

# Git Básico

---

**Arturo Silvelo**

Try New Roads

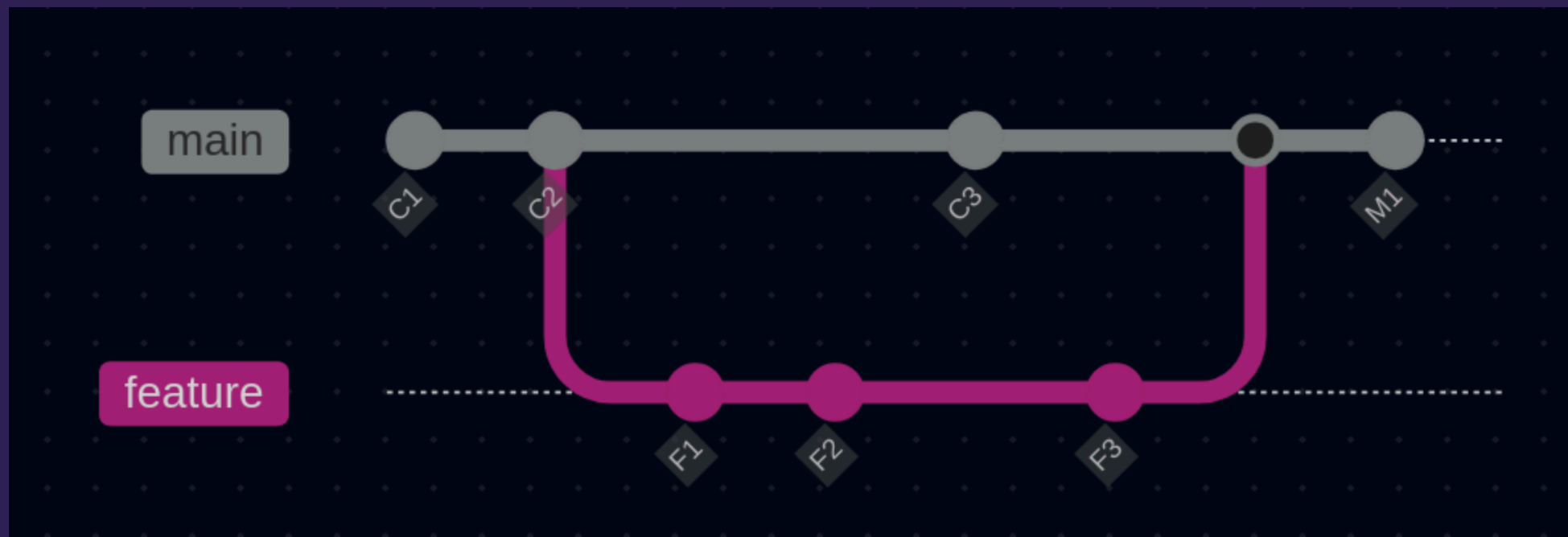
# Ramas

## ¿Qué es una rama?

---

Una **rama** (branch) en Git es una línea de desarrollo independiente que permite:

- Trabajar en nuevas características sin afectar el código principal
- Experimentar con cambios de forma segura
- Colaborar en paralelo con otros desarrolladores
- Mantener un historial limpio y organizado



## Ventajas de usar ramas

---

- **Aislamiento:** Los cambios en una rama no afectan otras ramas
- **Experimentación:** Puedes probar ideas sin riesgo
- **Colaboración:** Múltiples desarrolladores pueden trabajar simultáneamente
- **Organización:** Cada característica o bug fix tiene su propia rama
- **Historial limpio:** Facilita el seguimiento de cambios

## Crear una rama

---

Para crear una nueva rama se pueden usar diferentes comandos

- Crear solo la rama (sin cambiar a ella)

```
git branch feature
```

Para cambiar a la rama creada:

```
git switch feature  
# ó  
git checkout feature
```

- Crear la rama y cambiar a ella:

```
git switch -c feature  
git checkout -b feature
```

- Crear rama desde un commit específico:

```
git branch feature abc1234  
git switch -c feature abc1234
```

## Listar ramas

---

- Ver ramas locales:

```
git branch
```

La rama actual tendrá un asterisco al inicio.

- Ver todas las ramas (locales y remotas):

```
git branch -a
```



- Ordenar por fecha de modificación:

```
git branch --sort=-committerdate
```

- Ver información detallada:

```
git branch -v
```

# Trabajando Con Ramas

## Escenario inicial

---

Tu equipo está trabajando en un proyecto y necesitan manejar múltiples tareas simultáneamente.

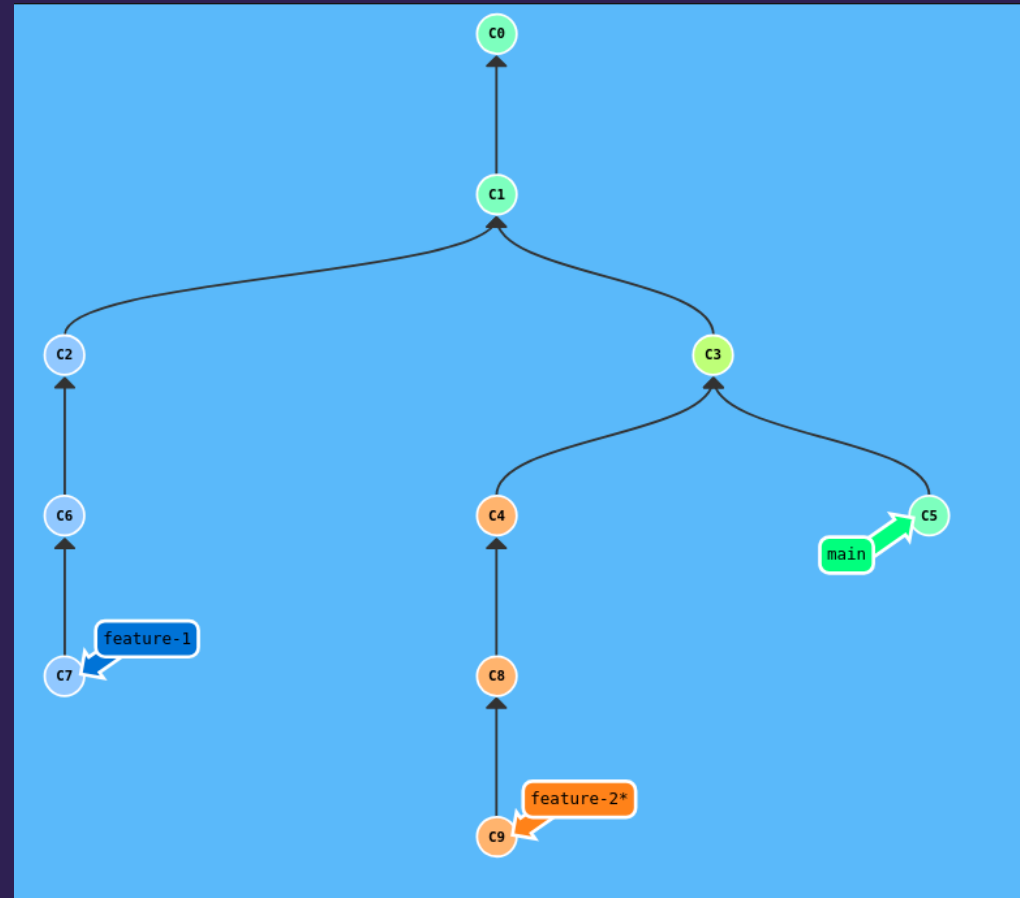
### Tareas a realizar:

1. **Nueva característica:** Te asignan desarrollar `feature-1`
2. **Bug crítico:** Durante el desarrollo encuentras un error que necesita solución inmediata
3. **Desarrollo paralelo:** Un compañero debe crear `feature-2` al mismo tiempo
4. **Segundo bug:** Se reporta otro error en la aplicación principal mientras todos trabajan

## Cargar el nivel

---

1. **Learn Git Branching**
2. Escribimos `import level`
3. Copiamos el fichero `ejercicios_1.json` y lo pegamos.



# Fusionar Ramas

## ¿Qué es fusionar (merge)?

---

**Fusionar** es el proceso de **integrar los cambios** de una rama en otra rama.

- Los commits de la rama origen se incorporan a la rama destino
- Se crea un historial unificado
- Permite combinar el trabajo de diferentes desarrolladores
- Mantiene la trazabilidad de los cambios

- Comando básico de fusión

```
git merge <rama-origen>
```

Ejemplo:

```
# 1. Cambiar a la rama destino  
git switch main  
  
# 2. Fusionar la rama feature  
git merge feature-1
```

 **Importante:** Siempre debes estar en la rama destino antes de hacer merge



## Tipos de merge

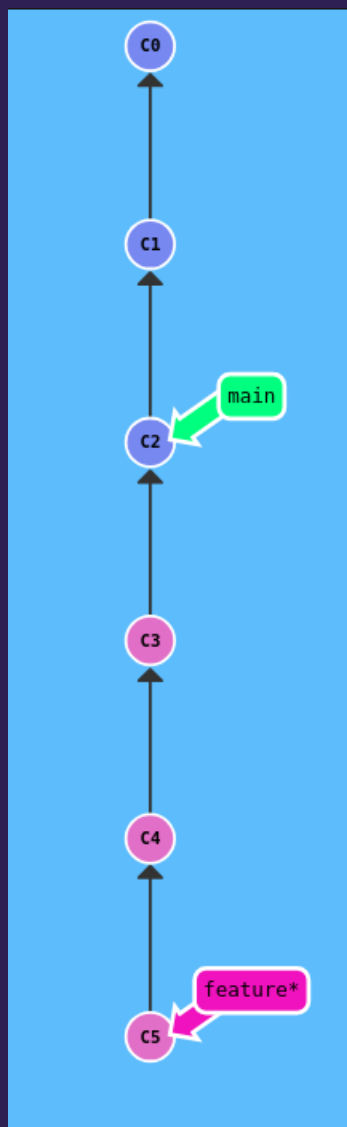
---

## Fast-forward

---

- Cuando no hay commits nuevos en la rama destino
- Simplemente mueve el puntero hacia adelante

```
git merge --ff-only my-branch
```



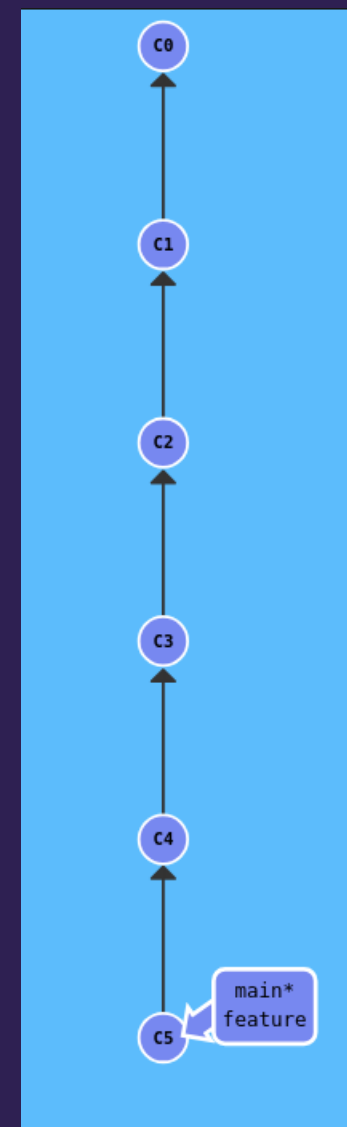
```
# Creamos la nueva rama
git switch -c feature

# Añadimos cambios
git commit
git commit
git commit

# Volvemos a la rama principal
git switch main

# Fusionamos las ramas
git merge --ff-only feature

# ó
git merge feature
```

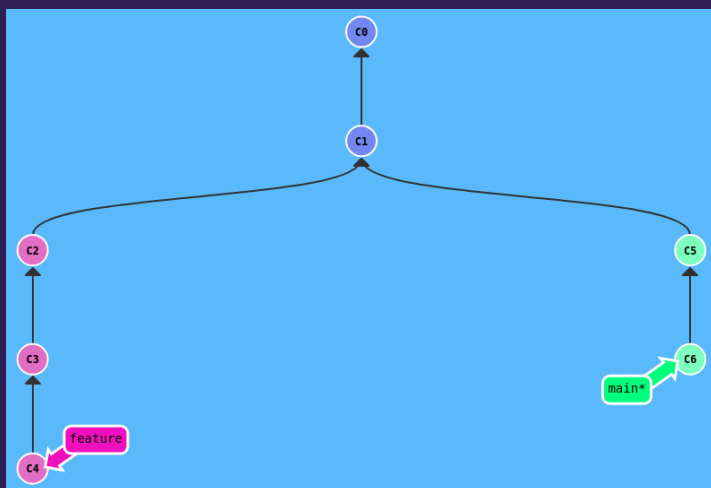


## No Fast-forward

---

- Cuando ambas ramas tienen commits nuevos
- Crea un commit de fusión que une ambas historias
- Preserva la estructura de ramas en el historial

```
git merge --no-ff my-branch
```



```
# Creamos la nueva rama  
git switch -c feature
```

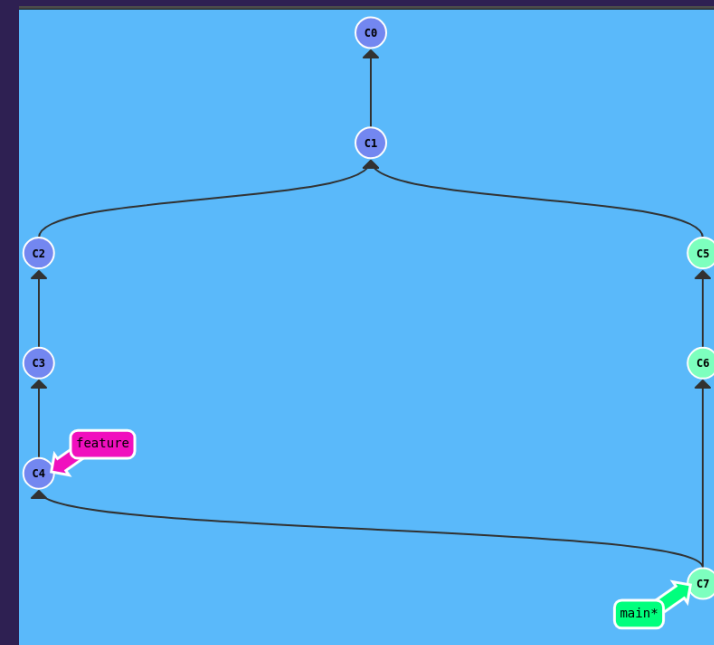
```
# Añadimos cambios  
git commit  
git commit  
git commit
```

```
# Volvemos a la rama principal  
git switch main
```

```
# Añadimos cambios  
git commit  
git commit  
git commit
```

```
# Fusionamos las ramas  
git merge --no-ff feature
```

```
# ó  
git merge feature
```

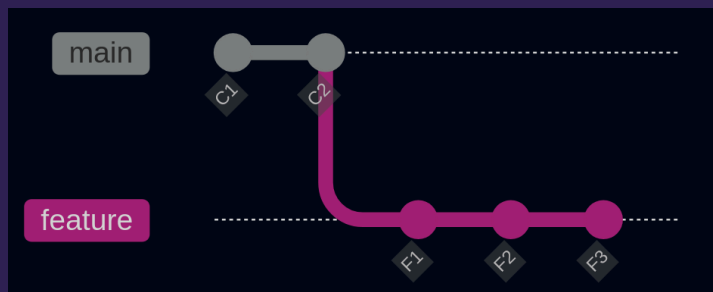


## Squash Merge

---

- Combina todos los commits de una rama en un solo commit
- Crea un historial más limpio y lineal
- Útil cuando la rama tiene muchos commits pequeños o experimentales

```
git merge --squash feature  
git commit -m "Add complete feature X"
```



```
# Creamos la nueva rama
git switch -c feature

# Añadimos múltiples commits
git commit -m "F1"
git commit -m "F2"
git commit -m "F3"

# Volvemos a main
git switch main

# Squash merge: combina todos en uno
git merge --squash feature
git commit -m "Feature Complete"
git commit -m "C3"
```

