Sistema para cadastramento de veículos

Para esse projeto foi criado uma API Rest utilizando **Java** e **Spring Boot** no back-end. Para o front-end foi criado um app **Angular** junto com **PrimeNG**. Como servidor foi utilizado o **Apache Tomcat 9.0.8**, já para o gerenciamento de dependências e build do back-end foi utilizado o **Maven** e no front-end o **Angular CLI** e por fim, o versionamento dos fontes foi utilizado o **git**.

Como havia o requisito de não ser utilizado banco de dados, no back-end foram utilizados atributos estáticos, tanto para os dados que carregará o combo de montadoras quanto para os veículos cadastrados.

Passos para compilar e realizar o deploy:

- 1. Build e deploy da API
 - 1. Baixar o projeto da api (https://github.com/Paulo-E-F-Fernandes/veiculos-api);
 - 2. Abrir o terminal no diretório raiz do projeto api e executar o comando **mvn clean install**;
 - 3. Abrir pelo browser a página do Tomcat (http://localhost:8080/) e depois o **Manager App**. Fazer o deploy do arquivo **veiculos-api.war** que se encontra no diretório **target** do projeto da api.
- 2. Build e deploy da UI
 - 1. Baixar o projeto da ui (https://github.com/Paulo-E-F-Fernandes/veiculos-ui);
 - OBS.: É necessário ter instalado o Angular CLI.
 - Abrir o terminal no diretório raiz do projeto ui e executar o seguinte comando: ng build --prod --base-href http://localhost:8080/veiculos/
 - **OBS.:** Os arquivos do app angular serão gerados no diretório **dist**.
 - 3. Criar o diretório **veiculos** dentro de **webapps** no diretório da instalação do tomcat e colar todos os arquivos do diretório **dist** no diretório **veiculos**.
 - 4. Abrir o browser e acessar a url http://localhost:8080/veiculos/index.html, com isso a página inicial do aplicativo será exibida.



5. **Observação final:** foi utilizada como URL base **http://localhost:8080** pois os serviços do angular estão apontando para esse URL. Para alterar o local em que será publicada a API, será necessário alterar a variável **recursoUrl** tanto do **MontadoraService** quanto do **VeiculoService** e também a URL do comando **ng build**.

Imagens do sistema e evidências de teste:

• Página inicial do sistema:



A1. O botão "**Adicionar**" inicia bloqueado e só será habilitado quando todos os campos obrigatórios estiverem preenchidos.

• Dados preenchidos:



• Veículo adicionado:

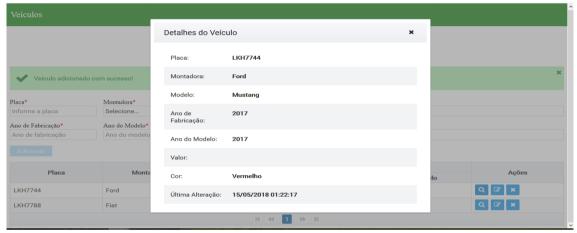


- A2. Visualizar detalhes do veículo.
- A3. Editar veículo.
- A4. Remover veículo.

Validação para não inserir outro veículo com a mesma placa:



Detalhamento do veículo:



• Dois veículos cadastrados:



Veículo removido:

Cadastro de Veículos





Dados alterados:



Alteração realizada:

Cadastro de Veículos

