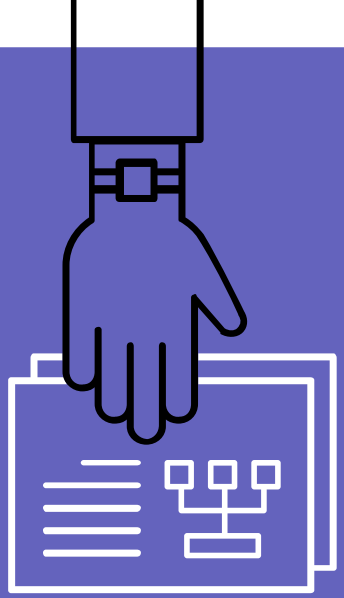


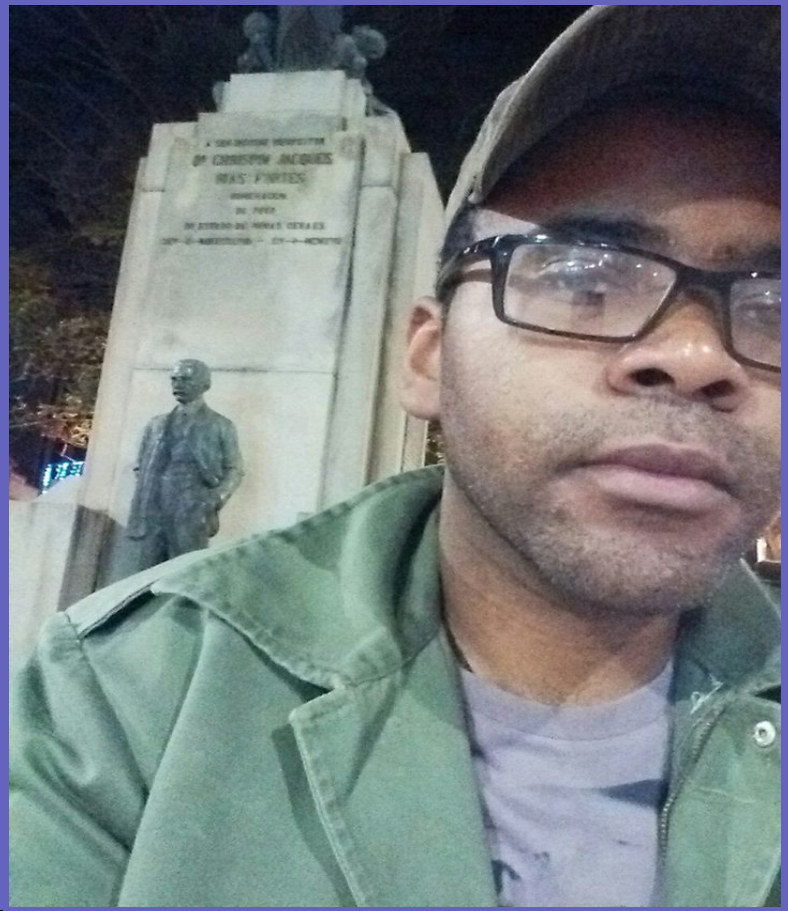
Github

Wagner Rodrigo da Silva

Curso: Engenharia de Software
Periodo: 7º

GitHub:

https://github.com/wagnerrodrigo

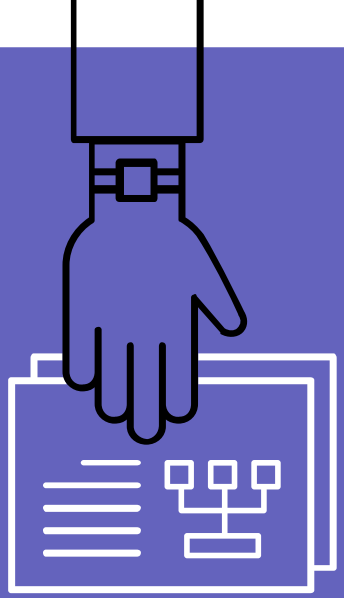


Mas o que é

Github

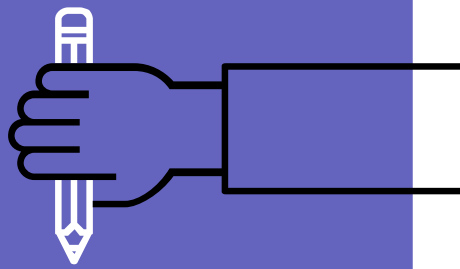
Que utiliza o GIT

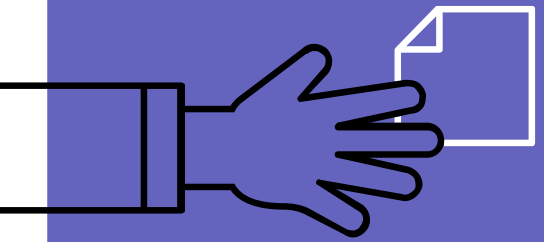
Então o que é o GIT



Controle de Versão

Mas o que é Controle de versão?

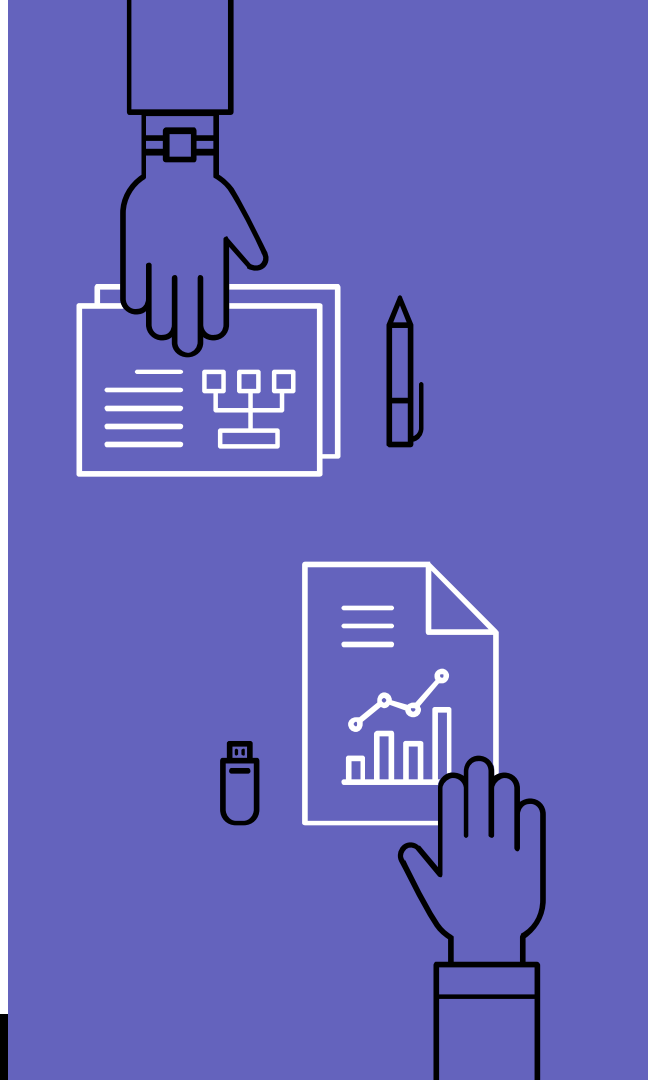




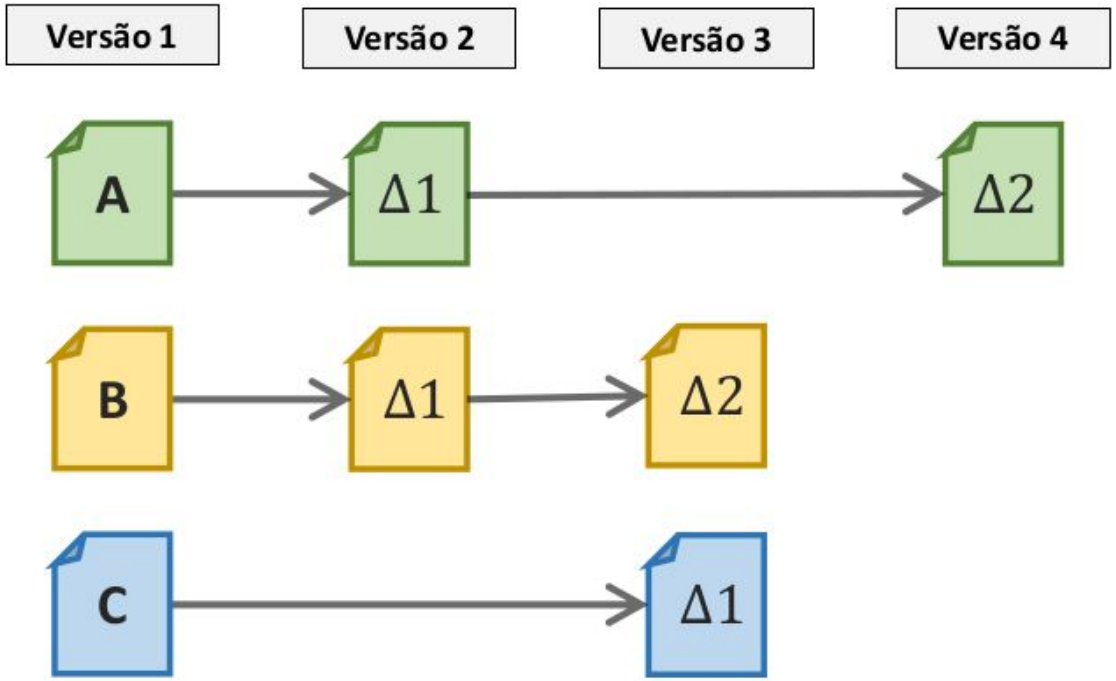
É um

- sistema que gerencia as várias etapas de um projeto à medida que seu desenvolvimento vai evoluindo, possibilitando observar (o versionamento) e retornar para alguma versão passada possibilitando uma correção para o projeto atual.

  - VCS (version control system)
  - SCM (source code management)

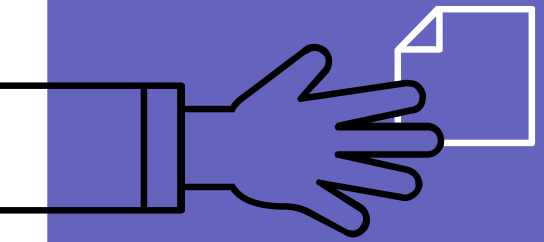


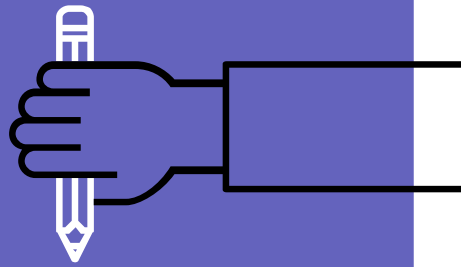
Sistema de Controle Convencional



Bkp 1, Bkp2,...
Projeto_inicio ....
Email
Whatsapp
Telegram

Tipos de Controle de versão



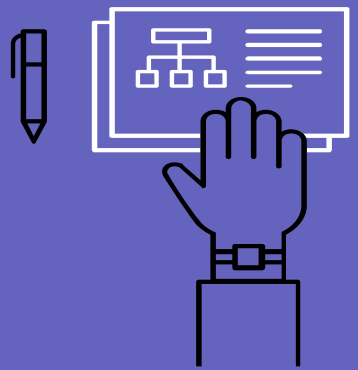


Controle de versão local

Controle de versão Centralizado

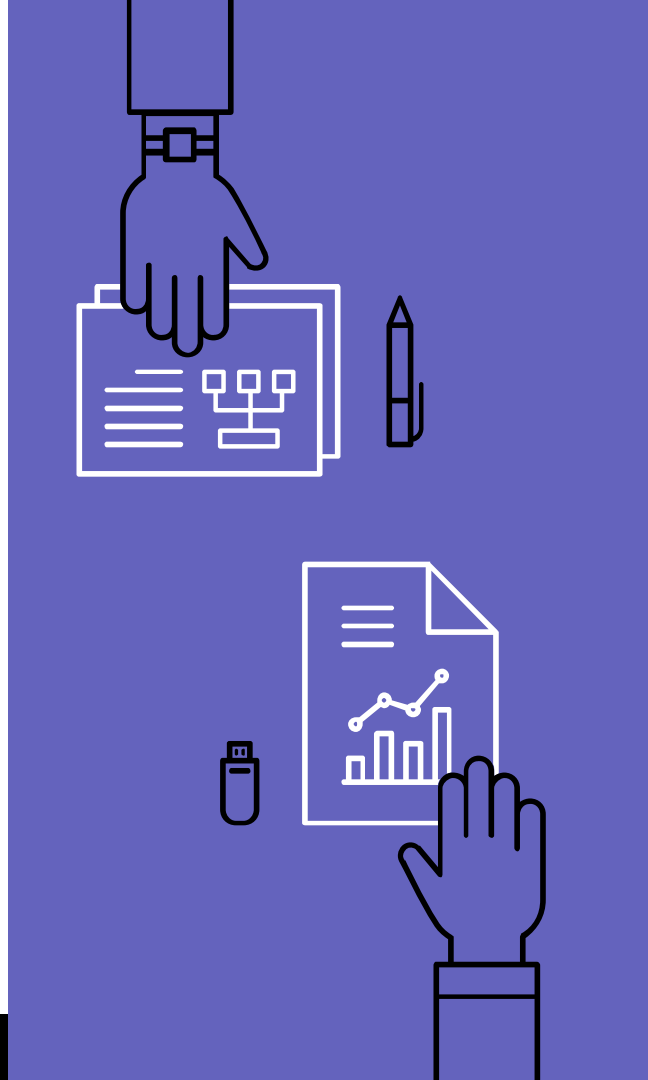
Controle de versão distribuído

Controle de versão local

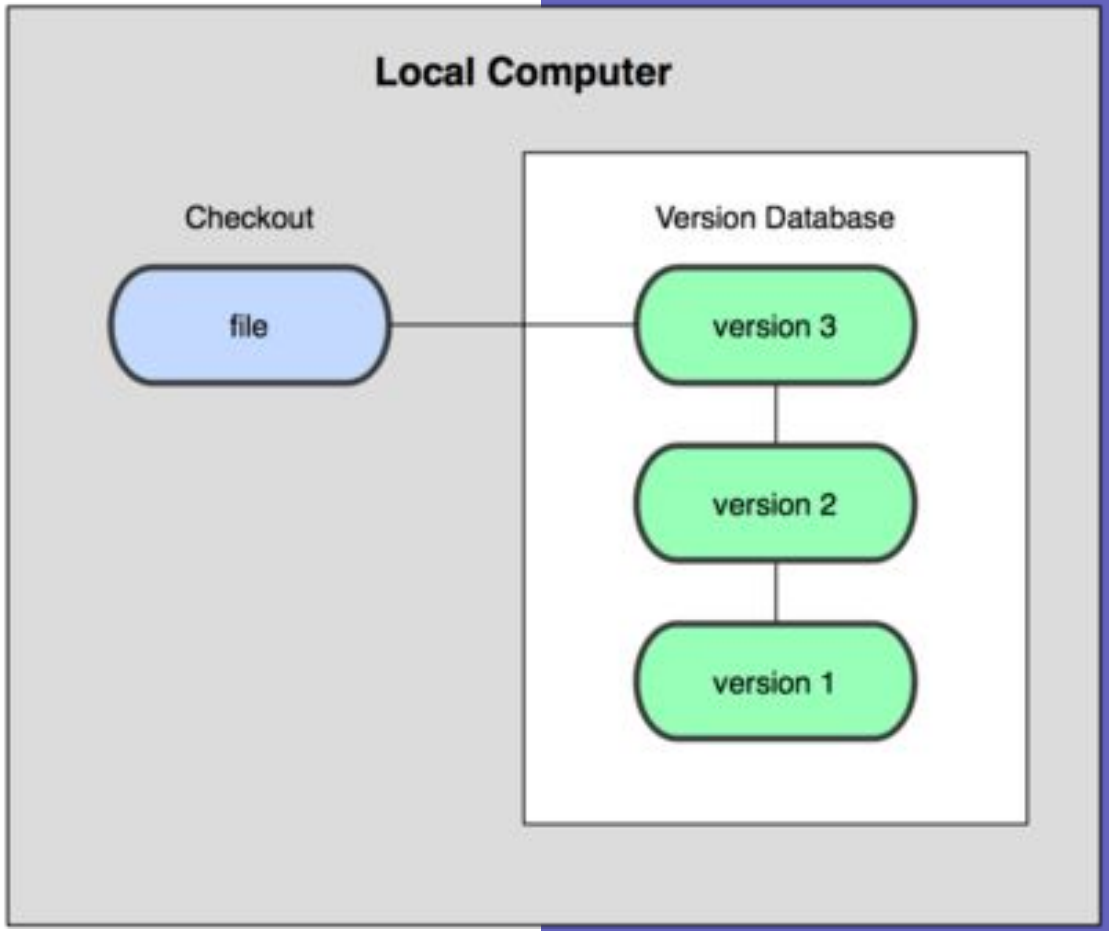


Source Code Control System (SCCS)

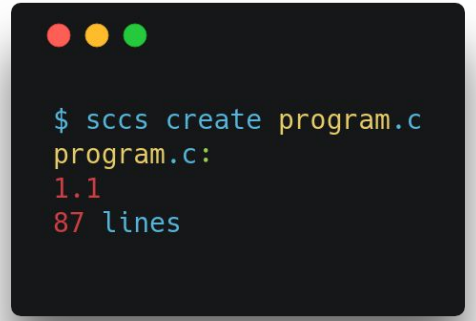
- Salvo somente na máquina de trabalho
- O desenvolvedor tem um histórico do projeto.
- Permite a recuperação de dados da última versão



SCCS


Local Computer
Checkout
file
Version Database
version 3
version 2
version 1

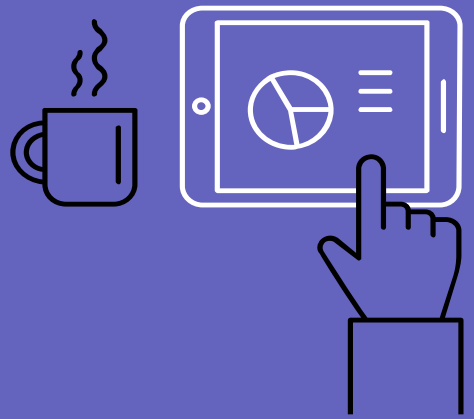
Um pouco de História



- O primeiro controle de versão surgiu em 1972 com o nome de Source Code Control System (SCCS) sendo o sistema mais utilizado para controle de versão até o surgimento do RCS

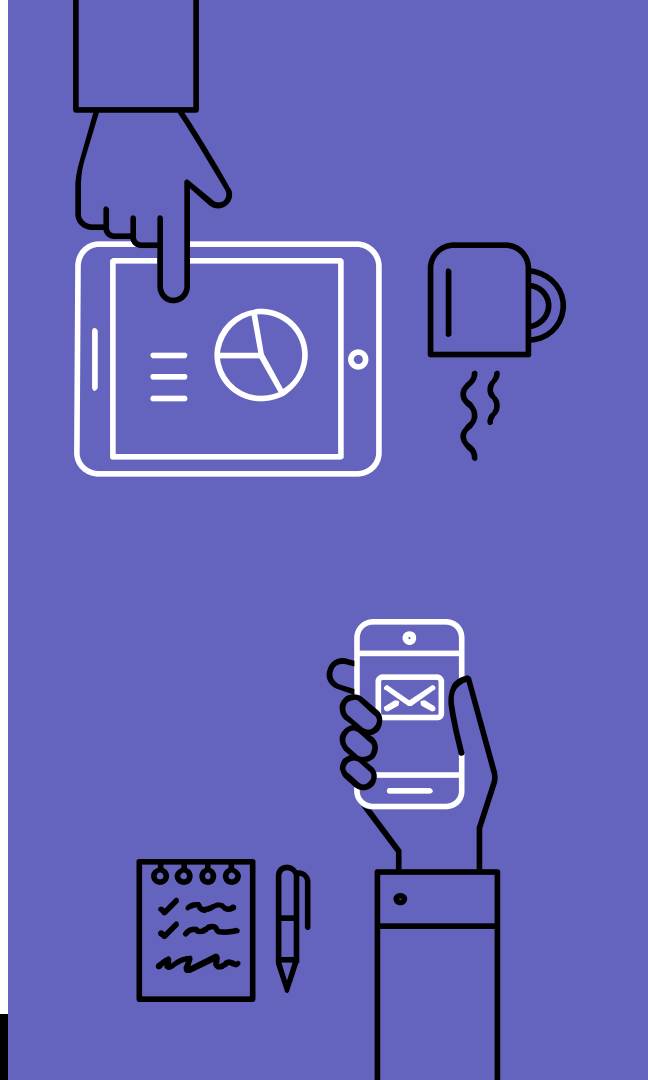
- Revision Control System ou *Sistema de Controle de Revisão (RCS)* foi lançado em 1982 por Walter F.Tichy, usava técnicas novas em relação ao SCCS, de ponto negativo era a capacidade de trabalhar somente com arquivos individuais

Controle de versão Centralizado

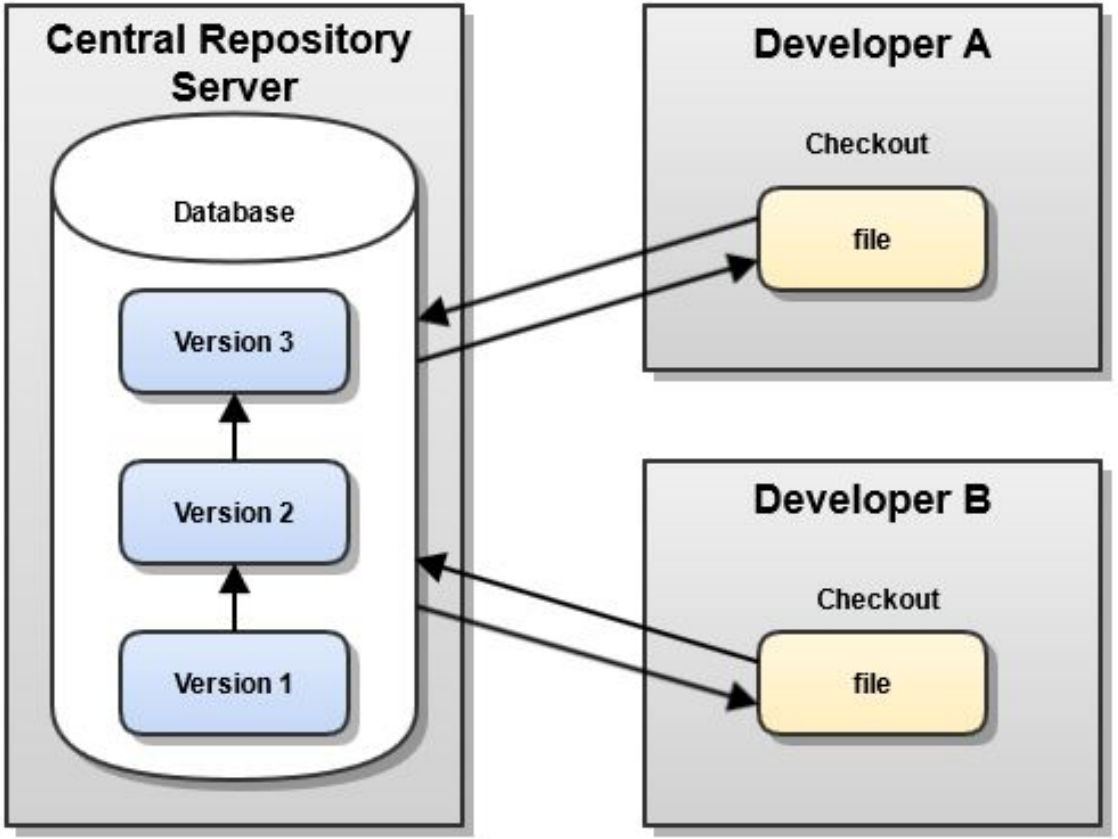


Centralized Version Control System(CVCS) ou CVS

- Modelo cliente-servidor
- Divisão do trabalho em equipes.
- Gerência do projeto pelo administrador do CVCSs(Sistema de Controle de versão Centralizado)



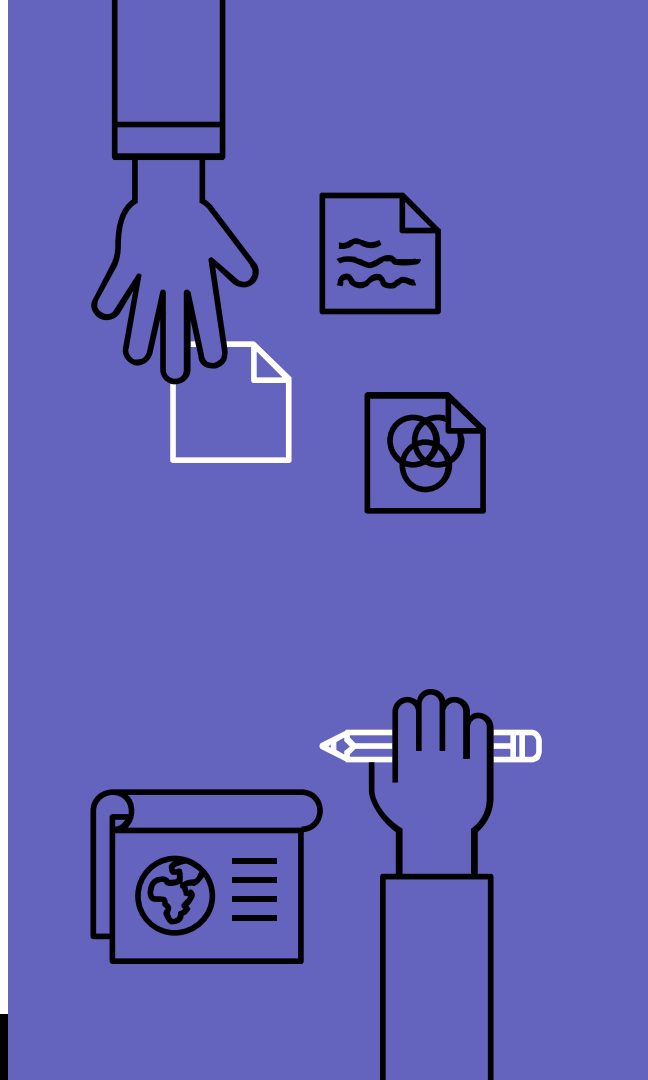
CVCS


Central Repository Server
Database
Version 3
Version 2
Version 1
Developer A
Checkout
file
Developer B
Checkout
file

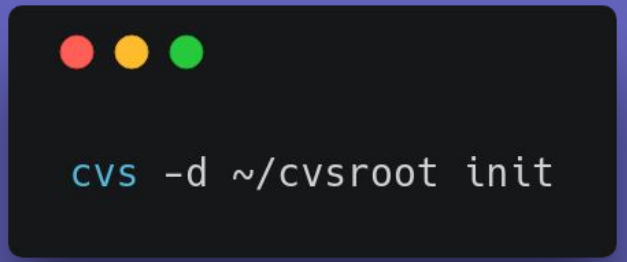
Principais representantes

- CVS (Concurrent Version System)
- **SVN (Apache Subversion)**
- P4 (Perforce)

  Games!!!!

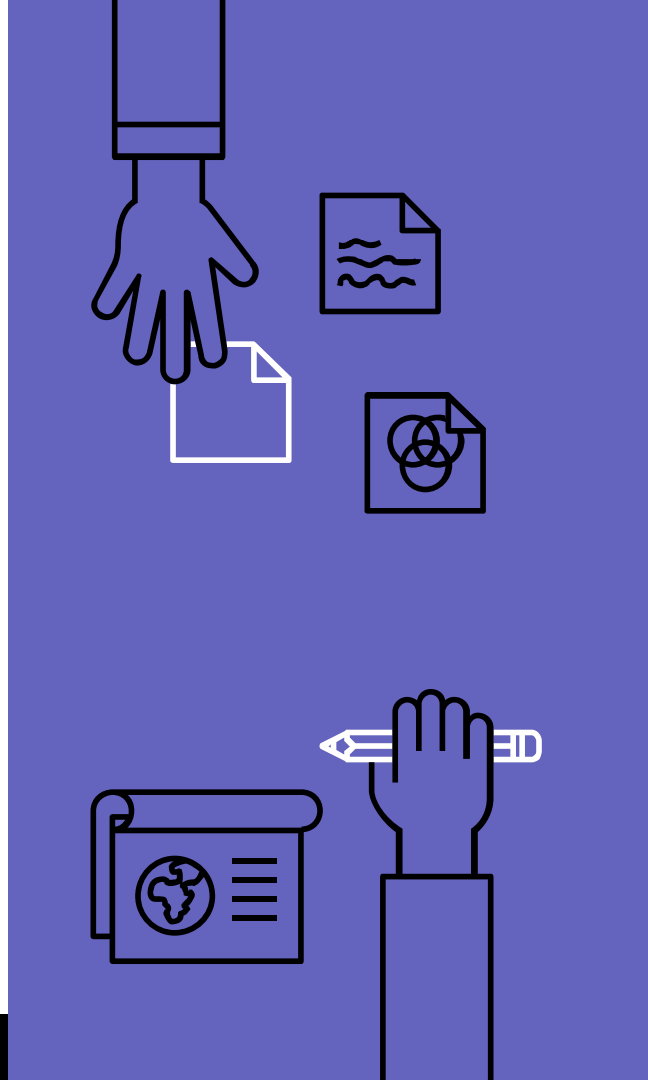


Comandos !!!



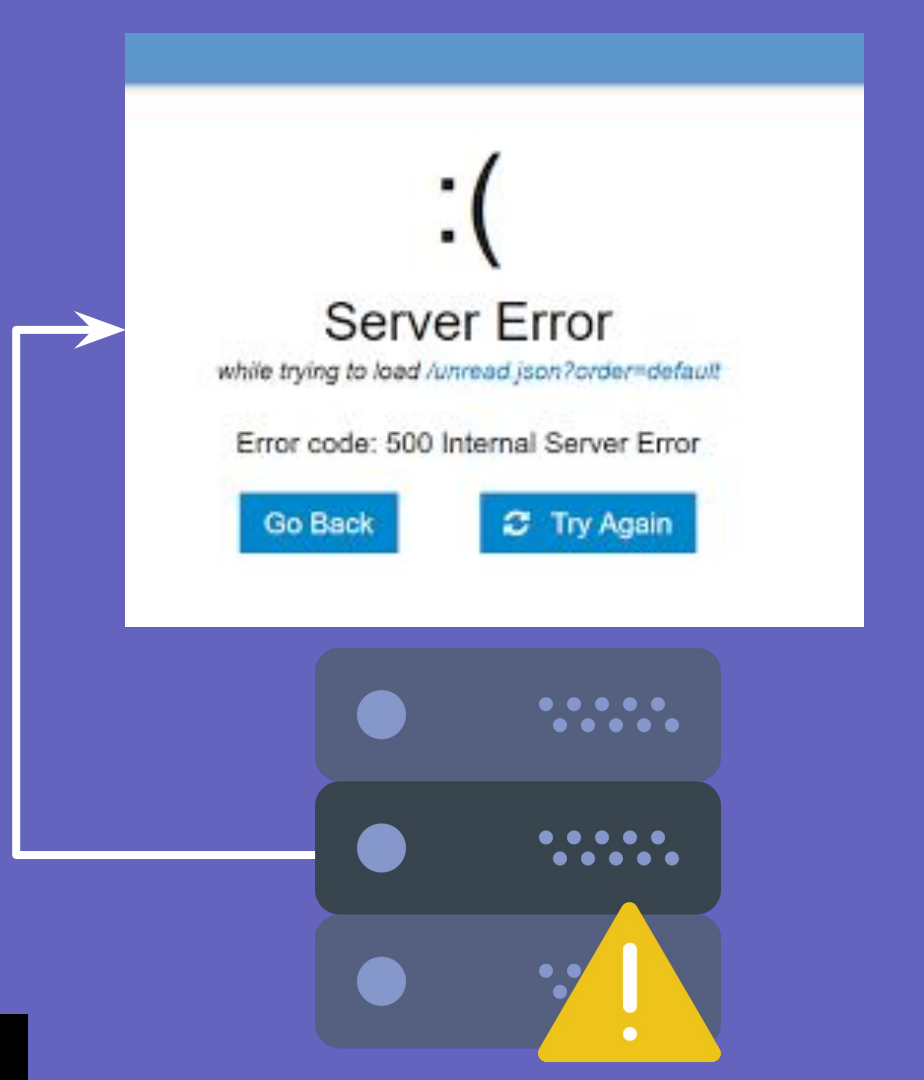
Apache Subversion

- Conhecido como SVN muito popular entre empresas de desenvolvimento "fábricas de software" o SVN é de código aberto criado em 2000 pela CollabNet e atualmente está sendo mantido pela Apache Software Foundation



Desvantagens

- Somente um repositório para o projeto.
- Disponibilidade do repositório.
- Permissão de acesso ao repositório.
- Defeito de hardware BKP?

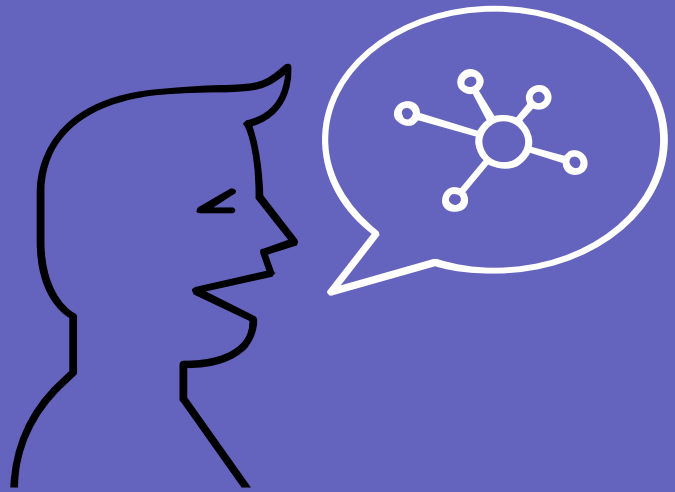


Vantagens

- Controle de acesso.
- Cópia de segurança.
- Controle de qualidade.
- Evolução do código.

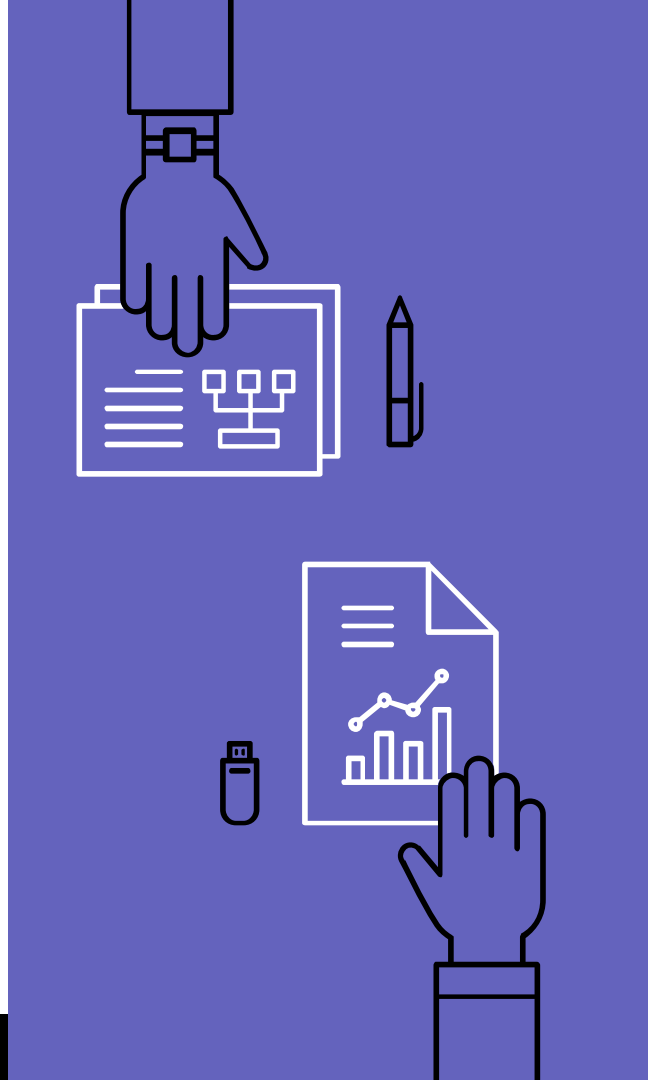
Gerencia de projetos

Controle de versão distribuído

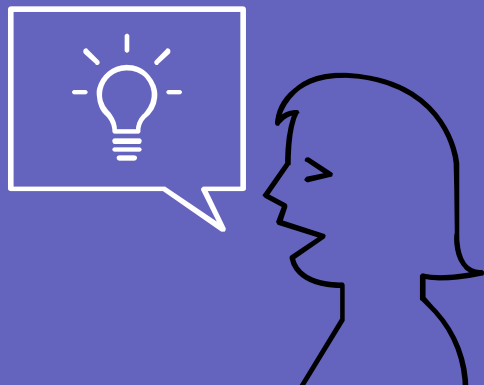


Distributed Control Version (DVC) ou (DVCS)

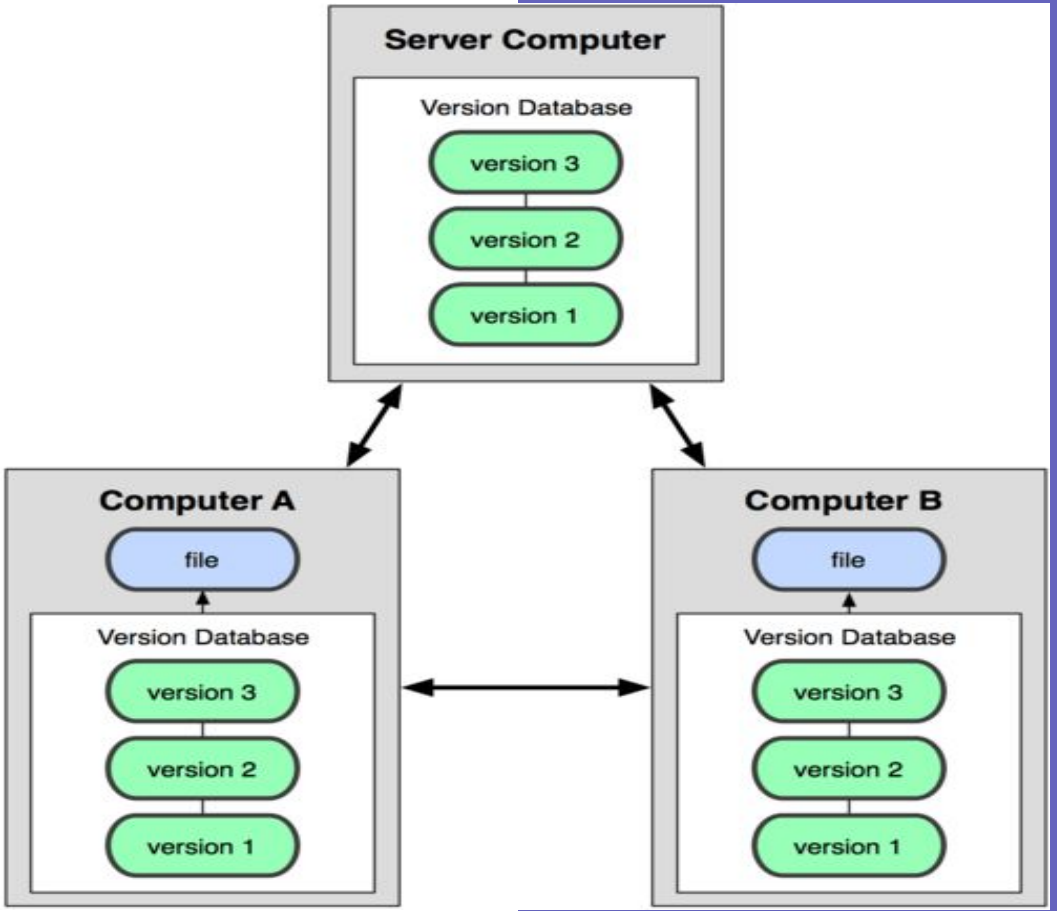
- Todos os membros da equipe tem os repositórios as últimas versões.
- "Não a necessidade de um repositório central"
- O desenvolvedor tem liberdade para trabalhar entre repositórios remotos.



O que significa ser distribuido

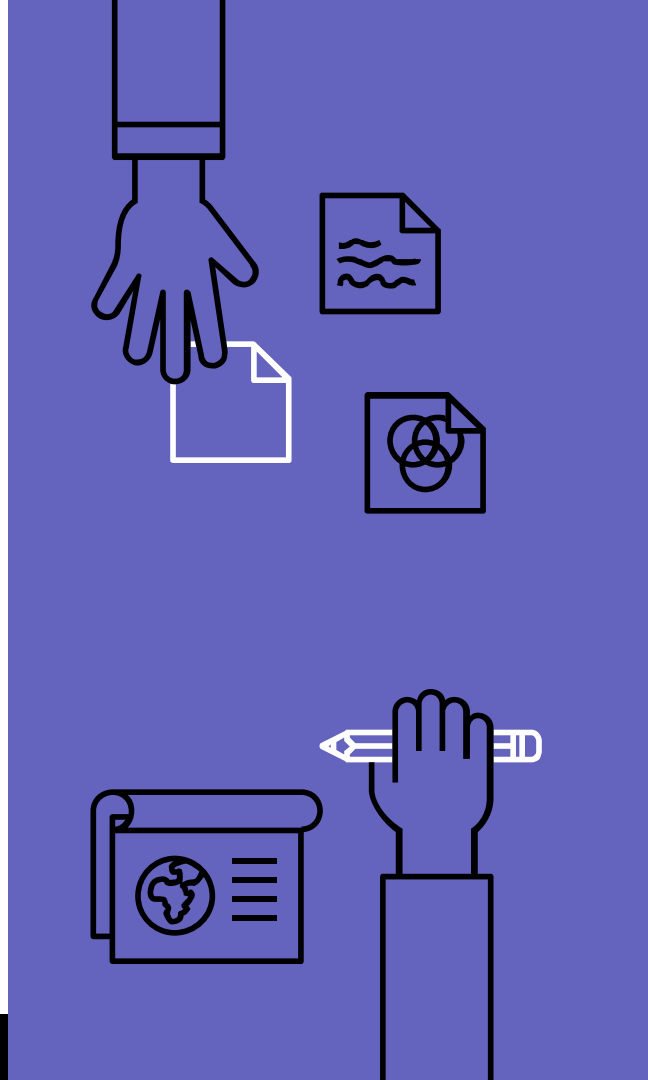


DVCS


Server Computer
Version Database
version 3
version 2
version 1
Computer A
file
Version Database
version 3
version 2
version 1
Computer B
file
Version Database
version 3
version 2
version 1

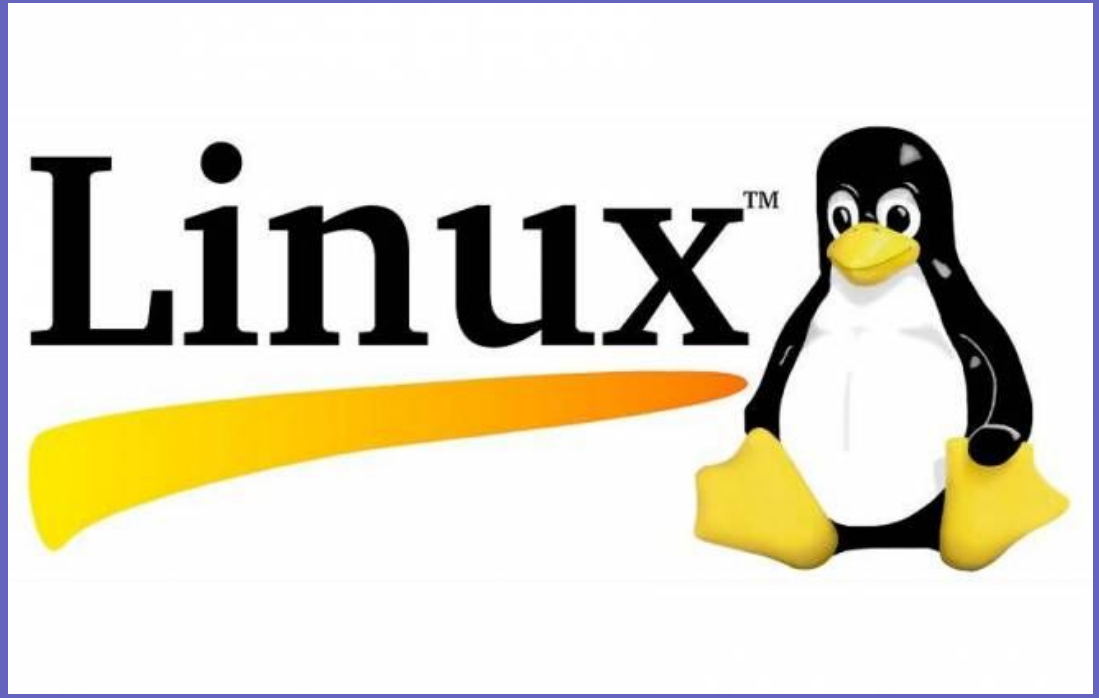
Principais representantes

- **Git**
- Mercurial
- Darcs (Darcs Advanced Revision Control System)



Comandos !!!

Surgimento do Git



Vantagens

- Design simples e muito ágil
- Suporte robusto a desenvolvimento não linear (milhares de branches paralelos)
- Totalmente distribuído capaz de lidar eficientemente com grandes projetos como o kernel do Linux (velocidade e volume de dados)

Desvantagens

- Mais complexo.
- Necessidade que os desenvolvedores conheça o funcionamento do git.
- A equipe deve escolher um fluxo de trabalho (workflow).
- Privacidade e segurança.


git

“

Não é

git

Apache Allura

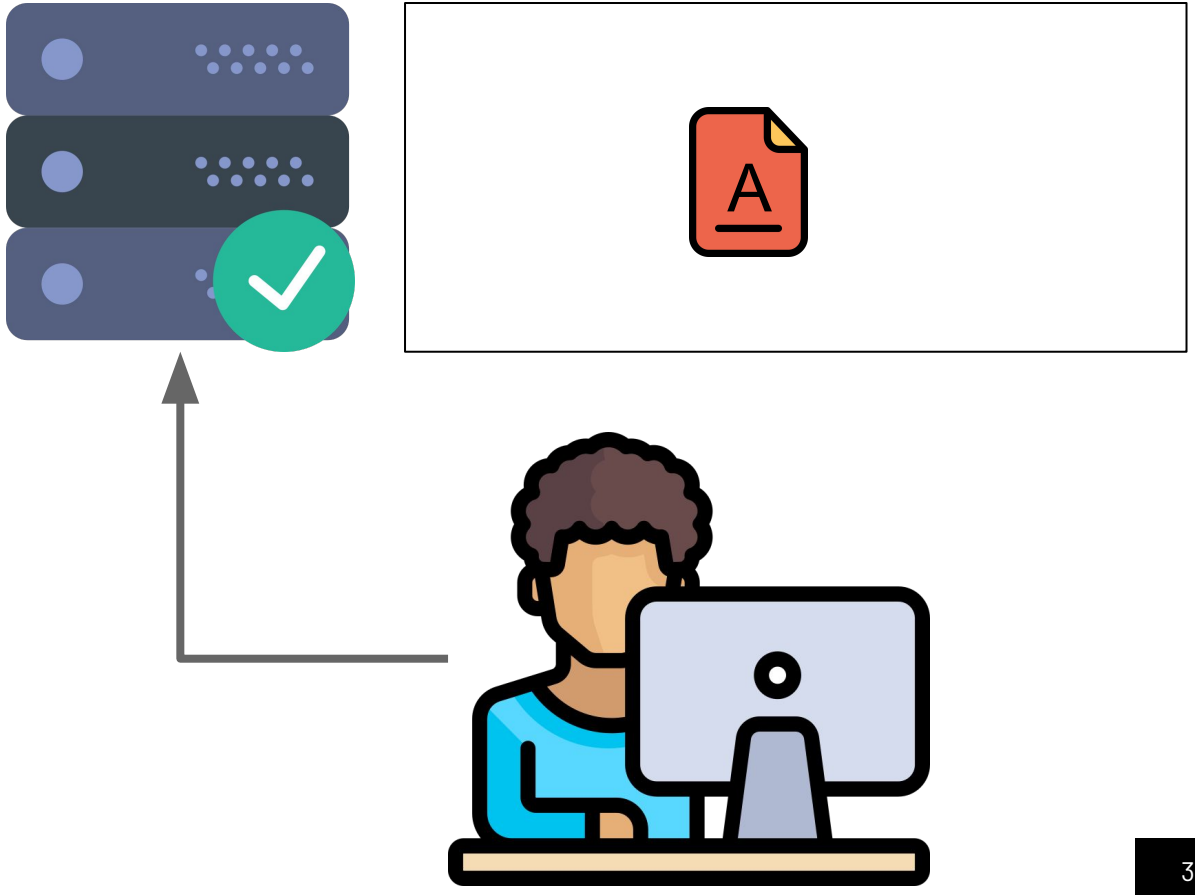
Cloud Source Repositories

Outros serviços de armazenamento de repositórios de códigos fonte

Trabalhando com o GIT localmente

Maria trabalhando
Está trabalhando no código na linha 47

Repositório Local



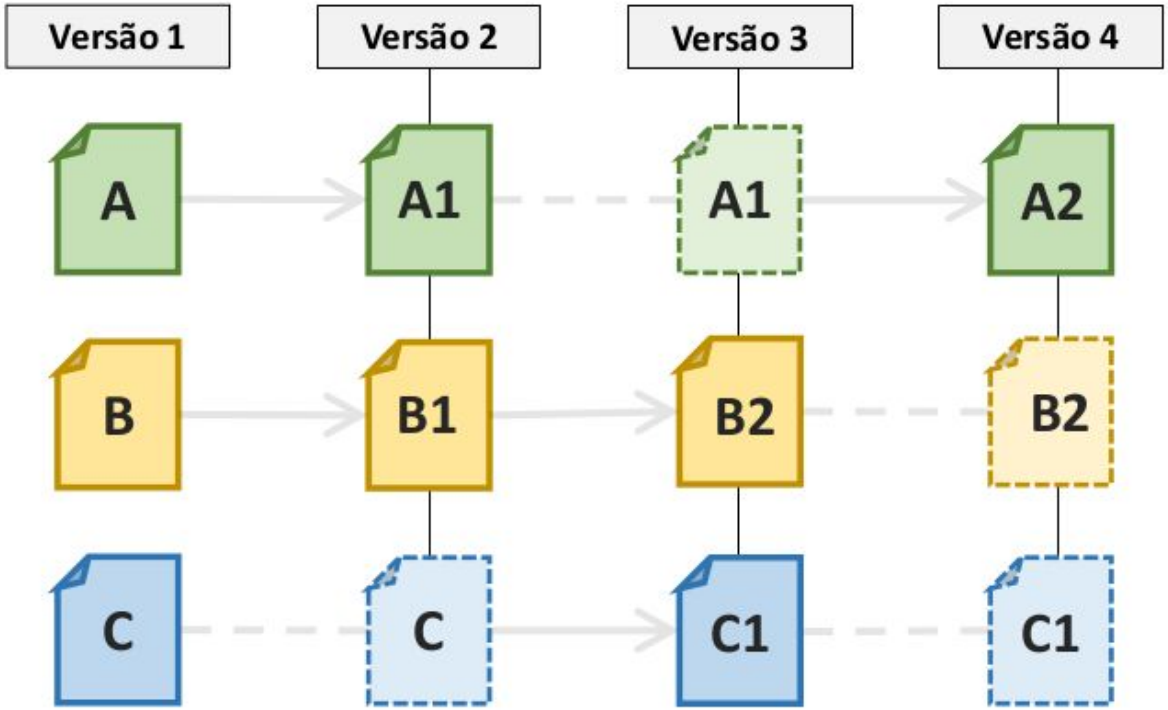
Maria envia o código para o Servidor local no qual é a sua própria máquina


Linha 101
A


Linha 25
A

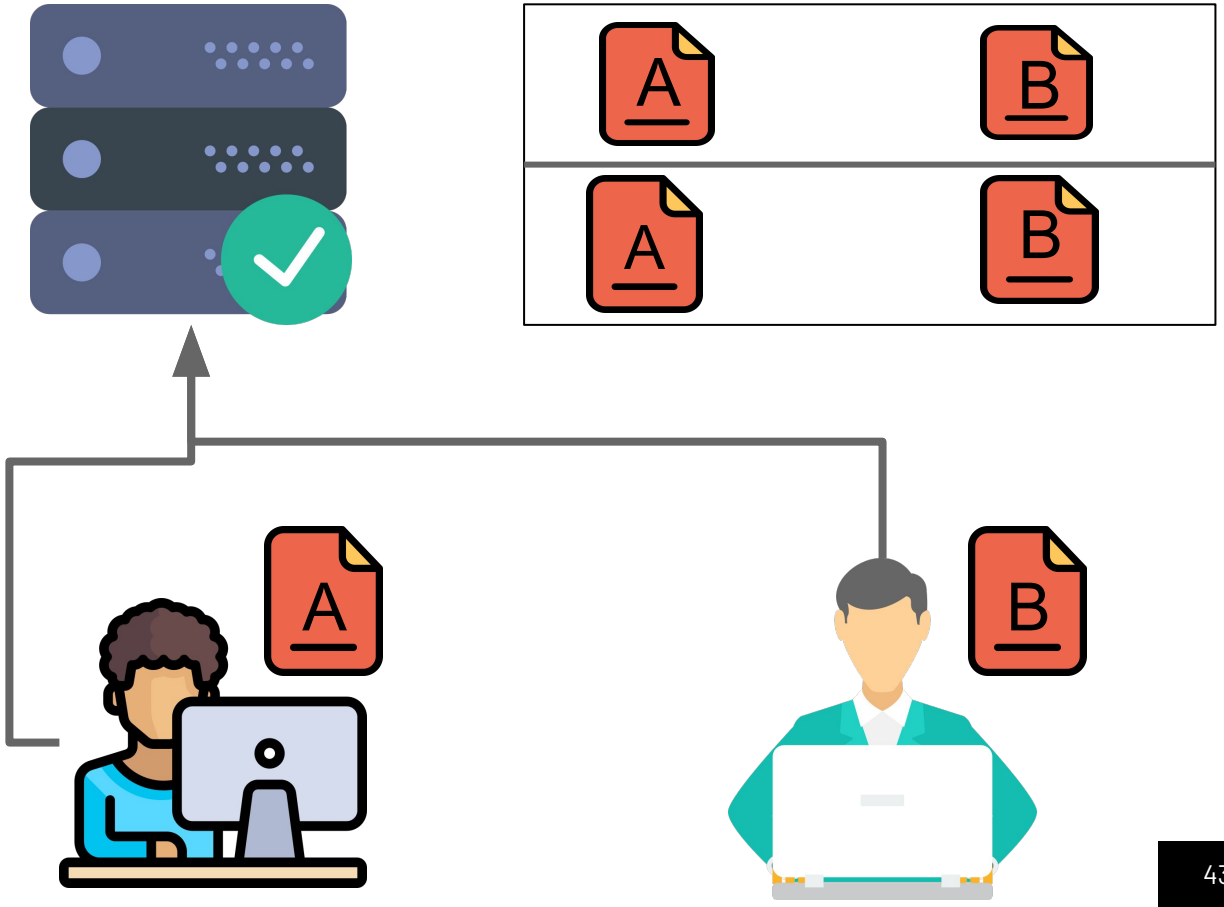
Assim por diante no final do dia de trabalho ou então em tempos regulares envia para o servidor local

Sistema de Controle do Git



Trabalhando com o GIT distribuído

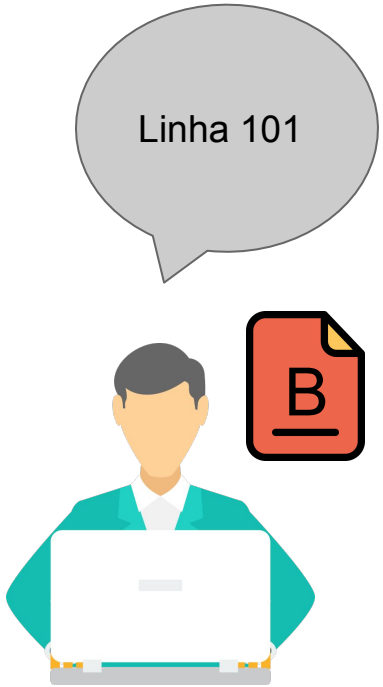
Repositório Remoto



<- Ultima verão

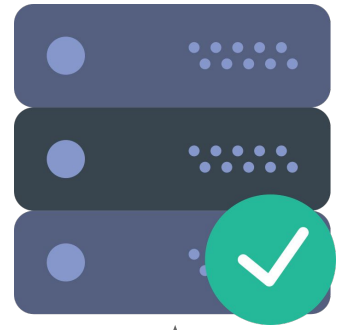
<- Historico

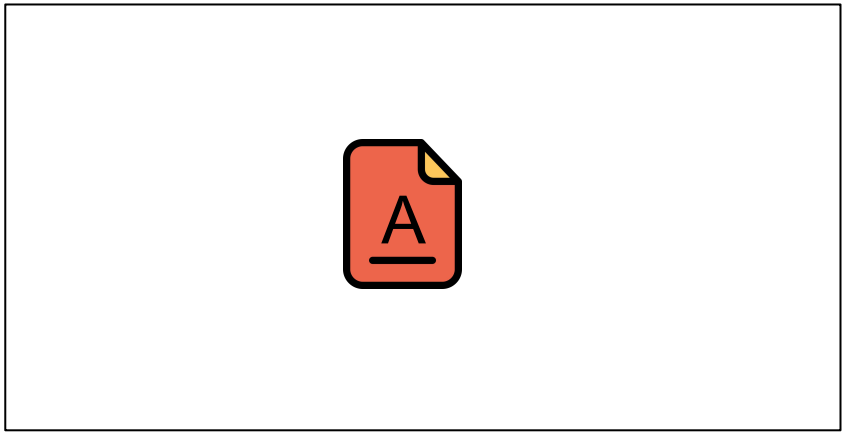
<- Desenvolvedores

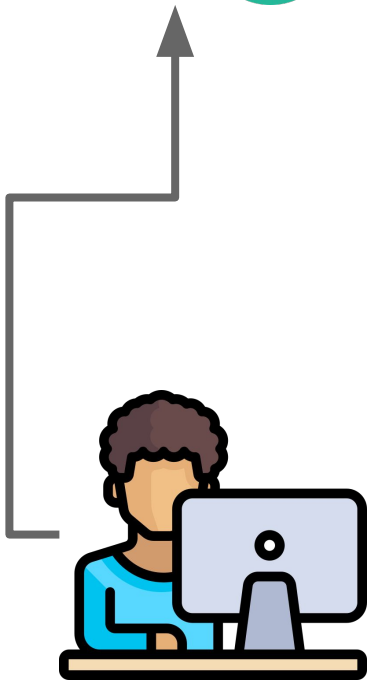


Maria trabalhando na linha 47

E José trabalhando na linha 101

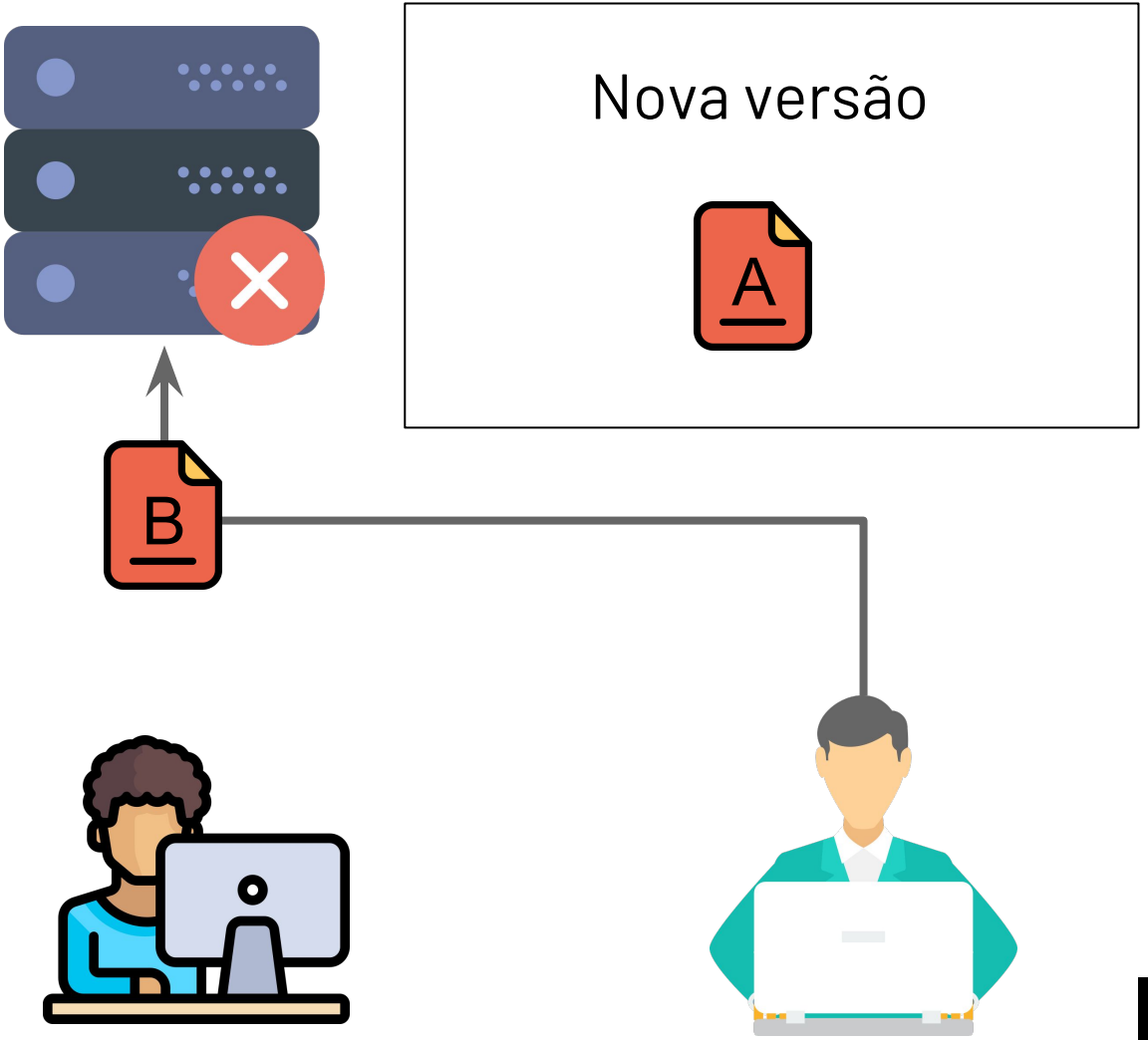




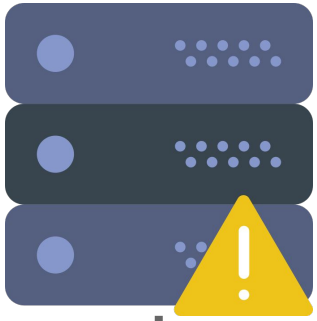


Maria envia o codigo para o servidor

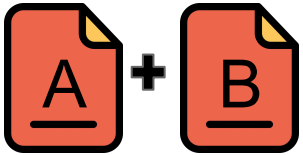
José ainda está trabalhando



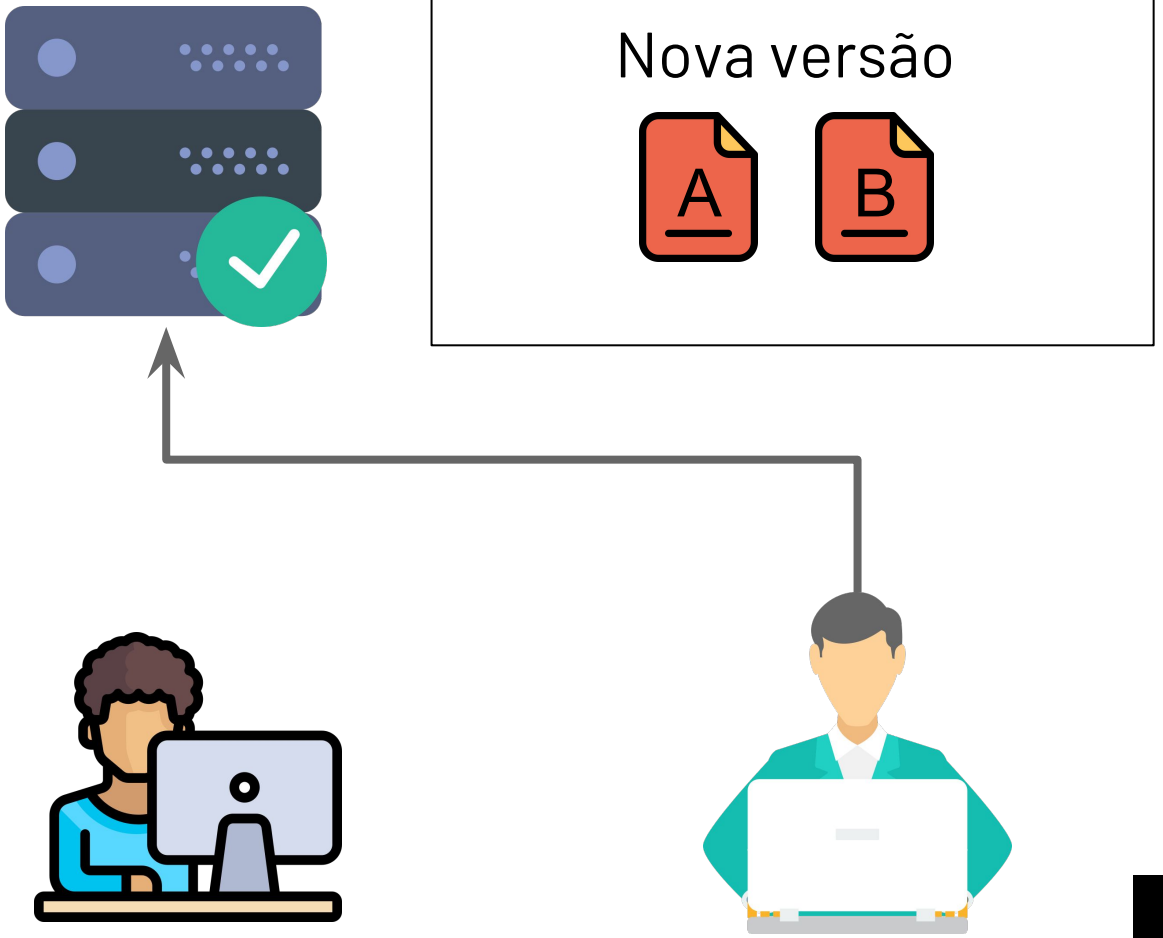
José envia seu código para o servidor e é alertado que existe uma versão nova no servidor



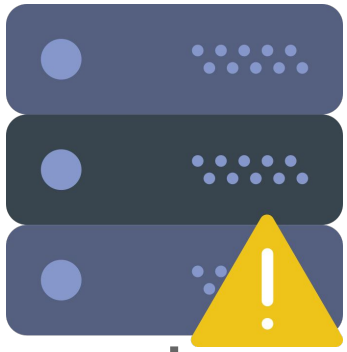
Nova versão



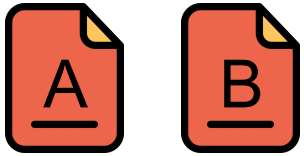
José baixa a versão mais nova do servidor e atualizar seu repositório local

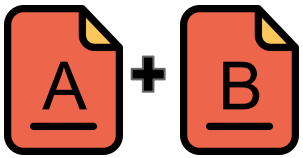


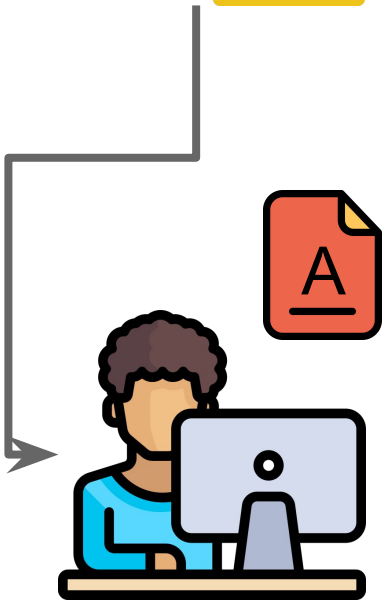
José envia novamente o seu código para o servidor



Nova versão

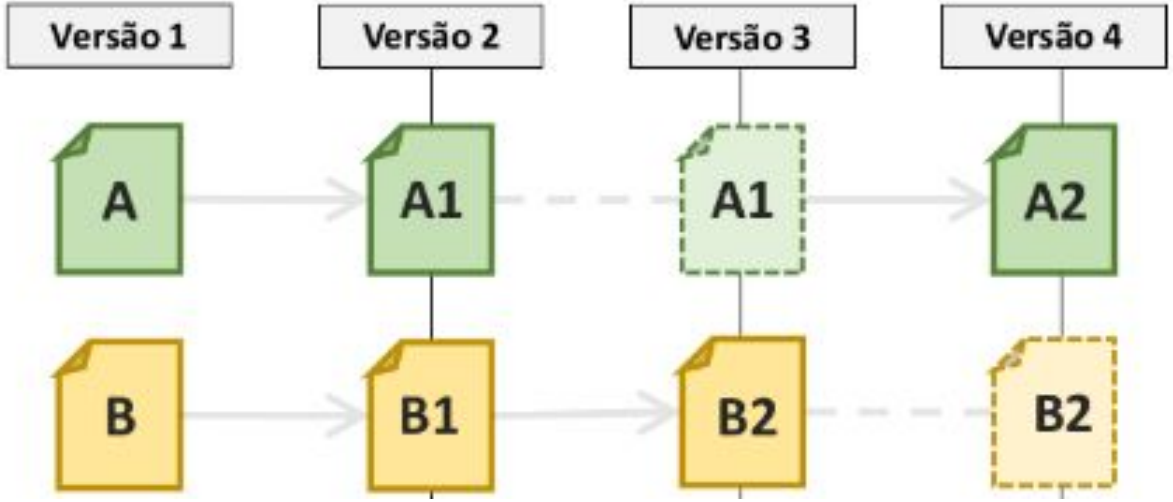


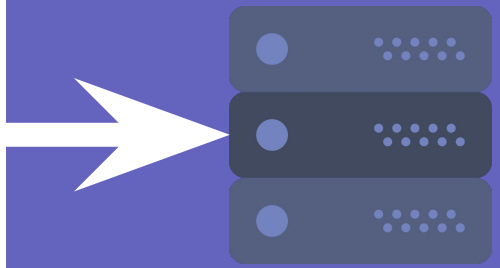




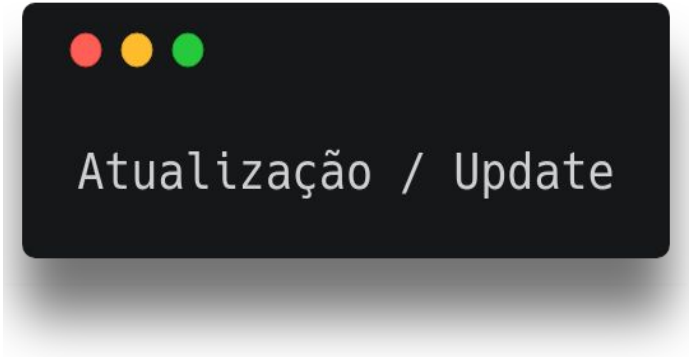
Maria percebeu que José havia enviado código para o servidor remoto e por isso já atualiza o seu repositório local

Sistema de Controle do Git

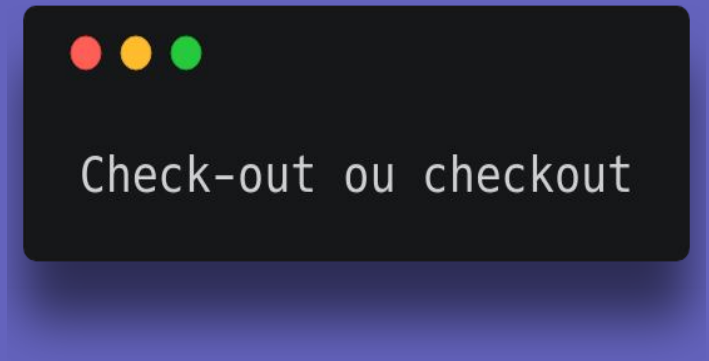




Terminologia

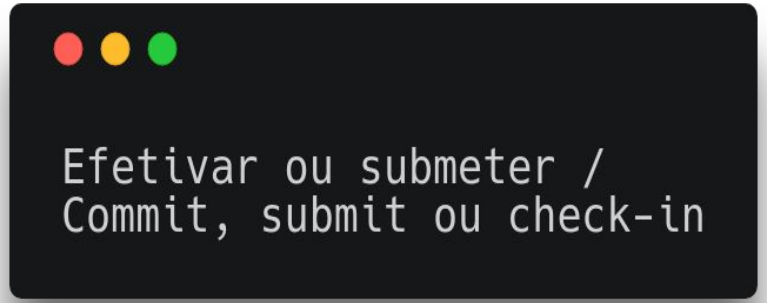


Atualiza o arquivo (pull) localmente baixando as novidades do Servidor, enviadas por outro desenvolvedor.

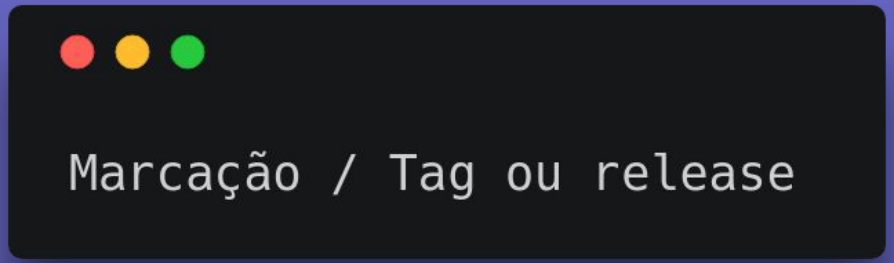


Quando não existe cópia local e é necessário **baixar (git clone)** todo o projeto do servidor. Trazendo os metadados junto com o projeto

É uma pasta no sistema operacional do desenvolvedor que mantém a cópia da última versão do projeto. É através da cópia local que o Controle de versão compara com a última versão do Servidor e sabe exatamente o que foi modificado.



Salva as alterações da cópia local.

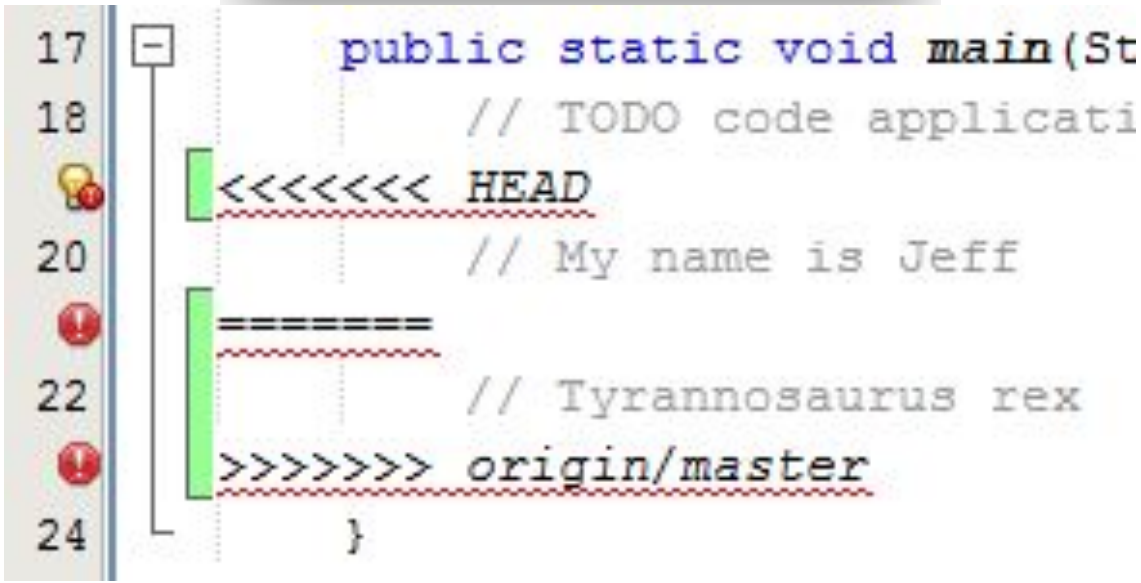


É atribuir um nome a um determinado "momento" do repositório de determinada informação.

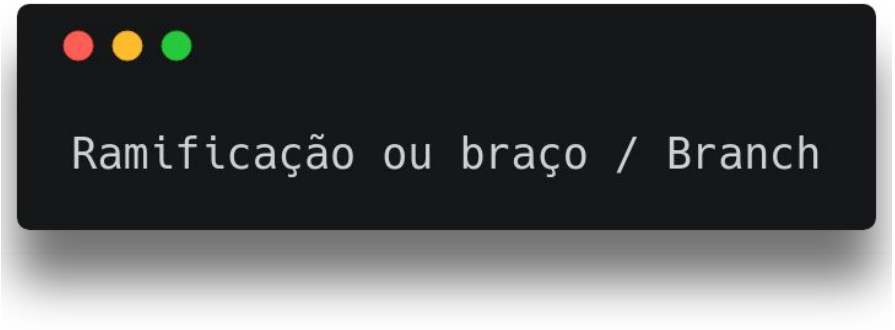
```
Mesclagem / Merge ou integration
```

E a mesclagem de um mesmo arquivo modificado pela pessoa A mesclando as modificações feita pela pessoa B **mantendo as duas alterações se possível**. O merge geralmente é feita localmente na atualização"**pull**" de um documento quando há uma **novas versão** no Servidor mais recente que a sua.

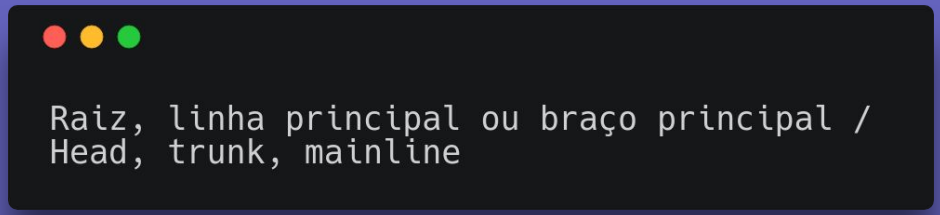
Conflito / Conflict



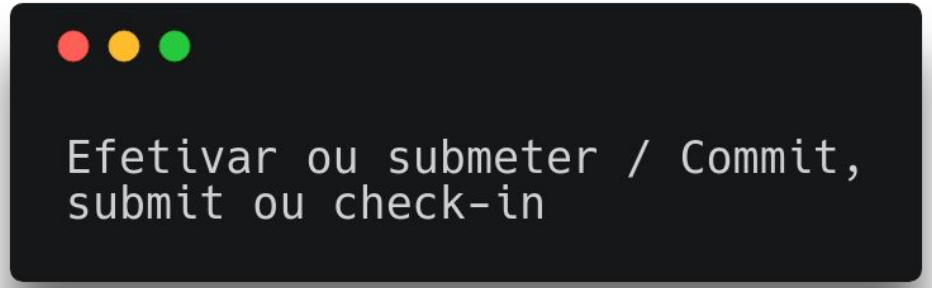
É a **mudança simultânea** de um mesmo documento por usuários diferentes.



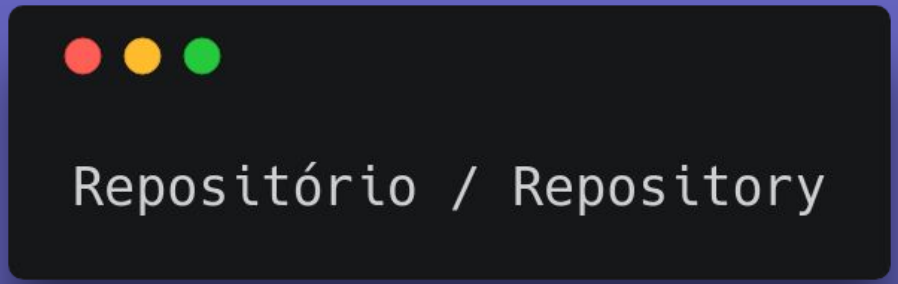
Quando o Desenvolvimento precisa ser dividida em duas ou mais ramos.



É o caminho de revisões que não se quebrou em um braço.

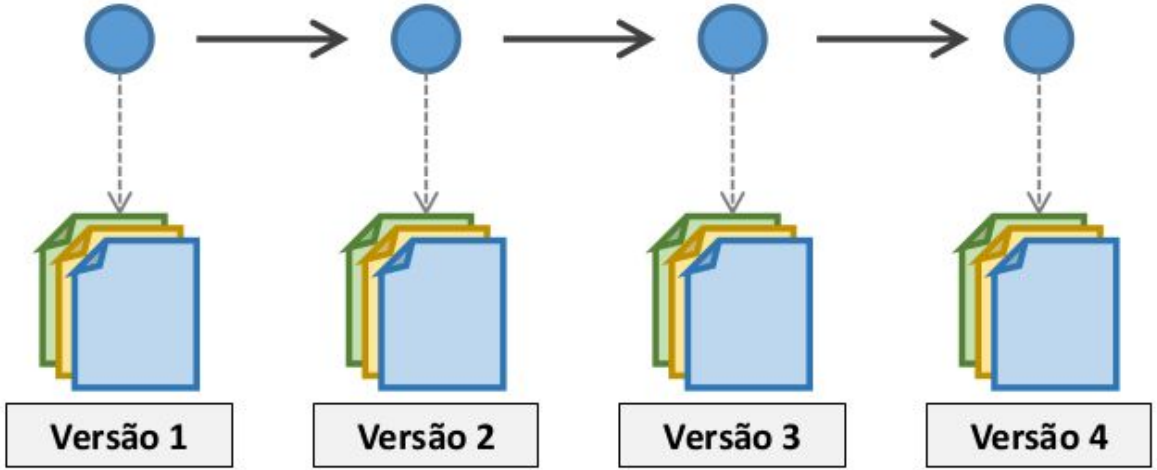


Quando o desenvolvedor enviar (**push**) as alterações da sua cópia local ao **Servidor remoto.**



Local no computador onde fica armazenado todas as versões, desde o **início do projeto até a o fim**. Onde pode existir mais de um repositório (Pasta, Diretórios, HD, etc).

Snapshots



Cada versão é uma "foto" do diretório de trabalho e será representado por um círculo azul denominado commit

Inicio

Primeiro commit

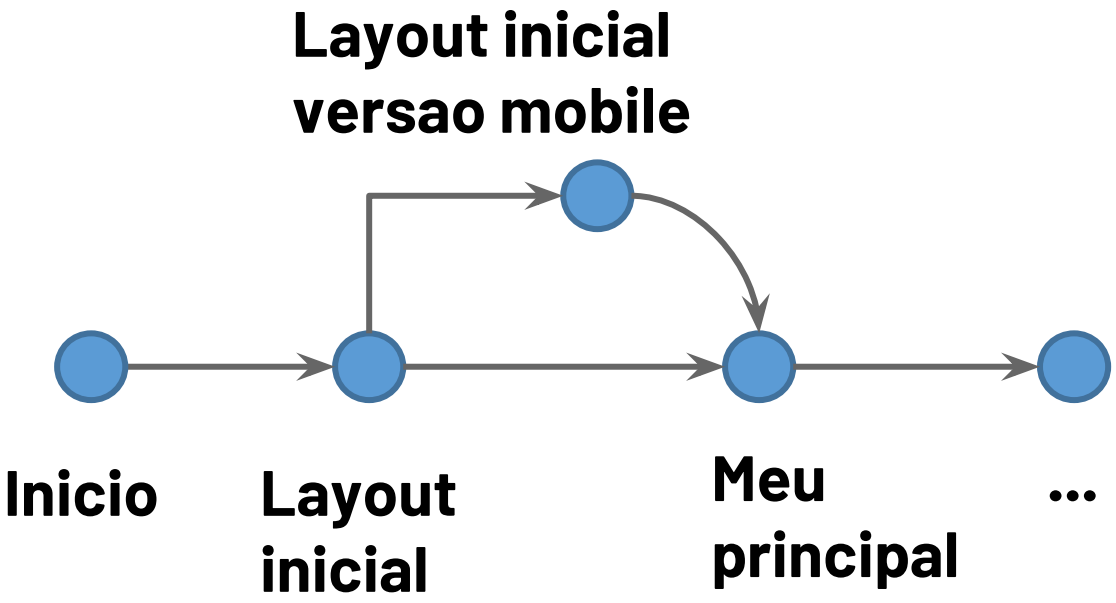
Inicio

Layout inicial

Meu principal

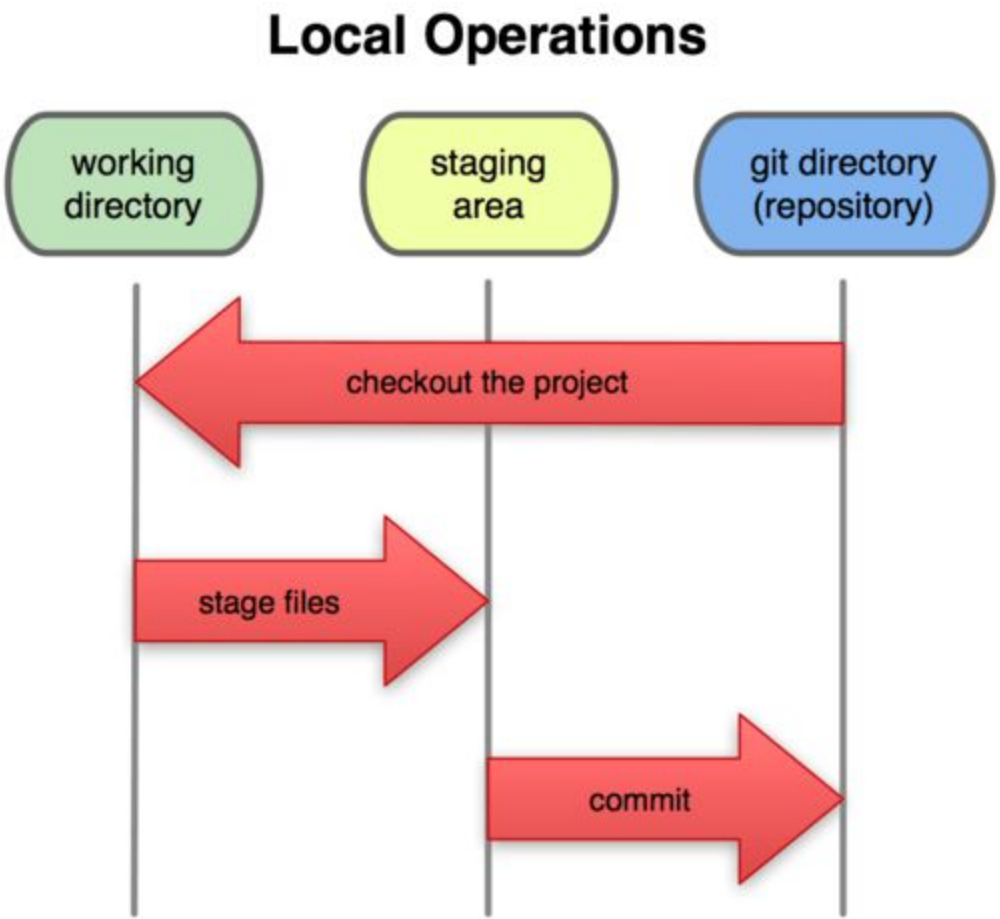
...

Desenvolvimento do projeto



Lembrando que no desenvolvimento do projeto outros caminhos pode ser tomados

Workflow basico de oparações no Git



1° modifica arquivos no seu diretório de trabalho.

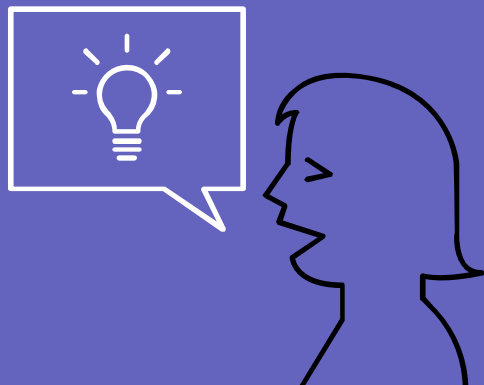
2° seleciona os arquivos, adicionando snapshots deles para sua área de preparação.

3° Com o commit os arquivos são levados da sua área de preparação para serem armazenados permanentemente no seu diretório Git.

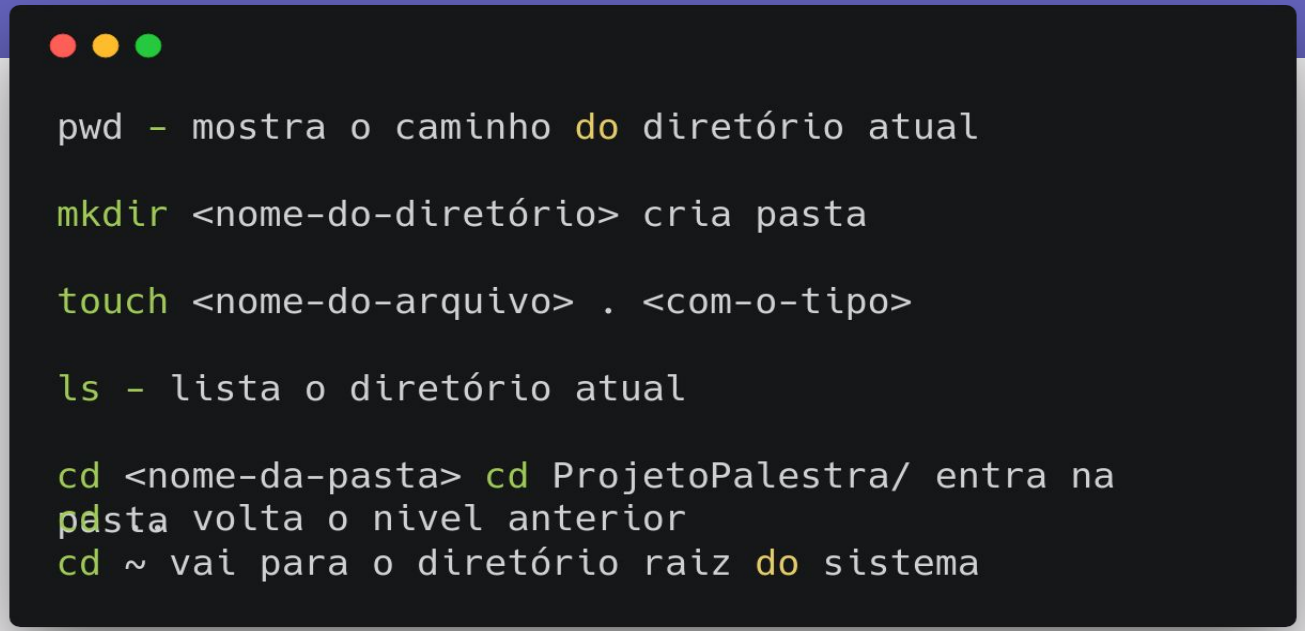
Comando basicos

Apesar de não ser nosso foco é interessante aprender a usar um interpretador de comando o **Shell**

O mais conhecido e **BASH**



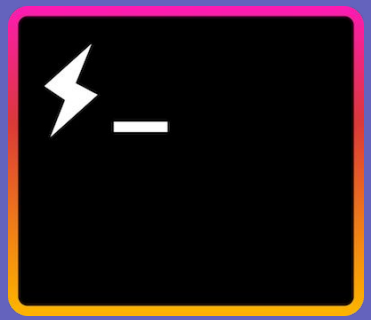
Um rápido tutorial de comandos do terminal



Terminal

Git Bash

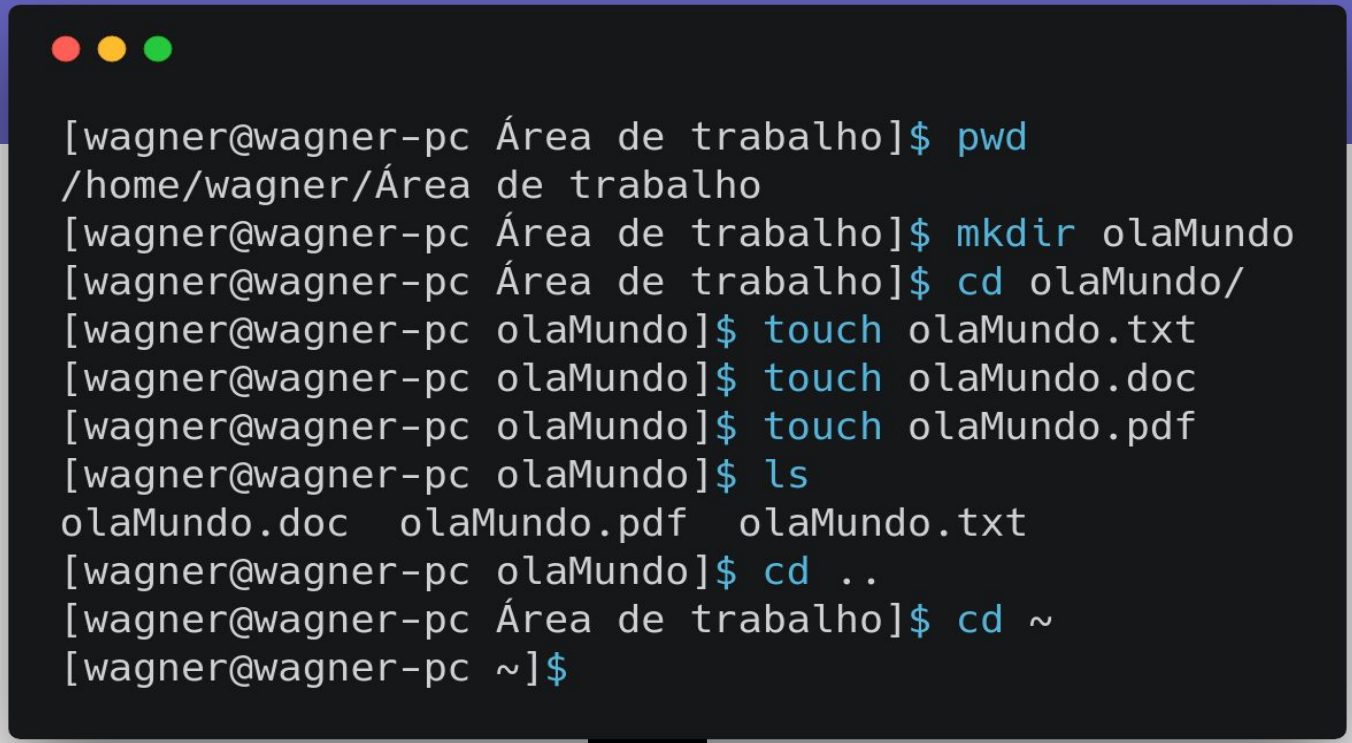
Cmder



hyperterminal

Cygwin

Teste !!!!


[wagner@wagner-pc Área de trabalho]$ pwd
/home/wagner/Área de trabalho
[wagner@wagner-pc Área de trabalho]$ mkdir olaMundo
[wagner@wagner-pc Área de trabalho]$ cd olaMundo/
[wagner@wagner-pc olaMundo]$ touch olaMundo.txt
[wagner@wagner-pc olaMundo]$ touch olaMundo.doc
[wagner@wagner-pc olaMundo]$ touch olaMundo.pdf
[wagner@wagner-pc olaMundo]$ ls
olaMundo.doc olaMundo.pdf olaMundo.txt
[wagner@wagner-pc olaMundo]$ cd ..
[wagner@wagner-pc Área de trabalho]$ cd ~
[wagner@wagner-pc ~]$

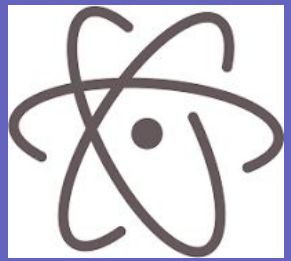
Escolha suas armas!!!

Sublime text

Meu favorito

Visual studio code

Em lua de mel com DEVs



Atom

Fui trocado!!

Code blocks

Ame ou odeie

Vim

hipster

Escolha suas armas!!!

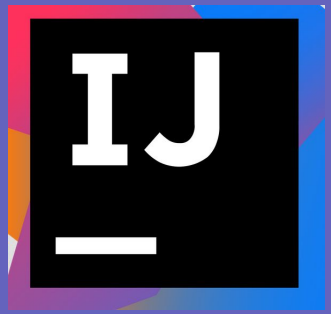
IDEs

Netbeans

Faculdade / Mercado

Eclipse

Lado negro da força



Intellij

Muito bom mas com custo elevado

Configuração do usuário

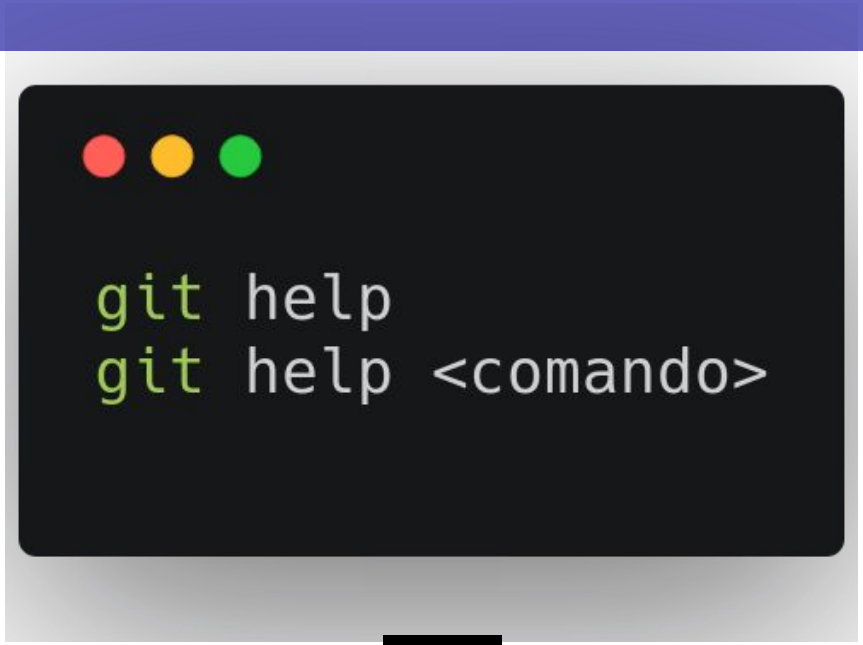
```
git config --global user.name "Wagner Rodrigo da silva"
git config --global user.email "wagnerrodrigo.pan@gmail.com"
```

Confirindo as informações

Confirindo as informações

```
[wagner@wagner-pc ProjetoPalestra]$ git config -l
user.name=Wagner Rodrigo da Silva
user.email=wagnerrodrigo.pan@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

Ficou perdido então ...



Exemplo

```
GIT-INIT(1)                                                Git Manual
                GIT-INIT(1)

NAME
       git-init - Create an empty Git repository or reinitialize an existing one

SYNOPSIS
       git init [-q | --quiet] [--bare] [--template=<template_directory>]
                 [--separate-git-dir <git dir>]
                 [--shared[=<permissions>]] [directory]

DESCRIPTION
       This command creates an empty Git repository - basically a .git directory with
subdirectories for objects,
       refs/heads, refs/tags, and template files. An initial HEAD file that references
the HEAD of the master branch
       is also created.
```

Iniciando o repositorio

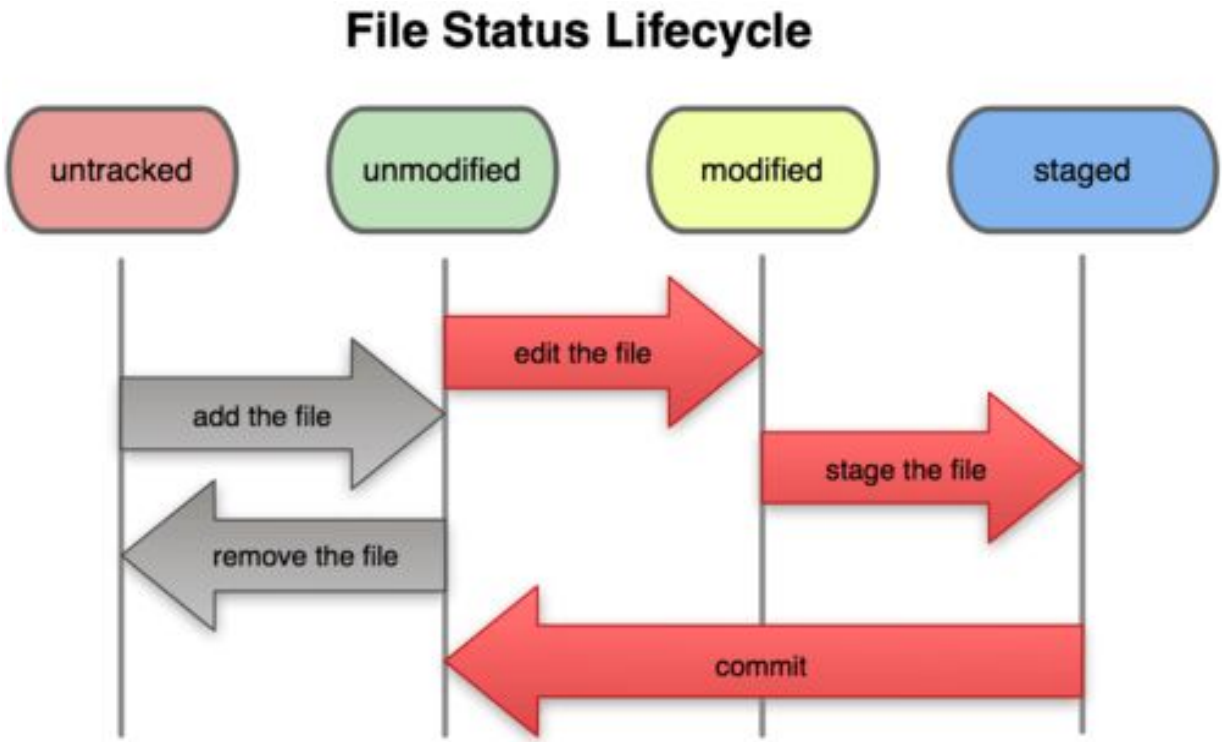
Verificando o que mudou

Nada mudou?


[wagner@wagner-pc ProjetoPalestra]$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)

É agora?

```
[wagner@wagner-pc ProjetoPalestra]$ 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)
```



UNTRACKED

O arquivo não está monitorado pelo git.

UNMODIFIED

Monitorado e sem alterações desde o último commit.

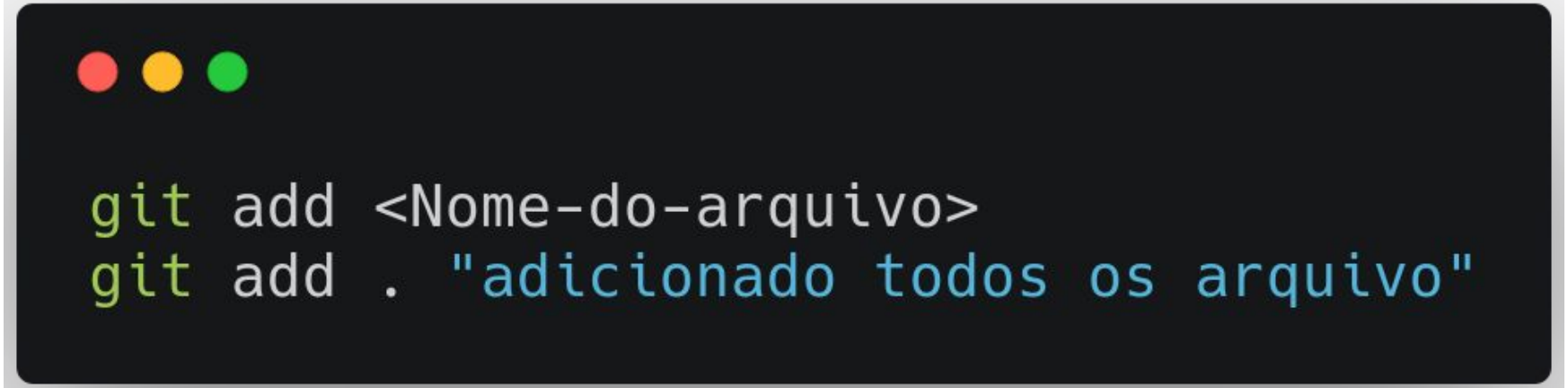
MODIFIED

Monitorado e com alterações desde o último commit.

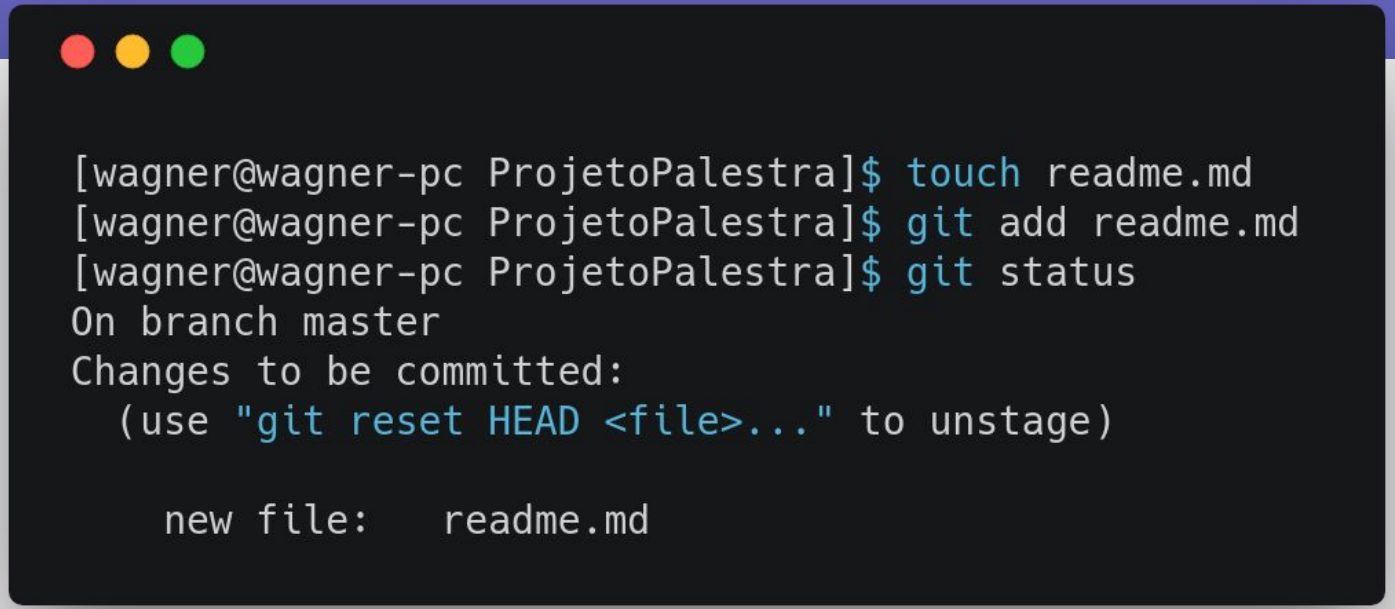
STAGED

Alteração marcada para publicação no próximo commit.

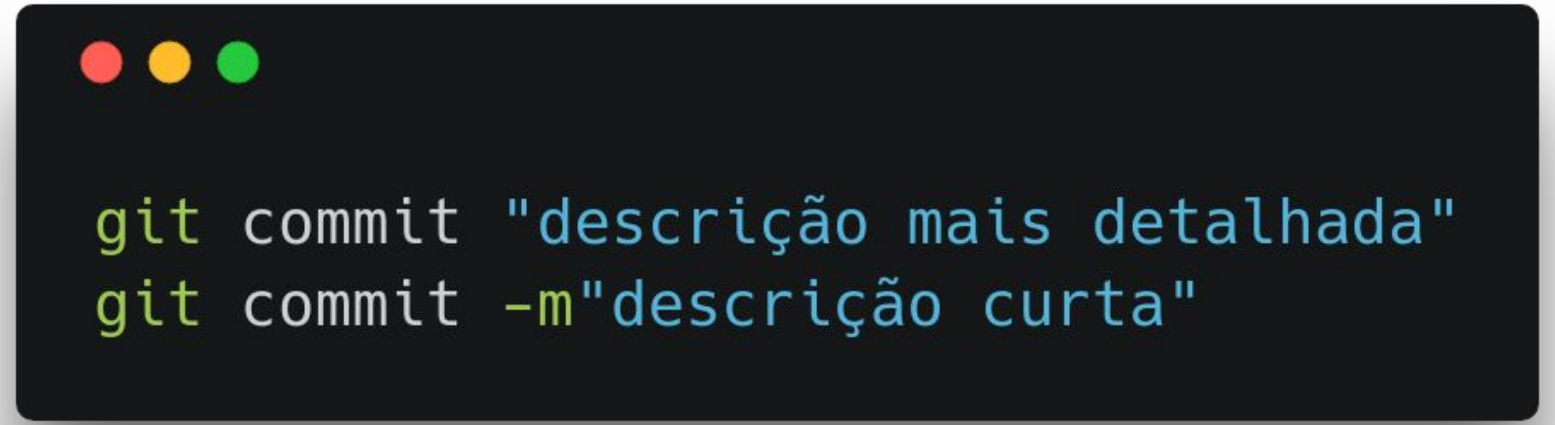
Adicionados arquivos



Exemplo


[wagner@wagner-pc ProjetoPalestra]$ touch readme.md
[wagner@wagner-pc ProjetoPalestra]$ git add readme.md
[wagner@wagner-pc ProjetoPalestra]$ git status
On branch master
Changes to be committed:
(use "git reset HEAD <file>..." to unstage)
new file: readme.md

Salvandos os arquivos “Commits”



Exemplo descrição curta

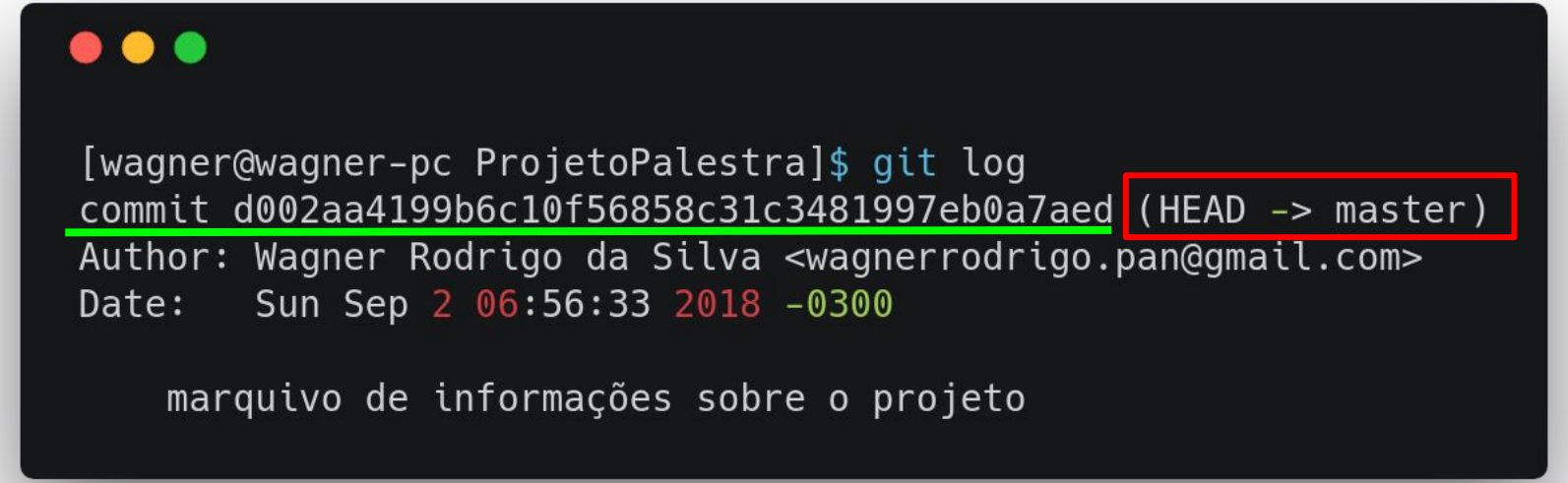
```
[wagner@wagner-pc ProjetoPalestra]$ git commit -m "arquivo de informações sobre o projeto"
[master (root-commit) 46801ef] arquivo de informações sobre o projeto
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Readme.md
```

Editor de texto muito conhecido no mundo linux

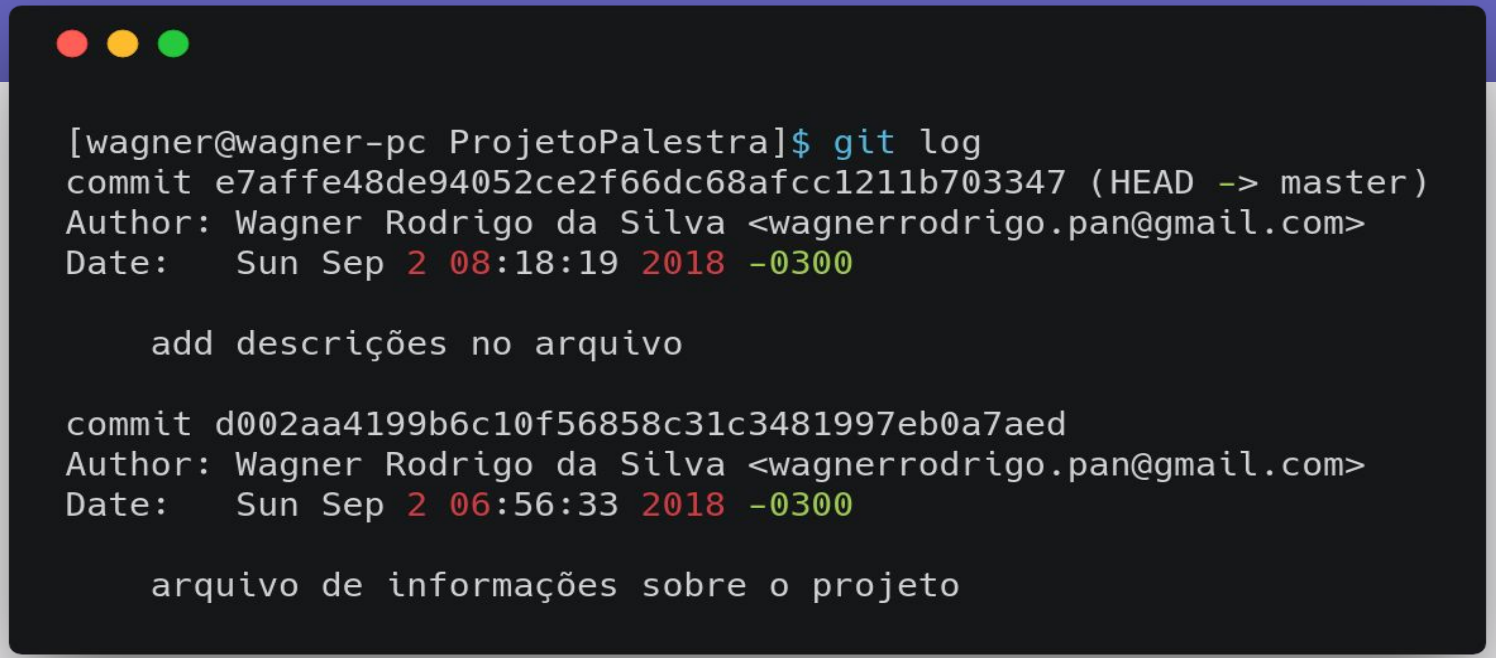
Verificando os “Commits”

```
git log
git log --pretty=oneline
git log --pretty=format:"%h - %an, %ar : %s"
git log --all --graph --decorate
git log --all --graph --decorate --oneline --simplify-by-decoration
```

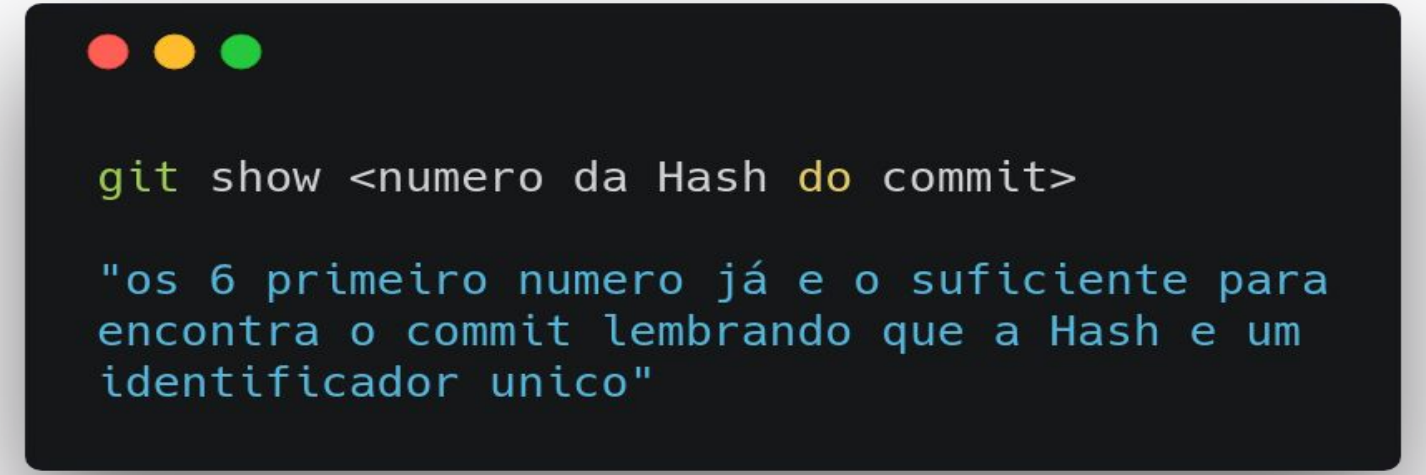
Buscar os “Commits”



Verificando os “Commits”



Verificando os “Commits”



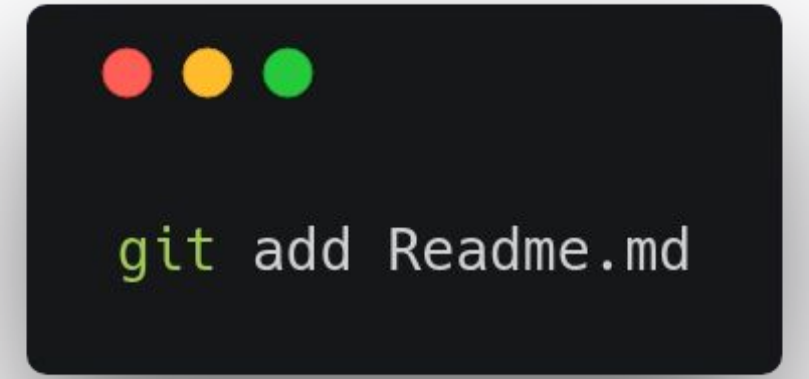
Verificando os “Commits”

```
[wagner@wagner-pc ProjetoPalestra]$ git show d002aa41
commit "d002aa4199b6c10f56858c31c3481997eb0a7aed"
Author: Wagner Rodrigo da Silva <wagnerrodrigo.pan@gmail.com>
Date:   Sun Sep 2 06:56:33 2018 -0300

    arquivo de informações sobre o projeto

diff --git a/Readme.md b/Readme.md
new file mode 100644
index 0000000..e69de29
```

Adicionado mais informações

```
# informações sobre o arquivo readme
- O arquivo Readme.md é renderizado no repositório git serve como uma carta de apresentação
- O Markdown e uma forma simplificada de escrever HTML muito utilizado em blog e posts de desenvolvedores
- O Medium utiliza o markdown


ERROOOOO......
```

Adicionado mais informações



Retirando da área de Stage

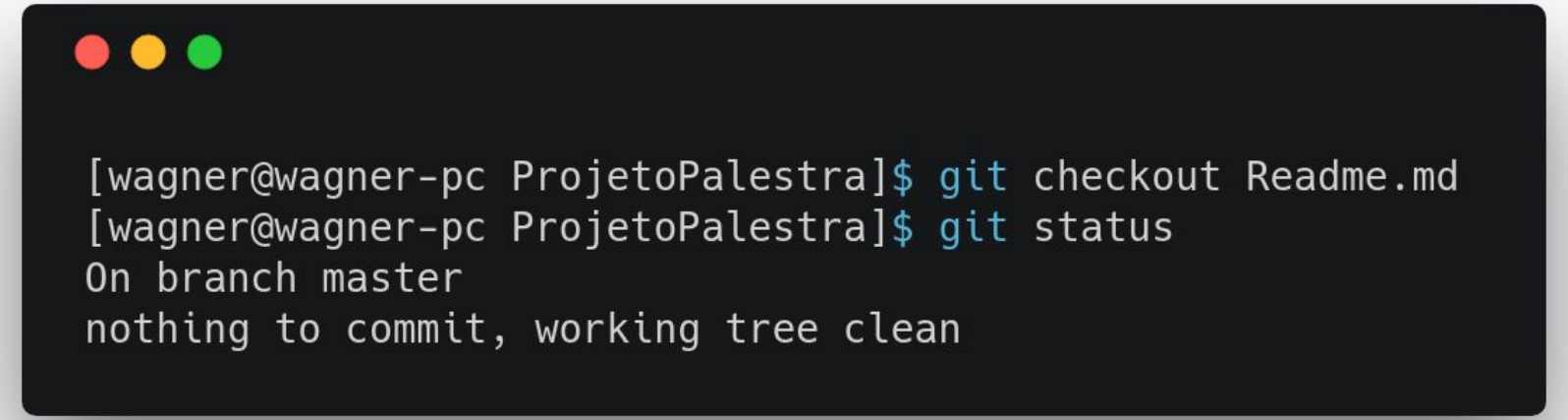
Cuidado com o checkout !!!

Retirando da area working directory

```
# informações sobre a os arquivos
- os arquivo são sobre o git e github
- esse arquivo e salvo na extensão md de markdown muito utilizado para
passar informações, posts de blog "em especial blog de devs" por se tratar
de um arquivo muito fácil de escrever e formatar.

Consertando o erro
```

Retirando da area working directory

Retirando da area working directory



Retirando da area working directory

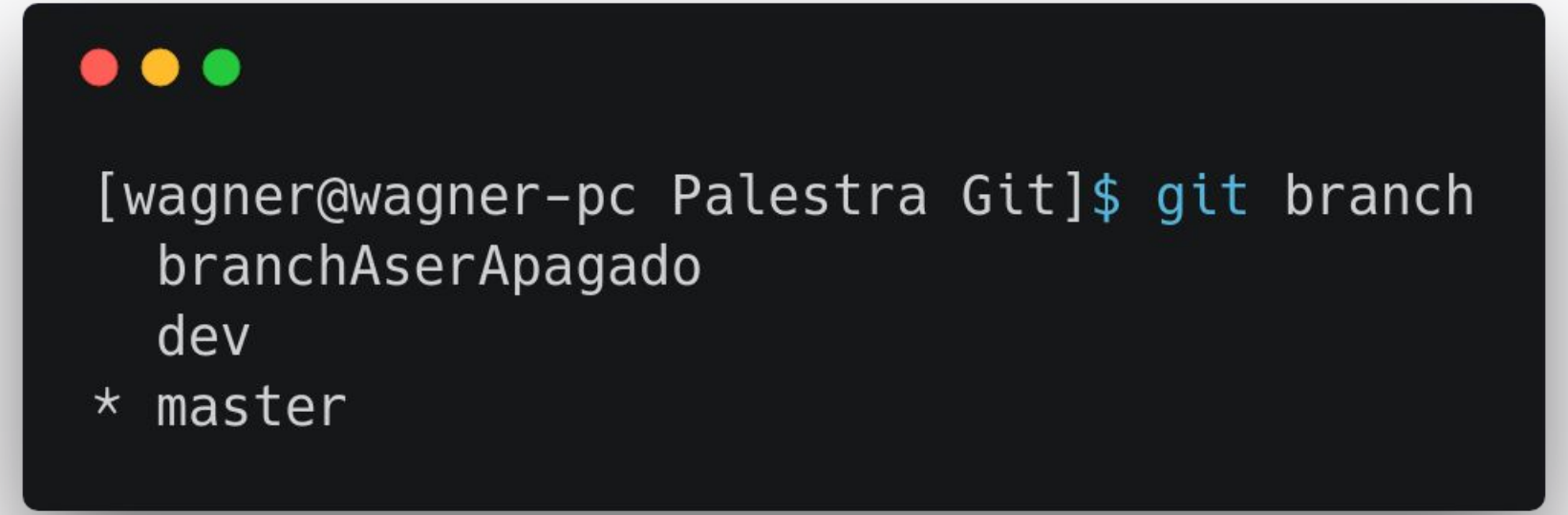
```
# informações sobre a os arquivos
- os arquivo são sobre o git e github
- esse arquivo e salvo na extensão md de markdown muito utilizado para
passar informações, posts de blog "em especial blog de devs" por se tratar
de um arquivo muito fácil de escrever e formatar.

ERRROO.........
```

Criando Branch

```
git branch "lista os branch e mostra o branch atual"

git branch <nome-do-branch>

git branch -v "lista o branch com o ultimo commit"

git checkou -b <nome-do-branch>
```

Criando Branch

Listando Branch

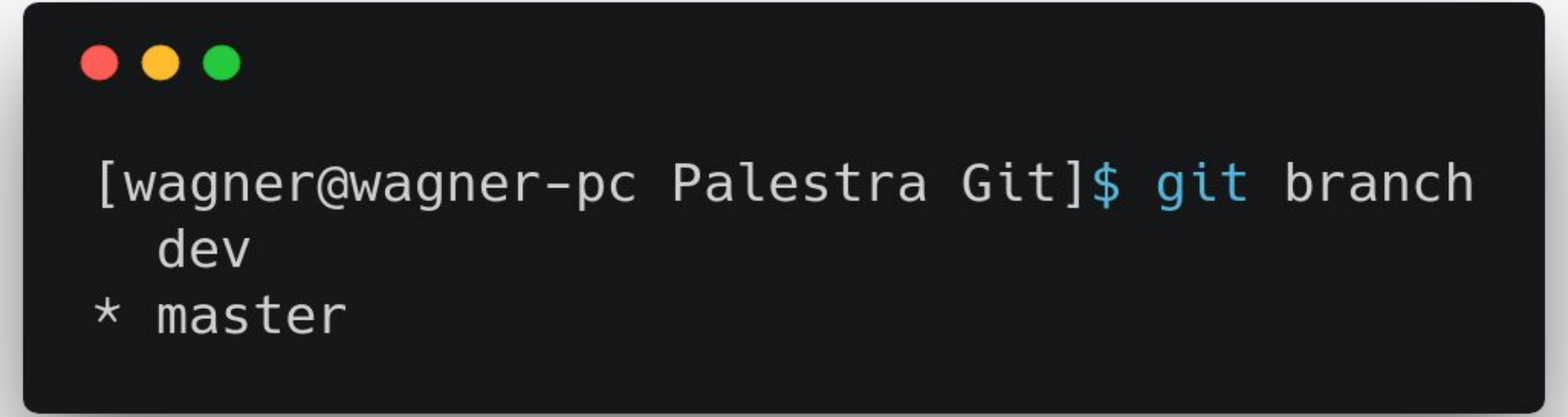


Apagando o Branch

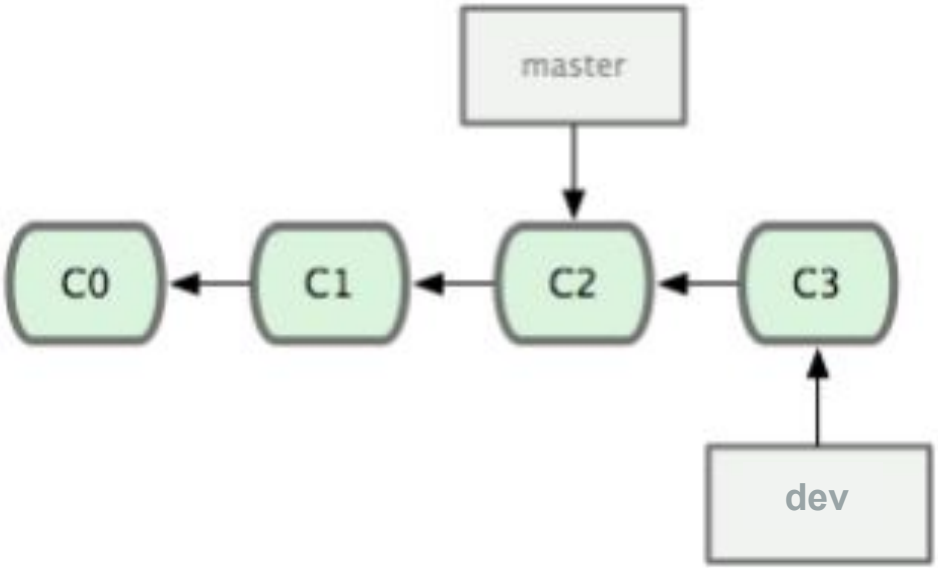
Deletando o branch

```
[wagner@wagner-pc Palestra Git]$ git branch
  dev
* master
[wagner@wagner-pc Palestra Git]$ git branch branchAserApagado
[wagner@wagner-pc Palestra Git]$ git branch
  branchAserApagado
  dev
* master
[wagner@wagner-pc Palestra Git]$ git branch -d branchAserApagado
Deleted branch branchAserApagado (was e4ecfa4).
[wagner@wagner-pc Palestra Git]$ git branch
  dev
* master
[wagner@wagner-pc Palestra Git]$
```

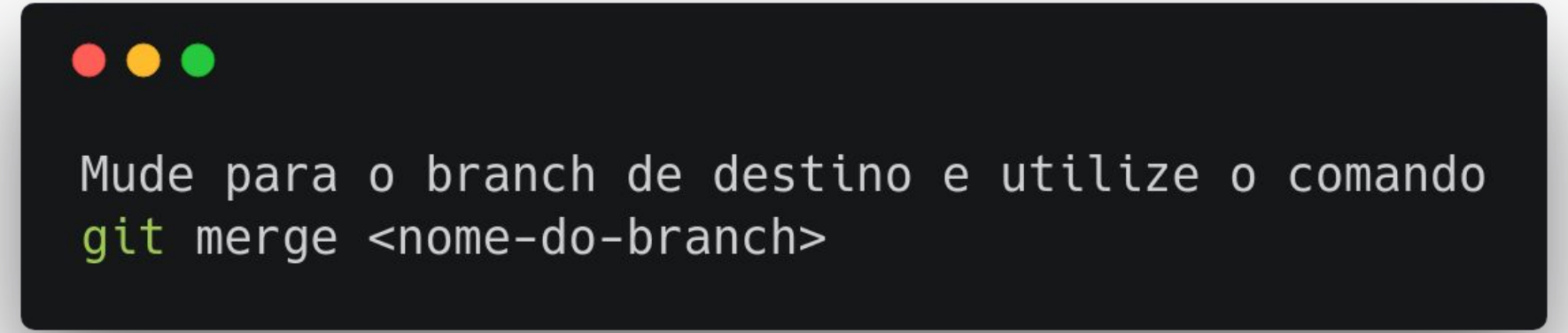
Depois de ser apagado



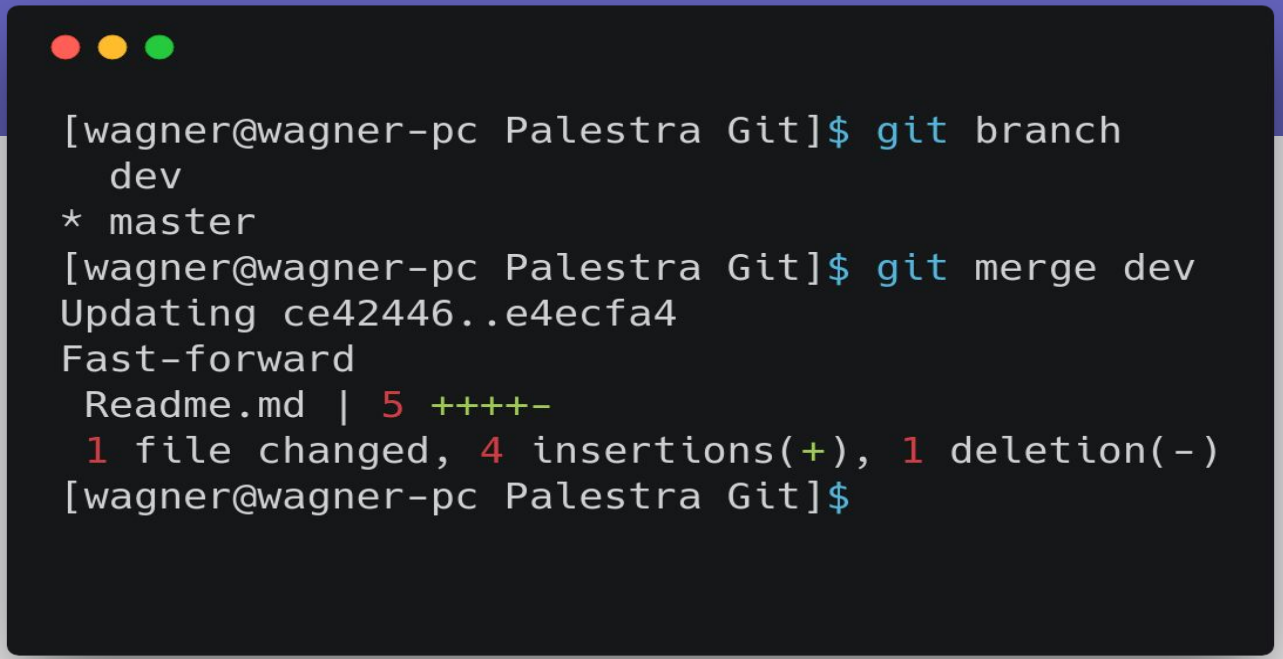
O commit feito no Branch



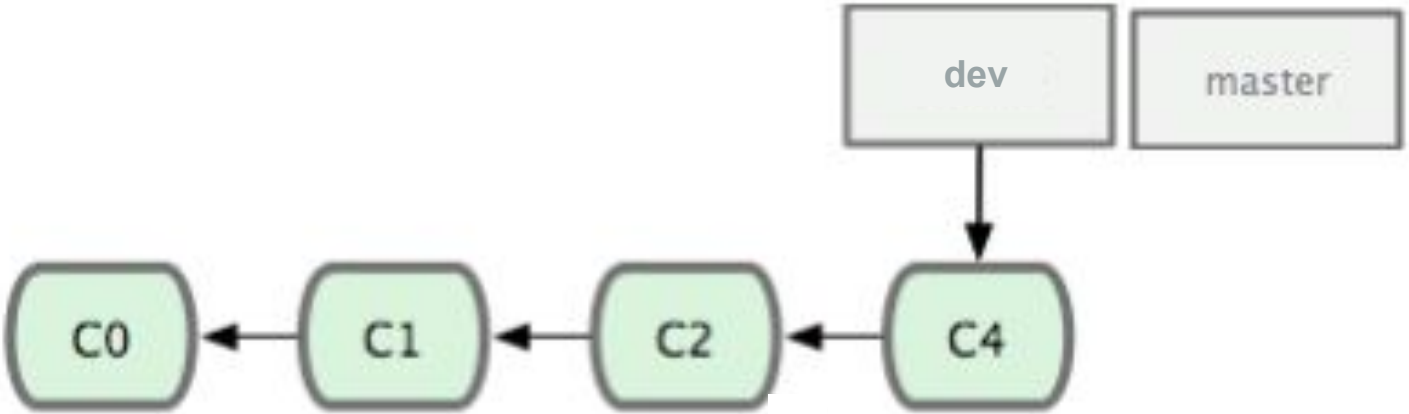
Merge


Mude para o branch de destino e utilize o comando
git merge <nome-do-branch>

Merge

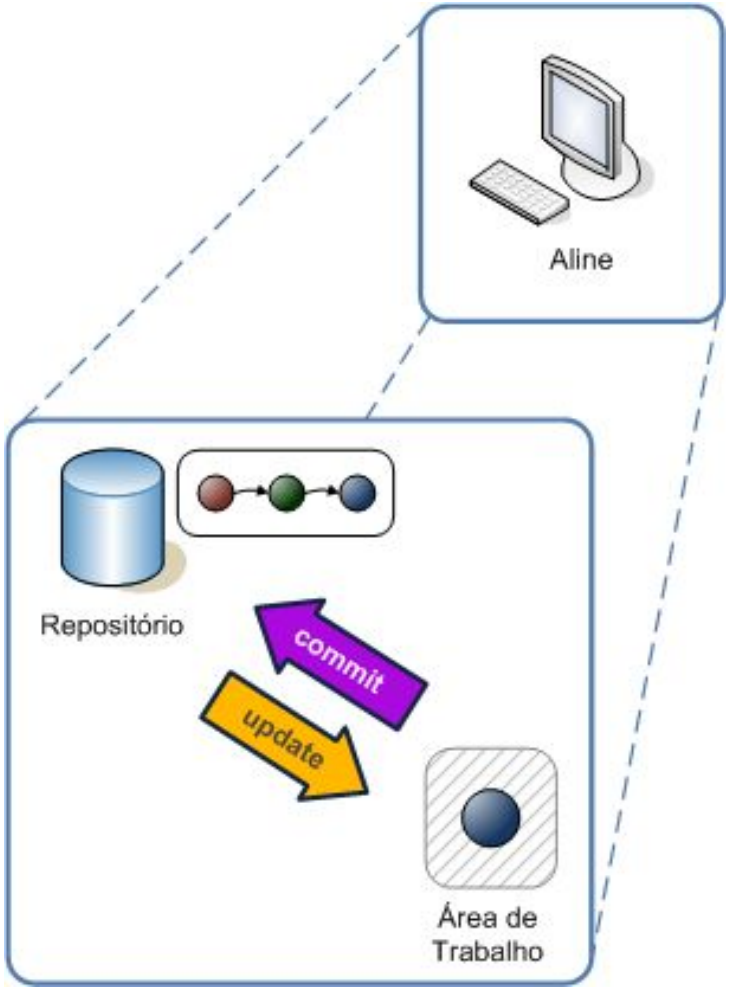

[wagner@wagner-pc Palestra Git]$ git branch
dev
* master
[wagner@wagner-pc Palestra Git]$ git merge dev
Updating ce42446..e4ecfa4
Fast-forward
Readme.md | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)
[wagner@wagner-pc Palestra Git]$

Merge



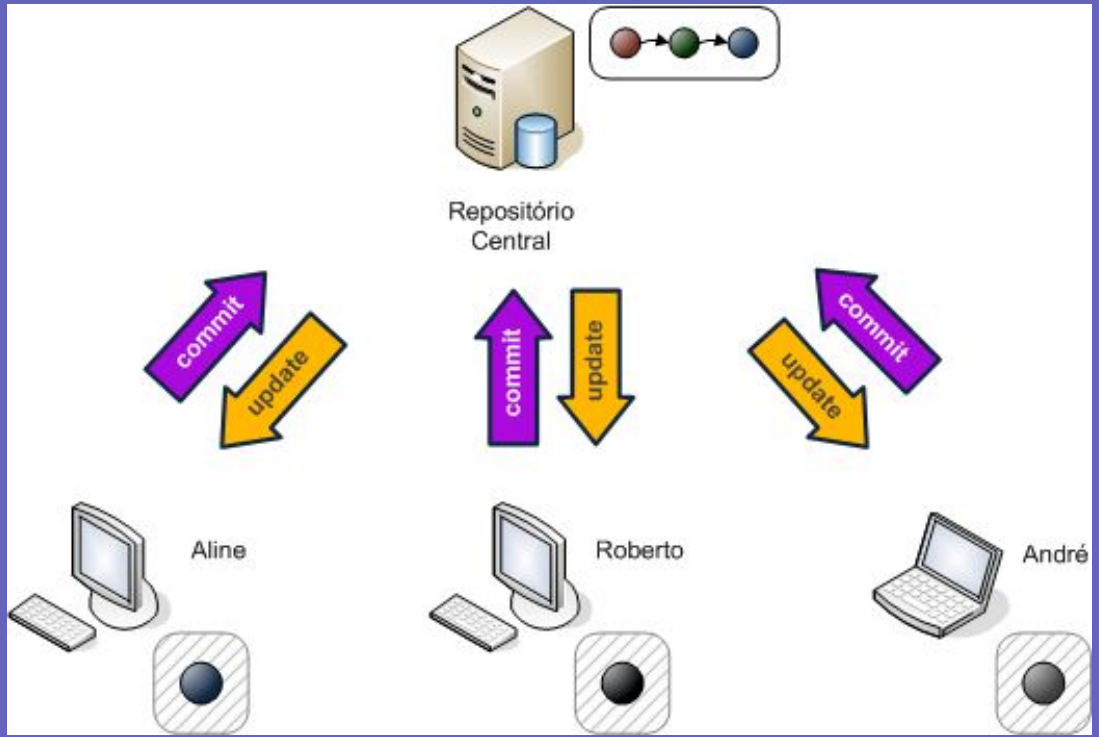
Dependendo da empresa e do workflow adotado o branch utilizado para fazer as modificações é apagado após o Merge

Repositórios



Repositórios Local

Repositórios Remoto



Git Push

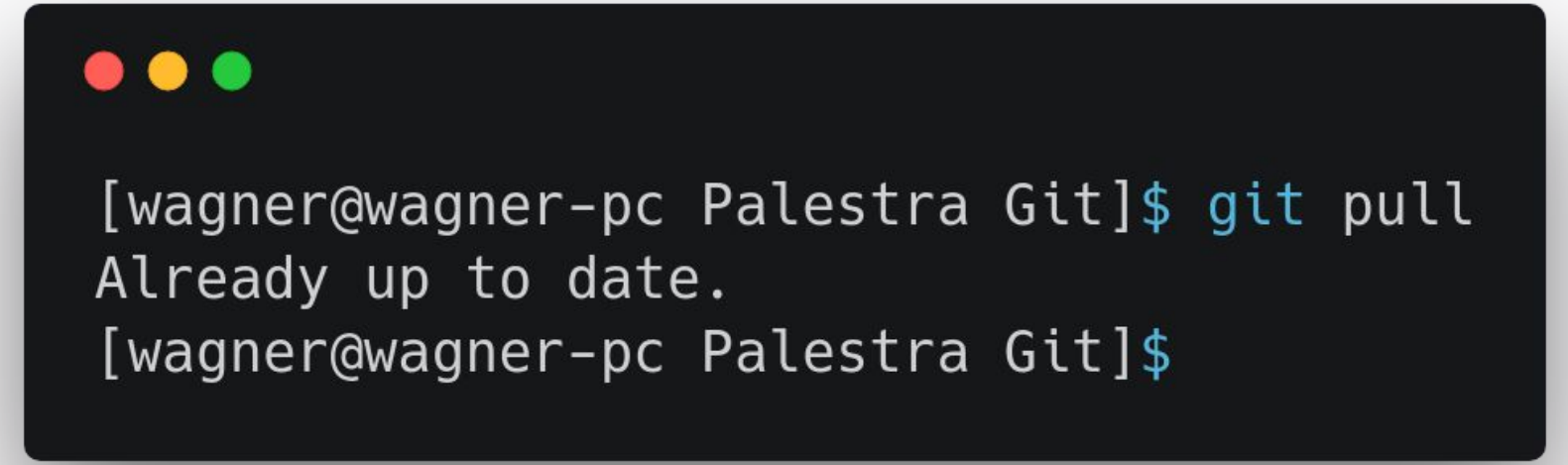
```
[wagner@wagner-pc Palestra Git]$ git push "envia o repositório local para o
repositório remoto"

[wagner@wagner-pc Palestra Git]$ git push -u origin master "comando usado a primeira
vez quando o repositório remoto e criado  depois utilizar o comando acima"
```

```
[wagner@wagner-pc Palestra Git]$ git push -u origin master
Username for 'https://github.com': wagnerrodrigo.pan@gmail.com
Password for 'https://wagnerrodrigo.pan@gmail.com@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 632 bytes | 632.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/wagnerrodrigo/Palestra-Git/pull/new/master
remote:
To https://github.com/wagnerrodrigo/Palestra-Git.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
[wagner@wagner-pc Palestra Git]$
```

Git Pull

```
git pull "atualiza o repositório local com o remoto"

O comando só funciona onde existe o repositório remoto!
se é a primeira que você cria o repositório remoto não tera arquivo nele
```

Git Pull

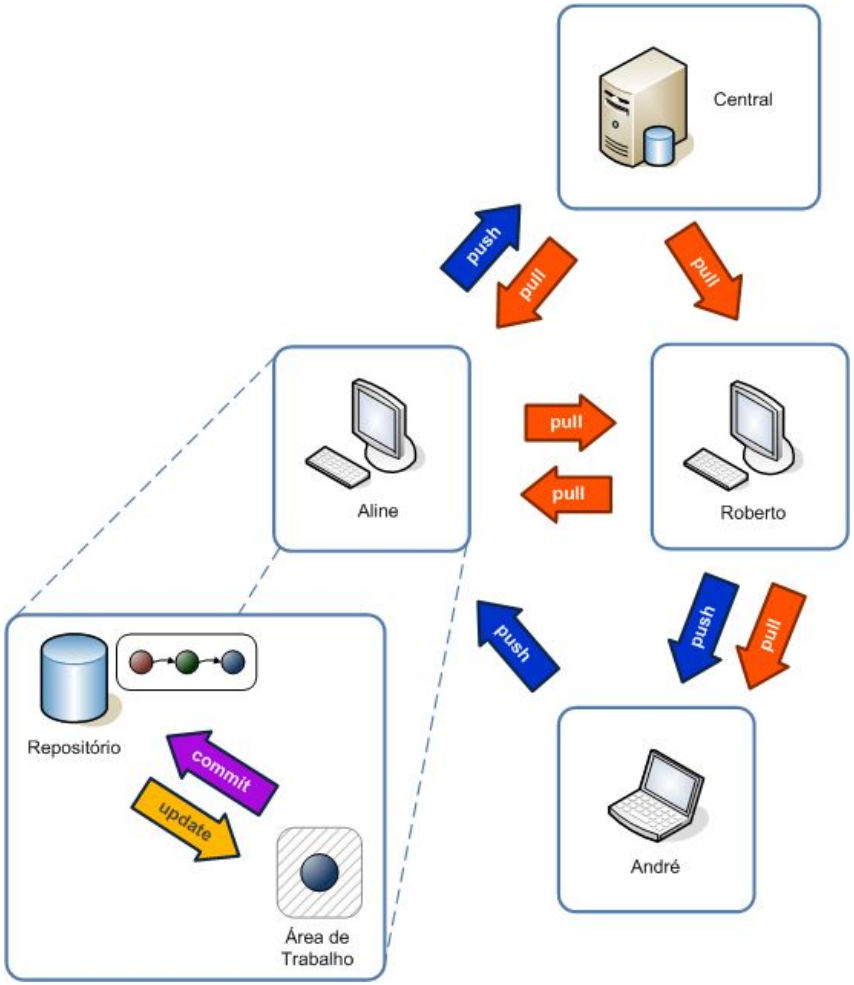

[wagner@wagner-pc Palestra Git]$ git pull
Already up to date.
[wagner@wagner-pc Palestra Git]$

Download de Repositórios Remotos

Git Clone

```
Como baixar um repositório com o histórico do código?
para isso utilizamos o comando

git clone <nome-do-repositório-remoto> "faz o download do repositório remoto"
```

Pull Request GitHub



Mas o que é Pull Request

O código modificados por um Dev deve ser "mergiados" pelo dono do repositório de origem

texto para novo ramo #1

Edit

Open **wagnerrodrigo** wants to merge 3 commits into `master` from `list`

Conversation 0 | Commits 3 | Checks 0 | Files changed 2 | +12 −1

wagnerrodrigo commented on 13 Mar 2017 — Owner

No description provided.

`texto para novo ramo` 67637f2

New changes since you last viewed — View changes

Wagner Rodrigo da Silva added some commits on 13 Mar 2017

`listas entre listas nao ordenada` 8813a97

`erro no novoRamo mudança de ramo faltava o commit` 6e98014

Add more commits by pushing to the **`list`** branch on **wagnerrodrigo/PJF**.

This branch has conflicts that must be resolved

Use the web editor or the command line to resolve conflicts.

Resolve conflicts

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe

Registrer Centro de Ensino Superior de Juiz de Fora #5279

Edit

Merged **philipto** merged 1 commit into `JetBrains:master` from `wagnerrodrigo:master` 23 days ago

Conversation 1 | Commits 1 | Checks 0 | Files changed 1 | +1 −0

wagnerrodrigo commented 23 days ago • edited

Registrer Centro de Ensino Superior de Juiz de Fora

`Centro de Ensino Superior de Juiz de Fora` a1a4d0c

philipto merged commit **a842392** into `JetBrains:master` 23 days ago — Revert

philipto commented 23 days ago

@wagnerrodrigo Pull request merged. Thank you!

Reviewers

No reviews

Assignees

No one assigned

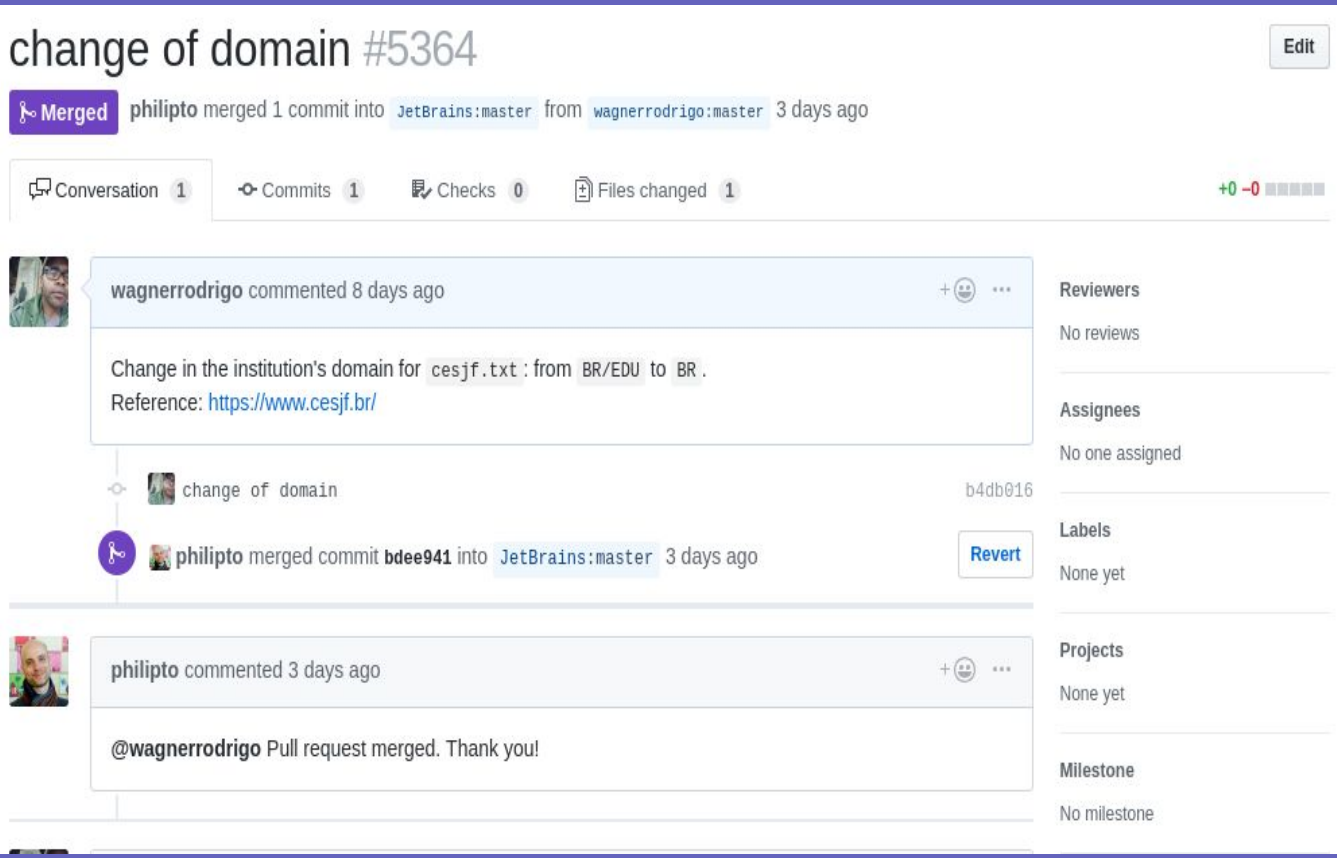
Labels

None yet

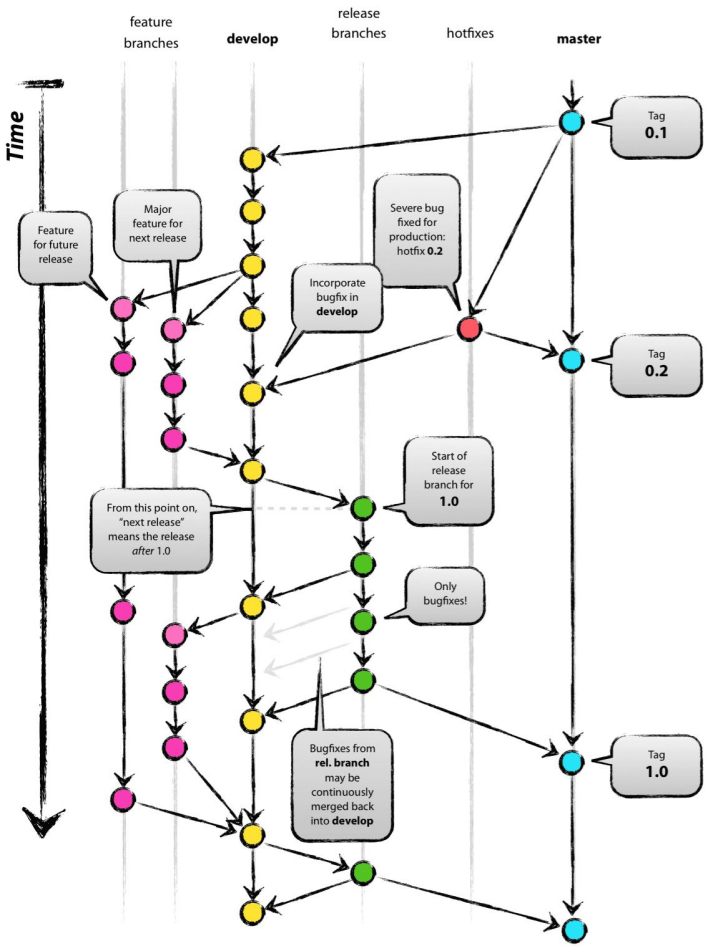
Projects

None yet

Milestone

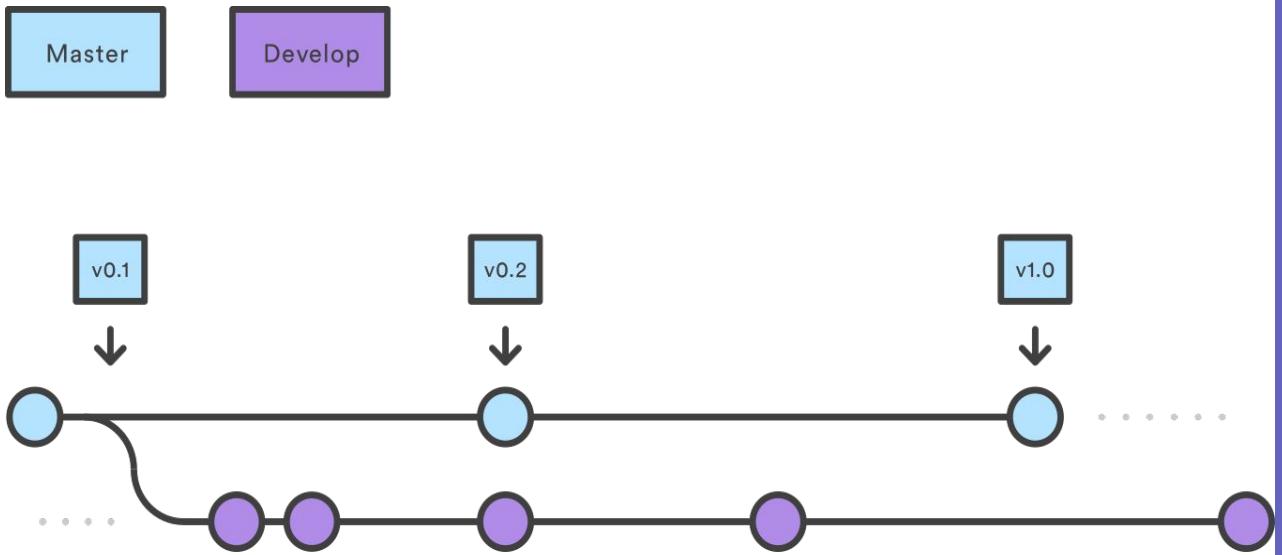

change of domain #5364
Edit
Merged
philipto merged 1 commit into JetBrains:master from wagnerrodrigo:master 3 days ago
Conversation 1
Commits 1
Checks 0
Files changed 1
+0 −0
wagnerrodrigo commented 8 days ago
Change in the institution's domain for cesjf.txt : from BR/EDU to BR .
Reference: https://www.cesjf.br/
change of domain
b4db016
philipto merged commit bdee941 into JetBrains:master 3 days ago
Revert
philipto commented 3 days ago
@wagnerrodrigo Pull request merged. Thank you!
Reviewers
No reviews
Assignees
No one assigned
Labels
None yet
Projects
None yet
Milestone
No milestone

Fluxo

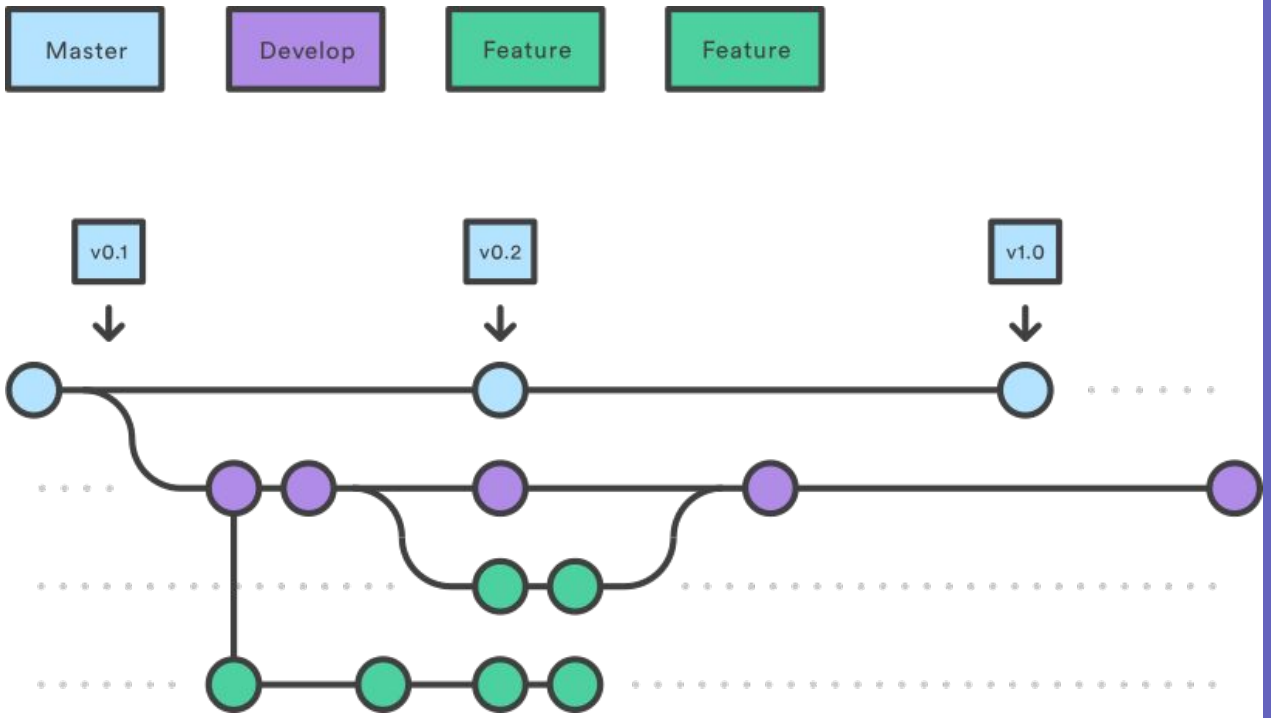


daniel kummer

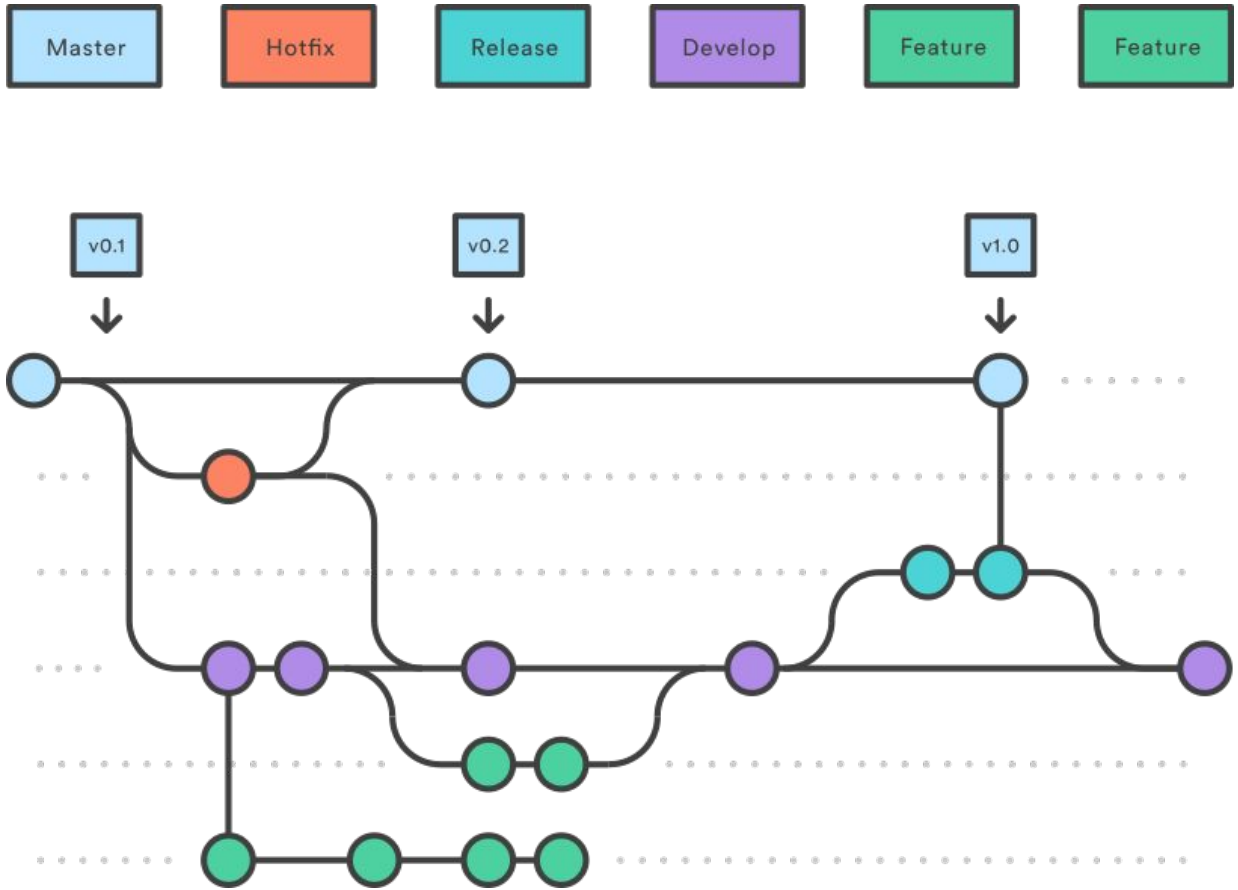
Forma de trabalhar com vários branch deixando cada um específico para uma única função



Master e Develop



Master ,
Develop,
Feature 1 ,
Feature 2



Master ,
HortFix,
Release,
Develop,
Feature 1 ,
Feature 2...

Cada empresa adota um fluxo de trabalho com os branch.

Extra

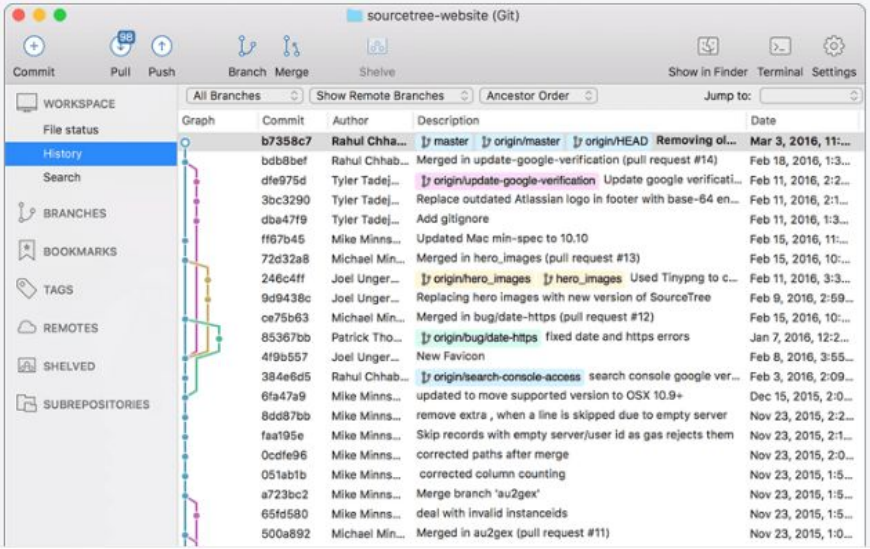
Sourcetree

Download free

Simplicity and power in a beautiful Git GUI

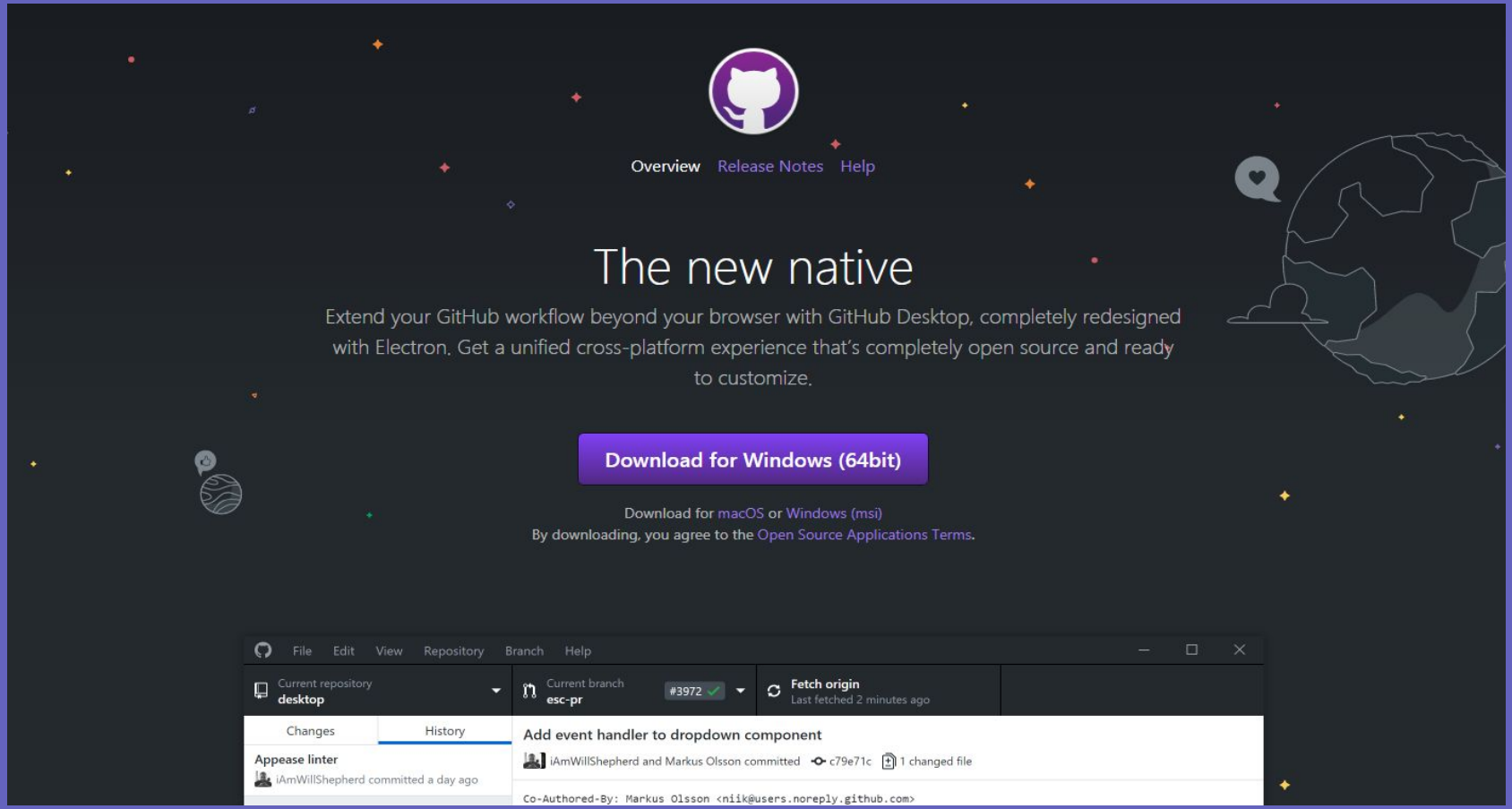
Download for Mac OS X

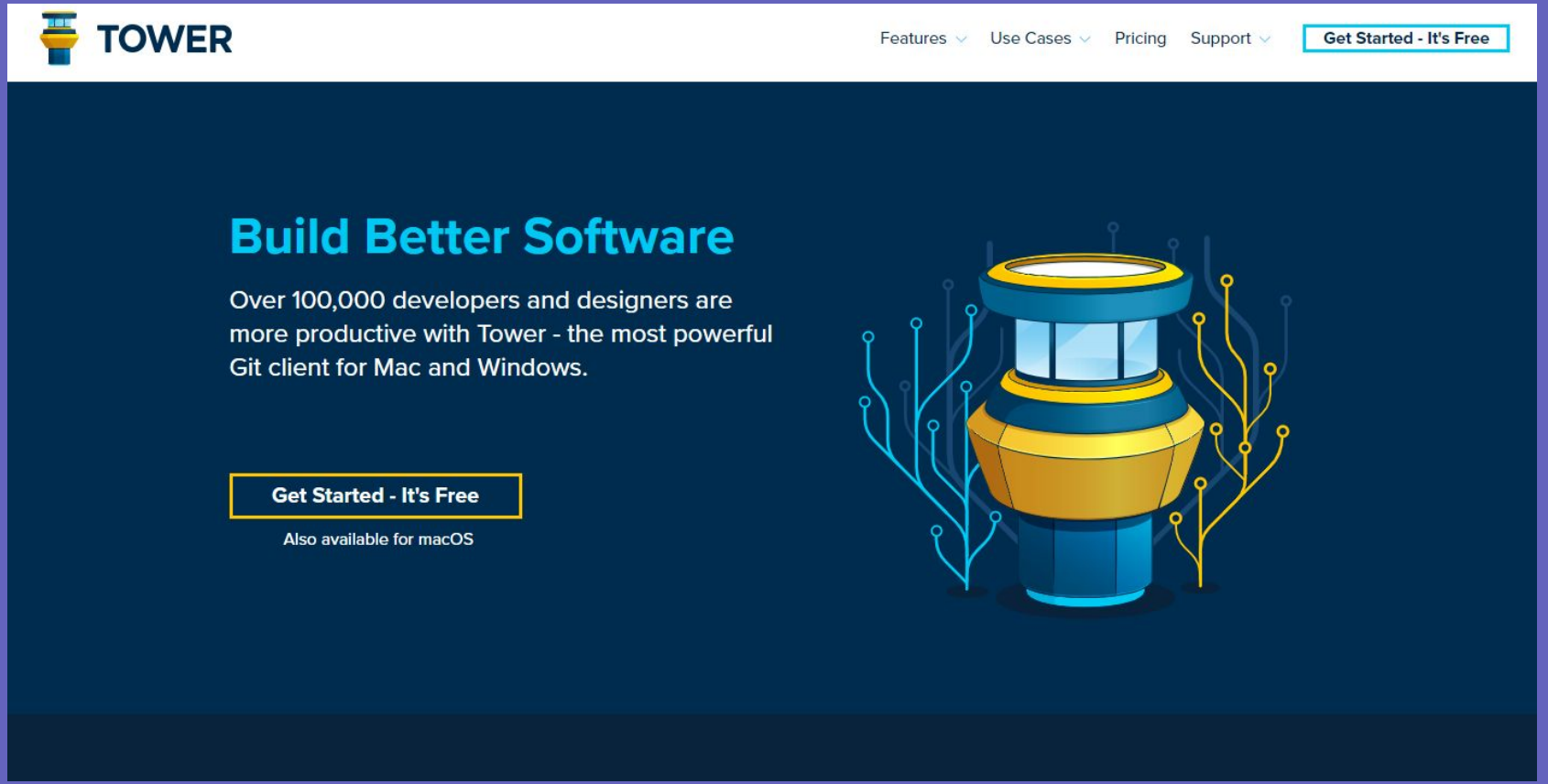
Also available for Windows



A free Git client for Windows and Mac


Git Client
Glo Boards
Pricing
Why GitKraken
My Account
axosoft
GitKraken
The legendary Git GUI client for Windows, Mac and Linux
Download Free Now
GITKRAKEN V4.0 - FOR UBUNTU LTS 14.04+, DEBIAN 8+
See All Platforms


Overview
Release Notes
Help
The new native
Extend your GitHub workflow beyond your browser with GitHub Desktop, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize.
Download for Windows (64bit)
Download for macOS or Windows (msi)
By downloading, you agree to the Open Source Applications Terms.
File
Edit
View
Repository
Branch
Help
Current repository
desktop
Current branch
esc-pr
#3972
Fetch origin
Last fetched 2 minutes ago
Changes
History
Add event handler to dropdown component
Appease linter
iAmWillShepherd committed a day ago
iAmWillShepherd and Markus Olsson committed
c79e71c
1 changed file
Co-Authored-By: Markus Olsson <niik@users.noreply.github.com>


TOWER
Features
Use Cases
Pricing
Support
Get Started - It's Free
Build Better Software
Over 100,000 developers and designers are more productive with Tower - the most powerful Git client for Mac and Windows.
Get Started - It's Free
Also available for macOS

Ferramenta de Merge

Meld

Get it | News | Features | Help | Wiki | Development

What is Meld?

Meld is a visual diff and merge tool targeted at developers. Meld helps you compare files, directories, and version controlled projects. It provides two- and three-way comparison of both files and directories, and has support for many popular version control systems.

Meld helps you review code changes and understand patches. It might even help you to figure out what is going on in that merge you keep avoiding.

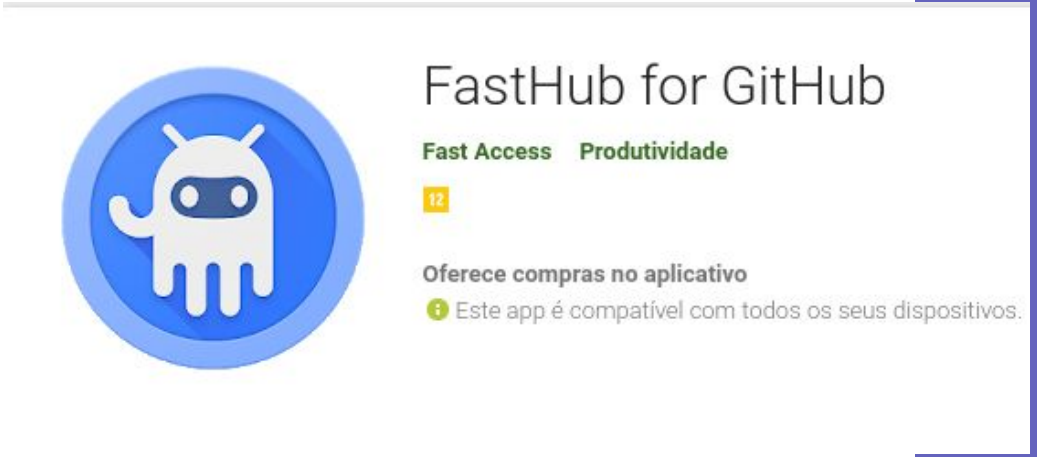
Git Quick Reference

Easy Intent Produtividade

L

Este app é compatível com todos os seus dispositivos.

Comando do Git app android



Cliente do gitHub para android

Obrigado!

Duvidas?

https://github.com/wagnerrodrigo

141