

# Controle de versão Distribuído

## Introdução

[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)

## Sobre o Professor



*Prof. Emerson Paduan*

### Formação em Ciência da Computação:

- Graduação (UEL)
- Mestrado (USP)

- 20 anos de Anhembi
- 10 anos coordenação
- Treinamento in-company

### Aulas em:

- ◆ Ciência da Computação,
- ◆ Sistemas de Informação,
- ◆ Design de Games
- ◆ Eng. da Computação,
- ◆ Tec. Gestão de TI,
- ◆ Tec. Análise e Desenv. de Sist.
- ◆ Ciência dos dados

[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)

## Avaliações

- A1 = 30 pontos
  - Dissertativa (profs Anhembi)
- A2 = 30 pontos
  - Objetiva (profs Nacional)
- A3 = 40 pontos
  - Prática (profs Anhembi)
- AI - substitui A1 ou A2
  - início do semestre seguinte

emerson@paduan.pro.br

## Contexto

Já passou por isso?

trabalho.doc  
trabalho-v02.doc  
trabalho-v03.doc  
...etc..  
trabalho-vfinal.doc  
trabalho-vfinal-ultima.doc  
trabalho-vfinal-ultima-mesmo.doc  
trabalho-vfinal-ultima-mesmo-entregue.doc

emerson@paduan.pro.br

## O que é?

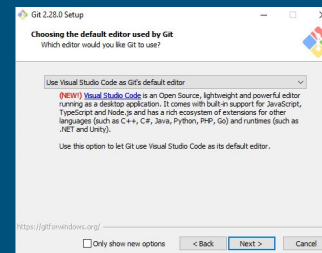
O que é controle de versão?

- 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

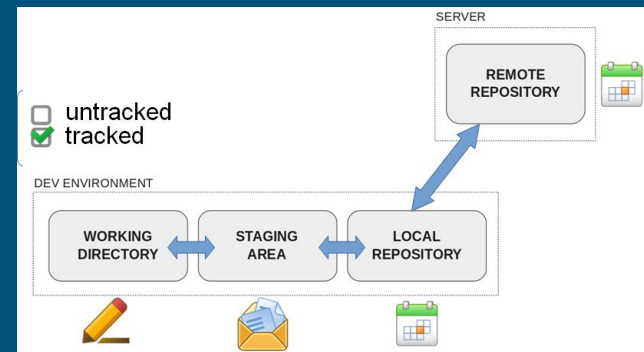
## Por que Git?

### 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

## Resumo 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

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