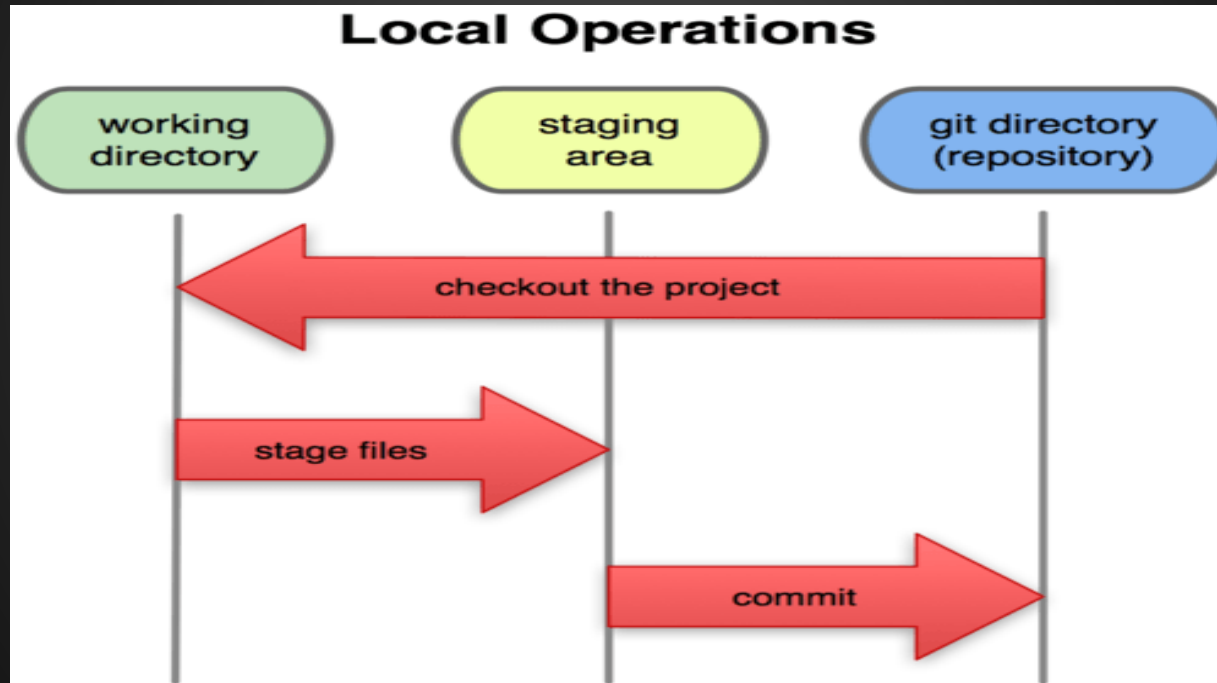


# Git - Tutorial

# Controle de Versão

O controle de versão é um sistema que registra as mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo de forma que você possa recuperar versões específicas.

# Tipos de Arquivos



# Instalando o Git

```
$ yum install git-core
```

```
$ apt-get install git
```

```
http://msysgit.github.com
```

# Configurações Básicas

```
$git config --global user.name "John Doe"
```

```
$ git config --global user.email: john@example
```

```
$ git config --global core.editor emacs
```

# Criando Um repositório

```
$ git init
```

```
$ git add
```

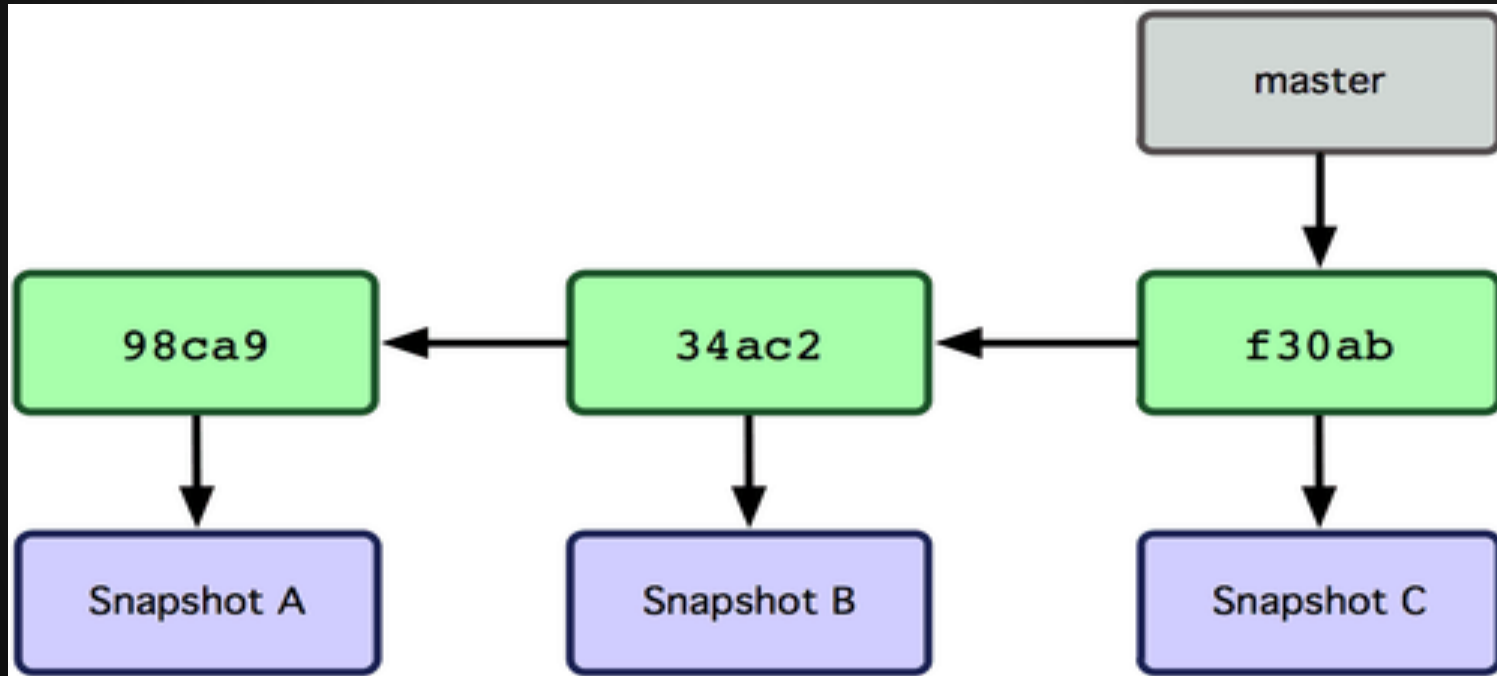
```
$ git commit
```

# Clonar repositório existente

```
$ git clone git://github.com/schacon/grit.git
```

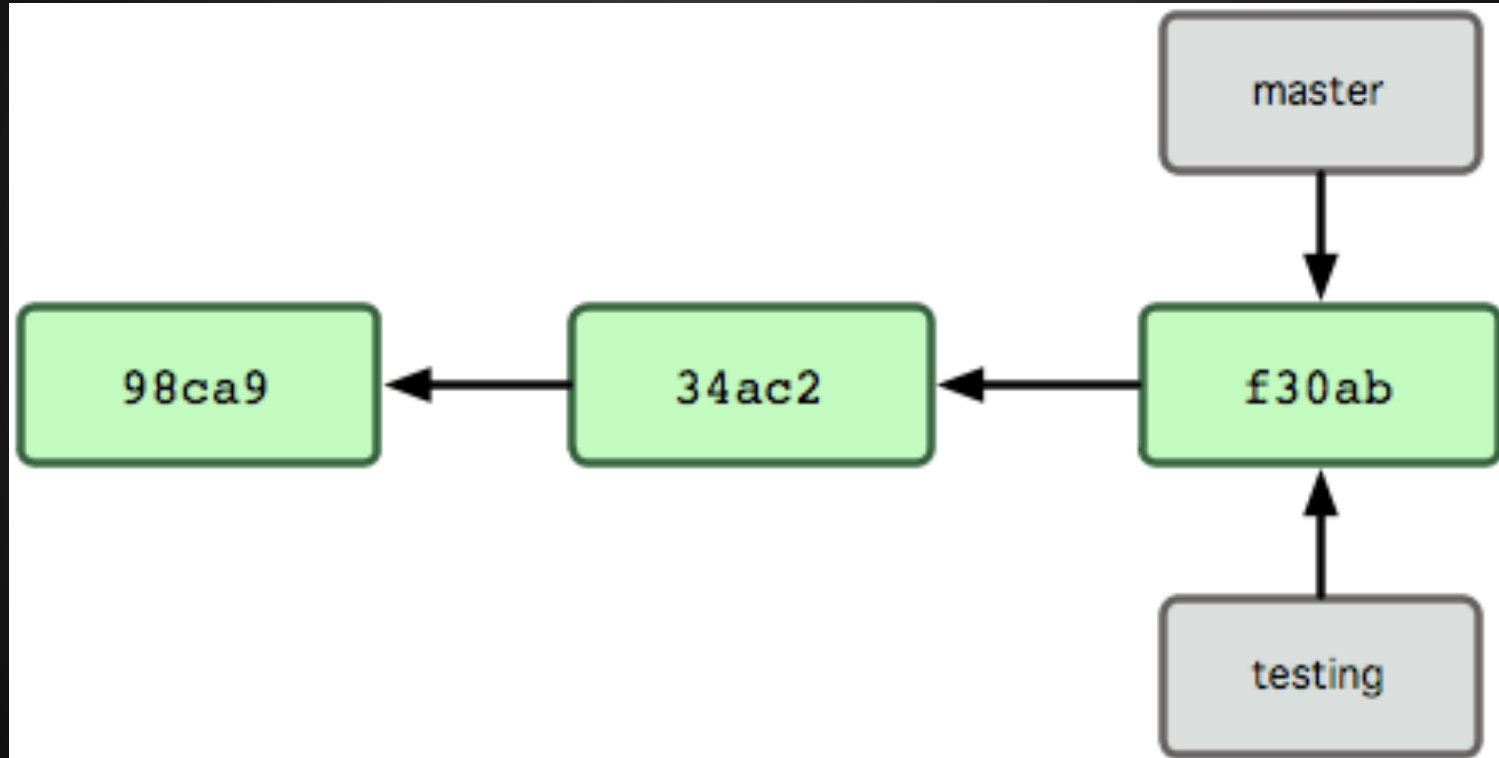
```
$ git clone git://github.com/schacon/grit.git  
mygit
```

# Conceito de Branch

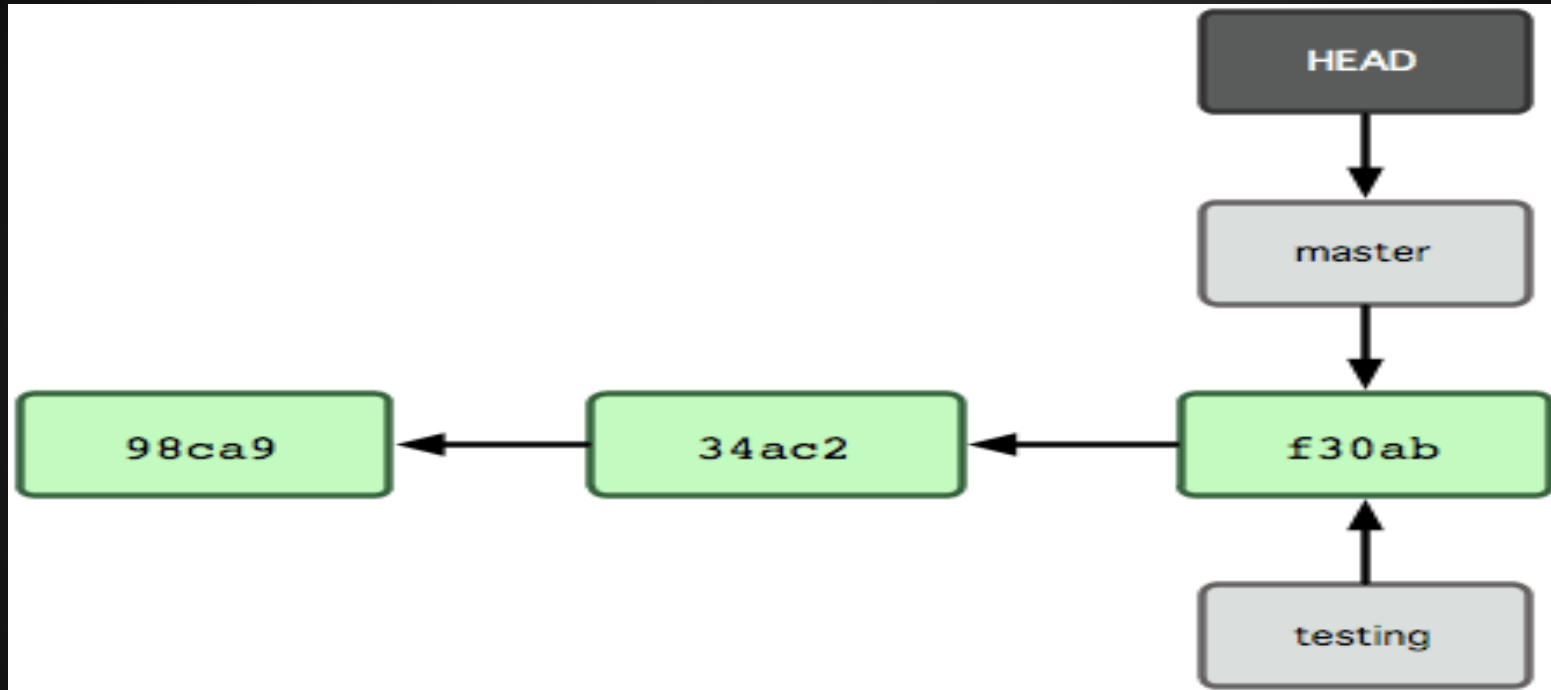




```
$ git branch testing
```

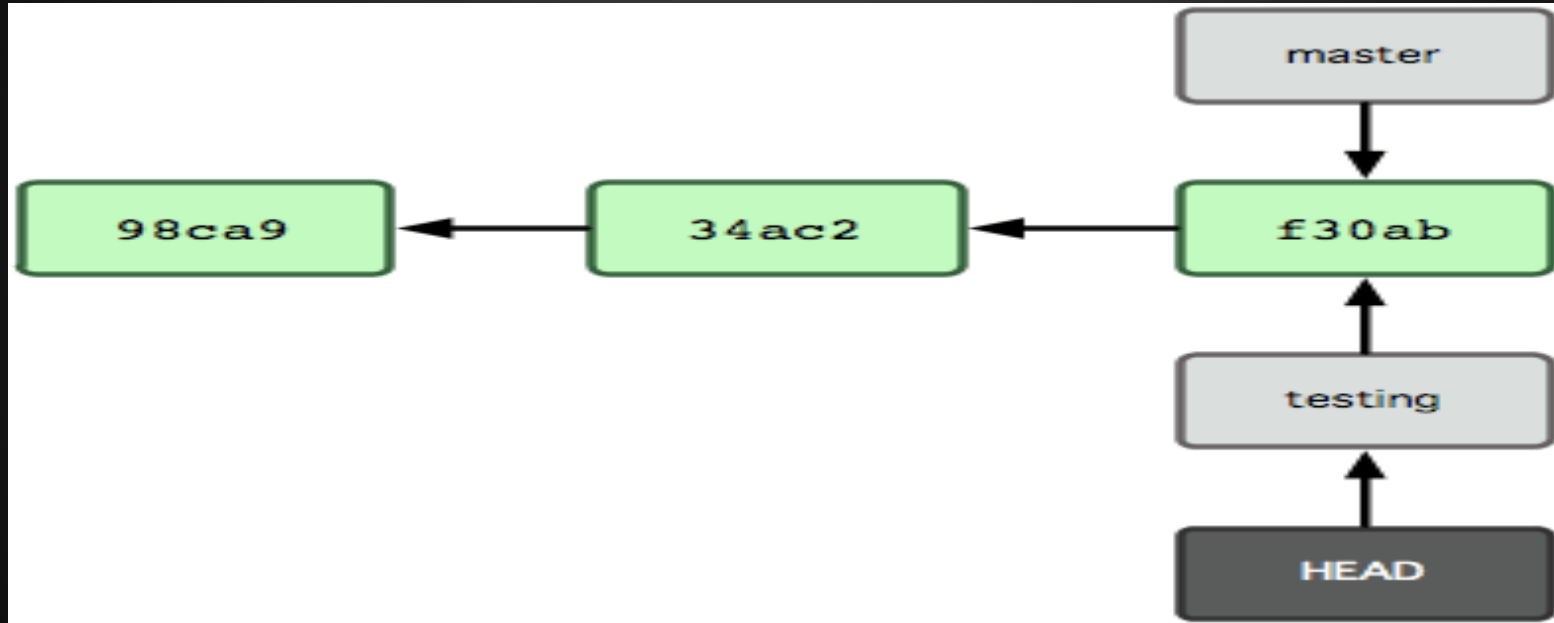


Múltiplos branches apontando para o histórico de commits.



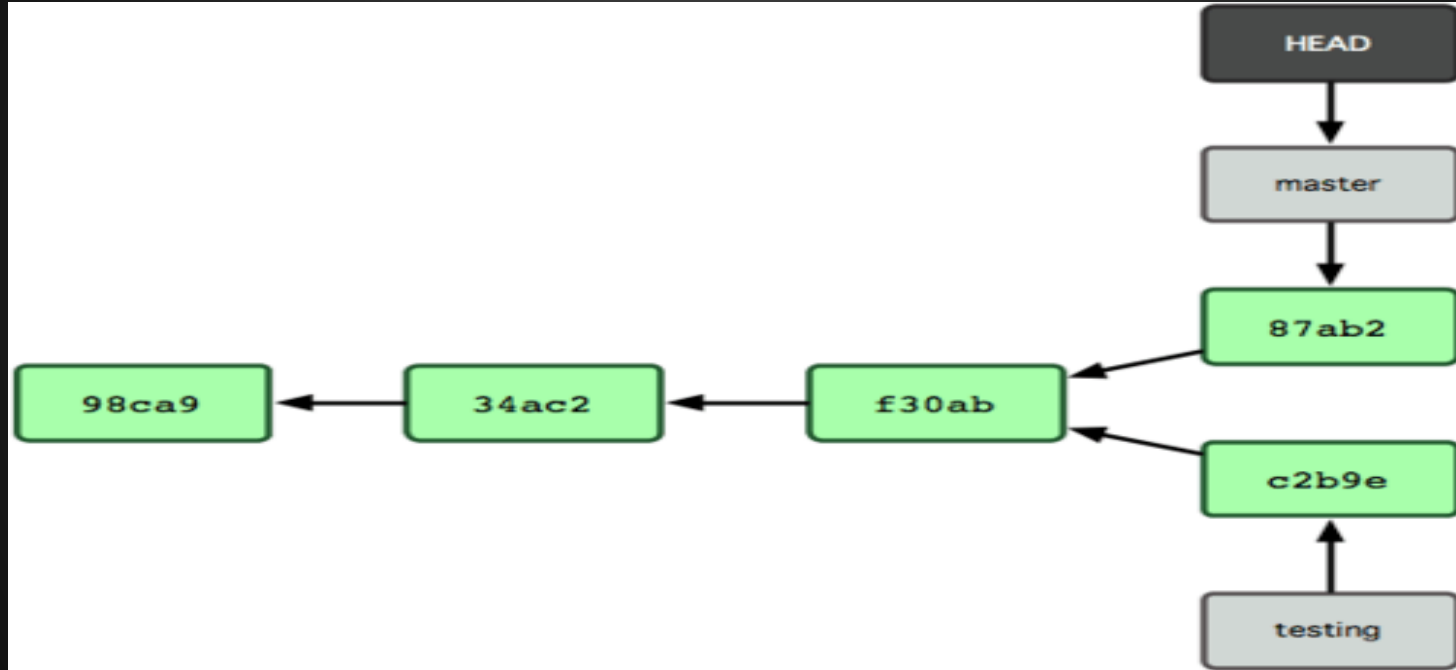
HEAD apontando para o branch em que você está

```
$ git checkout testing
```



O HEAD aponta para outro branch quando você troca de branches.

```
$ git checkout master
```



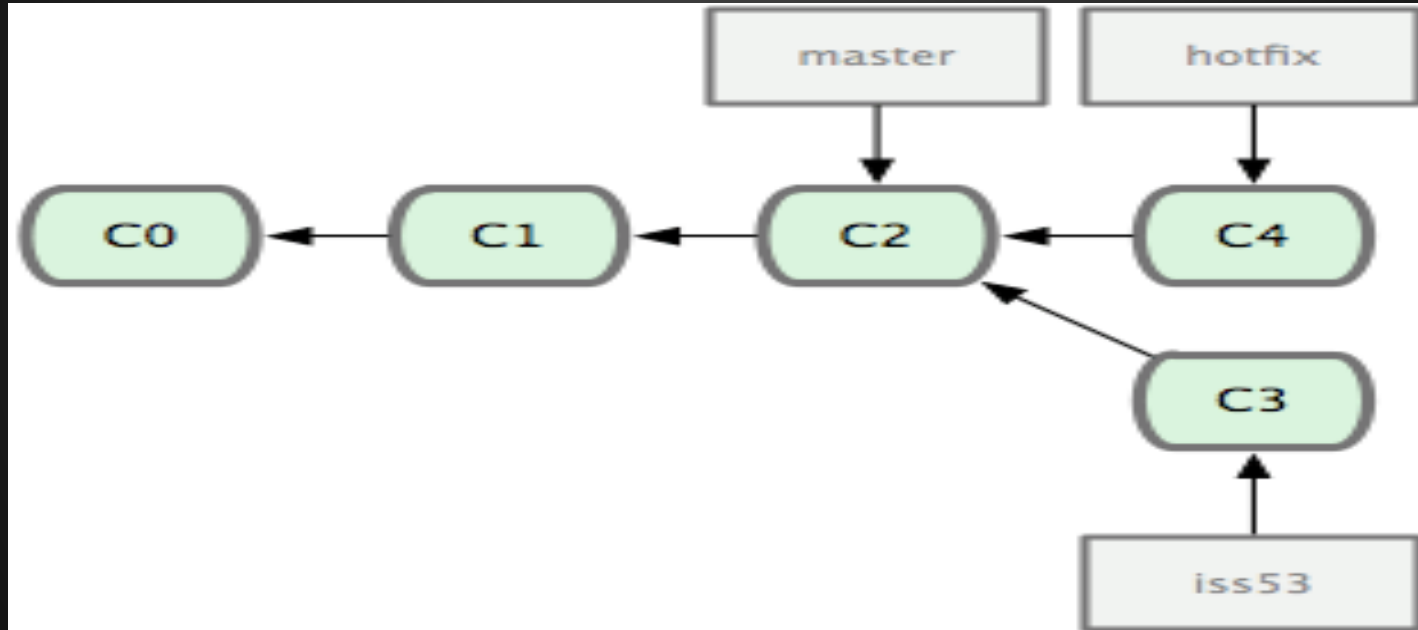
O histórico dos branches diverge.

# Comandos

```
$ git branch -d example
```

apaga o branch "example" se você  
não estive mais necessidade

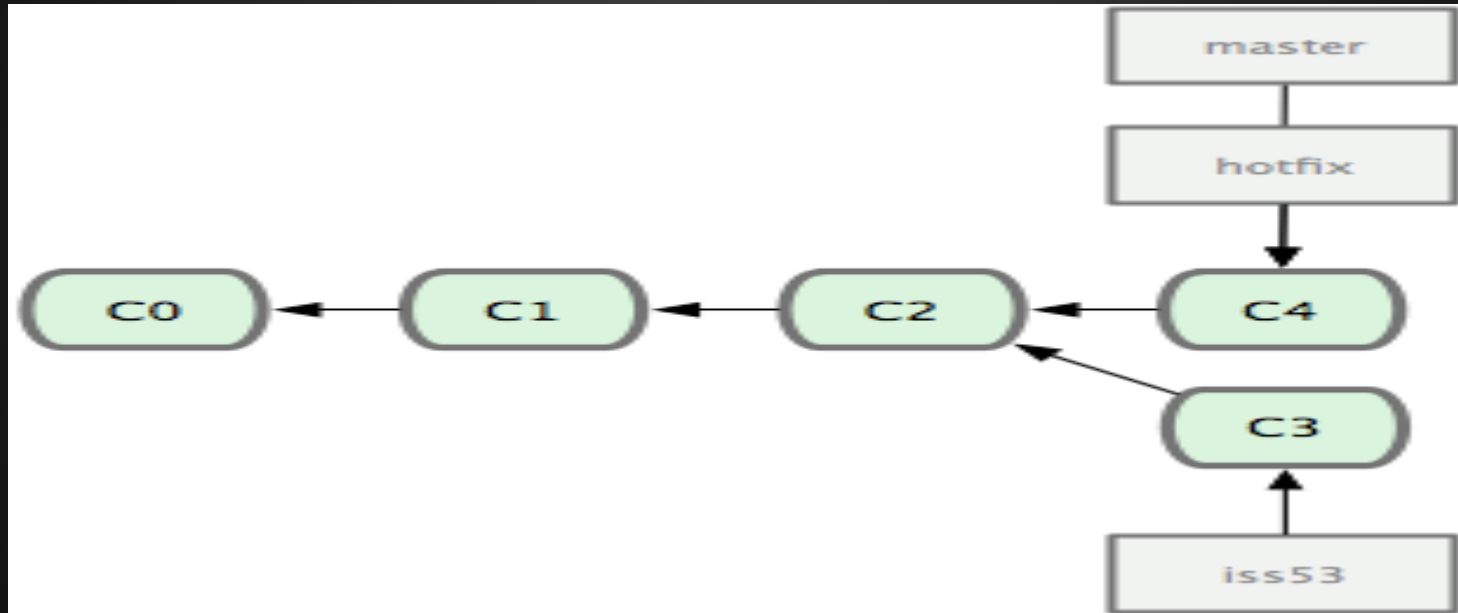
# Merge



branch de correção (hotfix) baseado num ponto de seu branch master.

\$checkout master

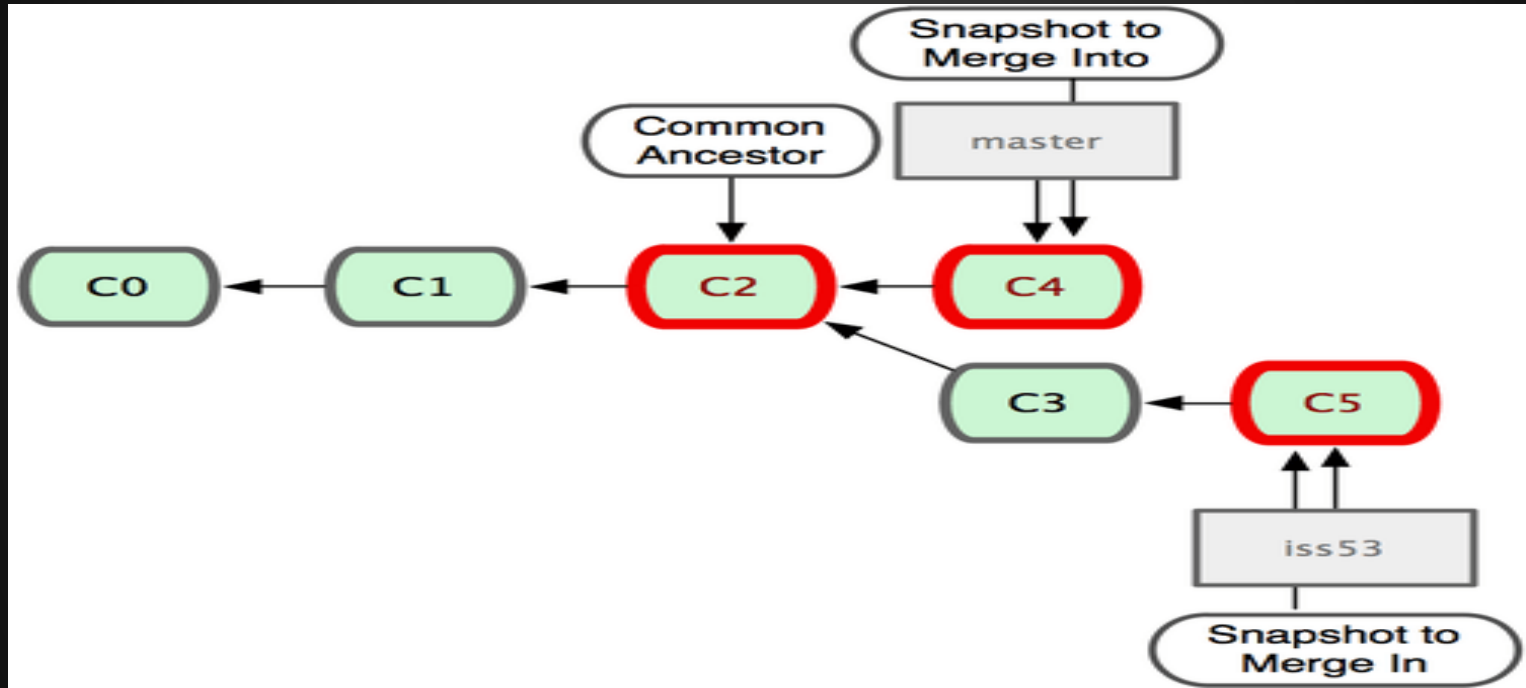
\$ git merge hotfix



Depois do merge seu branch master aponta para o mesmo local que o branch hotfix.

```
$checkout master
```

```
$ git merge iss53
```



Git identifica automaticamente a melhor base ancestral comum para o merge do branch.