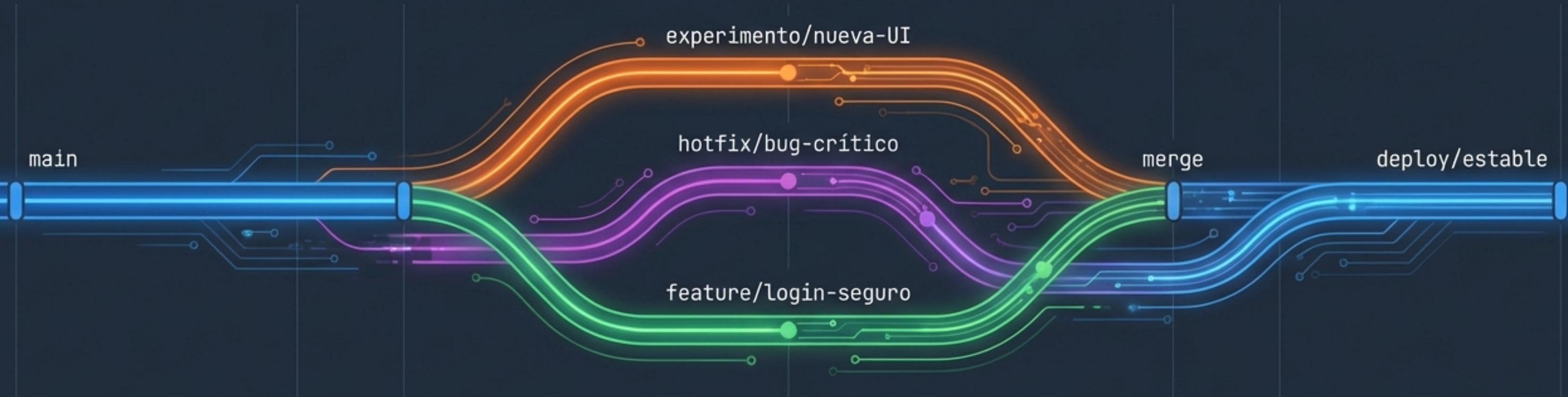


Dominando las Ramas en Git

Universos paralelos para tu código



JetBrains Mono

Las ramas te permiten divergir de la línea principal para experimentar, arreglar errores o añadir funciones sin romper lo que ya funciona.

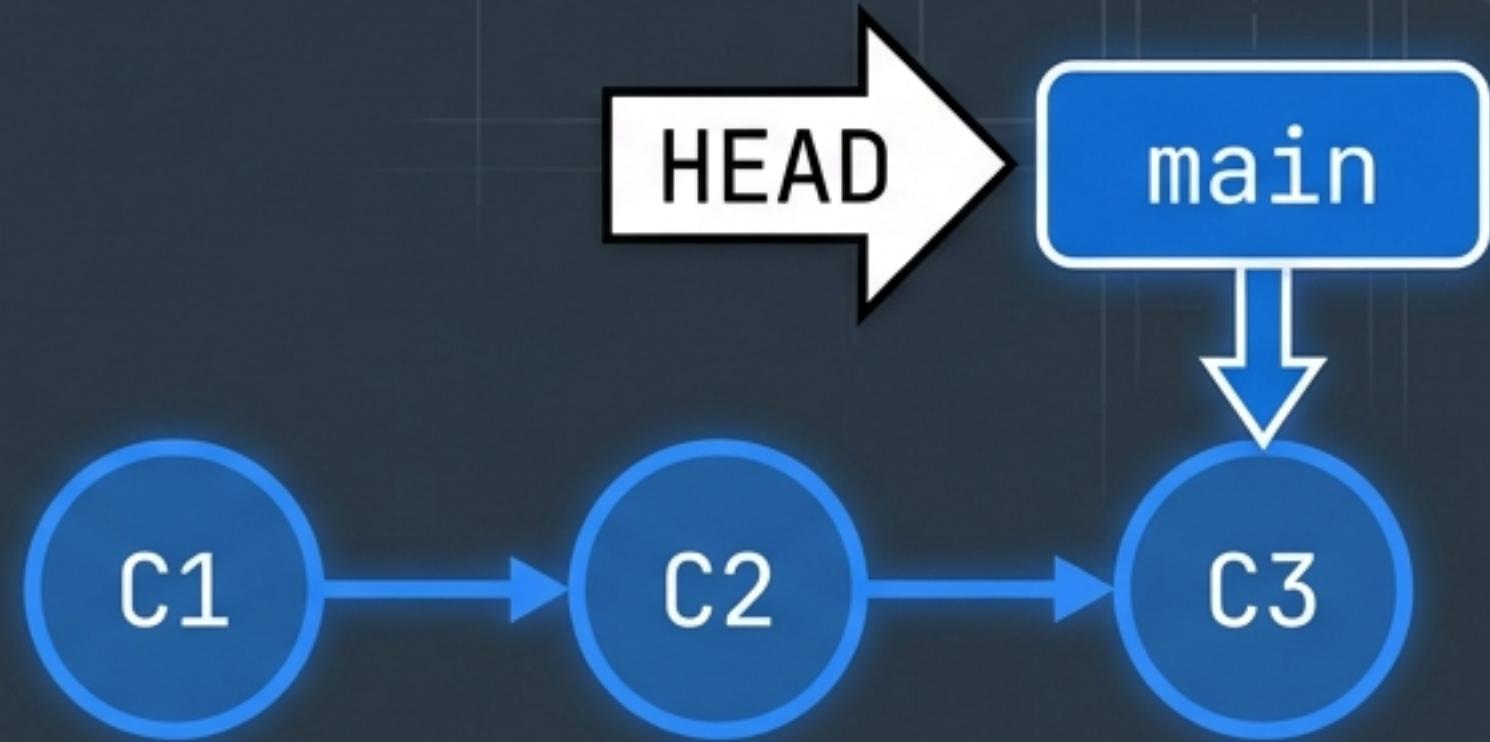
Inter

Basado en la documentación oficial de Git.

Anatomía de una Rama

Una rama no es una carpeta nueva.
Es una etiqueta móvil (puntero) que señala a un commit específico.

- **Puntero Móvil:** Cuando creas una rama, solo creas una nueva referencia.
- **HEAD:** Es el indicador de 'tú estás aquí'. Git sabe en qué rama trabajar mirando dónde apunta el HEAD.



El Kit de Herramientas

Ver

```
git branch
```

Lista ramas y marca la actual con *

Mover

```
git checkout <nombre>  
# o  
git switch <nombre>
```

Mueve el HEAD a la otra rama.

Crear

```
git branch <nombre>
```

Crea el puntero, pero no te mueve.

Borrar

```
git branch -d <nombre>
```

Elimina la etiqueta tras fusionar.

Ojos que ven: El Alias git lga

Para entender las ramas, necesitas ver el grafo. El log estándar es difícil de leer.

```
git config --global alias.lga "log --graph --oneline --all --decorate"
```

Ejecuta una sola vez. A partir de ahora, usa 'git lga'.

git log (Estándar)

```
commit a8f31d29b5c7e4e3a9b8d7f6c5a4b3d2e1
Author: Dev <dev@example.com>
Date:   Wed Nov 10 14:23:45 2023 -0500
```

Fix bug in authentication module.

```
commit a8f31d29b5c7e4e3a9b8d7f6c5a4b3d2e1
Author: Dev <dev@example.com>
Date:   Wed Nov 10 14:23:45 2023 -0500
```

Fix bug in authentication module.

```
commit a8f31d29b5c7e4e3a9b8d7f6c5a4b3d2e1
Author: Dev <dev@example.com>
Date:   Wed Nov 10 14:23:45 2023 -0500
```

git lga (Recomendado)

```
* a8f31d2 Fix bug in authentication module
  (HEAD -> main, origin/main)
```

```
  | 7e9c1a4 Update documentation for API
```

```
  | 4b3d2e1 Merge branch 'feature-x'
```

```
  |  \ 1c2d3e4 Implement feature-x (feature-x)
```

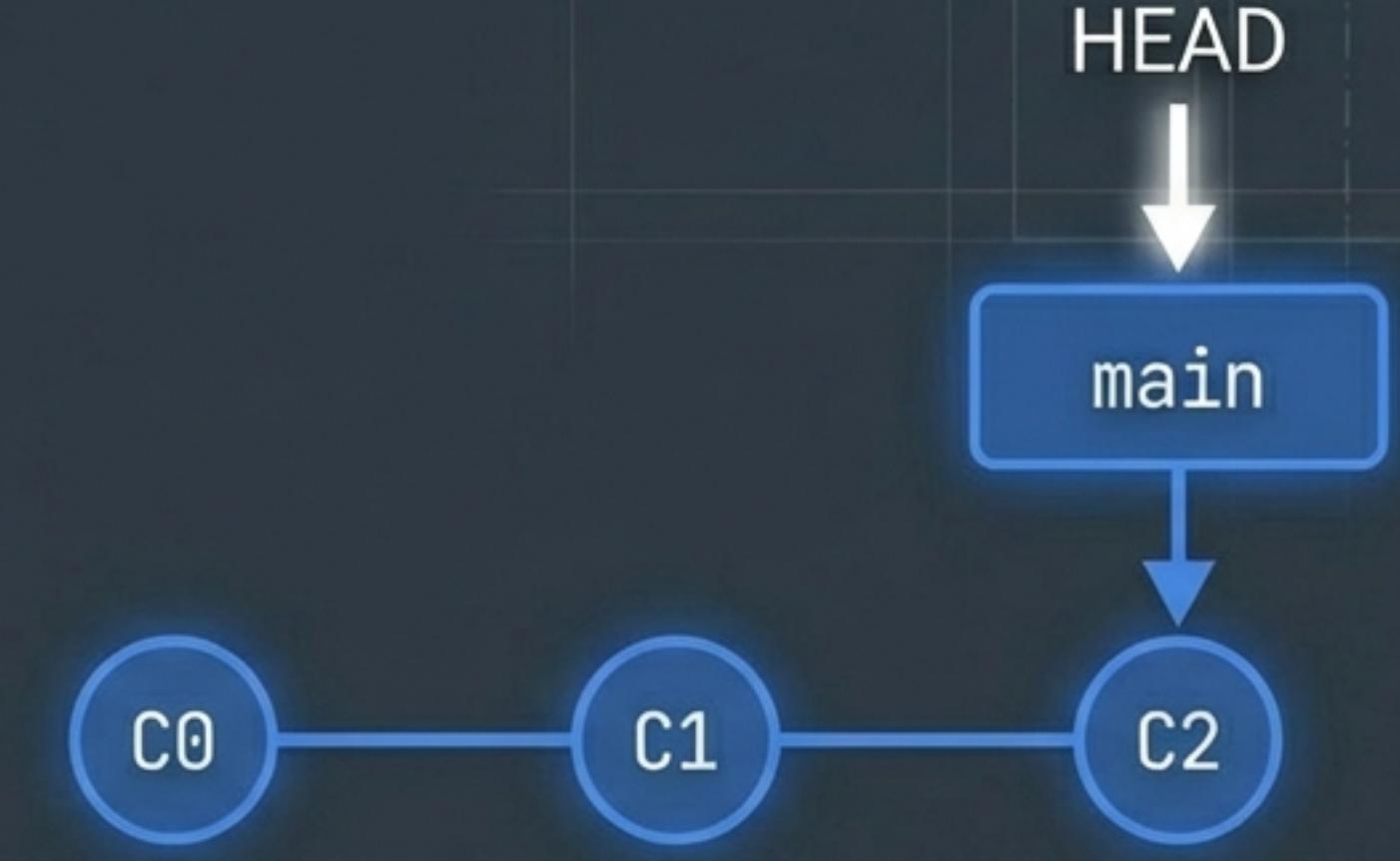
```
  /  * 8a7b6c5 Initial commit
```

Historia de una Rama: El Punto de Partida

Escenario: Estás trabajando en una web y quieres cambiar el color de fondo sin romper la versión pública.

Estado: Estás en la rama main.

```
$ git status  
On branch main
```



Creación y Movimiento

Creamos el “universo paralelo” y nos mudamos a él.

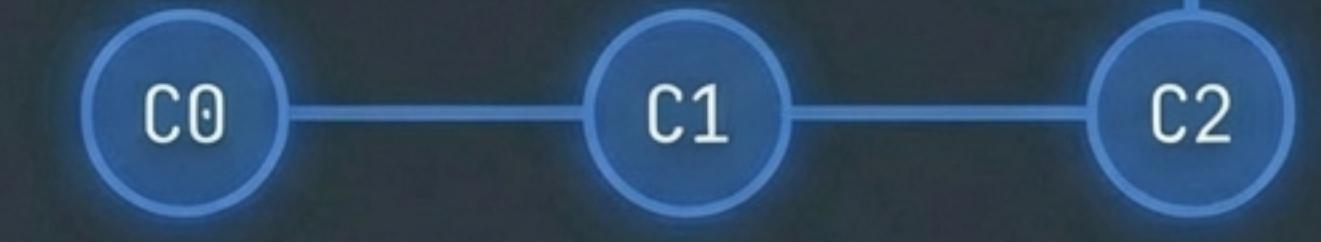
```
$ git branch experimento-fondo  
$ git checkout experimento-fondo  
Switched to branch 'experimento-fondo'
```

HEAD



experimento-fondo

main

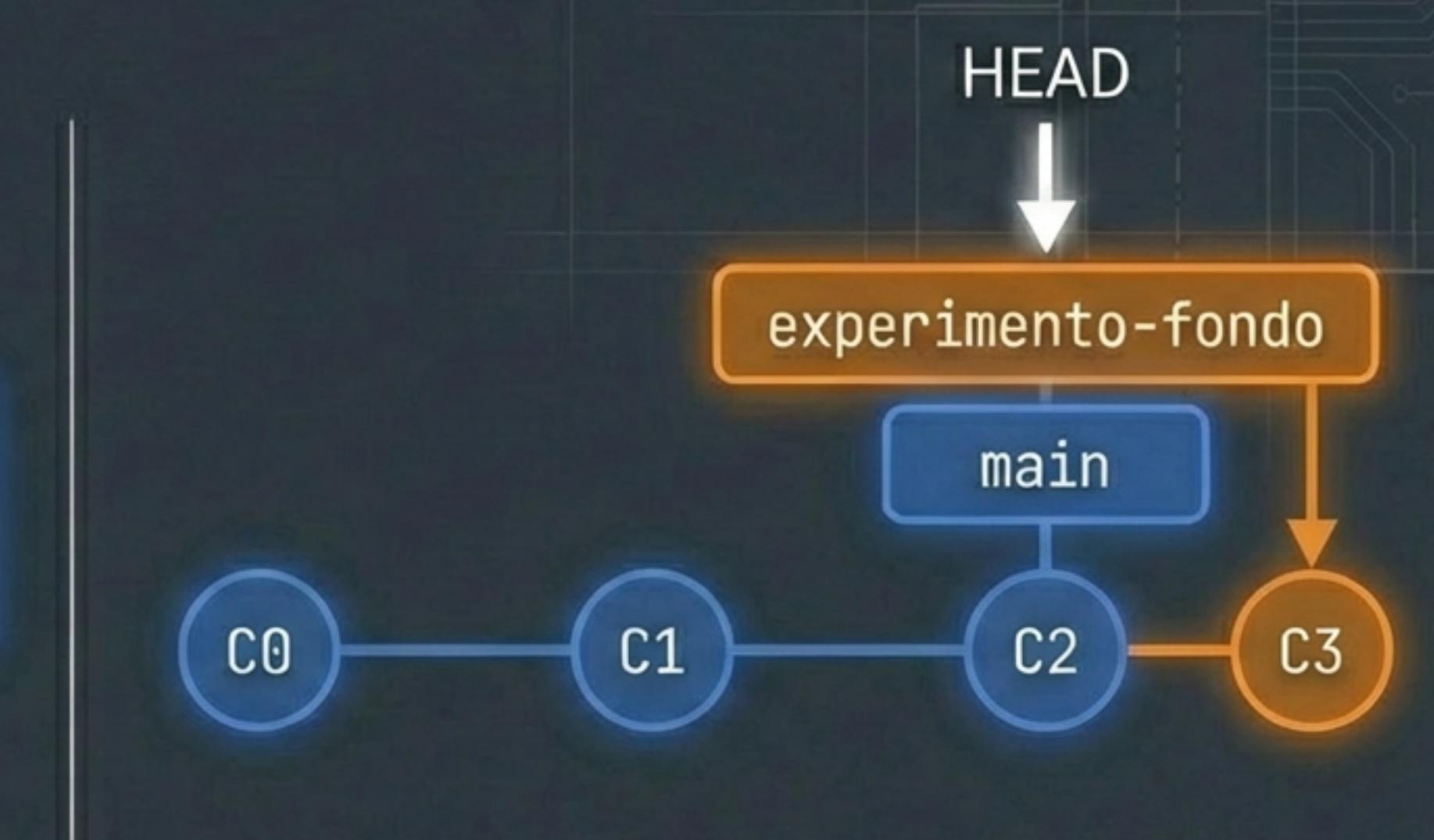


El historial se ve igual, pero tu lugar de trabajo ha cambiado.

La Divergencia

Modificamos estilos.css (fondo rojo) y hacemos commit.

```
$ git add estilos.css  
$ git commit -m "Fondo cambiado a rojo"
```



main está 'congelada' en el pasado. Tu experimento avanza seguro en su propia línea de tiempo.

Vuelta a la Realidad

El experimento funcionó. Volvemos al destino para prepararnos.

```
$ git checkout main
```



¡Magia! Al volver a main, el cambio de color desaparece de tu archivo. Tus cambios siguen guardados en la otra rama, esperando ser fusionados.

La Fusión Perfecta (Fast-Forward)

Como main no se movió mientras trabajabas, Git simplemente desliza el puntero hacia adelante.

```
$ git merge experimento-fondo  
Updating... Fast-forward
```



Una vez fusionado, borramos la rama con `git branch -d experimento-fondo`.

Cuando las Historias Divergen

Escenario: Merge Recursivo

Tú creas la rama
funcionalidad-X (botón).

Mientras tanto, tu compañero
actualiza main (logo).

Un Fast-Forward es imposible aquí.



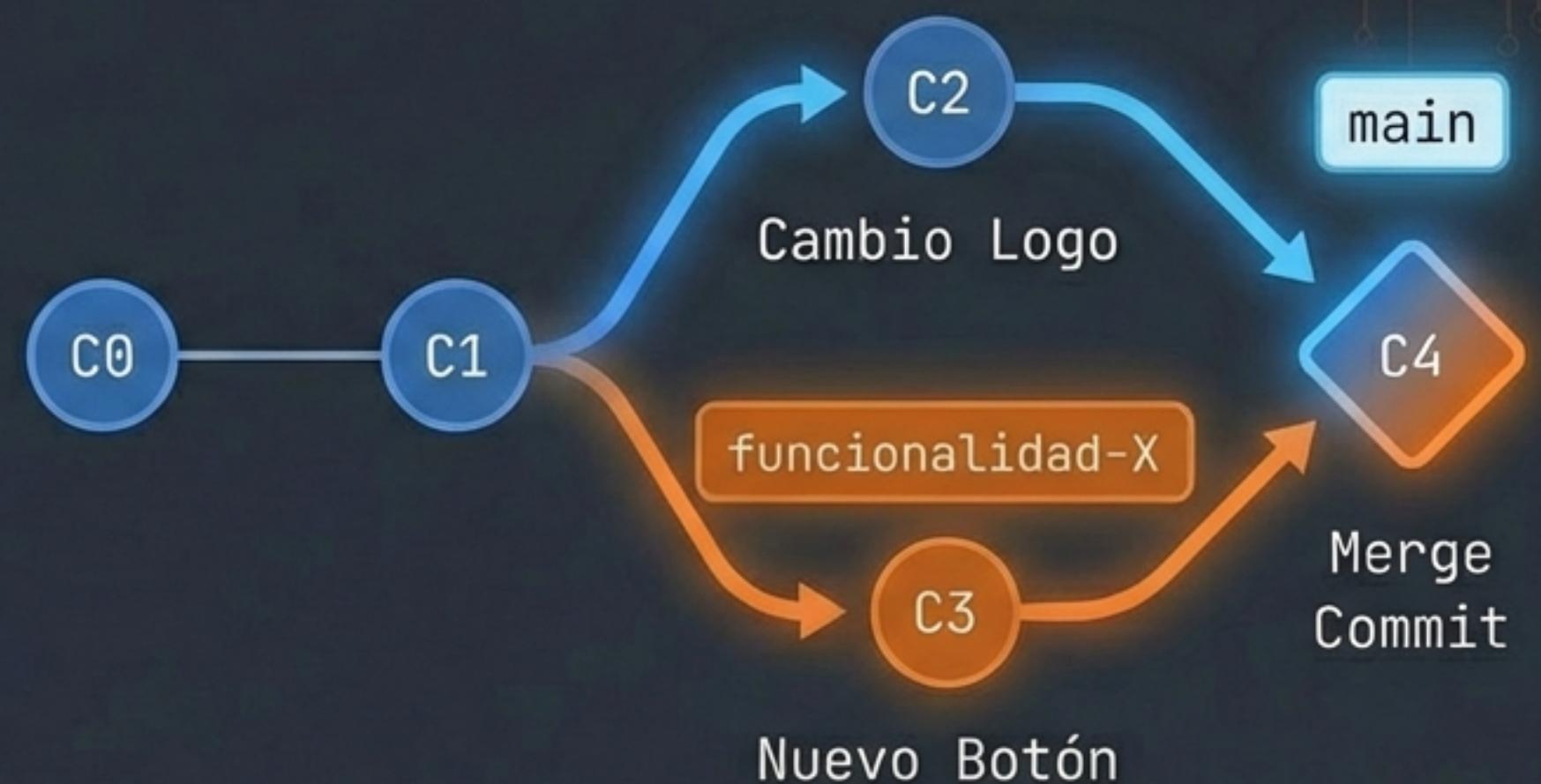
El “Merge Commit”

Git crea un nuevo commit específicamente para unir las dos historias.

Git abrirá tu editor (Vim/Nano) para el mensaje:

Tip Vim: :wq + Enter

Tip Nano: Ctrl+0, Enter, Ctrl+X



¡Conflict! Colisión de Universos



Timeline A



Timeline B



Acción: `git merge cambio-titulo`

Resultado: **⚠ CONFLICT** (content): Merge conflict in `titulo.txt`

Git detiene el proceso. Requiere intervención humana.

Anatomía de un Conflicto



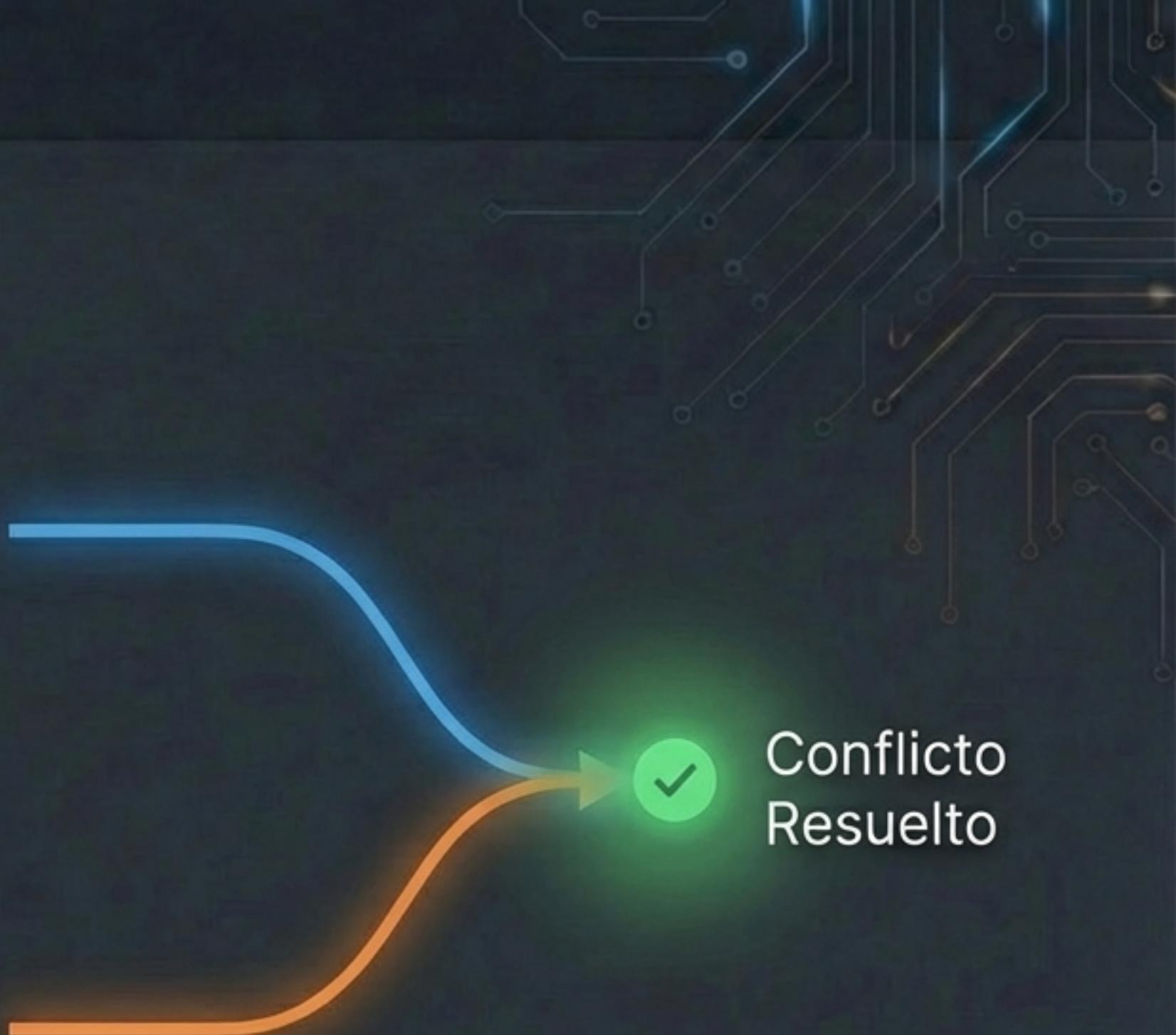
Lo que tienes en tu rama
actual (main)

La línea divisoria

Lo que viene de la rama
externa

Resolución y Paz

1. **Edita:** Borra las marcas (<<<, ===, >>>) y elige el texto final.
2. **Prepara:** git add titulo.txt
(Marca el conflicto como resuelto).
3. **Confirma:** git commit
(Cierra la fusión).



Resumen Maestro

Ver

git branch

git lga (alias)

Crear/Mover

git switch -c <rama>

Fusionar

git merge <rama> (Desde el destino)

Limpiar

git branch -d <rama>

Recurso Recomendado: "Learn Git Branching" (Web Interactiva)

Consejo: Las ramas son baratas y rápidas. ¡Úsalas para todo!