma simultánea a partir de un estado primario
- ▷ Permite mantener y producir diferentes versiones de un proyecto
- ▷ A menudo, las ramas se juntan para reunir esfuerzos aislados

# Ramas



# Razones para usar ramas

- ▷ Encapsula una fase de desarrollo (prototipo, alfa, beta, estable, etc)
- ▷ Aísla el desarrollo de una funcionalidad o el manejo de un *bug* (no importa si una rama contiene “poco” trabajo)
- ▷ Una rama individual puede representar el trabajo de una única persona

# Branch v/s Tag

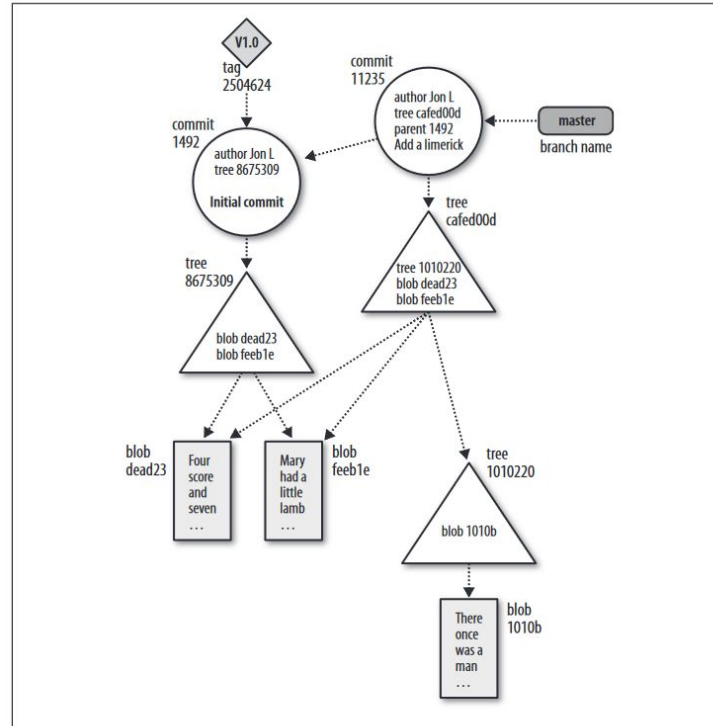


Figure 4-2. Git objects after second commit

# Branch v/s Tag

- ▷ Pueden parecer términos similares
- ▷ **Tag:** Nombre estático que no cambia a través del tiempo (punto de referencia)
- ▷ **Branch:** Es dinámica, y se mueve con cada commit que se realiza.
  - El nombre de una rama ayuda a seguir el desarrollo continuo de un proyecto



# Buenas prácticas

## Nombres de ramas:

- ▷ Puede ser un nombre arbitrario
- ▷ Por defecto, la rama principal se llama “master” o “main”
  - Suele contener la línea de desarrollo más robusta o estable
- ▷ Algunos nombres comunes:
  - develop
  - feature...
  - hotfix...

2.

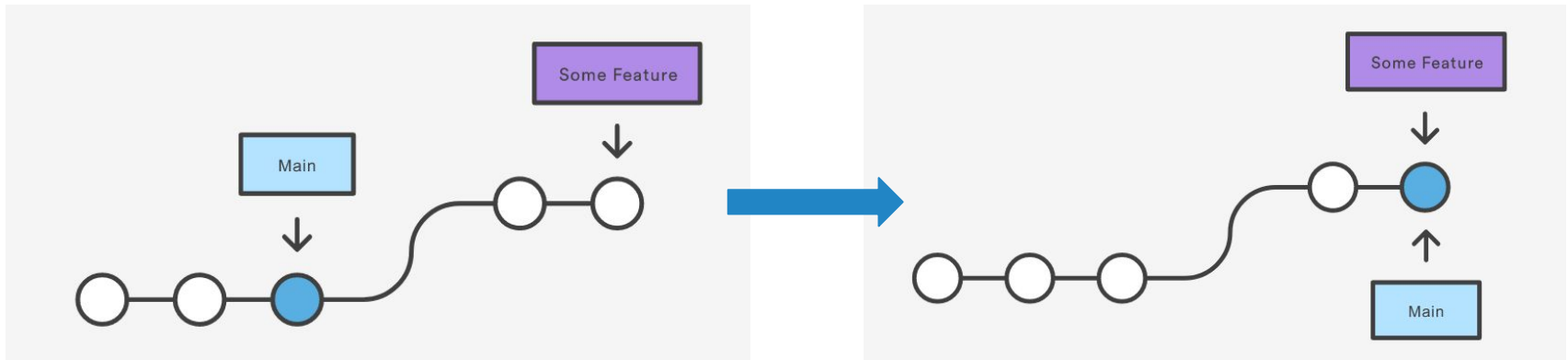
# Creación y merge de ramas

# Git merge

- ▶ Al momento de fusionar dos ramas, Git decide cuál algoritmo utilizará. Principalmente:
  - Fast-Forward
  - Recursive

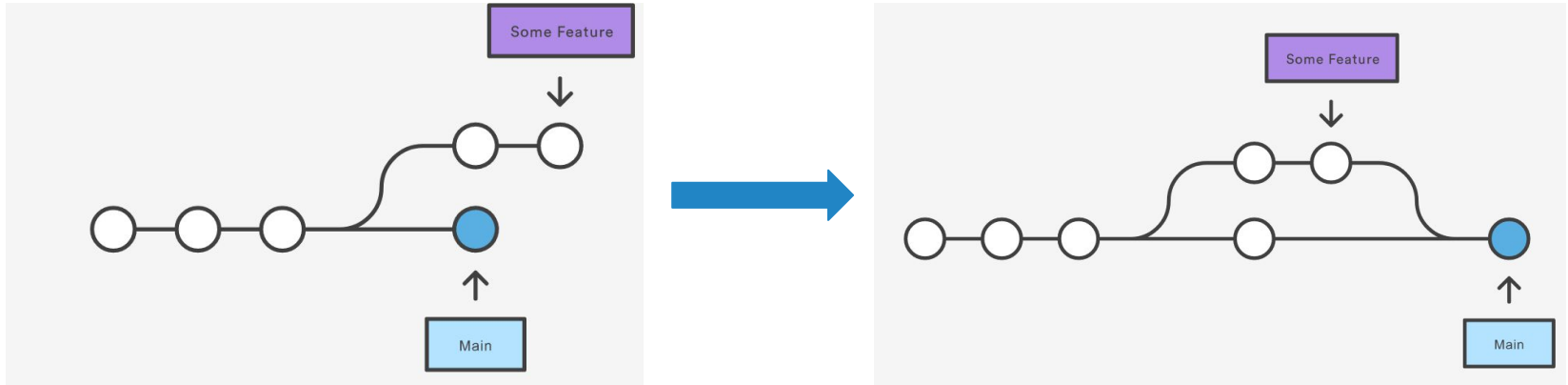
# Fast-forward

- Se utiliza cuando hay un desarrollo lineal entre las ramas que se quieren fusionar



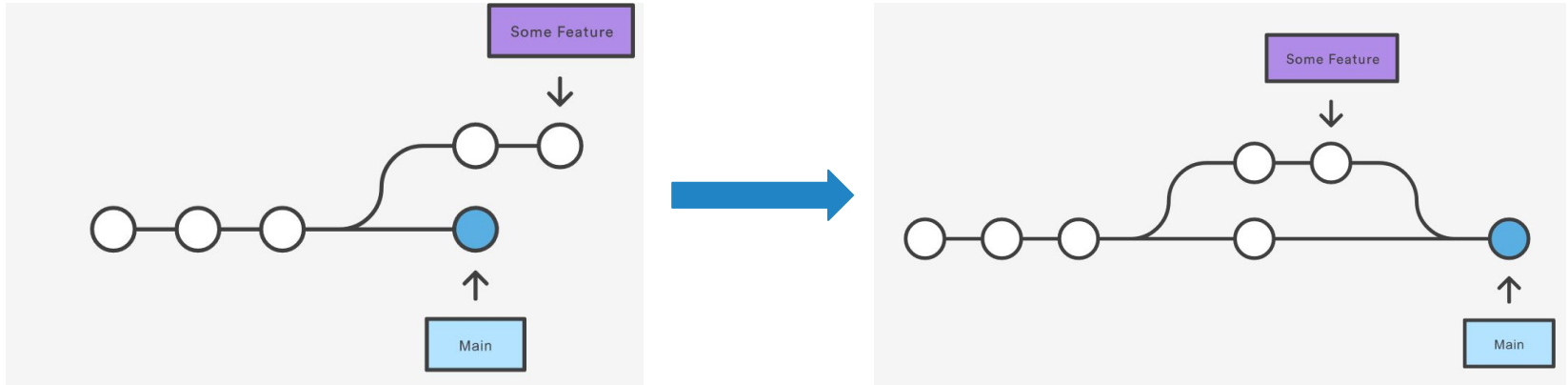
# Recursive

- ▷ No hay un desarrollo lineal entre las ramas (han divergido)

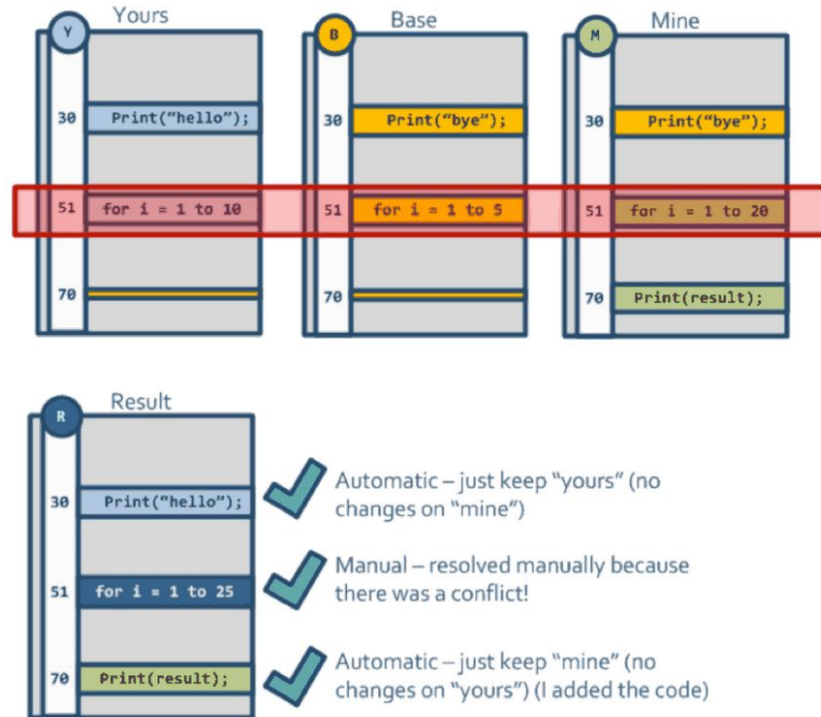


# Recursive

- También llamado “de 3 vías”: Se basa en el último commit de cada rama y en el commit antecesor común



# Recursive



(image from <http://www.drdoobs.com/tools/three-way-merging-a-look-under-the-hood/240164902>)

# Proyecto snake.py (lab 2)

- Arreglar error en variable difficulty
- En mensaje de GAME OVER se quiere poner el puntaje en amarillo





# Introducción a los Repositorios de Código Distribuido

*Manejo de ramas*

Diego Madariaga