

**Centro Universitário - Católica de Santa Catarina**  
**Engenharia de software**



**RepasseAuto: Sistema de repasse de veículos  
voltado apenas a empresários do ramo automotivo**

Autor: Gustavo Lipinski

Professor: Luiz Carlos Camargo

Joinville

2024

# Resumo

Este trabalho aborda o desenvolvimento do sistema RepassAuto, uma solução de gerenciamento voltada para microempresas do setor automotivo, especificamente focada no repasse de veículos. A proposta surgiu da necessidade de enfrentar desafios como a reposição de estoque, a gestão ineficiente de ofertas e finanças, e a falta de visibilidade online para essas empresas.

## 1. Introdução

### 1.1 Contexto

Uma das maiores dificuldades dos empresários do ramo automotivo é justamente a reposição do seu estoque, com mercadorias de qualidade e com margem de valor para poder trabalhar com o produto.

### 1.2 Justificativa

Dada a lacuna de soluções de repasse específicas para o setor automotivo, o RepassAuto representa uma contribuição significativa para a modernização e profissionalização das microempresas automotivas.

### 1.3 Objetivos

Desenvolver um sistema que permita às microempresas automotivas terem um fácil acesso a um sistema de repasse de carros que tem como objetivo facilitar e otimizar o processo de compra e venda de veículos entre empresas do setor.

## 2. Descrição

### 2.1 Tema do Projeto

Este projeto trata do desenvolvimento de um sistema de gerenciamento para empresas do setor automotivo, especialmente focado no repasse de veículos. O sistema terá as seguintes funcionalidades:

- Espaço do vendedor
  - Criação e gerenciamento de ofertas de veículos:
    - No espaço do vendedor, será possível criar e gerenciar ofertas de veículos para repasse. Cada oferta incluirá informações detalhadas sobre os veículos, como modelos, quilometragem, condições, históricos de manutenção e preços
  - Controle de orçamentos

- O sistema permitirá a criação e gerenciamento de orçamentos, onde serão informados os materiais utilizados, a quantidade, as horas trabalhadas, a descrição dos serviços realizados, assim como o cliente para o qual o serviço será prestado.
  - Controle financeiro:
    - Lucro por períodos
    - Faturamento por períodos
    - Gastos por períodos
  - Controle de clientes:
    - Criar Usuários para clientes
    - Principais clientes
    - Quantidades de veículos repassados
    - Lucro por cliente
- Espaço do Cliente
  - Landing Page
    - A landing page terá imagens de portfólio, detalhes sobre a empresa e informações de contato
  - Dashboard de cliente
    - Na landing page, haverá um botão de login que redireciona o cliente para um dashboard. No dashboard, o cliente poderá:
      - Solicitar novos serviços.
      - Entrar em contato com o vendedor.
      - Visualizar veículos e orçamentos em aberto, em execução e finalizados.

## 2.2 Problemas a resolver

- Falta de visibilidade online para microempresas automotivas:
  - O sistema visa proporcionar uma plataforma que aumenta a visibilidade dessas empresas e facilite a descoberta de veículos disponíveis para repasse.
- Gestão ineficiente de ofertas e finanças:
  - O sistema propõe uma solução integrada para criar e gerenciar ofertas, além de monitorar o desempenho financeiro de forma precisa e eficiente.
- Comunicação dificultada com clientes:
  - O sistema visa simplificar e centralizar a comunicação, oferecendo um canal direto e claro para interações, solicitações de serviços e acompanhamento de trasações.

## 2.3 Limitações

- Pagamentos não incluídos:

- As transações financeiras devem ser conduzidas por meios externos ao sistema.
- Emissão de notas fiscais não suportadas:
  - As empresas deverão utilizar outros métodos para gerar e gerenciar suas notas fiscais.
- Consulta de valores dos materiais utilizados nos orçamentos:
  - Os usuários precisarão inserir essas informações manualmente.

### 3. Especificação técnica

#### 3.1 Requisitos do software

##### 3.1.1 Requisitos funcionais

N°	Título	Descrição
RF.1	Incluir Ofertas De Veículos	Incluir uma nova oferta de veículo para repasse.
RF.2	Alterar Oferta de Veículo	Alterar uma oferta de veículo pelo ID.
RF.3	Consultar Oferta de Veículo	Consultar dados de uma oferta de veículo pelo ID.
RF.4	Listar Ofertas de Veículos	Listar todas as ofertas de veículos, com filtros por período, cliente, e ordenação por data.
RF.5	Excluir Oferta de Veículo	Excluir uma oferta de veículo pelo ID.
RF.6	Imprimir Oferta de Veículo	Imprimir uma oferta de veículo pelo ID.
RF.7	Incluir Cliente	Incluir um novo cliente.
RF.8	Alterar Cliente	Alterar os dados de um cliente pelo ID.
RF.9	Consultar Cliente	Consultar os dados de um cliente pelo ID.
RF.10	Listar Cliente	Listar todos os clientes, com filtros por nome, em ordem alfabética
RF.11	Desativar Cliente	Desativar um cliente pelo ID.
RF.12	Calcular Faturamento	Calcular o faturamento por período com base nas ofertas aprovadas.
RF.13	Calcular Lucro	Calcular o lucro por período com base nos gastos de materiais e valores das ofertas aprovadas.
RF.14	Calcular Gasto	Calcular os gastos totais por período com base nos

N°	Título	Descrição
RF.1	Incluir Ofertas De Veículos	Incluir uma nova oferta de veículo para repasse.
RF.2	Alterar Oferta de Veículo	Alterar uma oferta de veículo pelo ID.
		materiais utilizados nas ofertas.
RF.15	Incluir Material	Incluir um novo material.
RF.16	Alterar Material	Alterar Material por ID.
RF.17	Consultar Material	Consultar Material por ID.
RF.18	Listar Material	Listar todos os materiais salvos na base, com filtro por nome do material.
RF.19	Excluir Material	Excluir material por ID.
RF.20	Login	Realizar login e redirecionar o usuário para o dashboard adequado.
RF.21	Principais Clientes	Exibir os clientes com maior recorrência de transações.
RF.22	Serviços Prestados	Exibir a quantidade de serviços prestados em um período.
RF.23	Lucro por Cliente	Exibir uma lista de clientes com o lucro individual, ordenado do maior para o menor.
RF.24	Configuração de Landing Page - Inclusão	Incluir dados a serem exibidos na landing page, como logo da empresa, imagens do portfólio e dados de contato.
RF.25	Configuração de Landing Page - Alteração	Alterar dados exibidos na landing page, como logo da empresa, imagens do portfólio e dados de contato.
RF.26	Configuração de Landing Page - Visualização	Visualizar os dados exibidos na landing page, como logo da empresa, imagens do portfólio e dados de contato.
RF.27	Configuração de Landing Page - Exclusão	Excluir dados exibidos na landing page, como logo da empresa, imagens do portfólio e dados de contato. Caso os dados sejam excluídos, serão exibidos dados padrões em tela.

### 3.1.2 Requisitos Não Funcionais

N°	Título	Descrição
RNF.1	Responsividade	O sistema deverá ser responsivo, adaptando-se a diferentes dispositivos e tamanhos de tela.
RNF.2	Experiência do Usuário	O sistema deverá ser intuitivo e fácil de usar, proporcionando uma boa experiência ao usuário.
RNF.3	Segurança	O sistema deverá contar com autenticação utilizando tokens JWT para garantir a segurança.
RNF.4	Criptografia	Dados sensíveis, como senhas, deverão ser criptografados ao serem salvos no banco de dados.

### 3.1.3 Caso de uso

[Clique para acessar em PNG](#)

[Clique para acessar em XML](#)(Necessário login do Draw.io)

## 3.2 Considerações de Design

### 3.2.1 Escolhas de Design

Optou-se pela arquitetura RESTful devido à sua simplicidade, eficiência e compatibilidade com a web. Essa arquitetura utiliza padrões e protocolos web estabelecidos, como HTTP e JSON, o que facilita a implementação e integração com diversas plataformas. Além disso, a natureza sem estado do RESTful permite uma escalabilidade eficaz, adequando-se perfeitamente a ambientes distribuídos e em nuvem. Comparativamente, SOAP e RPC, embora úteis em cenários que exigem protocolos rigorosos, tendem a ser mais complexos e

pesados, o que pode impactar negativamente a performance em aplicações modernas voltadas para a web.

### 3.2.2 Visão Inicial da Arquitetura

- API:
  - Responsável pela lógica de negócios e comunicação entre os front-ends e banco de dados.
- Landing Page:
  - Um front-end que exibe informações sobre a empresa, como portfólio, detalhes de contato e outros dados configurados.
- Sistema de Gerenciamento:
  - Outro front-end dedicado à gestão das ofertas de veículos, clientes, materiais, finanças e configurações de landing page.

### 3.2.3 Padrões de Arquitetura

Será seguido o padrão de arquitetura RESTful, utilizando uma API e dois front-ends isolados. Essa abordagem garante que cada componente seja independente e facilmente escalável, mantendo uma comunicação eficiente através de chamadas HTTP.

### 3.2.4 C4

[Clique aqui para acessar a modelagem C4](#)

## 3.3 Stack Tecnológica

### 3.3.1 Linguagens de Programação

Optamos por utilizar Java devido à sua portabilidade, permitindo que programas sejam executados em múltiplos dispositivos através da máquina virtual Java. A orientação a objetos de Java facilita a organização e reutilização de código, essenciais para construir aplicações complexas de forma eficiente. Além disso, Java oferece gerenciamento automático de memória e robustas ferramentas de detecção de erros, aumentando a estabilidade das aplicações. Com uma vasta comunidade de desenvolvedores e um ecossistema rico em bibliotecas e frameworks, como Spring e Hibernate, Java agiliza o desenvolvimento. Suas funcionalidades de segurança integradas também protegem contra ameaças externas, tornando Java uma escolha madura, segura e adaptável às necessidades empresariais.

O JavaScript é fundamental para o desenvolvimento de aplicações modernas devido à sua capacidade de execução tanto no cliente quanto no servidor. A linguagem oferece diversos frameworks como React e Angular, proporcionando interatividade em páginas web e melhorando significativamente a experiência do usuário. Com uma comunidade ativa e uma abundância de bibliotecas e frameworks disponíveis, JavaScript facilita o desenvolvimento rápido e eficaz de aplicações robustas e responsivas. Portanto, utilizar JavaScript permite aproveitar uma linguagem dinâmica e versátil, adequada para uma ampla variedade de projetos web e móveis.

### 3.3.2 Frameworks e Bibliotecas

Para o backend, escolhemos o Spring Boot, que simplifica a configuração e desenvolvimento da aplicação, permitindo um foco maior na lógica de negócio. Suas configurações automáticas e gerenciamento de dependências tornam o desenvolvimento mais ágil, enquanto sua integração com Spring Security garante a segurança da aplicação.

Para o frontend, utilizaremos React devido à sua estrutura organizada que suporta padrões de design como MVC, facilitando a organização e manutenção do código. A capacidade de componentização de React permite a reutilização de código e a criação de interfaces ricas e interativas de forma modular, tornando-o ideal para desenvolver aplicações de página única (SPAs).

### 3.3.3 Ferramentas de Desenvolvimento e Gestão do Projeto

Jira: Optamos pelo Jira para organizar e acompanhar as atividades do projeto. Seus quadros Kanban e Issues proporcionam uma visualização clara do progresso de cada tarefa, incluindo descrições, tempo estimado para conclusão e outras informações relevantes.

JUnit: Utilizaremos JUnit para a criação de testes unitários, pois é um framework amplamente utilizado, robusto e simples de implementar.

Heroku: A plataforma Heroku será utilizada para o deploy da aplicação. Heroku é conhecida por sua facilidade de uso e por fornecer uma solução completa de hospedagem, além de oferecer opções de hospedagem gratuita.

## 3.4 Considerações de Segurança



Aplicamos o padrão de Bearer Tokens utilizando JWT para garantir a integridade das autenticações. Além disso, utilizaremos o SonarQube para identificar pontos de melhoria e fragilidades no código. Testes unitários serão realizados com JUnit para assegurar a qualidade do código.

## 4. Próximos Passos

Após a finalização deste documento, seguiremos o plano de desenvolvimento a seguir:

- Organização das issues no Jira
- Desenvolvimento da API
- Criação dos testes unitários
- Desenvolvimento do frontend de gerenciamento
- Criação da Landing Page
- Deploy da aplicação no Heroku

## 5. Referências

ESCOLA DNC. Java: A linguagem poderosa para projetos de desenvolvimento. Disponível em: <https://www.escoladnc.com.br/blog/java-a-linguagem-poderosa-para-projetos-de-desenvolvimento/>. Acesso em: 12 maio 2024.

PROGRAMATHOR. Vantagens & Desvantagens da linguagem Java. Disponível em: <https://programathor.com.br/blog/vantagens-desvantagens-da-linguagem-java/>. Acesso em: 12 maio 2024.

MEDIUM. Porque você deve usar Spring Boot. Disponível em: <https://medium.com/@duduxss3/porque-você-deve-usar-spring-boot-a22f75da0955>. Acesso em: 12 maio 2024.

DIGITAL INNOVATION ONE. Por que usar Angular?. Disponível em: <https://www.dio.me/articles/por-que-usar-angular>. Acesso em: 12 maio 2024.

ONE DAY TESTING. SonarQube: melhorando a qualidade e a padronização do seu código. Disponível em: <https://blog.onedaytesting.com.br/sonarqube-melhorando-a-qualidade-e-a-padronizacao-do-seu-codigo/>. Acesso em: 12 maio 2024.

DEV MEDIA. JUnit - Tutorial. Disponível em: <https://www.devmedia.com.br/junit-tutorial/1432>. Acesso em: 12 maio 2024.

IMAGINE DONE. Heroku vale a pena utilizar a plataforma?. Disponível em: <https://imaginedone.com.br/en/artigos/web-e-mobile/heroku-vale-a-pena-utilizar-a-plataforma/>. Acesso em: 12 maio 2024.

Considerações Professor/a:

Considerações Professor/a:

Considerações Professor/a: