

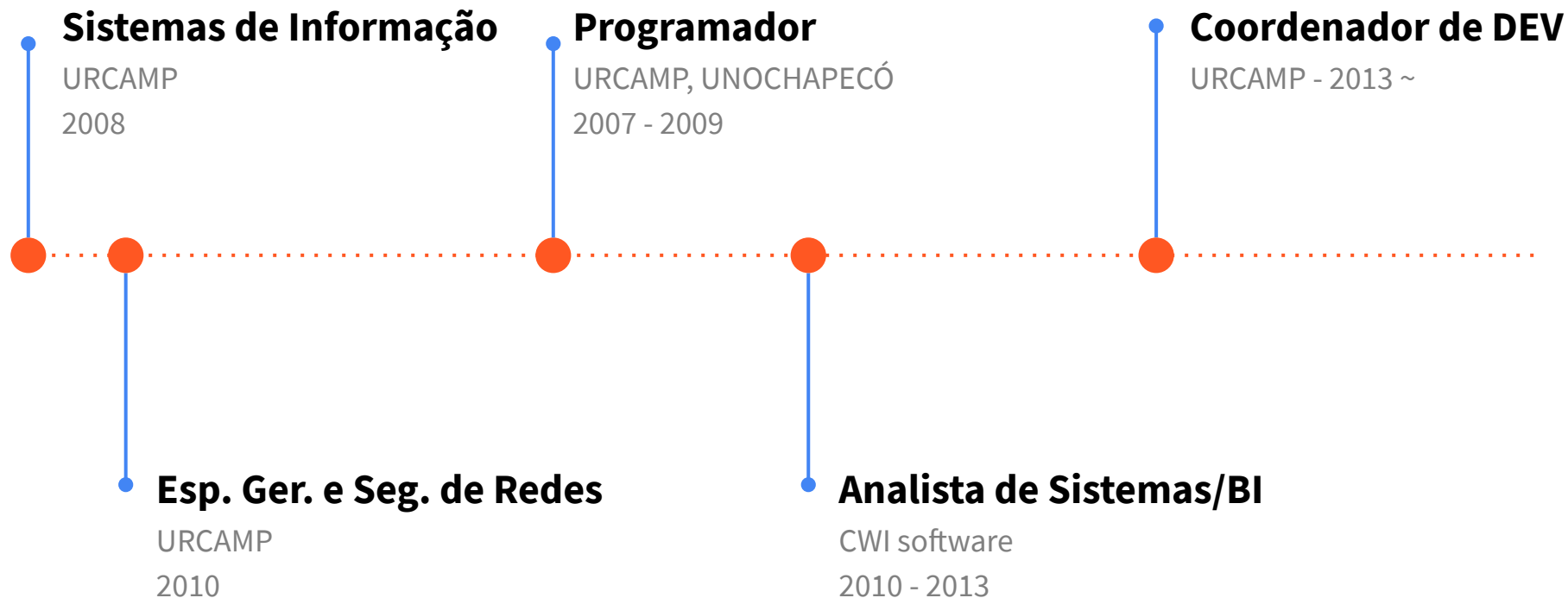
Versionamento de whatever com



git

Fernando Fagonde

Who am I?



Curriculum

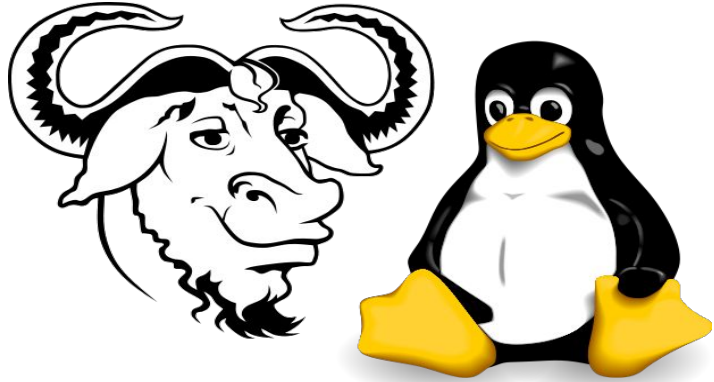


ODEBRECHT

Curriculum

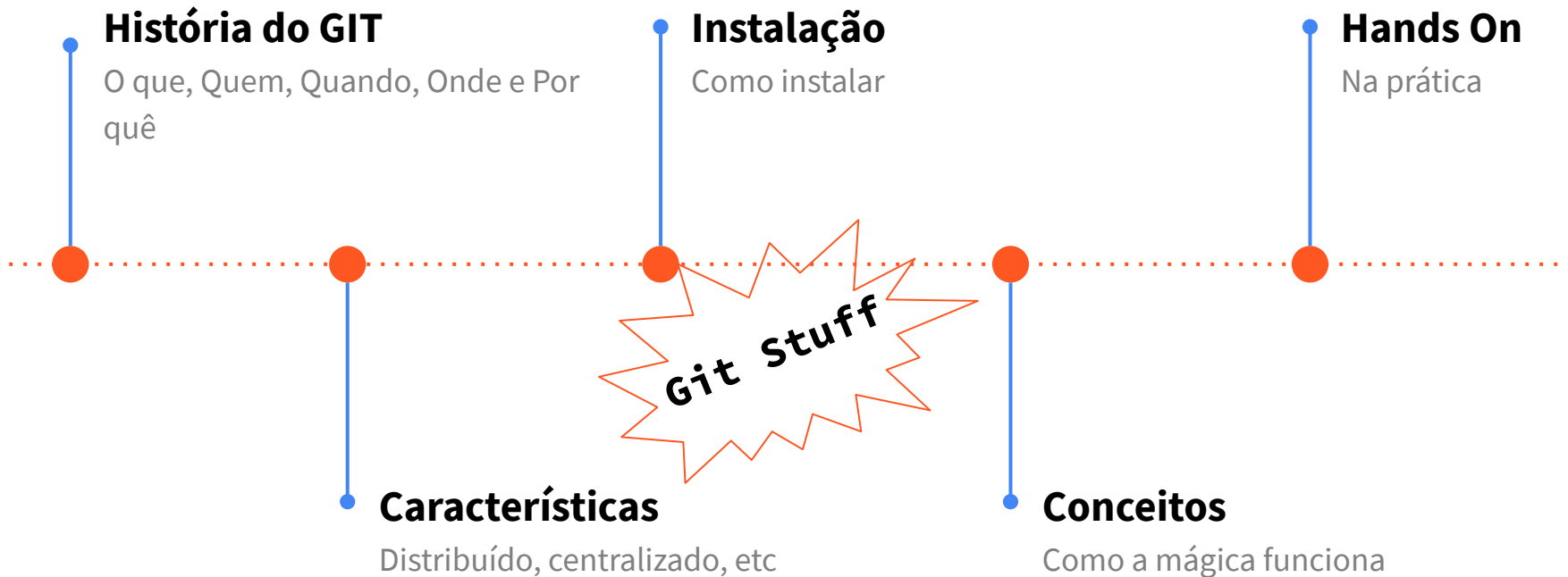


Experiência em




PostgreSQL


RoadMap





Sistema de Controle de Versão


 artigo.odt


 artigo (3ª cópia).odt


 ARTIGO_AGORA_VAI.odt


 artigo_final.odt

 artigo_final (outra cópia).odt


 artigo_final_valendo.odt


 artigo_final_valendo_de_verdade.odt


 artigo_final_valendo_de_verdade (outra cópia).odt


 artigo (outra cópia).odt


 artigo.odt


 artigo (3ª cópia).odt


 ARTIGO_AGORA_VAI.odt


 artigo_final.odt

 artigo_final (outra cópia).odt

 artigo_final_valendo.odt

 artigo_final_valendo_de_verdade.odt

 artigo_final_valendo_de_verdade (outra cópia).odt

 artigo (outra cópia).odt

Git Stuff

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.





GitHub



GitLab



Aplicações com muitos recursos que não existiriam sem o git.

Quem fez isso?



História do Git

Linus Torvalds

- Versionamento do Kernel do Linux de 1991 a 2002 era feito com arquivos compactados;
- Bitkeeper;

História do Git

Linus Torvalds

- Velocidade;
 - Design simples;
 - 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);
-

About

Documentation

Downloads

[GUI Clients](#)

[Logos](#)

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads



Mac OS X



Windows



Linux/Unix

Older [releases](#) are available and the [Git source repository](#) is on GitHub.



GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients](#) →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos](#) →

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads



Mac OS X



Windows



Linux/Unix

Older [releases](#) are available and the [Git source repository](#) is on GitHub.



GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients](#) →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos](#) →

```
fernando@fagonde:~/Downloads$ sudo apt install git
[sudo] senha para fernando:
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
git is already the newest version (1:2.11.0-3+deb9u3).
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
fernando@fagonde:~/Downloads$ █
```

Quem usa?

Companies & Projects Using Git

Google

facebook

Microsoft

twitter

LinkedIn

NETFLIX

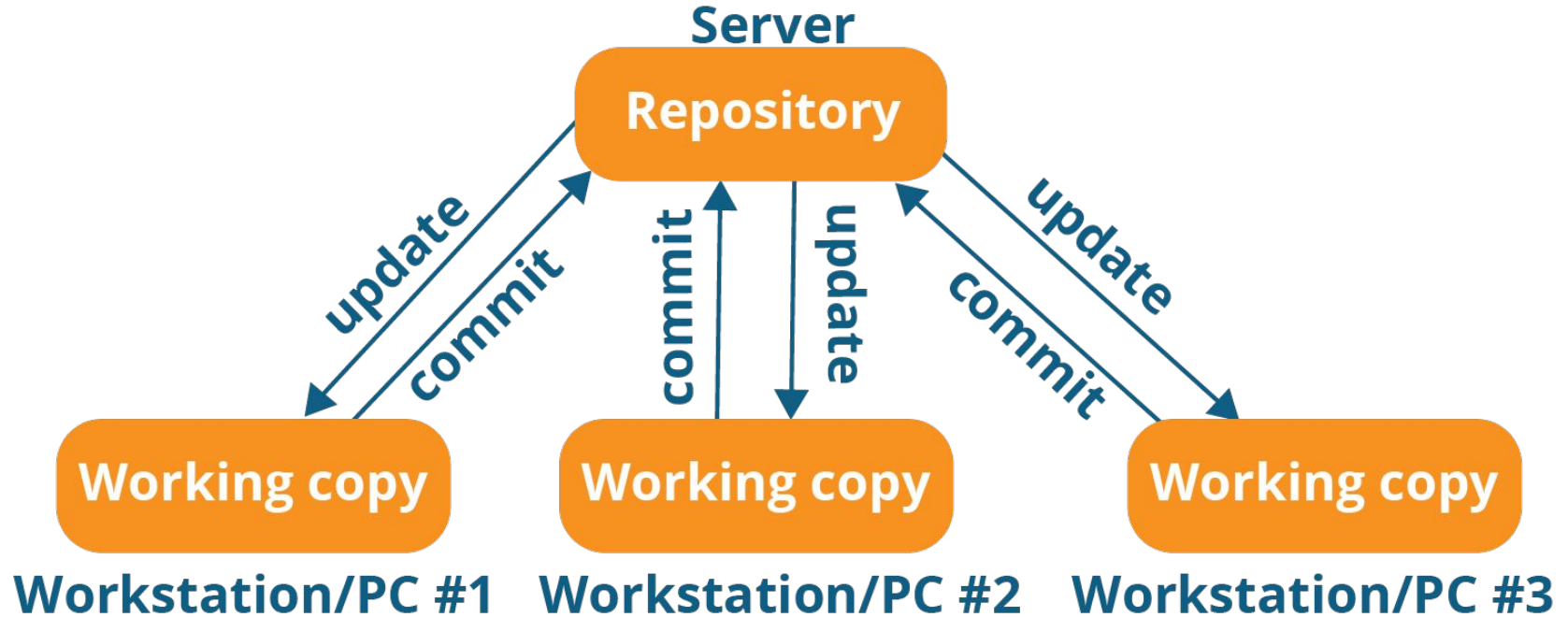


PostgreSQL

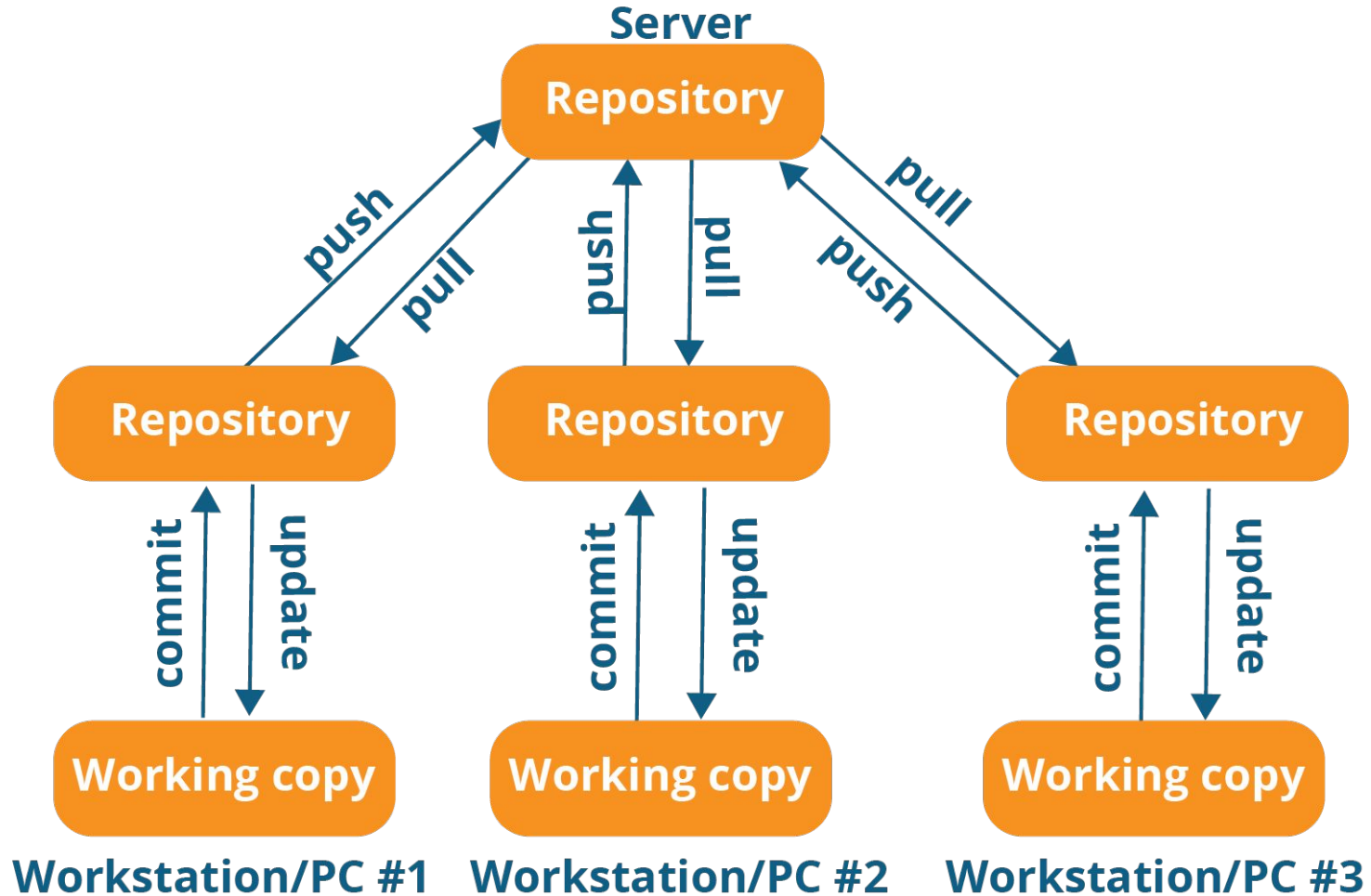


**Distribuído e
Centralizado**

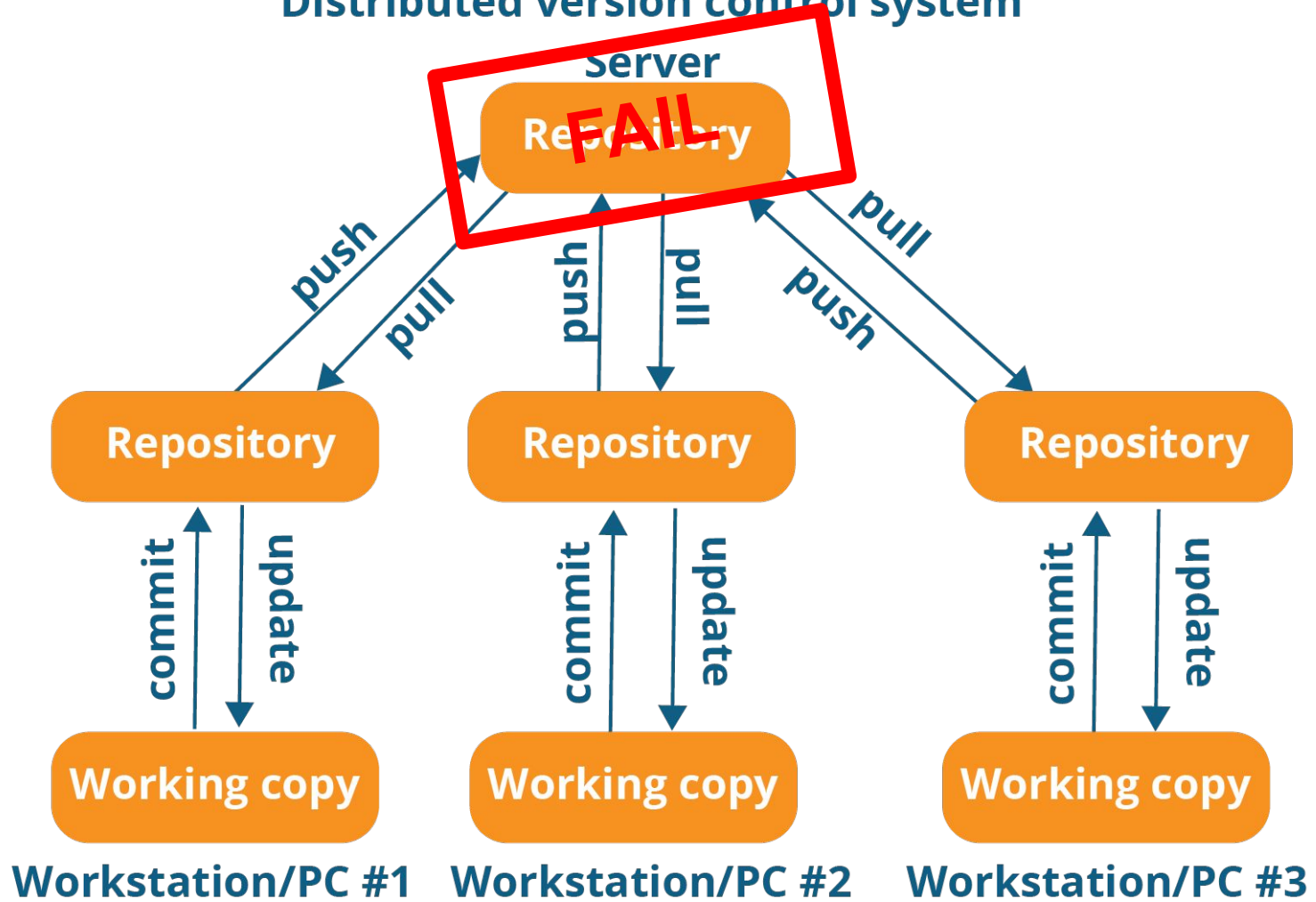
Centralized version control system



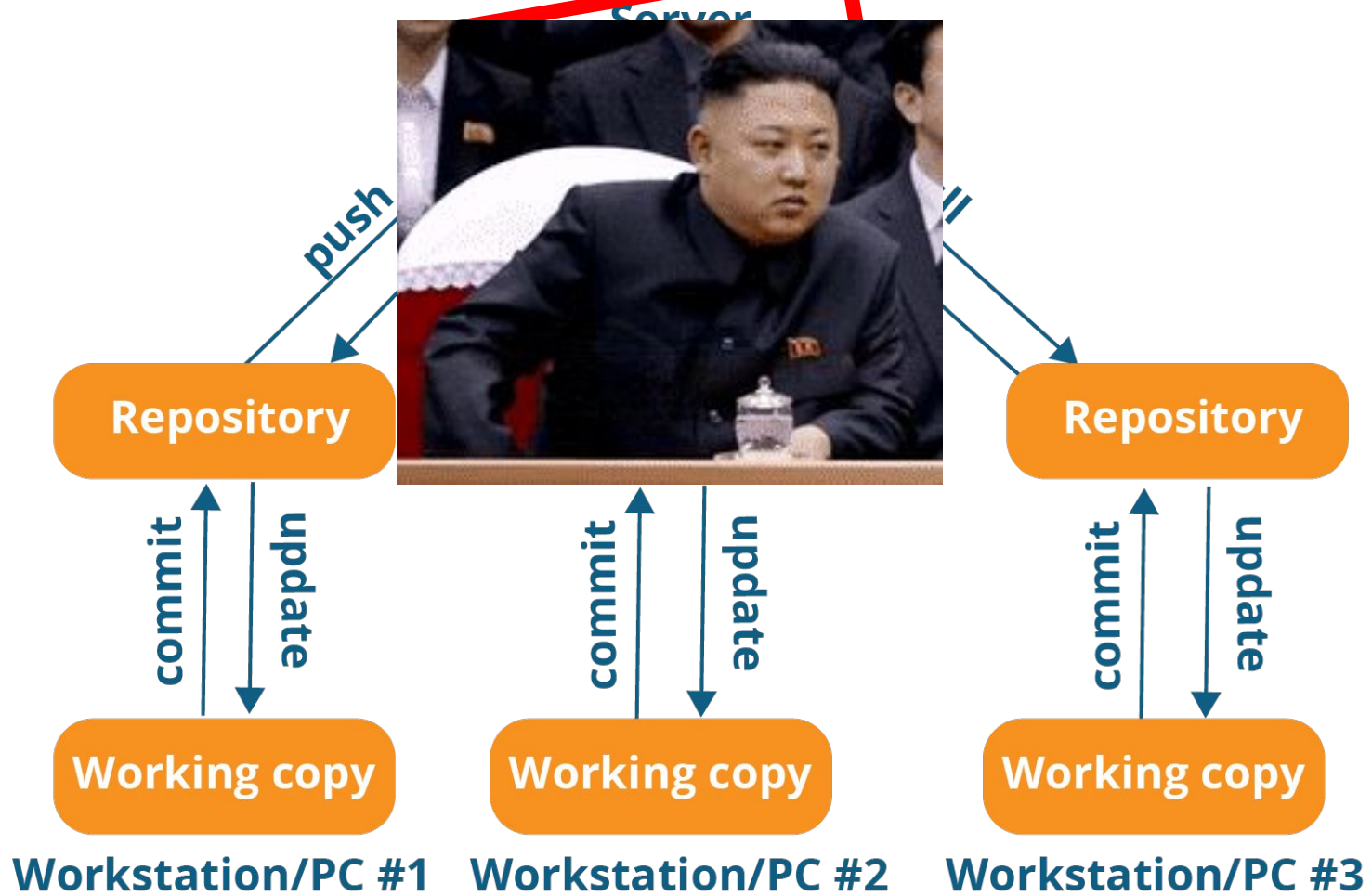
Distributed version control system



Distributed version control system



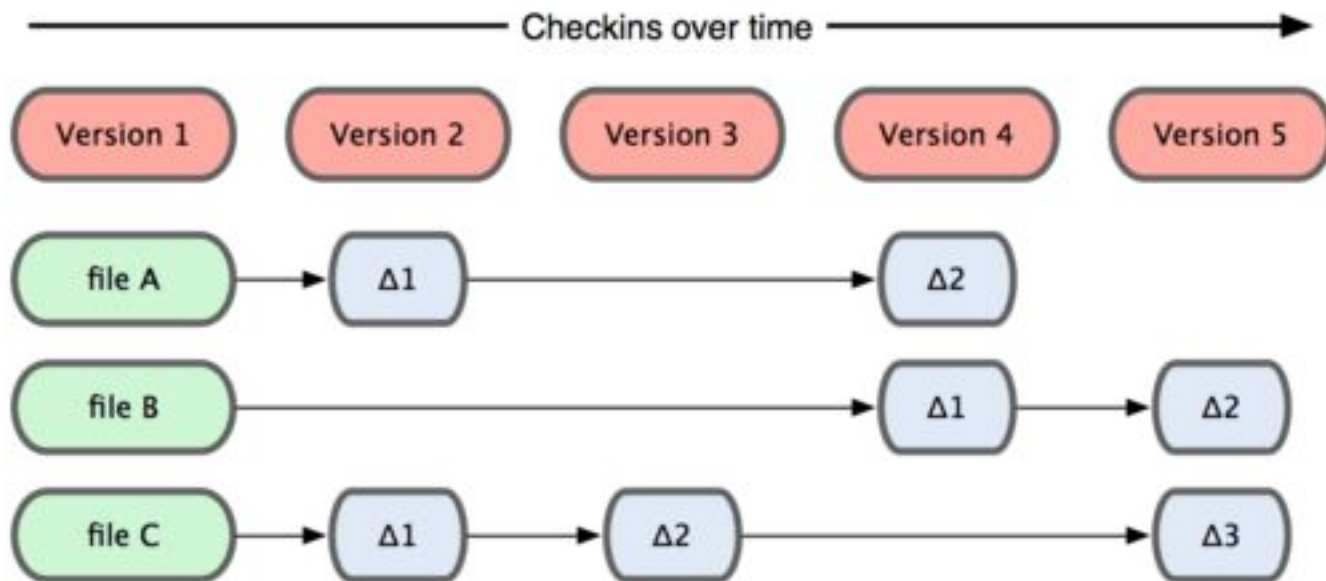
Distributed version control system



Basics

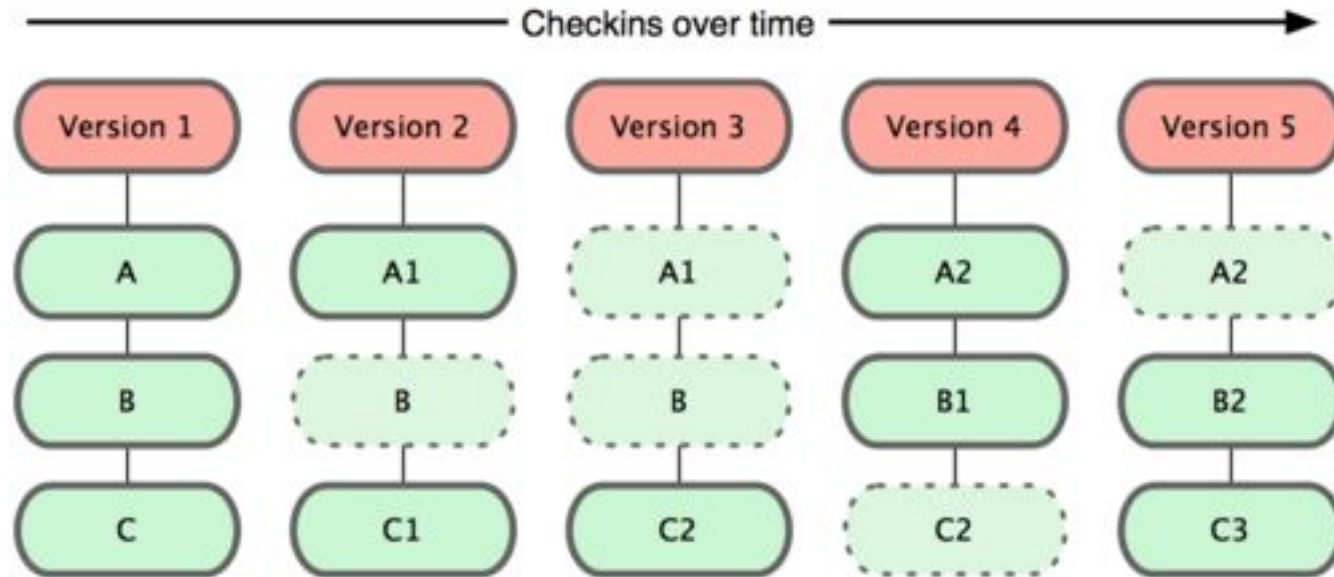
- Snapshots e não Diferenças
- 3 estados
- Branchs
 - origin master
- remote
- tag

Snapshot



Diferenças (não git)

Diferenças



Snapshot (git)

COMUNICADO

IMPORTANTE

3 estados

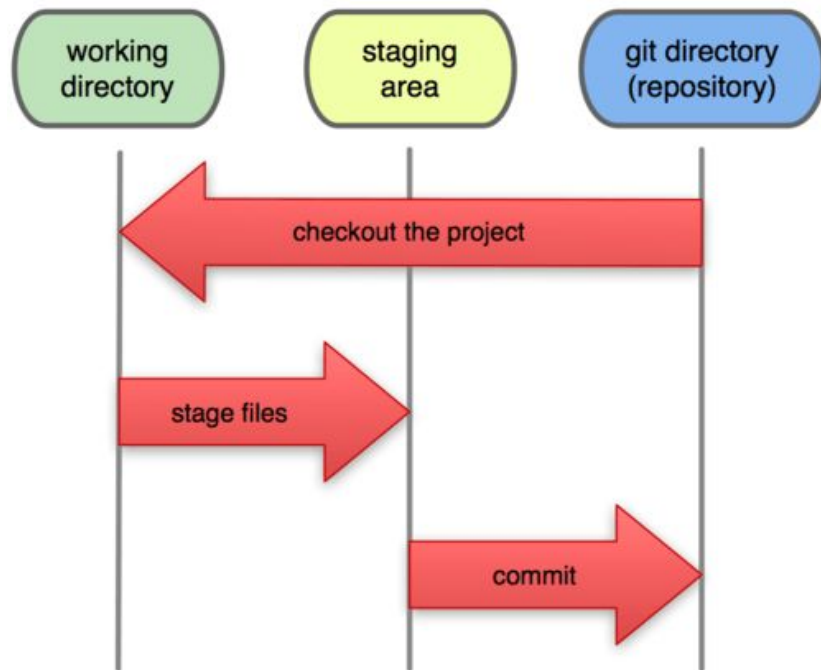
3 estados



3 estados

- Consolidado (committed)
- Modificado (modified)
- Preparado (staged)

Local Operations



O workflow básico do Git pode ser descrito assim:

1. Você modifica arquivos no seu diretório de trabalho.
2. Você seleciona os arquivos, adicionando snapshots deles para sua área de preparação.
3. Você faz um commit, que leva os arquivos como eles estão na sua área de preparação e os armazena permanentemente no seu diretório

—Git

Branches

BRANCHS

Convenção

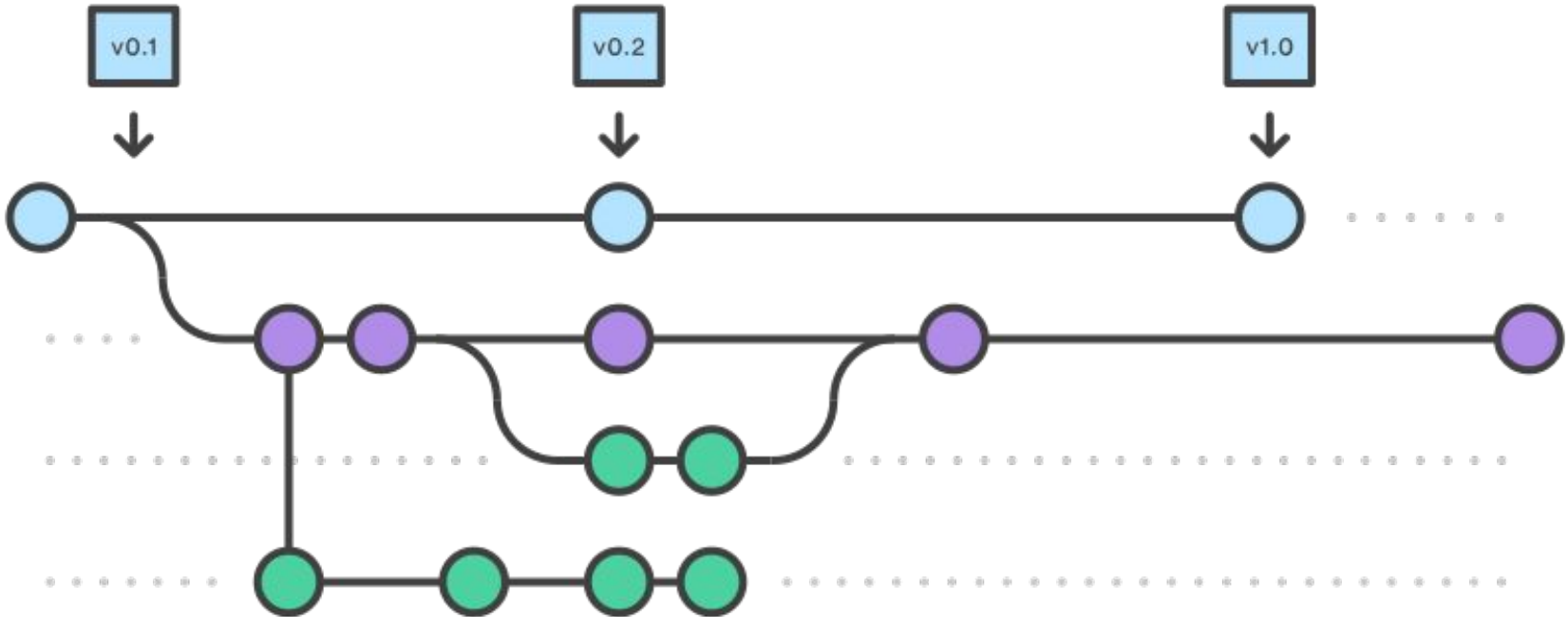


Master

Develop

Feature

Feature



Remote

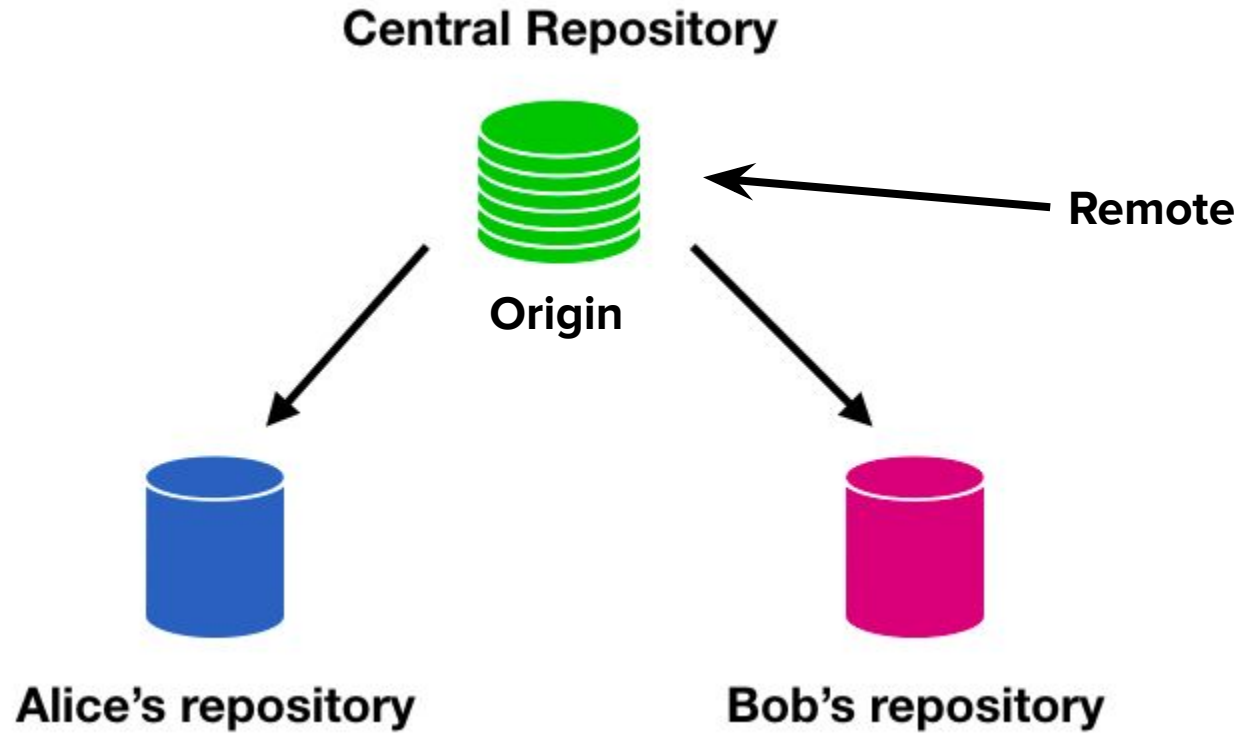
Central Repository

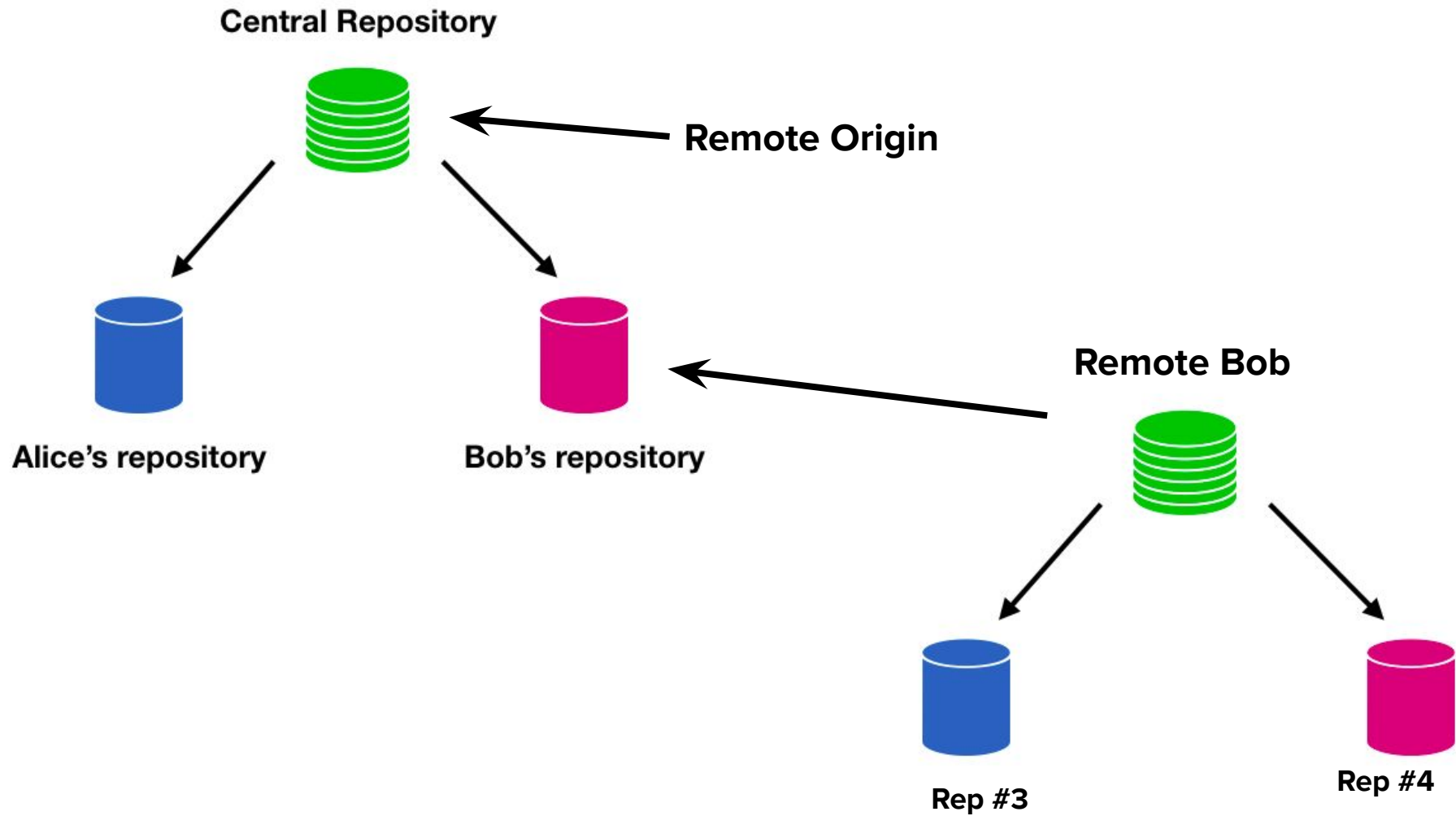


Alice's repository

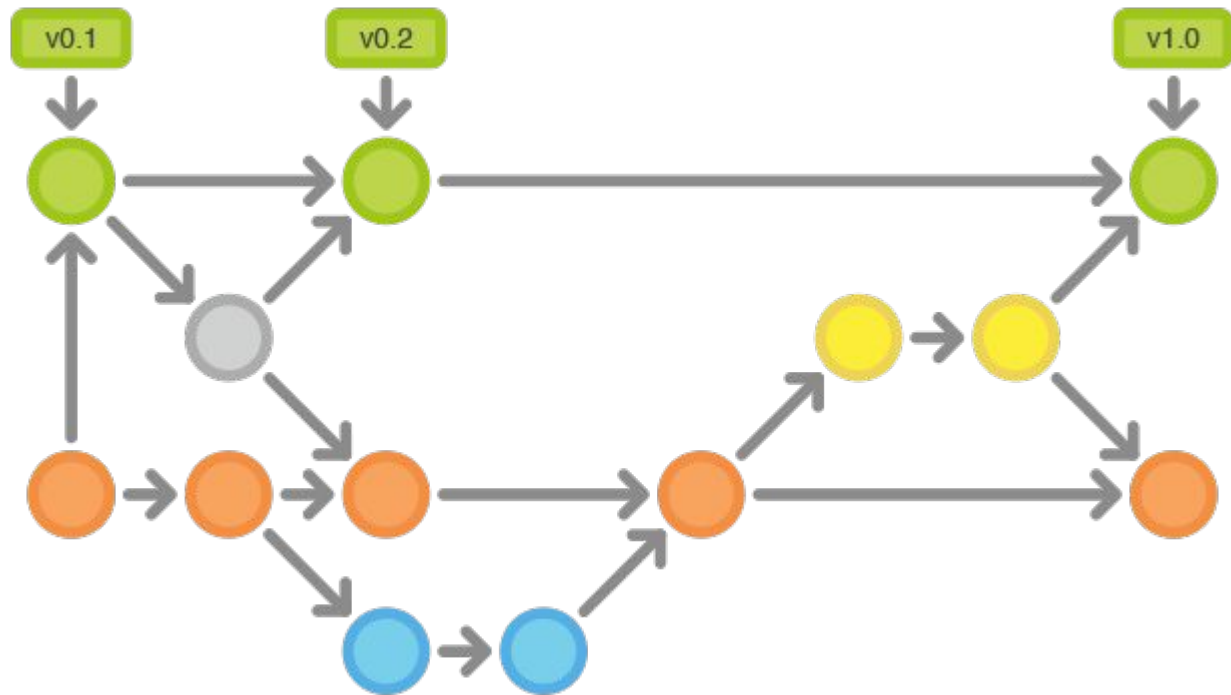


Bob's repository





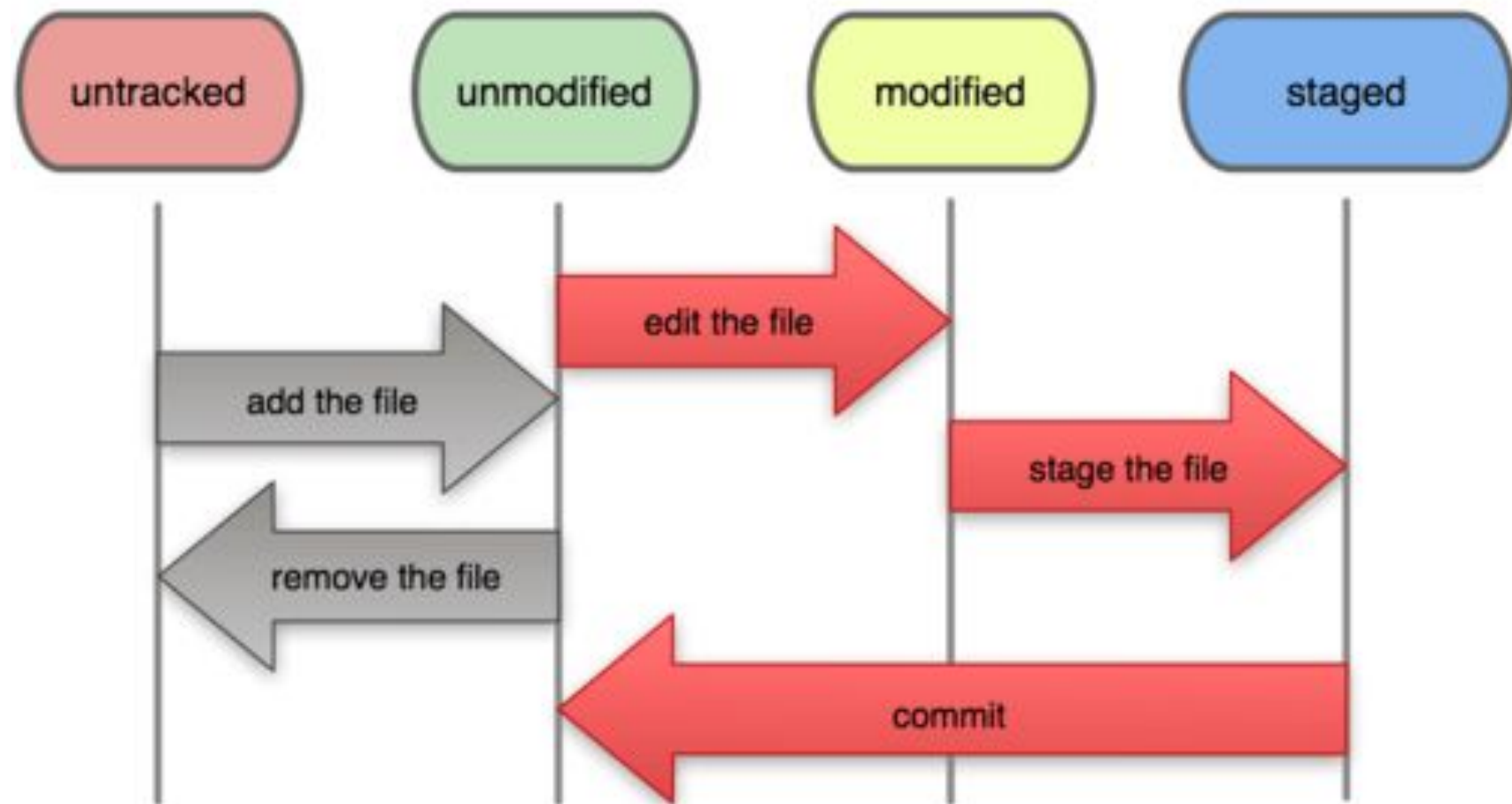
Tags





MÃO NA MASSA

File Status Lifecycle



Mão na massa

- File Status lifecycle
- Inicializando um Repositório em um Diretório Existente
- Clonando um Repositório Existente
- Gravando Alterações no Repositório
 - Verificando o Status de
___ Seus Arquivos

Mão na massa

- Monitorando Novos Arquivos
- Selecionando Arquivos Modificados
- Ignorando Arquivos
- Visualizando Suas Mudanças Seleccionadas e Não Seleccionadas
- Fazendo Commit de Suas Mudanças

Mão na massa

- Removendo arquivos
 - Movendo Arquivos
 - Visualizando o Histórico de Commits
 - Desfazendo Coisas
 - Modificando Seu Último Commit
 - Desfazendo um Arquivo Modificado checkout
-

Mão na massa

- Exibindo Seus Remotos
 - Adicionando Repositórios Remotos
 - Fazendo o Fetch e Pull de Seus Remotos
 - Pushing Para Seus Remotos
 - Removendo e Renomeando Remotos
-

Inicializando um Repositório em um Diretório Existente

- `git init`

Clonando um Repositório Existente

- git clone

**Gravando
Alterações no
Repositório**

Verificando o Status de Seus Arquivos

- `git status`

Monitorando* Novos Arquivos

*Dizer pro GIT cuidar desse arquivo

- criar um arquivo
 - git status
 - git add
 - git status
-

Selecionando Arquivos Modificados

- Alterar um arquivo
- `git status`
- `git add`
- `git status`
- altera de novo
- `git status`

—

Ignorando Arquivos

- .gitignore

Visualizando Suas Mudanças Selecionadas e Não Selecionadas

- git diff

Fazendo Commit de Suas Mudanças

- `git commit`
- `git commit -m`

Removendo Arquivos

- `git rm`

**Desfazendo
Coisas**

Fazendo Commit de Suas Mudanças

- git t

Removendo Arquivos

- `git rm`

Movendo Arquivos

- `mv README.txt README`
- `git rm README.txt`
- `git add README`

Visualizando o Histórico de Commits

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

Modificando Seu Último Commit

- `git commit --amend`

Desfazendo um Arquivo Modificado

- `git checkout --`

**Trabalhando
com Remotos**



Quase acabando...

Exibindo Seus Remotos

- `git remote`
- `git remote -v`

Adicionando Repositórios Remotos

- `git clone`
- `git remote add`

Fazendo o Fetch e Pull de Seus Remotos

- `git fetch`
 `[nome-remoto]`
- `git pull origin`
 `master`

Pushing Para Seus Remotos

- `git push`
 `[nome-remoto]`
 `[branch]`

Removendo e Renomeando Remotos

- `git remote rm [nome-remoto]`
- `git remote rename [old-name] [new-name]`

Outras coisas

- GUI;
- gitflow;
- bash;

- _ (ツ) _ / -

Dúvidas

<https://git-scm.com/book/pt-br/v1/>



@fernandofagonde



fernandofagonde@urcamp.edu.br



Muito
Obrigado!