

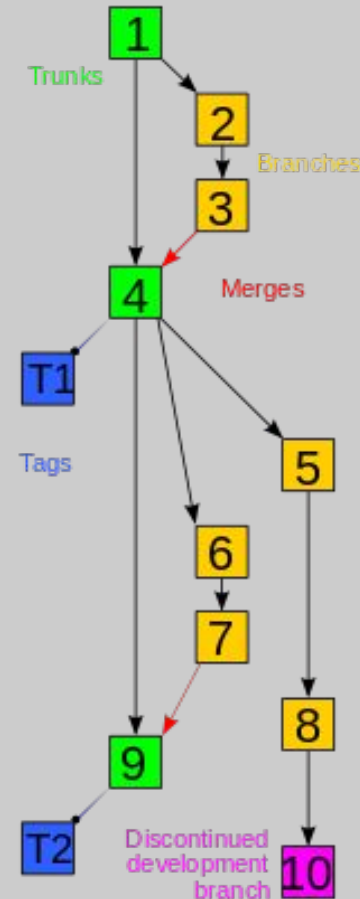
ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

GIT e GitHub

Controle de versões

Ferramentas que gerenciam mudanças em programas de computador, documentos, web sites, etc.

Com elas, é possível rastrear alterações no código, combinar diferentes atualizações, reverter as versões anteriores, etc.



GIT e GitHub

GIT é um sistema de controle de versões, que permite gerenciar e rastrear o histórico de alterações no código.

GitHub é um serviço de armazenamento em nuvem, que permite gerenciar repositórios criados com GIT.

Tutorial

Passo 0: Instalar GIT e criar uma conta no GitHub

- Instalação: <https://git-scm.com/downloads>
- Criar conta: <https://github.com/join>

Tutorial

Passo 1: Criar um repositório GIT local

- Usando o terminal, acessar a pasta do projeto que deseja criar o repositório**
- Para inicializar o repositório, digite: 'git init'**

Tutorial

Passo 2: Adicionar um novo arquivo no repositório

- Criar um arquivo no repositório**
- Verificar a saída do comando 'git status'**

Fluxo de trabalho

O repositório local consiste em três "árvores" mantidas pelo GIT:

- 1) A primeira é o diretório de trabalho, que contém os arquivos vigentes.
- 2) A segunda, "index", funciona como uma área temporária.
- 3) A terceira, "head", aponta para o último commit (confirmação) realizado.



Fonte: https://rogerdudler.github.io/git-guide/index.pt_BR.html

Tutorial

Passo 3: Adicionar um arquivo na área temporária (stage/index):

- Adicionar o arquivo usando o comando 'git add'**
- Verificar a saída do comando 'git status'**

Tutorial

Passo 4: Realizar um commit:

- Executar o comando: `git commit -m "mensagem"`**

Com o comando acima, as modificações são enviadas para a árvore 'head' no repositório local, mas ainda não foram enviadas ao repositório remoto (github).

Tutorial

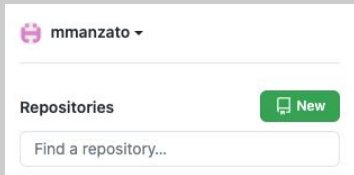
Passo 5: Criar um novo branch:

- Executar o comando: `git checkout -b <nome>`**
- Verificar todos os branches criados: `git branch`**

Ao fazer alterações (inclusive com commits) em um determinado branch, os demais branches não serão alterados até que seja feita a junção desses branches (merge).

Tutorial

Passo 6: Criar um novo repositório no GitHub:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * / Repository name *

Great repository names are short and memorable. Need inspiration? How about [didactic-disco](#)?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

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

```
echo "# Teste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/mmanzato/Teste.git
git push -u origin master
```

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

```
git remote add origin https://github.com/mmanzato/Teste.git
git branch -M master
git push -u origin master
```

Tutorial

Passo 7: Enviar (push) alterações de um branch para o GitHub:

- Executar os comandos:

```
git remote add origin <url do repositório>  
git push -u origin <branch>
```

Tutorial

Outras funcionalidades:

- Criar um pull request (PR): é uma maneira de alertar o dono de um repositório que se deseja realizar alguma alteração no código.
- Combinar um pull request (merge): permite juntar as alterações de um pull request com o branch principal.
- Refletir as alterações do repositório remoto no repositório local

Verificar:

<https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

GIT e GitHub