

# MecSystem: Sistema de Gestão para Oficinas Mecânicas de Pequeno Porte

Daniel Gandolfi, Rogério Xavier de Azambuja

Instituto Federal do Rio Grande do Sul - Câmpus Farroupilha - Av. São Vicente, 785 -  
CEP 95174-274 - Bairro Cinquentenário - Farroupilha - RS - Brasil

[danielgandolfi.10@gmail.com](mailto:danielgandolfi.10@gmail.com), [rogerio.xavier@farroupilha.ifrs.edu.br](mailto:rogerio.xavier@farroupilha.ifrs.edu.br)

**Abstract.** *This paper presents MecSystem, a Web-based system developed to optimize the management of small automotive repair shops. Its main purpose is to centralize the registration of clients, vehicles, and services, in addition to automating processes such as scheduling, budget creation, and service order issuance, with a focus on security and efficiency. The system was developed using technologies such as Next.js, Spring Boot, and PostgreSQL, aiming to improve organization, reduce errors, and increase productivity. The project also seeks to provide an intuitive and efficient interface, ensuring a smooth user experience in desktop-oriented web environments.*

**Keywords:** *Web development, service management, auto repair shops, process automation.*

**Resumo.** *Este artigo apresenta o MecSystem, um sistema Web desenvolvido para otimizar a gestão de oficinas mecânicas de pequeno porte. O objetivo principal é centralizar o cadastro de clientes, veículos e serviços, além de automatizar processos como agendamentos, criação de orçamentos e emissão de ordens de serviço, com foco na segurança e na eficiência. O sistema foi desenvolvido utilizando tecnologias como Next.js, Spring Boot e PostgreSQL, buscando melhorar a organização, reduzir erros e aumentar a produtividade. A proposta também visa oferecer uma interface intuitiva e eficiente, garantindo uma experiência fluida em ambientes Web para desktop.*

**Palavras-chave:** *desenvolvimento Web, gestão de serviços, oficinas mecânicas, automação de processos.*

## 1. Introdução

O setor automotivo desempenha um papel importante na economia moderna, garantindo a movimentação de pessoas e bens em larga escala. Nesse cenário, as oficinas mecânicas desempenham papel fundamental na manutenção e no funcionamento adequado dos veículos e frotas, assegurando segurança e eficiência.

Dados do setor apontam que o Brasil tem cerca de 117.857 oficinas de reparação automotiva, a maioria de pequeno porte, o que mostra a importância desse segmento para a economia do país (Peças e Veículos, 2023). Apesar disso, grande parte dessas

empresas trabalha com processos manuais, o que dificulta a organização interna e pode afetar a qualidade do atendimento.

Para atender essa necessidade, foi desenvolvido o MecSystem, um sistema Web voltado para ajudar na organização das oficinas mecânicas. Ele reúne em um só lugar o cadastro de clientes, veículos e serviços, além de permitir o uso de recursos como agendamentos e controle de ordens de serviço. O sistema também conta com autenticação para garantir a segurança das informações. A ideia é oferecer uma interface simples, moderna e fácil de usar, ajudando a oficina a manter tudo mais organizado no dia a dia.

Com isso, este Trabalho de Conclusão de Curso (TCC) tem como objetivo geral desenvolver uma aplicação Web que centralize e otimize os processos internos de uma oficina mecânica de pequeno porte, aumentando o controle operacional e a eficiência dos serviços. Para atingir esse objetivo, foram definidos os seguintes objetivos específicos: centralizar o cadastro de clientes, veículos e serviços; implementar funcionalidades para o gerenciamento de ordens de serviço e agendamentos; garantir a segurança dos dados processados por meio de autenticação; e adotar tecnologias modernas que proporcionem uma interface amigável e responsiva.

A escolha desse tema se deve ao fato de muitas oficinas de pequeno porte ainda trabalharem de forma manual, o que costuma gerar erros, perda de informações e dificuldade de controle. A digitalização ajuda a evitar esses problemas e deixa os processos mais rápidos e organizados. Além disso, estudos mostram que o uso de ferramentas digitais é um fator importante para melhorar a produtividade e o atendimento em pequenos negócios. Segundo o SEBRAE, sistemas de gestão e soluções digitais têm impacto direto na eficiência e na organização das empresas (SEBRAE, 2024a).

No restante deste trabalho é apresentado primeiramente o referencial teórico, onde são discutidos conceitos e estudos sobre digitalização de processos e gestão em oficinas mecânicas. Em seguida, são descritos os procedimentos metodológicos usados no desenvolvimento do projeto, incluindo os requisitos e as tecnologias aplicadas. Por fim, como o sistema ainda não foi implantado, são apresentados os resultados obtidos durante o desenvolvimento e os testes feitos em ambiente local, além das referências utilizadas.

## **2. Referencial Teórico**

O referencial teórico reúne os principais conceitos utilizados no desenvolvimento do sistema de gestão para oficinas mecânicas. A pesquisa abordou temas como gestão de serviços, tecnologias voltadas ao setor automotivo e formas de melhorar processos internos. Esses conteúdos servem para contextualizar o sistema e explicar as escolhas feitas no projeto.

## **2.1. Gestão de Oficinas Mecânicas**

O setor automotivo tem grande impacto na economia, principalmente na área de manutenção e reparo de veículos. Nas oficinas mecânicas, a boa gestão dos serviços influencia diretamente a experiência do cliente e os resultados do negócio.

Segundo o SEBRAE (2021b), pequenos negócios como oficinas precisam organizar seus processos para melhorar a qualidade dos serviços e se manter competitivos. Isso inclui controle de estoque, agendamentos e treinamento da equipe, além do uso de ferramentas tecnológicas.

## **2.2. Gestão de Serviços no Setor Automotivo**

A gestão de serviços no setor automotivo envolve tanto aspectos técnicos quanto a forma como o cliente percebe o atendimento. Um dos modelos usados para avaliar essa percepção é o SERVQUAL, criado por Parasuraman, Zeithaml e Berry (1985). Ele analisa dimensões como aparência da empresa, confiabilidade, rapidez no atendimento, segurança e atenção ao cliente.

Um estudo de Diehl (2015) mostra que, nas oficinas mecânicas, as dimensões mais valorizadas pelos clientes são a confiabilidade e a rapidez no atendimento, reforçando a importância de processos bem organizados.

## **2.3. Satisfação do Cliente e Fidelização**

A satisfação do cliente é essencial em serviços automotivos. Conforme Oliver (1980), ela ocorre quando o serviço recebido atende ou supera as expectativas do cliente. Para isso, é importante oferecer reparos de qualidade, boa comunicação e prazos claros. A fidelização surge quando o cliente confia no serviço e percebe consistência no atendimento, algo que pode ser reforçado com o uso de sistemas de gestão.

## **2.4. O Impacto da Tecnologia na Gestão de Oficinas Mecânicas**

A adoção de tecnologias em oficinas mecânicas tem revolucionado a gestão, proporcionando uma série de benefícios, como maior eficiência operacional e redução de erros humanos. Timmons (2016) destaca que os sistemas de gestão modernos possibilitam o controle de estoque, o agendamento de serviços, o faturamento e até a previsão de falhas em veículos com base em dados históricos e sensores de manutenção. Essa tecnologia não só otimiza as operações, mas também melhora a experiência do cliente, permitindo um atendimento mais ágil e preciso.

## **2.5. Boas Práticas de Desenvolvimento de Software**

A Engenharia de Software é uma disciplina que reúne princípios, métodos e boas práticas voltados ao desenvolvimento sistemático e controlado de sistemas computacionais. Conforme Sommerville (2019), a Engenharia de Software é composta por processos que começam na definição dos requisitos e seguem até a manutenção, garantindo que o sistema final seja confiável e atenda ao que foi planejado. Pressman (2021) destaca que um software de qualidade depende de um projeto bem elaborado, apoiado em uma arquitetura sólida, documentação clara e técnicas que promovam a reutilização e a facilidade de manutenção.

A qualidade do código é um fator determinante para o sucesso e a sustentabilidade de sistemas de software. Segundo Martin (2009), em Código Limpo, um código de fácil leitura, com funções bem definidas e responsabilidades claras, reduz o acúmulo de erros, facilita a manutenção e melhora a produtividade a longo prazo. Os princípios apresentados por Martin enfatizam aspectos como nomes significativos, simplicidade na estrutura de métodos, organização modular e ausência de duplicidade, garantindo que o software permaneça compreensível e evolutivo.

## **2.6. Arquitetura Web e APIs REST**

Segundo Fielding (2000), o estilo arquitetural Representational State Transfer (REST) define princípios para comunicação entre sistemas usando recursos identificados por Uniform Resource Identifiers (URIs), utilizando os verbos do protocolo HyperText Transfer Protocol (HTTP). Aplicações que seguem esse modelo ganham padronização, baixo acoplamento e facilidade de evolução. O uso correto dos métodos HTTP e dos códigos de status também ajuda na manutenção, nos testes e no monitoramento das Application Programming Interfaces (APIs).

## **2.7. Autenticação e Autorização com Auth0**

De acordo com a Auth0 (2025), as plataformas de autenticação como serviço oferecem soluções completas para controle de identidade e acesso, baseadas em padrões de segurança amplamente reconhecidos. O Auth0 é uma dessas plataformas, fornecendo um sistema de autenticação centralizado que permite gerenciar logins, permissões e tokens de acesso com segurança e escalabilidade. Segundo a documentação oficial, sua arquitetura baseia-se em protocolos modernos e no uso do JSON Web Token (JWT) para validação das sessões de usuário.

## **2.8. Banco de Dados Relacional e PostgreSQL**

De acordo com a PostgreSQL Global Development Group (2025), o PostgreSQL é um sistema gerenciador de banco de dados relacional que se destaca por sua conformidade

com o padrão Structured Query Language (SQL), estabilidade e suporte a transações: Atomicidade, Consistência, Isolamento e Durabilidade (ACID). Por meio de índices, chaves estrangeiras e integridade referencial, o PostgreSQL garante consistência e desempenho nas operações, sendo amplamente utilizado em aplicações corporativas de alta confiabilidade.

## **2.9. Segurança e Proteção de Dados**

Segundo a Lei nº 13.709/2018, conhecida como Lei Geral de Proteção de Dados (LGPD), o tratamento de dados pessoais deve seguir princípios como a necessidade, a finalidade, a segurança e a transparência. Essas diretrizes exigem que sistemas de informação adotem medidas técnicas e administrativas que assegurem a proteção e o uso responsável dos dados, incluindo o controle de acesso, a minimização de coleta e o registro de operações, reforçando a importância da conformidade legal em sistemas que tratam de informações sensíveis.

## **2.10. Usabilidade em Sistemas de Gestão**

Para Nielsen (1994), usabilidade é a facilidade com que o usuário aprende e usa uma interface. Em sistemas de gestão, isso é fundamental, pois telas claras e processos simples ajudam diretamente no desempenho e na satisfação do usuário.

## **3. Procedimentos Metodológicos**

Nesta seção, são descritas as etapas seguidas para alcançar os objetivos propostos, incluindo a seleção das tecnologias, ferramentas e metodologias que embasam o desenvolvimento do MecSystem. O projeto foi desenvolvido com foco em automação, organização e escalabilidade.

### **3.1. Tecnologias**

O desenvolvimento do MecSystem utilizou ferramentas já consolidadas no mercado, escolhidas pela eficiência e pela confiabilidade. Durante a construção do sistema, seus conceitos foram estudados e aplicados de forma prática, garantindo uma aplicação consistente e resultando em uma solução completa alinhada às necessidades do setor.

#### **3.1.1. Front-end**

- **React.js**

React (React, 2025) é uma biblioteca JavaScript amplamente utilizada para a construção de interfaces de usuário. Sua escolha se deu pela modularidade, facilidade de manutenção e reutilização de componentes, o que torna a aplicação mais organizada e consistente, permitindo identificar visualmente pontos-chave do código.

- **Next.js**

O *framework* Next.js (Next.js, 2025) foi escolhido por suas vantagens em renderização do lado do servidor conhecida como *Server Side Rendering* (SSR) e geração de páginas estáticas, *Static Site Generation* (SSG), resultando em uma aplicação mais rápida e com melhor desempenho em *Search Engine Optimization* (SEO). Além disso, o Next.js simplifica a configuração de rotas e otimiza a performance, sendo uma solução ideal para o escopo do projeto.

- **TypeScript**

Com a adoção do TypeScript (TypeScript, 2025), foi possível implementar tipagem estática no código, reduzindo erros durante o desenvolvimento e facilitando a manutenção a longo prazo. A ferramenta melhora significativamente a confiabilidade do sistema ao oferecer maior controle sobre as definições de tipos.

- **Material UI (MUI)**

O Material UI (MUI) (Material UI, 2025) segue os princípios do Material Design e oferece componentes prontos e personalizáveis, o que facilita a criação de interfaces modernas e responsivas. A biblioteca ajuda na montagem de telas claras e bem organizadas, priorizando a usabilidade e uma boa experiência para o usuário em ambiente Web.

### 3.1.2. Backend

- **Java com Spring Boot**

O desenvolvimento *back-end* do MecSystem foi implementado na linguagem Java, utilizando o Spring Boot (Spring Boot, 2025), um *framework* que facilita o desenvolvimento de aplicações robustas e escaláveis. Sua integração com o Spring Data JPA (Java Persistence API) facilita o mapeamento objeto-relacional e a persistência de dados no banco de dados PostgreSQL, reduzindo a complexidade do código e aumentando a produtividade.

O sistema expõe APIs RESTful que garantem comunicação eficiente e segura com o *front-end*, utilizando boas práticas de versionamento e controle de respostas do Hypertext Transfer Protocol (HTTP).

- **Auth0 (Autenticação e Autorização)**

Para a autenticação e controle de acesso dos usuários, foi implementada a integração com a plataforma Auth0 (Auth0, 2025), que oferece um serviço de identidade moderno e seguro.

O processo de autenticação é realizado por meio do Auth0 Universal Login (Auth0, 2025), um serviço hospedado na própria plataforma que centraliza o fluxo de login dos usuários. Após a autenticação, o Auth0 emite um JWT, que é utilizado nas requisições às APIs do sistema. Esse token é validado pelo Spring Boot, assegurando

que apenas usuários autenticados tenham acesso aos recursos protegidos. Com isso, o sistema ganha mais segurança, controle de permissões e rastreabilidade dos acessos.

### **3.1.3. Banco de Dados**

- **PostgreSQL**

O PostgreSQL foi adotado como sistema de gerenciamento de banco de dados relacional por sua estabilidade, desempenho e conformidade com os padrões SQL. Ele oferece suporte a consultas simples e complexas, controle transacional completo (ACID) e integridade referencial, requisitos essenciais para o armazenamento de informações de clientes, veículos, serviços e ordens de serviço.

### **3.1.4. Gerenciamento de Ambientes**

O ambiente de desenvolvimento do MecSystem foi configurado localmente, sem o uso de contêineres, utilizando ferramentas nativas das tecnologias empregadas: *Maven* (Maven, 2025) para gerenciamento de dependências no *back-end* e *npm* (npm, 2025) no *front-end*.

Essa abordagem facilitou o processo de desenvolvimento, depuração e integração entre as camadas da aplicação, mantendo a simplicidade e eficiência do ambiente. Além disso, o projeto segue boas práticas de versionamento com *Git* (Git, 2025), permitindo o controle de versões e a colaboração durante o desenvolvimento.

### **3.1.5. Considerações adicionais sobre as tecnologias**

A escolha das tecnologias do MecSystem levou em conta desempenho, escalabilidade, segurança e facilidade de manutenção. Cada ferramenta foi selecionada pensando no contexto do projeto e na necessidade de oferecer uma solução moderna para oficinas de pequeno porte.

A combinação entre Next.js, React, Spring Boot, PostgreSQL e Auth0 garante uma aplicação estável, segura e com arquitetura flexível, capaz de evoluir de forma contínua conforme as necessidades do sistema.

Essas tecnologias são amplamente usadas e têm forte suporte da comunidade, o que traz confiabilidade e facilita a manutenção a longo prazo. Isso ajuda o MecSystem a se manter atualizado e preparado para crescer no futuro.

## **4. Desenvolvimento**

Nesta seção, são apresentados os métodos de desenvolvimento e detalhadas as etapas práticas para a realização do trabalho.

#### 4.1. Análise e Classificação dos Requisitos

Os requisitos para o desenvolvimento do MecSystem foram levantados e classificados com base em entrevistas e observações realizadas em uma oficina mecânica localizada no município de Farroupilha (RS), em conjunto com o responsável pelo estabelecimento.

##### 4.1.1. Requisitos Funcionais (RF):

- **RF01:** Permitir o cadastro de clientes com nome, telefone, e-mail e endereço.
- **RF02:** Permitir o cadastro de veículos com informações como placa, modelo, marca e ano.
- **RF03:** Registrar os serviços realizados, vinculando-os ao veículo e ao cliente correspondente.
- **RF04:** Criar orçamentos contendo a descrição dos itens e os valores estimados.
- **RF05:** Converter orçamentos aprovados em ordens de serviço, permitindo sua edição até a finalização.
- **RF06:** Exibir na *dashboard* o resumo dos serviços realizados, com visualização filtrada por dia, mês e ano.
- **RF07:** Permitir o agendamento de serviços, com data, horário e vínculo ao cliente e veículo.
- **RF08:** Controlar o status das ordens de serviço (em andamento, concluída ou cancelada).

##### 4.1.2. Requisitos Não Funcionais (RNF):

- **RNF01:** O sistema deve ser acessível via navegadores Web modernos.
- **RNF02:** Assegurar a proteção dos dados por meio de autenticação via Auth0 e do uso de token JWT.
- **RNF03:** As respostas da interface devem ter um tempo médio de resposta inferior a 1 segundo.

##### 4.1.3. Requisitos de Sistema (RS):

- **RS01:** Utilizar banco de dados relacional PostgreSQL para o armazenamento das informações.
- **RS02:** Implementar APIs RESTful para a integração entre o *front-end* e o *back-end*.
- **RS03:** Utilizar arquitetura cliente-servidor, permitindo escalabilidade e manutenção simplificada.

##### 4.1.4. Regras de Negócio (RN):

- **RN01:** Orçamentos têm validade de 30 dias.

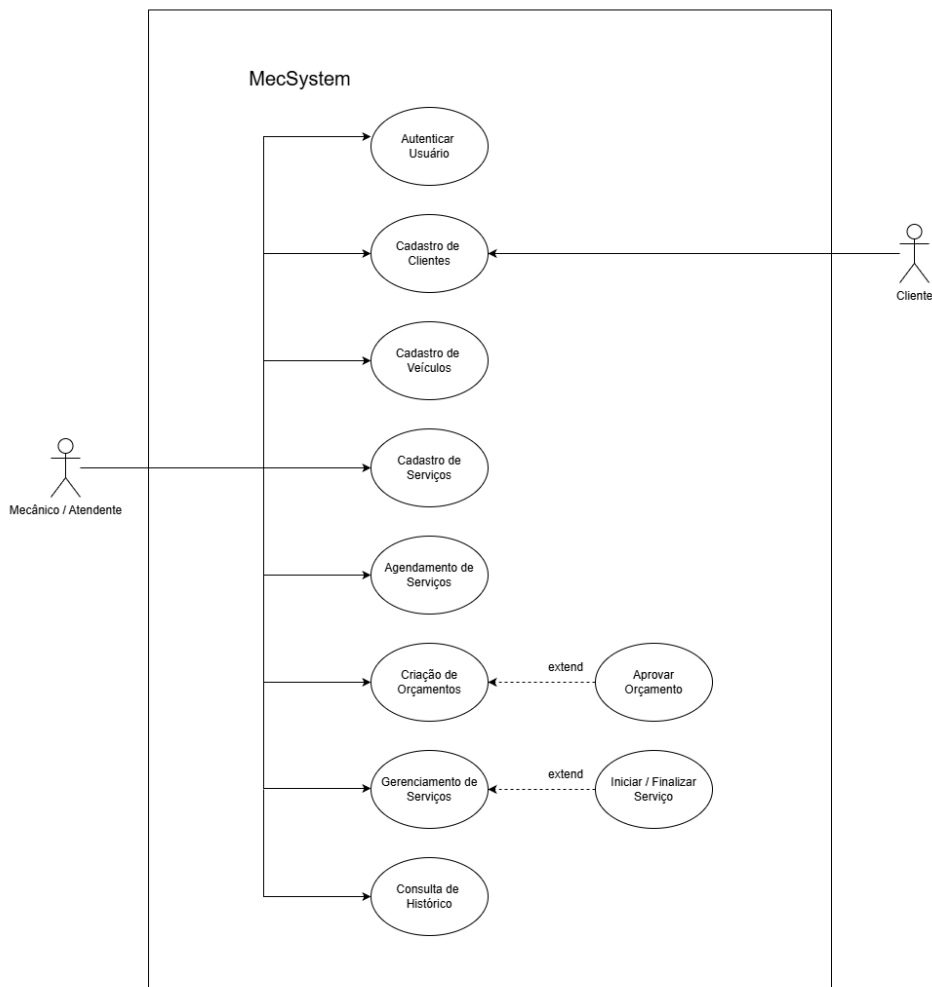
- **RN02:** Ordens de serviço só podem ser criadas após aprovação do orçamento.
- **RN03:** Cada ordem de serviço deve estar associada a um cliente e a um veículo previamente cadastrados.
- **RN04:** O *status* dos serviços deve refletir o andamento das atividades, permitindo o acompanhamento operacional por meio da *dashboard*.

## 4.2. Modelagem

Após o detalhamento dos requisitos do sistema, será apresentada nesta seção, a modelagem do sistema.

### 4.2.1. Caso de Uso

A Figura 1 apresenta o diagrama de caso de uso do MecSystem, ilustrando as principais interações entre o gestor da oficina e as funcionalidades do sistema. Este modelo destaca como as atividades essenciais de gestão foram estruturadas no contexto do sistema.



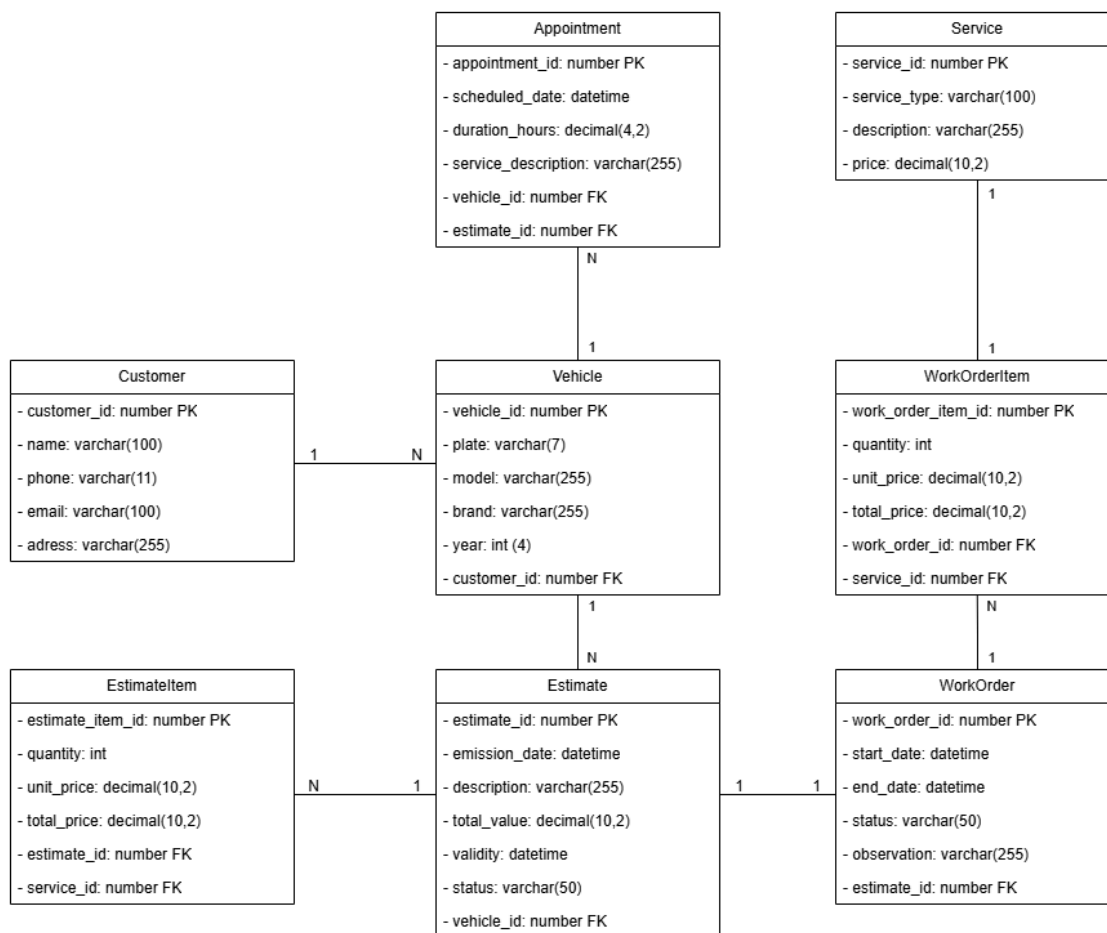
**Figura 1: Diagrama de Casos de Uso do MecSystem**

**Fonte: Elaboração própria, 2025.**

O diagrama de caso de uso apresenta o mecânico ou atendente como ator principal, responsável por operar as principais funcionalidades do sistema. Entre elas, destacam-se o cadastro de clientes e de veículos, a criação e aprovação de orçamentos, o agendamento e o gerenciamento de serviços e a consulta de históricos de atendimentos. Essas funcionalidades permitem o controle completo das atividades da oficina, desde o registro inicial até a finalização dos serviços, o que melhora a organização, facilita o acompanhamento e aumenta a eficiência do trabalho diário.

#### 4.2.2. Entidade e Relacionamento

A seguir, na Figura 2, é apresentado o modelo de entidade-relacionamento (E-R). Este modelo fornece uma visão estruturada dos principais elementos do sistema e das interações entre eles, demonstrando como os dados estão organizados e conectados para atender às necessidades de gestão das oficinas mecânicas.



**Figura 2: Modelo Entidade e Relacionamento do MecSystem**

**Fonte: Elaboração própria, 2025.**

O modelo de entidade e relacionamento do MecSystem representa as principais estruturas de dados e os vínculos entre os componentes do sistema. As entidades Cliente, Veículo, Serviço, Orçamento, Item de Orçamento, Ordem de Serviço, Item de

Ordem de Serviço e Agendamento compõem o núcleo da aplicação, refletindo os processos essenciais de gestão de uma oficina mecânica.

Cada cliente pode possuir um ou mais veículos cadastrados, e cada veículo pode estar associado a orçamentos, agendamentos e ordens de serviço. Os orçamentos registram os serviços previstos e seus valores estimados, que podem ser convertidos em ordens de serviço após aprovação. As ordens de serviço detalham as atividades executadas, incluindo os serviços realizados, os valores e as observações pertinentes.

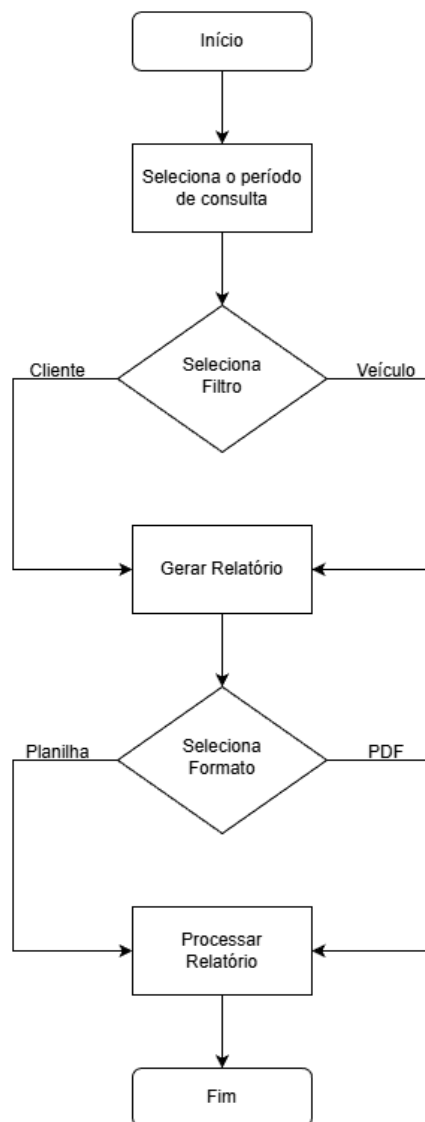
A entidade Serviço descreve as opções disponíveis de manutenção e reparo, sendo referenciada por itens de orçamento e de ordem de serviço. A entidade Agendamento permite o controle de horários e de disponibilidade da oficina, vinculando o serviço ao veículo do cliente.

Como a autenticação dos usuários é realizada por meio do Auth0, o sistema não mantém uma tabela interna de usuários. Em vez disso, cada operação pode registrar o identificador do usuário autenticado, proveniente do token JWT, garantindo rastreabilidade e controle de ações no sistema.

Essa estrutura assegura a integridade referencial, consistência e organização dos dados, permitindo o gerenciamento eficiente das informações e o acompanhamento das atividades operacionais de forma centralizada.

#### **4.2.3. Processo de Geração de Relatórios**

A Figura 3 apresenta uma representação do processo de geração de relatórios previsto para o MecSystem. Esse fluxo descreve as etapas básicas para selecionar o período de consulta, escolher o filtro entre cliente ou veículo, gerar o relatório e definir o formato de exportação, que pode ser planilha ou PDF.



**Figura 3: Diagrama de Relatórios**

**Fonte: Elaboração própria, 2025.**

Embora essa funcionalidade ainda não esteja implementada, ela faz parte da evolução planejada do sistema e complementa as operações já existentes, ampliando seu valor gerencial.

### 4.3. Protótipo e Desenvolvimento do Sistema

Após a modelagem conceitual e definição dos requisitos, foi desenvolvido o protótipo funcional do MecSystem, que evoluiu para a aplicação final. O sistema foi construído com base na arquitetura cliente-servidor, integrando toda a camada de *front-end* da aplicação, desenvolvida em Next.js e React, ao *back-end* construído em Spring Boot, com banco de dados PostgreSQL.

O objetivo central do desenvolvimento foi disponibilizar uma aplicação responsiva, intuitiva e eficiente para o gerenciamento das atividades de uma oficina mecânica, abrangendo o controle de clientes, veículos, orçamentos, ordens de serviço e agendamentos.

### 4.3.1. Protótipo e Interfaces

O protótipo inicial do MecSystem foi desenvolvido com base nos requisitos levantados, priorizando clareza, fluidez e eficiência de uso. As interfaces foram projetadas com base em princípios de usabilidade, garantindo uma navegação simples e objetiva, com foco nas operações diárias da oficina.

As principais telas do sistema são apresentadas nas Figuras 4 a 7, contendo a *Dashboard*, os Agendamentos, o módulo de Orçamentos e a tela de Ordens de Serviço. Cada uma delas desempenha um papel fundamental no fluxo operacional da aplicação, abrangendo o processo completo, desde o cadastro e o planejamento até a execução dos atendimentos.

Além das interfaces principais, o sistema conta com os módulos de cadastro, edição e exclusão dos clientes, veículos e serviços, garantindo que as informações cheguem atualizadas na interação dos próximos processos.

Os módulos de Itens de Orçamento e de Itens de Ordem de Serviço possibilitam o detalhamento dos serviços executados, especificando quantidades, valores unitários e totais. Essa estrutura garante transparência nas informações apresentadas ao cliente, além de facilitar o controle operacional da oficina.

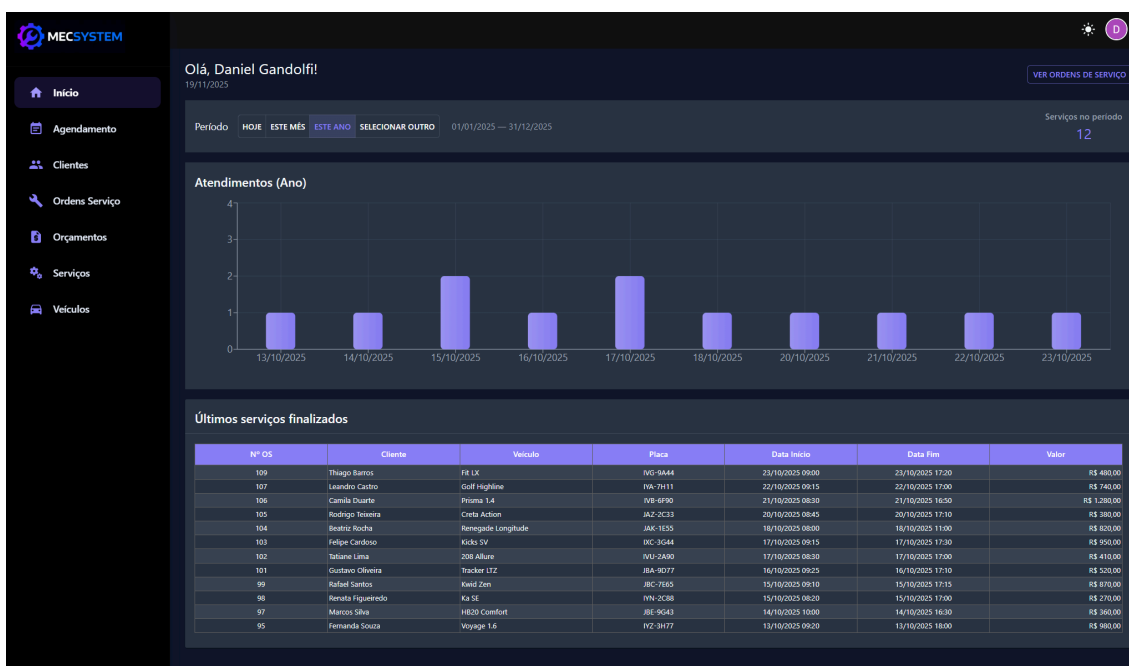


Figura 4: Tela principal (Dashboard)

Fonte: Elaboração própria, 2025.

A *Dashboard* do MecSystem apresenta uma visão consolidada das principais atividades da oficina, permitindo ao gestor acompanhar o desempenho operacional em diferentes períodos. Na parte superior da interface, é possível selecionar intervalos de tempo como “Hoje”, “Este Mês”, “Este Ano” ou definir um período personalizado, o que atualiza dinamicamente os dados exibidos.

O painel principal contém um gráfico de barras que representa a quantidade de atendimentos realizados por dia dentro do período selecionado, facilitando a visualização do fluxo de serviços ao longo do tempo. Logo abaixo, a *Dashboard* apresenta uma tabela com os serviços finalizados mais recentes, incluindo informações como número da ordem de serviço, cliente, veículo, placa, datas de início e término, status e valor total.

Essa composição visual oferece ao gestor um panorama claro e objetivo das operações recentes, apoiando a tomada de decisão e o acompanhamento do desempenho da oficina.

The screenshot shows the 'Agendamentos' (Appointments) interface. At the top right, there is a button 'ADICIONAR AGENDAMENTO'. Below it, there are filters for 'Período' (set to 'Personalizado'), 'Início' (01/09/2025), and 'Fim' (31/10/2025), along with a 'LIMPAR FILTROS' button. There are also buttons for 'EXPORTAR', 'COLUNAS', and a search bar. The main table contains 8 rows of appointment data. At the bottom, it shows 'Mostrando 1 - 8 de 8 registros' and navigation controls for 'Anterior', 'Página 1 de 1', 'Próximo', and a page size dropdown set to '30'.

	Nº	Data	Duração	Descrição do Serviço	Veículo	Placa	Valor	Nº Orçamento	Cliente
✎	38	30/10/2025 12:00	3 horas	Revisão preventiva completa	Onix LT	IVX-5D23	R\$ 380,00	12	Carlos Menezes
✎	37	28/09/2025 13:00	3 horas	Troca de fluido de direção e freio	Versa Advance	JCA-7H30	R\$ 610,00	36	Eduardo Ramos
✎	36	23/09/2025 12:00	4 horas	Troca de correia dentada e bomba d'água	HB20S Vision	JBG-6E22	R\$ 970,00	35	Larissa Melo
✎	35	18/09/2025 11:00	1 horas	Troca de palhetas e limpeza de vidro	Polo MSI	JAD-8B99	R\$ 210,00	34	André Carvalho
✎	34	14/09/2025 13:00	4 horas	Troca de embreagem parcial e regulagem	S10 LTZ	IVW-4D12	R\$ 720,00	33	Maurício Pereira
✎	33	10/09/2025 12:30	6 horas	Troca de junta do cabeçote	EcoSport Titanium	JAS-3H10	R\$ 1.450,00	32	Sabrina Lopes
✎	32	06/09/2025 11:00	4 horas	Revisão de freios e limpeza geral	Sanderó Stepway	IVP-2C66	R\$ 850,00	31	Ricardo Almeida
✎	31	02/09/2025 12:00	2 horas	Troca de fluido de transmissão	Cronos Precision	JAE-7F32	R\$ 370,00	30	Patrícia Souza

**Figura 5: Interface de agendamento de serviços**

**Fonte: Elaboração própria, 2025.**

A interface de Agendamentos permite o controle e o acompanhamento das atividades programadas na oficina, exibindo todos os serviços marcados na agenda conforme o período selecionado. Nela são apresentadas informações detalhadas sobre cada agendamento, como data e horário, descrição do serviço, veículo, placa, cliente, valor estimado e tempo previsto de execução, o que auxilia na distribuição equilibrada das tarefas entre os mecânicos e evita sobreposição de horários.

Além disso, a tela oferece filtros por período, opções para limpar os filtros ativos, exportar a lista de agendamentos e adicionar novos registros, facilitando a organização e o planejamento das atividades futuras. Essa funcionalidade contribui para uma gestão mais eficiente da carga de trabalho, garantindo previsibilidade no atendimento e melhor aproveitamento dos recursos da oficina.

Orçamentos ADICIONAR ORÇAMENTO +

Período: Mês Status: Todos LIMPAR FILTROS

EXPORTAR COLUMNAS

	Nº	Status	Data Criação	Descrição	Valor	Validade	Veículo	Placa	Cliente
>	86	Aprovado	19/10/2025	Revisão completa de 80.000 km	R\$ 1.150,00	26/10/2025	Versa Advance	JCA-7H30	Eduardo Ramos
>	85	Pendente	18/10/2025	Troca de amortecedores dianteiros	R\$ 750,00	25/10/2025	HB20S Vision	JBG-6E22	Larissa Melo
>	84	Pendente	17/10/2025	Troca de fluido e ajuste de freio	R\$ 450,00	24/10/2025	Polo MSI	JAD-8B99	André Carvalho
>	83	Pendente	16/10/2025	Revisão completa com troca de filtros	R\$ 860,00	23/10/2025	S10 LTZ	IVW-4D12	Maurício Pereira
>	82	Pendente	15/10/2025	Troca de óleo e filtro de ar	R\$ 420,00	22/10/2025	EcoSport Titanium	JAS-3H10	Sabrina Lopes
>	81	Aprovado	14/10/2025	Troca de bomba d'água e correia	R\$ 970,00	21/10/2025	Sandero Stepway	IVP-2C66	Ricardo Almeida
>	80	Pendente	13/10/2025	Troca de embreagem completa	R\$ 1.350,00	20/10/2025	Cronos Precision	JAE-7F32	Patrícia Souza
>	79	Pendente	12/10/2025	Troca de fluido de direção e alinhamento	R\$ 390,00	19/10/2025	Fit LX	IVG-9A44	Thiago Barros
>	78	Pendente	11/10/2025	Troca de palhetas e limpeza de vidro	R\$ 210,00	18/10/2025	Yaris XL	JBB-8E70	Juliana Freitas
>	77	Pendente	10/10/2025	Troca de amortecedores dianteiros	R\$ 780,00	17/10/2025	Golf Highline	IVA-7H11	Leandro Castro
>	76	Pendente	09/10/2025	Troca de junta e vedação	R\$ 1.280,00	16/10/2025	Prisma 1.4	IVB-6F90	Camila Duarte
>	75	Pendente	08/10/2025	Troca de palhetas e vidros	R\$ 330,00	15/10/2025	Creta Action	JAZ-2C33	Rodrigo Teixeira
>	74	Pendente	07/10/2025	Revisão de freios e fluido	R\$ 510,00	14/10/2025	Renegade Longitude	JAK-1E55	Beatriz Rocha
>	73	Cancelado	06/10/2025	Troca de correia e tensionador	R\$ 440,00	13/10/2025	Kicks SV	IXC-3G44	Felipe Cardoso
>	72	Aprovado	05/10/2025	Troca de embreagem completa	R\$ 1.150,00	12/10/2025	208 Allure	IVU-2A90	Tatiane Lima
>	71	Pendente	04/10/2025	Troca de amortecedores traseiros	R\$ 720,00	11/10/2025	Tracker LTZ	JBA-9D77	Gustavo Oliveira
>	70	Pendente	03/10/2025	Troca de lâmpadas e fusíveis	R\$ 280,00	10/10/2025	T-Cross Highline	JDF-1B90	Amanda Costa
>	69	Pendente	02/10/2025	Revisão completa 60.000 km	R\$ 950,00	09/10/2025	Kwid Zen	JBC-7E65	Rafael Santos
>	68	Aprovado	01/10/2025	Troca de fluido de freio	R\$ 360,00	08/10/2025	Ka SE	IYN-2C88	Renata Figueiredo
>	62	Aprovado	26/10/2025	Troca de correia dentada	R\$ 480,00	02/11/2025	Onix LT	IVX-5D23	Carlos Menezes
>	61	Pendente	25/10/2025	Troca de fluido de freio e direção	R\$ 630,00	01/11/2025	Versa Advance	JCA-7H30	Eduardo Ramos

**Figura 6: Tela gerenciamento de orçamentos**

**Fonte: Elaboração própria, 2025.**

O módulo de Orçamentos é responsável pelo controle das estimativas de serviços solicitados pelos clientes. Nessa tela, o gestor pode cadastrar novos orçamentos, visualizar os existentes e acompanhar o status de cada um, que pode variar entre pendente, aprovado ou cancelado.

Cada orçamento contém informações detalhadas, como a descrição do serviço, a data de criação, o valor total, a validade, o veículo e o cliente associado. O sistema também permite filtrar os registros por período e status, além de exportar a listagem para consulta externa. Quando um orçamento é aprovado, ele é automaticamente convertido em uma ordem de serviço, mantendo o vínculo entre os itens orçados e os serviços executados. Esse processo reduz erros manuais e garante a integridade das informações ao longo das etapas do atendimento.

Ordem de Serviços

Período: Personalizado Início: 01/10/2025 Fim: 31/10/2025 Status: Todos LIMPAR FILTROS

EXPORTAR COLUNAS Buscar...

	Nº	Status	Data Início	Data Fim	Observação	Valor	Veículo	Placa	Cliente
>	113	Pendente	-	-	Revisão completa e troca de fluido realizadas.	R\$ 540,00	Polo MSI	JAD-8899	André Carvalho
>	112	Pendente	-	-	Bomba d'água e correia sendo substituídas.	R\$ 870,00	EcoSport Titanium	JAS-3H10	Sabrina Lopes
>	111	Pendente	-	-	Palhetas e limpeza de vidro concluídas.	R\$ 210,00	Sandero Stepway	IVP-2C66	Ricardo Almeida
>	110	Pendente	-	-	Revisão elétrica e troca de fusíveis em andamento.	R\$ 950,00	Cronos Precision	JAE-7F32	Patrícia Souza
>	109	Finalizado	23/10/2025 09:00	23/10/2025 17:20	Troca de óleo de motor e filtro realizada.	R\$ 480,00	Fit LX	IVG-9A44	Thiago Barros
>	108	Pendente	-	-	Troca de fluido de transmissão em execução.	R\$ 360,00	Yaris XL	JBB-8E70	Juliana Freitas
>	107	Finalizado	22/10/2025 09:15	22/10/2025 17:00	Revisão de arrefecimento concluída.	R\$ 740,00	Golf Highline	IVA-7H11	Leandro Castro
>	106	Finalizado	21/10/2025 08:30	21/10/2025 16:50	Embreamento e alinhamento finalizados.	R\$ 1.280,00	Prisma 1.4	IVB-6F90	Camila Duarte
>	105	Finalizado	20/10/2025 08:45	20/10/2025 17:10	Troca de correia e tensionador realizada.	R\$ 380,00	Creta Action	JAZ-2C33	Rodrigo Teixeira
>	104	Finalizado	18/10/2025 08:00	18/10/2025 11:00	Amortecedores traseiros substituídos.	R\$ 820,00	Renegade Longitude	JAK-1E55	Beatriz Rocha
>	103	Finalizado	17/10/2025 09:15	17/10/2025 17:30	Revisão de freios e suspensão concluída.	R\$ 950,00	Kicks SV	IXC-3G44	Felipe Cardoso
>	102	Finalizado	17/10/2025 08:30	17/10/2025 17:00	Troca de palhetas e limpeza de vidro finalizada.	R\$ 410,00	208 Allure	IVU-2A90	Tatiane Lima
>	101	Finalizado	16/10/2025 09:25	16/10/2025 17:10	Troca de pastilhas e fluido concluída.	R\$ 520,00	Tracker LTZ	JBA-9D77	Gustavo Oliveira
>	100	Cancelado	16/10/2025 08:15	16/10/2025 16:45	Serviço cancelado pelo cliente.	R\$ 1.350,00	T-Cross Highline	JDF-1B90	Amanda Costa
>	99	Finalizado	15/10/2025 09:10	15/10/2025 17:15	Revisão elétrica e limpeza de bicos executada.	R\$ 870,00	Kwid Zen	JBC-7E65	Rafael Santos
>	98	Finalizado	15/10/2025 08:20	15/10/2025 17:00	Troca de fluido de direção realizada.	R\$ 270,00	Ka SE	IVN-2C88	Renata Figueiredo
>	97	Finalizado	14/10/2025 10:00	14/10/2025 16:30	Troca de velas e filtro de combustível concluída.	R\$ 360,00	HB20 Comfort	JBE-9G43	Marcos Silva
>	96	Em Andamento	14/10/2025 08:40	-	Troca de embreamento completa em processo.	R\$ 1.120,00	Corolla XEi	IXW-8C22	Luciana Alves

**Figura 7: Interface de gerenciamento de ordens de serviço**

Fonte: Elaboração própria, 2025.

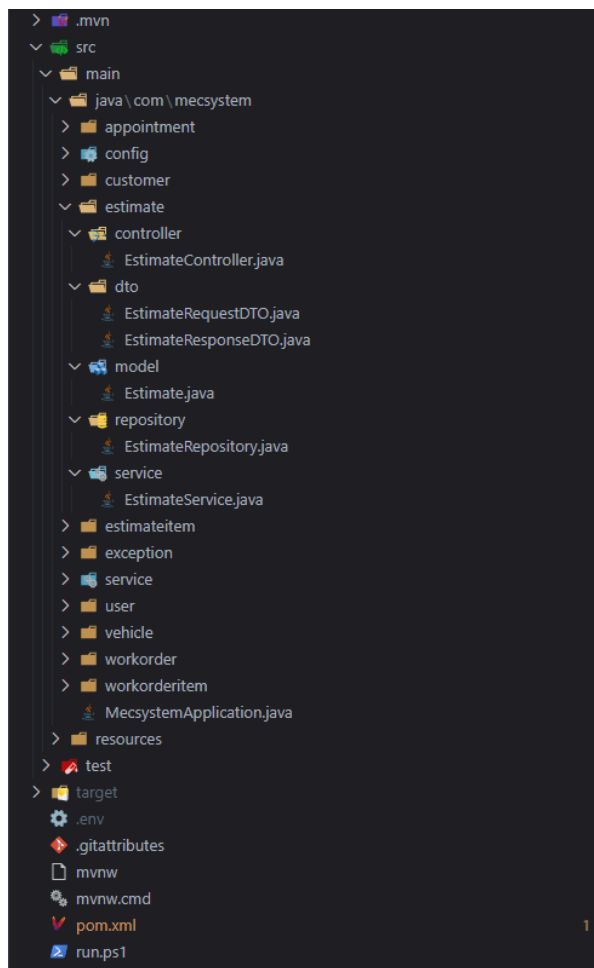
A interface Ordens de Serviço representa a etapa central do fluxo operacional do sistema. Nela são exibidas todas as ordens geradas a partir de orçamentos aprovados, permitindo o acompanhamento completo do status de cada serviço, sendo eles: Pendente, Em Andamento, Finalizado ou Cancelado.

O painel apresenta informações como data de início e término, descrição do serviço realizado, observações inseridas pelo mecânico, valor total, veículo atendido, placa e cliente associado. Também estão disponíveis filtros por período e status, além da opção de exportar os dados para controle externo ou para análises administrativas.

Esse módulo garante o monitoramento detalhado da execução dos serviços, permitindo que o gestor acompanhe a evolução de cada atendimento e identifique gargalos no processo produtivo da oficina.

#### 4.3.2. Trechos de Código-Fonte

Os trechos a seguir representam partes essenciais do funcionamento do sistema. O primeiro exemplo mostra a estrutura de uma rota de API REST desenvolvida com Spring Boot, responsável pelo gerenciamento de clientes:



**Figura 8: Estrutura de pastas do back-end**

**Fonte: Elaboração própria, 2025.**

A Figura 8 apresenta a estrutura *back-end* do sistema, que foi organizada conforme o padrão *Model-View-Controller* (MVC) e os princípios de responsabilidade única. A aplicação foi estruturada em pacotes distintos: *controller*, *service*, *repository*, *dto* e *model*, separando claramente as camadas de controle, negócio, persistência e transferência de dados.

```

@RestController
@RequestMapping("/estimates")
public class EstimateController {
    @Autowired
    private EstimateService service;

    @PostMapping
    public ResponseEntity<Map<String, Object>> create(@RequestBody EstimateRequestDTO dto) {
        Long id = service.insertEstimate(dto);

        Map<String, Object> response = new HashMap<>();
        response.put(key:"estimate_id", id);

        return ResponseEntity.ok().body(response);
    }

    @PatchMapping("/{id}")
    public ResponseEntity<Void> update(@PathVariable("id") Long id, @RequestBody EstimateRequestDTO estimate) {
        service.updateEstimate(id, estimate);
        return ResponseEntity.noContent().build();
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> delete(@PathVariable("id") Long id) {
        service.deleteEstimate(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public List<EstimateResponseDTO> findAll() {
        return service.findAll();
    }

    @GetMapping("/{id}")
    public ResponseEntity<EstimateResponseDTO> findById(@PathVariable("id") Long id) {
        EstimateResponseDTO estimate = service.findById(id);
        return ResponseEntity.ok(estimate);
    }
}

```

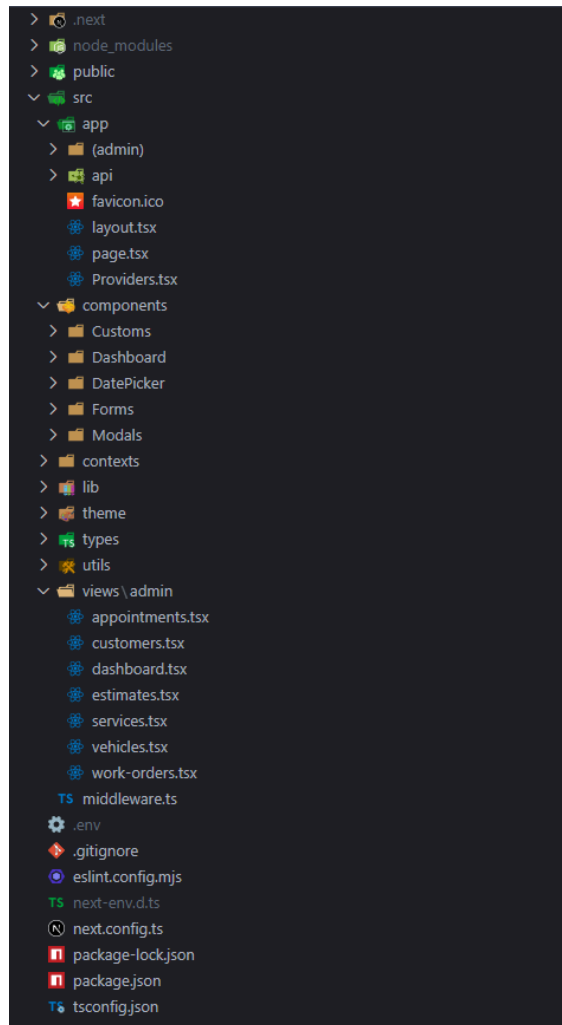
**Figura 9: Controlador REST responsável pelo gerenciamento de orçamentos**

**Fonte: Elaboração própria, 2025.**

A Figura 9 exibe o código-fonte da classe *EstimateController*, responsável pelo gerenciamento das rotas da *API REST* do módulo de Orçamentos. Essa classe utiliza as anotações *@RestController* e *@RequestMapping* para definir os endpoints e segue os princípios *RESTful*, permitindo a comunicação entre o *front-end* e o *back-end* por meio dos métodos *HTTP GET*, *POST*, *PATCH* e *DELETE*.

Cada rota apresentada é tratada de forma independente, utilizando a classe *ResponseEntity* para padronizar as respostas e retornar os códigos de status apropriados, como *200 OK*, *201 Created* e *204 No Content*. O controlador repassa as operações de negócio à camada de serviço, que realiza a validação dos dados e a interação com o repositório responsável pelo acesso ao banco de dados PostgreSQL.

A seguir, são apresentados exemplos que ilustram a estrutura e a lógica de funcionamento do módulo.



**Figura 10: Estrutura de pastas do front-end**

**Fonte: Elaboração própria, 2025.**

A Figura 10 apresenta a estrutura de diretórios do *front-end* do sistema MecSystem, desenvolvido com Next.js e TypeScript. O projeto foi organizado de forma modular, visando a reutilização de componentes, a clareza no fluxo de dados e a manutenção simplificada.

A pasta *components* reúne os elementos visuais reutilizáveis da interface, como botões, formulários e tabelas. As *views* contêm as telas principais do sistema, como Clientes, Veículos, Serviços, Orçamentos e Ordens de Serviço. Já o *contexts* centraliza os estados globais, incluindo o controle de autenticação e a exibição de mensagens, e a pasta *lib/api* armazena as configurações para comunicação com o *back-end*.

```

return (
  <Grid container>
    <Grid size={12}>
      <CustomCard
        title={<Typography variant="h5">Clientes</Typography>}
        subtitle={
          <Button
            variant="contained"
            endIcon={<AddBoxIcon />}
            onClick={() => setOpenModalAddEdit(true)}
          >
            Adicionar Cliente
          </Button>
        }
      >
      {loading ? (
        <Skeleton height={300} />
      ) : (
        <CustomTable data={data} columns={columns} />
      )}
    </CustomCard>
  </Grid>
  <ModalAddEdit
    open={openModalAddEdit}
    onClose={handleCloseModalAddEdit}
    title={dataRow ? "Editar Cliente" : "Adicionar Cliente"}
    dataRow={dataRow}
    renderForm={(data) => (
      <CustomerForm
        defaultValues={data || null}
        onSubmit={(values) => {
          if (data) {
            handleEdit(values);
          } else {
            handleSubmit(values);
          }
        }}
      />
    )}
  />
  <ModalDelete
    open={openModalDelete}
    onClose={handleCloseModalDelete}
    title="Excluir cliente"
    message={`o cliente ${dataRow?.name} ?`}
    onConfirm={handleDelete}
  />
</Grid>
);
}

```

Figura 11: Trecho do componente de gerenciamento de clientes

Fonte: Elaboração própria, 2025.

A Figura 11 apresenta um trecho do código-fonte da interface Clientes, implementada em *React* com *TypeScript* e *Material UI*. Esse componente é responsável

por renderizar a interface de listagem e de gerenciamento de clientes, incluindo a exibição da tabela principal e os modais de adição, edição e exclusão de registros.

O código mostra a utilização de componentes reutilizáveis, como *CustomCard* e *CustomTable*, que padronizam o visual e reduzem a duplicação de código. Além disso, o uso de modais dinâmicos e condicionais de renderização demonstra uma abordagem reativa, na qual o estado da aplicação define o comportamento da interface.

### 4.3.3. Boas Práticas de Desenvolvimento

Este trabalho foi guiado pelos princípios descritos no livro Código Limpo (Clean Code) de Robert C. Martin, que apresenta as boas práticas de programação como uma base sólida na construção de software, respeitando a clareza, a modularidade e a manutenção do código. Cada ponto serviu como uma inspiração para tornar o código de fácil compreensão e expansão. Dentre esses pontos, destacam-se:

- Nomes significativos para classes, funções e variáveis, facilitando a leitura e o entendimento do código;
- Funções curtas e coesas, responsáveis por uma única tarefa;
- Separação de responsabilidades, seguindo o princípio de *Single Responsibility Principle* (SRP);
- Eliminação de duplicidades, priorizando a reutilização e abstração de componentes;
- Tratamento adequado de exceções, garantindo estabilidade nas operações;

A aplicação consistente desses princípios contribuiu para a qualidade geral do sistema, reduzindo a complexidade, facilitando futuras manutenções e garantindo conformidade com padrões profissionais de engenharia de software.

## 4.4. Resultados e Discussão

O desenvolvimento do MecSystem resultou em uma aplicação Web funcional voltada à gestão de oficinas mecânicas de pequeno porte. O sistema atende aos requisitos definidos neste projeto e cumpre os objetivos propostos, ajudando no controle das principais operações da oficina.

Durante os testes em ambiente local, o sistema se mostrou estável e com bom desempenho. A navegação entre as telas foi simples e o tempo de resposta ficou baixo, o que deixou o uso do sistema mais ágil. Todas as funcionalidades principais, como cadastro de clientes e veículos, criação de orçamentos, conversão em ordens de serviço e controle de agendamentos, funcionaram conforme o planejado, validando a modelagem e a arquitetura estabelecidas. Os testes foram conduzidos de forma manual e automatizada, incluindo verificações do fluxo completo das funcionalidades, testes de integração entre o *front-end* e a API e testes de comportamento do banco de dados durante operações de leitura e escrita. Embora não tenha sido possível realizar testes

com usuários reais da oficina, os cenários simulados permitiram avaliar o funcionamento geral do sistema e identificar pontos de melhoria para etapas futuras.

O módulo de *Dashboard* oferece uma visão consolidada das atividades da oficina, permitindo acompanhar o volume de serviços diários, mensais e anuais. Essa funcionalidade contribui para o planejamento das operações e para o acompanhamento do desempenho geral da empresa. Além disso, a integração com o Auth0 garante a segurança do acesso, assegurando que apenas usuários autenticados possam acessar os dados do sistema.

Com base nas funcionalidades implementadas e nos testes realizados, o MecSystem mostrou que pode ajudar na organização e na digitalização dos processos internos da oficina. Entre os benefícios mais relevantes observados destacam-se:

- Centralização das informações, eliminando registros manuais e reduzindo falhas no controle de serviços;
- Organização operacional, permitindo o acompanhamento completo de clientes, veículos, orçamentos e ordens de serviço;
- Indicativos de eficiência no atendimento, com acesso rápido às informações e histórico de serviços;
- Segurança dos dados, garantida pela autenticação via Auth0 e uso de tokens JWT;
- Facilidade de manutenção do código, obtida pelo uso de boas práticas de desenvolvimento;
- Escalabilidade, permitindo futuras expansões, como integração com sistemas financeiros ou relatórios analíticos.

Como não foi possível realizar testes com usuários reais, os testes preliminares indicam ganhos qualitativos relevantes, como a padronização das rotinas e a maior organização das informações. Com isso, o MecSystem apresenta viabilidade técnica e tem potencial para melhorar a gestão de oficinas mecânicas.

Além disso, também foi percebida a necessidade de um módulo de relatórios. O fluxo dessa funcionalidade já foi representado no diagrama apresentado na Figura 3 deste trabalho, mostrando como deverá funcionar nas próximas versões.

## **5. Conclusão**

O desenvolvimento do MecSystem possibilitou a criação de uma aplicação Web voltada à modernização do gerenciamento de oficinas mecânicas de pequeno porte. O projeto atendeu aos objetivos propostos, oferecendo uma solução que centraliza informações e automatiza processos antes realizados manualmente.

O uso de tecnologias consolidadas, como Spring Boot, PostgreSQL, Next.js e Auth0, contribuiu para a construção de um sistema seguro, estável e com uma boa

estrutura de manutenção. A aplicação das boas práticas em “Código Limpo” como separação adequada de responsabilidades, padronização de nomenclaturas, modularização e organização das camadas de software favoreceu a legibilidade do código-fonte, facilitou o entendimento das rotinas implementadas e reduziu a probabilidade de defeitos.

Além de atender aos requisitos funcionais, o sistema demonstra potencial para evolução, permitindo futuras integrações com módulos financeiros, relatórios analíticos e implantação em nuvem. Esses avanços poderão ampliar ainda mais a utilidade prática do projeto e fortalecer seu papel como ferramenta de apoio à gestão de pequenas empresas do setor automotivo.

Por fim, o MecSystem evidencia a importância do uso de tecnologias modernas e metodologias bem estruturadas no desenvolvimento de sistemas corporativos, contribuindo para a digitalização de processos e para a eficiência operacional no contexto das oficinas mecânicas. A construção da aplicação também reforçou a utilização dos conhecimentos adquiridos ao longo da formação em Análise e Desenvolvimento de Sistemas, integrando fundamentos de programação, modelagem de dados, engenharia de software e tecnologias Web com as práticas atuais da comunidade de desenvolvimento de software.

Como trabalhos futuros, está prevista a implementação do módulo de relatórios, que deve complementar o que já foi desenvolvido e oferecer mais recursos de análise para a oficina. Também faz parte do planejamento colocar o sistema em ambiente produtivo, realizar testes com usuários reais e criar um módulo de controle de estoque. Mais adiante, uma versão mobile também poderá ser considerada. Esses avanços devem ajudar na evolução do MecSystem e torná-lo ainda mais completo no uso diário.

## 6. Referências

APACHE. **Maven Documentation**. 2025. Disponível em: <https://maven.apache.org>. Acesso em: 13 nov. 2025.

AUTH0. **Auth0 Documentation**. Disponível em: <https://auth0.com/docs>. Acesso em: 25 out. 2025.

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)**. Diário Oficial da União, Brasília, DF, 15 ago. 2018.

DIEHL, A. **Qualidade em serviços em oficinas mecânicas**. SciELO, 2015. Disponível em: <https://www.scielo.br>. Acesso em: 30 nov. 2024.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000.

GIT. **Git Documentation**. 2025. Disponível em: <https://git-scm.com>. Acesso em: 10 nov. 2025.

MARTIN, Robert C. **Código Limpo: Habilidades Práticas do Agile Software**. Rio de Janeiro: Alta Books, 2009.

NIELSEN, J. **10 Heuristics for User Interface Design**. Nielsen Norman Group, 1994. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso em: 25 out. 2025.

MATERIAL UI. **Material UI - Overview**. Disponível em: <https://mui.com/material-ui/getting-started>. Acesso em: 03 nov. 2024.

NEXT.JS. **Introduction to Next.js**. Disponível em: <https://nextjs.org/docs>. Acesso em: 03 nov. 2024.

NPM. **npm Documentation**. 2025. Disponível em: <https://www.npmjs.com>. Acesso em: 10 nov. 2025.

OLIVER, R. L. **A cognitive model of the antecedents and consequences of satisfaction decisions.** *Journal of Marketing Research*, v. 17, n. 4, p. 460-469, 1980.

PARASURAMAN, A.; ZEITHAML, V. A.; BERRY, L. L. **SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality.** *Journal of Retailing*, v. 64, n. 1, p. 12-40, 1985.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation.** Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 25 out. 2025.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: Uma Abordagem Profissional.** 9. ed. Porto Alegre: AMGH, 2021.

REACT. **React Documentation.** Disponível em: <https://react.dev>. Acesso em: 25 out. 2025.

SEBRAE. **Gestão de Pequenos Negócios. 2024a.** Disponível em: <https://www.sebrae.com.br>. Acesso em: 30 nov. 2024.

SEBRAE. **Transformação Digital nos Pequenos Negócios. 2024b.** Disponível em: <https://www.sebrae.com.br>. Acesso em: 10 nov. 2025.

SPRING BOOT. **Spring Boot.** Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 03 nov. 2024.

TIMMONS, J. A. **Gestão de Processos em Negócios de Pequeno Porte.** Disponível em: <https://www.fia.com.br>. Acesso em: 30 nov. 2024.

TYPESCRIPT. **TypeScript Documentation.** Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 25 out. 2025.