

# Controle de versão

## Introdução

emerson@paduan.pro.br

## O que é?

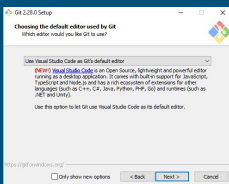
O que é controle de versão?

- Um sistema que mantém um registro das alterações realizadas, permitindo saber quais foram as alterações realizadas (histórico).
- Permite saber quem fez e quando fez as alterações
- Permite REVERTER as alterações feitas
- Permite o desenvolvimento colaborativo

emerson@paduan.pro.br

## Git

<https://git-scm.com/downloads>



Controle de versão

emerson@paduan.pro.br

## Por que Git?

Vantagens?

- Faz o controle de versão distribuída
- Usuários mantêm uma cópia do código completo em sua máquina local
- Desempenho
- Uma das mais utilizadas pelos desenvolvedores

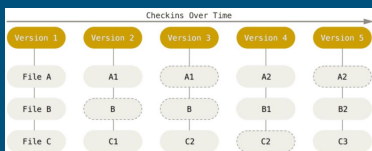
emerson@paduan.pro.br

## Commit



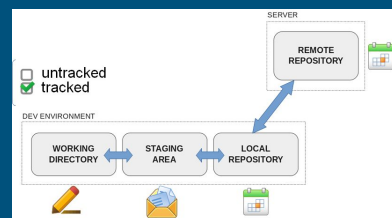
Maneira como o Git mantém um histórico do seu código.

Basicamente ele grava como estava o código em um determinado ponto (tempo).



emerson@paduan.pro.br

## Áreas do Git



emerson@paduan.pro.br

## Iniciando...

```
$> git init [diretório]
```

Cria um repositório GIT no diretório indicado, ou no diretório atual.

emerson@paduan.pro.br

## Configurações

Caso não estejam aparecendo as cores no terminal

```
$> git config --global color.ui auto
```

Configurando o usuário:

```
$> git config [--global] user.name seuNomeDeUsuário
```

```
$> git config [--global] user.email seuE-mail
```

emerson@paduan.pro.br

## Status

```
$> git status
```

Lista como estão os arquivos untracked, staged, unstaged.

emerson@paduan.pro.br

## Adicionando ao stage

```
$> git add <arquivo(s)>
```

```
$> git add * / git add .
```

Adiciona dos arquivos *modificados* para o stage.

emerson@paduan.pro.br

## Commit

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

Cria um commit (snapshot) dos arquivos que estão no stage.

Um commit é identificado por um *texto* informado pelo programador e um *hash code*.

emerson@paduan.pro.br

## Log

```
$> git log
```

Mostra o histórico dos commits.

emerson@paduan.pro.br