



CENTRO DE TECNOLOGIA DA INFORMAÇÃO
DESENVOLVIMENTO WEB BÁSICO

Versionamento de códigos

Geucimar Brilhador
geucimar@up.edu.br

Atualizado em 04/2021

Redes sociais



<https://linkedin.com/in/gilbriatore>

GitHub

<https://github.com/gilbriatore>



<https://youtube.com/gilbriatore>



<https://gitlab.com/gilbriatore>

O que é e para que serve um sistema de versionamento?

- Um sistema de controle de versões (ou versionamento) é um software que tem a finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer. Esses sistemas são comumente utilizados no desenvolvimento de software para controlar as diferentes versões — histórico e desenvolvimento — dos códigos-fontes e também da documentação.

Principais benefícios

- Segurança;
- Trabalho em equipe;
- Controle de histórico e rastreabilidade;
- Marcação e resgate de versões estáveis;
- Ramificação de projeto.

O que é o Git?

- Git é um software de versionamento de linha de comando usado para criar versões de códigos ou documentos e voltar versões anteriores facilmente, assim como analisar o histórico do que já foi feito.

CURSO DE CONTROLE DE VERSÃO

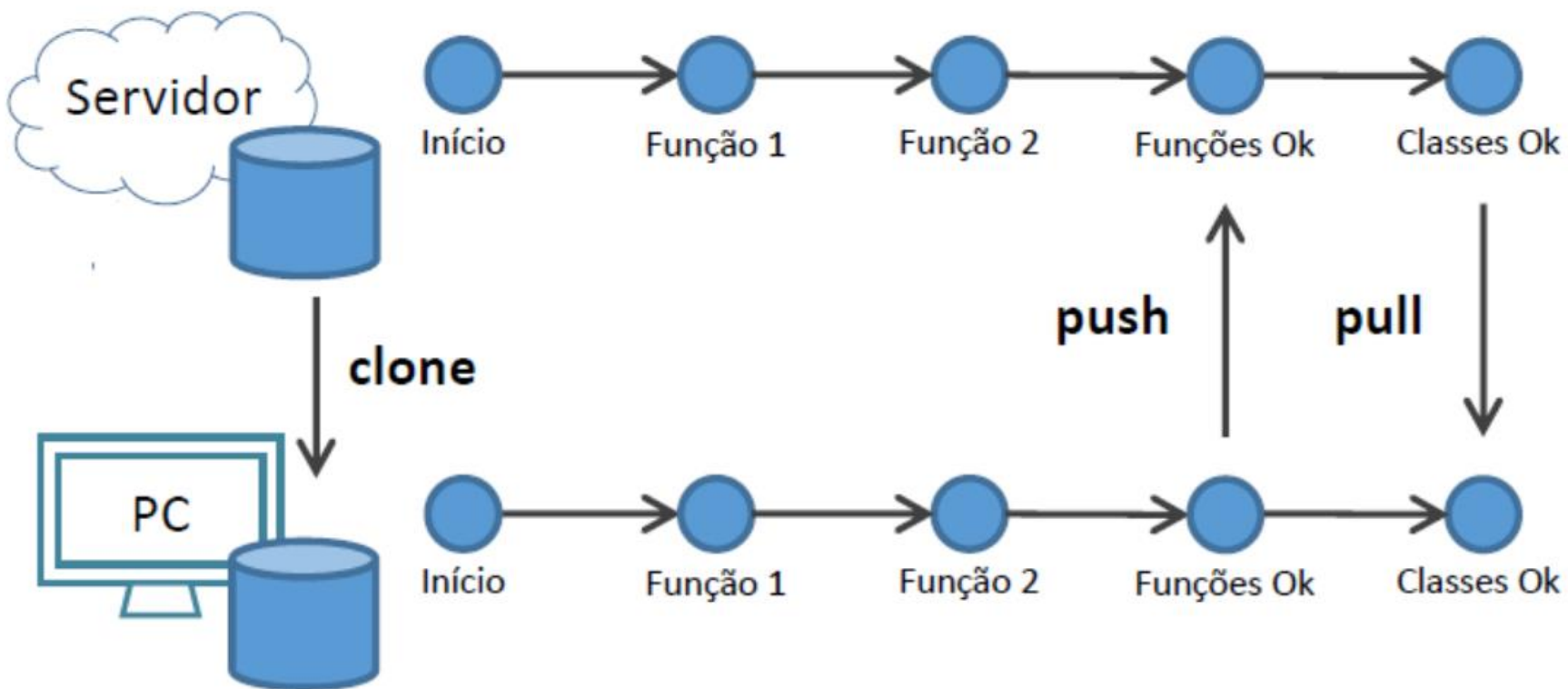


<https://tinyurl.com/9bm3yrpr>

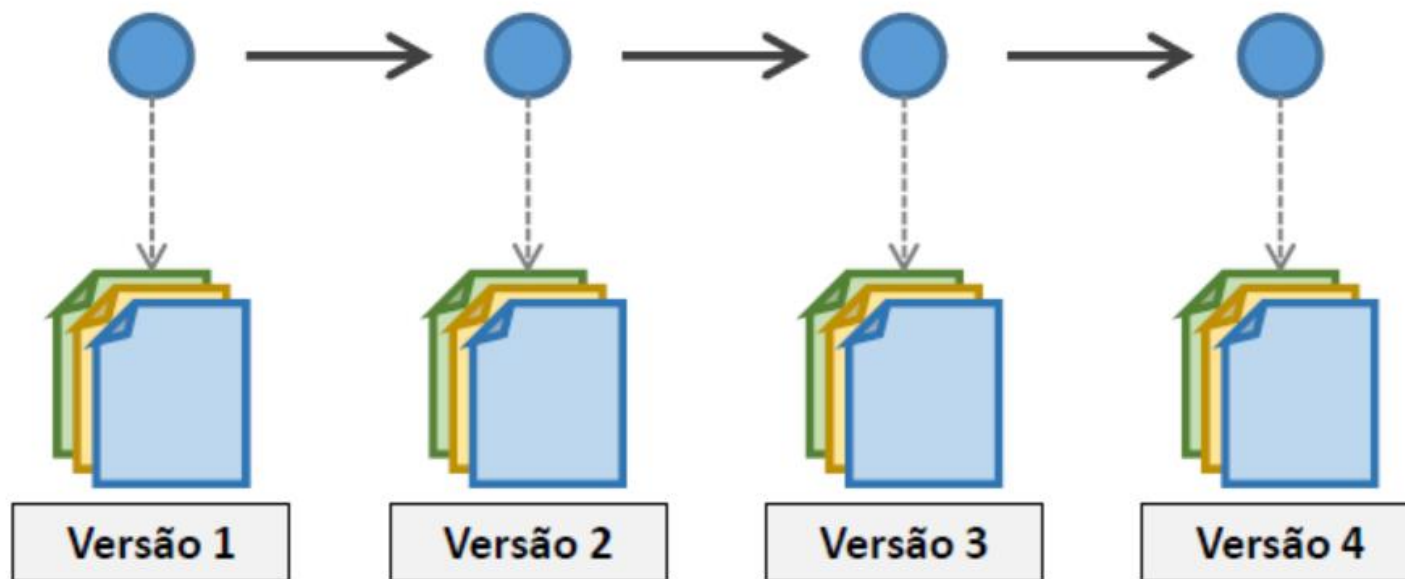
Servidores de hospedagem de códigos...



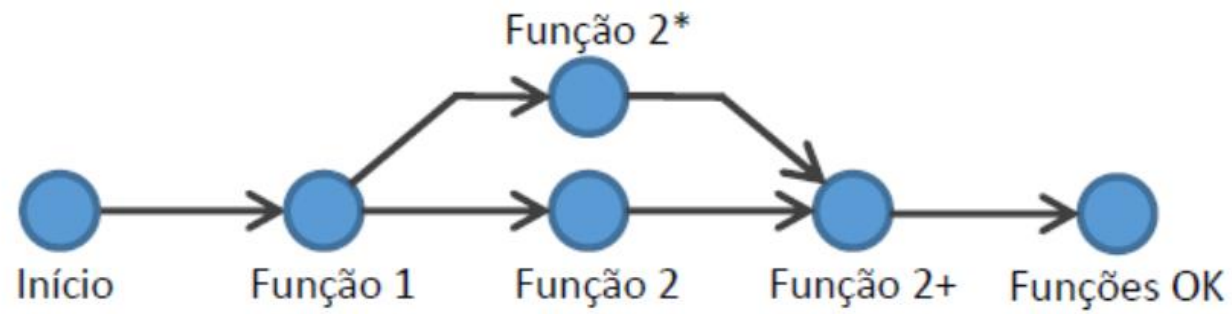
Operações



Snapshots



Branchs



Instalação (<https://git-scm.com>)



Search entire site...

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



macOS



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 →](#)

Configuração da credencial

```
git config --global user.name "nome de usuario"
```

```
git config --global user.email "meu_email@exemplo.com"
```

1. Clonar repositório remoto, codificar e versionar

```
git clone https://gitlab.com/xxxx.git
```

```
git add .
```

```
git commit -m "Atualização..."
```

```
git push
```

2. Codificar, adicionar repositório remoto e versionar

```
git init
```

```
git config --global core.autocrlf true
```

```
git remote add origin https://gitlab.com/xxxx.git
```

```
git pull origin master --allow-unrelated-histories
```

```
git add .
```

```
git commit -m "Versão inicial"
```

```
git push --set-upstream origin master
```

3. Codificar, criar repositório local e versionar (**gitlab** > private)

```
git init
```

```
git add .
```

```
git commit -m "Versão inicial"
```

```
git push --set-upstream https://gitlab.com/xxxx.git master
```

3. Codificar, criar repositório local e versionar (github)

```
git init
```

```
git add .
```

```
git commit -m "Versão inicial"
```

```
gh repo create
```

```
git push -u origin master
```