



# BEM-VINDOS AO PROGRAMA DE TALENTOS 2022!

**Git / Python / FastAPI**

# APRESENTAÇÃO



**Fernando Celmer**

FernandoCelmer

Site: <https://fernandocelmer.com/>

Github: <https://github.com/FernandoCelmer>

Linkedin: <https://www.linkedin.com/in/fernando-celmer/>

Discord: **fernando.celmer#0679**

# CRONOGRAMA

- Git (Github)

- Dia 1 - 15-08-2022

- Python Básico

- Dia 2 - 16-08-2022

- Framework FastAPI

- Dia 3 - 17-08-2022

- Dia 4 - 18-08-2022

- Dia 5 - 19-08-2022

# AVISOS

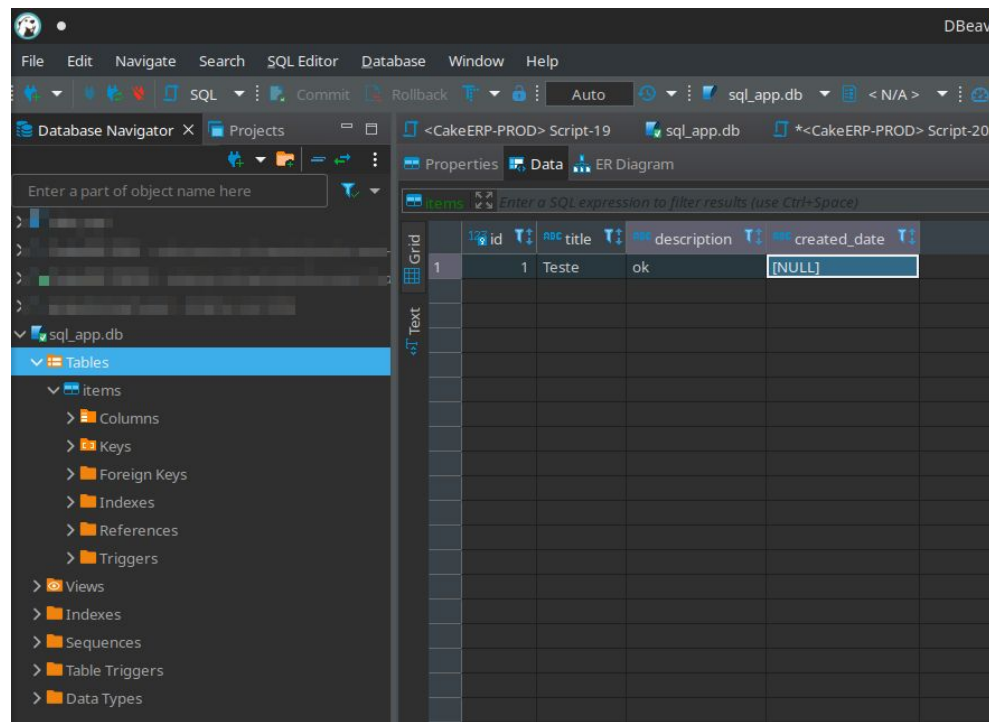
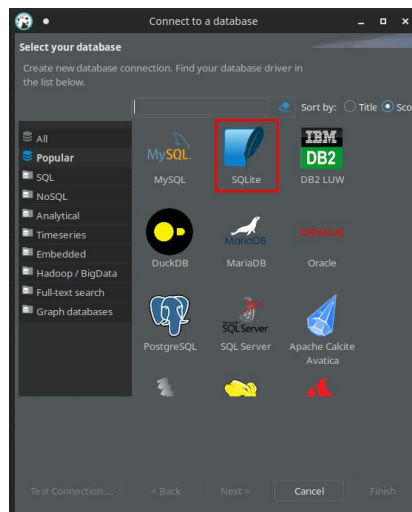
<https://discord.gg/MrYP3XRb>

- Grupo Discord
  - Entrega Actividades
  - Prova Final
  - Datas Importantes
-

DIA 5  
PYTHON FASTAPI

# VISUALIZAÇÃO DO BANCO DE DADOS

- Menu >
- Database >
- New Database Connection



# CONFIGURAÇÕES DE VARIÁVEIS DE AMBIENTE

Vamos configurar as nossas variáveis de ambiente para as informações sensíveis que estão na nossa aplicação de teste como por exemplo a nossa URL de conexão ao banco de dados.

Que fica no arquivo `database.py`

Primeiro vamos instalar mais um requerimentos:

```
pip install python-dotenv
```

E executar novamente o comando para gerar o arquivo de requirements.txt

```
pip freeze > requirements.txt
```

## CRIAR ARQUIVO .ENV

Primeiro vamos criar um arquivo chamado `.env`

E vamos inserir a seguinte variável dentro dele:

```
SQLALCHEMY_DATABASE_URL=sqlite:///./sql_app.db
```



# CONFIGURAÇÕES DE VARIÁVEIS DE AMBIENTE

```
import os
```

```
from dotenv import load_dotenv
```

```
load_dotenv()
```

```
SQLALCHEMY_DATABASE_URL = os.getenv("SQLALCHEMY_DATABASE_URL")
```

# TESTES UNITÁRIOS - PYTEST

Pytest é uma estrutura de teste Python que se originou do projeto PyPy. Ele pode ser usado para escrever vários tipos de testes de software, incluindo testes de unidade, testes de integração, testes de ponta a ponta e testes funcionais. Seus recursos incluem testes parametrizados, fixtures e reescrita de assert.



# O QUE É DOCKER?

Docker é uma ferramenta opensource escrita em GO para gerenciamento e criação de containers.

# SERVIDOR FÍSICO / MÁQUINA VIRTUAL / CONTAINER

**Servidor** é uma máquina física qualquer onde você instala um sistema operacional baseado em **Linux**. Quando é **máquina virtual**, você executa um novo sistema operacional dentro do seu sistema operacional que já está sendo executado na sua **máquina física**. O container não precisa rodar um novo sistema operacional dentro do sistema operacional da minha máquina física.

Quando é utilizado uma **máquina virtual**, ela contém o seu próprio **kernel**, as suas próprias bibliotecas etc. Em um container não, quando eu estou utilizando o container, somente o processo que eu pedir vai estar em execução. As bibliotecas e os binários, ele compartilha da máquina host.

# COMANDOS BÁSICOS DOCKER

`sudo docker ps` | Faz a listagem dos containers que estão ativos no momento.

`sudo docker ps -a` | Com o parâmetro `-a` é feito a listagem de todos os containers inativos.

`sudo docker run -i -t -p 8080:80 ubuntu:14.10` | Cria um novo container com ubuntu. Pode ser colocado parametros adicionais como a porta que vai ser redirecionado esse container.

`sudo docker stop ID` | Para a execução de um container.

# DIFERENÇA ENTRE DIFERENÇA ENTRE O DOCKERFILE E O DOCKER-COMPOSE:

Dockerfile configura as imagens para um container e o Docker-compose configura as imagens para os containers e a conexão entre eles...

# CONFIGURAÇÃO DOCKERFILE

Criar arquivo chamado `Dockerfile` sem nenhuma extensão.

```
FROM python:3.9
```

```
COPY ./app_aula ./app_aula
```

```
COPY ./requirements.txt ./requirements.txt
```

```
RUN pip install --no-cache-dir -r ./requirements.txt
```

```
CMD ["uvicorn", "api:app", "--host", "0.0.0.0", "--port", "80"]
```

# EXECUTANDO CONTAINER

Construindo a imagem Docker

- `sudo docker build --tag fastapi-aula/dev --file Dockerfile .`

Iniciando o container Docker

- `sudo docker run --name aula_fastapi -d -p 80:80 fastapi-aula/dev`



# O QUE É DEPLOY EM PROGRAMAÇÃO?

O verbo **deploy**, em inglês, quer dizer **implantar**.

Em **programação**, seu sentido está intimamente relacionado à sua tradução literal: fazer um deploy, em termos práticos, significa colocar no ar alguma aplicação que teve seu desenvolvimento concluído.

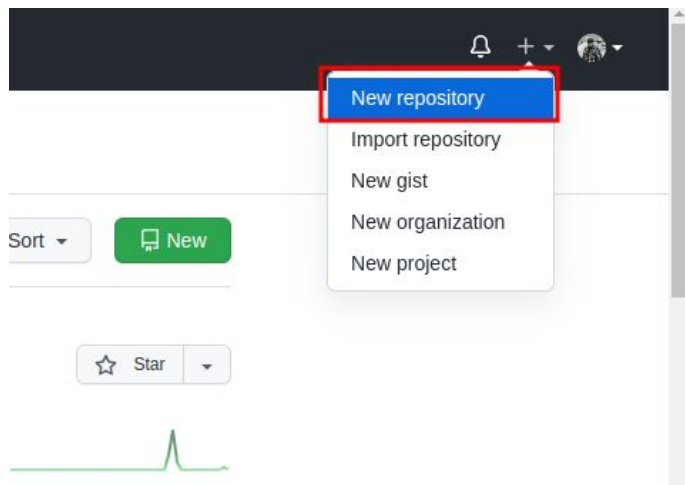
Esta tarefa é extremamente comum dentro do escopo de trabalho dos **programadores**, embora seja muito comumente associada somente aos profissionais de infraestrutura, ou DevOps.

Quando um **projeto** é finalizado por um desenvolvedor e, após seus **testes**, é finalmente hospedado e colocado no ar, ele passa pelo processo de deploy.

De mesmo modo, quando um sistema sofre alguma melhoria ou alteração em seu **código-fonte**, implementar essa alteração ao sistema que está no ar também é um tipo de deploy.

# criação novo repositório

Criação de novo repositório no Github para colocar utilizar no deploy com Heroku.



## Create a new repository

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

### Repository

Start your repository

No template

### Owner

Fer

Great repository

### Description

Private

Public

### 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.](#)

.gitignore template: None

### Choose a license

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

License: MIT License

This will set `master` as the default branch. Change the default name in your [settings](#).

**Info** You are creating a private repository in your personal account.

Create repository

Create repository

# criação novo repositório

## 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.](#)

.gitignore template: None ▾

### Choose a license

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

License: MIT License ▾

This will set  master as the default branch. Change the default name in your [settings](#).

 You are creating a private repository in your personal account.

Create repository

/ cakeerp-test-fastapi Private

 Pull requests  Actions  Projects  Security  Insights  Settings

 master ▾

 1 branch

 0 tags

Go to file

Add file ▾

Code ▾



FernandoCelmer Initial commit



LICENSE

Initial commit



README.md

Initial commit

README.md

## cakeerp-test-fastapi

 Clone



HTTPS SSH GitHub CLI

<https://github.com/FernandoCelmer/cake> 

Use Git or checkout with SVN using the web URL.

 Download ZIP

# AJUSTES NOVO REPOSITÓRIO

- Criar pasta `app`
- Colocar todos os arquivos na pasta `app`
- Ajustar `imports`
- instalar `python-dotenv`
- instalar `uvicorn`
- Gerar novos `requerimentos`



Heroku é uma plataforma de nuvem como serviço que suporta várias linguagens de programação. Uma das primeiras plataformas em nuvem, o Heroku está em desenvolvimento desde junho de 2007, quando suportava apenas a linguagem de programação Ruby, mas agora suporta Java, Node.js, Scala, Clojure, Python, PHP e Go



# CONFIGURAÇÃO RUNTIME

Arquivo runtime.txt

python-3.9.7

# CONFIGURAÇÃO PROFILE

Arquivo runtime.txt

```
web: uvicorn app.api:app --host=0.0.0.0 --port=${PORT:-5000}
```

PROVA

eeeeeh!



# PROVA

## O que é para fazer?

1 - Um projeto mínimo e funcional de uma API em FastAPI, com os principais tópicos explicados durante os cinco dias de curso. O objetivo desta API é realizar um CRUD (Create, Read, Update, Delete) em uma ou mais tabela do banco de dados.

O projeto deve ter no mínimo:

- Models
- Schemas
- Endpoints (GET/POST/PUT)
- Banco de Dados SQLite

## Onde fazer as atividades?

1 - Em um novo repositório que deve ser criado no seu perfil com o nome de `cakeerp-test-fastapi`. Fica a critério do aluno colocar em produção essa aplicação, mas não é obrigatório.

## Como enviar a sua prova?

Envie um E-mail para [fernando.celmer@cakeerp.com](mailto:fernando.celmer@cakeerp.com) com o link do repositório.

## LINKS

- [Informações Básicas de Docker](#)
- [Pytest - Helps you write better programs](#)
- [Heroku Cloud Application Platform](#)
- [O que é Deploy?](#)
- [Docker](#)