

DevOps - Trabalho 1

Alunos

Enzo Youji Murayama - 813606

Lucas Sciarra Gonçalves - 811948

Esta atividade consiste em selecionar uma aplicação — preferencialmente uma que exija múltiplos contêineres para sua execução — e transformá-la em uma aplicação compatível com execução via contêineres.

Visão geral da aplicação

A aplicação foi desenvolvida utilizando o **ABP Studio**, uma ferramenta baseada no framework **ABP (AspNet Boilerplate)**, que facilita a criação de aplicações modernas em .NET, com arquitetura modular, boas práticas de desenvolvimento e suporte integrado para autenticação, autorização, multitenancy, entre outros recursos.

Trata-se de uma aplicação web com backend em ASP.NET Core e frontend baseado em Angular. O projeto segue uma estrutura modular, separando as responsabilidades em camadas distintas, o que facilita sua manutenção, escalabilidade e containerização.

Este projeto se enquadra como uma **aplicação corporativa genérica**, servindo como uma base flexível e escalável para o desenvolvimento de sistemas empresariais diversos, como ERPs, CRMs, portais administrativos, ou outras soluções internas personalizadas. Ele também pode ser classificado como um boilerplate modular, permitindo a fácil extensão e adaptação conforme as necessidades do negócio.

A aplicação inclui funcionalidades básicas como autenticação de usuários, gerenciamento de permissões, auditoria, e uma estrutura pronta para integração com bancos de dados relacionais. Devido à sua arquitetura distribuída, a aplicação se beneficia da utilização de múltiplos contêineres — por exemplo, um contêiner para o backend, outro para o frontend, e um para o banco de dados — o que a torna um excelente caso para orquestração com Docker Compose.

Dockerfiles

Localização dos dockerfiles no projeto:

- **Backend (ASP.NET Core)**
Devops.Trabalho1/src/Devops.Trabalho1.HttpApi.Host/Dockerfile

- **Frontend (Angular)**

Devops.Trabalho1/angular/Dockerfile

- **Banco de Dados (PostgreSQL)**

Devops.Trabalho1/src/Devops.Trabalho1.DbMigrator/Dockerfile

- **Serviço de Autenticação**

Devops.Trabalho1/src/Devops.Trabalho1.AuthServer/Dockerfile

Contêineres e tecnologias utilizadas

Para este trabalho, como realizamos em dupla, precisamos utilizar cinco contêineres, que são os seguintes:

- **Backend (ASP.NET Core)**

Responsável pela lógica de negócios e pelas APIs da aplicação. Este contêiner executa o projeto de backend gerado pelo ABP Studio.

- **Frontend (Angular)**

Contêiner dedicado à interface do usuário. Ele executa o projeto frontend da aplicação, também gerado pelo ABP Studio, e consome as APIs expostas pelo backend.

- **Banco de Dados (PostgreSQL)**

Utilizado para armazenar os dados persistentes da aplicação.

- **Serviço de Autenticação**

Responsável pela autenticação e emissão de tokens de acesso (JWT). Este contêiner centraliza a gestão de usuários, permissões e autenticação, conforme o modelo de segurança do ABP.

- **Redis**

Utilizado como mecanismo de cache distribuído e gerenciamento de sessões, melhorando a performance da aplicação e suportando cenários com múltiplas instâncias.

Dependências

Para executar a aplicação em contêineres, é necessário que o ambiente de desenvolvimento ou de produção possua as seguintes dependências instaladas:

1. Git

Site oficial: <https://git-scm.com>

2. Docker

Versão recomendada: Docker Engine 20.10 ou superior

Site oficial: <https://www.docker.com>

3. Docker Compose

Versão recomendada: 1.29 ou superior

Site oficial: <https://docs.docker.com/compose/>

Como rodar o projeto?

Vamos começar clonando o repositório em nossa máquina:

```
Shell
git clone https://github.com/lsgonc/devops-trabalho1.git
```

Agora, para executar o projeto, basta utilizar o compose.

Vá para: `devops-trabalho1/Devops.Trabalho1/etc/docker-compose/`

E execute:

```
Shell
docker compose build
docker compose up
```

O frontend deve abrir em: `localhost:4200`

you can also access:

<https://localhost:44348>

<https://localhost:44321/swagger>