Vítor Milanez **RA:804319**

Descrição da aplicação escolhida para a prática de Devops

- O sistema apresentado neste repositório tem como objetivo cadastrar locações de clientes em locadoras, armazenando no banco de dados informações relevantes como a data e hora da locação, além de identificar qual cliente alugou com qual locadora.
- O projeto conta com um sistema de login com diferentes níveis de acesso: clientes, locadoras e administradores. Cada perfil possui permissões específicas para execução de suas respectivas tarefas.
- Idiomas suportados: Português-BR, Inglês e Alemão.

Funcionalidades

- O projeto teve foco em cobrir os requisitos do seguinte arquivo: Requisitos A 1 Servlets.pdf

Requisitos, Instalação e Execução

Antes da containerização, era necessário ter o Java (preferencialmente versão 8 ou superior) e o gerenciador de dependências Maven instalados localmente para executar a aplicação com o seguinte comando:

```
mvn clean package tomcat7:run.
```

Após a adoção do Docker, a execução foi simplificada. Agora, basta ter:

- Docker instalado (https://docs.docker.com/get-docker/))
- Docker Compose instalado (https://docs.docker.com/compose/install/))
- Arquivo .env no diretório bikes-rent com o seguinte conteúdo:

```
DB_HOST=db
DB_USER=user
DB_PASSWORD=password
DB_PORT=3306
DB_NAME=bikeRentSystem
```

Para executar a aplicação:

1. Clone o repositório:

```
git clone https://github.com/VMila/Pratica-Devops.git
cd Pratica-Devops/bikes-rent
```

2. Inicie os containers com:

```
docker-compose up --build
```

3. Acesse a aplicação em:

http://localhost/bikes-rent/

O Nginx te redirecionará automaticamente para o backend da aplicação.

Arquitetura com Docker

A aplicação agora está conteinerizada usando Docker Compose, com 3 containers:

- nginx: atua como proxy reverso, roteando as requisições HTTP para o backend.
- backend: aplicação Java rodando com Maven/Tomcat.
- db: banco de dados MySQL 8 com inicialização via script SQL.
- mailhog: serviço para enviar e-mails em ambiente de desenvolvimento/testes.

Todos os serviços rodam em uma rede Docker interna chamada `bike-network`.

Tecnologias Utilizadas

- Java EE | Servlets
- Tailwind CSS
- JavaScript
- Docker (com Docker Compose)