

# Git

Do noob até o Wizard lv12

Mateus Koppe

## Um pouco de história

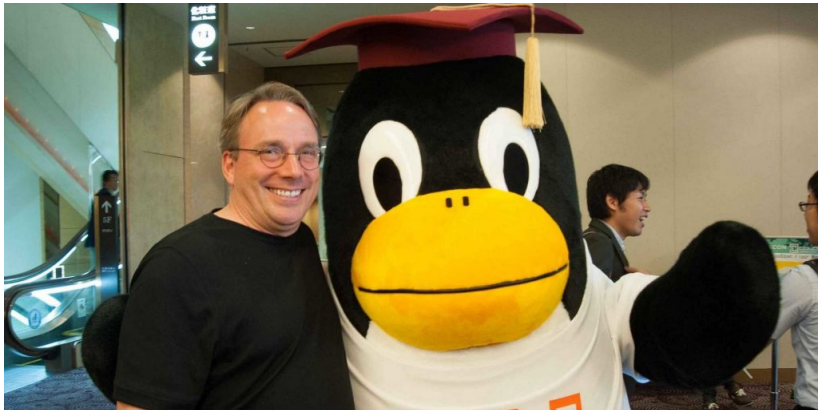


Figure 1: Linus Torval criador do Git

Foi criado em 2005 por Linus Torval para o versionamento do kernel Linux e foi rapidamente adotado por outros projetos.

# Sobre a apresentação

O repositório está na URL

<https://github.com/mateusKoppe/git-basic-guide>

# Como instalar

## Debian based

```
sudo apt install git
```

## Red-Hat based

```
sudo dnf install git
```

## Arch based

```
sudo pacman -S git
```

## Windows

Não sei, procura lá

# Setup

```
git config --global user.name <name>
```

```
git config --global user.email <email>
```

## Sobre `--global`

Utilize apenas no seu computador pessoal, caso você queira definir a configuração apenas para um repositório específico use `--local` (já é o padrão)

# Criando um repositório

```
# Inicia um repositório vazio  
git init
```

# Comandos e conceitos básicos

- ▶ `git-add`
- ▶ `git-status`
- ▶ `git-commit`
- ▶ `git-log`
- ▶ `git-diff`



## git-add

*# Adiciona arquivos para serem trackeados*  
`git add <files>`

## git-status

```
# Recebe status do repositório, dos arquivos  
git status
```

## git-diff

*# Exibe as diferenças que não foram adicionadas*  
`git diff`

*# Exibe as que foram adicionadas:*  
`git diff --cached`

# git-commit

*# Cria um commit*

```
git commit
```

*# De forma rápida:*

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

# git-log

*# Exibe os logs*

```
git log
```

*# Uma linha por log*

```
git log --oneline
```

# Branches

- ▶ `git-branch`
- ▶ `git-checkout`
- ▶ `git-stash`
- ▶ `git-merge`

# git-branch

*# Lista as branches criadas e exibe a branch atual*

```
git branch
```

*# Cria uma nova branch baseada na branch atual*

```
git branch <name>
```

*# Deleta uma branch*

```
git branch -d <name>
```

# git-checkout

*# Troca de branch*

```
git checkout <name>
```

*# Cria e troca de branch*

```
git checkout -b <name>
```



# git-stash

*# Salva as mudanças de uma branch e reseta-a*  
`git stash`

*# Retorna as mudanças que foram salvas para a branch*  
`git stash apply`

## git-merge

*# Junta os commits da branch atual com a branch alvo*  
`git merge <name>`

# Remote

Git não faria sentido se não houvesse uma forma de armazenar os repositório em algum lugar onde possa ser compartilhado para outras pessoas.

Crie um repositório online

- ▶ `git-remote`
- ▶ `git-push`
- ▶ `git-pull`
- ▶ `git-clone`

# Criando um repositório no Github

Owner

Repository name \*



mateusKoppe ▾

/

Great repository names are short and memorable. Need inspiration? How about **fantastic-guide**?

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.


Add .gitignore: **None** ▾



Add a license: **None** ▾







Create repository


# Criando um repositório no Github



 mateusKoppe / **git-basic-guide**



 Unwatch  1



 Star  0


 Fork  0


 Code


 Issues  0


 Pull requests  0

 Projects  0

 Wiki

 Security

 Insights


 Settings

## Quick setup — if you've done this kind of thing before

or

HTTPS


SSH



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


## ...or create a new repository on the command line

```
echo "# git-basic-guide" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mateusKoppe/git-basic-guide.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin https://github.com/mateusKoppe/git-basic-guide.git
git push -u origin master
```



## git-remote

*# Lista os repositórios adicionados*

```
git remote
```

*# Adiciona um repositório remoto*

```
git remote add <name> <url>
```

*# Por convenção o repositório principal geralmente*

*# é nomeado como origin*

```
git remote add origin <url>
```

# git-push

*# Push = Empurra*

*# Envia os commits da branch selecionada*

*# para o remote selecionado*

`git push <remote> <branch>`

*# O mais comum é*

`git push origin master`

# git-pull

```
# Pull = Puxa  
# Atualiza a branch selecionada de acordo com o  
# remote selecionado  
git pull <remote> <branch>
```



## git-clone

*# Clona um repositório online*

```
git clone <url> [<folder>]
```

# Gitignore

Caso seja necessário que o repositório ignore algum arquivo ou algumas pasta é possível criar um `.gitignore`, nele você insere quais arquivos deverão ser ignoradas no repositório.

## Variável HEAD

Reflete o branch e commit atual. Você também pode utilizar ~<n> para referenciar commits anteriores.

# Macetes

```
git push origin HEAD
```

```
git reset --head HEAD~1
```