

Controle de Versões com Git

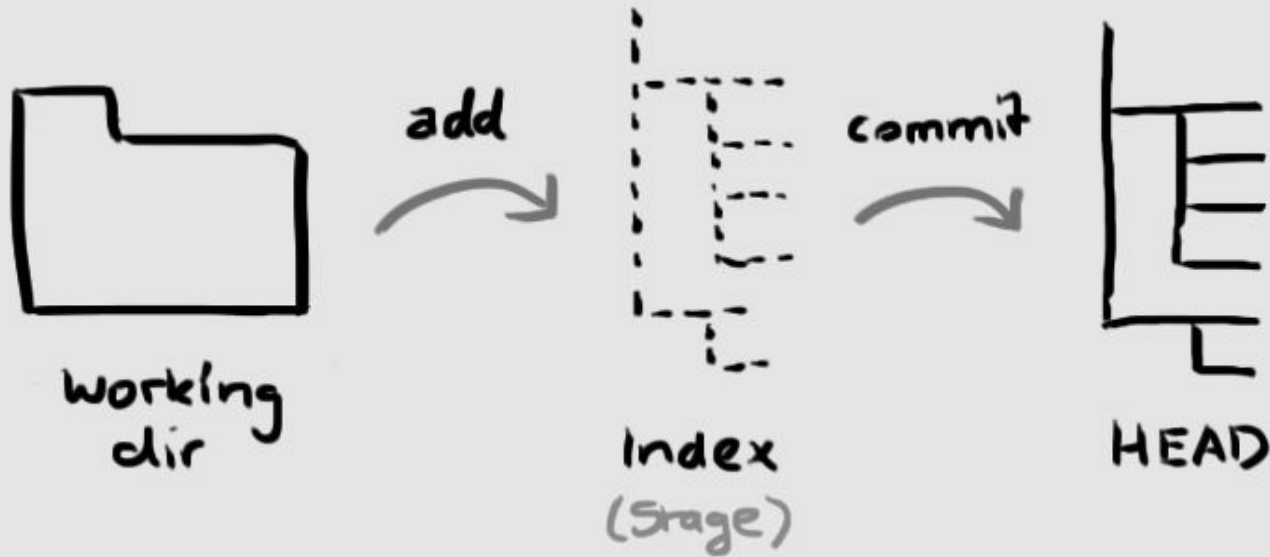
Prof. Carlos Fran

Gitlab / Github: @carlosfran

Git

- Sistema de controle de versões distribuído
- Criado por Linus Torvald (criador do kernel Linux)

Fluxo de trabalho



Working Dir

[1]

```
$ git clone user@server:repository
```

```
$ git clone https://gitlab.com/carlosfran/hellogit.git
```

[2]

```
$ git init
```

Index

```
$ git add <arquivos>
```

```
$ git add .
```

```
$ git add *
```

```
$ git add novo_arquivo.txt
```

HEAD

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

```
$ git commit -m "Adicionado ícones no menu"
```

Enviando versão

[1]

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

```
$ git push origin master
```

[2]

```
$ git remote add origin <servidor>
```

Prática 1 - Hello

1. Crie um repositório

2. Clonar repositório

\$ git clone https://gitlab.com/username/repository.git

3. Fazer alterações

\$ git add *

\$ git commit -m "Adicionado o username @carlosfran"

\$ git push origin <branch>

Branches



Branches

(1) **master** - branch principal (produção)

(1.1) **develop** - branch de desenvolvimento usado como base para inserção de novas funcionalidades.

(1.1.1) **features** - contém novas funcionalidades

(1.2) **hotfixs** - contém o código com correção de erros do atual master.

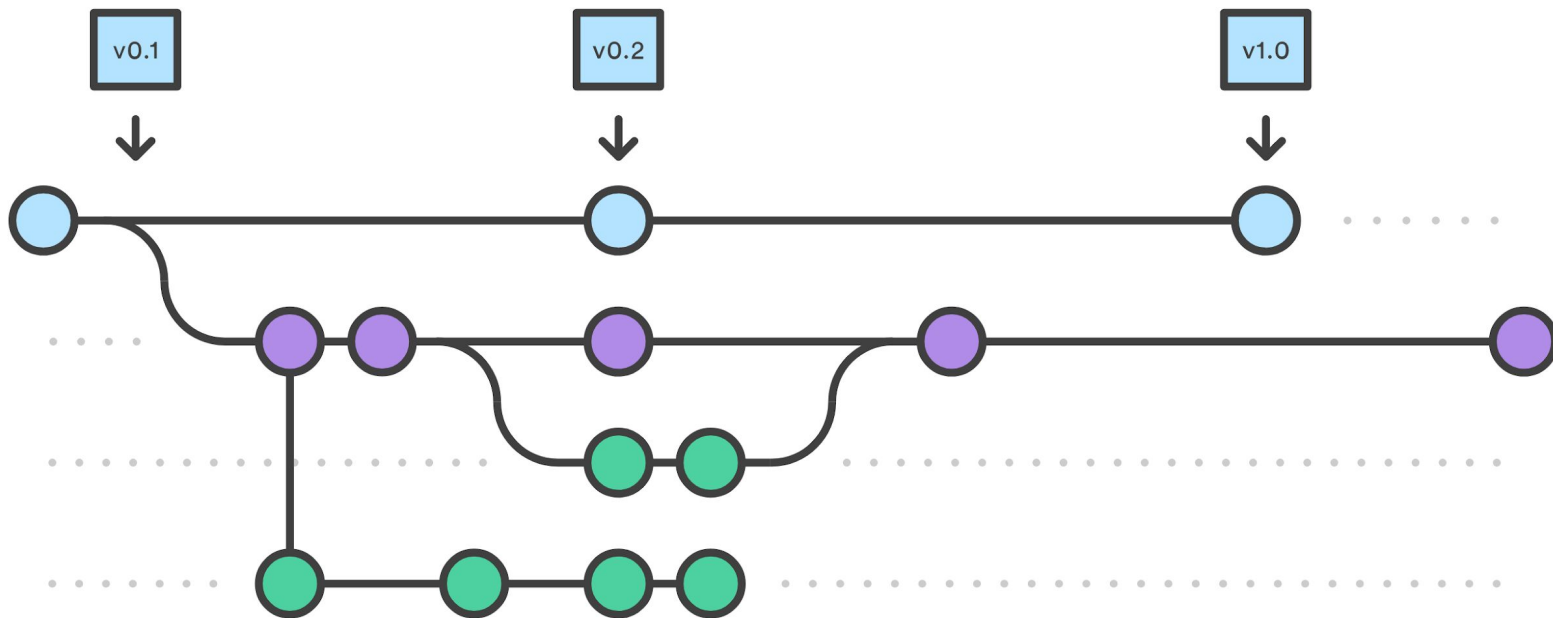
(1.3) **releases** - versões mais confiáveis que as encontradas no branch develop e prováveis candidatas à master.

Master

Develop

Feature

Feature



Fluxo entre branches

- **Visualizando o fluxo**

https://danielkummer.github.io/git-flow-cheatsheet/index.pt_BR.html

Working in feature

```
$ git checkout <branch>
```

```
$ git add *
```

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

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

Atualizar & mesclar

```
$ git pull
```

```
$ git merge <branch>
```

```
$ git add *
```

```
$ git diff <branch_source> <branch_destination>
```

Deu problema!!!

```
[working_dir <- HEAD]
```

```
$ git checkout -- <arquivo>
```

```
[reset arquivo de trabalho - working_dir, index e HEAD]
```

```
$ git fetch origin
```

```
$ git reset --hard origin/master
```

```
http://rogerdudler.github.io/git-guide/index.pt\_BR.html
```

Prática 2 - Gato

- # 1 - Criar projeto no GitLab
- # 2 - Seguir os passos para inclusão do projeto
- # 3 - Adicionar o professor como membro
- # 4 - Criar uma branch codeigniter
- # 5 - fazer as alterações do projeto para uso do CI
- # 6 - Atualizar repositório (add, commit, push)

Contatos

- E-mail: carlos.dantas@ifrn.edu.br
- Site: **carlosfran.info**
- Gitlab e GitHub: @**carlosfran**