

git básico

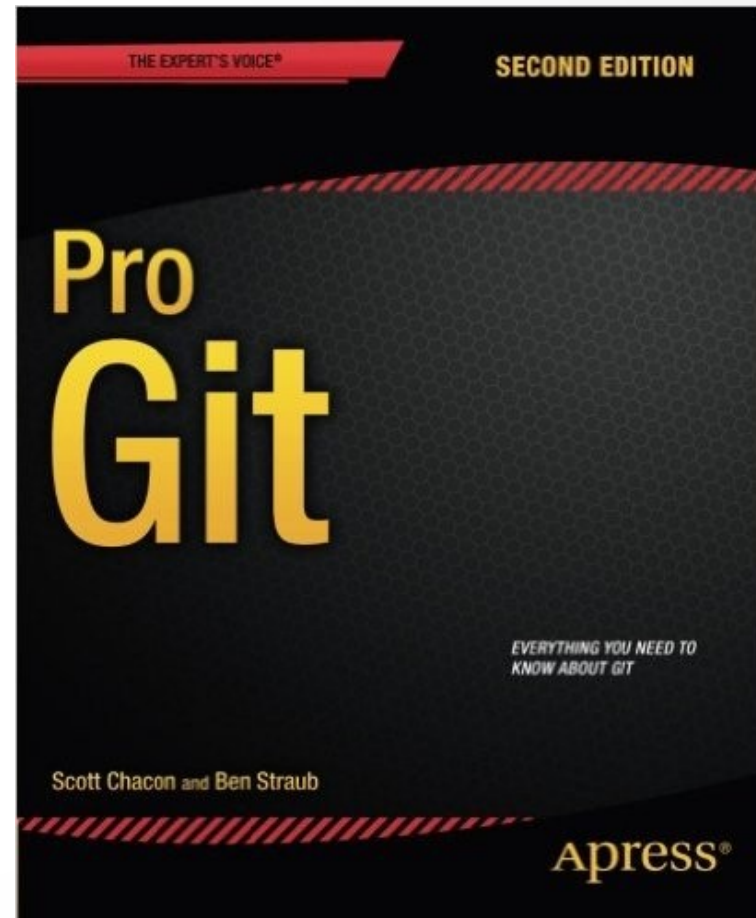
gmgall@lncc.br

<http://cursos.gmgall.net>

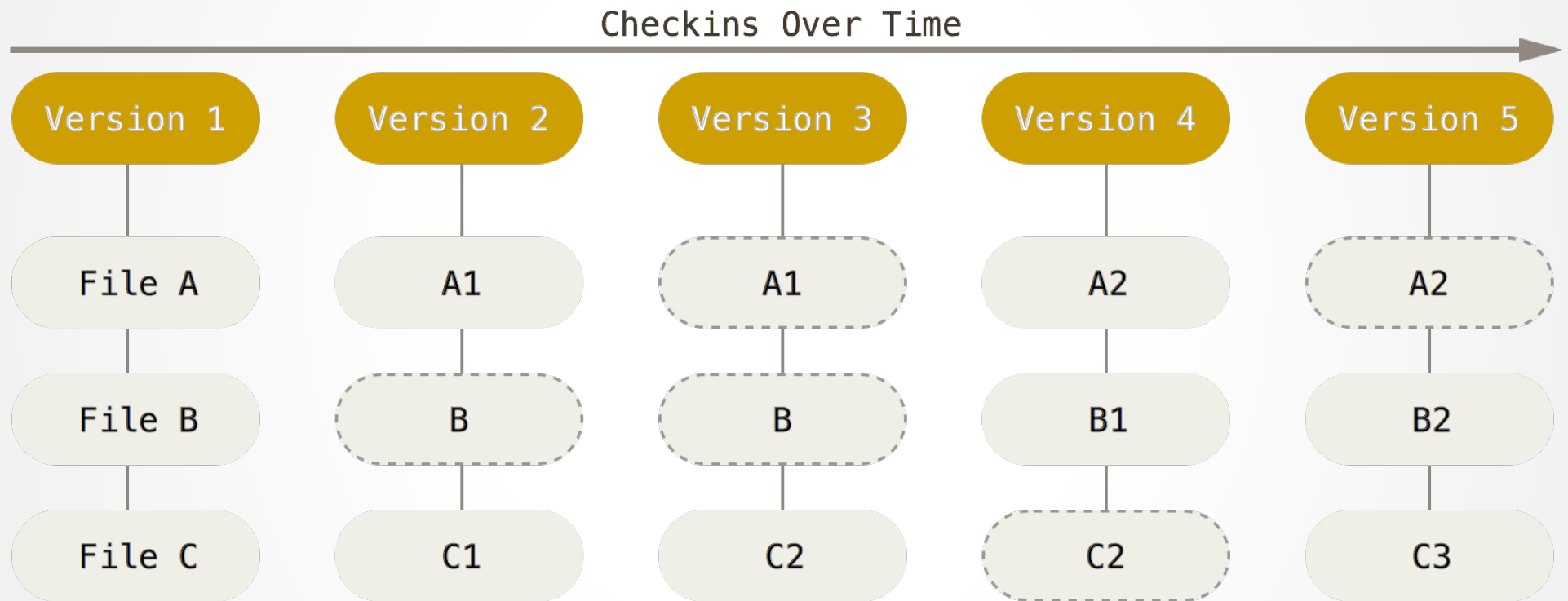
Pro Git de Scott Chacon e Ben Straub

- **ISBN-13:** 978-1484200773
- **ISBN-10:** 1484200772

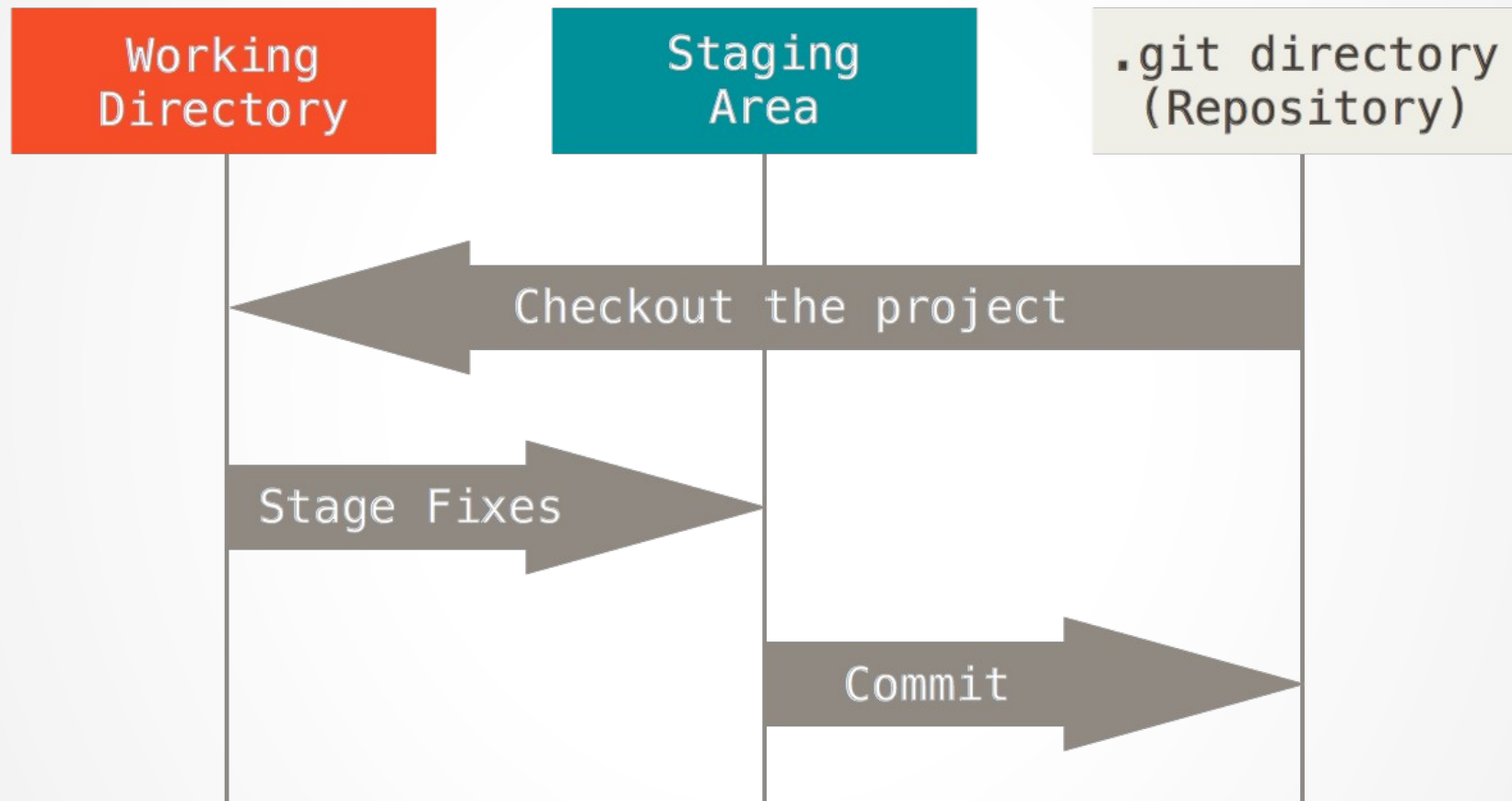
<https://git-scm.com/book/en/v2>



Como o git lida com os dados



Os 3 estados



Veja interativamente: <http://ndpsoftware.com/git-cheatsheet.html>

Criando um repositório git

```
$ git init
```

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

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

Verificando o estado

```
$ git status  
No ramo master
```

Submissão inicial.

nada para enviar (crie/copie arquivos e use "git add" para registrar)

Verificando o estado

```
$ echo 'Olá mundo' > README
```

```
$ git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
  (use "git add <file>..." to include in what will  
  be committed)
```

```
    README
```

```
nothing added to commit but untracked files present  
(use "git add" to track)
```

Verificando o estado

```
$ git add README
```

```
$ git status
```

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   README
```


Verificando o estado

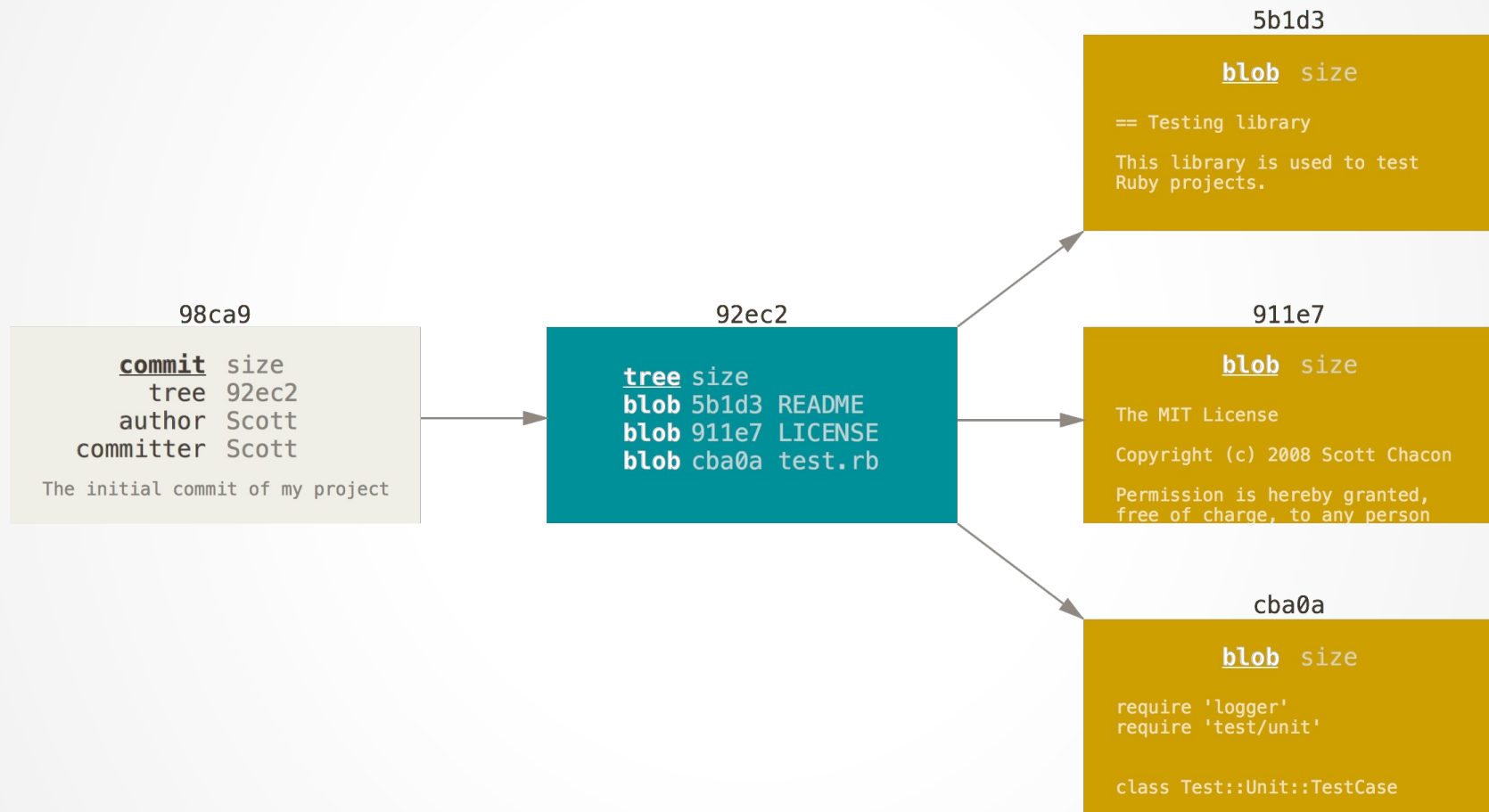
```
$ git commit
[master (root-commit) 6835649] Commit inicial
 1 file changed, 1 insertion(+)
 create mode 100644 README
$ git status
On branch master
nothing to commit, working directory clean
```

Histórico de commits

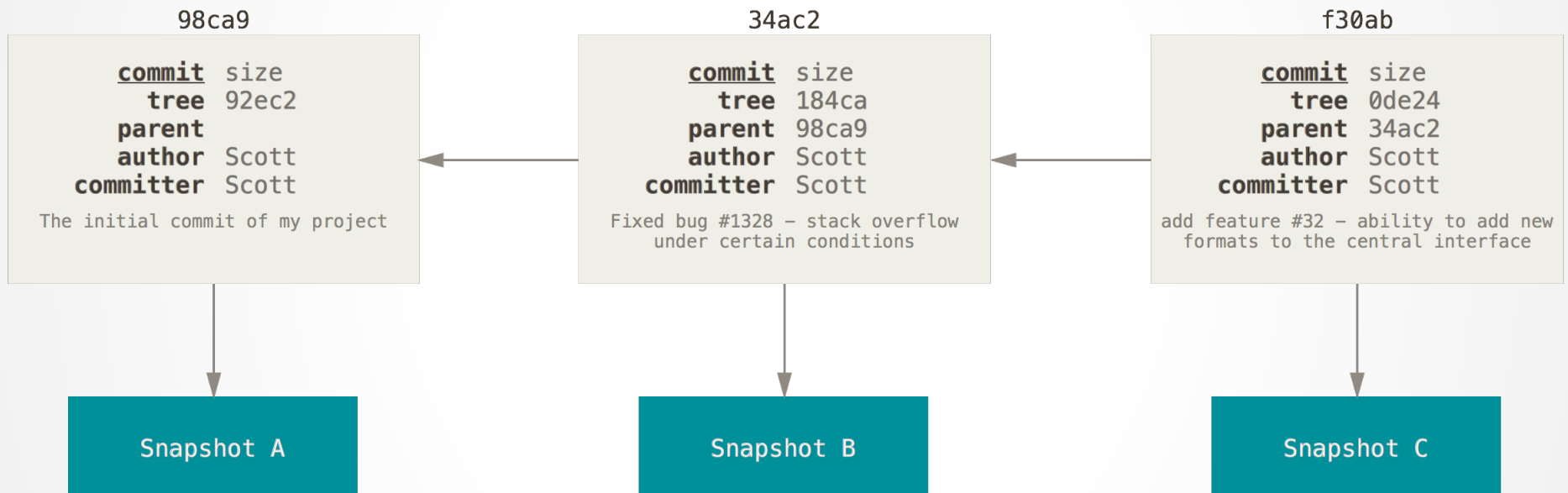
```
$ git log
```

```
$ gitk # interface grafica
```

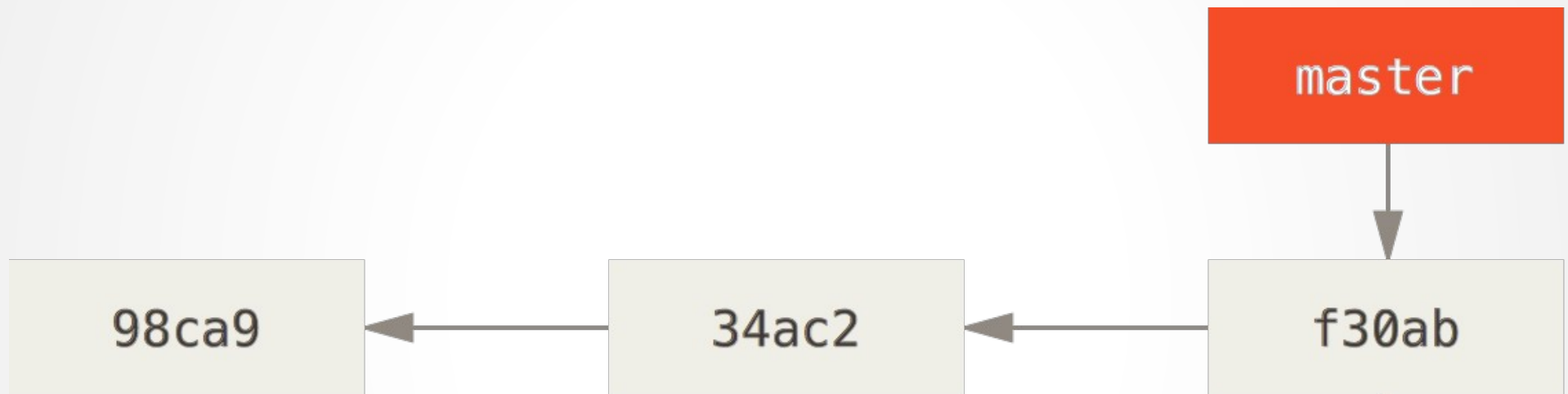
branches – entendendo os objetos num repositório git



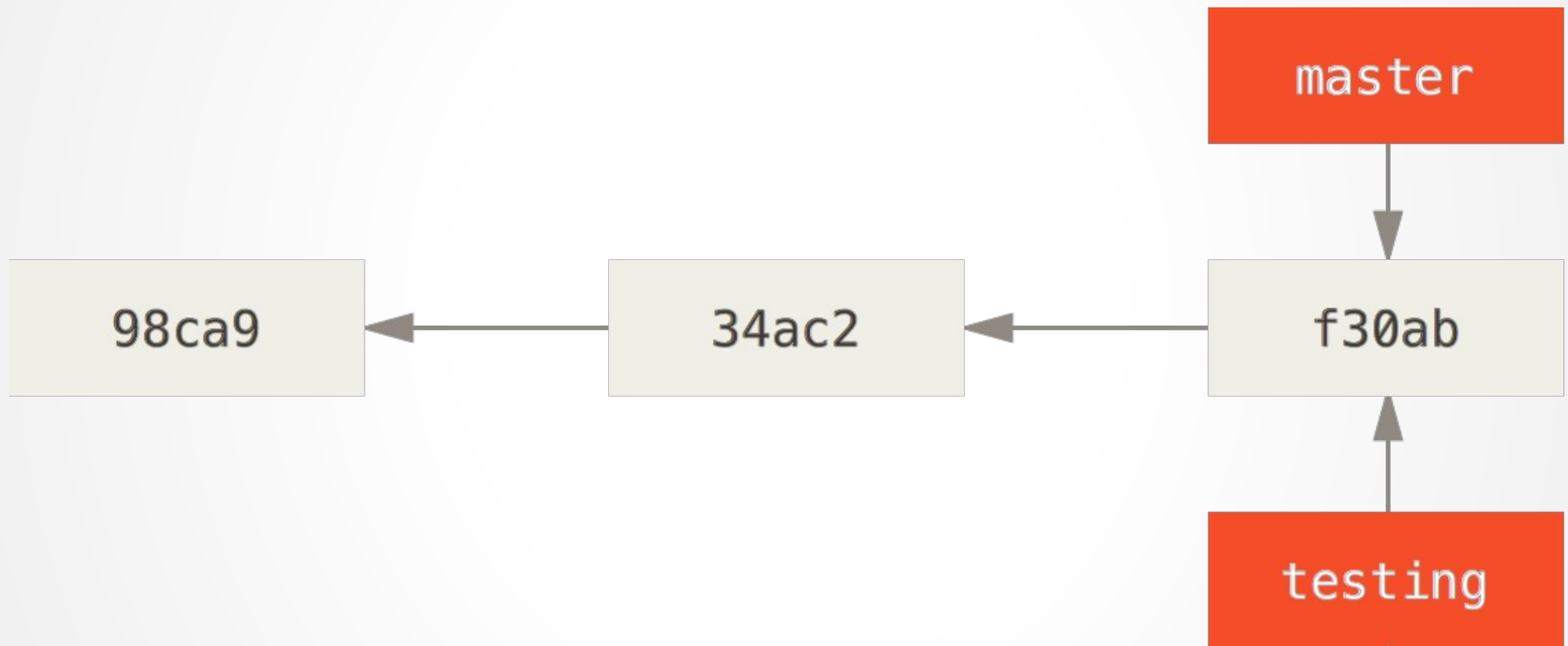
branches – entendendo os objetos num repositório git



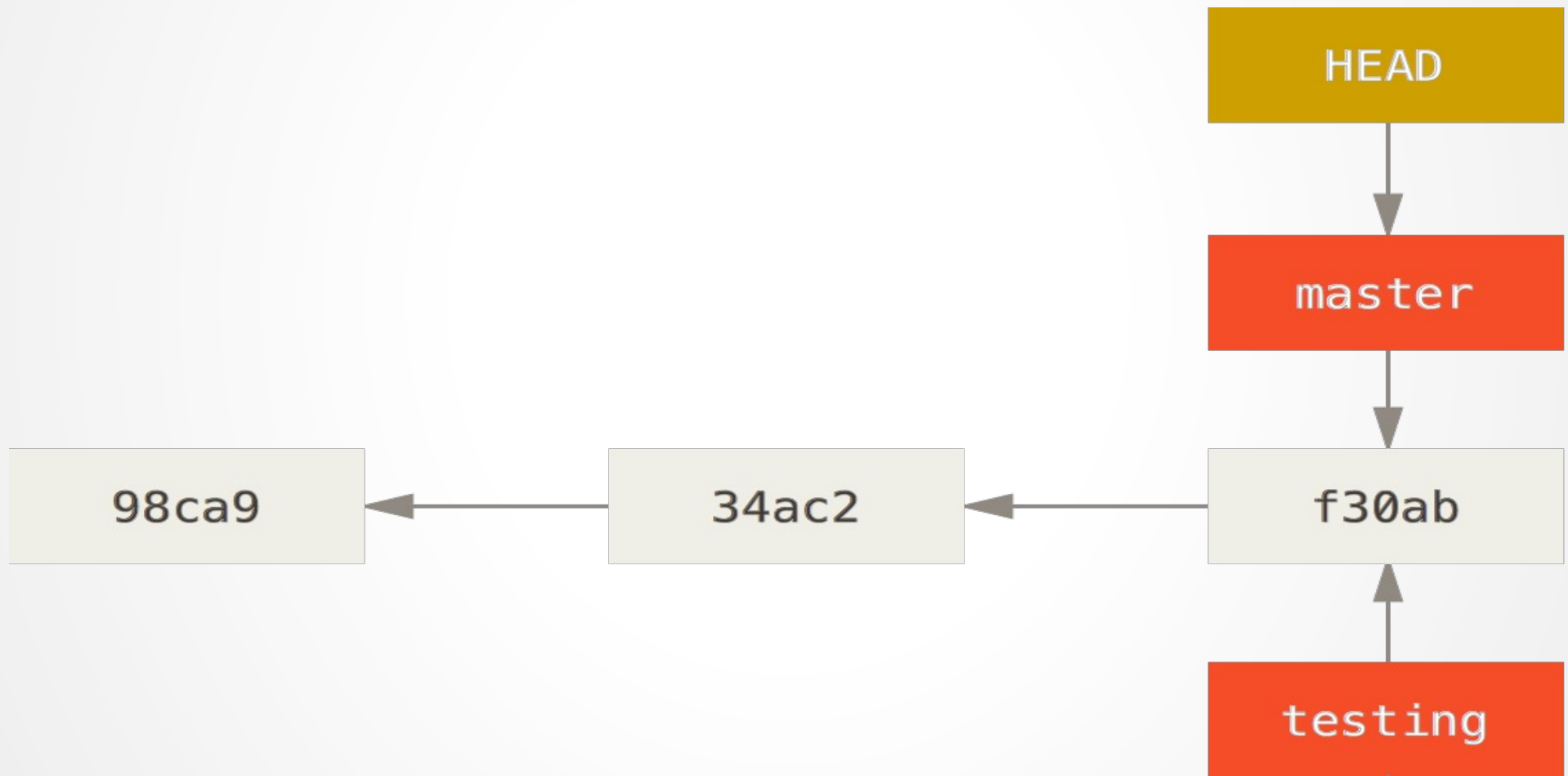
branches



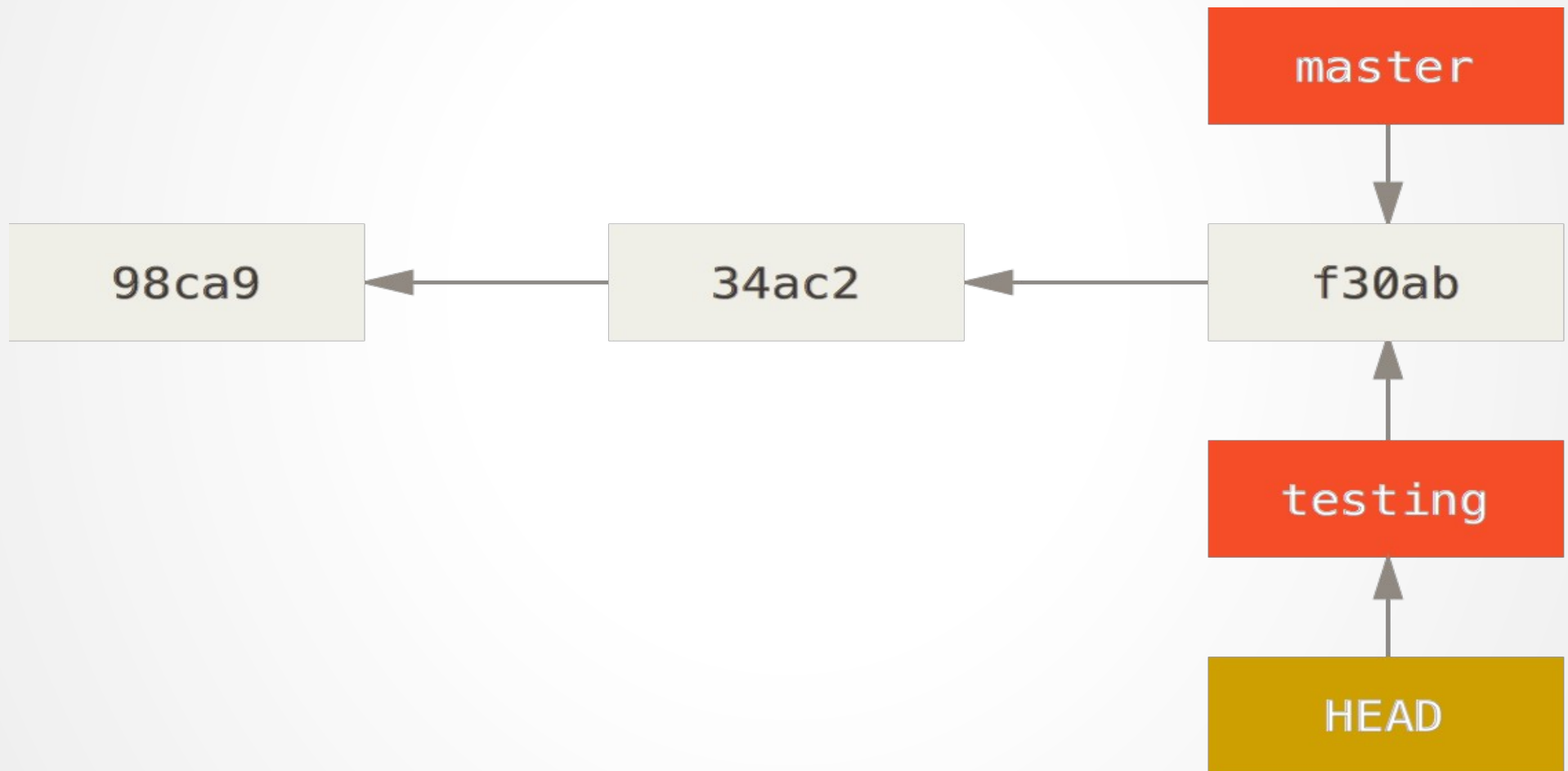
branches



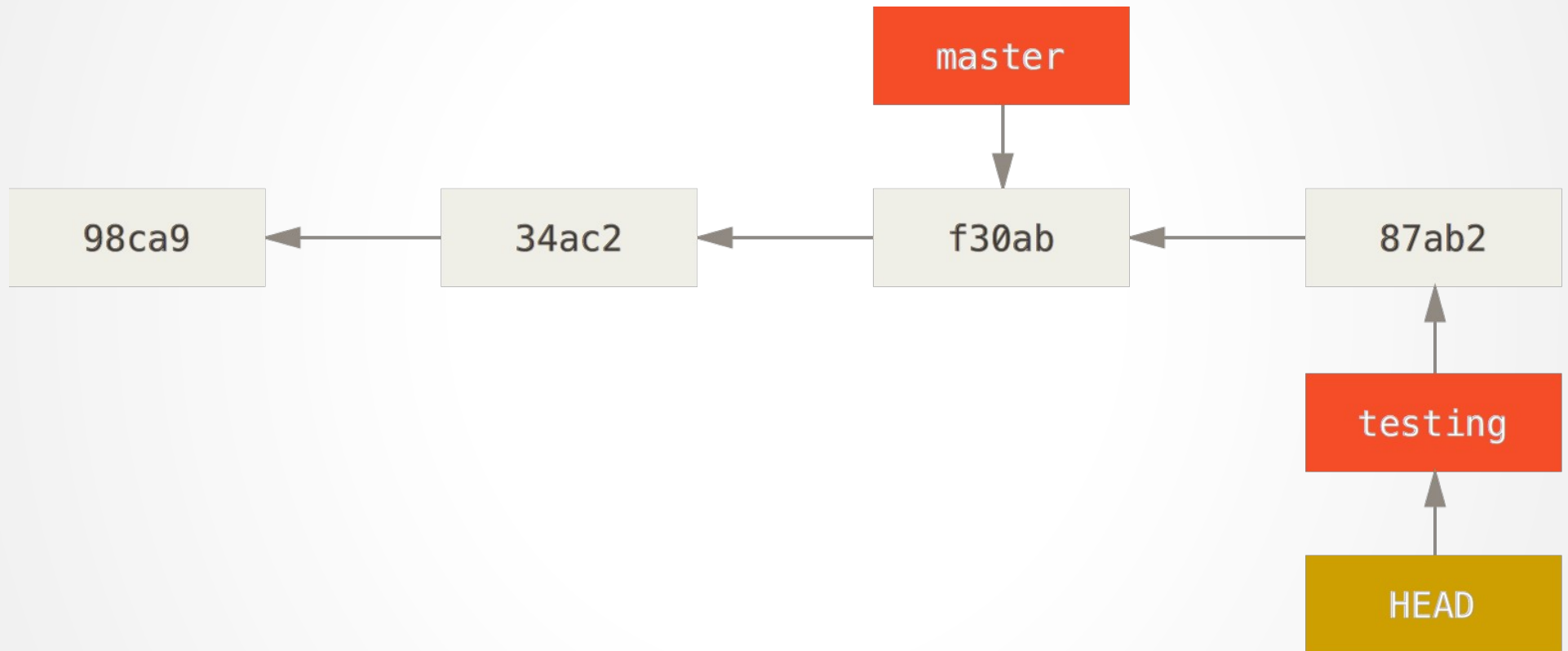
branches – em qual branch estou?



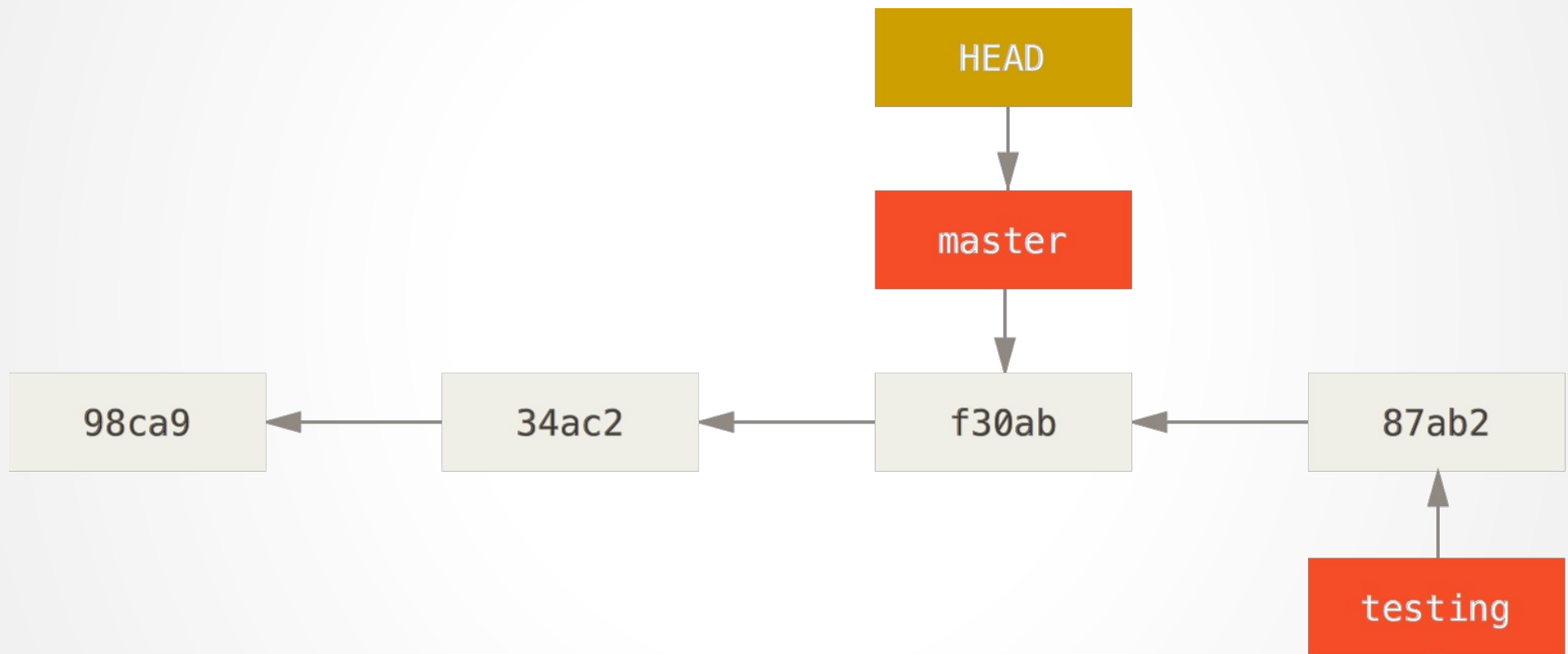
branches – em qual branch estou?



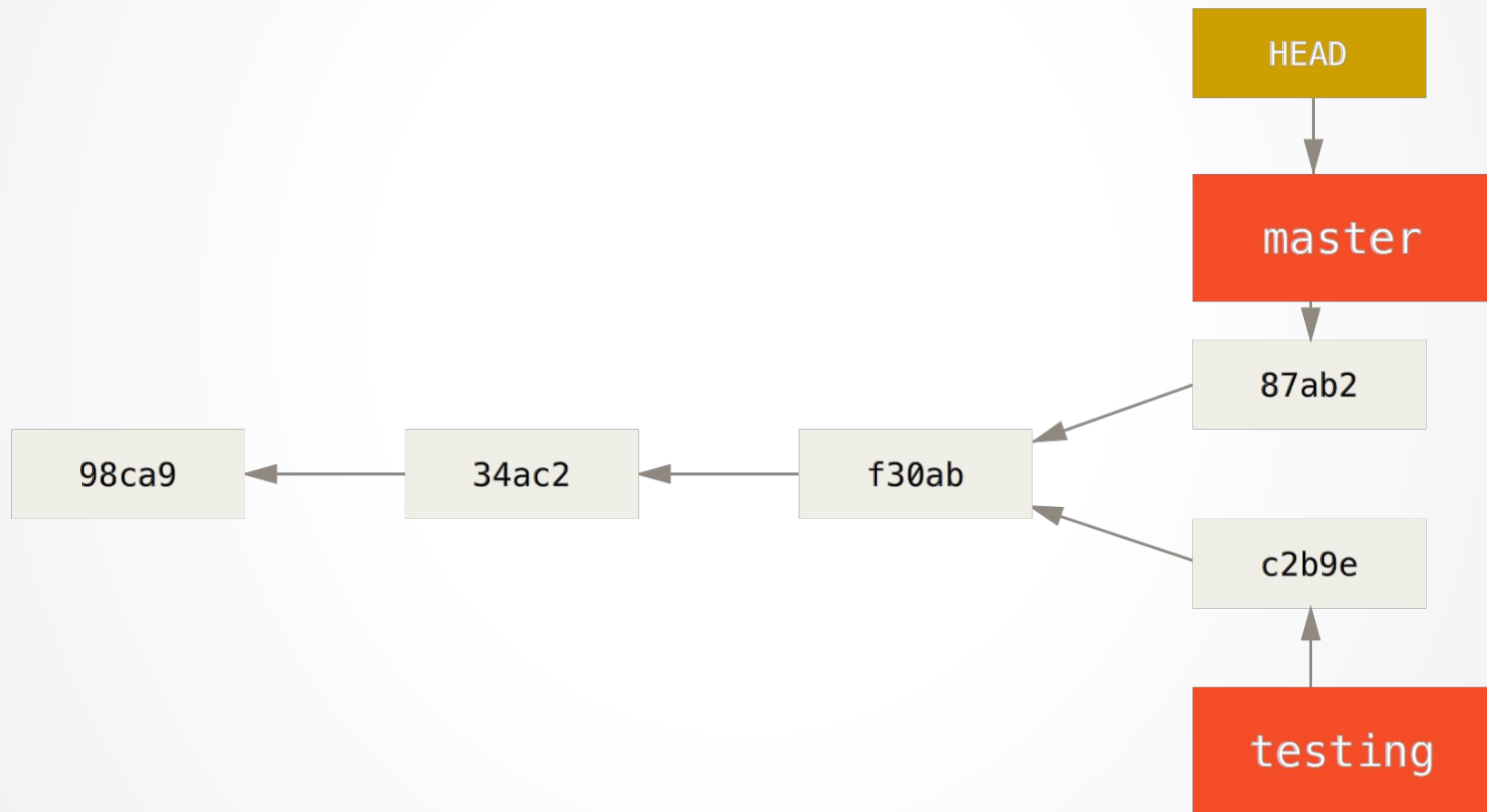
branches



branches



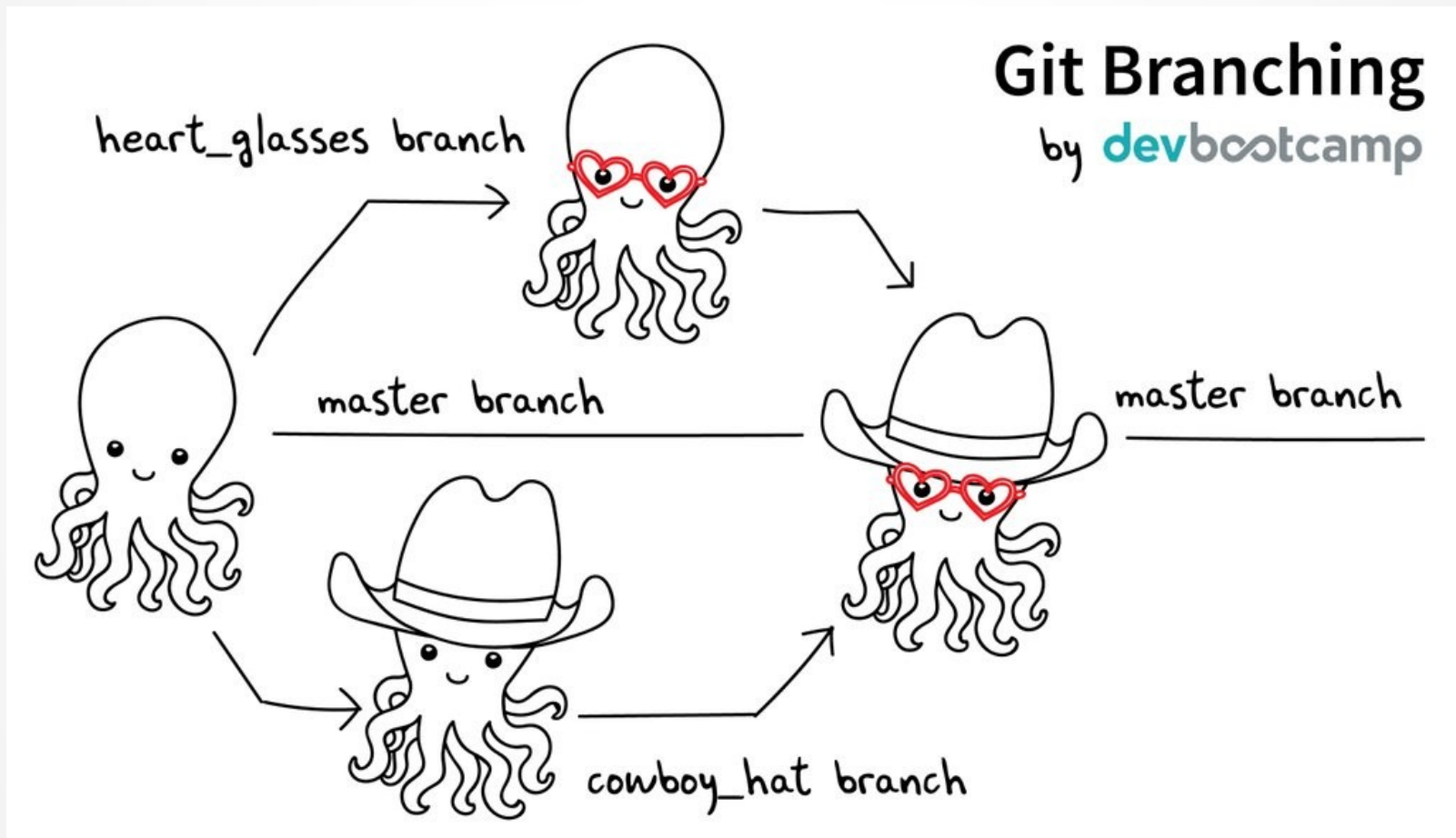
Histórico divergente



Demonstração!

- `git log --oneline --decorate --graph --all`
- `gitk --all`

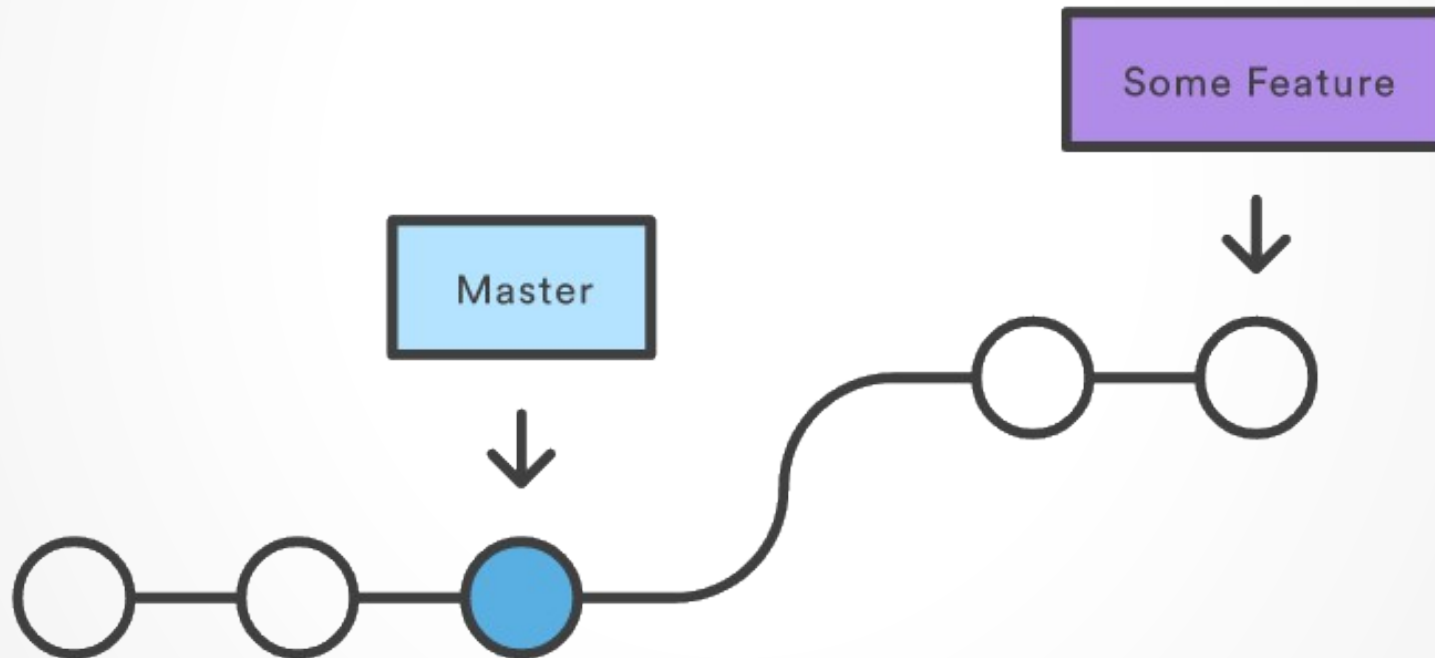
merge



Desenho de [Jennifer Gilbert](#)

merge – fast forward (antes)

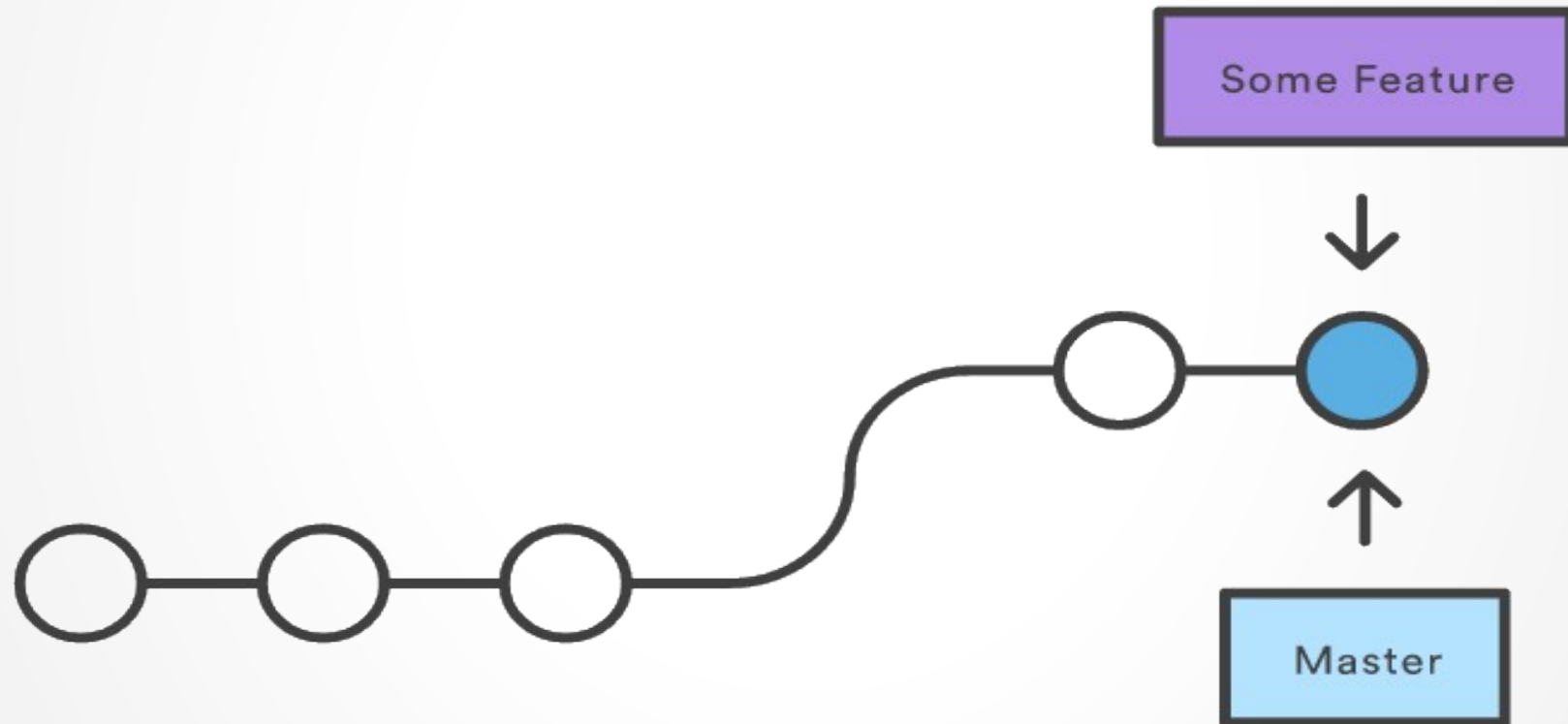
Before Merging



Fonte: [Atlassian](#)

merge – fast forward (depois)

After a Fast-Forward Merge



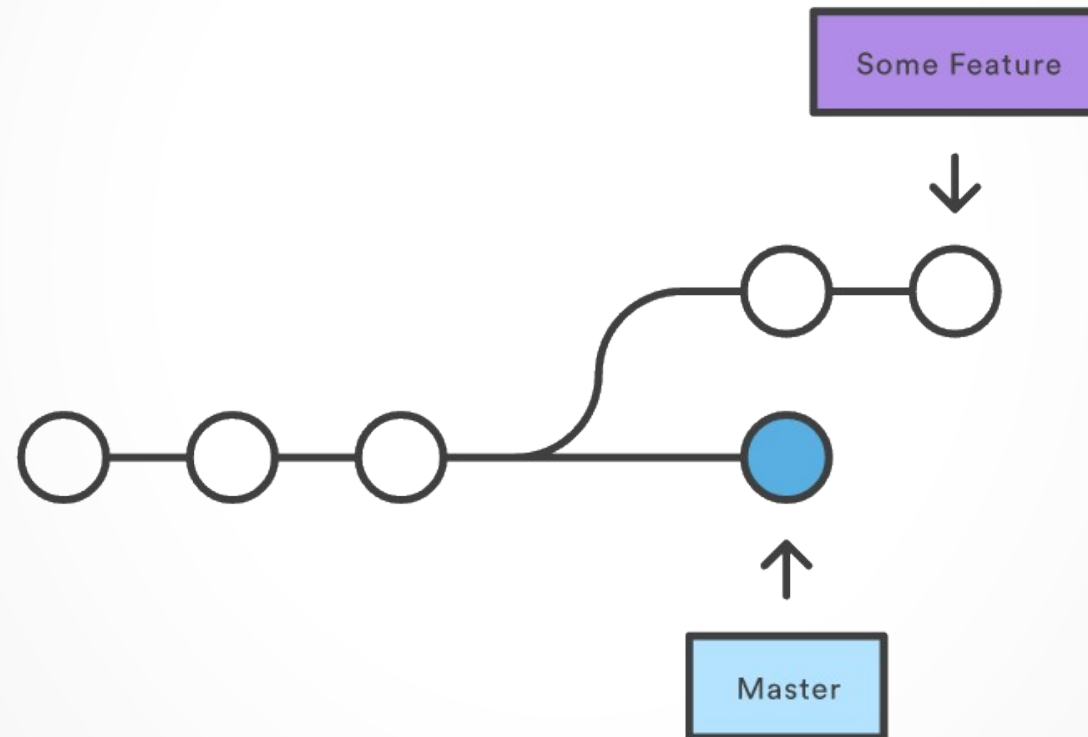
Fonte: [Atlassian](#)

\o/ Demonstração \o/

- Vamos criar um projeto simples e adicionar uma nova funcionalidade via *branches* e *merge*

merge – 3 way merge (antes)

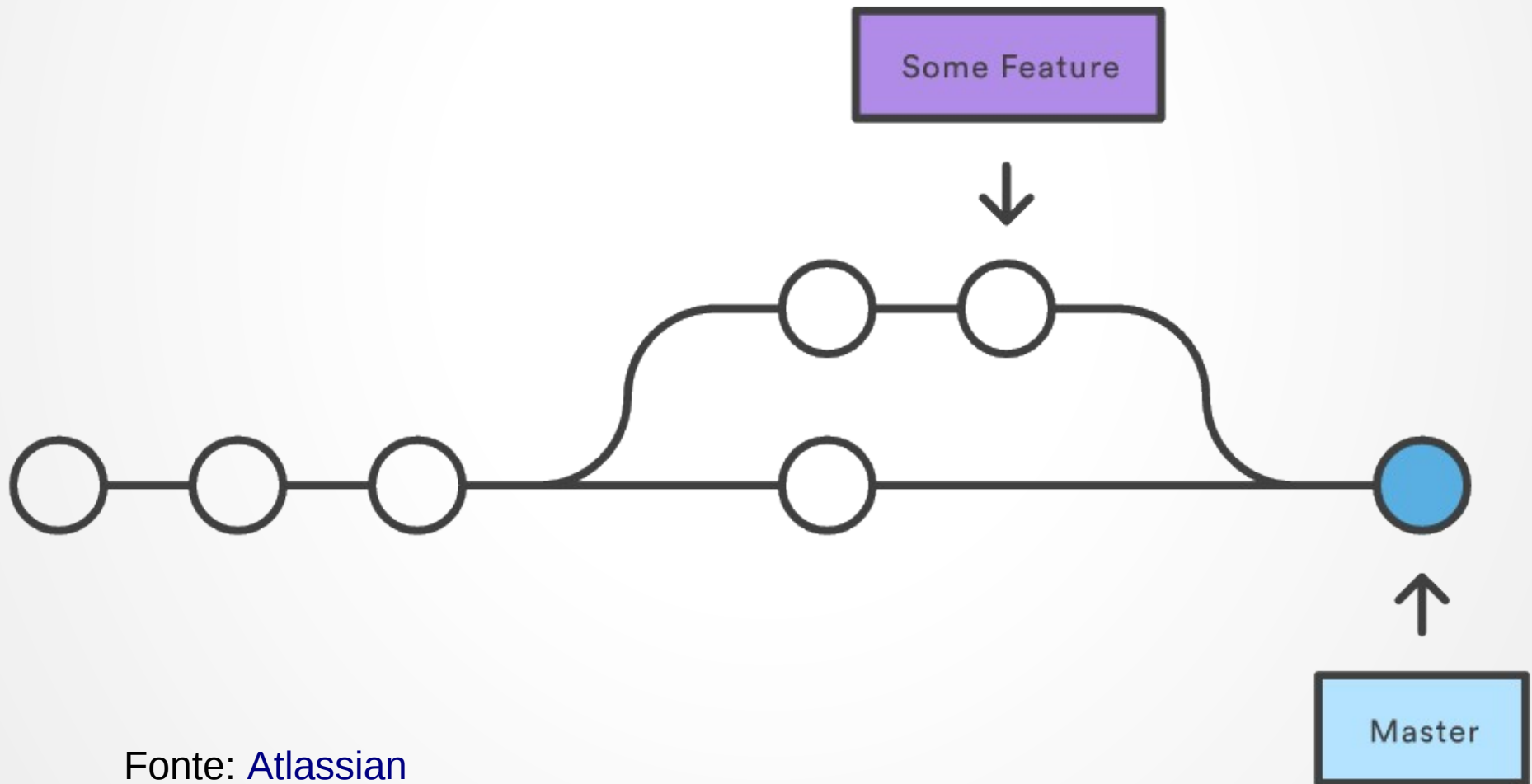
Before Merging



Fonte: [Atlassian](#)

merge – 3 way merge (depois)

After a 3-way Merge



Fonte: [Atlassian](#)

\o/ Demonstração \o/

- Vamos criar uma funcionalidade nova: contagem do total de espécies.
- Infelizmente seremos interrompidos pelo chefe mala e teremos que fazer modificações no master antes de terminar.

merge – conflitos



\o/ Demonstração \o/

- Você resolve trocar sed por cat...
- Mas no meio do caminho percebe um bug no código

(J°□°)J ~ **LL**

Repositórios remotos

```
$ git remote
```

```
$ git remote add origin URL
```

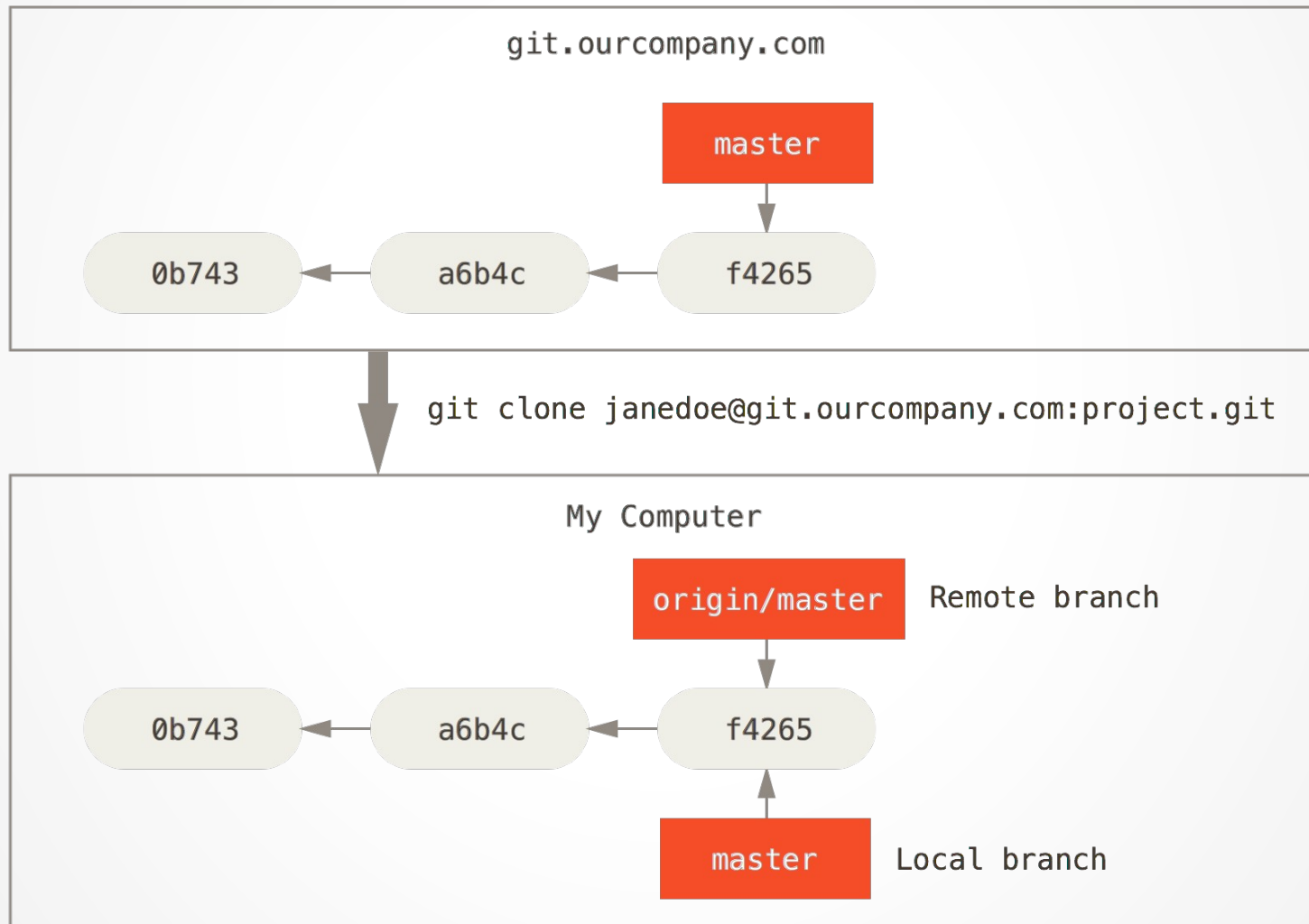
```
$ git push origin master
```

```
# faz o upload do branch local master para o remote  
origin
```

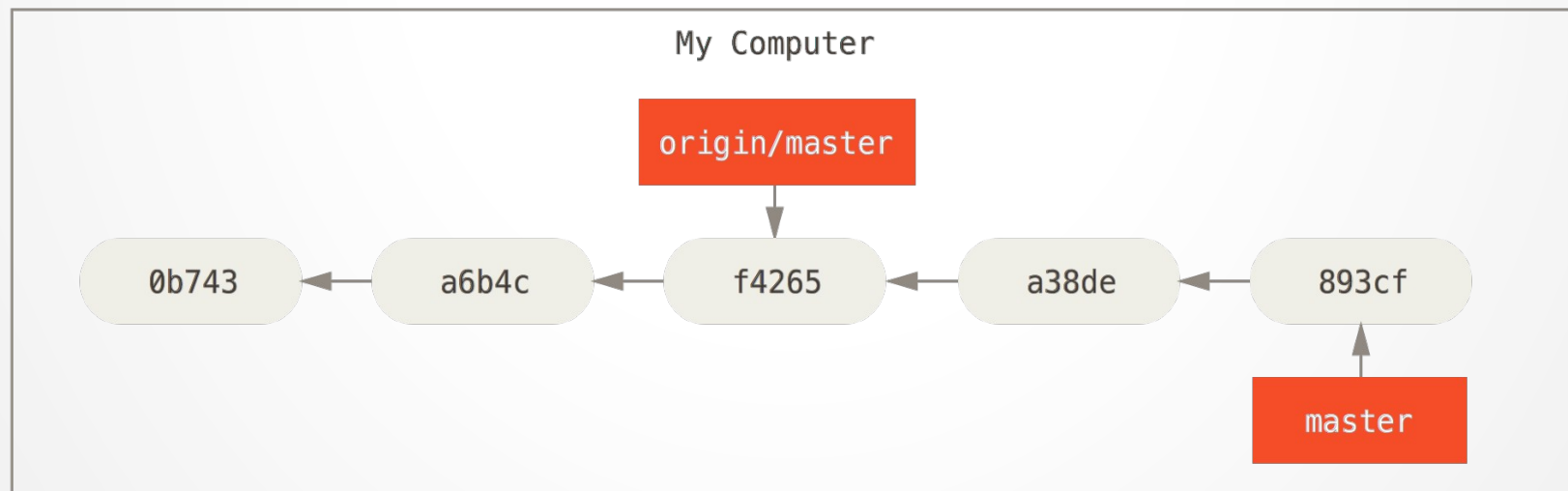
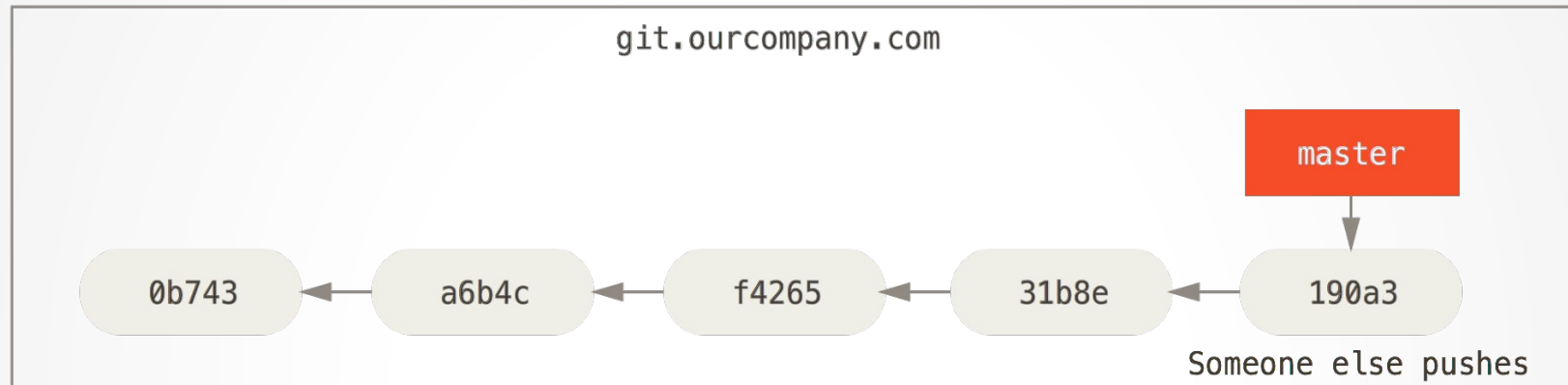
\o/ Demonstração \o/

- Vamos colocar nosso projeto de exemplo no github.

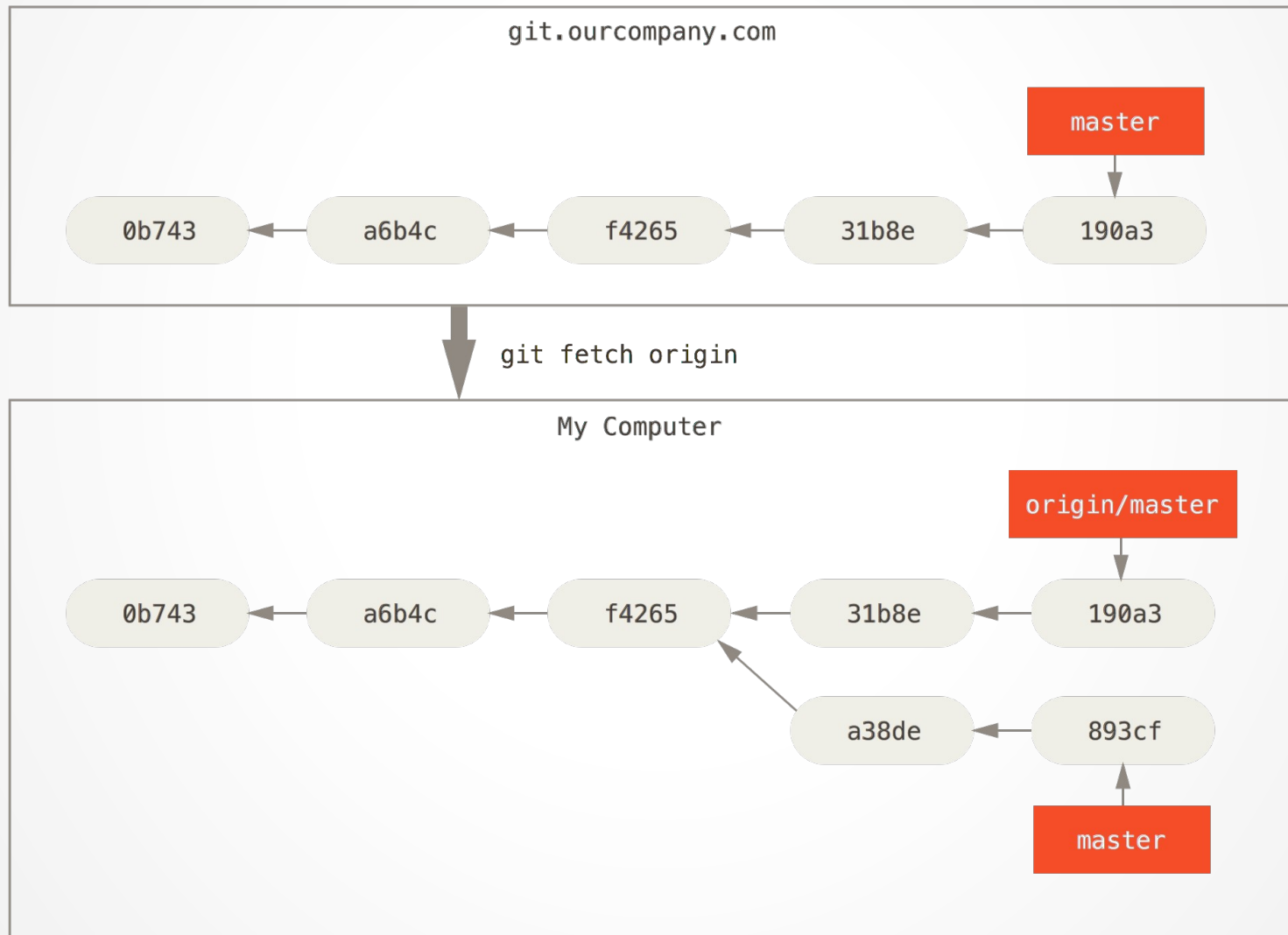
branches remotos - clone



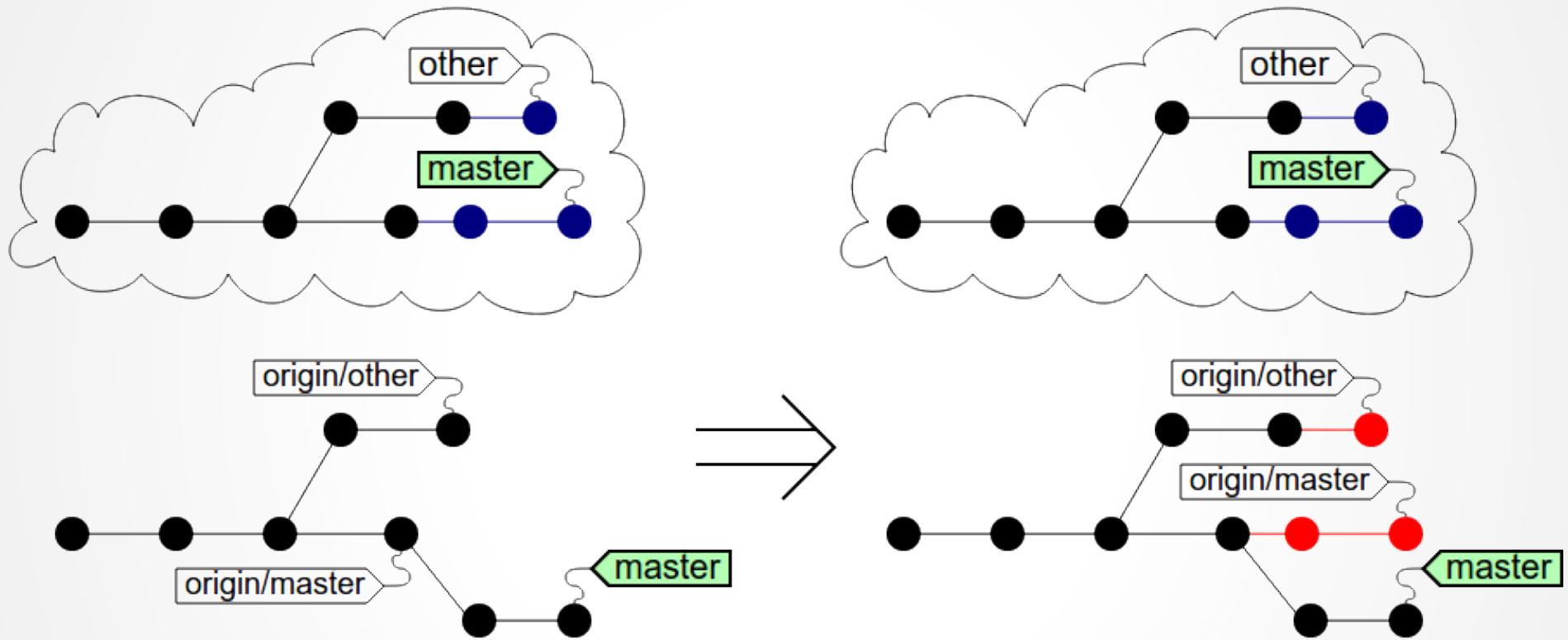
branches remotos - divergindo



branches remotos - fetch

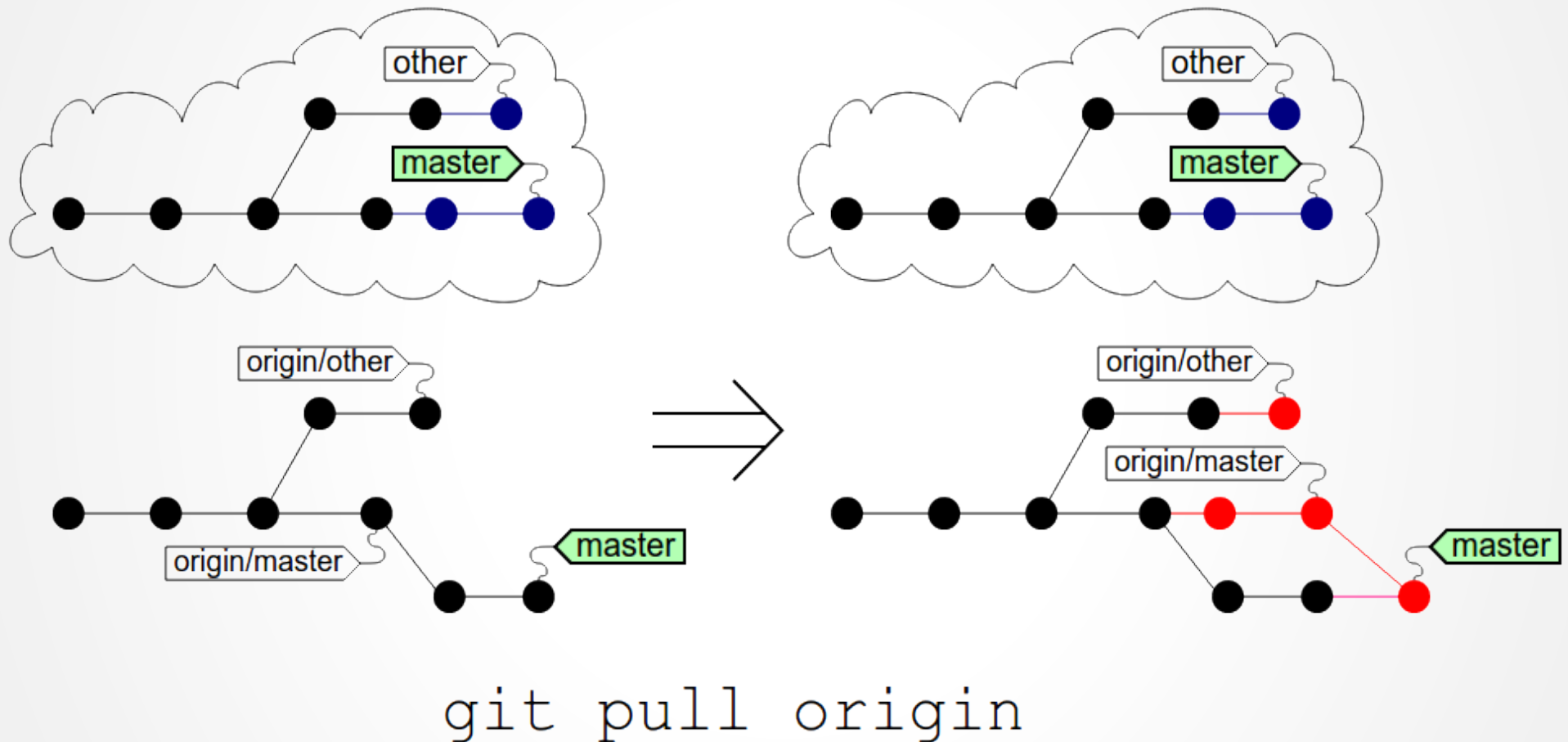


fetch vs pull

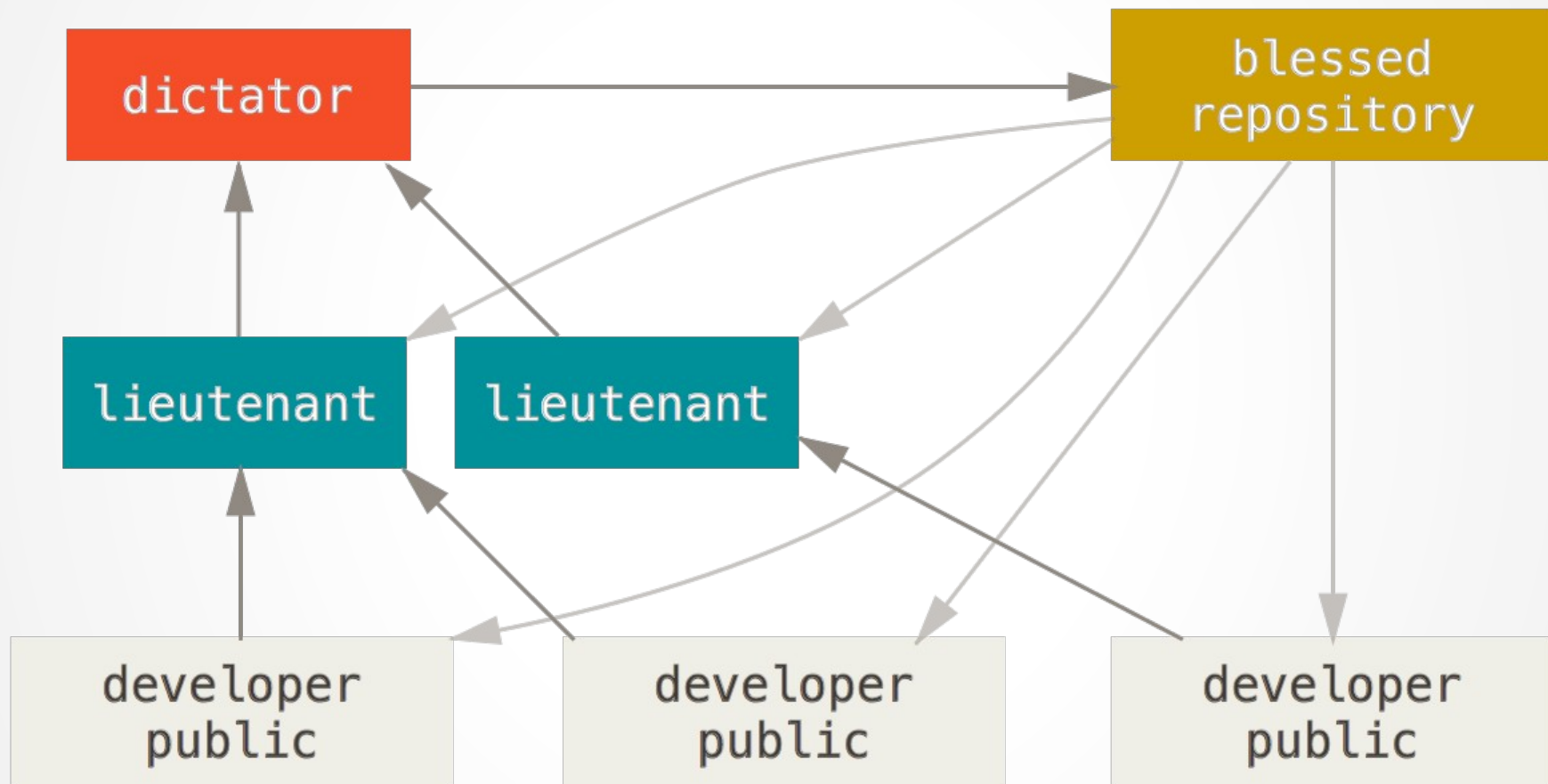


`git fetch origin`

fetch vs pull



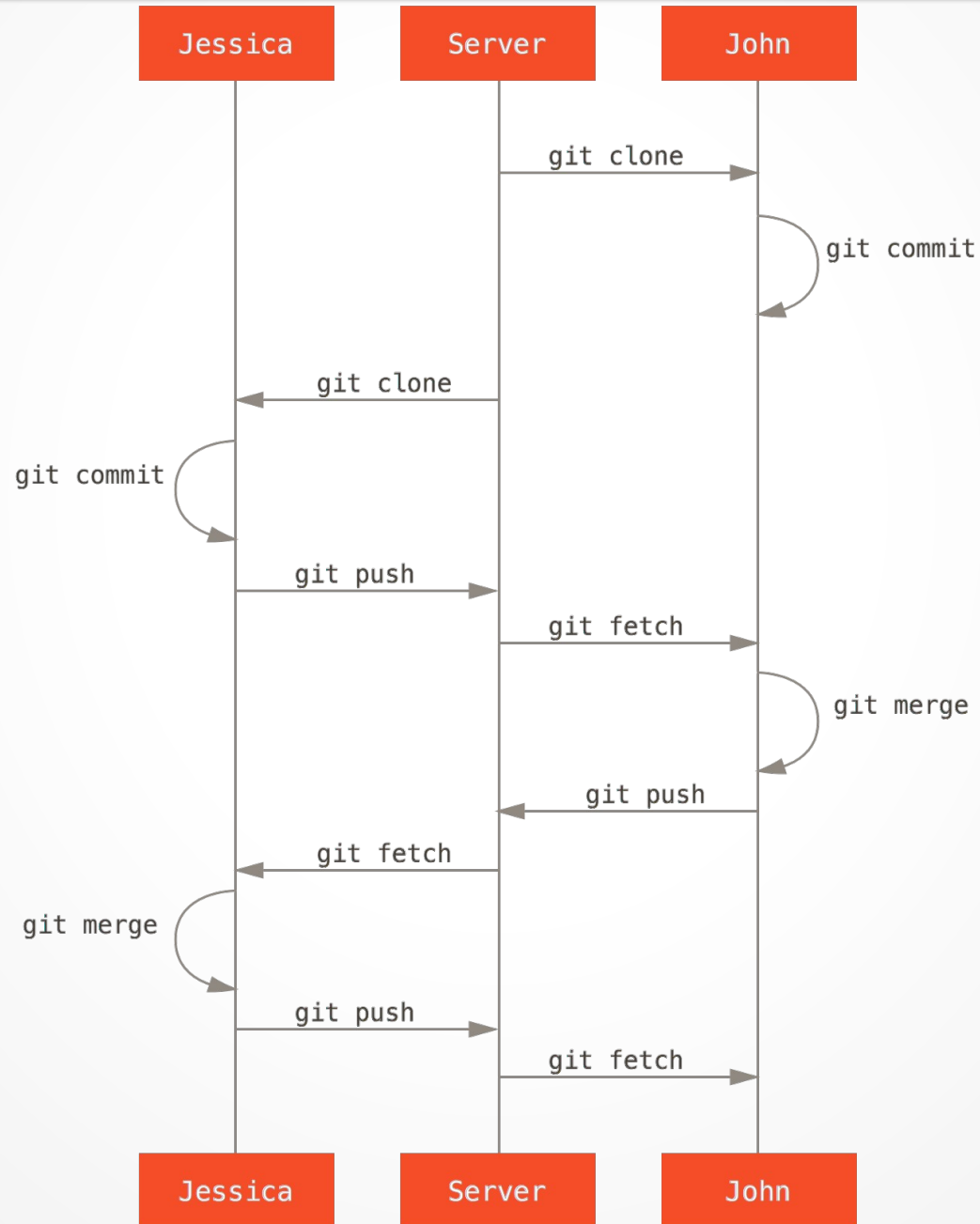
Modelos de colaboração




Modelos de colaboração (para projetos pequenos)

- Repositório comum
 - Repositório central
 - Colaboradores com permissão de push
- Pull Request (github)
 - Cada colaborador tem uma cópia do repositório central
 - Colaboradores fazem alterações nessas cópias
 - Alterações ao repositório central são **propostas** via *Pull Request*

Repositório comum



Pull Request (github)

- Visite a página do projeto e clique em  **Fork**
- **Na sua cópia**, crie um branch para a nova funcionalidade
- Faça os commits necessários
- Faça push **para sua cópia**
- Abra um Pull Request no Github
- Discuta a modificação, fazendo mais commits se necessário
- O mantenedor do projeto faz o merge ou fecha o Pull Request

Resumo desse processo: [How to Github](#)



That's all Folks!