

Introdução ao Git

Raniere Gaia Costa da Silva¹

24 de maio de 2013

¹r.gaia.cs@gmail.com

Esta apresentação é baseada na apresentação “Intro to Git” de Scott Chacon que encontra-se disponível em

<https://github.com/schacon/git-presentations>.

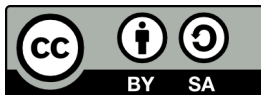
Os arquivos desta apresentação encontram-se disponíveis em

<https://gitorious.org/raniere-presentations/tech>.

Licença

Salvo indicado o contrário, esta apresentação está licenciada sob a licença Creative Commons Atribuição-Compartilhagual 3.0 Não Adaptada. Para ver uma cópia desta licença, visite

<http://creativecommons.org/licenses/by-sa/3.0/>.



1 O que é o Git?

2 Work flow

Git é um sistema de controle de versão distribuído livre desenvolvido² para ser rápido e eficiente.

²A primeira versão foi escrita por Linus Torvalds, o criador do [Linux](#). 

Quase tudo é local

O Git pode ser utilizado sem conexão de internet ou rede local:

- realizar um *diff*,
- visualizar o histórico de um arquivo,
- criar uma nova versão (*commit*),
- retornar para uma versão anterior,
- mudar de ramo,
- juntar ramos (*merge*).

Por este motivo, ele é rápido.

Configuração

Arquivos de configuração

- 1 /etc/gitconfig
- 2 ~/.gitconfig
- 3 .git/config

```
$ git config --global user.name "Seu Nome"  
$ git config --global user.email "seu.email@foobar.com"  
$ git config --global core.editor gedit  
$ git config --global color.ui true
```

Arquivos a serem ignorados

- 1 .gitignore

```
$ git config --global core.excludesfile ~/.gitignore-global
```

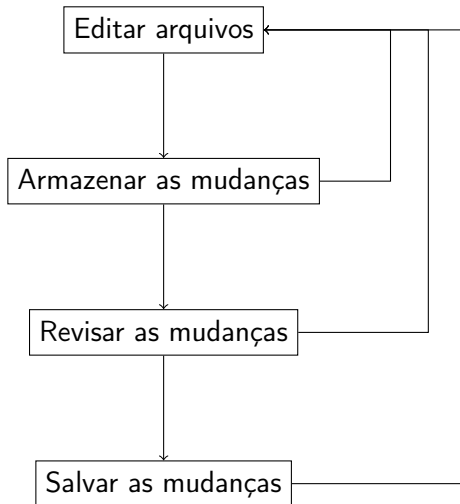
Criando um repositório

```
$ mkdir meu_projeto  
$ cd meu_projeto  
$ git init
```

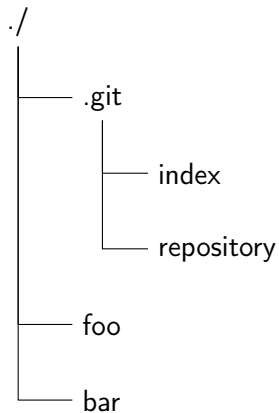
```
$ cd meu_projeto; tree -a
.
+-- .git
    |-- branches
    |-- config
    |-- description
    |-- HEAD
    |-- hooks
    |   |-- applypatch-msg.sample
    |   |-- commit-msg.sample
    |   |-- post-update.sample
    |   |-- pre-applypatch.sample
    |   |-- pre-commit.sample
    |   |-- prepare-commit-msg.sample
    |   |-- pre-rebase.sample
    |   |-- update.sample
    |-- info
    |   |-- exclude
    |-- objects
    |   |-- info
    |   |-- pack
    |-- refs
        |-- heads
        |-- tags
```

10 directories, 12 files

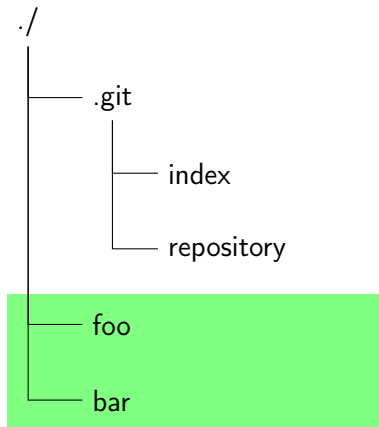
Workflow básico



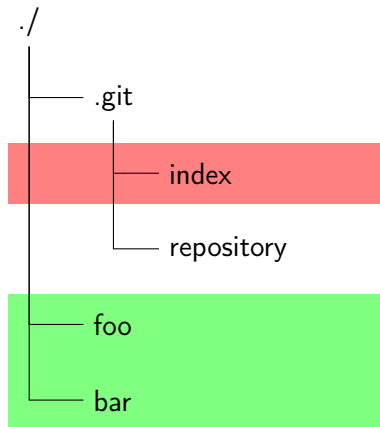
Estrutura de arquivos (ilustrativa)



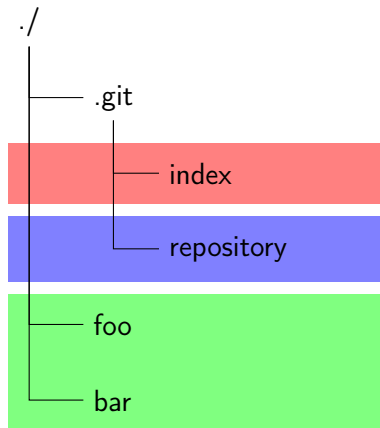
Estrutura de arquivos (ilustrativa)



Estrutura de arquivos (ilustrativa)



Estrutura de arquivos (ilustrativa)



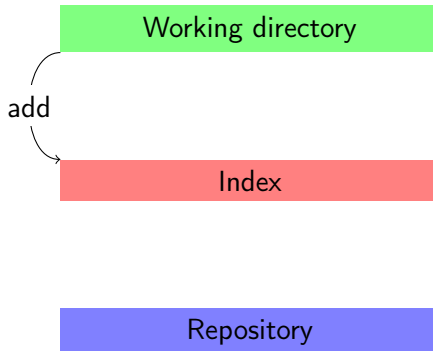
Workflow básico (comandos)

Working directory

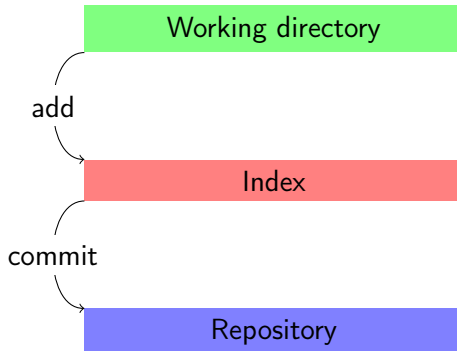
Index

Repository

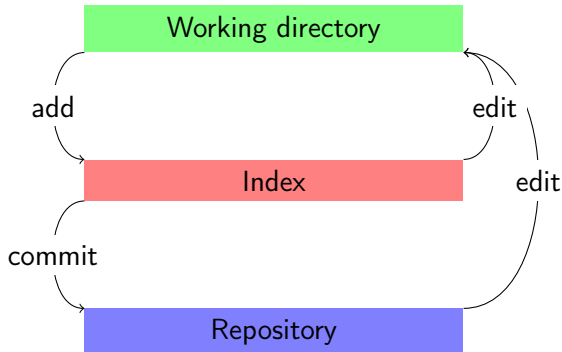
Workflow básico (comandos)



Workflow básico (comandos)



Workflow básico (comandos)



Ajuda

```
$ git help
usage: git [--version] [--help] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]
```

The most commonly used git commands are:

| | |
|--------|--|
| add | Add file contents to the index |
| branch | List, create, or delete branches |
| commit | Record changes to the repository |
| diff | Show changes between commits, commit and working tree, etc |
| init | Create an empty Git repository or reinitialize an existing one |
| log | Show commit logs |
| show | Show various types of objects |
| status | Show the working tree status |

See 'git help <command>' for more information on a specific command.

Status

Descrição

Mostra o estado atual do repositório git corrente.

```
$ git status
```

Add

Descrição

Adiciona o conteúdo/diff de arquivos ao index.

```
$ git add path/to/file
```

```
$ git add path/
```

```
$ git add .
```

```
$ git add -A|--all
```

```
$ git add -u|--update
```

Commit

Descrição

Salva as modificações no repositório como uma nova versão.

```
$ git commit
```

```
$ git commit -a
```

```
$ git commit -m "Some message"
```

Log

Descrição

Mostra o histórico (mensagem de log) das versões (commits).

```
$ git log
```

```
$ git log -N
```

```
$ git log --decorate
```

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

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

Show

Descrição

Mostra vários objetos.

```
$ git show objeto
```

Diff

Descrição

Mostra a diferença entre duas versões (salvas ou não).

```
$ git diff
```

```
$ git diff --staged
```

```
$ git diff commit_hash
```

```
$ git diff commit_hash01 commit_hash02
```


Mais informações

- 1 Documentação (GPL): <http://git-scm.com/documentation>,
- 2 Pro Git (CC-BY-NC-SA): <http://git-scm.com/book>,
- 3 Pragmatic Guide to Git
- 4 Pragmatic Version Control Using Git
- 5 Version Control with Git: Powerful tools and techniques for collaborative software development

Obrigado!

`r.gaia.cs@gmail.com`