



# Git & Github



Renatha Victor



Felipe Rodrigues



John Oliver

- **Introdução**

- Controle de versão
- História do git
- Principais conceitos do git

- **Essencial do Git**

- Inicializando um repositório
- Configuração inicial do git
- Ciclo de vida do git
- Desfazendo as coisas

- **Trabalhando com branchs**

- Criando uma branch nova
- Trabalhando na nova branch
- Juntando branches
- Resolvendo conflitos de junção

- **Trabalhando com repositórios remotos**

- Criando um repositório remoto
- Ligando o repositório local com o remoto
- Editando o repositório remoto
- Atualizando o repositório local com o remoto

- **Extras**

- Ignorando arquivos com `.gitignore`
- Kahoot



# Git e Github

## Registro de Modificações

É utilizado principalmente para gerenciar versões de software e manter tudo registrado em um histórico.

## Desenvolvimento colaborativo

O software pode ser desenvolvido por um ou mais colaboradores onde os integrantes de um projeto podem enviar correções, atualizações e etc.



## Controle de Modificações

As alterações enviadas ao projeto principal não compromete o mesmo, pois cabe ao dono do projeto a inclusão ou não das alterações efetuadas.

## Reversão de Modificações

Podemos retroceder sempre que necessário, voltando a um estado anterior antes da modificação.

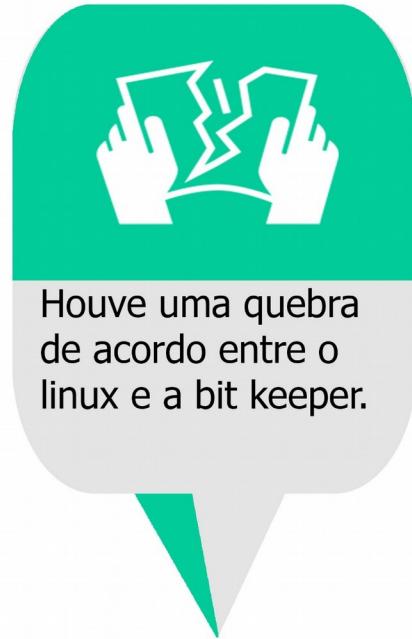
# História do Git



Foi o primeiro sistema de controle de versão distribuído.

## Bit Keeper

Sistema de versionamento que guardava todo o código do kernel do linux.



Houve uma quebra de acordo entre o linux e a bit keeper.

## Quebra de acordo

A bit keeper exigia um pagamento para manter o código do kernel guardado e Linus Torvalds rompeu o contrato.

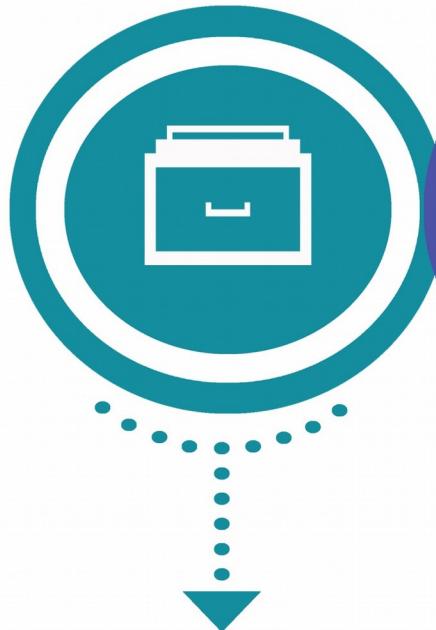


Linus torvalds criou seu próprio sistema de versionamento

## Criação do Git

O git trouxe uma série de vantagens em relação ao sistema da bit keeper, como velocidade e desempenho.

# Principais conceitos do git



**Repositório**

**Local onde fica todos os arquivos do projeto, inclusive os históricos e versões.**



**Commit**

**Coleção de alterações realizadas, sempre que necessário você pode retroceder até um commit.**



**Branch**

**Cada branch representa uma versão do projeto e em uma branch pode haver vários commits.**



**Fork**

**Uma cópia de um projeto existente para seguir em uma direção.**



**Merge**

**É a capacidade de incorporar alterações do git, onde acontece junções de branch.**



## Inicializando um repositório:

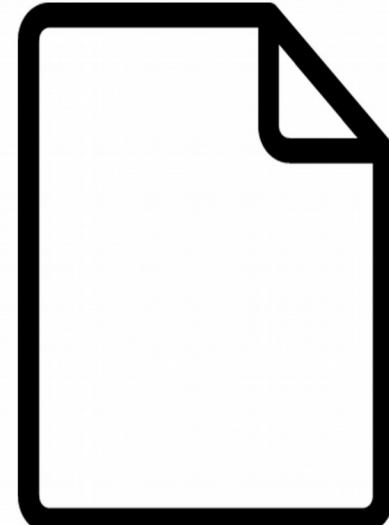
```
$ git init repositorio
```

```
$cd repositorio
```

```
$ ls -a
```

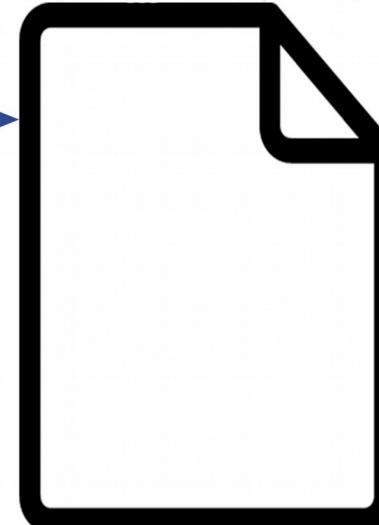
```
./ ../ .git/
```

working directory



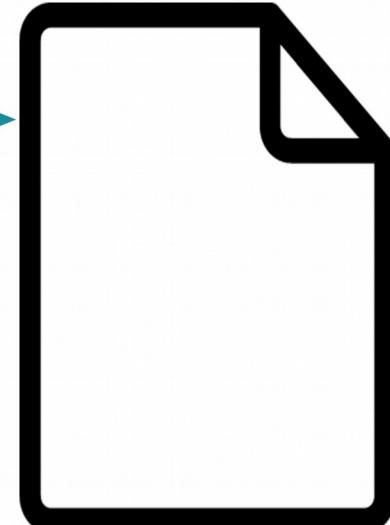
git add

Staging area (index)



git commit

Repository



## Configurando o repositório

```
$ git config --global user.nome "nome sobrenome"
```

```
$git config --global user.email "nome@email.com"
```



## Criando arquivos:

```
touch README.md
```

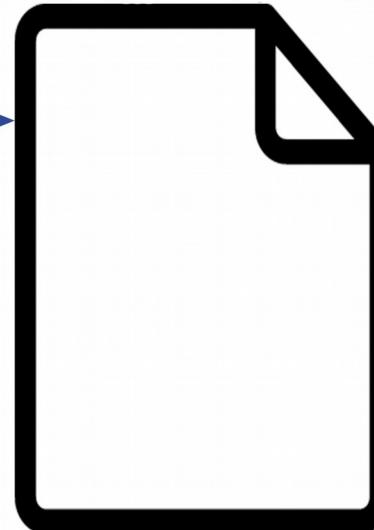
## # Alunos:

working directory



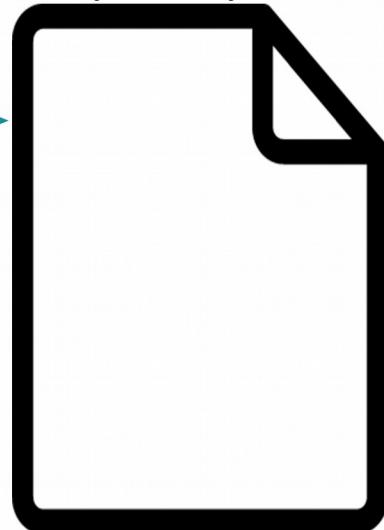
git add

Staging area (index)



git commit

Repository



```
$ git status  
On branch master
```

No commits yet

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

README.md

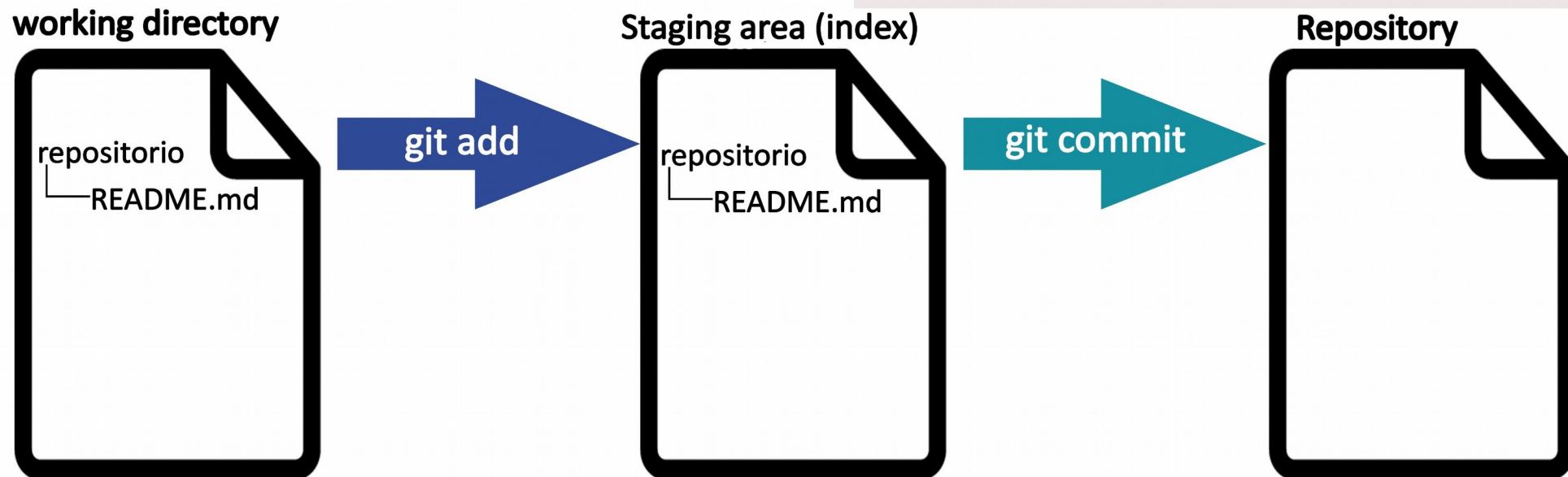
nothing added to commit but untracked files present (use "git add" to track)



## Adicionando o README.md ao Stage area

```
$ git add README.md
```

```
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  
new file: README.md
```

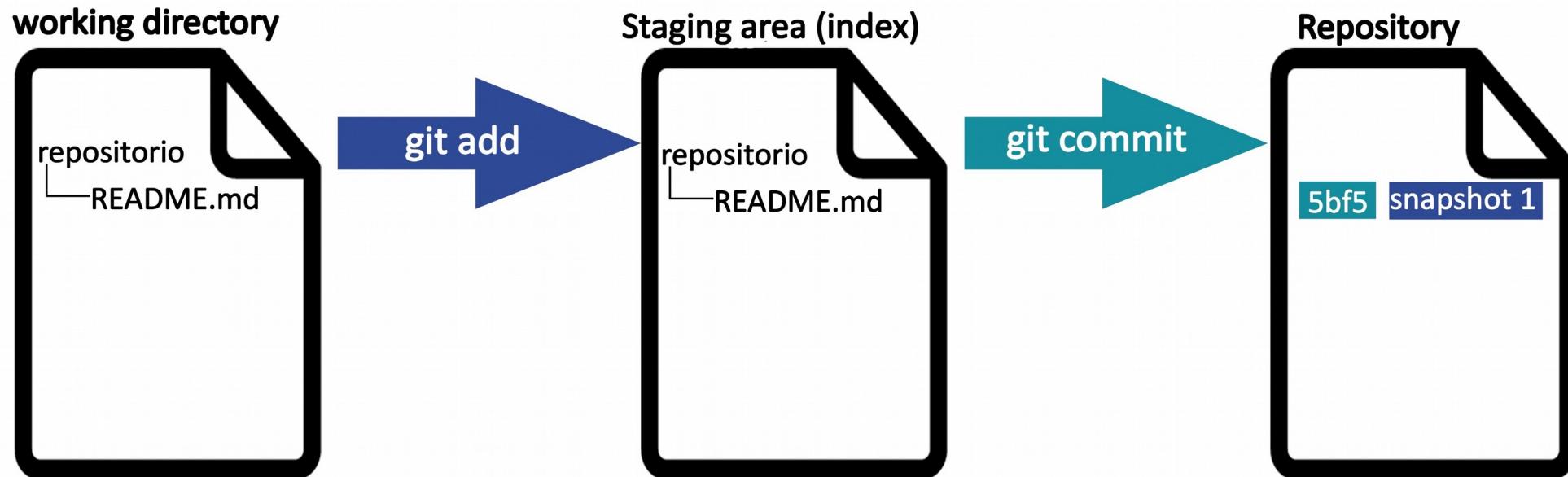




## Criando o primeiro commit

```
$ git commit -m "primeiro commit"
```

```
$ git status  
[master (root-commit) 5bf530e] primeiro commit  
1 file changed, 1 insertion(+)  
create mode 100644 README.md
```





## Exibindo a versão criada:

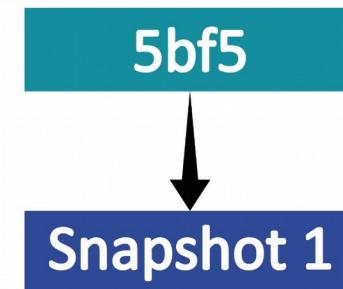
```
$ git log
```

```
commit 5bf5823b0f40af253343dd9a4d891a42e992f44b (HEAD -> master)
```

```
Author: John Oliver <johnoliver021@gmail.com>
```

```
Date: Sun Nov 18 13:13:43 2018 -0200
```

primeiro commit





## Editando arquivos

# Alunos:

- nome aluno

```
$ git status
```

```
On branch master
```

```
Changes not staged for commit:
```

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

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified: README.md
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

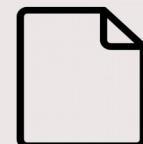
untracked

unmodified

modified

staged

readme.MD





## desfazendo alterações

```
$ git chekout -- README.md
```

```
$ git status  
On branch master  
nothing to commit, working tree clean
```

untracked

unmodified

modified

staged





## Editando arquivos

```
# Alunos:  
- nome aluno  
$ git add .
```

```
$ git status  
On branch master  
Changes to be committed:  
(use "git reset HEAD <file>..." to unstage)  
modified: README.md
```

untracked

unmodified

modified

staged

readme.MD





## desfazendo alterações

```
$ git reset HEAD README.md
```

```
$ git status
```

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: README.md

no changes added to commit (use "git add" and/or "git commit -a")

untracked

unmodified

modified

staged

readme.MD





## Commitando alterações

```
$ git add .
```

```
$ git commit -m "alterando o arquivo"
```

```
[master 1c8d546] alterando o arquivo
```

```
1 file changed, 2 insertions(+)
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

untracked

unmodified

modified

staged





## Exibindo a versão criada:

```
$ git log
```

```
commit 1c8d5466fad29feaf7fb891a6f14418fe92305e3 ( HEAD -> master)
```

```
Author: John Oliver <johnoliver021@gmail.com>
```

```
Date: Sun Nov 18 13:13:43 2018 -0200
```

alterando o arquivo

```
commit 5bf5823b0f40af253343dd9a4d891a42e992f44b ( HEAD -> master)
```

```
Author: John Oliver <johnoliver021@gmail.com>
```

```
Date: Sun Nov 18 13:13:43 2018 -0200
```

primeiro commit



## desfazendo o commit

```
$ git reset --soft 5bf5823b0f40...
```

```
$ git status  
On branch master  
Changes to be committed:  
(use "git reset HEAD <file>..." to unstage)  
modified: README.md
```

untracked

unmodified

modified

staged

readme.MD





## desfazendo o commit

```
$ git commit -m "alterando o arquivo"
```

```
$ git reset --mixed 5bf5823b0f40...
```

```
$ git status
```

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: README.md

no changes added to commit (use "git add" and/or "git commit -a")

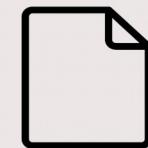
untracked

unmodified

modified

staged

readme.MD





## desfazendo o commit

```
$ git add .
```

```
$ git status  
On branch master  
nothing to commit, working tree clean
```

```
$ git commit -m "alterando o arquivo"
```

```
$ git reset --hard 5bf5823b0f40...
```

untracked

unmodified

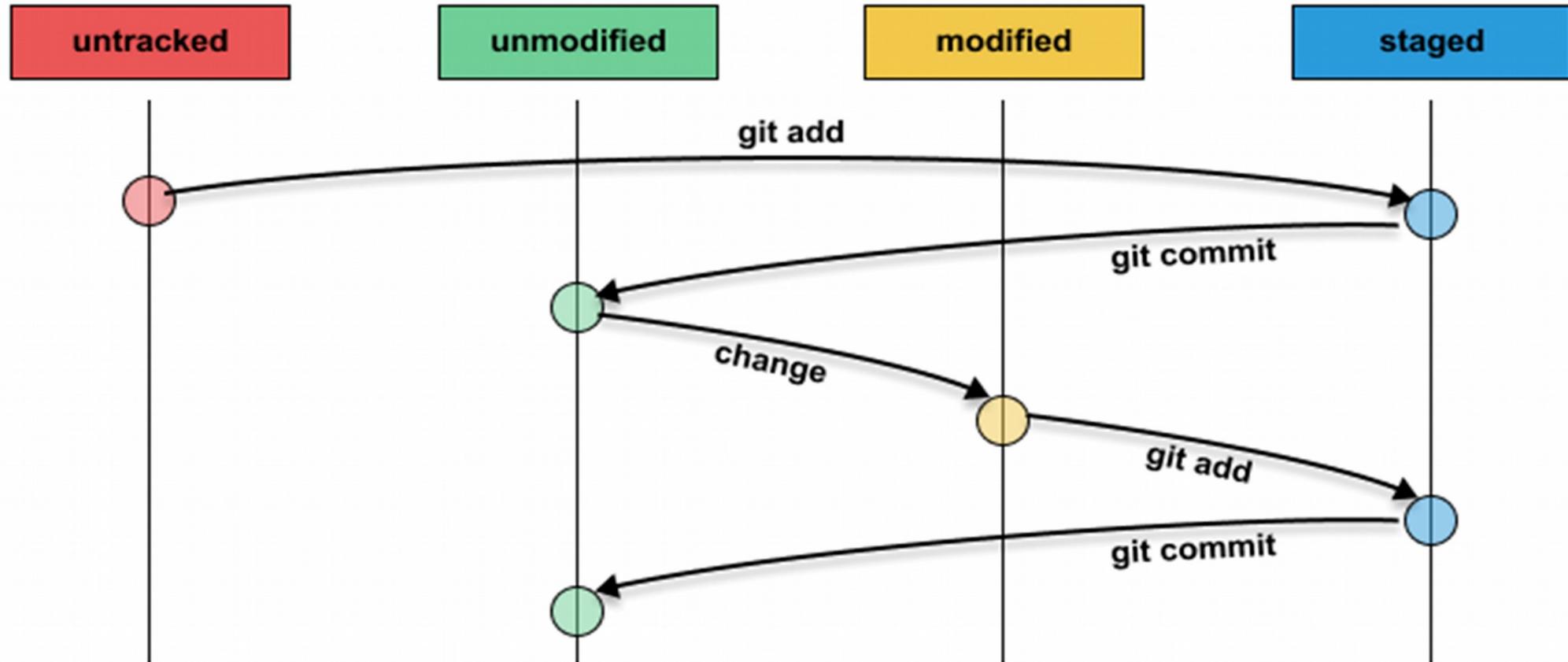
modified

staged





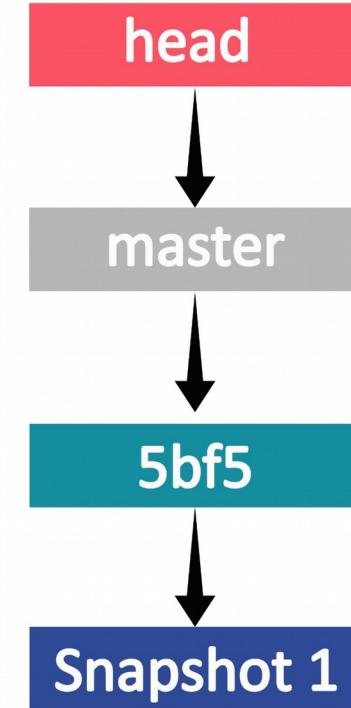
## Ciclo de vida





## Exibindo a versão criada:

# Alunos:

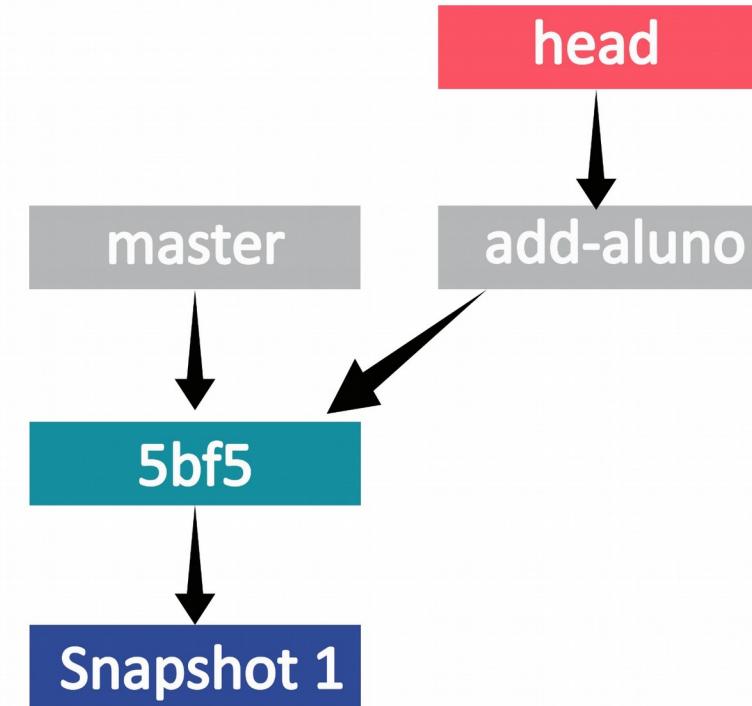




## Criando uma nova branch

```
$ git checkout -b add-aluno
```

```
Switched to a new branch 'add-aluno'
```





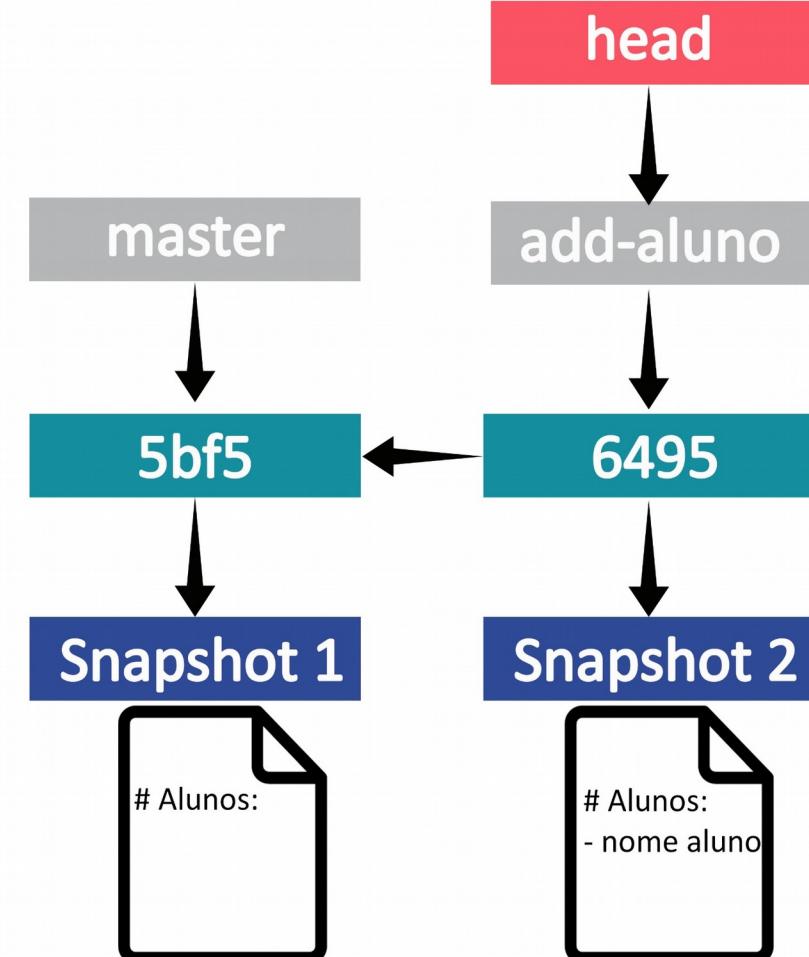
## Criando uma nova versão

```
# Alunos:
```

```
- nome aluno
```

```
$ git add .
```

```
$ git commit -m "adicionando aluno"
```

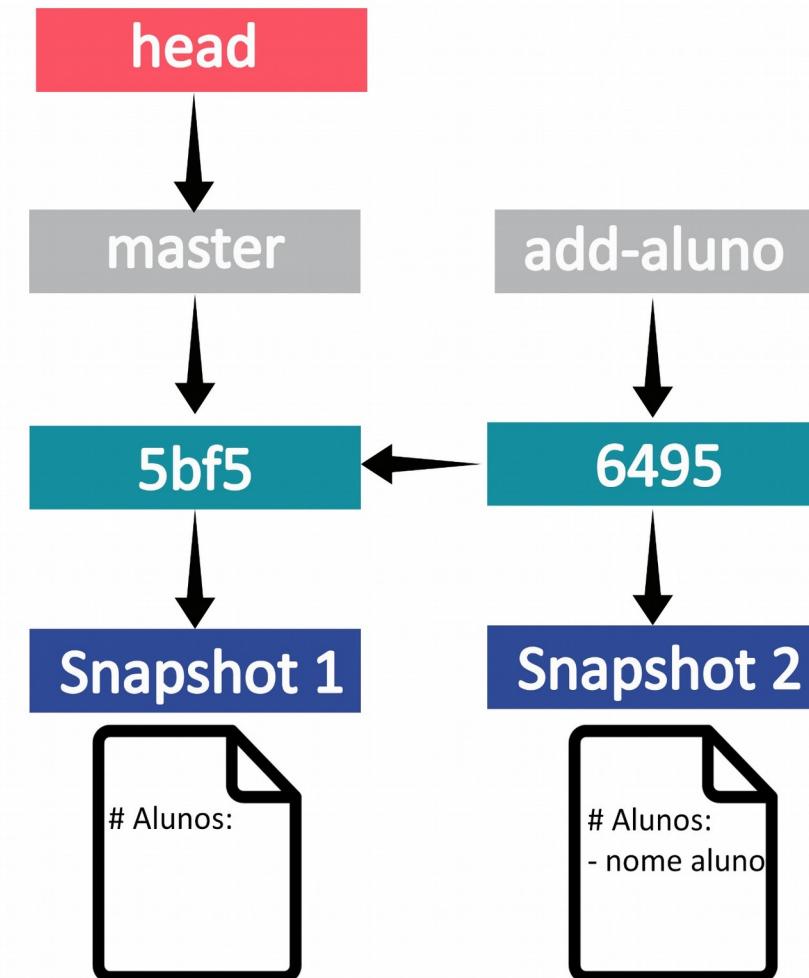




## Voltando pra branch master

```
$ git checkout master
```

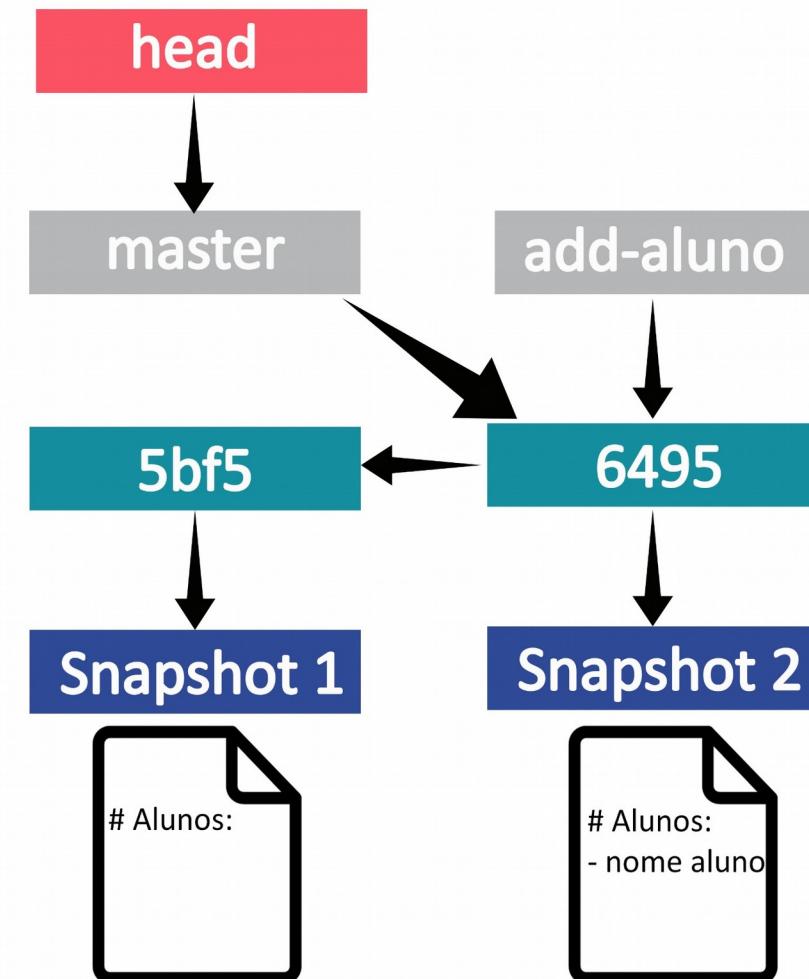
```
Switched to a new branch 'master'
```





## Fazendo o merge da master com a branch add-aluno

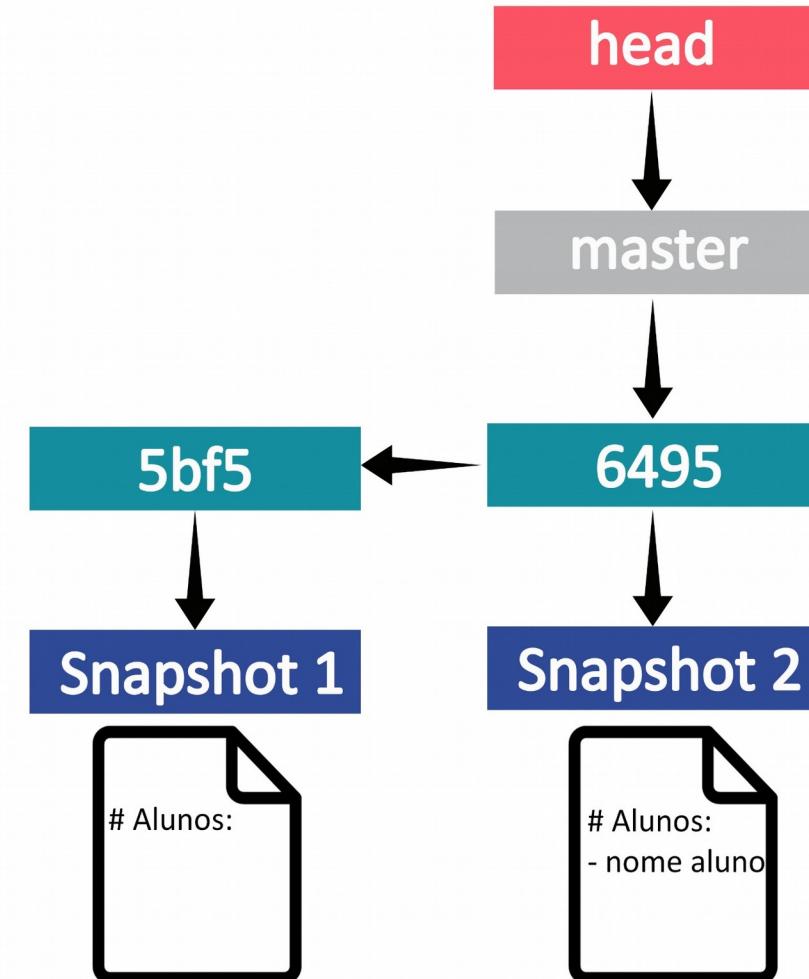
\$ git merge add-aluno





## Deletando a branch add-aluno

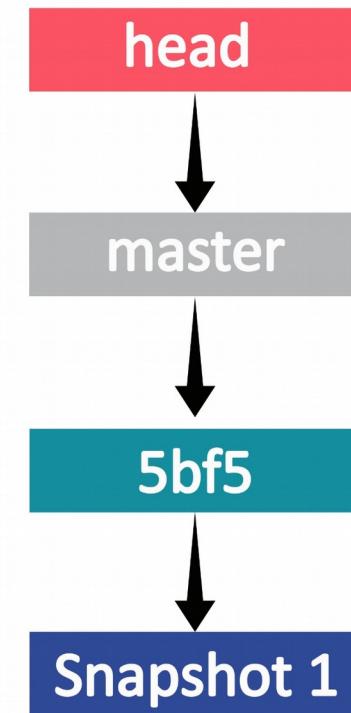
```
$ git branch -D add-aluno
```





## Voltando pra primeira versão

```
$ git reset --hard 5bf5....
```

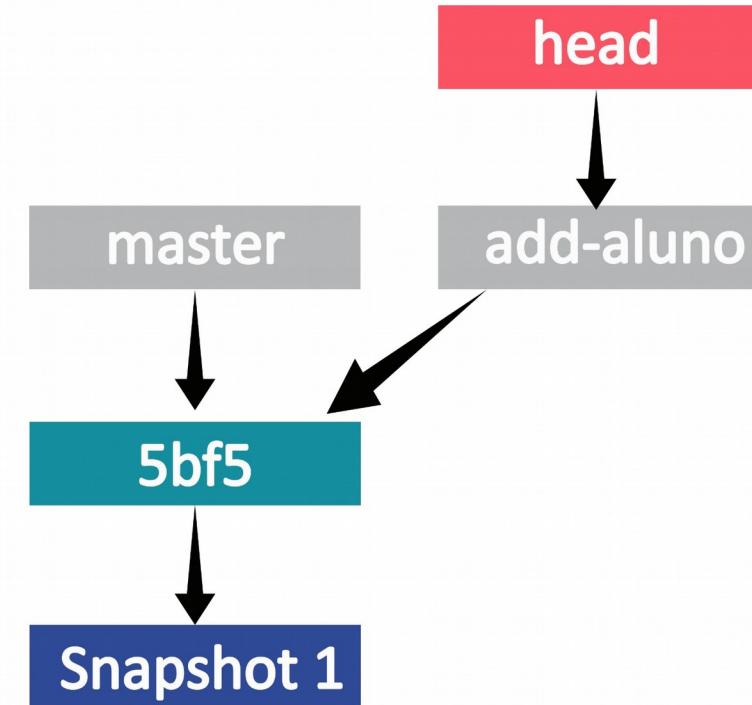




## Criando uma nova branch

```
$ git checkout -b add-aluno
```

```
Switched to a new branch 'add-aluno'
```





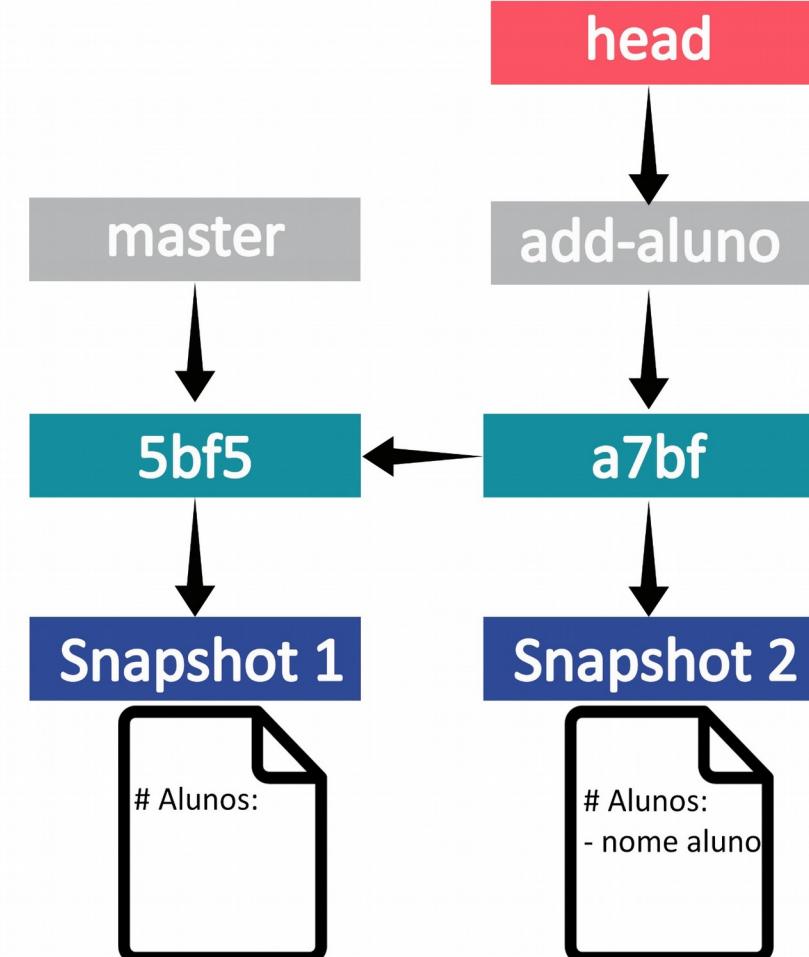
## Criando uma nova versão

```
# Alunos:
```

```
- nome aluno
```

```
$ git add .
```

```
$ git commit -m "adicionando aluno"
```

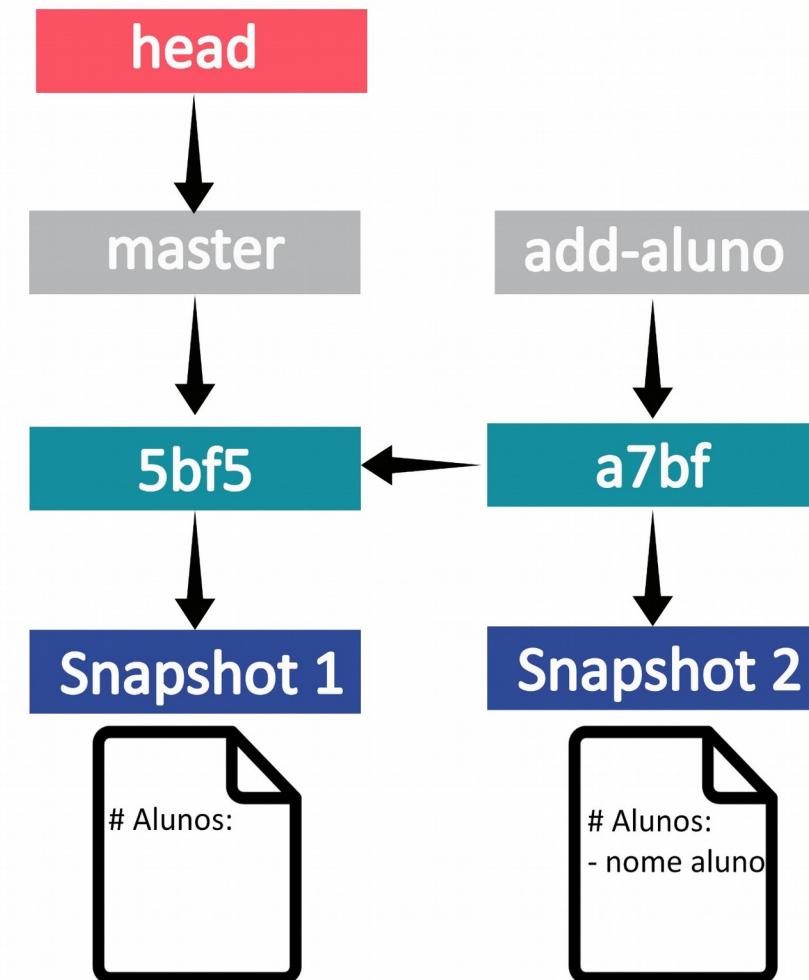




## Voltando pra branch master

```
$ git checkout master
```

```
Switched to a new branch 'master'
```

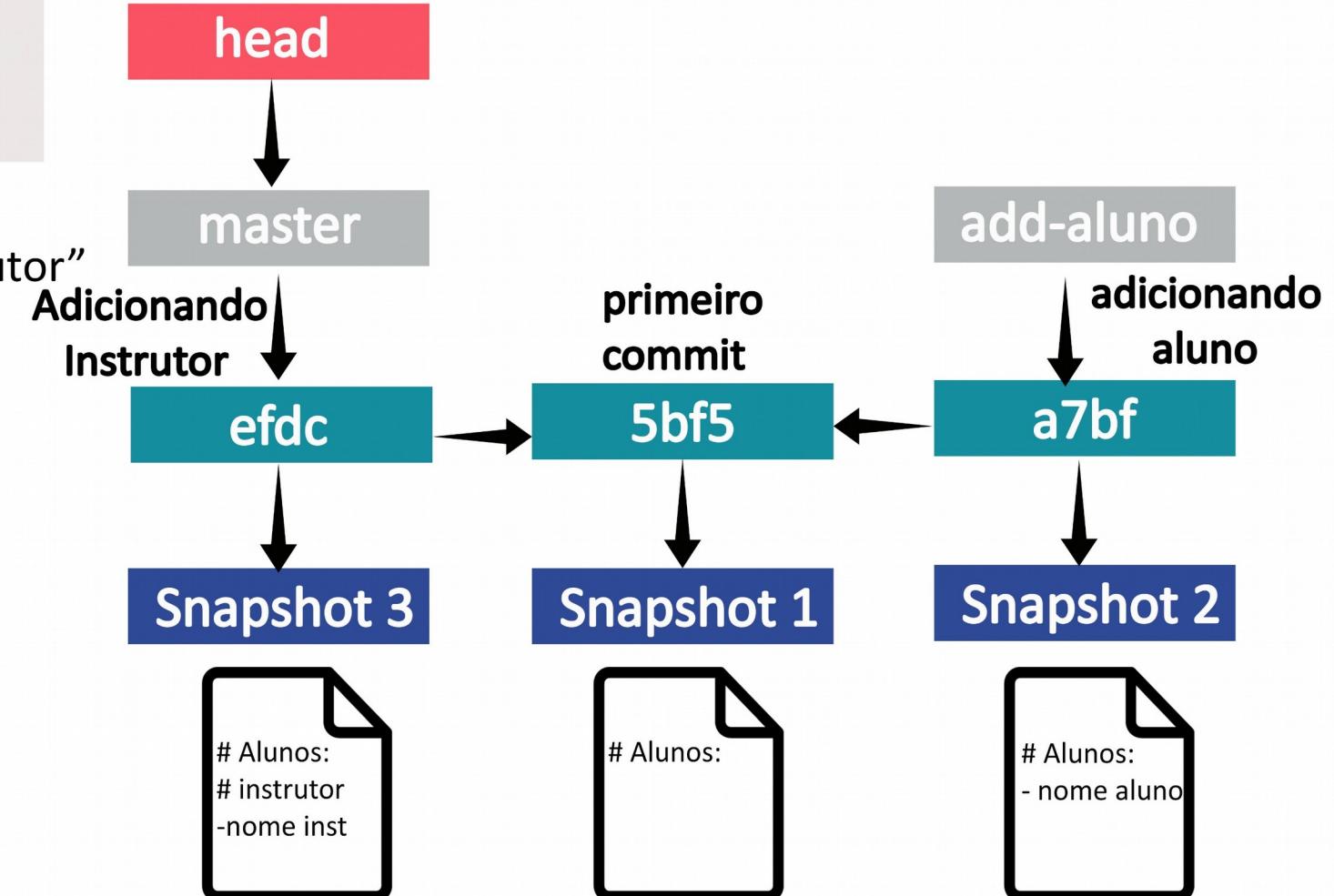




## Adicionando o instrutor

```
# Alunos:  
#Instrutor  
-nome instrutor
```

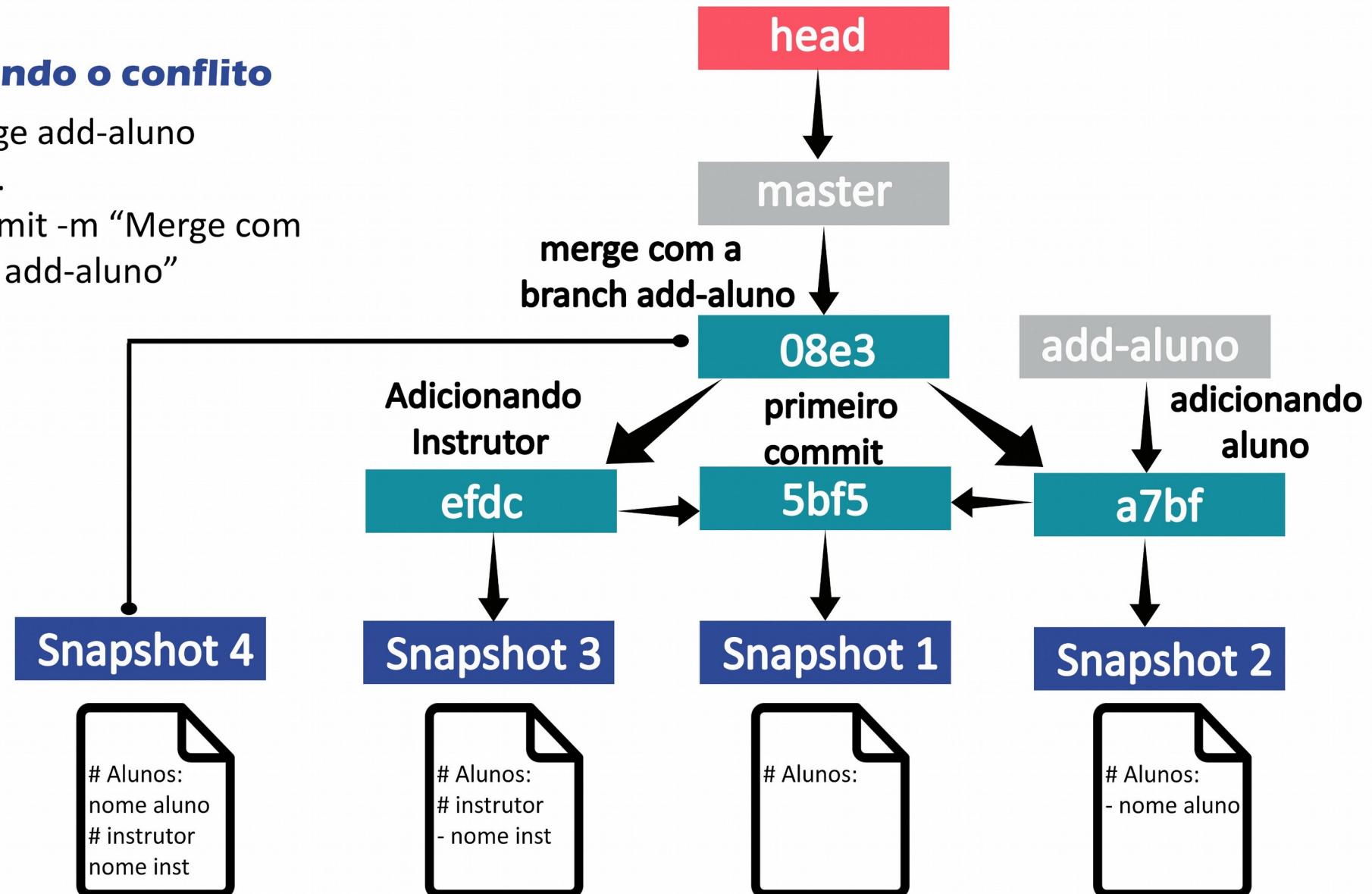
```
$ git add .  
$ git commit -m "adicionando instrutor"
```





## Resolvendo o conflito

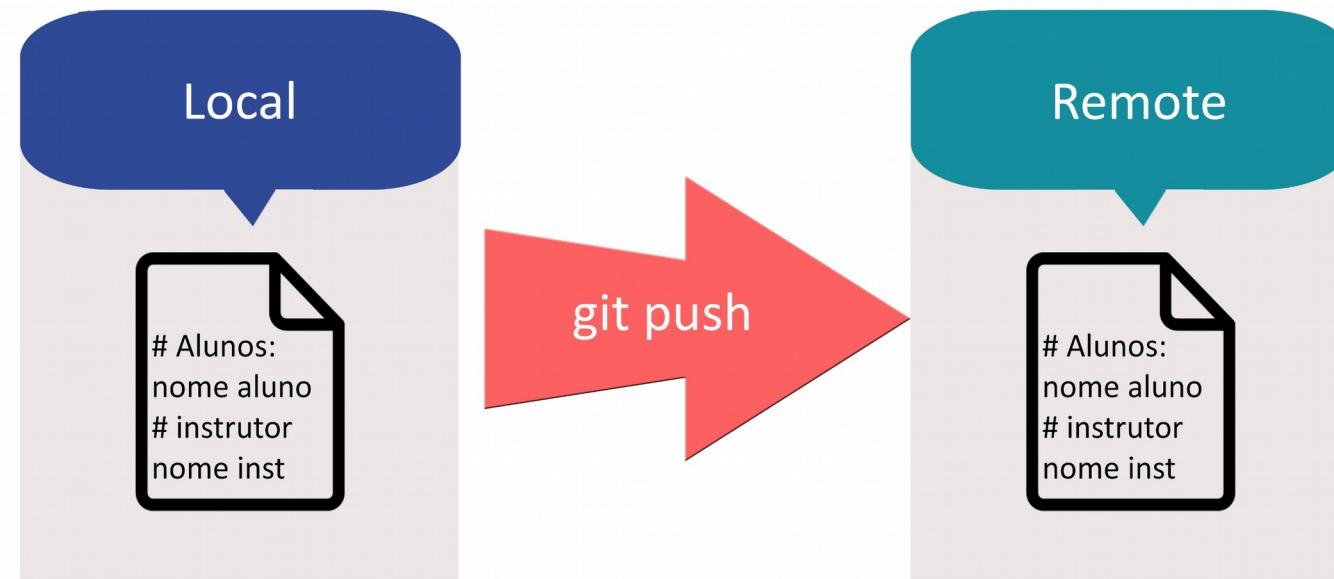
```
$ git merge add-aluno  
$ git add .  
$ git commit -m "Merge com a  
a branch add-aluno"
```





## Ligando o repositório local com o remoto

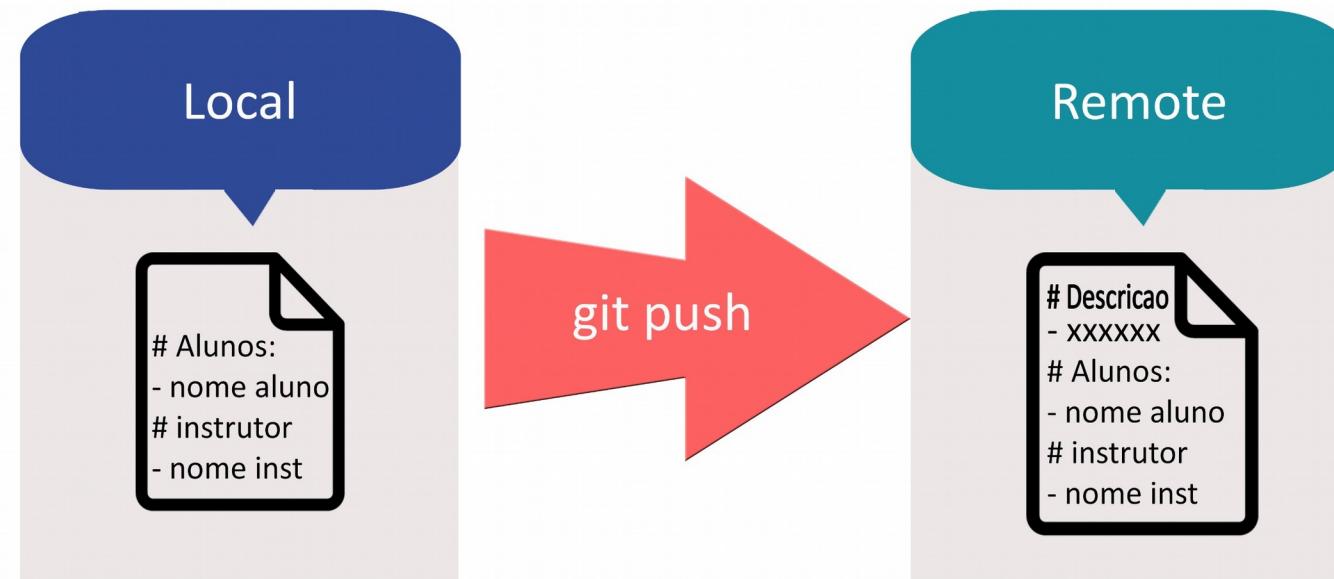
```
git remote add origin https://github.com/nomeUser/Git-Course.git  
git push -u origin master
```





## Editando o arquivo no repositório remoto

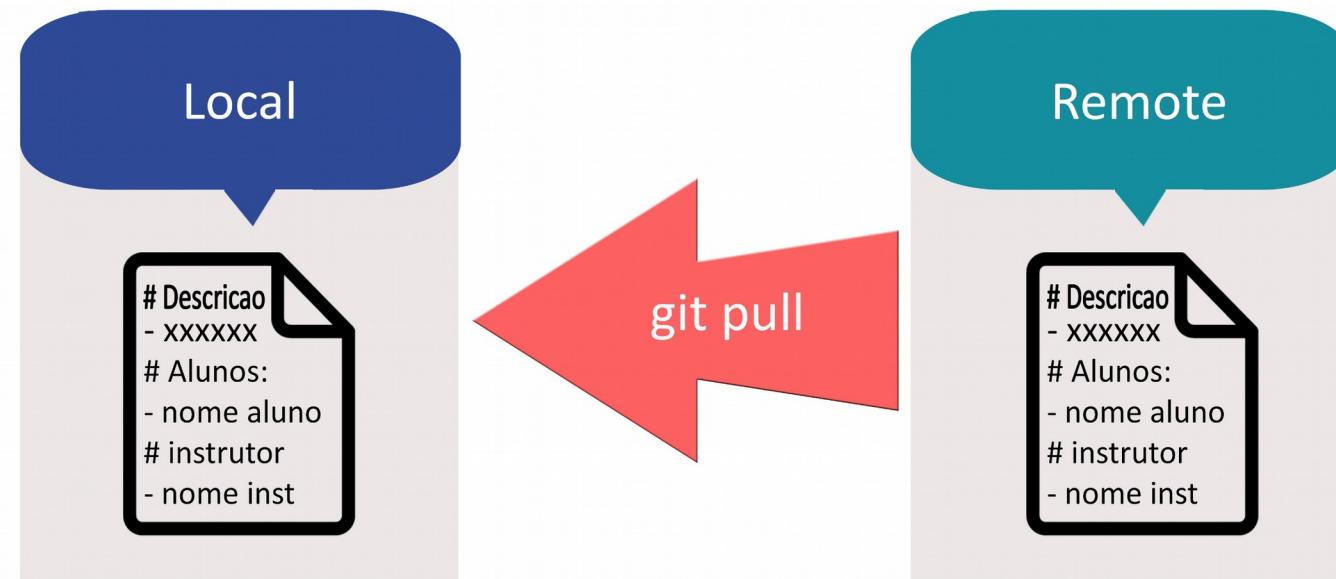
```
# Descrição  
- curso de git e github para iniciantes
```





## Atualizando o repositório local com o repositório remoto

git pull origin master





## Ciclo de vida

```
git remote add origin https://github.com/nameUser/Git-Course.git  
git push -u origin master
```

**Local**



localrepo

remote  
repo

git push

git pull

**Remote**

