

Centro Universitário Unidombosco
Curso: Análise e Desenvolvimento de Sistemas
Matéria: Projeto Integrado - ADS

Resenha do Projeto Sistema de Comércio de Automóveis

Aluno: Hedgar Alves da Silva
Maceió-AL

1. Introdução

O Sistema de Comércio de Automóveis foi desenvolvido com o objetivo de oferecer uma solução tecnológica eficiente para a gestão de vendas de veículos. Com a integração entre as tecnologias Django (backend), React (frontend) e PostgreSQL (banco de dados), o sistema proporciona funcionalidades como cadastro de clientes, registro de automóveis e acompanhamento das vendas, além de contar com interfaces responsivas e uma API REST. Este projeto também se destacou por incluir boas práticas de desenvolvimento, como a utilização de contêineres Docker para facilitar o deployment e a escalabilidade.

A escolha deste tema se deu pela relevância do mercado de automóveis e a necessidade de soluções digitais para negócios que operam nesse setor. O projeto foi desenvolvido em um ambiente de aprendizado, sendo estruturado para simular desafios encontrados em cenários reais de desenvolvimento de software.

2. Desafios Encontrados

Durante o desenvolvimento, enfrentamos diversos desafios, tanto técnicos quanto conceituais. O principal desafio técnico foi a integração entre o backend e o frontend. Apesar de Django e React serem ferramentas amplamente utilizadas, garantir que as comunicações entre a API REST e a interface gráfica fossem fluidas exigiu um entendimento detalhado sobre o consumo de APIs e o tratamento de respostas assíncronas.

Outro desafio significativo foi a configuração do banco de dados PostgreSQL, especialmente para garantir que as migrações fossem executadas corretamente e que a estrutura suportasse as operações necessárias. Além disso, configurar o ambiente Docker para orquestrar os serviços do backend, frontend e banco de dados exigiu ajustes para evitar conflitos de rede e garantir a persistência dos dados.

No aspecto conceitual, foi desafiador desenhar um modelo de dados que refletisse adequadamente as relações entre clientes, automóveis e vendas. Foi necessário considerar casos específicos, como a exclusão lógica de registros e a necessidade de histórico para auditoria.

3. Dificuldades e Facilidades

As principais dificuldades estiveram relacionadas ao entendimento e implementação inicial das ferramentas. Por exemplo, configurar corretamente o

React para consumir a API REST exigiu familiaridade com bibliotecas como Axios, enquanto, no backend, foi necessário lidar com permissões e autenticações.

Por outro lado, o uso de ferramentas modernas e bem documentadas, como Django e Docker, tornou o processo mais eficiente. A robustez do Django para lidar com operações no banco de dados e a agilidade do React em construir interfaces dinâmicas foram aspectos que facilitaram o desenvolvimento. Além disso, o uso do Docker foi um grande facilitador na criação de um ambiente isolado, evitando problemas de dependências e configurações inconsistentes.

4. Considerações Finais

O desenvolvimento do Sistema de Comércio de Automóveis foi uma experiência rica em aprendizado e prática de conceitos fundamentais para o desenvolvimento de sistemas de informação modernos. Apesar dos desafios enfrentados, foi possível concluir o projeto com sucesso, entregando um sistema funcional e escalável.

Além do aprendizado técnico, este projeto também trouxe insights sobre a importância do planejamento e da comunicação entre os componentes do sistema. A utilização de práticas como a containerização e a separação entre backend e frontend demonstrou ser essencial para garantir a modularidade e a facilidade de manutenção.

Por fim, este projeto reafirmou a relevância do uso de ferramentas modernas e a aplicação de boas práticas no desenvolvimento de software. Os resultados alcançados são a base para futuros projetos mais complexos, e os aprendizados aqui obtidos servirão como guia para superar novos desafios.

5. Referências

- Django Project: <https://www.djangoproject.com>
- React Documentation: <https://reactjs.org>
- PostgreSQL Documentation: <https://www.postgresql.org>
- Docker Documentation: <https://docs.docker.com>
- Axios - Promise based HTTP client: <https://axios-http.com>