

Aplicação escolhida:

A aplicação foi originalmente demonstrada em aula, contendo apenas uma versão em terminal(Python) para gerenciamento de estudantes. Para criação de mais containers, desenvolvi uma versão web(Java/Tomcat), que oferece as mesmas funcionalidades, porém com uma interface gráfica

Ambas as versões compartilham o mesmo banco de dados MongoDB, permitindo que as alterações feitas em uma interface sejam refletidas imediatamente na outra. Isso garante consistência nos dados e flexibilidade no uso.

Containers e tecnologias usadas:

MongoDB - Banco de dados:

- Banco que armazena os registros dos estudantes
- Foi usada a imagem oficial do MongoDB
- Usa as variáveis de ambiente via .env(usuário e senha root)
- Os dados são salvos em um volume para não serem perdidos quando reiniciar o container
- Conectado a rede app_network para comunicação com os outros serviços
- Tanto a versão web quanto a versão terminal se conectam a esse MongoDB

Tomcat - versão web

- Uso da imagem oficial do Tomcat
- Depende do Mongo para estar rodando
- Variáveis de ambiente via .env
- O arquivo .war é copiado para a pasta webapps do Tomcat, que faz o deploy automático
- Conectado a app_network para acessar o MongoDB
- É possível acessar em <http://localhost:8080/students/students>

Terminal - versão em python

- Baseado no Dockerfile no diretório students-terminal
 - PyMongo para conexão com o MongoDB
 - Scripts
- Usa imagem base do python
- Também depende do Mongo para estar ativo
- Usa as credenciais do Mongo definidas no .env

Manual de instalação

- Necessário ter Docker e Docker Compose instalados
- Para a versão web: Maven, caso precise reconstruir o WAR

Passos:

- Se necessário reconstruir o WAR, no diretório students-web executar o comando `mvn clean package`
- Para iniciar todos os containers, executar na pasta Estudantes, `docker-compose up --build`

Acessar as interfaces:

- Na web: <http://localhost:8080/students/students>
- No terminal: `docker exec -it terminal python ./main.py`

