

Git - Introdução

Abel Siqueira

30 de Maio de 2015

Por quê?

"FINAL".doc



FINAL.doc!



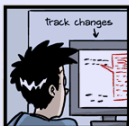
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



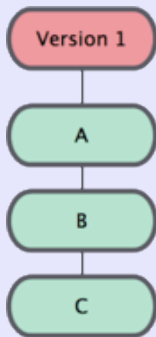
FINAL_rev.18.comments7.
corrections9.MORE.30.doc



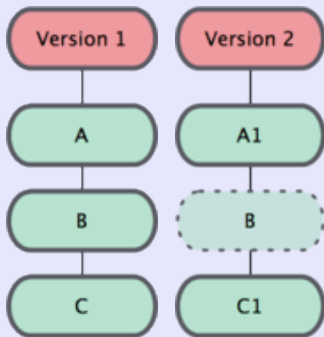
FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCCHOOL????.doc



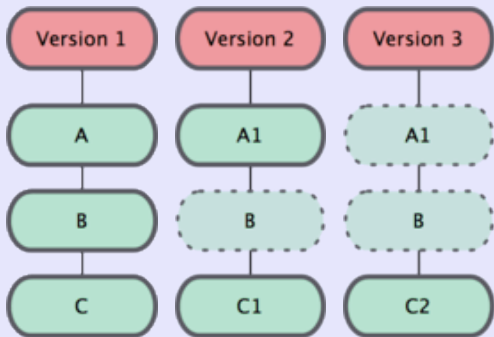
Como funciona?



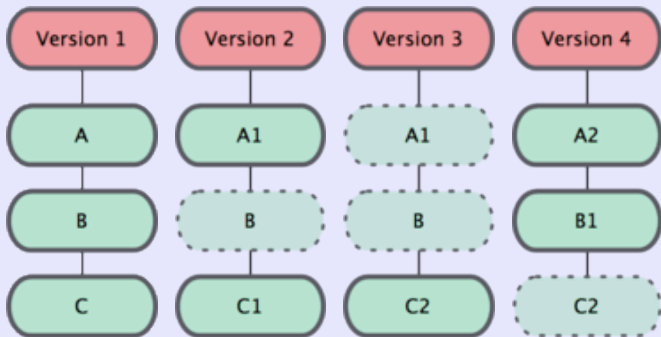
Como funciona?



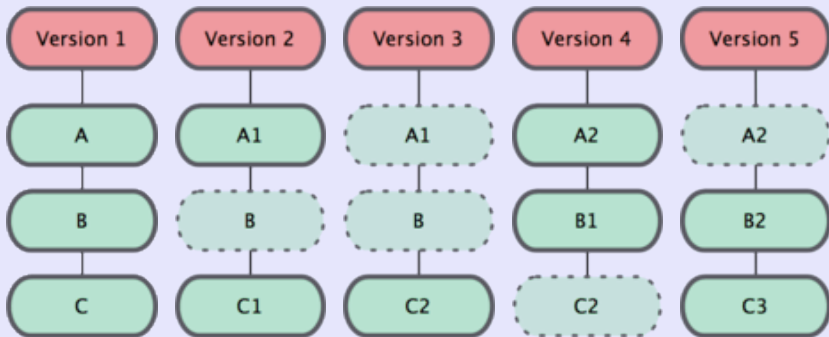
Como funciona?



Como funciona?



Como funciona?



Começando

Uma vez na vida

Começando

Uma vez na vida por computador

Começando

Uma vez na vida por computador

```
$ git config --global user.name "Seu nome"
```

```
$ git config --global user.email seu@email.com
```

```
$ git config --global color.ui auto
```

Começando

```
$ mkdir aula-git
```

```
$ cd aula-git
```

```
$ git init
```

```
$ ls -la
```

Começando

```
$ mkdir aula-git
```

```
$ cd aula-git
```

```
$ git init
```

```
$ ls -la
```

```
$ ls .git
```

Começando

```
$ mkdir aula-git
```

```
$ cd aula-git
```

```
$ git init
```

```
$ ls -la
```

```
$ ls .git # Não mexer
```

Criando e adicionando alguma coisa

```
$ # Escreva 'Meu noem eh Abel' no arquivo teste
```

Criando e adicionando alguma coisa

```
$ # Escreva 'Meu noem eh Abel' no arquivo teste
```

```
$ git status
```

Criando e adicionando alguma coisa

```
$ # Escreva 'Meu noem eh Abel' no arquivo teste
```

```
$ git status
```

```
$ git add teste
```


Criando e adicionando alguma coisa

```
$ # Escreva 'Meu noem eh Abel' no arquivo teste
```

```
$ git status
```

```
$ git add teste
```

```
$ git status
```

Primeiro commit

```
$ git commit -m 'Primeiro commit'
```

Primeiro commit

```
$ git commit -m 'Primeiro commit'
```

```
$ git status
```

Primeiro commit

```
$ git commit -m 'Primeiro commit'
```

```
$ git status
```

```
$ git log
```

Modificando e adicionando mais arquivos

```
$ # Corrija teste (noem para nome)
```

Modificando e adicionando mais arquivos

```
$ # Corrija teste (noem para nome)
```

```
$ git status
```

```
$ git diff
```

Modificando e adicionando mais arquivos

```
$ # Corrija teste (noem para nome)
```

```
$ git status
```

```
$ git diff
```

```
$ # Escreva 'batata' no arquivo compras
```

```
$ git status
```

Modificando e adicionando mais arquivos

```
$ git add teste
```


Modificando e adicionando mais arquivos

```
$ git add teste
```

```
$ # Mude eh para é em teste
```

Modificando e adicionando mais arquivos

```
$ git add teste
```

```
$ # Mude eh para é em teste
```

```
$ git status
```

Modificando e adicionando mais arquivos

```
$ git add teste
```

```
$ # Mude eh para é em teste
```

```
$ git status
```

```
$ git add compras
```

```
$ git status
```

Modificando e adicionando mais arquivos

```
$ git add teste
```

```
$ # Mude eh para é em teste
```

```
$ git status
```

```
$ git add compras
```

```
$ git status
```

```
$ git commit -am 'Segundo commit'
```

```
$ # -am é o mesmo que -a -m
```

Modificando e adicionando mais arquivos

```
$ git add teste
```

```
$ # Mude eh para é em teste
```

```
$ git status
```

```
$ git add compras
```

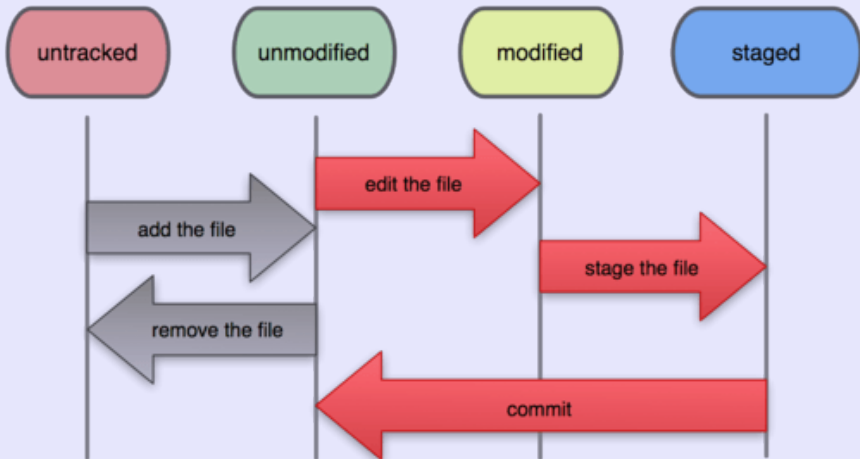
```
$ git status
```

```
$ git commit -am 'Segundo commit'
```

```
$ # -am é o mesmo que -a -m
```

```
$ git log
```

File Status Lifecycle



Já posso escrever meu código?

Já posso escrever meu código?

SIM

Já posso escrever meu código?

SIM

Mas espere. Tem mais.

Mensagens de commit

```
$ Escreva '# Meu código' no arquivo README.md
```

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

Mensagens de commit

```
$ Escreva '# Meu código' no arquivo README.md
```

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

```
$ git commit # Sem -m
```

Mensagens de commit

Título (imperativo, 50-80 caracteres)

Texto da mensagem (70-100 caracteres por linha)

Pode-se escrever a vontade, principalmente para explicar o motivo do commit.

Mensagens de commit

Faça o commit com o título “Adiciona README.md” e o texto “README.md adicionado porque o Abel mandou”.

```
$ git log
```

Mensagens de commit

Faça o commit com o título “Adiciona README.md” e o texto “README.md adicionado porque o Abel mandou”.

```
$ git log
```

```
$ git log --oneline
```

Mensagens de commit

Faça o commit com o título “Adiciona README.md” e o texto “README.md adicionado porque o Abel mandou”.

```
$ git log
```

```
$ git log --oneline
```

```
$ git log --graph
```

```
$ git log --graph --oneline
```

```
$ git log --graph --oneline --decorate
```

Log bonito

```
$ git log --pretty=format:“%ad (%ar) %n%an  
(%ae)%n%s%n%n%b” --graph --date=iso
```


Log bonito

```
$ git log --pretty=format:“%ad (%ar) %n%an  
(%ae)%n%s%n%n%b” --graph --date=iso # Vamos definir  
esse comando como meulog
```

Log bonito

```
$ git log --pretty=format:“%ad (%ar) %n%an  
(%ae)%n%s%n%n%b” --graph --date=iso # Vamos definir  
esse comando como meulog
```

```
$ git config --global alias.meulog ‘log  
--pretty=format:“%ad (%ar) %n%an (%ae)%n%s%n%n%b”  
--graph --date=iso’
```

Log bonito

```
$ git log --pretty=format:“%ad (%ar) %n%an  
(%ae)%n%s%n%n%b” --graph --date=iso # Vamos definir  
esse comando como meulog
```

```
$ git config --global alias.meulog ‘log  
--pretty=format:“%ad (%ar) %n%an (%ae)%n%s%n%n%b”  
--graph --date=iso’
```

```
$ git meulog
```

Help

```
$ git help comando
```

Sumário

```
$ git init
```

```
$ git add
```

```
$ git commit
```

```
$ git status
```

```
$ git log
```

```
$ git diff
```

```
$ git help
```

Faça

Commits frequentes

Commits de assuntos relacionados

Boas mensagens

Não Faça

Commits monolíticos

Inclusão de arquivos que são gerados (facilmente)

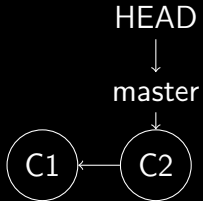
Commits sem testar o trabalho

Fim

Fim?



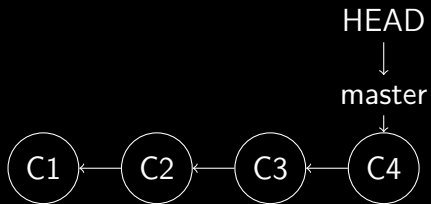
```
$ git commit
```



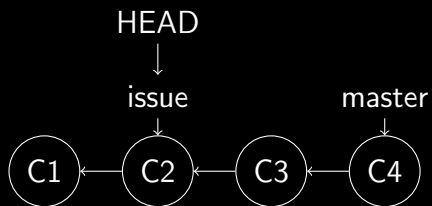
```
$ git commit
```



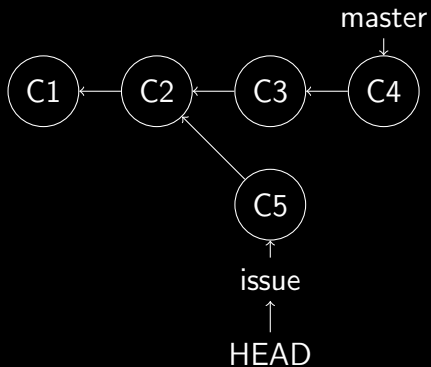
```
$ git commit
```



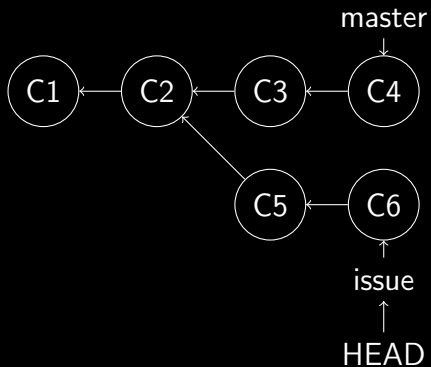
```
$ git commit
```



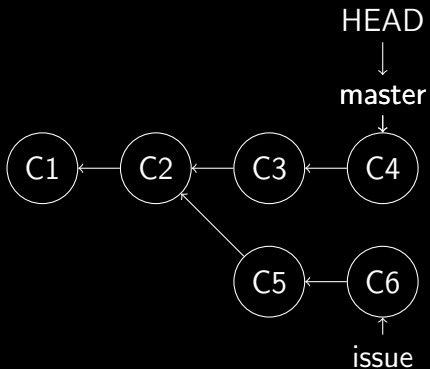
```
$ git checkout -b C2
```



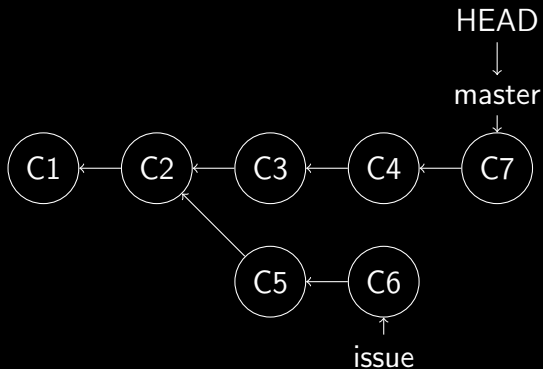
```
$ git commit
```



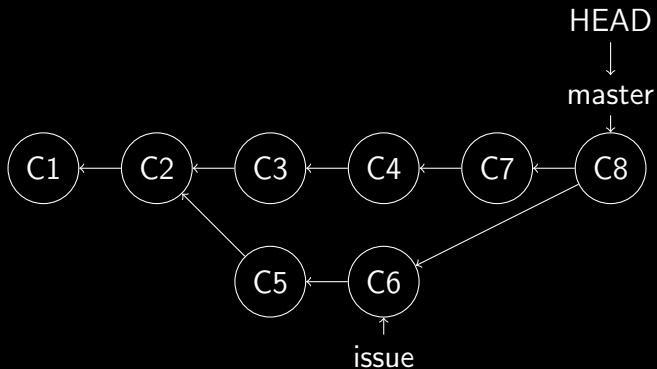
```
$ git commit
```

```
$ git checkout master
```



```
$ git commit
```



```
$ git merge issue
```

Sumário 2

```
$ git checkout
```

```
$ git merge
```

Fim