
Desmistificando o



git

Ícaro Jerry

\$whoami

- Graduado em ADS, IFBA
- Mestrando do PPGESP, IFBA
- Analista de Sistemas, Escavador
- Entusiasta do Software Livre



O que seria esse tal de git?

- Sistema de Controle de Versão **Multiplataforma**
- Um projeto de **Software Livre**
- Utilizado em **diversos** artefatos
- Release **inicial** em **2005**
- Mesmo **criador do Linux**
- Primeira release em **43 dias**
- Atualmente mantido por Junio Hamano



—

Mas por que eu
deveria usar o **git**?



"FINAL".doc



FINAL.doc!



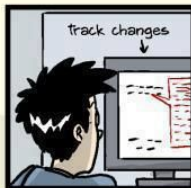
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



JORGE CHAM © 2012

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



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



–

“Deixar que um programa lide com o **controle de versão** do seu projeto permite que você **foque no seu projeto.**”

Mas por que eu deveria usar o git?

- Fácil
- Leve (~44MB)
- Rápido
- Escalável
- Distribuído
- Software Livre



Mas por que eu deveria usar o git?

- **Registro de alterações** com autor, data e hora
- **Sem perdas** de arquivos versionados
- Possibilidade de **trabalho remoto**
- Diversidade de **materiais disponíveis**
- Permite **desenvolvimento não-linear**
- **Alerta de conflitos** entre o trabalho de duas ou mais pessoas



—
In a Nutshell, Git...

... teve **56,494 commits** feitos
por **1.739 contribuidores**
representando **465.301 linhas**
de código...



Fonte

<https://www.openhub.net/p/git>

—
In a Nutshell, Git...

... é principalmente **escrito** em
C, com pelo menos **19%** do seu
código-fonte sendo
comentários ...



Fonte

<https://www.openhub.net/p/git>

—
In a Nutshell, Git...

... estimados **127 anos** de esforço,
começando em **abril de 2005** e
terminando com o sua modificação mais
recente há **4 dias** atrás ...



Fonte

<https://www.openhub.net/p/git>



Required for any path

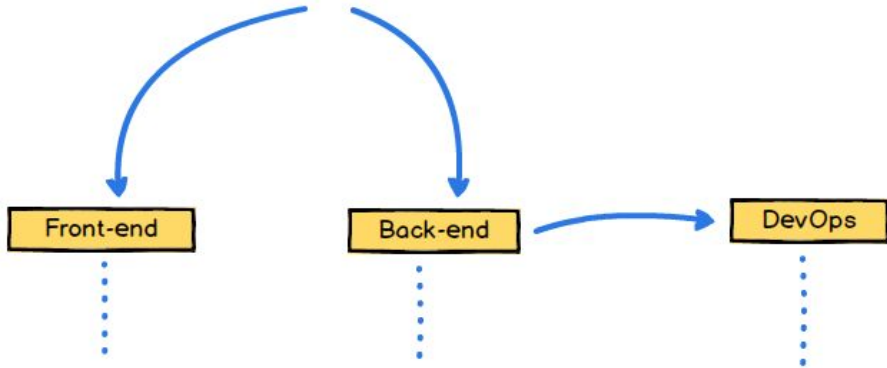
- Git - Version Control
- Basic Terminal Usage
- Data Structures & Algorithms
- SOLID, KISS, YAGNI
- GitHub
- Licenses
- Semantic Versioning
- SSH
- HTTP/HTTPS and APIs
- Design Patterns
- Character Encodings

Legends

- Personal Recommendation!
- Available Options

Web Developer in 2019

Choose your path



—

Mas quem usa o
git?

Google



Microsoft

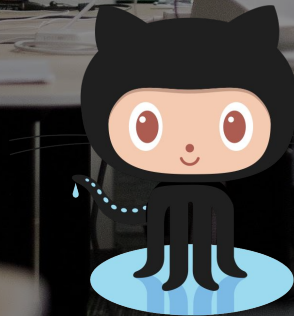
LinkedIn



NETFLIX

facebook

Plataformas on-line de hospedagem de projetos





As plataformas on-line Git

Atualmente diversas plataformas de hospedagem de projetos estão disponíveis. Com elas é possível **ampliar o poder de gerenciamento dos seu projetos!**

- **Gerenciamento de Times**
- **Codereview**
- **Issues**
- **Pull Requests**
- **Ferramentas de Integração**
- **Fork**
- **Wikis**
- **Portfólio Pessoal**



Página

<https://bitbucket.org>

Bitbucket



7 anos de existência

Pertence a Atlassian, dona do Trello

Integrada ao Jira, HipChat e Confluence

10 mi de usuários

28 mi de repositórios

1 mi de organizações

- Paypal, Ford



Página

<https://gitlab.com>

GitLab



8 anos de existência

Software Livre, Licença MIT

Desde 2013, GitLab Community Edition e GitLab Enterprise Edition

Usado por mais de 100k de Organizações

- Bayer, NASA, Sony e Uber
- IPqM (Marinha do Brasil), SERPRO



Página

<https://github.com>

GitHub



11 anos de existência

36 mi de usuários

2,1 mi de organizações

- Facebook, Microsoft, Airbnb, Spotify, Slack

100 mi de projetos

Microsoft compra a plataforma em 2018 (\$7,5 bi)

Atualmente é a maior e mais popular host de Git do mundo



Search or jump to...

Pull requests Issues Marketplace Explore



Microsoft

Report abuse

Open source, from Microsoft with love

Redmond, WA <https://opensource.micro...> opensource@microsoft.c... Verified

Repositories 2.6k Packages People 4.2k Projects 2

Pinned repositories

<p>vscode</p> <p>Visual Studio Code</p> <p>TypeScript ★ 81.1k 🗄️ 11.5k</p>	<p>TypeScript</p> <p>TypeScript is a superset of JavaScript that compiles to clean JavaScript output.</p> <p>TypeScript ★ 52.5k 🗄️ 7.2k</p>	<p>dotnet</p> <p>This repo is the official home of .NET on GitHub. It's a great starting point to find many .NET OSS projects from Microsoft and the community, including many that are part of the .NET Foundation.</p> <p>HTML ★ 11.4k 🗄️ 1.8k</p>
<p>calculator</p> <p>Windows Calculator: A simple yet powerful calculator that ships with Windows</p> <p>C++ ★ 18k 🗄️ 2.8k</p>	<p>monaco-editor</p> <p>A browser based code editor</p> <p>JavaScript ★ 16.5k 🗄️ 1.4k</p>	<p>terminal</p> <p>The new Windows Terminal, and the original Windows console host -- all in the same place!</p> <p>C++ ★ 50.4k 🗄️ 4.9k</p>

Find a repository... Type: All Language: All

Recognizers-Text

Microsoft.Recognizers.Text provides recognition and resolution of numbers, units, and date/time expressed in multiple languages (CN, EN, FR, ES, PT, DE. Partial support for NL, JA, KO). Contributions are greatly welcome! Packages are available at <https://www.nuget.org/profiles/Recognizers.Text> and <https://www.npmjs.com/~recognizers.text>

nlp datetime parser-library timex entity-extraction nlp-entity-extraction numex



Top languages

C# TypeScript JavaScript C++ PowerShell

Most used topics

azure microsoft python



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



Linus Torvalds
torvalds

Follow

Linux Foundation

Portland, OR

Block or report user

Organizations



Overview [Repositories 6](#) [Projects 0](#) [Stars 2](#) [Followers 98k](#) [Following 0](#)

Popular repositories

[linux](#)

Linux kernel source tree

C 79k 27.5k

[uemacs](#)

Random version of microemacs with my private modifications

C 414 63

[test-tlb](#)

Stupid memory latency and TLB tester

C 227 72

[pesconvert](#)

Brother PES file converter

C 117 12

[subsurface-for-dirk](#)

Forked from Subsurface-divelog/subsurface

Do not use - the real upstream is Subsurface-divelog/subsurface

C++ 92 32

[libdc-for-dirk](#)

Forked from Subsurface-divelog/libdc

Only use for syncing with Dirk, don't use for anything else

C 48 19

2,359 contributions in the last year



Contribution activity

2019

August 2019

2018

Created 76 commits in 1 repository

On a beautiful April day on GitHub...

80k

Repositories
Updated

7k

People pushed
their first repository

4k

People forked
a repository
for the first time

3k

People created
their very first
pull request

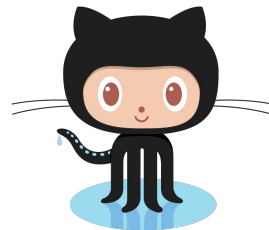
30k

Issues created

12k

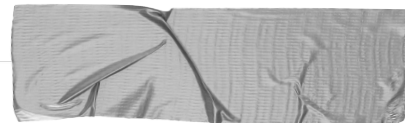
Pull Requests
created

And **6k** people signed up to join the fun.



GitHub

Relatório de atividades de um único dia no
GitHub, feito em maio de 2012.




Página

<https://github.blog/2012-05-01-data-at-github/com>

Principais projetos de código aberto

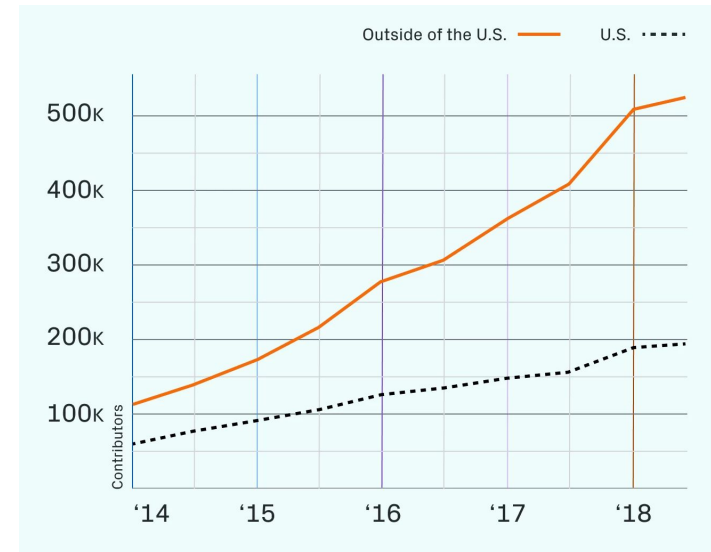
	Contributors
1 Microsoft/vscode	19k
2 facebook/react-native	10k
3 tensorflow/tensorflow	9.3k
4 angular/angular-cli	8.8k
5 MicrosoftDocs/azure-docs	7.8k
6 angular/angular	7.6k
7 ansible/ansible	7.5k
8 kubernetes/kubernetes	6.5k
9 npm/npm	6.1k
10 DefinitelyTyped/DefinitelyTyped	6.0k



GitHub

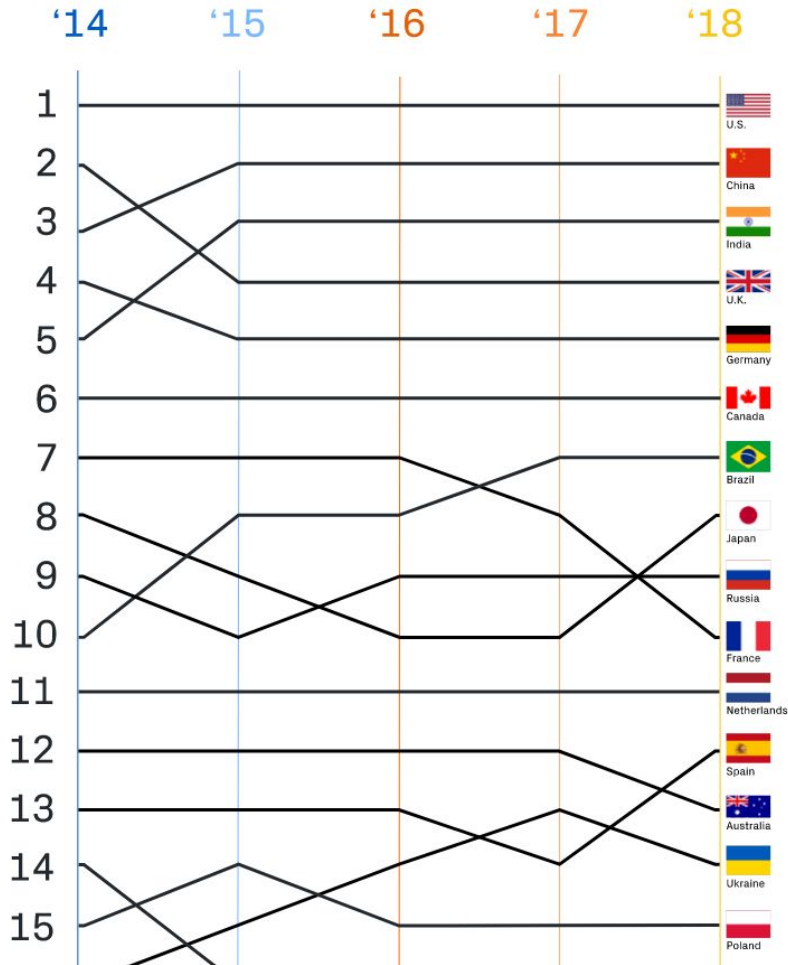


Onde esses repositórios são criados

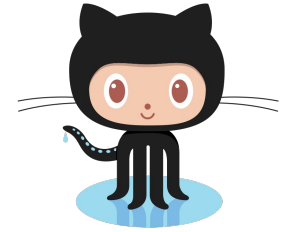


Página

<https://github.blog/2018-11-08-100m-repos/>

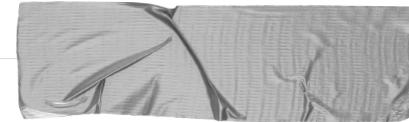


GitHub



Mais algumas informações interessantes...

Prometo que é a última :D



Página

<https://octoverse.github.com>



GitHub Education Package



→ <https://education.github.com/pack>

Ferramentas Gráficas

git



LAZY GIT



Ferramentas Gráficas Git

O Git possui ferramentas GUI embutidas, porém existem diversas outras feitas por terceiros que adicionam uma nova experiência de usuário

→ **Gitk (oficial)**

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

→ **Git-gui (oficial)**

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

→ **GitKraken**

<https://www.gitkraken.com/git-client>

→ **Lazygit**

<https://github.com/jesseduffield/lazygit>



github.com/jesseduffield/lazygit master x (ruby-2.4.1)

5m  



libgit2 index_dirty

Viewing 466/466 Show All
Filter (38 + Option + f)

LOCAL 2/2
ethomson
index_dirty
master 10<

REMOTE 406/406
Libgit2
appveyor
bindings
libgit2sharp
020_2
022_1
brianmario
attr-from-tree
revwalk-filter
trailer-info
trailer-list
bug
status_case
clay-test
cmn
atexit-skeleton
auth-retry
bump-pretend-git
cancellation
commit-on
commit-to-memory
commit-with-signature
config-header-common
config-regex-is-normalised

master h h
gh-pages

master

ethom...

Merge pull request #4608 from ... yesterday
generated docs
openssl: remove leftover #ifdef
Merge pull request #4640 from mkeeler/...
Merge pull request #4633 fro... 5 days ago
worktree: add functions to get ... 6 days ago
tests: fix issue with /tmp paths on macOS...
Merge pull request #4173 from tlennou/...
Merge pull request #4525 from piks-u/piks/...
Merge pull request #4580 from piks-u/piks/...
Merge pull request #4636 fro... a week ago
Merge pull request #4635 from tlennou/fl...
Merge pull request #4577 from csware/re...
tests: free the worktree in add_with_expli...
retspec: check for valid parameters in git...
remote: repo is optional here
worktree: fix calloc of the wrong object ty...
local: fix a leaking reference when iteratin...
Fix deletion of unrelated branch on workt...
Merge pull request #4631 from andreash...
fixed stack smashing due to wrong size of...
Index: commit the changes to the index p...
Index: test dirty index bit
Index: add a dirty bit reflecting unsaved c...
stash: use _an_index not _the_index
checkout: always set the index in checkou...
Index:reuc tests: test that checkout succe...
Index:reuc tests: add config-regex-is-n...

commit: 934486

index: commit the changes to the index properly

Now that the index has a "dirty" state, where it has changes that have

Edward Thomson authored 11/16/2017 @ 5:19 PM parent: 51c2aa

10 modified

Name ↑ Path Tree View all files

- src/checkout.c
- src/index.c
- src/index.h
- src/status.c
- tests/checkout/tree.c
- tests/checkout/typechange.c
- tests/cherrypick/workdir.c
- tests/index/names.c
- tests/index/reuc.c
- tests/rebase/submodule.c

90% Feedback PRO



Ferramentas Gráficas Git

→ **GitEye**

<https://www.collab.net/products/giteye>

→ **Smartgit**

<https://www.syntevo.com/smartgit/>

→ **Tig**

<https://jonas.github.io/tig/>

→ **Gitcola**

<http://git-cola.github.io/>

→ **Mais Opções**

<https://git-scm.com/downloads/guis/>

Hand-On

Git





0. Instalação do Git

O **Git** é **multiplataforma** e há uma instalação específica para cada SO

→ **Página oficial para instalação**

`https://git-scm.com/downloads`

→ **No meu caso, uso ArchLinux**

```
$ sudo pacman -S git
```

→ **Verificando instalação**

```
$ git --version
```

→ **Comandos de apoio**

```
$ git help
```




1. Configuração

A **primeira** coisa que devemos nos preocupar é a configuração

→ **Identidade**

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

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

→ **Editor**

```
$ git config --global core.editor <nome-editor>
```

→ **Mais comandos**

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

```
$ git help config
```



2. Criando um Repositório

Basicamente há duas formas de **criar um repositório, local ou remoto**

→ Local

- ◆ Criar uma pasta com nome do projeto e acessá-la

```
$ git init
```

→ Remoto

- ◆ Git em máquina ou servidor proprietário
- ◆ Utilizar alguma plataforma de Git on-line



3. Git ignore

É importante informar ao Git quais **arquivos você não deseja acompanhar**.

→ **Criar o arquivo *.gitignore***

◆ *.class

◆ public/

◆ .env

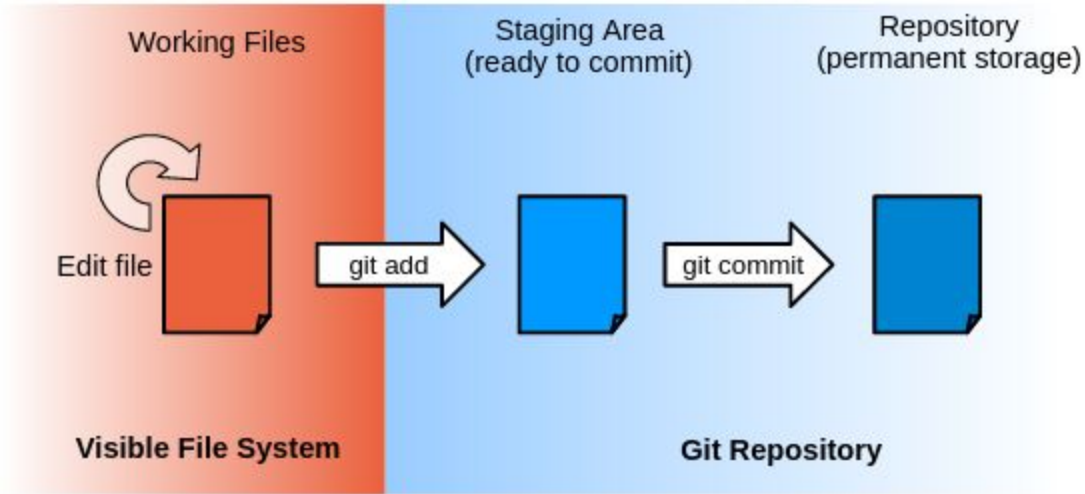
◆ *.log

→ **Mais detalhes**

<https://help.github.com/en/articles/ignoring-files>

—
Mas, antes de continuar,
vamos **entender**
como funciona o **git!**

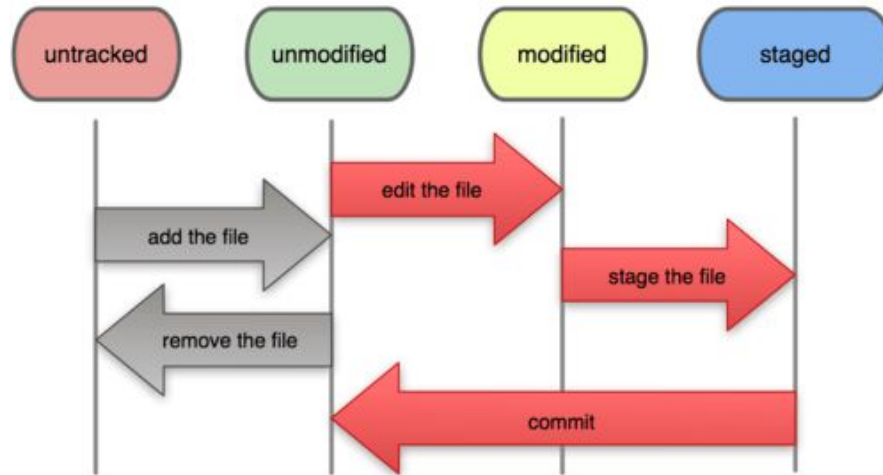
5. Workflow



<https://jdblischak.github.io/2014-09-18-chicago/novice/git/01-backup.html>

6. Ciclo de vida

File Status Lifecycle



<https://git-scm.com/book/pt-br/v1/Git-Essencial-Gravando-Alterações-no-Repositoryo>



6. Manipulando arquivos

→ **Exibir o status do espaço de trabalho**

```
$ git status
```

→ **Adicionar arquivos a Staging Area**

```
$ git add <nome-do-arquivo>
```

```
$ git add <arquivo-1 arquivo-2 arquivo-3>
```

```
$ git add < *.formato>
```

```
$ git add <diretorio/>
```

```
$ git add .
```



6. Manipulando arquivos

→ **Mover/Renomear arquivos do repositório**

```
$ git mv <nomeArquivo diretorioDeDestino/>
```

```
$ git mv <nomeArquivo novoNome/>
```

→ **Remover arquivos do repositório**

```
$ git rm <nomeArquivo>
```

```
$ git rm <diretorio/>
```




6. Manipulando arquivos

→ Verificando suas alterações

```
$ git diff
```

```
$ git diff <arquivo>
```

```
$ git diff <arquivo> <branch>
```

```
$ git diff HEAD <arquivo>
```

→ Efetivando as modificações

```
$ git commit
```

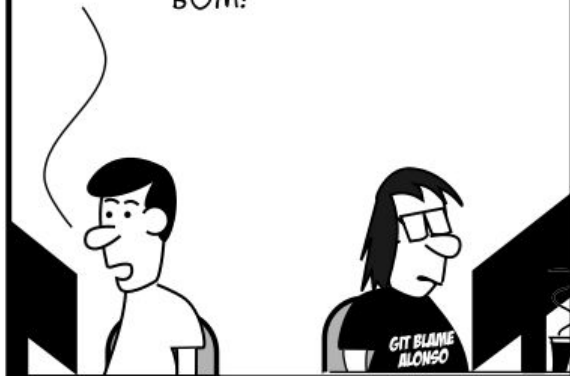
```
$ git commit -m "mensagem do commit"
```

ALONSO, VOCÊ SUBIU A ALTERAÇÃO COM O TEXTO DO COMMIT "FEITO CONFORME SOLICITADO"! ISSO É MUITO VAGO, PRECISA MELHORAR.

AH, TÁ. PODE DEIXAR QUE EU JÁ ARRUMO...

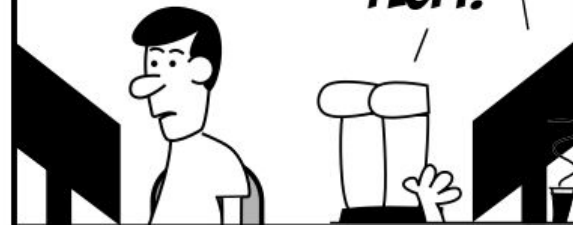


PRONTO, OLHA LÁ QUE ACHO QUE AGORA FICOU BOM!



Realizadas todas as alterações conforme elas foram solicitadas no código fonte do projeto atual

PLOFT!





7. Histórico

→ **Verificando alterações do repositório**

```
$ git log
```

```
$ git log --pretty=format:"%h - %an, %ar : %s"
```

```
$ git log -p
```



8. Desfazendo Alterações

→ **Desfazer um Arquivo Modificado:**

```
$ git checkout -- <file>
```

```
$ git checkout HEAD~1
```

```
$ git checkout <sha1>
```

→ **Desfazer alteração local**

```
$ git reset HEAD <file>
```

```
$ git reset [--hard] HEAD~1
```



8. Desfazendo Alterações

→ **Desfaz todos commits até o commit indicado**

```
$ git revert <sha1>
```

→ **Corrigir o último commit**

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



9. Repositório Remoto

→ **Enviando modificações para servidor remoto**

```
$ git push <repositorio> <branch>
```

→ **Coletando o estado atual do servidor remoto**

```
$ git pull <repositorio> <branch>
```



9. Repositório Remoto

→ **Conectando repositório**

```
$ git remote add origin <url-repositorio>
```

→ **Clone**

```
$ git clone <url-repositorio>
```

VIDA DE PROGRAMADOR

.COM.BR

```
real historia;  
string sender;  
sender = "estagiario";
```

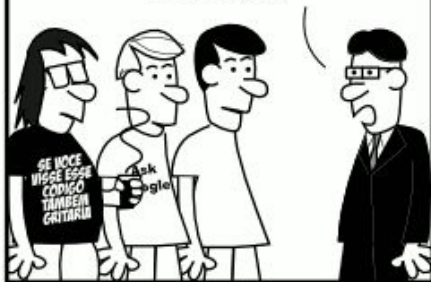


#891

ALGUÉM AÍ ESTÁ
MEXENDO NO ARQUIVO
FOO.BAR?



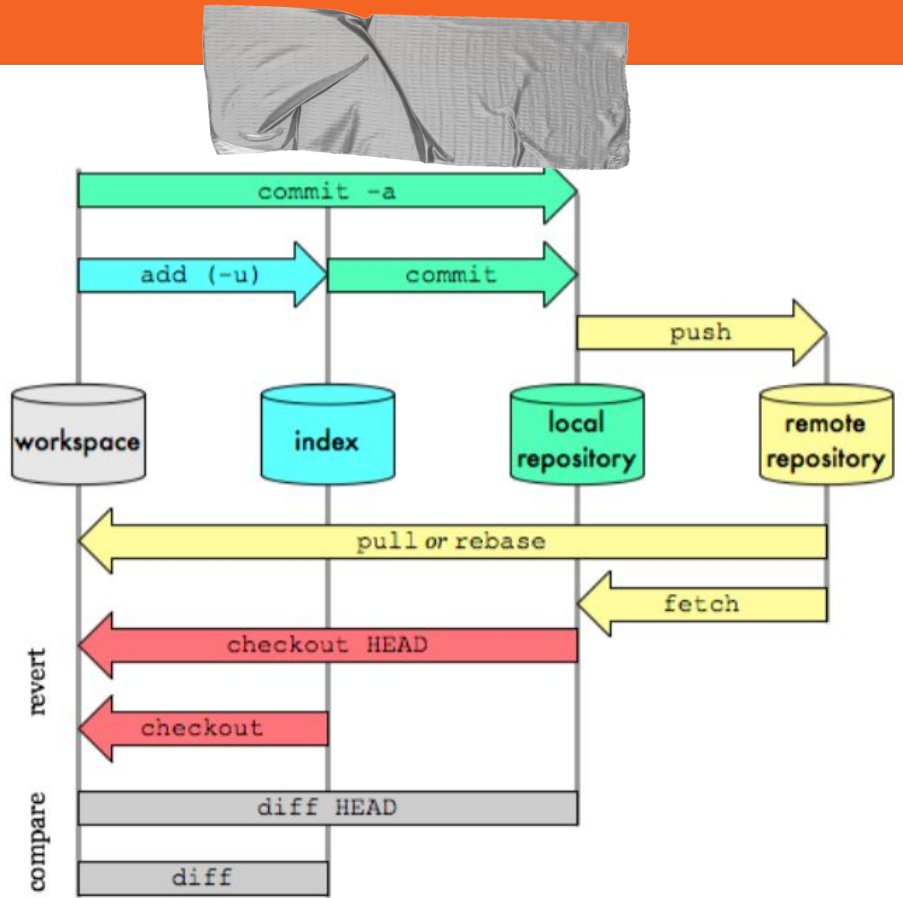
VAMOS IMPLANTAR UM SISTEMA
DE CONTROLE DE VERSÕES,
PARA ACABAR COM O CONTROLE
DE GRITOS



DIAS DEPOIS...

GALERA, GUENTA A
MÃO AÍ QUE VOU
DAR UM PUSH!





<https://hackernoon.com/top-5-free-courses-to-learn-git-and-github-best-of-lot-2f394c6533b0>

Photo by [Paul Gilmore](#) on [Unsplash](#)



Git Flow



9. Branch

→ **Criando uma nova branch**

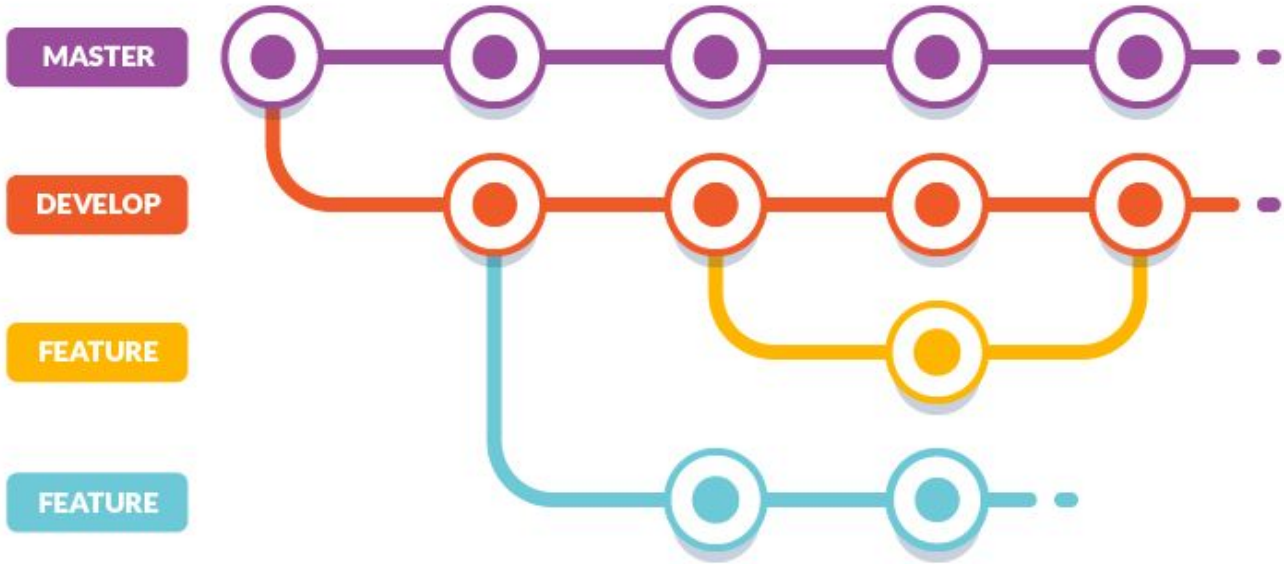
```
$ git branch <nome-branch>
```

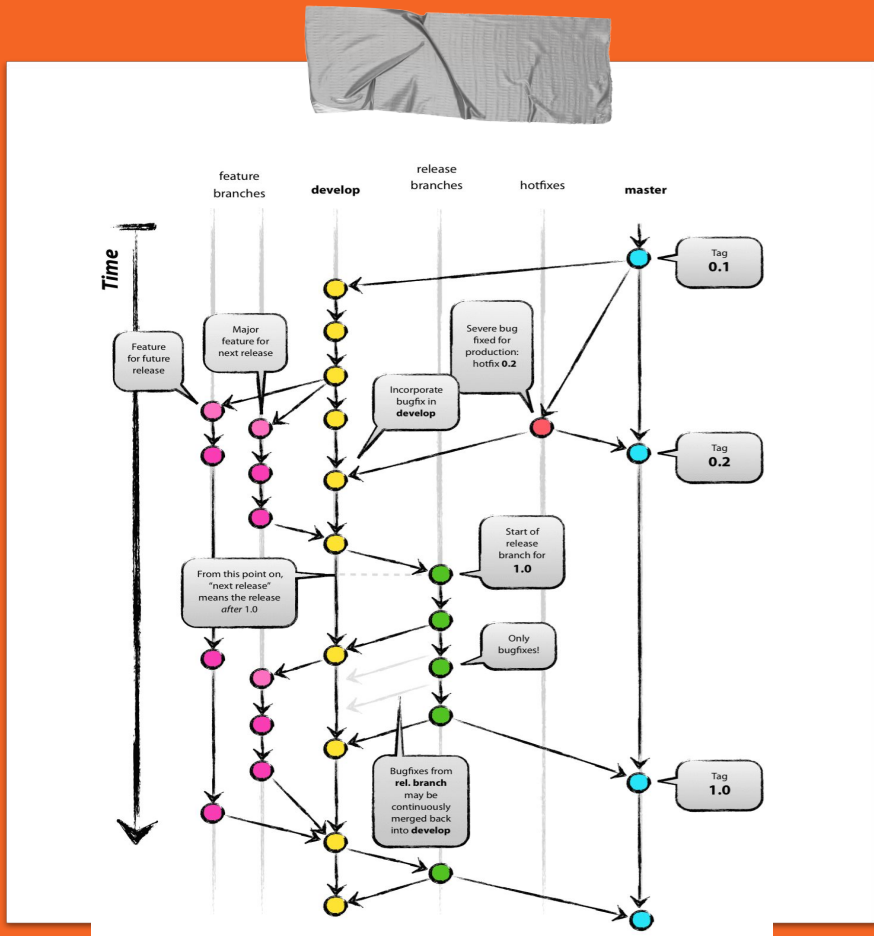
→ **Mudando de branch**

```
$ git checkout <nome-branch>
```

→ **Ou os dois de uma vez**

```
$ git checkout -b <nome-branch>
```



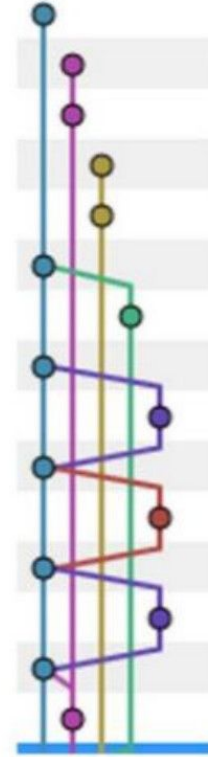




QUANDO VOCÊ
PERCEBE QUE DÁ
PRA FAZER UM
SOLO DE GUITARRA
COM O SEU
REPOSITÓRIO...

VIDA DE
PROGRAMADOR
COMER

(IMAGEM REAL DE REPOSITÓRIO
ENVIADA POR DAGNELSON)





9. Branch

→ **Listar branches disponíveis**

```
$ git branch
```

→ **Atualizar uma branch**

```
$ git fetch origin <branch-origem>
```

→ **Visualizar histórico para todas branches**

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

```
$ git log --graph --pretty=format:'%Cred%h%Creset  
-%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset'  
--abbrev-commit --date=relative
```

—
Depois de tantas

branches abertas e

pessoas trabalhando em

paralelo, **o que fazer?**



GIT MERGE





9. Merge

- **Para unir as modificações de uma branch para outra**

```
$ git checkout <branch-a-receber-alterações>
```

```
$ git merge <branch-das-alteracoes>
```

- **Apagar uma branch**

```
$ git branch -d <branch>
```

- **Gerenciar conflitos de merge**

```
$ git mergetool
```

- **Fork & Pull Request**



10. Tag

→ **Listar tags / Visualizar detalhes de uma tag**

```
$ git tag
```

```
$ git show v2.19.0
```

→ **Mudar status do repositório para uma tag**

```
$ git checkout v2.19.0
```

```
$ git checkout -b feature-bugfix v2.19.0
```

→ **Criar uma tag com status atual do repositório**

```
$ git tag -a v2.19.1 -m "Version v2.19.1, bugfix"
```



11. Encerramento

- Git é uma ferramenta **poderosíssima**
- Grande aliada para **gerenciar projetos** de **todos portes**
- **Diversos recursos e comandos** disponíveis
- Possui uma comunidade ativa e bastante madura
- **Diversos materiais/cursos** disponíveis
- **Um mundo de grandes possibilidades**



Links Importantes

- <https://git-scm.com>
- <https://git.wiki.kernel.org>
- <https://guides.github.com>
- <https://github.com/git/git>
- <https://www.gharchive.org/>



In case of fire



1. `git commit`



2. `git push`



3. leave building



Obrigado!



 [linkedin.com/in/icarojerry/](https://www.linkedin.com/in/icarojerry/)

 github.com/IcaroJerry

 icaro.jerry@gmail.com