Controle de versão Distribuído

Introdução

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

Softwares

JDK 17
https://www.oracle.com/java/technologies/downloads/

VS Code
https://code.visualstudio.com/

Git
https://ahttps://git-scm.com

GitHub
https://github.com/

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

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

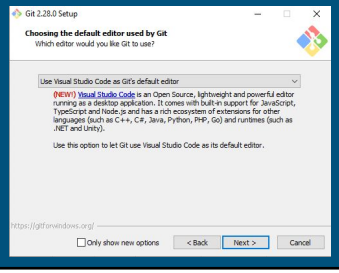
O que é?

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

Git

https://git-scm.com/downloads



Controle de versão

Por que Git?

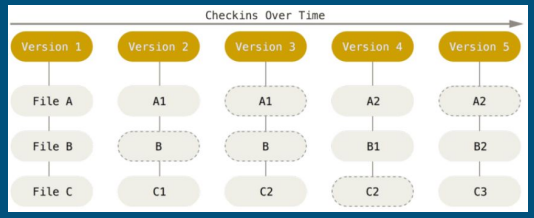
Vantagens?

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

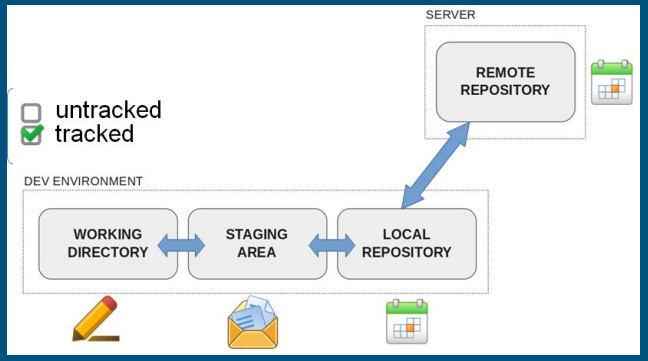
Commit

Maneira como o Git mantém um histórico do seu código.

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



Resumo Git



Iniciando...

$> git init [diretório]

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

Configurações

Configurando o usuário:

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

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

Status

$> git status

Lista como estão os arquivos untracked, staged, unstaged.

Adicionando ao stage

$> git add <arquivo(s)>

$> git add * / git add .

Adiciona dos arquivos *modificados* para o stage.

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***.

Log

$> git log [–oneline]

Mostra o histórico dos commits.

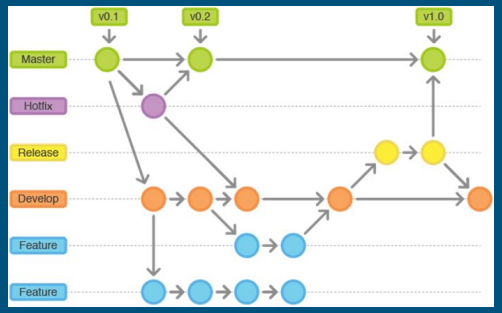
.gitignore

Arquivo texto que indica arquivos e/ou pastas que devem ser ignorados pelo git.

Podem ser usados padrões de nomes como "*.class", por exemplo

Diretórios são indicados usando "/" no final como: dir/

Branches



Branches

$> git branch <nome da branch>
Cria uma nova branch (ramificação).

$> git branch
Exibe todas as branches. A atual é mostrada com um *

$> git branch -r
Exibe todas as remotas

$> git branch -a
Exibe todas inclusive as remotas

Branches

$> git checkout <nome da branch> / git switch <nome da branch>
Alterna para uma outra branch.

$> git branch –d <nome da branch>
Apaga uma branch.

Branches

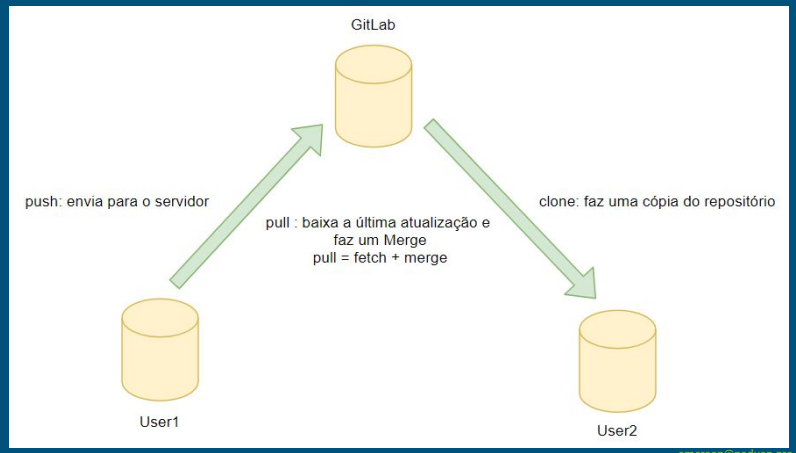
$> git branch -m <new-name>
Alterna o nome de uma branch.

$> git branch -m <old-name> <new-name>
Alterna o nome de uma branch quando você não está na branch.

Repositórios Git

https://github.com/

https://gitlab.com/



Remote

$> git clone <url do repositório remoto>
Clona na máquina local um repositório remoto

$> git remote add origin <url remoto>
Adiciona (configura) um repositório remoto para executar o push.

$> git push [origin] [branch]
Envia o commit branch para o origin remoto.

$ git pull <remote>
Faz uma cópia da branch atual do repositório remoto, para o repositório local fazendo um merge na cópia local.

Remote

$> git remote show origin
mostra informações sobre o repositório remoto

Exercício

1) Criar uma pasta local ("exerciciogit" por exemplo)
2) Iniciar um repositório do git nesta pasta
3) Criar um novo arquivo com algum conteúdo
4) Fazer um commit com o novo arquivo (após colocar no stage)
5) Criar um repositório no Github
6) Enviar para o Github o repositório criado na sua máquina local

Resposta

git add .

git commit -m " [*mensagem*] "

git remote add origin [*link_do_repositório*]

git push origin master

Ferramentas visuais