



Git

the information manager from
hell



Git **who?**

Git é um sistema de versionamento distribuído.

ou "the stupid content tracker", como descrito pelo criador, Linus, em 2005

Sistema de controle de versões mais utilizado atualmente

Velocidade e grande fluxo de dados

Desenvolvimento paralelo

Distribuído e simples

Poderoso - workflows e tools

[Primeiro commit: "git - o gerenciador de informações do inferno"](#)

[Entrevista com Linus sobre o surgimento do Git.](#)



Myreli.

myreli.github.io

myreli.barros@compasso.com.br | borbamyreli@gmail.com



individual

corporativo

Github Campus Expert

2015

2016

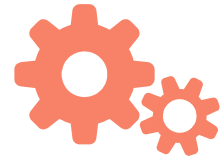
2017

2018

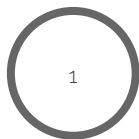


colaborativo

Github Campus
Advisor

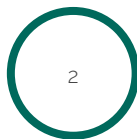


Cronograma (ideia)



Conceitos Básicos

Introdução a sistemas de versionamento e conceitos introdutórios.



Git man

Conceitos básicos de Git e Github.



Git commit

Versionamento individual



Git blame

Versionamento colaborativo

Versionamento

Introdução aos sistemas de controle de versão.



git



```
Ctrl + Z  
Ctrl + Z  
Ctrl + Z  
Ctrl + Z  
Ctrl + Z  
Ctrl + Z  
Ctrl + Z
```



O que e **por** quê?

VSC (version control system)

- arquivo.txt, arquivo-2.txt, arquivo-2-agoravai.txt, arquivo-FINAL.txt
- ctrl-z, ctrl-y
- SVN, Mercurial, Git, ...

Sistemas de controle de versão servem para gerenciar as versões de um arquivo ou projeto.



CVCSs **VS** DVCSs

Centralizados

- CVS, Subversion, ...
- Um servidor com todo o versionamento e vários clientes que solicitam a última versão de lá
- Se o servidor falha? Ops.

Descentralizados

- Git, Mercurial, ...
- Clientes possuem todo o projeto, versionamento e histórico
- Se alguém falha? Todo o resto é backup.



Primeiros passos

Introdução ao Git e Github.

Noções básicas

Conceitos básicos sobre Git e Github — a
teoria do versionamento.





Git **VS** Github

Git - Sistema de Versionamento

Git é uma ferramenta de versionamento de arquivos, pode ser usada em qualquer rede — ou em nenhuma, somente local.

Github - hosting

Github é um serviço de hospedagem de repositórios com suporte a Git ou SVN. Está para o Git como SourceForge está para SVN.



Github é o host com mais repositórios **Git**

E tem o octocat, então foi o
escolhido para o curso.

Existem alternativas como
Bitbucket, GitLab, etc...





Por trás do **git**

Snapshots

cada *commit* faz snapshot das alterações

Processamento local

Diminui muito o tempo de operação

Deltas

cada commit faz o delta de alterações

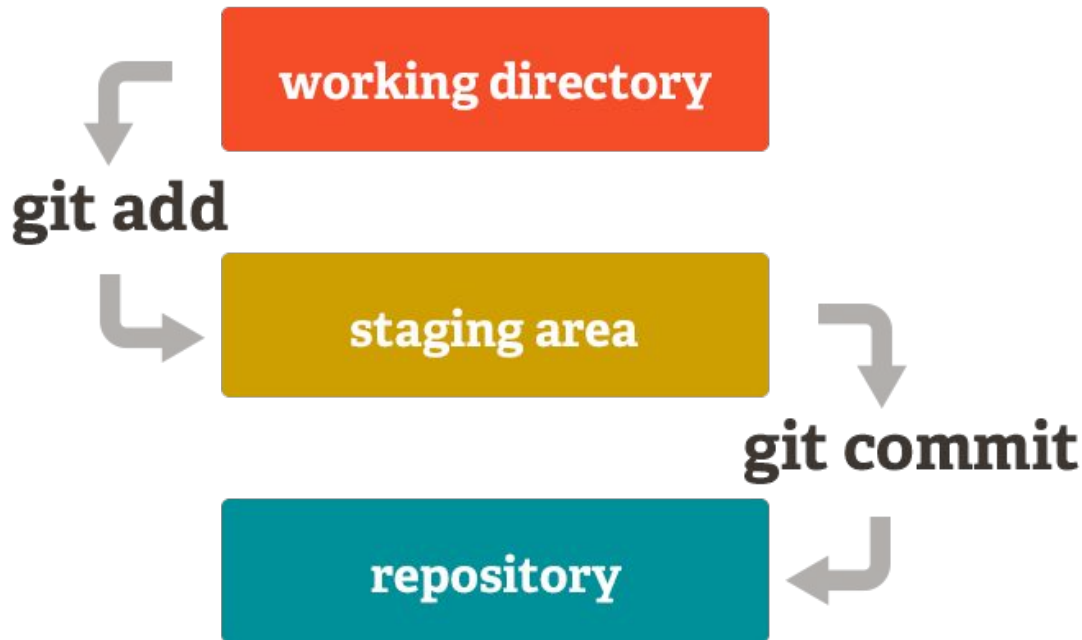
Processamento local e remoto

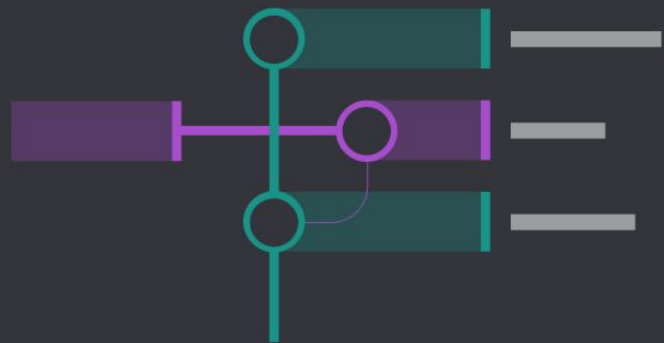
Ocorre muita interferência de rede



States do Git

modified
staged
committed





X



<https://git-scm.com/downloads>
<https://support.gitkraken.com/pdf/infographic/gitkraken-vs-cli.pdf>



git command --options

estrutura mais básica dos comandos git

git help [<args>]

ajuda sobre os comandos e utilização

git [flags] <command> [<args>]

the stupid content tracker

GIT TOOLS



CTRL+Z



**REWRITING
YOUR CODE**



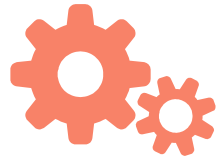
inglipose

How to maintain a working
version of you code

3

Versionando com Git

Everyday Git



git config [args]

<https://git-scm.com/docs/git-config>

Escopo gitconfig

repositório < usuário < sistema

--local

--global

--system

Configurações

```
$ git config --list
```

```
$ git config --show-origin <arg>
```

Identidade

```
$ git config --global user.name  
"Myreli Borba Barros"
```

```
$ git config --global user.email  
myreli.barros@compasso.com.br
```

Editor

```
$ git config core.editor notepad
```



git **init** [args]

<https://git-scm.com/docs/git-init>

Cria um repositório no diretório atual

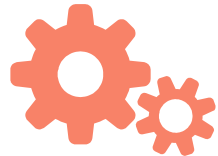
cria o diretório .git contendo espaços para os objetos, heads e tags e a branch master.

Criar repositório

```
$ git init planetas
```

Saber o que está acontecendo

```
$ git status
```



Registrando alterações

Controle de alterações nos arquivos

```
$ echo '#planetas' > README.md
```

```
$ ls
```

```
$ git status --short
```

```
$ echo 'saturno' > nomes
```

```
$ git status
```

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

Inspecionando mudanças

```
$ git diff
```

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

Mover e excluir arquivos

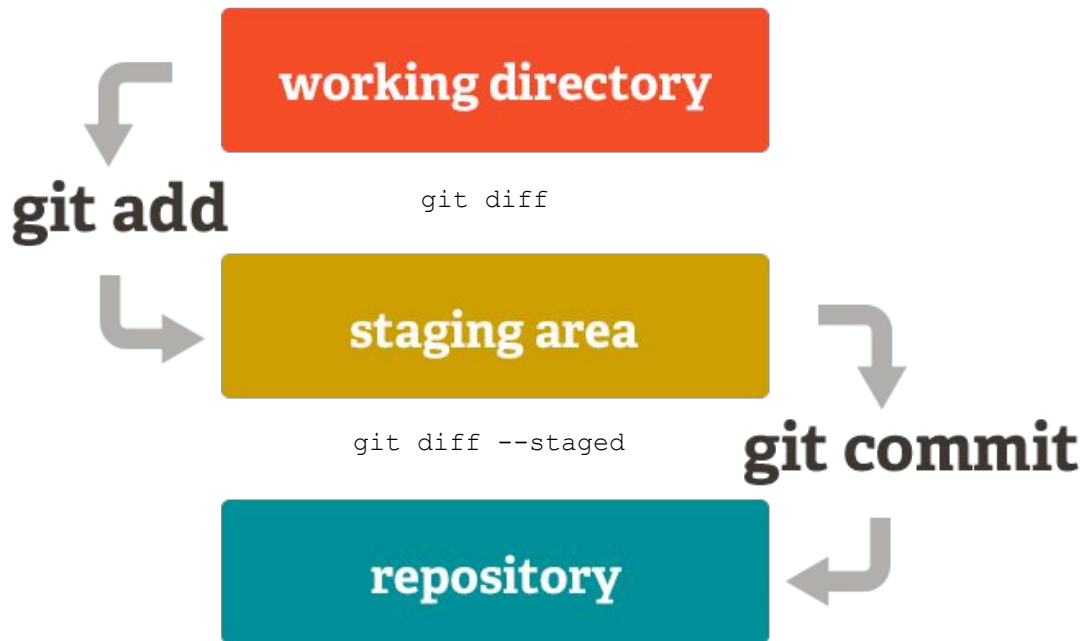
```
$ git mv antigo novo
```

```
$ git rm arquivo
```



States do Git

modified (or
untracked)
staged
committed





git commit

Snapshot

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

```
$ git status
```

```
$ git log
```

Corrigindo commit

```
$ git commit --amend
```

Verificar histórico

```
$ git log --pretty="%h - %s"
```

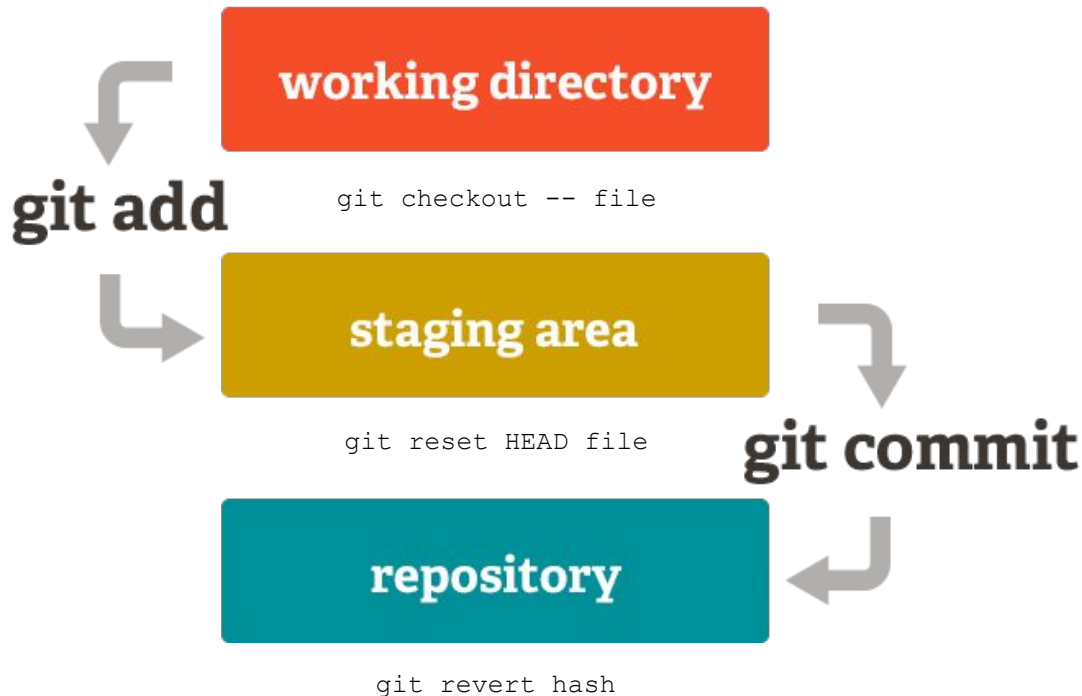


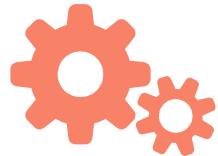
States do Git

modified (or
untracked)

staged

committed





git tag

Listar tags

```
$ git tag
```

```
$ git show v1.0
```

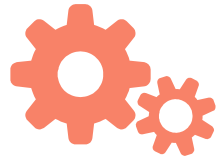
Criando tag

```
$ git tag -a v1.0 -m "primeira  
versão"
```

Criando tag leve (temporária)

```
$ git tag v1.0-leve
```

```
$ git tag -d v1.0-leve
```



git branch

Criar branch

```
$ git branch desenv
```

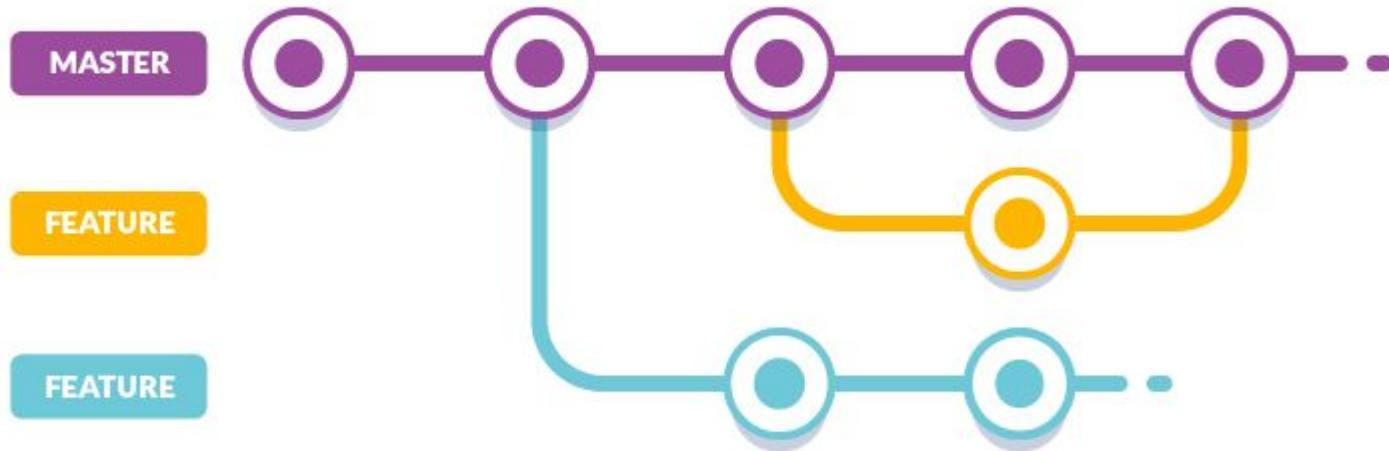
```
$ git checkout desenv
```

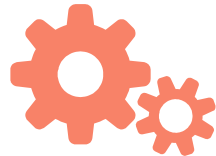
Atalho:

```
$ git checkout -b novabranch
```

Histórico com branches

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





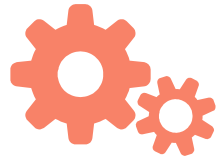
git merge

```
$ git checkout master
```

```
$ git merge desenv
```

Conflitos nos merges

<https://media.giphy.com/media/cFkiFMDg3iFol/giphy.gif>



git origin

```
$ git remote add origin  
https://github.com/user/repo.git
```

Criar repositório no GitHub

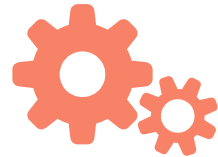
```
$ git push -u origin master  
$ git push origin --tags
```

<https://github.com/new>

```
$ git remote show origin
```

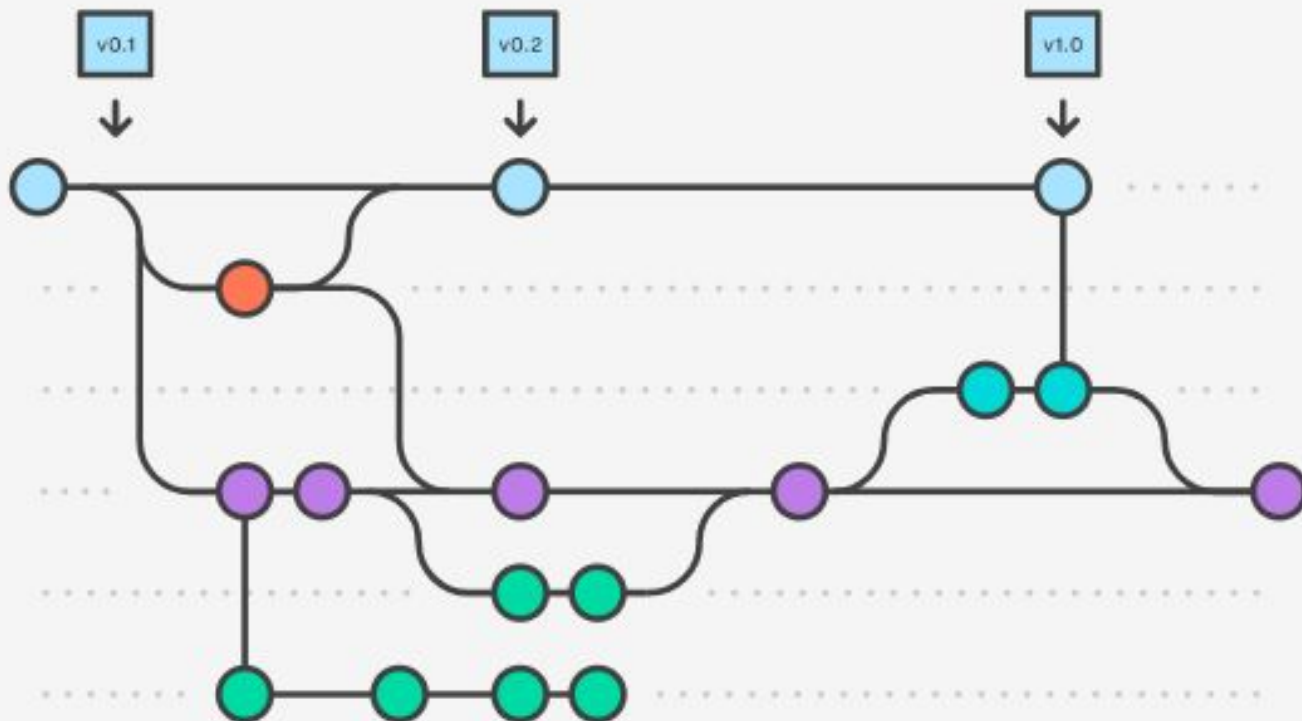
Versionando com Git

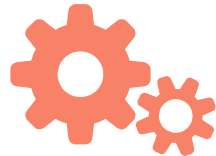
Tópicos avançados



git workflow

- Fluxo de branches
- Frequencia de commits
- Uso de tags





git **stash**

```
$ git stash
```

```
$ git stash list
```

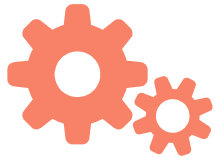
```
$ git stash apply
```

```
$ git stash drop
```

```
$ git stash pop
```



**KEEP
CALM
AND
GIT
STASH**

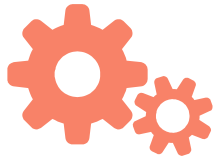


git updates

```
$ git fetch
```

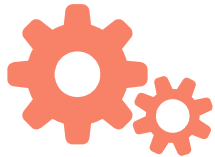
```
$ git pull
```

```
$ git push
```



git cherry-pick

```
$ git cherry-pick <commit-hash>
```

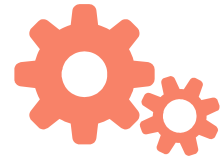


git rebase

```
$ git checkout -b feature
```

```
// alterações
```

```
$ git rebase master
```



git sources

<https://git-scm.com/docs>

<https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>

http://rogerdudler.github.io/git-guide/index.pt_BR.html

[Pro Git book](#) 2nd Edition by Scott Chacon and Ben Straub



Git

the information manager from
hell