

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

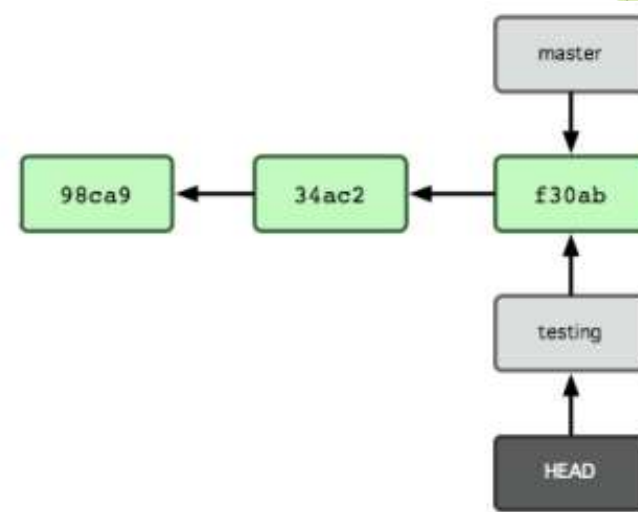
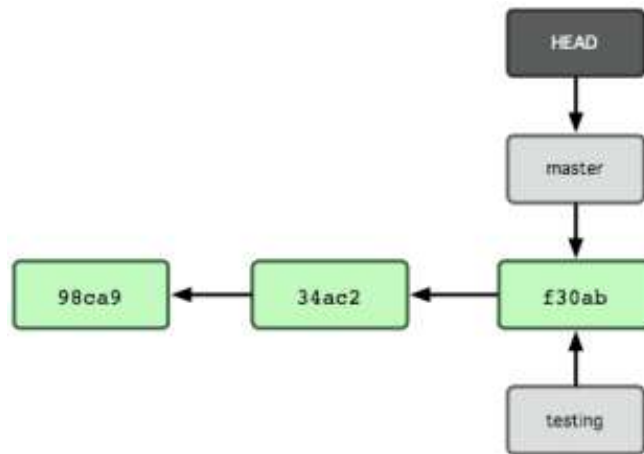
GIT - RAMAS

# Ramas en GIT

- ▶ Por defecto existe la rama(branch) master
- ▶ Similar al trunk de SVN desde el punto de vista estructural.
- ▶ NO ES IGUAL SEMÁNTICAMENTE AL trunk
  - ▶ El branch master se considera que contiene el código que se puede poner en producción.
  - ▶ El branch master es una referencia
- ▶ Listar branches / Averiguar branch actual:
  - ▶ `$ git branch [-v] -a`
  - ▶ La referencia HEAD apunta al branch actual

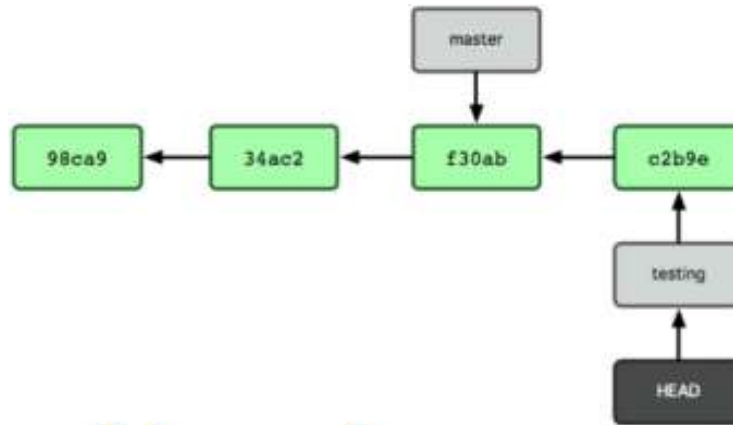
# Ramas en GIT

- ▶ Crear un branch (local)
  - ▶ `git branch <nombre branch>`
  - ▶ Crea un branch a partir del branch actual
- ▶ Pasar a trabajar a otro Branch
  - ▶ `git checkout <nombre branch>`
- ▶ Los dos comandos anteriores a la vez:
  - ▶ `git checkout -b <nombre branch>`

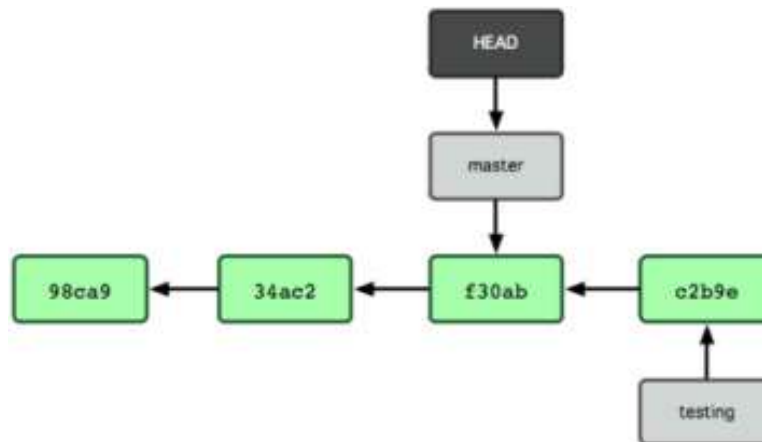


# Ramas en GIT

- Al hacer commit se realizarán sobre el branch activo

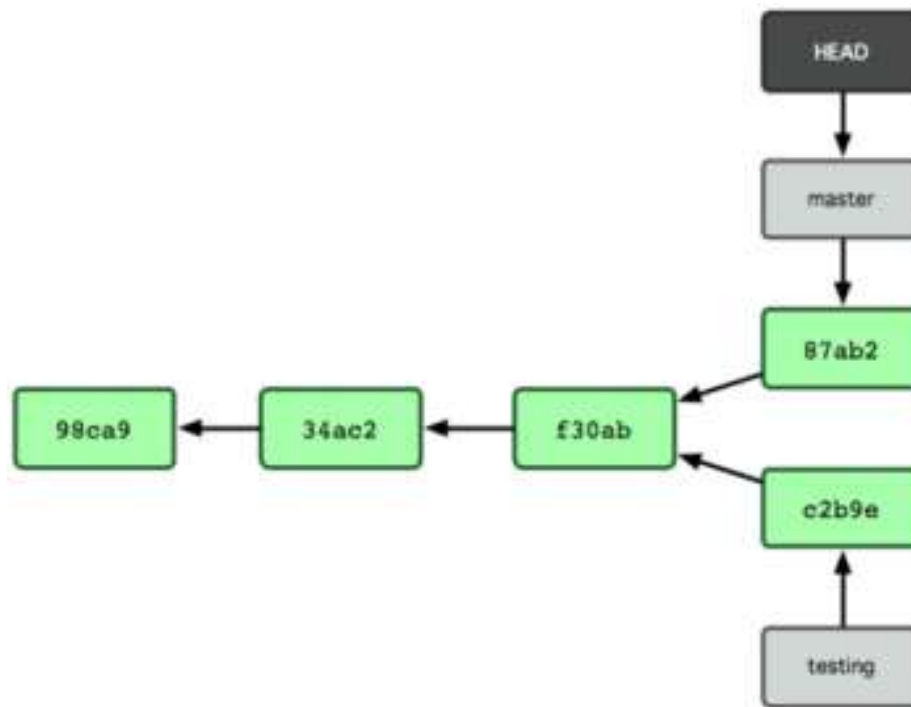


- Podemos volver al branch master cuando queramos



# Ramas en GIT

- ▶ Al modificar el branch master la estructura del repositorio cambia:
  - ▶ La historia de los branches diverge
  - ▶ Es necesario hacer un “merge” (reconciliar) los cambios



# Ramas en GIT

## FLUJO DE TRABAJO CON BRANCHES

1. Crear un branch cuando tengo que hacer una tarea o quiero experimentar algo.
2. Trabajar sobre el branch (desarrollar, hacer pruebas)
3. Nos aseguramos que la copia de trabajo está limpia. Es decir, que no hay ningún cambio pendiente
4. Actualizamos nuestro branch de trabajo con los cambios que haya habido en master
5. Cuando estamos contentos con el trabajo hacemos un “merge” del trabajo en el branch master

# Ramas en GIT

- ▶ ¿Cómo hacemos el merge?
  - ▶ Checkout del branch donde vamos a integrar los cambios
    - ▶ `$ git checkout master`
  - ▶ Integramos los cambios
    - ▶ `$ git merge tarea`
- ▶ Cuando se realizan los merges es posible que haya que resolver conflictos
- ▶ Conflictos: modificaciones sobre un mismo archivo que git no sabe resolver

# Ramas en GIT

- ▶ ¿Cómo averiguo los branches que hay?
  - ▶ `$ git branch [-a -v]`
  - ▶ El branch activo aparece con un '\*'
- ▶ ¿Cómo averiguo que branches NO están integrados con el branch activo?
  - ▶ `$ git branch --no-merged`
- ▶ ¿Cómo averiguo que branches SÍ están integrados con el branch activo?
  - ▶ `$ git branch --merged`
- ▶ Una vez que un branch está integrado puedo eliminarlo si lo deseo
  - ▶ `$ git branch -d <branch>`



# Ejercicios - Ramas en GIT

- ▶ 1. Probar en clase el ejercicio sobre ramas que aparece en el siguiente enlace:

[https://www.jesusamieiro.com/wp-content/uploads/2014/03/20141009\\_Git\\_Ejemplo\\_GPUL\\_UDC.pdf](https://www.jesusamieiro.com/wp-content/uploads/2014/03/20141009_Git_Ejemplo_GPUL_UDC.pdf)

- ▶ 2. Ejercicio para ramificar y fusionar:

<https://git-scm.com/book/es/v1/Ramificaciones-en-Git-Procedimientos-b%C3%A1sicos-para-ramificar-y-fusionar>

- ▶ 3. Ejercicio con ramas remotas