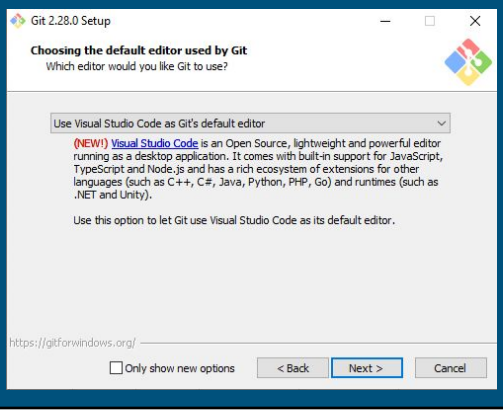
Controle de versão

Git / GitHub
Introdução

emerson@paduan.pro.br

Git

https://git-scm.com/downloads



Controle de versão

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

O que é?

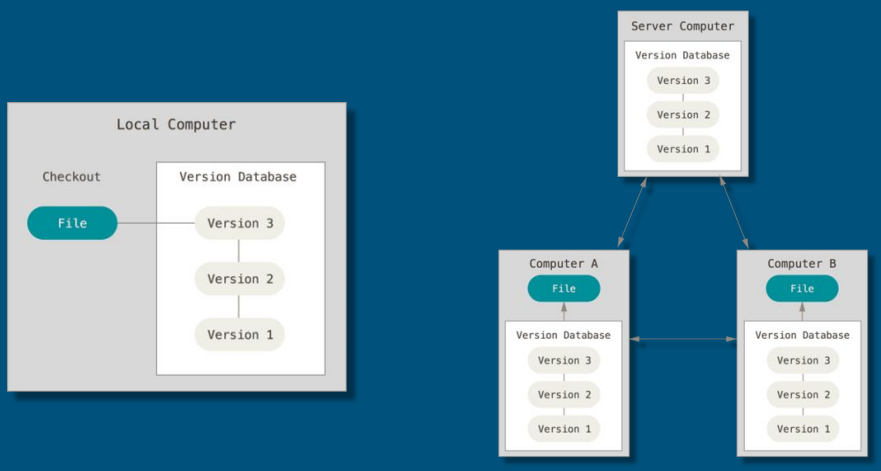
O que é controle de versão?

- 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

Repositórios

Conjunto de arquivos e o histórico de alterações daqueles arquivos.

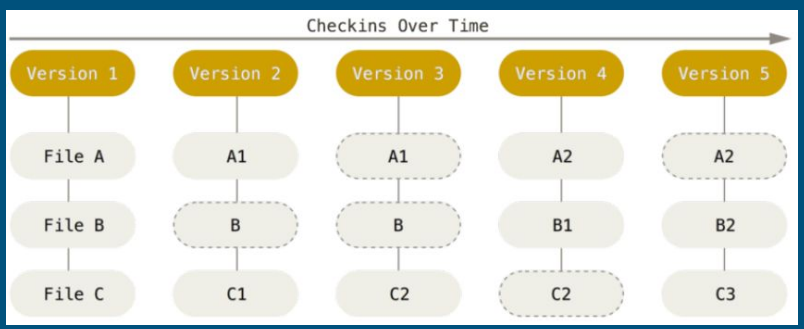
Normalmente trata-se de todos os Snapshots de um projeto.



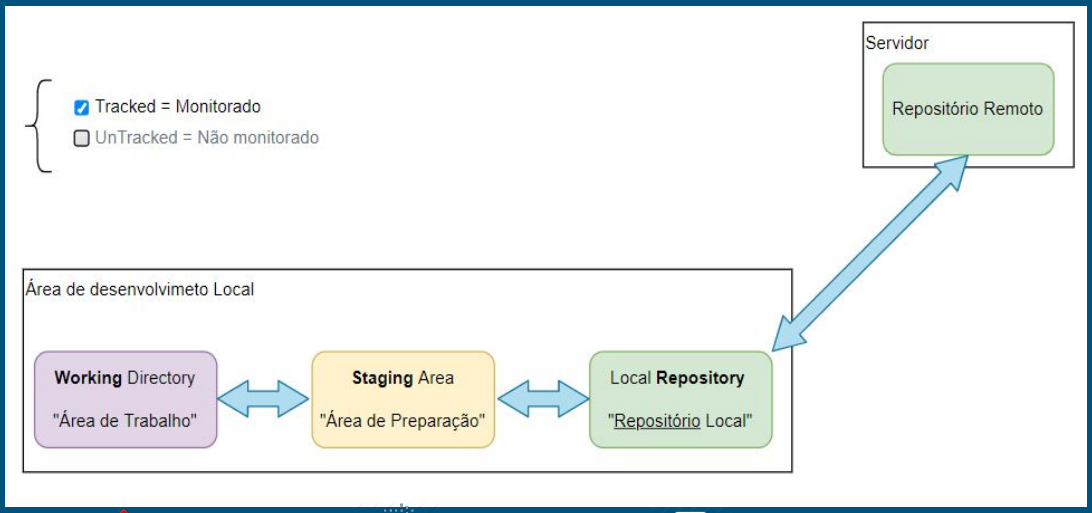
Commit = Foto

Maneira como o Git mantém um histórico do seu código.

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



Áreas do 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

Mostra o histórico dos commits.

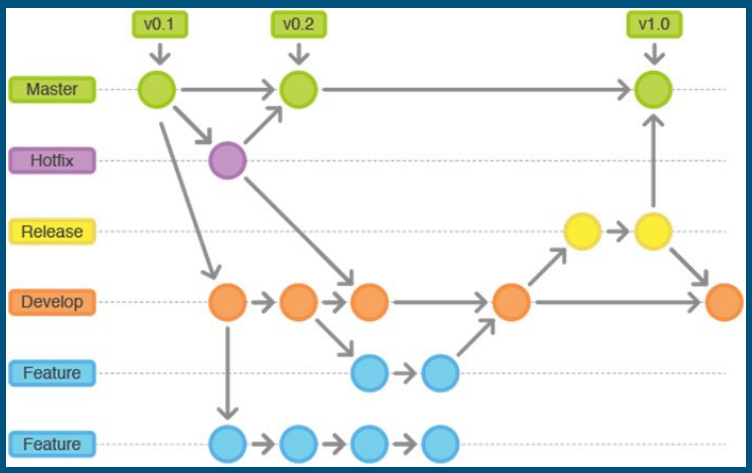
.gitignore

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 = ramificações



Branches

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

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

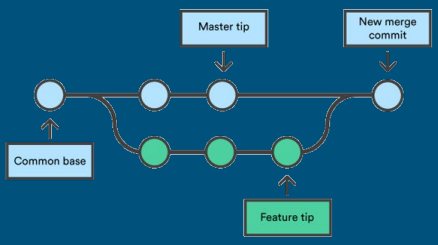
$> git checkout <nome da branch> /no git + atual --> git switch <nome da branch>
Alterna para uma outra branch.

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

Merge = juntar

$> git merge <branch1> <branch2>

Junção de duas banches.



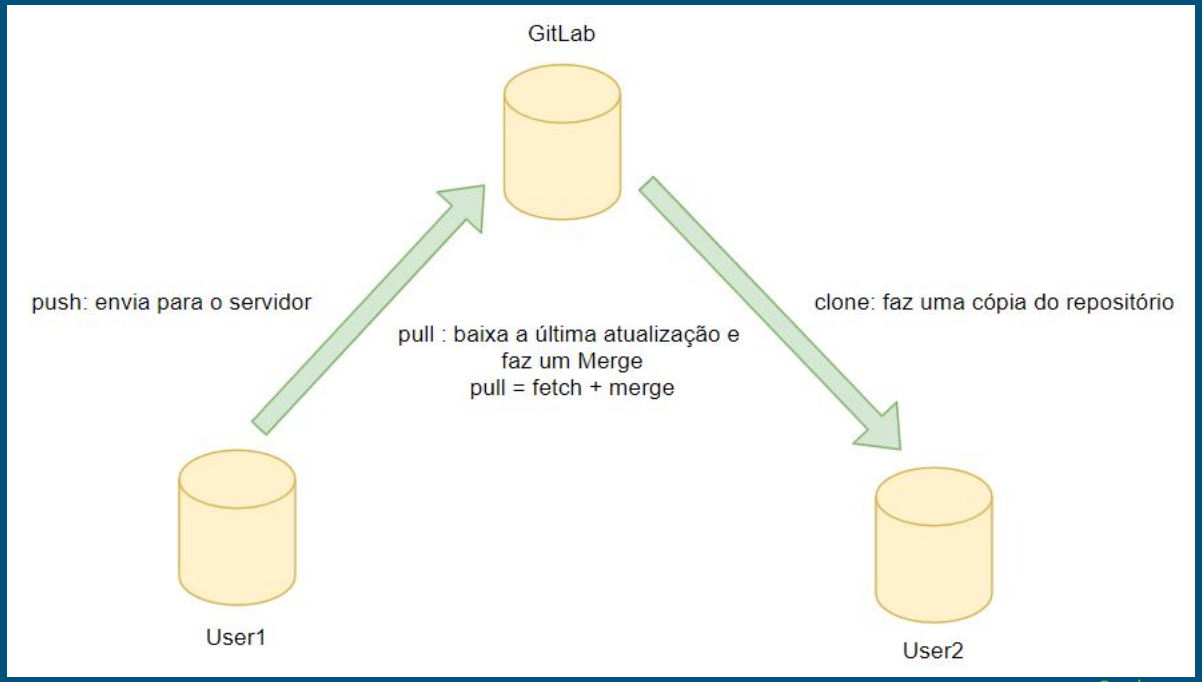
Repositórios Git

GitHub

https://github.com/

GitLab

https://gitlab.com/



Remote

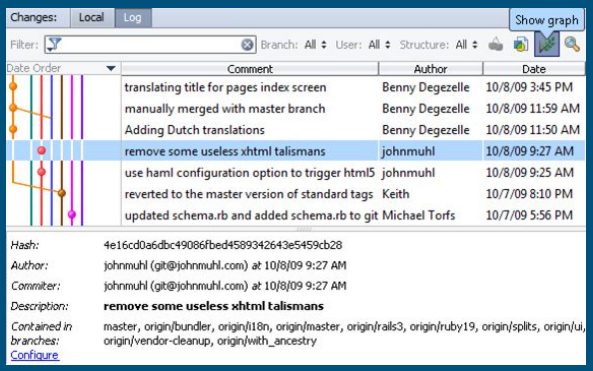
$> git clone <url do repositório remoto>
Clona na máquina local um repositório remoto

$> 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.

Ferramentas visuais



Letras no VS Code

A - Added (This is a new file that has been added to the repository)

M - Modified (An existing file has been changed)

D - Deleted (a file has been deleted)

U - Untracked (The file is new or has been changed but has not been added to the repository yet)

C - Conflict (There is a conflict in the file)

R - Renamed (The file has been renamed)