



Processo de integração da aplicação em container docker

Docente: Rezembrim de Paula Soares

Discentes: Ian Guilherme Duarte da Silva - 20232014050004

Jordan Júlio Francelino da Silva - 20241014050014

Pedro Jordan Cassiano Costa - 20221014050034

Tamires Angélica da Silva Bezerra - 20221014050033

- Sumário:

- Soluções
- Objetivo
- Etapas Realizadas
- Ferramentas e suas funções
- Conclusão

- Elementos para criação do projeto

- 1- Banco de Dados - PostgreSQL
- 2- Backend Principal - Integração com API do SUAP e Gerador dos QRs - Python
- 3- Frontend Principal - Dashboard / Interface no SUAP - Django/Python
- 4- Proxy Reverso - Nginx
- 5- Gerenciador de DNS interno - Próprio do Campus CNAT
- 6- Docker - Containers

- Objetivo:
- Executar a aplicação de geração de QR Codes e integração SUAP em ambiente Docker para garantir portabilidade e isolamento.

- Etapas Realizadas

1. Criação do Dockerfile com base em Python 3.11-slim.
2. Instalação das dependências (FastAPI, uvicorn, qrcode).
3. Criação do docker-compose.yml para orquestrar o serviço.
4. Testes locais com docker compose up → API acessível em localhost:8000/docs.
5. Verificação de resposta do endpoint /gera_qr e geração correta dos QR Codes.

- Banco de Dados - PostgreSQL:
- **Objetivos:** armazenamento, segurança e integração.

- Backend - Python:

- **Objetivo:** gerar QR Codes e autenticar matrículas via SUAP

- Frontend / Dashboard (Django)
- **Objetivo:** exibir presença dos alunos e relatórios para professores.

- Proxy Reverso - Nginx
- **Objetivo:** Rotear conexões entre containers.

- Docker - Gerenciador de Containers:
- **Objetivo:** Orquestrar o ambiente de serviços isolados.

- Conclusões da Sprint
- Aplicação 100% containerizada, pronta para integração futura com banco de dados e proxy reverso (Nginx).