



# Práctica GIT 2

## 1.-Introducción

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

## 2.-Enunciado

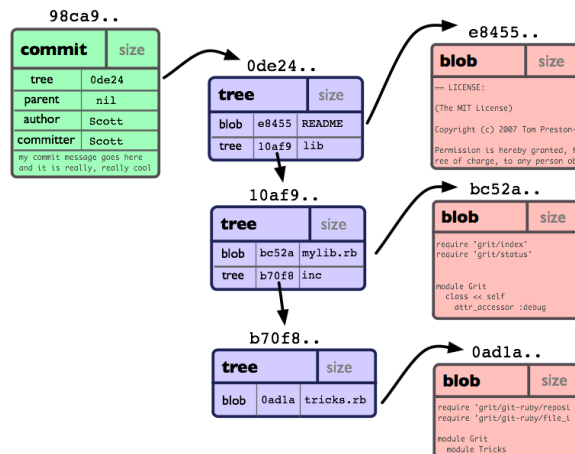
Para realizar estos ejercicios es necesario que hagas uso de la cuenta de GitHub que has creado en clase. Por cada ejercicio es necesario que crees un repositorio en tu cuenta.

### 2.1.- Individual

#### 2.1.1.- Ejercicio 1

Para entender cómo funciona GIT es necesario saber cómo lo hace por dentro. Para ello se necesita conocer estos términos: **SHA**, **blob**, **tree**, **commit**, **tag**.

Define estos términos conociendo la información que alberga cada uno y como se relacionan entre ellos para hacer funcionar un repositorio. Posteriormente explica lo que ve en esta imagen detallando información.



Para facilitar la búsqueda de información se sugiere utilizar el siguiente enlace:  
[http://shafiul.github.io/gitbook/1\\_the\\_git\\_object\\_model.html](http://shafiul.github.io/gitbook/1_the_git_object_model.html)

### 2.1.2.- Repositorio remoto

Lo ideal para realizar este ejercicio, es jugar con dos usuarios para cómo se notifican los cambios. Podéis decirle a un compañero que os cree un repositorio remoto y os pase el enlace, paso 1).

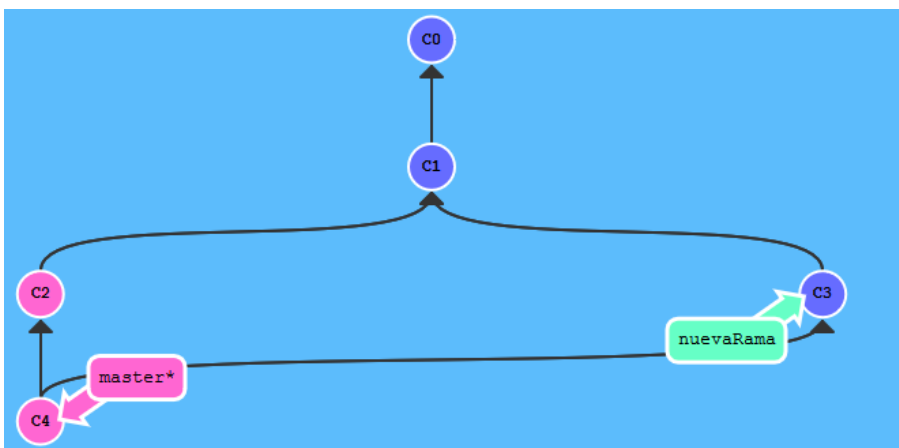
1. Colaborar en el repositorio remoto, crea uno para ello.
2. Clónalo su repositorio.
3. Añadir varios ficheros con algún contenido.
4. Añadir los cambios a la zona de intercambio temporal.
5. Hacer un commit con el mensaje “subida ejercicio avanzado.”
6. Subir los cambios al repositorio remoto.
7. Haz una bifurcación del repositorio remoto anterior.
8. Clonar el repositorio de nuevo.
9. Crear una nueva rama y activarla.
10. Añade varios ficheros con contenido.
11. Añadir los cambios a la zona de intercambio temporal.
12. Hacer un commit con el mensaje “cambios en la nueva rama.”
13. Subir los cambios de la rama creada en el punto 9 al repositorio remoto.
14. Hacer un *Pull Request* de los cambios en la rama.

### 2.1.3.- Programación en GIT

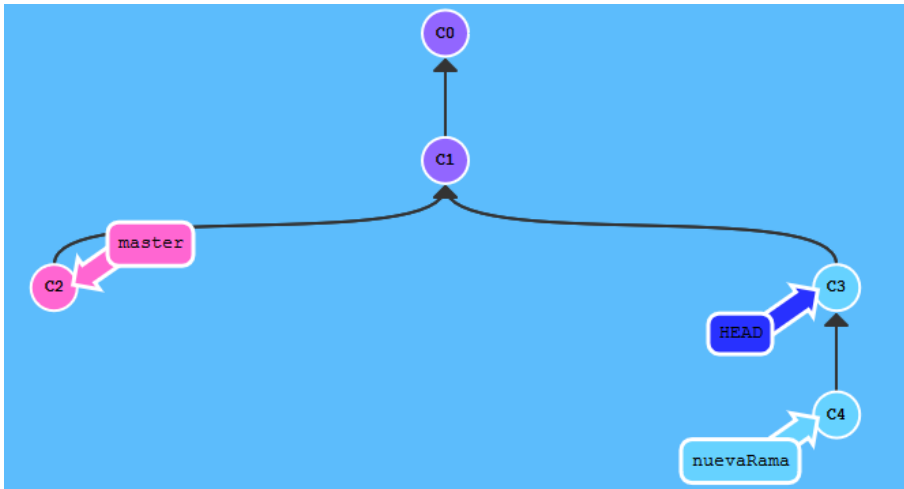
Existen múltiples herramientas de aprendizaje de GIT, en este ejercicio, vamos a utilizar [learninggit](#). Esta herramienta nos va a permitir profundizar en los comandos que hemos visto en clase: clone (para hacer commit en repositorio remoto tienes que hacer git fakeTeamwork cantidad\_commits), merge, fetch, push, pull, [branch](#), [rebase](#), [reset](#), [revert](#), ...

En este ejercicio tienes que indicar la serie de pasos a seguir para obtener los resultados de la imagen en cuestión:

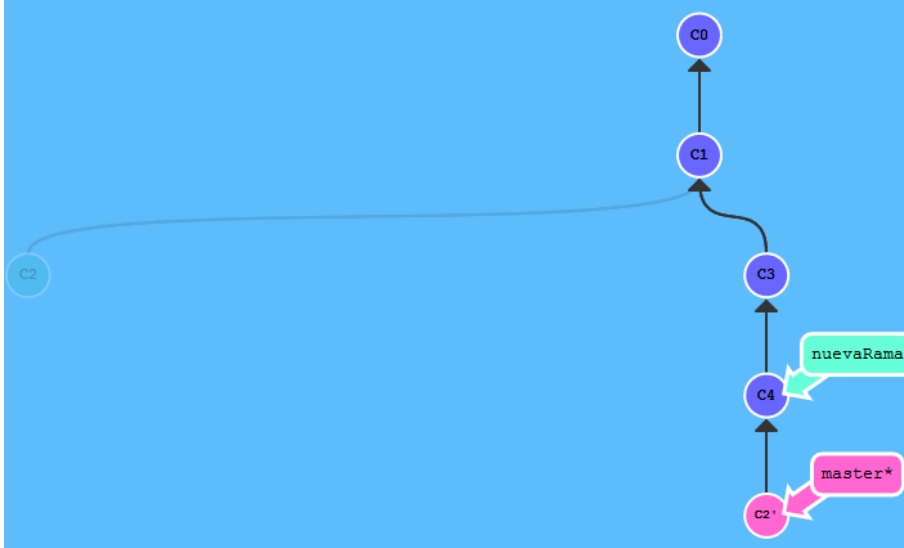
1)



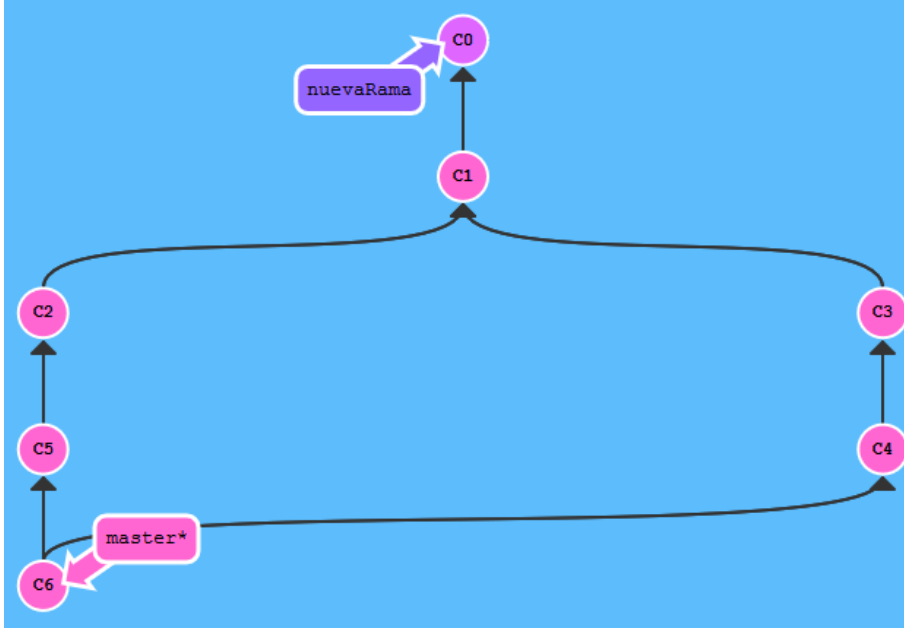
2)



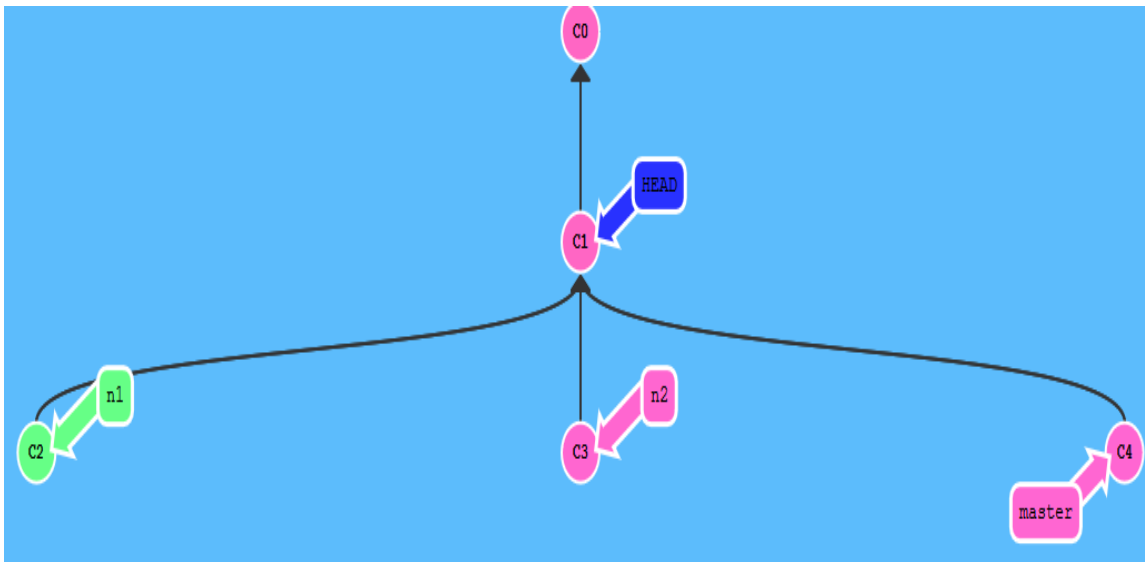
3)



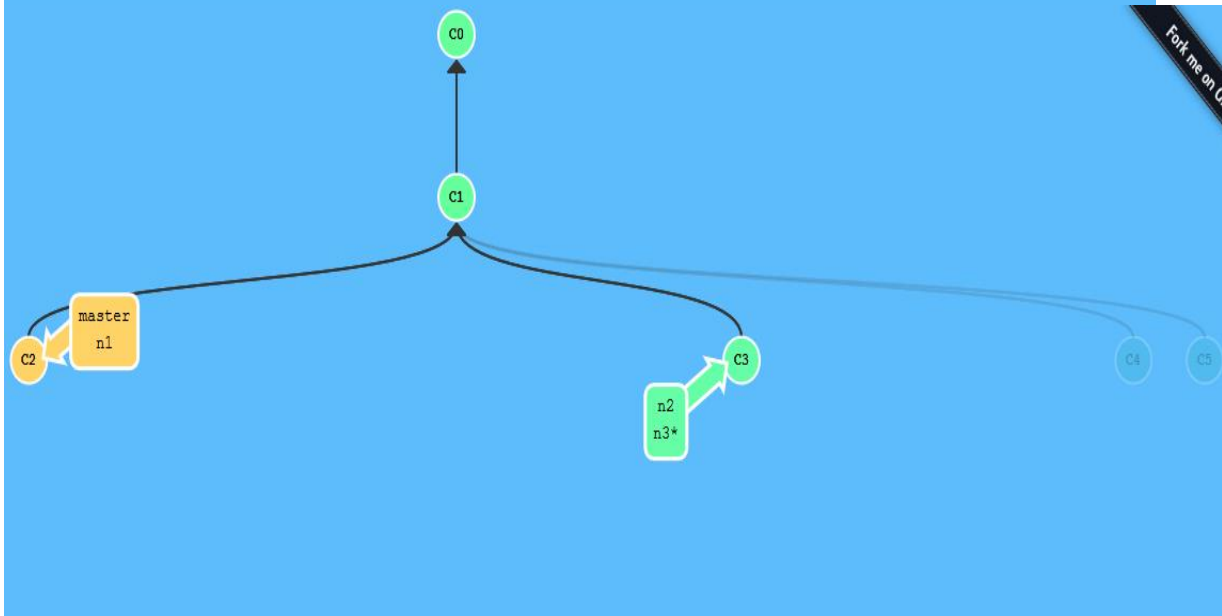
4)



5)



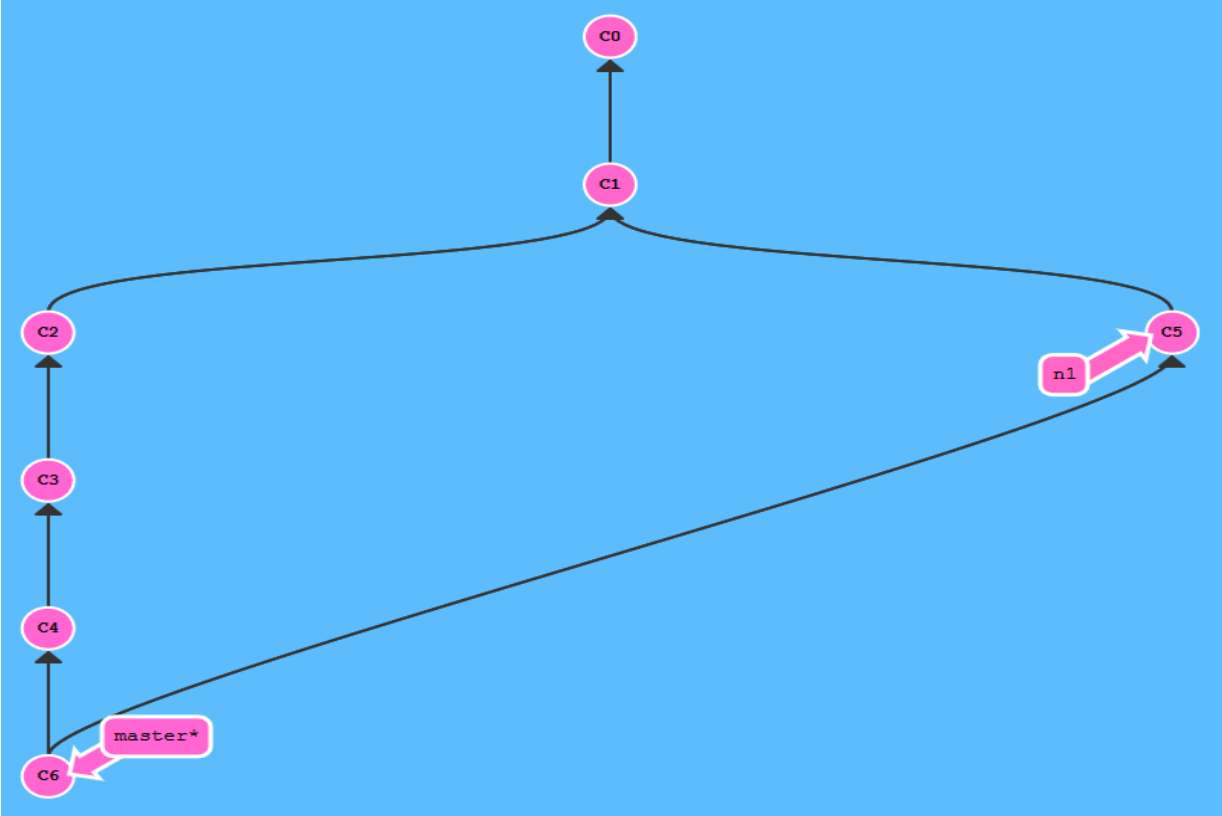
6)



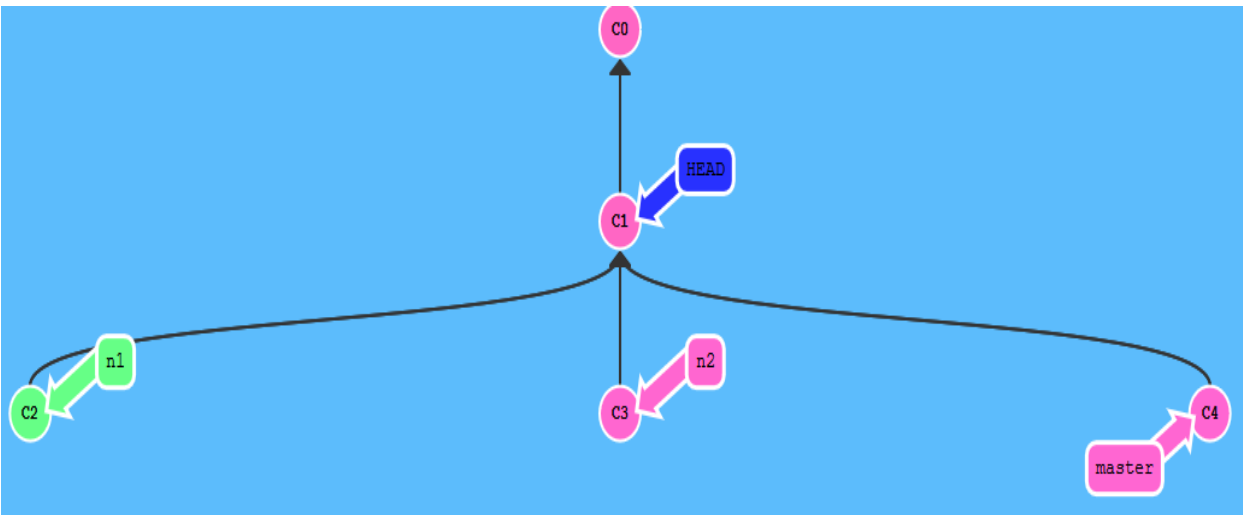
7)



8)



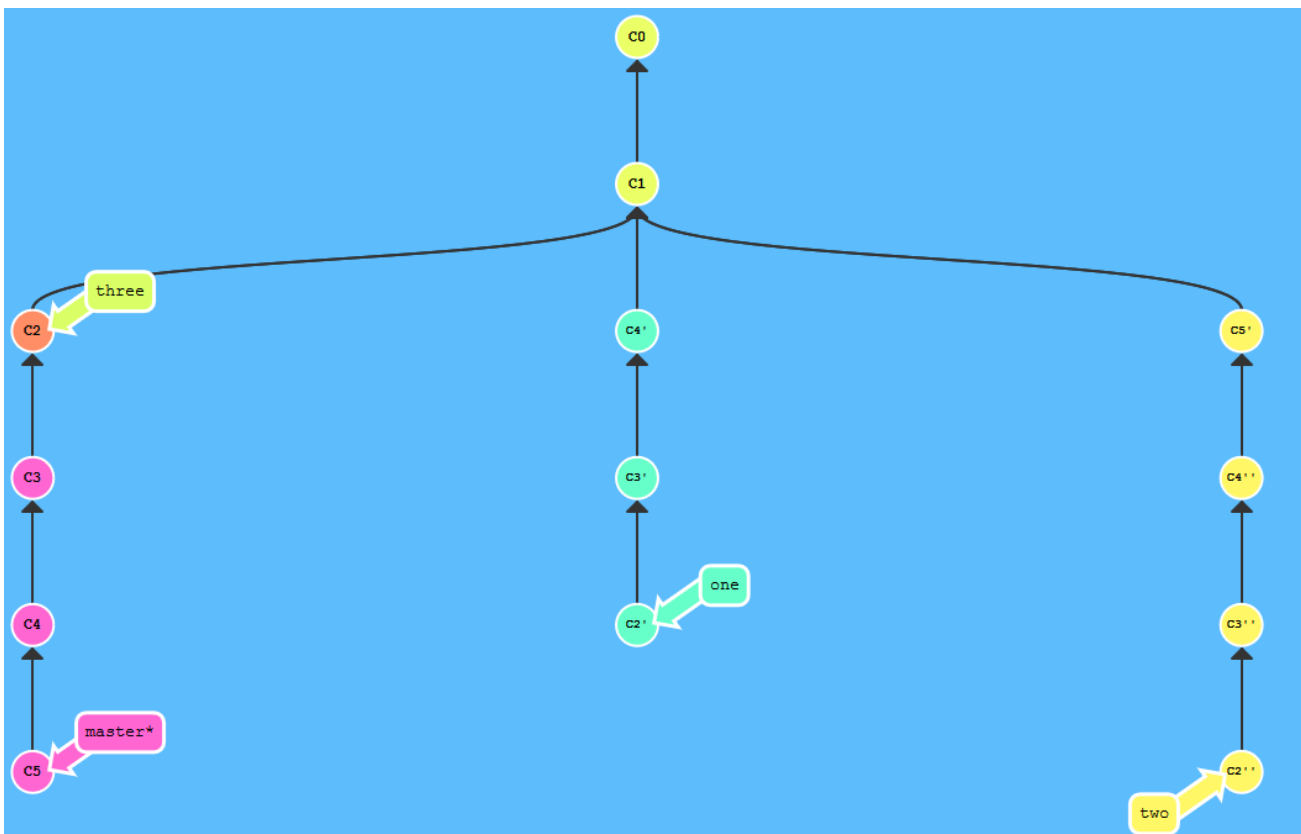
9)



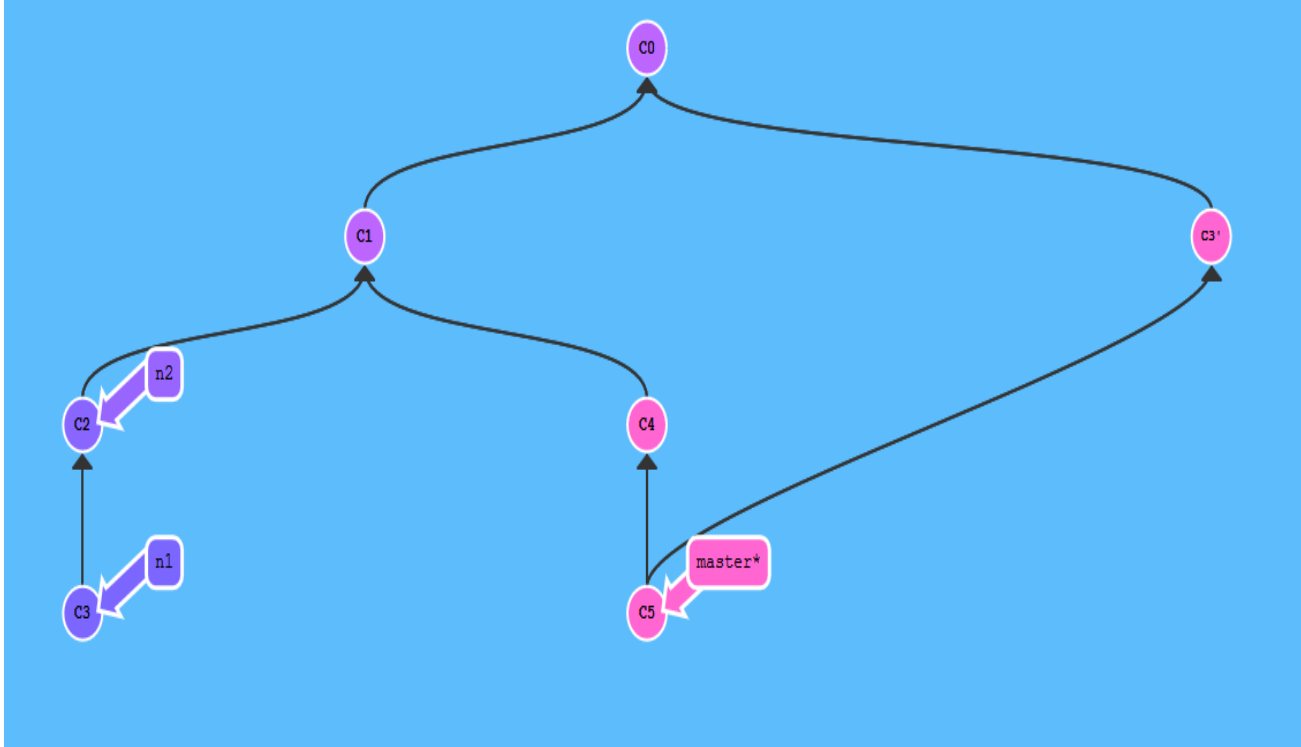
10)



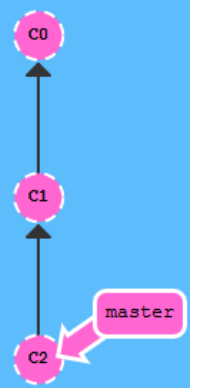
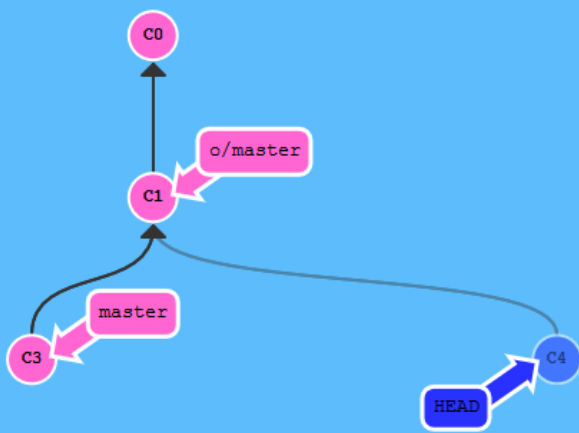
11)



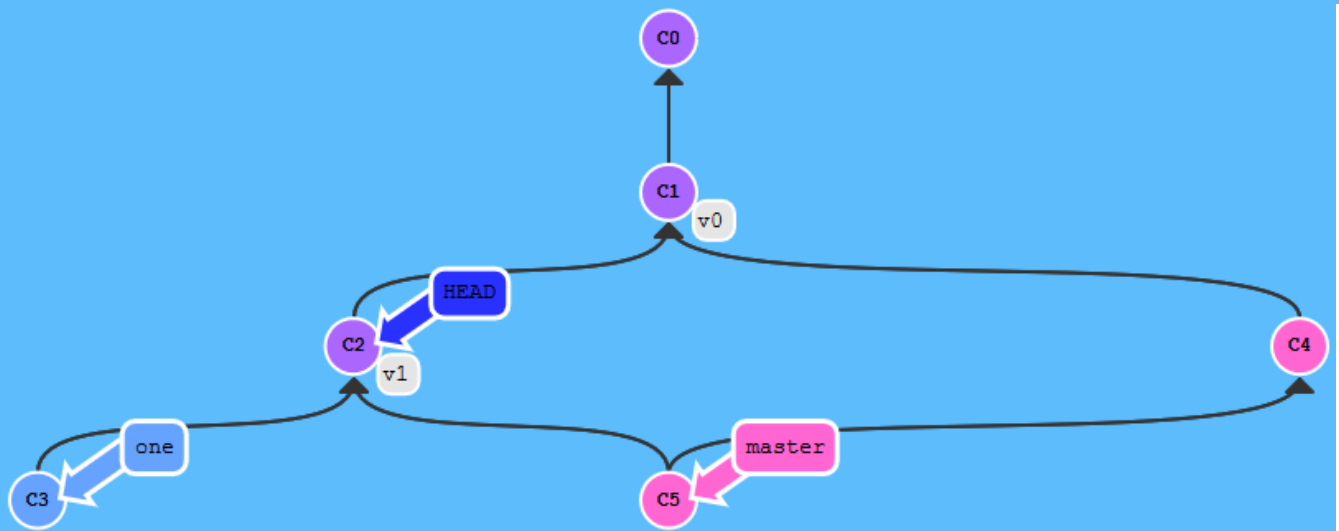
12)



13)

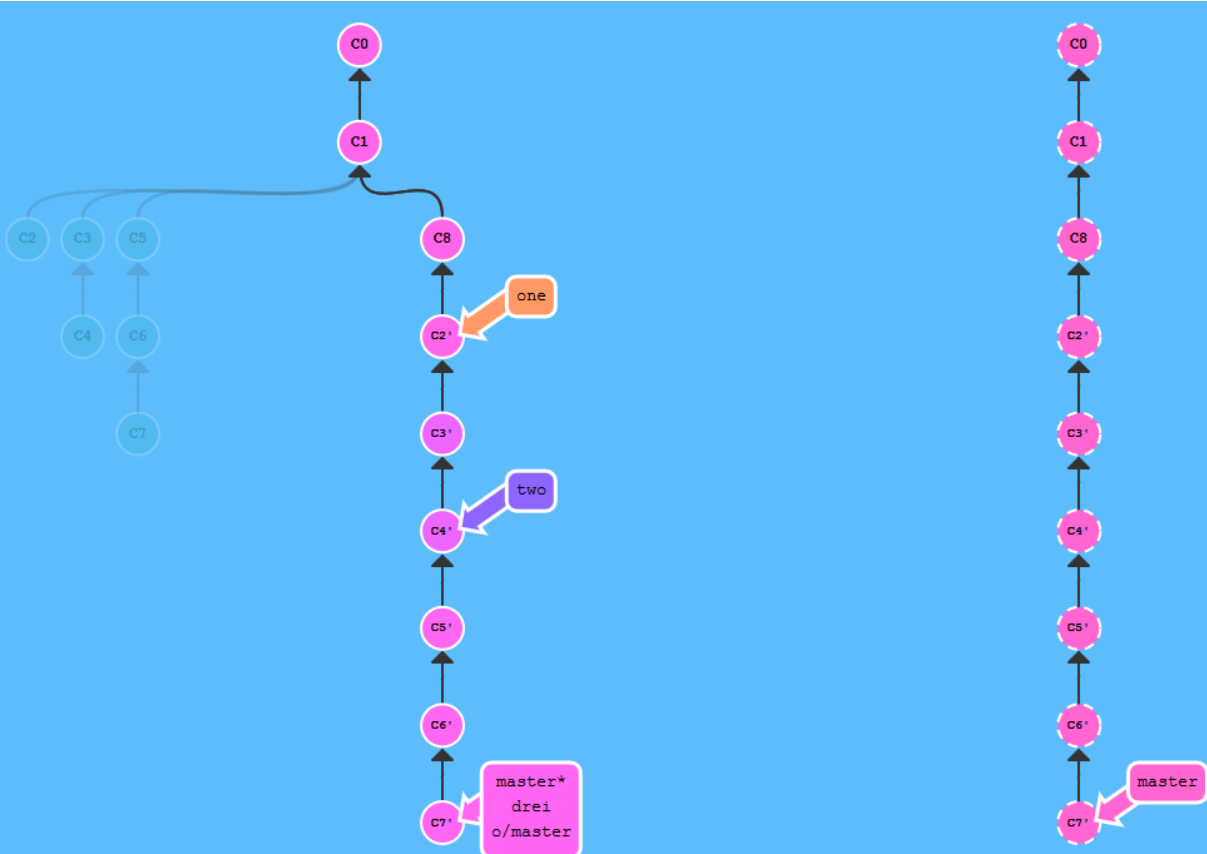


14)





15)



16)

