



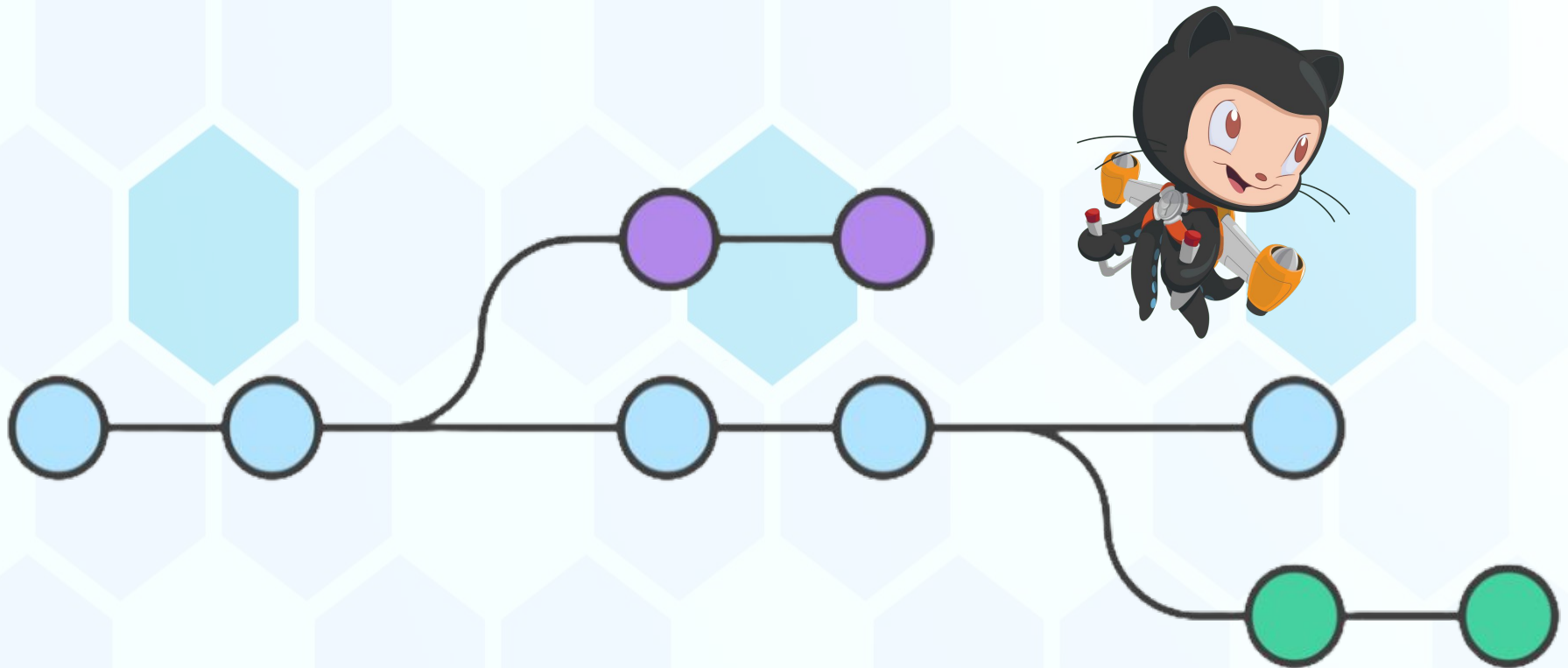
UFABC

IEEE Student Branch



A Student Chapter of the IEEE Computer Society

Introdução ao Git / GitHub



Palestrante



Raul Duarte

Técnico Eletroeletrônico; Graduando em Ciência da Computação e Engenharia da Informação pela UFABC; Presidente da Computer Society UFABC.

O Que Veremos?

❖ Teoria 01:00

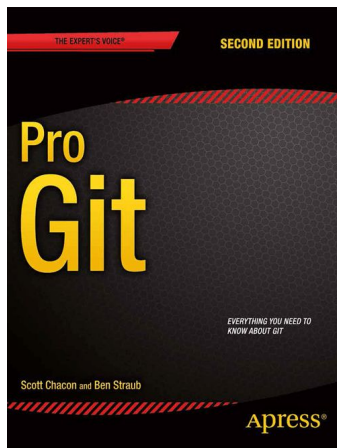
- Controle Versão
- História do Git
- Fluxo de Estados
- Instalação e Configuração
- Noções Básicas
- Boas Práticas
- Branching
- Github
- Integrações

❖ Intervalo 00:15

❖ Projeto Dirigido 01:30

❖ Tirando Dúvidas 00:15

Referências



[Git - Book](#)

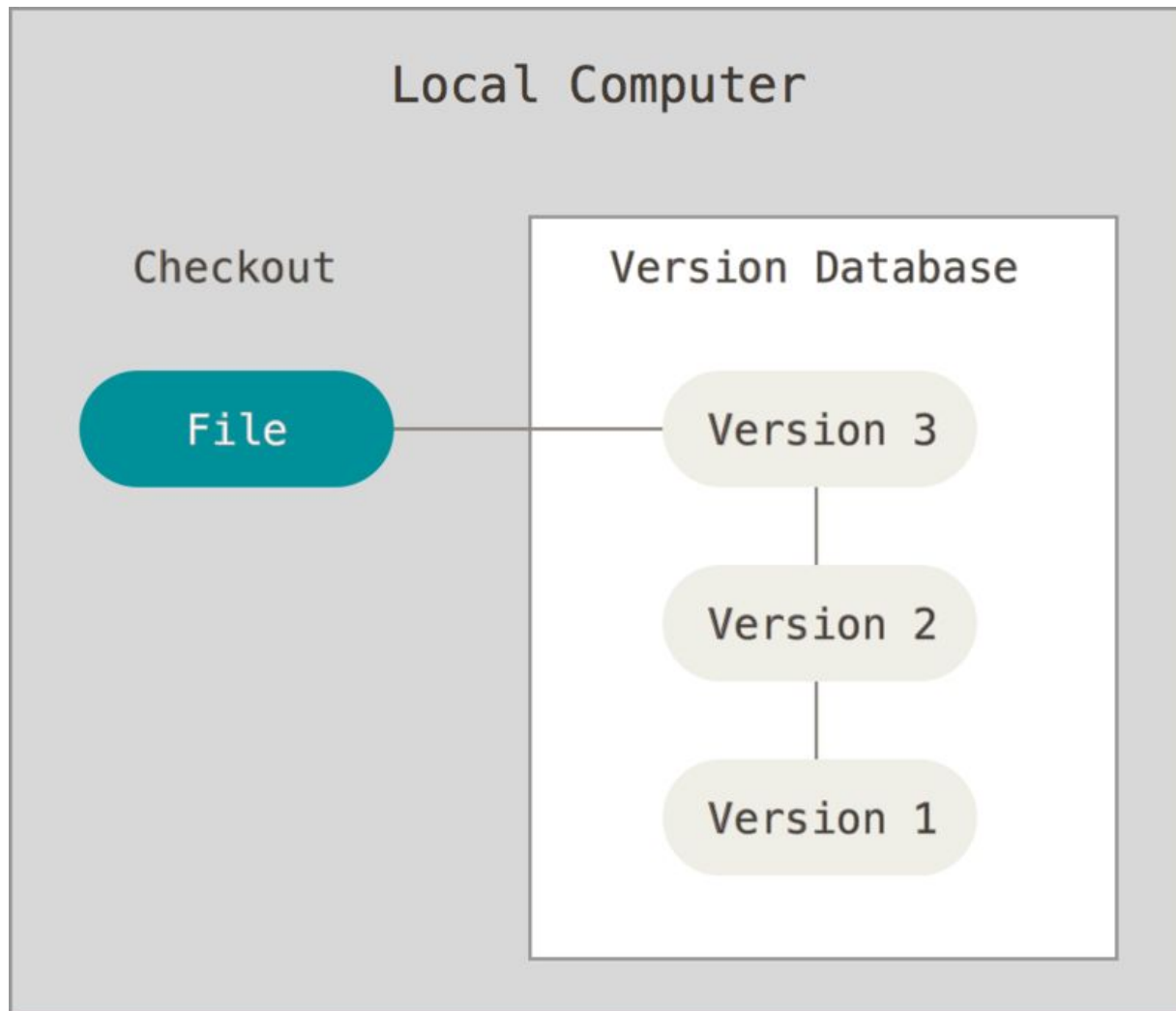


[Git User Manual](#)

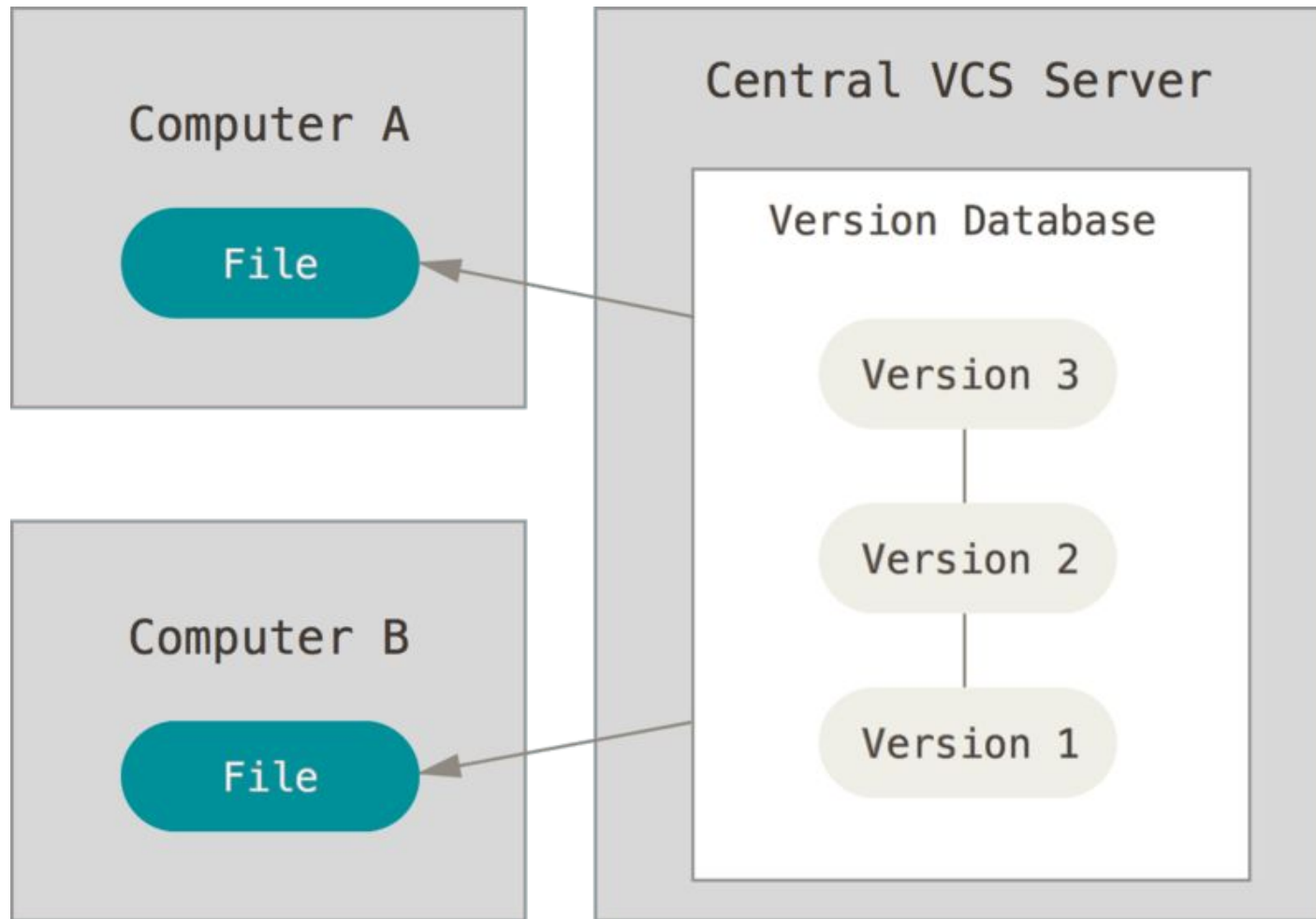


1. Controle de Versão

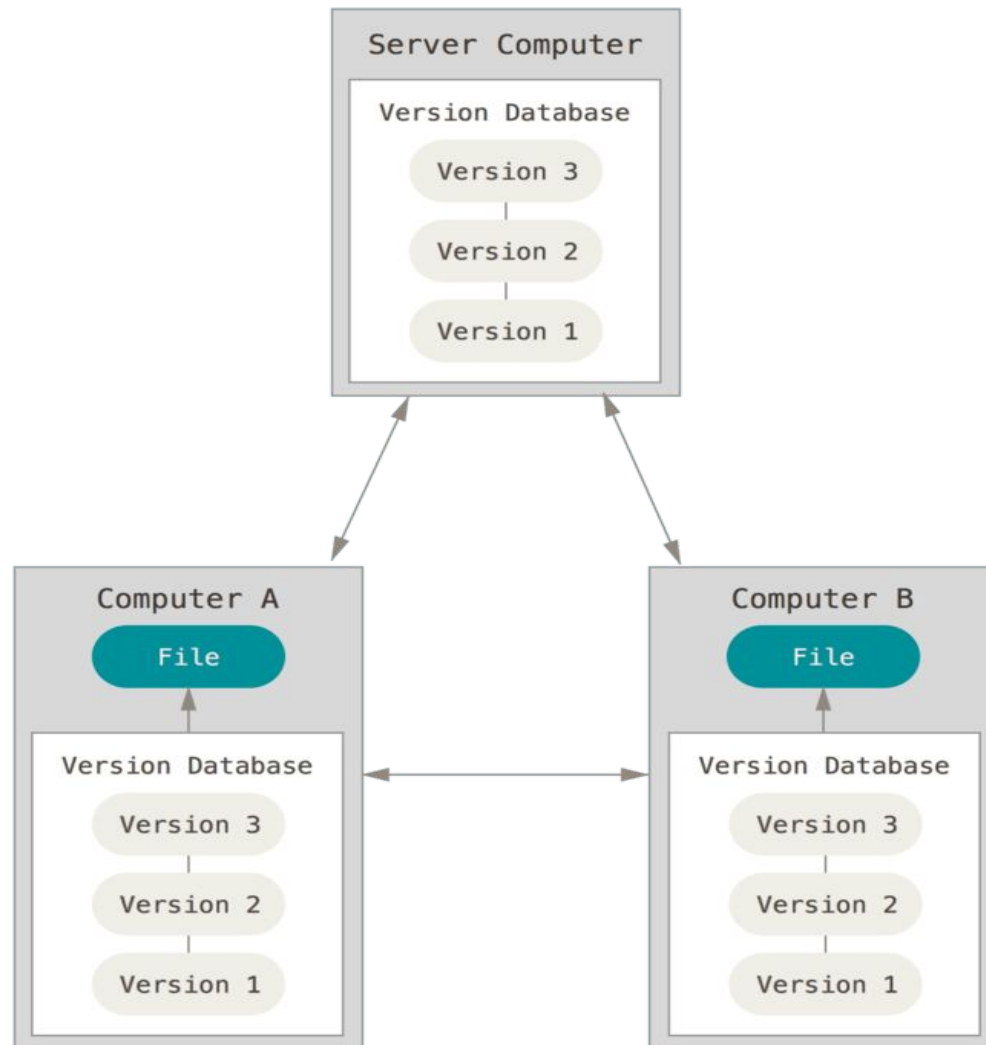
Controle de Versão Local



Controle de Versão Centralizado



Controle de Versão Distribuído





2. História do Git

História do Git

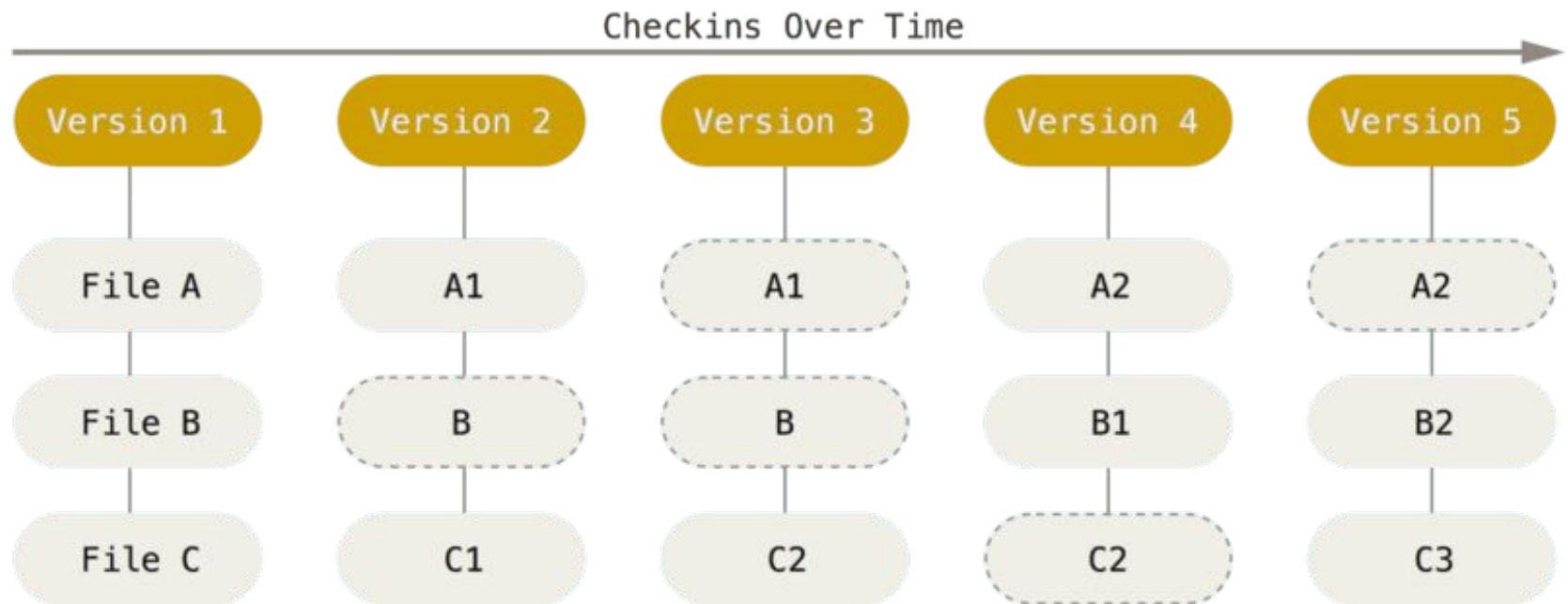


- (1991-2002) Desenvolvimento do Núcleo do Linux
- (2002 - 2005) Uso da DVCS proprietária BitKeeper
- (2005) Desenvolvimento do Git

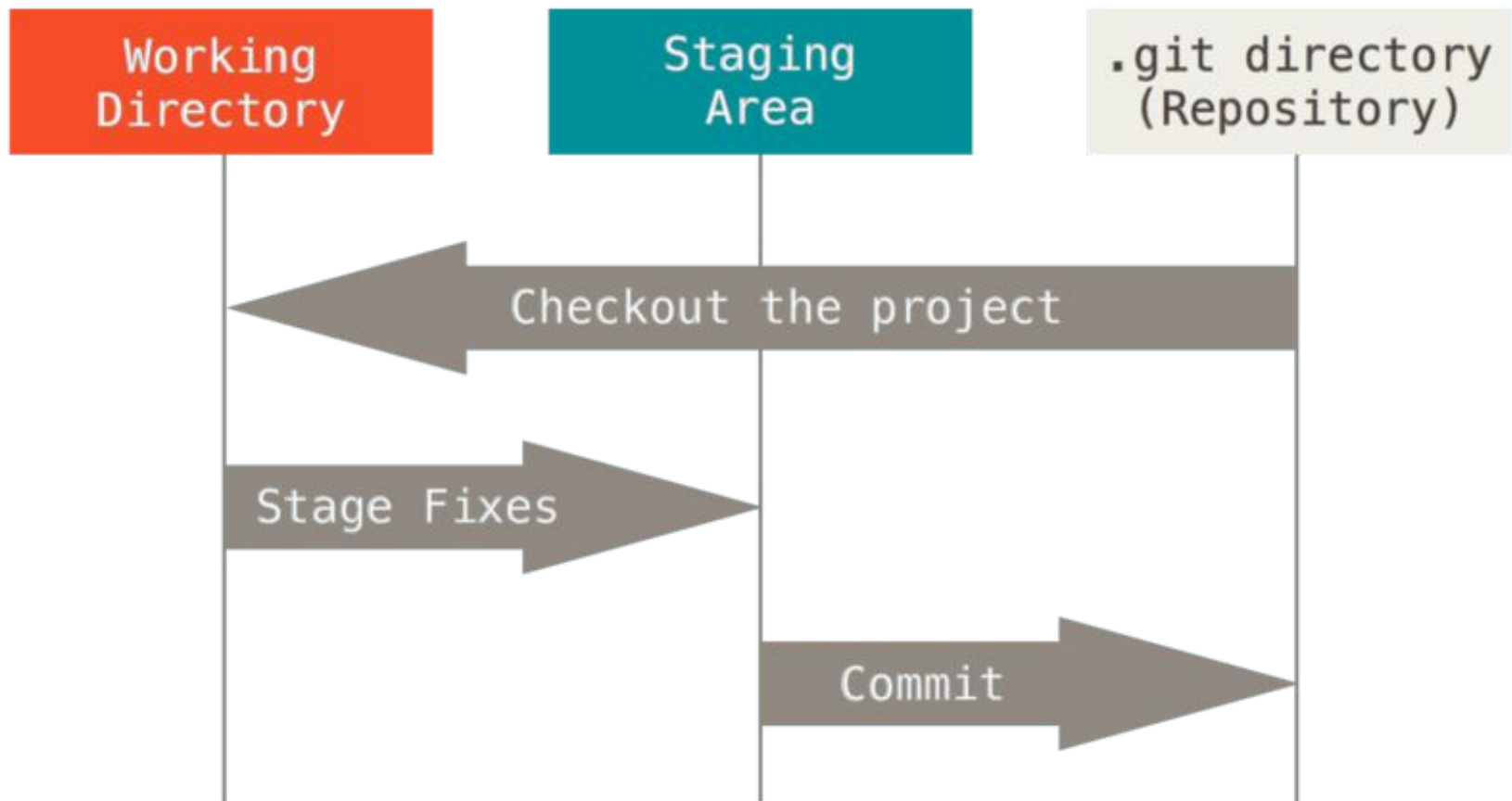


3. Fluxo de Estados

Imagens de Arquivos



Os 3 Estados do Git





4. Instalação e Configuração

Instalação



GitHub Desktop

<https://git-scm.com/download/win>



git --version

<https://git-scm.com/download/mac>



sudo apt-get
install git

<https://git-scm.com/download/linux>

Configuração

Identidade

```
$ git config --global user.name "Raul Duarte"
```

```
$ git config --global user.email raul.duarte@example.com
```

Editor

```
$ git config --global core.editor vscode
```

```
$ git config --global core.editor "'C:/ProgramFiles/Notepad++  
/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"
```

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



5. Noções Básicas

Acessando a Documentação

```
$ git help <verb>
```

```
$ git <verb> --help
```

```
$ man git-<verb>
```



Iniciando um Repositório

Diretório Existente

```
$ git init
```

```
$ git add *.c
```

```
$ git add LICENSE
```

```
$ git commit -m 'Initial project version'
```

Clonando um Repositório

```
$ git clone https://github.com/libgit2/libgit2 mylibgit
```

```
$ gh repo clone libgit2/libgit
```

```
$ git@github.com:libgit2/libgit.git
```

Os 4 principais Comandos

```
git add <file>
```

```
git commit -m '<message>'
```

```
git status
```

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



Visualizando Commits

```
$ git log
```

```
$ git log --stat
```

```
$ git log --pretty=oneline
```

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

```
$ git log --pretty=format:"%h %s" --graph
```

```
$ git log --since=2.weeks
```

Visualizando Mudança entre Commits

```
$ git diff
```

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

```
$ git diff --cached
```



O Arquivo .gitignore

O Git usa um arquivo oculto chamado .gitignore para indicar quais arquivos ou tipos de arquivos (exemplo: .pkl, .ipynb_checkpoints) não devem ser rastreados para o versionamento.



Modificando o Último Commit

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



Desfazendo o Último Commit

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



Comandos Úteis

Removendo Arquivos

```
$ git rm
```

```
$ git rm --cached
```

Mover ou Renomear

```
$ git mv
```



Usando Tags

```
$ git tag
```

```
$ git tag -a v1.4 -m "my version 1.4"
```

```
$ git show v1.4
```

```
$ git tag -a v1.2 9fceb02
```

```
$ git tag -l "v1.8.5*"
```

```
$ git tag -d v1.4-lw
```





6. Boas Práticas

Boas Práticas no Commit

Use o imperativo para commits

- ❖ “Adiciona feature” ao invés de “Adicionando feature” ou “Adicionada feature”.

A primeira linha deve ter no máximo 72 caracteres

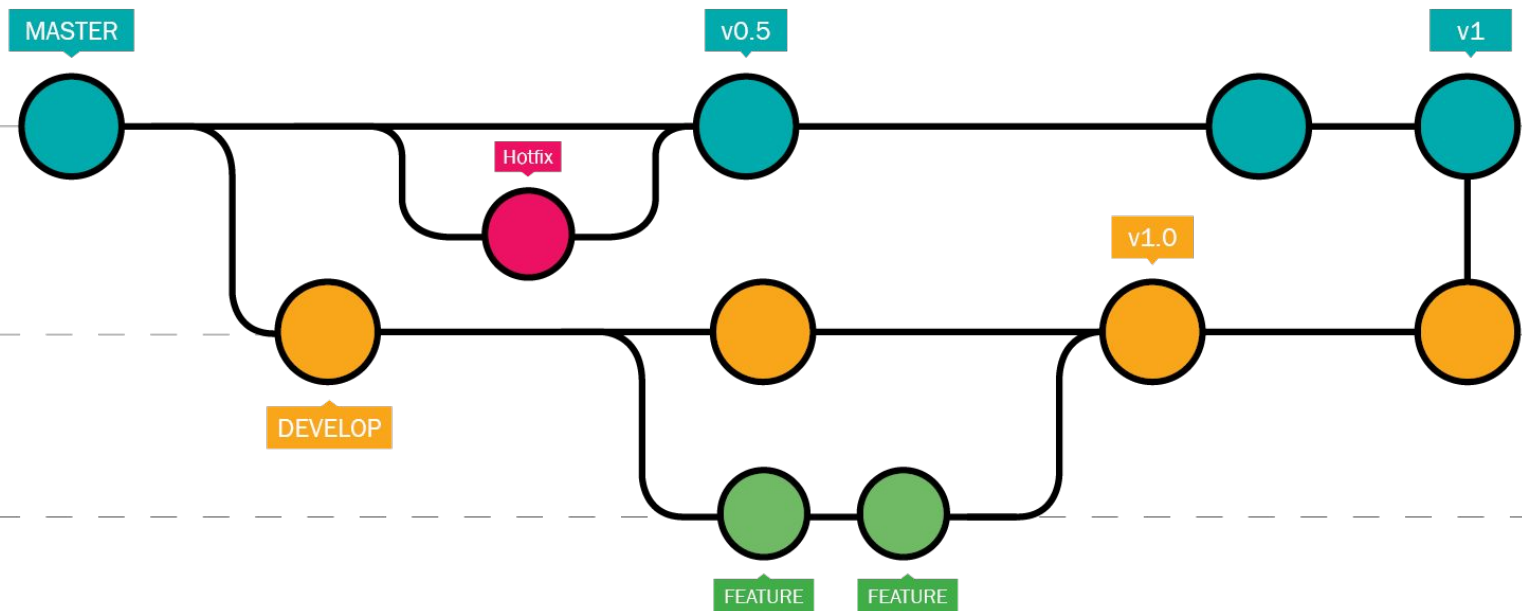
Descreva as alterações com detalhes no corpo do commit

Faça um commit no seu projeto apenas quando existirem alterações suficientemente relevantes

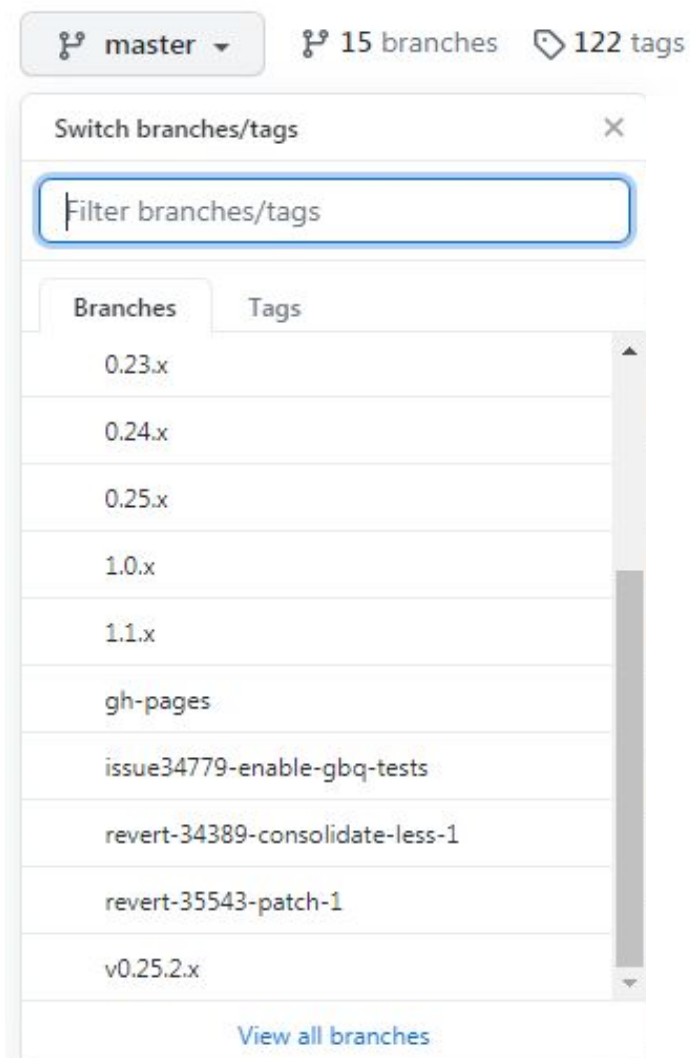


7. Branching

Branching



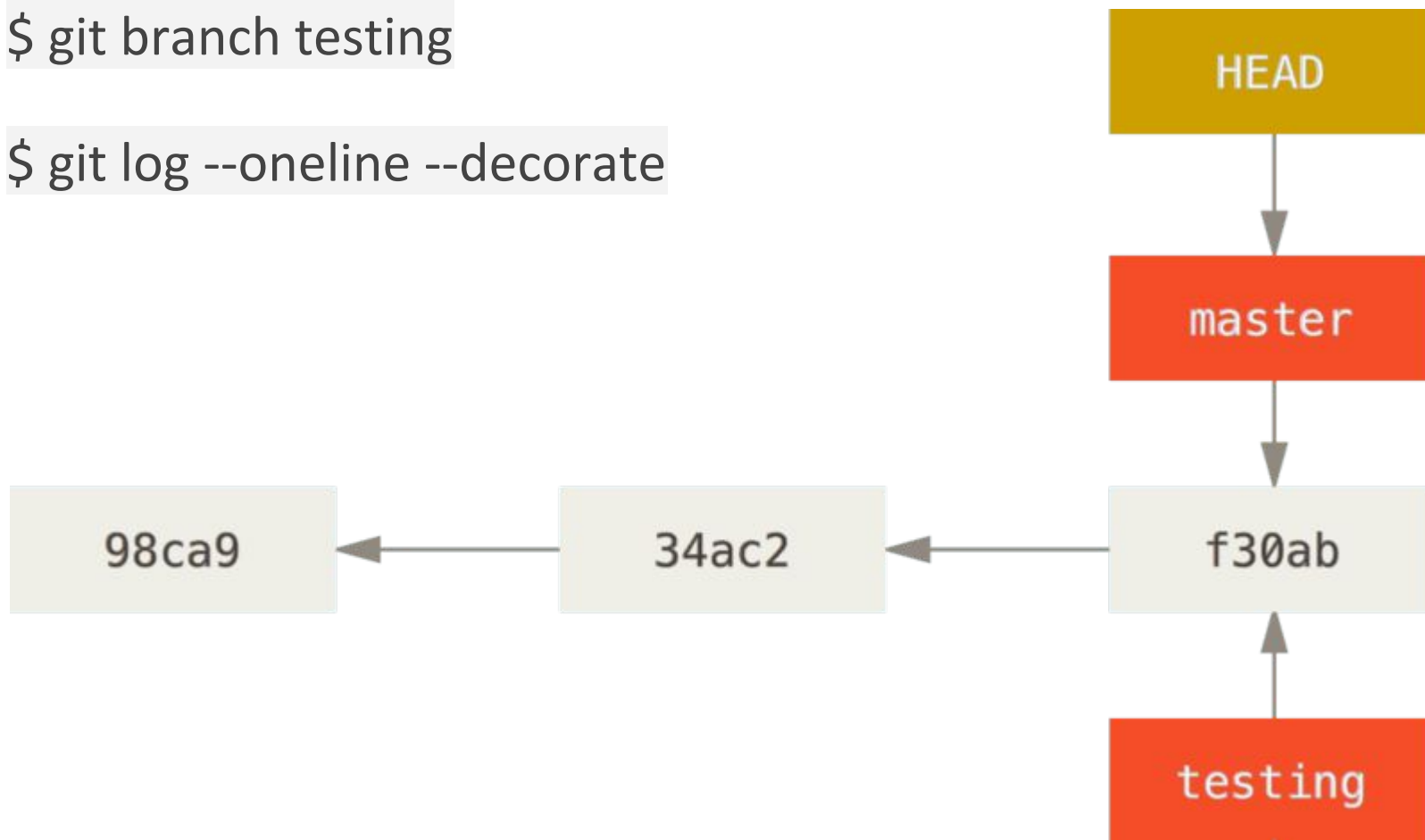
Branches do Pandas



Criando uma Branch

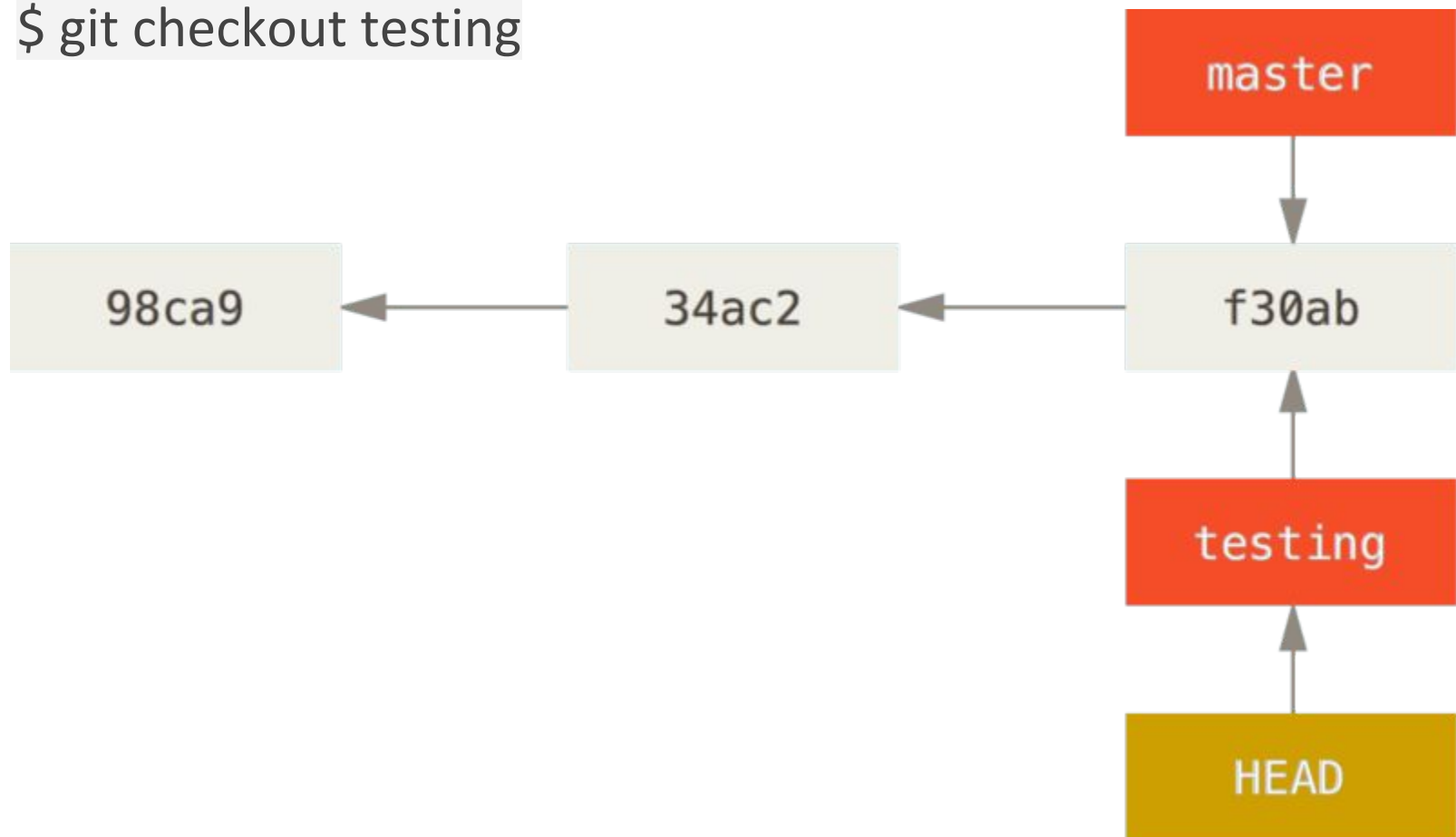
```
$ git branch testing
```

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



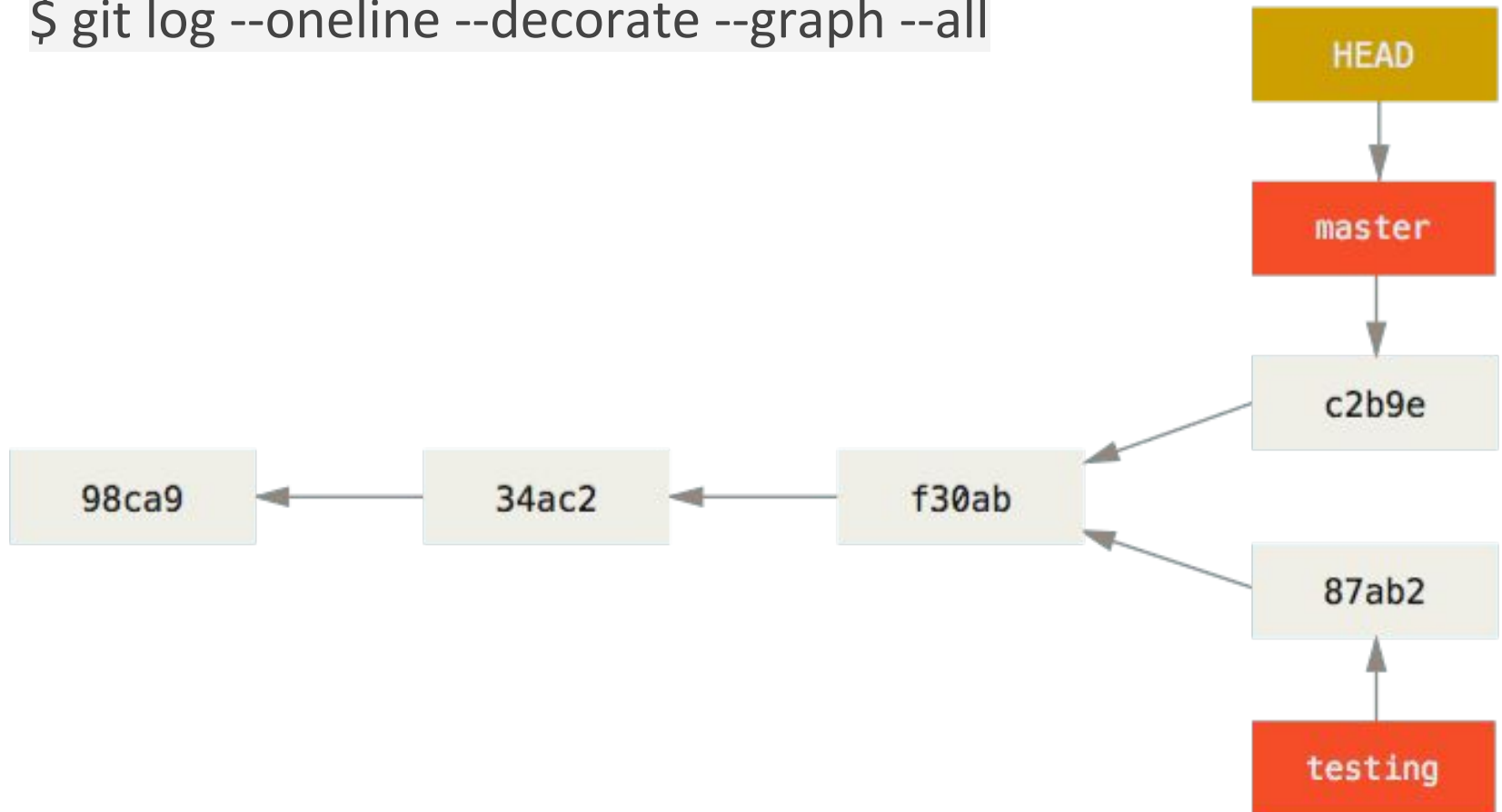
Alternando entre Branches

```
$ git checkout testing
```



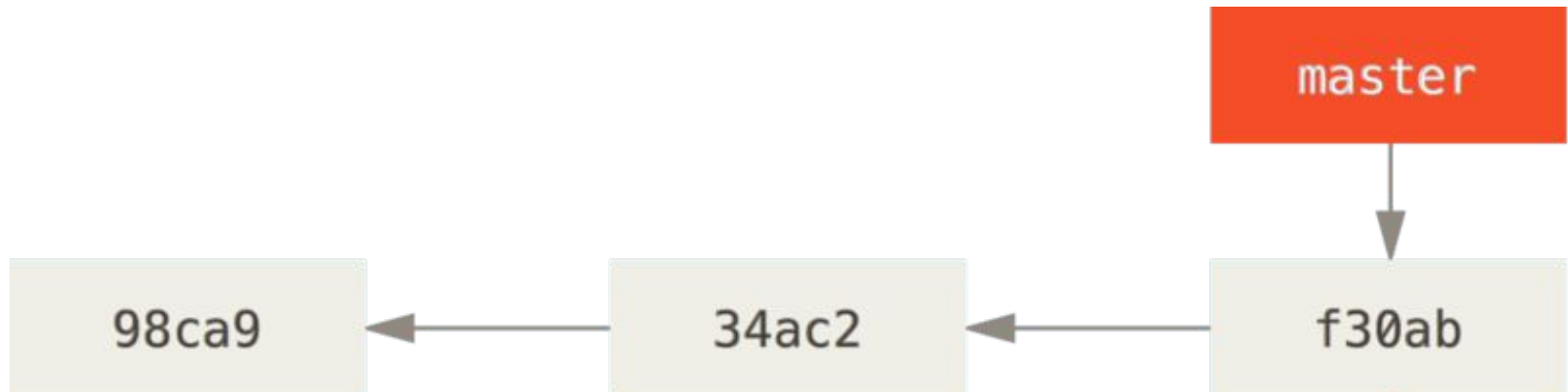
Visualizando as Ramificações

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

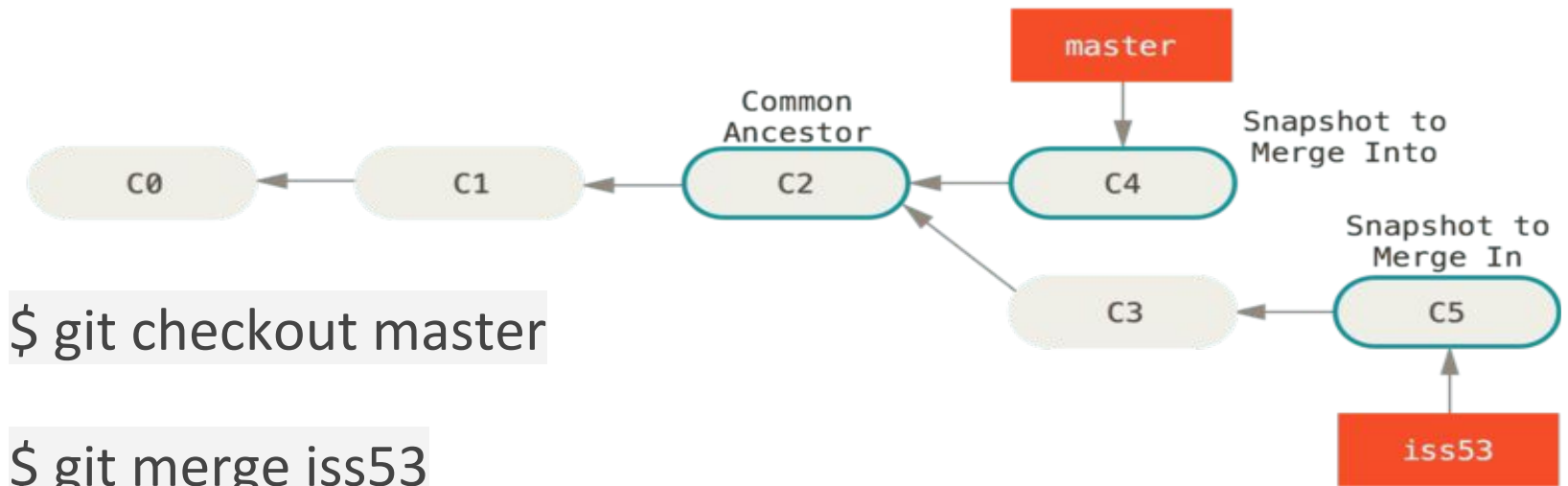


Removendo uma Branch

```
$ git branch -d testing
```



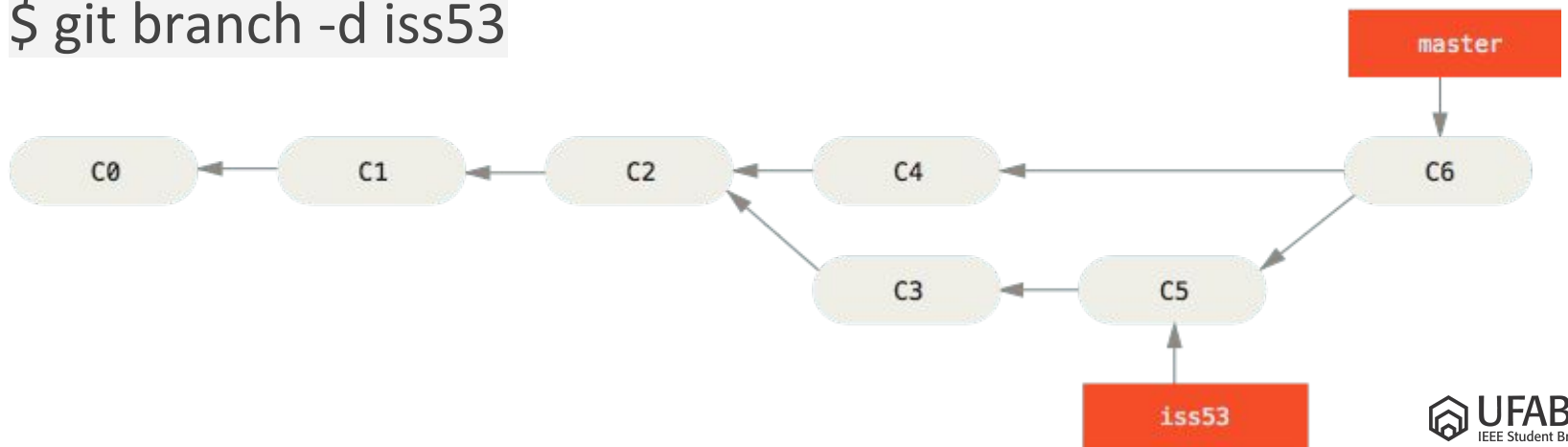
Merging



```
$ git checkout master
```

```
$ git merge iss53
```

```
$ git branch -d iss53
```



Conflitos

```
$ git merge iss53
```

Auto-merging index.html

CONFLICT (content): Merge conflict in index.html

Automatic merge failed; fix conflicts and then commit the result.

Identificando o Conflito

```
$ git status
```

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

Resolvendo o Conflito

```
$ sudo nano index.html
```

```
<<<<<<< HEAD:index.html
```

```
<div id="footer">contact : email.support@github.com</div>
```

```
=====
```

```
<div id="footer">
```

```
  please contact us at support@github.com
```

```
</div>
```

```
>>>>>>> iss53:index.html
```

Finalizando o Merging

```
$ git add index.html
```

```
$ git status
```

```
$ git commit
```



Comandos Úteis

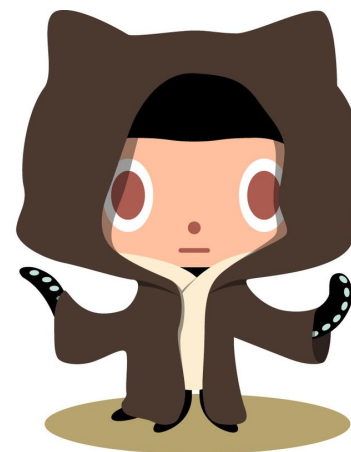
```
$ git branch
```

```
$ git branch -v
```

```
$ git branch --merged
```

```
$ git branch --no-merged
```

```
$ git checkout -b version2 v2.0.0
```





9. GitHub

GitHub

O GitHub é uma corporação multinacional americana que fornece hospedagem para desenvolvimento de software e controle de versão usando Git.



Configuração da Conta

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 50 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.


Criando um Repositório

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *

 RaulDuarte ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about [bookish-happiness](#)?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license


A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository


Fork

 Watch ▼

108

 Unstar

3.8k

 Fork

516




franz

Forked from meetfranz/franz

 Star

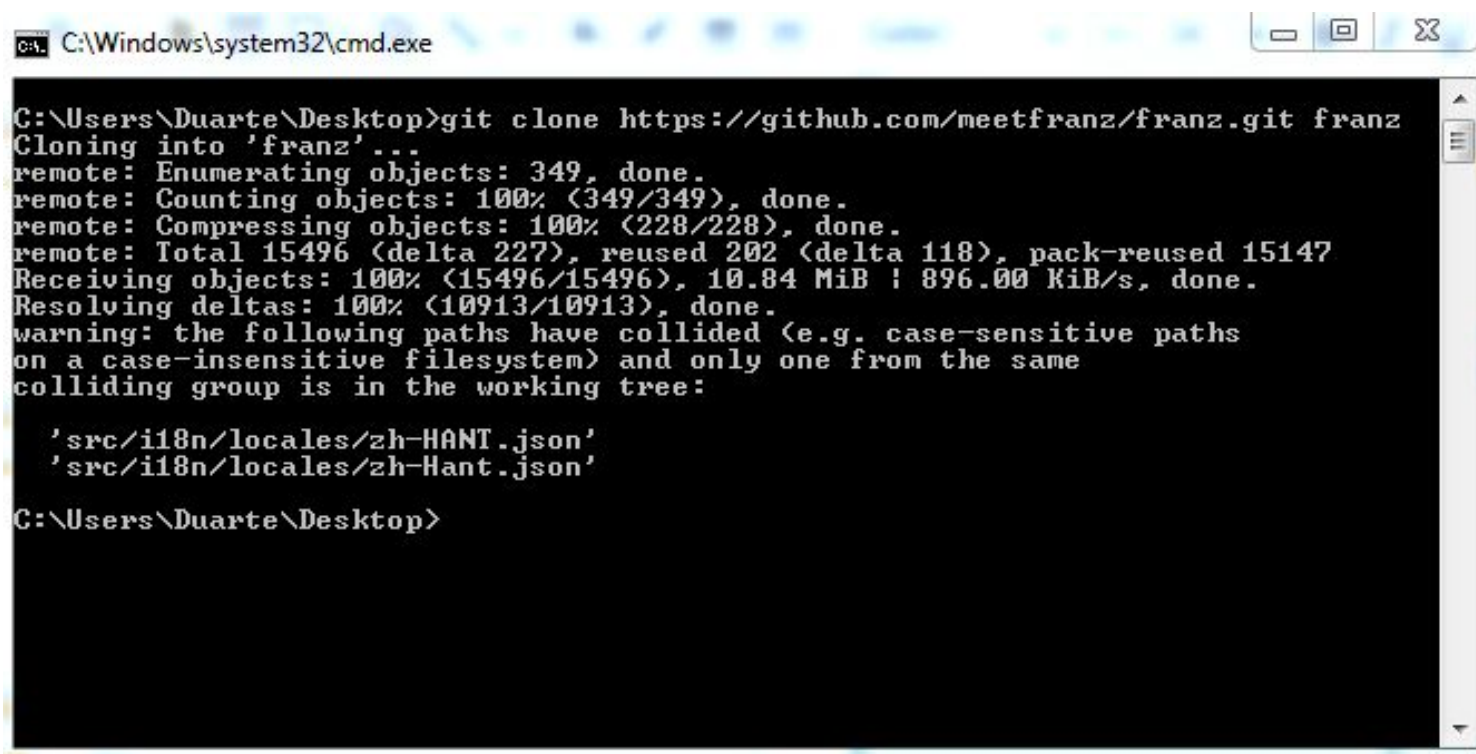
Franz is a free messaging app for services like WhatsApp, Slack, Messenger and many more.



 JavaScript  517  Apache License 2.0 Updated 2 days ago

Clonando um Repositório

```
$ git clone https://github.com/meetfranz/franz.git franz
```



```
C:\Windows\system32\cmd.exe

C:\Users\Duarte\Desktop>git clone https://github.com/meetfranz/franz.git franz
Cloning into 'franz'...
remote: Enumerating objects: 349, done.
remote: Counting objects: 100% (349/349), done.
remote: Compressing objects: 100% (228/228), done.
remote: Total 15496 (delta 227), reused 202 (delta 118), pack-reused 15147
Receiving objects: 100% (15496/15496), 10.84 MiB | 896.00 KiB/s, done.
Resolving deltas: 100% (10913/10913), done.
warning: the following paths have collided (e.g. case-sensitive paths
on a case-insensitive filesystem) and only one from the same
colliding group is in the working tree:

    'src/i18n/locales/zh-HANT.json'
    'src/i18n/locales/zh-Hant.json'

C:\Users\Duarte\Desktop>
```

Repositórios Remotos

```
$ git remote -v
```

```
$ git remote add
```

```
$ git remote show
```

```
$ git remote rename
```

```
$ git remote remove
```



Clonando Repositórios

```
$ git clone
```



Atualizando Repositório Remoto

```
$ git push origin <branch>
```

Branches

```
$ git push --set-upstream origin
```

```
<corrected-branch-name>
```

```
$ git push origin --delete <bad-branch-name>
```

Tags

```
$ git push origin <tag>
```

```
$ git push origin --tags
```

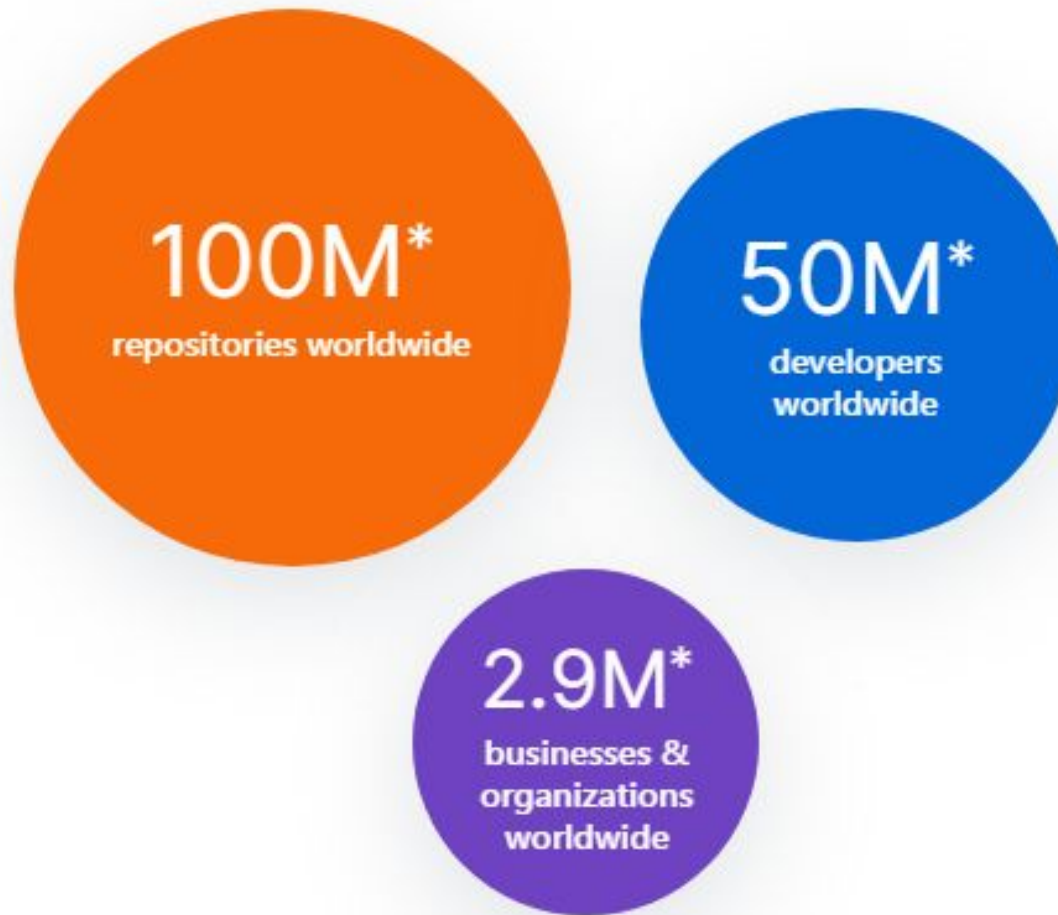
Atualizando Repositório Local

```
$ git pull
```

```
$ git fetch
```



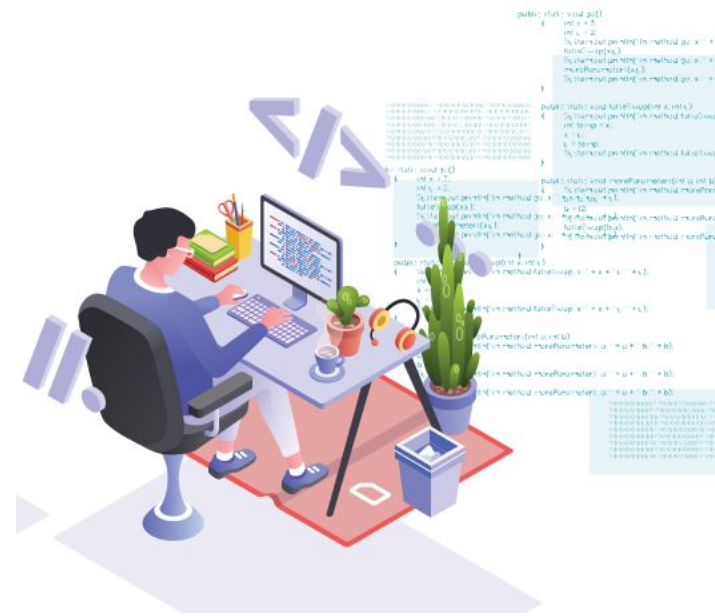
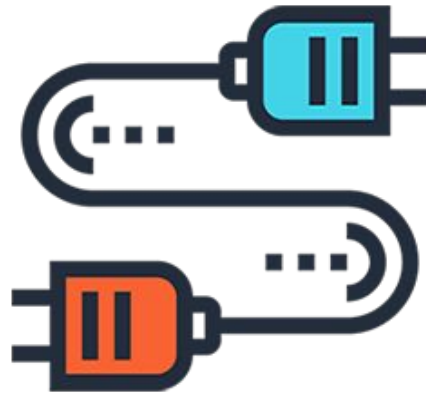
GitHub





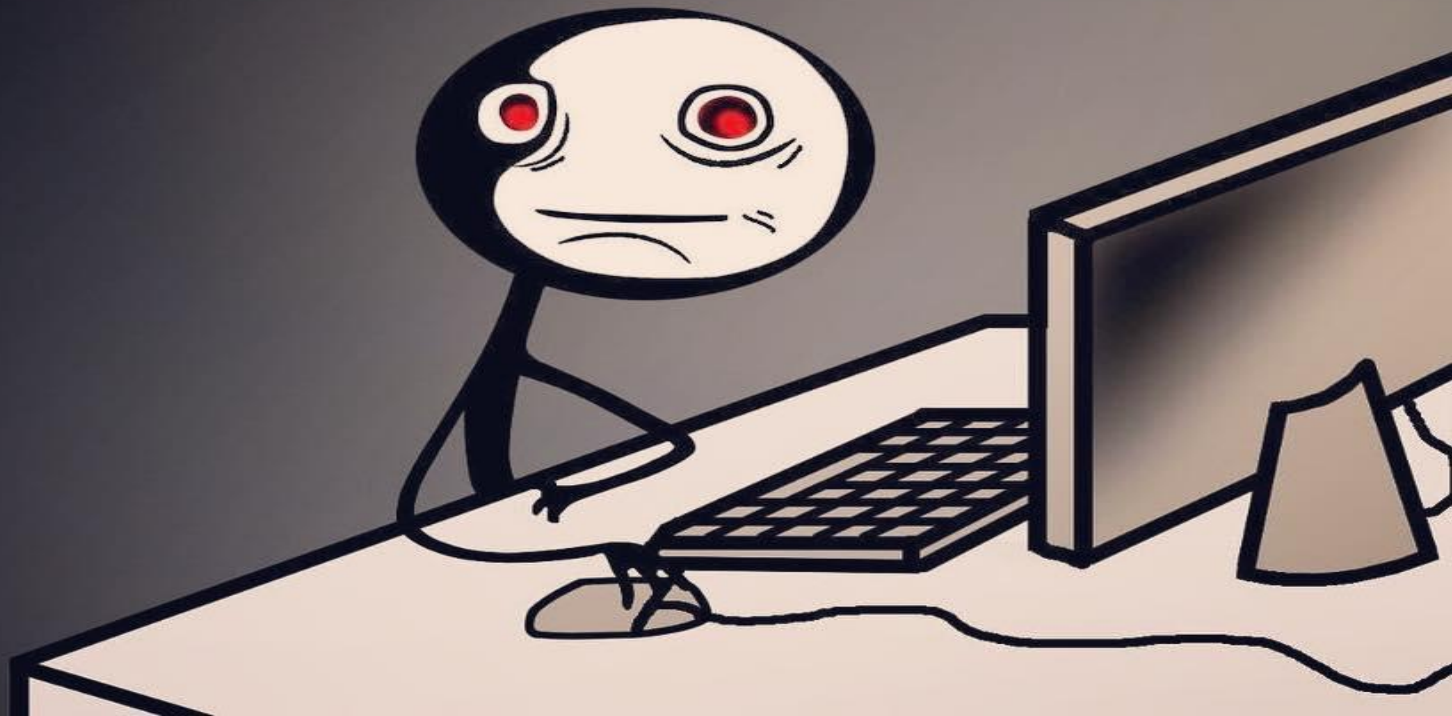
10. Integrações

Integrações



Programador

Mamífero noturno de olhos vermelhos. Capaz de conversar com objetos inanimados e converter café em código.





11. Projeto Dirigido

Projeto Dirigido

Objetivo:

Simular o desenvolvimento de um projeto em ambiente colaborativo através do Git/Github.

Repositório:

<https://github.com/raulduarte/treinamento-git-github>

Template:

<https://templatemo.com/tm-539-simple-house>

Comunicação:

Salas do Discord

❖ Massas

- Galeria
- Artigo

❖ Saladas

- Galeria
- Artigo

❖ Doces

- Galeria
- Artigo

❖ Sobre \ Home

Tarefas

1. Entender o projeto;
2. Montar as equipes;
3. Criar as branches;
4. Construir o projeto;
5. Fazer o push das alterações;
6. Fazer o merge das branches;
7. Fazer o pull do projeto finalizado.



12. Tirando Dúvidas

Palestrante



Nome

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sed libero eget erat ultrices commodo. In accumsan eget augue sit amet.

Palestrantes



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sed libero eget erat ultrices commodo. In accumsan eget augue sit amet tempus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sed libero eget erat ultrices commodo. In accumsan eget augue sit amet tempus.

