

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**MATHEUS DA SILVA GOMES
PATRICK LUIZ SILVA**

**APP IFRETE
Intermediação de Fretes Urbanos para Redução da Ociosidade**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Alexandre Gomes da Silva

**FRANCA/SP
2025**

APP IFRETE

Matheus da Silva Gomes¹

Patrick Luiz Silva²

Resumo

O transporte de cargas intraurbano contribui para a economia de cidades de médio porte, como as do interior de São Paulo, Brasil. Neste contexto, caminhoneiros autônomos que realizam fretes urbanos enfrentam desafios como a informalidade na captação de serviços, perda de tempo e baixa rentabilidade. Este projeto propõe o desenvolvimento de um aplicativo móvel para intermediação específica de fretes urbanos dentro da mesma cidade, visando conectar motoristas e contratantes de forma prática, segura e eficiente. A solução busca reduzir a ociosidade nos deslocamentos urbanos e facilitar a contratação de fretes locais e rápidos por meio de funcionalidades como geolocalização precisa, comunicação direta e filtros de busca adequados ao contexto urbano (tipo de veículo, capacidade, distâncias reduzidas). Com base em levantamento de requisitos reais e metodologias ágeis, o aplicativo foi concebido como um MVP funcional, desenvolvido com React Native, JavaScript e MongoDB. O projeto demonstra a viabilidade técnica e social da proposta para o nicho de fretes urbanos, buscando contribuir com a modernização de um segmento local historicamente marcado pela informalidade e carente de soluções digitais. Ao otimizar a conexão dentro do perímetro urbano, o aplicativo visa gerar ganhos operacionais e melhorar o dia a dia dos profissionais envolvidos assim como dos contratantes.

Palavras-chave: Aplicativo móvel. Cidades médias. Fretes urbanos. Intermediação de serviços. Ociosidade de caminhões.

Abstract

Urban cargo transportation is essential to the economy of medium-sized cities, such as those in the interior of São Paulo, Brazil. In this context, self-employed truck drivers performing urban freight face challenges such as informal service contracting, time loss, and low profitability. This project proposes the development of a mobile application specifically for urban freight intermediation within the same city, aiming to connect drivers and clients in a practical, secure, and efficient manner. The solution seeks to reduce idle time in urban routes and facilitate the hiring of local and short-distance freight services through features such as precise geolocation, direct communication, and search filters tailored to the urban context (vehicle type, capacity, and short distances). Based on real user requirements and agile methodologies, the application was designed as a functional MVP, developed using React Native, JavaScript, and MongoDB. The project demonstrates the technical and social viability of the proposed solution for the urban freight niche, contributing to the modernization of a traditionally informal and underserved segment. By optimizing local connections

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: matheus.recin@outlook.com

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: patrick.silva106@fatec.sp.gov.br

within the urban perimeter, the application aims to improve operational efficiency and enhance the quality of life for professionals involved in these specific cities..

Keywords: *Mobile application. Medium-sized cities. Urban freight. Service intermediation. Truck idleness.*

1 Introdução

A logística de transporte de cargas constitui um setor relevante para a economia, sobretudo em regiões onde o comércio local depende diretamente do deslocamento terrestre de mercadorias. Contudo, a realidade enfrentada por muitos caminhoneiros autônomos ainda é marcada pela informalidade na captação de serviços, o que acarreta incertezas, perda de tempo e baixa rentabilidade.

O presente trabalho tem como tema o desenvolvimento de uma solução digital voltada à intermediação de fretes urbanos, com o intuito de conectar caminhoneiros e contratantes de forma ágil, segura e eficiente. Diante desse contexto, a questão-problema que orienta esta pesquisa é: como reduzir a ociosidade dos caminhoneiros e otimizar a contratação de fretes por meio de uma solução tecnológica acessível e funcional?

Parte-se da hipótese de que o uso de um aplicativo móvel, com funcionalidades específicas para a publicação, busca e negociação de fretes, pode atenuar significativamente os desafios enfrentados tanto por motoristas quanto por clientes. O objetivo geral do projeto é desenvolver uma plataforma que centralize essas interações de forma intuitiva e confiável. Entre os objetivos específicos, destacam-se: identificar as reais necessidades dos usuários, implementar funcionalidades como geolocalização e canais de comunicação diretos, além de garantir a usabilidade e acessibilidade da solução proposta. A justificativa para esta iniciativa reside na carência de soluções tecnológicas eficientes no setor local de transportes, o que representa uma oportunidade para inovação social. Além disso, a adesão dos caminhoneiros a recursos tecnológicos já é significativa: segundo pesquisa da CNT (2019), 98% utilizam smartphones para acesso à internet, e 55,4% utilizam a internet com fins profissionais. Esses dados reforçam a viabilidade de uma solução digital que facilite o contato entre motoristas e contratantes.

Este trabalho apresenta-se como relevante por propor uma inovação aplicável a um setor tradicionalmente informal e desassistido por recursos tecnológicos. A proposta visa contribuir para o desenvolvimento regional, impactando positivamente a qualidade de vida dos motoristas e a eficiência das atividades comerciais que dependem de transporte de mercadorias.

Quanto à metodologia, foram utilizadas abordagens qualitativas, com levantamento de requisitos fundamentado em estudos de caso, observações empíricas e análises baseadas em experiências reais relatadas por profissionais da área. Ferramentas de modelagem como as matrizes SWOT e 5W2H foram aplicadas para estruturar os aspectos estratégicos do projeto, além da notação BPMN para mapear os fluxos de funcionamento da solução. O desenvolvimento técnico foi realizado utilizando as tecnologias React Native, JavaScript e MongoDB, com foco na construção de um MVP funcional, responsivo e escalável.

Este projeto está organizado em sete capítulos, os quais foram estruturados de forma lógica para apresentar, justificar, desenvolver e avaliar a solução proposta frente ao problema identificado.

O Capítulo 1 – Introdução contextualiza a problemática enfrentada por motoristas na busca por fretes e apresenta a proposta do aplicativo como uma solução tecnológica. Neste capítulo são definidos o tema, a questão-problema, os objetivos, a justificativa, a relevância social e acadêmica do trabalho, além da metodologia adotada e da organização dos capítulos seguintes.

O Capítulo 2 – Viabilidade do Projeto avalia a viabilidade técnica e operacional da solução desenvolvida. São analisados os riscos, as restrições e os recursos necessários para garantir que o projeto seja exequível e sustentável dentro do escopo proposto.

O Capítulo 3 – Levantamento de Requisitos apresenta as necessidades do público-alvo com base em relatos reais e modelagem progressiva. São descritos os requisitos funcionais e não funcionais do sistema, além das técnicas utilizadas para capturar e documentar essas informações.

O Capítulo 4 – Ferramentas e Métodos descreve os recursos técnicos empregados no desenvolvimento do sistema, como linguagens de programação, banco de dados, plataformas de desenvolvimento e padrões arquiteturais, além das metodologias utilizadas para organização do projeto.

O Capítulo 5 – Desenvolvimento relata como foi construída a aplicação, destacando as principais etapas da implementação, os desafios enfrentados, as decisões técnicas e o funcionamento das funcionalidades principais do sistema.

O Capítulo 6 – Resultados e Discussão apresenta os resultados obtidos com a entrega do MVP (Produto Mínimo Viável), destacando a validação parcial da solução, os retornos obtidos junto aos usuários-alvo e a análise crítica do que foi alcançado em relação aos objetivos propostos.

Por fim, o Capítulo 7 – Considerações Finais, Referências e Anexos retoma os principais pontos abordados, apresenta as conclusões obtidas, propõe futuras melhorias, traz as referências bibliográficas utilizadas ao longo do projeto e os anexos complementares, como entrevistas e modelos de telas.

1.1 Termo da Abertura do Projeto (TAP)

O Termo de Abertura do Projeto (TAP) é um documento que autoriza formalmente o início de um projeto. Ele concede ao gerente de projetos a autoridade para aplicar os recursos organizacionais nas atividades do projeto e estabelece as bases para o sucesso ao definir claramente os objetivos do projeto, suas restrições e premissas, responsabilidades e as expectativas das partes interessadas (Pmi, 2021).

Esse artefato detalha as etapas necessárias ao desenvolvimento do software, consolidando informações essenciais sobre o contexto do projeto, como a justificativa de negócio, os requisitos preliminares e os riscos potenciais identificados. Ao reunir esses elementos, o TAP proporciona uma visão integrada dos recursos e esforços requeridos para alcançar os objetivos propostos, servindo como base para o planejamento e a execução (Kerzner, 2017).

A importância deste artefato está descrita no PMBOK Guide como um dos processos essenciais para iniciação. Ele é fundamental para fornecer uma compreensão clara do escopo, dos objetivos e das expectativas do projeto, o que ajuda a minimizar os desvios e assegurar o alinhamento com as necessidades do negócio (Pmi, 2021).

1.1.1 Situação Atual

Atualmente, muitos caminhoneiros da cidade enfrentam dificuldades para encontrar fretes de forma eficiente. Grande parte depende do boca a boca ou de se posicionar fisicamente em locais estratégicos, como postos ou centros de carga, aguardando ser contratado. Isso gera incerteza, tempo ocioso e custos extras com deslocamento, sem garantia de serviço.

Motoristas usam redes sociais, grupos de mensagens e contatos informais para buscar oportunidades. Já os clientes, como comércios e pessoas físicas, geralmente ligam para conhecidos ou procuram por indicações. Esses métodos são pouco organizados, geram atrasos, exigem tempo e muitas vezes não são eficazes para encontrar o transporte adequado no momento certo.

As soluções atuais são descentralizadas, informais e imprevisíveis. Caminhoneiros perdem tempo sem retorno garantido e muitas vezes aceitam fretes mal remunerados por necessidade. Já os contratantes podem enfrentar dificuldade em encontrar caminhoneiros disponíveis, confiáveis ou com o tipo de veículo necessário para sua carga.

A digitalização dos serviços tem crescido em todas as áreas, e o setor de transporte local ainda carece de uma solução tecnológica eficiente e acessível. Resolver esse problema agora pode gerar benefícios econômicos, otimizar a logística local, facilitar a vida de motoristas e clientes, além de uma nova oportunidade de negócio sustentável e escalável na região.

1.1.2 Justificativa do Projeto

Por meio de análise do setor de transporte de carga, foram identificados pontos fracos críticos que comprometem sua eficiência, como viagens de retorno vazias, dificuldades na busca por cargas e motoristas, comunicação ineficiente entre empresas, motoristas e clientes, e a falta de parcerias estratégicas. Esses problemas representam oportunidades de melhoria que justificam o desenvolvimento de um aplicativo para conectar fretistas a pessoas e empresas. A importância de uma solução sistêmica reside na capacidade de integrar todos os envolvidos motoristas, clientes e empresas em uma plataforma única, eliminando problemas operacionais e criando oportunidades de parcerias antes inacessíveis.

1.2.3 Propósito do Projeto e Metas

A implementação do aplicativo de fretes tem como objetivo principal simplificar e otimizar o processo de conexão entre motoristas de caminhão e pessoas ou empresas que necessitam de transporte de cargas. A proposta surge da necessidade de enfrentar desafios recorrentes do setor, como a dificuldade de encontrar serviços de forma eficiente, a comunicação informal e ineficaz entre as partes envolvidas, e o desperdício de tempo e recursos com deslocamentos sem garantia de retorno.

Para resolver essas questões, o projeto busca oferecer uma solução acessível, moderna e funcional. As metas envolvem o desenvolvimento de uma interface amigável, que torne o uso da plataforma fácil e rápido, favorecendo a adesão tanto de motoristas quanto de contratantes. Também se pretende promover uma comunicação mais direta e eficiente entre as partes, garantindo agilidade nos contatos e nas negociações.

Além disso, o projeto visa fortalecer a confiança entre usuários, por meio da validação de identidades e do acompanhamento da qualidade dos serviços prestados. Com o uso de recursos de geolocalização, será possível melhorar o rastreamento e a segurança das cargas durante o transporte. Outro ponto importante será a criação de um sistema de avaliações e histórico de serviços, contribuindo para a construção de uma rede de reputação confiável e transparente.

1.2.4 Descrição do Projeto

O projeto consiste no desenvolvimento de um aplicativo mobile para intermediar serviços de frete entre caminhoneiros e clientes da cidade, visando solucionar a dificuldade de encontrar ou divulgar serviços de transporte de forma eficiente. A proposta contempla o planejamento, construção e entrega de uma plataforma digital que conecte motoristas e contratantes de forma ágil, transparente e segura, promovendo a valorização do trabalho local e otimizando o tempo e os recursos de ambas as partes.

O projeto será estruturado em fases, conforme definido na EAP, iniciando com o levantamento e análise de requisitos, onde serão identificadas as necessidades dos usuários e as funcionalidades essenciais do sistema. Em seguida, será realizada a modelagem de ferramentas de apoio à gestão, como a Matriz SWOT, para avaliar forças, fraquezas, oportunidades e ameaças, e a matriz 5W2H, para planejar as ações

do projeto com clareza e objetividade. Também será feito o mapeamento do ambiente de implantação, que envolverá tecnologias como JavaScript, React Native para o desenvolvimento da aplicação mobile, e MongoDB para armazenamento dos dados.

Entre as principais funcionalidades previstas estão: a publicação de demandas de frete por clientes, o acesso dos motoristas às cargas disponíveis, o contato rápido via integração com WhatsApp, e a localização de cargas e veículos por geolocalização. O sistema também contará com validação de identidade dos motoristas, avaliação de prestadores de serviço e registro histórico das transações, proporcionando mais segurança, confiabilidade e controle para todos os envolvidos.

Ao final das fases de desenvolvimento, testes e validação, o aplicativo será entregue em um ambiente operacional, com documentação técnica e suporte básico para implantação. O projeto busca não apenas entregar um produto funcional, mas também consolidar uma solução útil e escalável, que contribua de forma concreta para o cenário logístico da cidade e região.

1.2.5 Premissas

A construção de soluções tecnológicas eficazes demanda não apenas domínio técnico, mas também o uso de ferramentas e metodologias adequadas em cada fase do desenvolvimento. Dentre os princípios norteadores deste projeto, destaca-se o monitoramento constante e a validação contínua, com o intuito de criar um software eficiente, eficaz, intuitivo e com boa usabilidade. De acordo com Sommavilla e Peres (2019), a qualidade de um software está diretamente relacionada à sua capacidade de atender às reais necessidades dos usuários, sendo essencial a realização de testes iterativos, revisões frequentes e melhoria contínua ao longo do processo de desenvolvimento.

Desde o início, a aplicação foi concebida como uma solução mobile, considerando a crescente adesão dos usuários ao uso de smartphones para fins profissionais, especialmente no setor de transporte. A escolha por uma plataforma móvel visa proporcionar praticidade, mobilidade e acessibilidade, permitindo que caminhoneiros e contratantes realizem interações diretamente de seus dispositivos, em tempo real e com maior eficiência.

A gestão do tempo e do escopo foi fundamental para o projeto. Adotou-se um cronograma com entregas por etapas, decompondo o processo em atividades menores e exequíveis. Essa estratégia contribuiu para a produtividade da equipe e permitiu ajustes iterativos durante o desenvolvimento. Conforme destaca Anderson (2011, p. 127), a "decomposição de tarefas combinada com métodos visuais como o Kanban" – utilizado neste projeto por meio de quadros colaborativos – favorece "o controle do fluxo de trabalho, o acompanhamento transparente das etapas e a otimização simultânea do tempo e da qualidade das entregas".

Diversas ferramentas de apoio ao gerenciamento do projeto foram empregadas. O Miro foi utilizado como quadro colaborativo para brainstorming e estruturação de ideias. O Microsoft Word e Excel foram amplamente usados para documentação e planejamento. Já o Git e o GitHub desempenharam papel essencial no controle de versão e na colaboração entre os membros da equipe. O Bloco de Notas foi útil para anotações rápidas, enquanto o OneDrive garantiu o armazenamento seguro e compartilhado de arquivos.

Na fase de modelagem e diagramação, ferramentas distintas foram aplicadas conforme a complexidade dos artefatos gerados. O Camunda e o Visual Paradigm foram empregados na modelagem de processos com a notação BPMN (Business Process Model and Notation). Já o Miro, StarUML e Lucidchart foram utilizados para construção de diagramas baseados na UML, como casos de uso.

A prototipação da interface foi realizada com o uso do Figma, ferramenta reconhecida por permitir o design colaborativo de interfaces responsivas e de alta fidelidade. Nas etapas iniciais, o Canvas foi utilizado para estruturar a proposta de valor e definir os elementos funcionais essenciais da aplicação.

Do ponto de vista metodológico, o projeto foi fundamentado em princípios da Engenharia de Software, utilizando técnicas consolidadas como elicitação de requisitos (entrevistas, observações e análise de tarefas), matrizes de rastreabilidade, mapeamento da questão-problema e diagramas UML. Como observam Castro e Moreira (2022), o uso dessas práticas são essenciais para alinhar as expectativas dos usuários às funcionalidades do sistema, garantindo consistência e qualidade durante todo o ciclo de desenvolvimento.

1.2.6 Restrições

Durante o desenvolvimento do projeto, algumas restrições e dificuldades podem impactar diretamente o cronograma, os recursos e a execução das atividades previstas. A primeira limitação identificada está relacionada à disponibilidade de tempo da equipe, considerando que os integrantes conciliarão o projeto com outras demandas acadêmicas e pessoais. Isso pode afetar prazos de entrega, especialmente em fases que exigem maior dedicação, como testes e validações.

Outra restrição relevante é a limitação de recursos financeiros e técnicos, o que pode restringir o uso de serviços pagos, como servidores em nuvem, APIs comerciais ou ferramentas de desenvolvimento avançadas. A equipe buscará contornar esse desafio utilizando tecnologias de código aberto e infraestrutura gratuita ou de baixo custo, sempre que possível.

Também há a restrição quanto ao acesso ao público-alvo para testes mais amplos. Como o projeto é voltado a caminhoneiros da cidade, pode haver dificuldades logísticas para engajar motoristas durante o processo de levantamento de requisitos, testes de usabilidade e validações finais.

Por fim, por se tratar de uma solução que envolve validação de identidade e geolocalização, a equipe deverá considerar os cuidados legais e técnicos necessários para garantir privacidade, segurança e conformidade com normas de proteção de dados. Essas exigências podem limitar certas implementações ou demandar soluções adicionais que impactem o escopo inicial.

1.2.7 Stakeholders

- Motoristas e clientes voluntários;
- Equipe de desenvolvimento;

1.2.8 Riscos

Segurança:

O principal risco relacionado à segurança é o uso indevido da plataforma por pessoas não verificadas, o que pode gerar fraudes, golpes ou até riscos físicos, espe-

cialmente considerando que motoristas e clientes lidarão com cargas e endereços reais. Além disso, existe o risco de vazamento de dados pessoais, caso não sejam adotadas boas práticas de proteção de informações. Para mitigar esse risco, será necessário implementar autenticação segura, validação de identidade e armazenamento adequado dos dados.

Qualidade:

Há risco de entrega de funcionalidades com desempenho insatisfatório ou usabilidade ruim, devido à limitação de tempo e recursos. Interfaces pouco intuitivas, falhas na geolocalização ou instabilidade no envio de mensagens podem comprometer a experiência do usuário e, consequentemente, a adoção do aplicativo. Para reduzir esse risco, será essencial realizar testes com usuários reais durante as fases de prototipagem e validação.

Cronograma:

O cronograma pode ser comprometido por atrasos nas etapas de levantamento de requisitos, codificação ou testes, especialmente considerando a carga de trabalho acadêmica dos membros da equipe e possíveis dificuldades técnicas durante o desenvolvimento. Dependências externas, como a disponibilidade de usuários para teste, também podem impactar o andamento. A definição de marcos intermediários e uso de metodologias ágeis pode ajudar a manter o projeto sob controle.

2 Viabilidade do Projeto

A viabilidade do aplicativo IFrete foi avaliada por meio do Business Model Canvas (BMC), representando as bases estratégicas e operacionais do projeto. A proposta visa conectar clientes a motoristas de frete, otimizando rotas, reduzindo custos e evitando viagens vazias. O público-alvo inclui caminhoneiros autônomos, empresas com demanda logística e pessoas físicas que necessitam de fretes ou mudanças.

A solução será entregue por meio de um aplicativo mobile, com suporte via WhatsApp. O relacionamento com os usuários será sustentado por notificações, avaliações e suporte técnico personalizado. A monetização se dará por meio de comissão sobre fretes intermediados, planos premium para motoristas e anúncios logísticos.

Os recursos essenciais incluem equipe de desenvolvimento, banco de dados, servidores e sistema de geolocalização, utilizando ferramentas acessíveis como Node.js, MongoDB Community, React Native e Expo. As atividades principais envolvem manutenção da plataforma, atendimento ao cliente e estratégias de marketing digital. Parcerias com cooperativas, prefeituras e plataformas de pagamento podem ampliar o alcance e fortalecer a rede de operação.

O modelo demonstrou viabilidade tanto técnica quanto comercial, com foco na entrega de um MVP funcional. A estrutura proposta permite testes de mercado com baixo custo inicial e escalabilidade futura, validando o projeto como uma solução inovadora para o setor de transportes.

2.1 Canvas de Negócio (*Business Model Canvas* - BMC)

O Business Model Canvas (BMC) é uma ferramenta visual utilizada para descrever, analisar e inovar modelos de negócio de forma simples e estruturada. Ele é dividido em nove blocos, que representam as áreas essenciais de uma organização, permitindo uma visão sistêmica e estratégica do funcionamento do empreendimento.

Segundo Osterwalder (2011), o BMC é “uma linguagem comum para descrever, visualizar, avaliar e mudar modelos de negócio, permitindo decisões mais conscientes e ágeis por parte dos empreendedores”. Sua importância reside no fato de facilitar o planejamento, a validação e a comunicação do modelo de negócios, sendo amplamente utilizado por startups, incubadoras, investidores e empresas consolidadas.

Abaixo, está a figura1 com a distribuição do Canvas do nosso projeto:

Figura 1 - Canvas de Negócio do projeto

Business Model Canvas - IFrete



Fonte: Autores (2025)

Para o nosso projeto, o Business Model Canvas (BMC) permitiu estruturar e visualizar de forma clara e integrada todos os elementos-chave do modelo de negócio proposto para o aplicativo de conexão entre motoristas de frete e clientes. A utilização do BMC facilita a identificação de oportunidades de melhoria, o alinhamento entre a proposta de valor e as reais necessidades dos usuários — sejam eles motoristas, empresas ou pessoas físicas — e contribui para uma tomada de decisão mais estratégica e fundamentada. Ao apresentar uma visão sistêmica do projeto, o BMC orienta o planejamento, o desenvolvimento e a validação da solução, aumentando suas chances de sucesso no mercado. A seguir, cada bloco do BMC será explicado detalhadamente, com base na Figura 1 apresentada anteriormente.

Segmento de Clientes: O aplicativo tem como foco três perfis principais de usuários. O primeiro é composto por caminhoneiros autônomos e pequenos transportadores, que enfrentam dificuldades para encontrar cargas no retorno de suas viagens. Em segundo lugar, o sistema atende empresas de pequeno, médio e grande porte que possuem demanda recorrente por entregas, buscando otimizar seus custos logísticos. Por fim, o público também inclui pessoas físicas que precisam de serviços de mudanças residenciais ou pequenos transportes urbanos.

Proposta de Valor: A principal proposta do aplicativo é oferecer uma solução eficiente para os gargalos logísticos existentes, conectando motoristas e clientes de forma direta. Isso elimina a necessidade de intermediários, reduz os custos de transporte, evita viagens sem carga e melhora o aproveitamento das rotas, além de aumentar a visibilidade dos serviços disponíveis tanto para motoristas quanto para contratantes.

Canais: A interação entre o serviço e os usuários será feita, prioritariamente, por meio de um aplicativo mobile de fácil acesso. Além disso, haverá um site institucional com informações detalhadas e suporte ao usuário. A comunicação com o público será reforçada através de redes sociais e pelo WhatsApp, que funcionará como canal de atendimento direto.

Relacionamento com Clientes: A gestão do relacionamento será estruturada em diversas frentes, incluindo um sistema de atendimento automatizado e humano via chat, notificações push com atualizações de status dos fretes, mecanismos de avaliação mútua entre motoristas e clientes, e suporte técnico eficiente para garantir uma boa experiência de uso.

Fontes de Receita: O modelo de monetização do aplicativo será baseado principalmente em comissões cobradas sobre os fretes fechados na plataforma. Haverá também planos de assinatura premium voltados para motoristas, que poderão ter benefícios como destaque nas buscas. Além disso, o app poderá exibir anúncios de empresas parceiras, como oficinas mecânicas, seguradoras e postos de combustível.

Recursos-Chave: Para operar com eficiência, o projeto depende de recursos fundamentais como uma equipe de tecnologia da informação e desenvolvimento, infraestrutura em nuvem que proporcione escalabilidade e segurança, sistemas integrados de pagamento e tecnologias de geolocalização para rastreamento e otimização de rotas.

Atividades-Chave: Entre as atividades essenciais para o funcionamento do negócio estão o desenvolvimento e manutenção contínua do aplicativo, ações de marketing digital para captação de usuários, atendimento ao cliente para suporte e fidelização, e análise constante de dados para promover melhorias baseadas no comportamento e nas necessidades dos usuários.

Parcerias-Chave: As parcerias desempenham um papel estratégico no modelo de negócio. O projeto buscará colaborar com cooperativas de transporte e sindicatos

de caminhoneiros, empresas de logística e e-commerce, prefeituras e associações comerciais locais. Também serão importantes os acordos com plataformas de pagamento, como Mercado Pago, PagSeguro ou Pix, para facilitar as transações dentro do app.

Estrutura de Custos: Os custos principais do projeto incluem o desenvolvimento inicial da plataforma e suas atualizações, a hospedagem e segurança dos dados em servidores confiáveis, investimentos em marketing digital para divulgação e aquisição de usuários, bem como a manutenção de um serviço de atendimento e suporte técnico de qualidade.

2.2 Matriz SWOT

A Análise SWOT (Strengths, Weaknesses, Opportunities and Threats), também conhecida como matriz FOFA, é uma ferramenta de gestão estratégica utilizada para avaliar fatores internos e externos que afetam o desempenho de uma organização ou projeto. Sua principal função é oferecer uma visão estruturada das potencialidades e vulnerabilidades internas, bem como das oportunidades e ameaças do ambiente externo. Segundo Kotler e Keller (2012), a SWOT é essencial para alinhar as estratégias da organização aos seus recursos, capacidades e ao contexto de mercado, proporcionando uma base sólida para a tomada de decisões.

Os elementos internos da análise – Forças e Fraquezas – dizem respeito a aspectos sob controle direto da organização, como infraestrutura, tecnologia, equipe e processos. As Forças representam vantagens competitivas, enquanto as Fraquezas apontam limitações que podem comprometer o sucesso do projeto. Já os fatores externos – Oportunidades e Ameaças – estão ligados a mudanças no mercado, avanços tecnológicos, comportamentos de consumo, concorrência e políticas públicas. Hill e Jones (2012) ressaltam que reconhecer essas variáveis é fundamental para identificar riscos e antecipar tendências que podem influenciar os resultados da organização.

A simplicidade e aplicabilidade da matriz SWOT a tornam uma ferramenta estratégica valiosa, especialmente em projetos inovadores. No caso deste aplicativo de fretes, ela permite mapear os principais desafios enfrentados por motoristas e clientes, oferecendo subsídios para desenvolver soluções mais alinhadas à realidade de ambos. Ao analisar os dois perfis separadamente, torna-se possível elaborar

estratégias personalizadas, fortalecer os pontos positivos do sistema e minimizar riscos operacionais.

A seguir, será apresentado as matrizes SWOT 1 e 2 que foram elaboradas para a análise detalhada dos fatores que impactam o projeto em questão.

Quadro 1 – Análise SWOT dos motoristas

Forças	Oportunidades
• Possibilidade de obter mais fretes • Otimização de rotas com retorno remunerado • Maior autonomia e visibilidade no mercado	• Expansão do uso de tecnologia entre caminhoneiros • Incentivos governamentais para transporte eficiente • Acesso a novos mercados e clientes via aplicativo
Fraquezas	Ameaças
• Necessidade de adaptação tecnológica • Dificuldade inicial com o uso do app • Confiabilidade no sistema de pagamento	• Aumento da concorrência entre motoristas na plataforma • Instabilidade econômica impactando o setor • Mudanças na legislação de transporte

Fonte: Autores (2025)

Quadro 2 – Análise SWOT dos clientes

Forças	Oportunidades
• Facilidade para encontrar motoristas • Economia ao negociar diretamente • Praticidade e agilidade na contratação de fretes	• Expansão dos serviços de entrega e mudanças. • Crescimento do e-commerce demandando mais entregas. • Parcerias com empresas de transporte.
Fraquezas	Ameaças
• Falta de confiança inicial nos motoristas • Pouco conhecimento sobre fretes autônomos • Dificuldade de avaliar a qualidade do serviço	• Fraudes e problemas com entregas • Falta de regulamentação clara • Dependência de conexão com internet e dispositivos móveis

Fonte: Autores (2025)

A análise das Matrizes apresentadas destaca áreas-chave que precisam de intervenção para melhorar a fidelização de clientes. As principais fraquezas estão

relacionadas à confiança e ao desconhecimento sobre o funcionamento do serviço de fretes autônomos. Muitos usuários ainda preferem contratar motoristas tradicionais por medo de fraudes ou falta de garantia de qualidade, o que exige ações claras de comunicação e segurança por parte do aplicativo.

2.3 Plano de Ação 5W2H do Projeto

A matriz 5W2H é uma ferramenta gerencial amplamente utilizada para planejamento, execução e acompanhamento de ações em projetos e processos organizacionais. Conforme Oliveira (2024), esta técnica se destaca por sua simplicidade e objetividade, sendo ideal para transformar ideias em planos de ação claros, com responsabilidades bem definidas. O nome 5W2H deriva das iniciais, em inglês, de sete perguntas fundamentais: What (o que será feito?), Why (por que será feito?), Where (onde será feito?), When (quando será feito?), Who (quem fará?), How (como será feito?) e How much (quanto custará?). Essas questões orientam a execução de tarefas e garantem uma visão clara e detalhada de cada etapa do processo.

Segundo Oliveira (2024), a 5W2H facilita a organização das tarefas, o alinhamento entre os membros da equipe e o monitoramento dos resultados, sendo bastante útil tanto em ambiente empresariais quanto em projetos acadêmicos e startups. Sua importância também está ligada à clareza que proporciona no acompanhamento de planos e metas, permitindo identificar rapidamente falhas na execução e facilitar ajustes de rota quando necessário. Em projetos de desenvolvimento de software, como no caso do aplicativo de fretes, a matriz pode ser utilizada para organizar entregas, distribuir tarefas na equipe e alinhar os objetivos com os recursos disponíveis.

Figura 2 – Matriz 5W2H

5W2H	O que?	Por que?	Onde?	Quem?	Quando?	Como?	Quanto?
Solução 1	Implementar validação dos documentos do motorista	Garantir a autenticidade dos perfis cadastrados, promovendo segurança para os clientes e compliance com a legislação de transporte.	Backend do sistema	Equipe de desenvolvimento	Maio e Junho de 2025	Gerando integração com OCR gratuito e validações via API pública ou análise manual	Cálculo das métricas
Solução 2	Implementar avaliação dos motoristas	Permitir que clientes deem feedback, melhorando a qualidade do serviço, aumentando a confiança e auxiliando na gestão da reputação dos motoristas.	No sistema na camada Clientes	Equipe de desenvolvimento	Maio e Junho de 2025	Criando um sistema de avaliação por estrelas e comentários atrelados aos serviços finalizados	Cálculo das métricas
Solução 3	Implementar geolocalização em tempo real	Acompanhar o deslocamento do motorista em tempo real, agregando valor ao serviço e melhorando a experiência do cliente.	No sistema de fretes	Equipe de desenvolvimento	Maio e Junho de 2025	Utilizando a API do Expo Location e integração com mapa (Leaflet ou Mapbox)	Cálculo das métricas

Fonte: Autores (2025)

3 Levantamento de Requisitos

O levantamento de requisitos é uma etapa que permite identificar e compreender as necessidades e expectativas dos usuários e demais stakeholders. Esse processo serve de base para que a equipe de desenvolvimento possa projetar e implementar uma solução condizente com o problema identificado e com o contexto de uso real. Segundo Sommerville (2011), o levantamento de requisitos é uma das atividades mais críticas dentro da Engenharia de Software, pois erros cometidos nessa fase tendem a se propagar para as demais etapas do projeto, elevando os custos e comprometendo a qualidade do produto final.

De acordo com Castro e Moreira (2022), a elicitação bem-sucedida depende não apenas da escolha adequada das técnicas, mas também da capacidade da equipe em interpretar corretamente os dados coletados e transformá-los em requisitos funcionais (o que o sistema deve fazer) e não funcionais (como o sistema deve se comportar). No caso deste projeto, os requisitos funcionais envolvem, por exemplo, o cadastro de fretes, o filtro por tipo de carga ou veículo e a comunicação entre contratante e motorista. Já os requisitos não funcionais contemplam aspectos como segurança, desempenho e usabilidade.

3.1 Elicitação e especificação dos Requisitos

A importância da elicitação de requisitos está em garantir que o sistema a ser desenvolvido atenda, de forma precisa, às reais necessidades dos usuários. Segundo Sommerville (2011), “a elicitação é uma das etapas mais críticas da engenharia de requisitos, pois erros ou omissões nessa fase tendem a se propagar por todo o ciclo de vida do software.” Já Pressman (2016) destaca que “o sucesso de um projeto depende da compreensão clara do problema a ser resolvido”, reforçando a importância de métodos adequados de coleta de requisitos.

A origem da ideia do projeto surgiu a partir de uma dor real identificada na experiência pessoal de um dos integrantes da equipe, cujo pai é motorista de caminhão. O relato das dificuldades enfrentadas por ele e por colegas de profissão ao tentar encontrar fretes ou se tornar visível para possíveis contratantes motivou a equipe a buscar uma solução prática e eficiente. Com base nesse contexto, os primeiros requisitos foram formulados de forma exploratória, utilizando como base um modelo de usuário representativo e sua vivência no setor. A partir desse ponto inicial, os requisitos foram estruturados com o apoio de ferramentas de modelagem e validação progressiva, a fim de representar as necessidades reais do público-alvo.

Diante da limitação de acesso a um público mais amplo em um primeiro momento, optou-se por uma abordagem qualitativa, utilizando como ponto de partida o relato do usuário representativo. A equipe realizou conversas informais com outros motoristas da região, que compartilharam percepções similares em relação às dificuldades enfrentadas no cotidiano profissional, especialmente quanto à ineficiência na divulgação de disponibilidade, à insegurança nas negociações e à falta de um canal direto com os contratantes.

Essas interações, embora não formalizadas como entrevistas estruturadas, contribuíram significativamente para o refinamento dos requisitos iniciais, permitindo identificar padrões de comportamento e necessidades recorrentes entre os potenciais usuários. A partir dessas informações, foram descritas funcionalidades essenciais como: cadastro de demandas de frete, exibição de oportunidades para motoristas, contato direto por aplicativo de mensagens, geolocalização para facilitar o rastreamento, validação de identidade e um sistema de avaliação pós-serviço.

Os requisitos foram organizados com o auxílio de histórias de usuários (user stories), uma técnica amplamente adotada em metodologias ágeis por permitir

representar as funcionalidades do ponto de vista do usuário final. Essa abordagem favorece uma comunicação mais clara entre os membros da equipe de desenvolvimento e mantém o foco nas necessidades reais dos envolvidos.

As perguntas e respostas derivadas das conversas com os motoristas, bem como os primeiros rascunhos dos requisitos, estão reunidos no **Apêndice 1**, ao final deste documento, como forma de documentar e justificar as decisões de projeto tomadas com base na experiência prática do usuário-chave.

3.2 BPMN

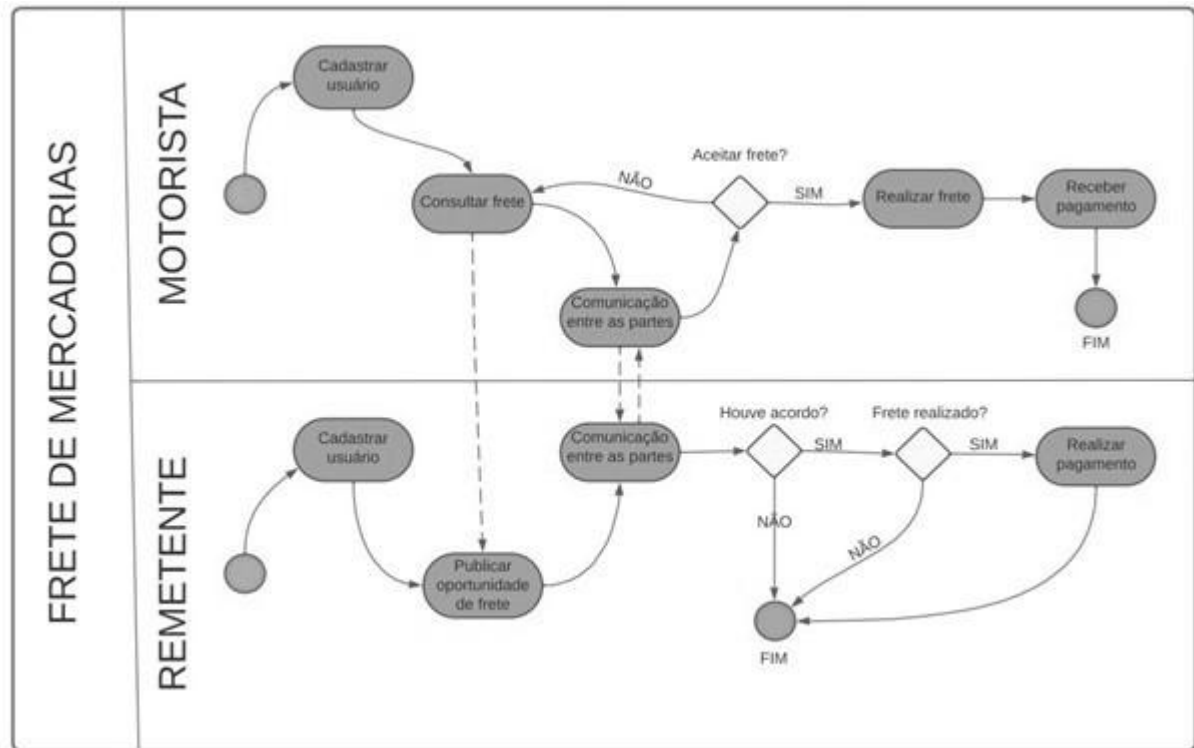
O BPMN (Business Process Model and Notation) é uma notação gráfica padronizada desenvolvida pelo Object Management Group (OMG) para descrever os processos de negócios de maneira clara, compreensível e estruturada. De acordo com o próprio OMG (2011), "o BPMN fornece uma notação compreensível por todos os participantes do processo de negócio, incluindo analistas de negócios, desenvolvedores técnicos e gestores." Ou seja, ele atua como uma ponte entre o desenho do processo e sua implementação prática em sistemas.

A principal função da BPMN é permitir o mapeamento dos fluxos de trabalho e operações dentro de uma organização, representando de forma visual as etapas, decisões e interações entre atores envolvidos em um processo. Segundo Dumas et al. (2018), "o BPMN ajuda na documentação, análise e melhoria dos processos de negócio, promovendo o alinhamento entre as necessidades do negócio e os sistemas de informação que os apoiam."

No contexto do projeto de um aplicativo de fretes, a modelagem BPMN é essencial para visualizar e entender o fluxo completo das interações entre os usuários (clientes e motoristas) e o sistema. Ela permite identificar os pontos de entrada de dados, os caminhos alternativos que um processo pode seguir e as possíveis melhorias operacionais. Ao representar graficamente o fluxo, a equipe consegue alinhar as expectativas, facilitar a comunicação entre os membros do time e identificar gargalos ou falhas antes da implementação.

Portanto, a utilização do BPMN neste projeto contribui não apenas para a clareza dos processos, mas também para a qualidade final do sistema, pois orienta o desenvolvimento com base em um modelo previamente validado com os requisitos e expectativas do usuário.

Figura 3 – BPMN do projeto



Fonte – Autores (2025)

3.3 Requisitos Funcionais

Os requisitos funcionais representam as funcionalidades específicas que um sistema deve possuir para atender às expectativas de seus usuários e partes interessadas. Eles definem o que o sistema deve fazer, descrevendo comportamentos, reações a entradas específicas e interações esperadas com o ambiente e com o usuário.

Segundo Sommerville (2011), "requisitos funcionais são declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em situações particulares". Ou seja, trata-se de funcionalidades observáveis e mensuráveis, que podem ser verificadas durante o desenvolvimento e nos testes do sistema.

Wieggers e Beatty (2013) complementam ao afirmar que requisitos funcionais são cruciais para garantir que o sistema atenda aos objetivos do negócio, pois descrevem o comportamento que o software deve apresentar para entregar valor. Eles constituem a base do contrato entre o cliente e a equipe de desenvolvimento.

Esses requisitos podem envolver ações explícitas (como registrar um frete ou enviar uma mensagem), além de comportamentos implícitos (como validações automáticas ou cálculos de distância). Incluem tanto os requisitos diretamente expressos pelos usuários quanto aqueles derivados da análise do domínio. A seguir, são apresentados o índice e a documentação detalhada dos requisitos funcionais identificados para o sistema.

Índice de requisitos funcionais:

- RF01 – Cadastro de motoristas
- RF02 – Cadastro de cliente
- RF03 – Publicação de frete
- RF04 – Consultar frete
- RF05 – Alterar dados cadastrados
- RF06 – Excluir conta
- RF07 – Canal de comunicação
- RF08 – Validação de credenciais
- RF09 – Geolocalização do motorista
- RF10 – Sistema de avaliação de motoristas
- RF11 – Consultar histórico de serviços
- RF12 – Filtro de busca
- RF13 – Notificações
- RF14 – Confirmação de aceite
- RF15 – Visualizar perfil do motorista
- RF16 – Upload de documentos

Quadro 3 – Requisitos Funcionais do sistema

RF01 -Cadastro de motoristas	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve permitir que motoristas criem perfis.		
RF02 -Cadastro de clientes	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve permitir que clientes criem perfis.		
RF003 -Publicação de frete	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta

		☐ Média ☐ Baixa
Descrição: O sistema deve permitir que clientes publiquem serviços de frete.		
RF04 -Consultar frete	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir que motoristas consultem serviços de frete.		
RF05 -Alterar dados cadastrados	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir que os usuários do sistema alterem seu cadastro.		
RF06 - Excluir conta	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir que os usuários do sistema excluam sua conta do app.		
RF07 -Canal de comunicação	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir que usuários entrem em contato com outros usuários.		
RF08 -Validação de credenciais	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve realizar a validação das credenciais dos motoristas.		
RF09 -Geolocalização do motorista	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir a geolocalização de motoristas.		
RF10 -Avaliação de motoristas	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir a avaliação de motoristas após a conclusão de um serviço.		
RF11 -Histórico de serviços	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir consultar o histórico de serviços efetuados.		
RF12 -Filtro de busca	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir realizar busca parametrizada.		

RF13-Notificações	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir o envio de notificações para clientes e motoristas sobre serviços.		
RF14-Confirmação de aceite.	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir que motoristas aceitem um serviço de frete.		
RF15-Consulta de perfil motorista.	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir clientes consultar o perfil do motorista.		
RF16-Upload de documentos	Categoria: ☐ Oculto ☒ Evidente	Prioridade: ☒ Altíssima ☐ Alta ☐ Média ☐ Baixa
Descrição: O sistema deve permitir o upload de documentos para motoristas.		

Fonte: Autores (2025)

3.4 Requisitos Não Funcionais

Os Requisitos Não Funcionais (RNF) descrevem as características de qualidade que o sistema deve possuir, como desempenho, segurança, usabilidade e compatibilidade. Eles não dizem o que o sistema faz, mas como ele deve funcionar para garantir uma boa experiência ao usuário e atender às restrições técnicas e organizacionais. Segundo Sommerville (2011), “requisitos não funcionais são restrições sobre os serviços ou funções oferecidas pelo sistema, como restrições de tempo, requisitos de desenvolvimento, padrões de qualidade ou limitações do sistema operacional.” Já Pressman (2016) destaca que “esses requisitos afetam diretamente a experiência do usuário, a estabilidade e a escalabilidade da aplicação, sendo tão críticos quanto os funcionais.” Portanto, os RNF são fundamentais para assegurar que o sistema seja eficiente, seguro, confiável e acessível, complementando os requisitos funcionais no sucesso do produto. A seguir, são apresentados o índice e a documentação detalhada dos requisitos não funcionais identificados para o sistema.

Quadro 4 – Requisitos Não Funcionais do sistema

RNF01 - Usabilidade.	Tipo: Usabilidade	Obrigatoriedade: () Desejável (X) Obrigatório	Permanência: (X) Permanente () Transitório
Descrição: A interface do sistema deve ser intuitiva, permitindo uso fluido mesmo por usuários com baixa familiaridade tecnológica.			
RNF02 - Portabilidade.	Tipo: Portabilidade	Obrigatoriedade: () Desejável (X) Obrigatório	Permanência: (X) Permanente () Transitório
Descrição: O sistema deve estar disponível para interfaces Android e iOS.			
RNF03 – Segurança dos dados	Tipo: Segurança	Obrigatoriedade: () Desejável (X) Obrigatório	Permanência: (X) Permanente () Transitório
Descrição: As senhas dos usuários devem ser armazenadas utilizando hash seguro, como bcrypt..			
RNF04 – Integração	Tipo: Confiabilidade	Obrigatoriedade: (X) Desejável () Obrigatório	Permanência: (X) Permanente () Transitório
Descrição: A comunicação entre cliente e motorista deve ocorrer por meio da API oficial do WhatsApp.			
RNF05 – Autorização para acesso a recursos.	Tipo: Confiabilidade e Usabilidade	Obrigatoriedade: () Desejável (X) Obrigatório	Permanência: (X) Permanente () Transitório
Descrição: O sistema deve solicitar permissão explícita para acesso e uso de funcionalidades do aparelho, como câmera, localização e notificações, conforme boas práticas da LGPD.			

Fonte: Autores (2025)

3.5 Regras de Negócio

As regras de negócio são diretrizes que definem, restringem ou orientam os processos e operações realizados por uma organização. Elas estabelecem como a empresa deve operar em diferentes cenários, descrevendo ações, condições e restrições que refletem as políticas internas, estratégias e regulamentações externas. Quando aplicadas em sistemas de informação, essas regras funcionam como uma camada lógica independente que pode ser ajustada sem a necessidade de alterar o código da aplicação, conferindo flexibilidade e agilidade à manutenção e evolução do software. (IBM, acesso em 19 jun. 2025).

Segundo García e Kaczmarek (2020, p. 77-89), as regras de negócio "servem para traduzir as políticas da empresa em orientações claras e objetivas para os

sistemas de software", assegurando que a solução tecnológica reflita fielmente as práticas organizacionais. Isso permite alinhar as funcionalidades do sistema às necessidades do negócio, promovendo consistência, eficiência e coerência operacional.

Sua importância também se manifesta na capacidade de reduzir erros humanos, ao estabelecer critérios objetivos para a execução de tarefas e ao substituir verificações manuais por decisões automatizadas. De acordo com Rubinoff e Sharma (2021), as regras de negócio oferecem uma base clara para a automação de processos, ajudando a manter a conformidade regulatória e a eficiência operacional. Além disso, tornam a lógica de negócio visível e compreensível até mesmo para profissionais sem conhecimento técnico, favorecendo auditorias, transparência e a evolução contínua do sistema.

Dessa forma, as regras de negócio não apenas sustentam a integridade funcional dos sistemas de software, mas também garantem que estes acompanhem a dinâmica organizacional, promovendo robustez, flexibilidade e facilidade na implementação de novas funcionalidades. Neste documento, as regras de negócio deste projeto são apresentadas no Quadro 3, listadas com seus respectivos identificadores, nomes e descrições.

Quadro 5 – Regras de Negócio do sistema.

RN01 – CPF do cliente ou motorista
Descrição: Só serão permitidos cadastros com CPF válidos quanto a existência e formato. O cadastro de uma pessoa física só será concluído após validação do CPF e caso não exista duplicação na base de dados.
RN02 – Validação de CNH
Descrição: Apenas serão aceitos cadastros de motoristas com CNH validada pelo sistema.
RN03 – Política de rastreo
Descrição: A opção de geolocalização só deve ser habilitada com o consentimento prévio do usuário, através de notificação enviada pelo sistema.
RN04 – Sistema de avaliação
Descrição: Só será permitido editar ou excluir comentários sobre avaliação de motoristas, pelo usuário que gerou a avaliação.
RN05 – Publicação de serviço
Descrição: Só será permitido gerar uma publicação após preenchimento dos campos obrigatórios: endereços de origem e destino e tipo de carga.
RN06 – Consulta de serviço

Descrição: O sistema deve buscar apenas serviços com status em aberto, serviços rejeitados pelo usuário autenticado não devem ser retornados.
RN07 – Comunicação
Descrição: O sistema deve permitir que usuários entrem em contato com a parte interessada, apenas quando um motorista alterar o status de um serviço para aceito.
RN08 – Busca personalizada
Descrição: Os usuários podem aplicar filtros de busca por status do serviço, origem e destino.
RN09 – Política para notificar
Descrição: A opção de notificação local só deve ser habilitada com o consentimento prévio do usuário, através de alerta enviado pelo sistema para coleta da autorização.
RN10 – Confirmação de aceite
Descrição: Após o aceite, o frete muda de status para “aceito”, sendo automaticamente removido da listagem de outros motoristas.

Fonte: Autores (2025)

3.6 Casos de Uso

Os casos de uso são diagramas com representações sobre como as tarefas do sistema serão executadas por funcionalidades, passo a passo pelos usuários, sem a necessidade de especificação de como tais funcionalidades serão implementadas. Utiliza de uma linguagem simples e de fácil compreensão para que se possa ter uma visão geral de como o sistema irá se comportar, identificando os atores e funcionalidades do sistema. (Guedes, 2018).

Os casos de uso funcionam como histórias que ilustram como o sistema é utilizado na prática, promovendo uma comunicação eficaz entre desenvolvedores, analistas, clientes e demais stakeholders. No contexto da UML, esses casos são elementos fundamentais para a modelagem da funcionalidade do sistema, permitindo compreender claramente as interações entre usuários e sistema.

Além disso, os casos de uso são importantes ferramentas para a identificação de requisitos funcionais e não funcionais, como desempenho e segurança, e auxiliam na elaboração de testes mais eficazes. Para Jacobson et al. (1992), essa técnica é essencial para garantir a qualidade e o sucesso de um projeto de software. Sua aplicação contínua durante o desenvolvimento facilita a validação do sistema,

contribui para uma visão compartilhada entre os envolvidos no projeto e apoia futuras atualizações e manutenções com maior clareza e controle.

Índice de casos de uso:

UC01 – Cadastrar usuário

UC02 – Publicar frete

UC03 – Consultar frete

UC04 – Alterar dados cadastrados

UC05 – Excluir conta

UC06 – Enviar mensagem

UC07 – Validar credenciais

UC08 – Habilitar geolocalização

UC09 – Avaliar motoristas

UC10 – Consultar histórico de serviços

UC11 – Filtrar busca

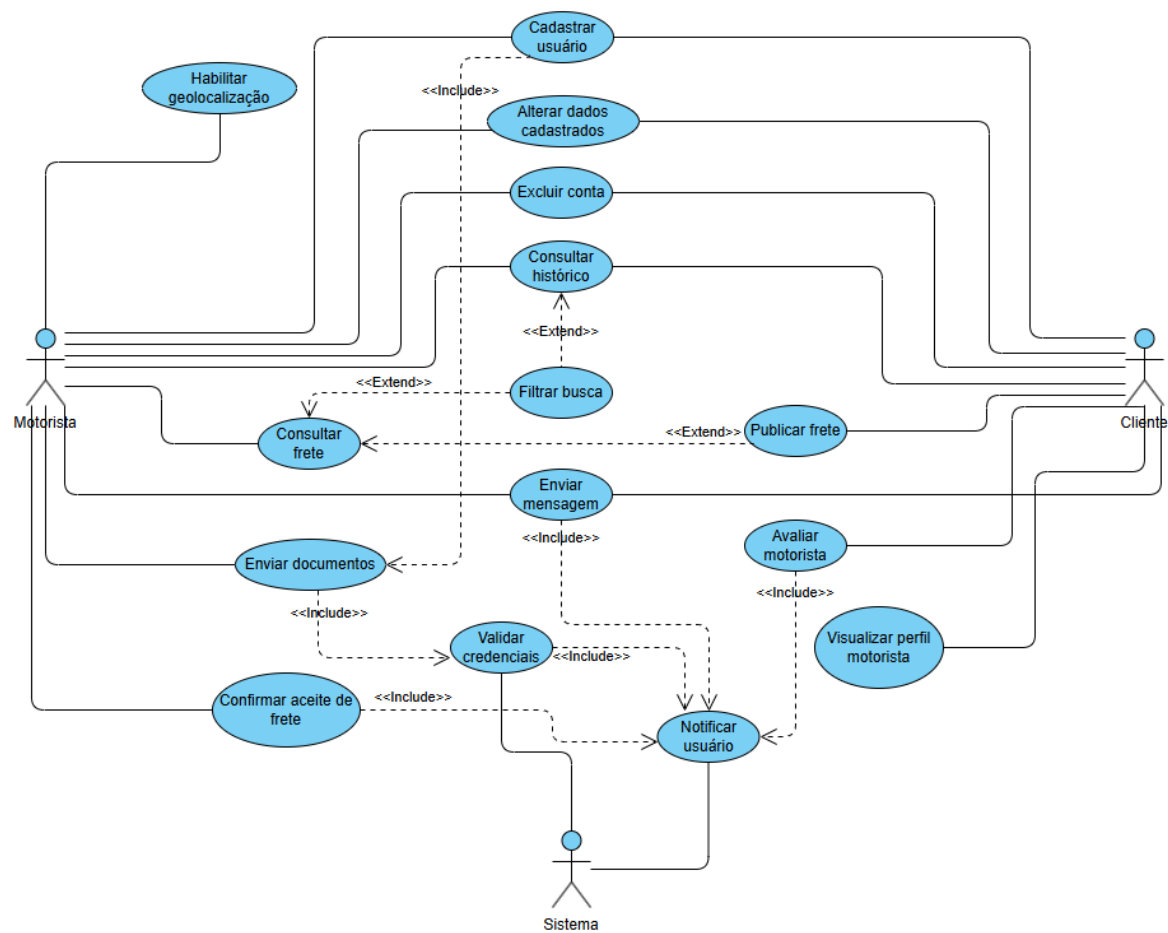
UC12 – Notificar usuário

UC13 – Confirmar aceite de frete

UC14 – Visualizar perfil do motorista

UC15 – Enviar documentos

Figura 4 – Use Case Cadastrar Usuários



Fonte: Autores (2025)

Quadro 6 – Especificação dos casos de uso

Caso de Uso – Cadastrar usuários	
ID	UC 01
Descrição	Este caso de uso tem por objetivo cadastrar novos usuários no sistema.
Ator Primário	Usuário do sistema (motorista ou cliente)
Ator Secundário	Sistema
Pré-condição	Nenhuma
Cenário Principal	1. O use case inicia com o usuário selecionando a opção “Cadastrar” na tela de login. 2. O sistema carrega o formulário de cadastro. 3. O usuário preenche o nome, e-mail e senha. 4. O usuário seleciona o tipo motorista. 5. O usuário preenche os campos obrigatório CNH e Tipo de veículo. 6. O usuário seleciona a opção cadastrar. 7. O sistema valida os dados inseridos pelo usuário. 8. O sistema realiza o registro do novo usuário na base de dados. 9. O sistema apresenta mensagem de sucesso. 10. O sistema redireciona o usuário para tela de login.
Pós-condição	Para ativar o cadastro, no primeiro login o motorista é solicitado a anexar a CNH e foto de perfil (UC15) para validação do sistema (UC07).

Cenário Alternativo	4a – O usuário seleciona o tipo cliente. 5ª – O usuário preenche o telefone e campos opcionais.
Caso de Uso – Publicar frete	
ID	UC 02
Descrição	Este caso de uso tem por objetivo a publicação de novo frete
Ator Primário	Usuário do sistema (cliente)
Pré-condição	O usuário precisa estar autenticado no sistema
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção “Publicar Serviço”. 2. O sistema carrega um formulário de publicação, contendo campos para descrever as especificações do frete. 3. O usuário preenche os campos obrigatórios e opcionais. 4. O usuário seleciona a opção publicar. 5. O sistema valida os dados informados. 6. O sistema apresenta uma tela de sucesso. 7. O sistema redireciona o usuário para página de histórico.
Pós-condição	Nenhuma
Cenário Alternativo	5a – O sistema apresenta mensagem de erro, caso haja inconsistência nos dados informados solicitando a correção.
Caso de Uso – Consultar frete	
ID	UC 03
Descrição	Este caso de uso tem por objetivo a consulta de fretes disponíveis
Ator Primário	Usuário do sistema (motorista)
Pré-condição	O usuário precisa estar autenticado no sistema e com CNH validada.
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção “Consultar Serviços”. 2. O sistema carrega uma página com uma lista de propostas disponíveis. 3. O usuário seleciona uma proposta. 4. O sistema carrega os dados do serviço.
Pós-condição	O usuário aceita ou rejeita o serviço.
Cenário Alternativo	Nenhum
Caso de Uso – Alterar dados cadastrados	
ID	UC 04
Descrição	Este caso de uso tem por objetivo a alteração de dados cadastrados
Ator Primário	Usuários do sistema (motorista ou cliente)
Pré-condição	O usuário precisa estar autenticado no sistema
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção “Cadastro”. 2. O sistema carrega o formulário com os dados do usuário. 3. O usuário altera os dados. 4. O usuário seleciona a opção salvar alterações. 5. O sistema valida os dados inseridos. 6. O sistema carrega mensagem de sucesso e redireciona para home.
Pós-condição	Nenhuma
Cenário Alternativo	Nenhum
Caso de Uso – Excluir conta	
ID	UC 05
Descrição	Este caso de uso tem por objetivo excluir conta de usuário
Ator Primário	Usuários do sistema (motorista ou cliente)

Pré-condição	O usuário precisa estar autenticado no sistema
Cenário Principal	1. O use case inicia quando o usuário seleciona a opção “Cadastro”. 2. O sistema carrega a tela de cadastro. 3. O usuário seleciona a opção excluir conta. 4. O sistema apresenta mensagem pedindo a confirmação. 5. O usuário confirma a exclusão do cadastro. 6. O sistema exclui o cadastro do usuário do sistema removendo todos os dados do usuário da base de dados do sistema. 7. O sistema redireciona para página de login.
Pós-condição	Nenhuma
Cenário Alternativo	Nenhum
Caso de Uso – Enviar mensagem	
ID	UC 06
Descrição	Este caso de uso tem por objetivo o envio de mensagens entre usuários
Ator Primário	Usuários do sistema (motorista ou cliente)
Pré-condição	O usuário precisa estar autenticado e na tela de histórico de serviços.
Cenário Principal	1. O use case inicia quando o motorista seleciona o serviço. 2. O usuário seleciona a opção “Entrar em Contato”. 3. O sistema abre uma janela com as informações do cliente e um botão com ícone do whatsapp. 4. O usuário interage com o botão do whatsapp. 5. O sistema redireciona o usuário para uma conversa no whatsapp. 6. O sistema gera um histórico de conversa no aplicativo.
Pós-condição	As partes interessadas combinam o serviço, o motorista realiza o frete e por fim altera o status do serviço no aplicativo para concluído.
Cenário Alternativo	1a – O use case inicia quando o cliente seleciona o serviço.
Caso de Uso – Validar credenciais	
ID	UC 07
Descrição	Este caso de uso tem por objetivo validar a CNH e foto do motorista
Ator Primário	Sistema
Pré-condição	O motorista fez o upload da CNH e foto de perfil para o aplicativo (UC15).
Cenário Principal	1. O use case inicia quando o sistema recebe um novo registro de documento e foto de motorista. 2. Escaneamento da CNH via OCR. 3. Validação de dados com órgão governamental. 4. Validação para comparar selfie com foto da CNH. 5. Validação aprovada e registrada. 6. O sistema envia notificação ao motorista.
Pós-condição	Nenhuma
Cenário Alternativo	5a – Validação reprovada 6a – O sistema envia notificação ao motorista informando as divergências.
Caso de Uso – Habilitar geolocalização	
ID	UC 08
Descrição	Este caso de uso tem por objetivo habilitação de geolocalização para rastreamento de serviço
Ator Primário	Usuários do sistema (cliente ou motorista)
Pré-condição	O usuário precisa estar autenticado no sistema
Cenário Principal	1. O use case inicia quando um serviço recebe o status iniciado. 2. O sistema solicita a permissão de acesso à geolocalização do dispositivo. 3. O usuário autoriza o acesso à localização ao confirmar a solicitação. 4. O sistema habilita a geolocalização em segundo plano.

	5. O sistema passa a registrar e disponibilizar os dados de localização para acompanhamento do serviço..
Pós-condição	O sistema mantém a geolocalização ativa durante a execução do serviço. Quando o motorista altera o status do serviço para finalizado, a geolocalização é desabilitada automaticamente.
Cenário Alternativo	3a – O usuário nega a solicitação de uso da geolocalização. 4a – O sistema exibe uma mensagem informando que o rastreamento não será possível e o serviço poderá ser cancelado. 5a – O fluxo termina ou segue com limitações.
Caso de Uso – Avaliar motoristas	
ID	UC 09
Descrição	Este caso de uso tem por objetivo a avaliação de motoristas após término de serviço
Ator Primário	Usuário do sistema (cliente)
Pré-condição	O serviço precisa estar com status finalizado.
Cenário Principal	1. O caso de uso inicia quando o usuário acessa a página de "Histórico de Serviços" e seleciona um serviço finalizado. 2. O sistema carrega a tela de detalhes do serviço, exibindo a opção "Avaliar Motorista". 3. O sistema gera um código de avaliação único vinculado à corrida (ex: AVA123456789). 4. O sistema exibe os dados do motorista (nome, foto, placa do veículo) e um campo de avaliação com estrelas (1 a 5) e comentário opcional. 5. O usuário seleciona a quantidade de estrelas que deseja atribuir ao motorista. 6. O sistema registra a nota e verifica se o campo de comentário foi preenchido. 7. O usuário confirma a avaliação clicando em "Enviar Avaliação". 8. O sistema armazena os dados da avaliação e associa ao perfil do motorista. 9. O sistema exibe uma mensagem de confirmação de que a avaliação foi enviada com sucesso.
Pós-condição	A avaliação é registrada e passa a compor a média pública de avaliações do motorista. O cliente não poderá avaliar o mesmo serviço novamente.
Cenário Alternativo	5a – O usuário não deseja avaliar o motorista: 5a.1 – O usuário clica em “Pular avaliação”. 5a.2 – O sistema registra que o serviço foi encerrado sem avaliação.
Caso de Uso – Consultar histórico de serviços	
ID	UC 10
Descrição	Este caso de uso tem por objetivo permitir que o cliente ou motorista consulte a lista de serviços realizados ou solicitados, com detalhes de cada um.
Ator Primário	Usuário do sistema (cliente ou motorista)
Pré-condição	O usuário precisa estar autenticado no sistema.
Cenário Principal	1. O caso de uso inicia quando o usuário seleciona a opção “Histórico de Serviços” no menu principal. 2. O sistema carrega uma página com os serviços relacionados ao perfil do usuário autenticado (cliente ou motorista). 3. O usuário visualiza uma lista contendo data, horário, status (finalizado, cancelado, etc.), nome do outro envolvido (motorista ou cliente), e valores. 4. O usuário seleciona um item da lista para visualizar os detalhes do serviço. 5. O sistema exibe informações completas do serviço: data, trajeto, valor, tempo estimado e real, status, avaliações (se houver).

	6. O usuário pode filtrar os resultados por data, status ou tipo de serviço.
Pós-condição	Nenhuma
Cenário Alternativo	Nenhum
Caso de Uso – Filtrar busca	
ID	UC 11
Descrição	Este caso de uso tem por objetivo permitir que o usuário aplique filtros em buscas de serviços para refinar os resultados com base em parâmetros específicos, como data, status, tipo de serviço ou localização.
Ator Primário	Usuário do sistema (cliente ou motorista)
Pré-condição	O usuário deve estar em uma interface com uma lista consultável (ex: histórico de serviços, motoristas disponíveis etc.)
Cenário Principal	1. O caso de uso inicia quando o usuário acessa uma lista de serviços e seleciona a opção do ícone de lupa “Filtrar”. 2. O sistema exibe os campos de filtro disponíveis, como data inicial/final, status do serviço (pendente, finalizado, cancelado), tipo de carga, entre outros. 3. O usuário preenche os critérios desejados nos campos fornecidos. 4. O sistema valida os parâmetros preenchidos (ex: intervalo de datas coerente). 5. O usuário confirma a filtragem clicando em “Aplicar filtros”. 6. O sistema atualiza a lista de serviços com base nos critérios informados. 7. O sistema exibe a quantidade de resultados encontrados ou informa que nenhum item corresponde aos filtros. 8. O usuário pode limpar os filtros ou modificar os critérios para uma nova busca.
Pós-condição	A lista de serviços é atualizada de acordo com os filtros aplicados, permitindo ao usuário localizar rapidamente informações específicas.
Cenário Alternativo	4a – O usuário preenche critérios inválidos (ex: data final anterior à inicial): 4a.1 O sistema exibe uma mensagem de erro indicando o problema e impede a aplicação do filtro. 7a – Nenhum serviço corresponde aos filtros aplicados: 7a.1 O sistema informa: “Nenhum resultado encontrado para os critérios informados.” 7a.2 O sistema oferece a opção de redefinir filtros ou retornar à lista completa.
Caso de Uso – Notificar usuário	
ID	UC 12
Descrição	Este caso de uso tem por objetivo notificar motoristas e clientes, sobre atualizações no status de seus serviços.
Ator Primário	Usuário do sistema (motorista ou cliente)
Pré-condição	O usuário precisa estar autenticado no sistema
Cenário Principal	1. O caso de uso inicia quando o status de um serviço é alterado (ex.: “Aceito”, “Em andamento”, “Finalizado” ou “Cancelado”). 2. O sistema identifica o usuário relacionado ao serviço (cliente e/ou motorista). 3. O sistema gera uma notificação com informações resumidas: tipo de alteração, data/hora, ID do serviço e mensagem informativa. 4. A notificação é registrada na central interna de notificações do aplicativo. 5. O usuário acessa a área de notificações no menu do aplicativo. 6. O sistema exibe uma lista com todas as notificações relacionadas à conta do usuário.

	7. O usuário seleciona uma notificação para visualizar detalhes do serviço correspondente. 8. O sistema direciona o usuário para a tela de detalhes do serviço mencionado. 9. O sistema marca a notificação como "lida" após sua visualização.
Pós-condição	A notificação é armazenada no histórico do aplicativo e pode ser consultada posteriormente pelo usuário.
Cenário Alternativo	Nenhum
Caso de Uso – Confirmar aceite de frete	
ID	UC 13
Descrição	Este caso de uso tem por objetivo permitir que o motorista confirme o aceite de um frete previamente visualizado no aplicativo, sinalizando sua disponibilidade para realizar o serviço.
Ator Primário	Usuário do sistema (motorista)
Pré-condição	O usuário precisa estar autenticado no sistema e deve ter um serviço disponível selecionado na tela.
Cenário Principal	1. O caso de uso inicia quando o motorista visualiza os detalhes de um frete disponível e seleciona a opção “Aceitar Frete”. 2. O sistema redireciona o usuário para página de “Histórico de Serviços”. 3. O usuário seleciona o serviço aceito. 4. O sistema carrega uma tela com as informações do serviço e as opções para “Entrar em Contato”, “Iniciar Serviço” e “Cancelar”. 5. O motorista confirma sua intenção clicando no botão. 6. O sistema registra o aceite do frete e altera o status do serviço para “Aceito” removendo da área de serviços disponíveis.
Pós-condição	O motorista entra em contato com cliente, ou inicia o serviço ou cancela o serviço.
Cenário Alternativo	Nenhum
Caso de Uso – Visualizar perfil do motorista	
ID	UC 14
Descrição	Este caso de uso tem por objetivo que o cliente consiga visualizar o perfil do motorista que aceitou o frete
Ator Primário	Usuário do sistema (cliente)
Pré-condição	O usuário precisa estar autenticado no sistema e o serviço precisa estar com status aceito.
Cenário Principal	1. O use case inicia quando o usuário abre a tela de “Histórico de Serviços” e seleciona um serviço aceito. 2. O sistema abre o serviço com os detalhes e as opções “Entrar em Contato” e “Ver perfil motorista”. 3. O usuário seleciona a opção “Ver perfil motorista”. 4. O sistema carrega uma tela com as informações sobre o motorista.
Pós-condição	Nenhuma
Cenário Alternativo	Nenhum
Caso de Uso – Enviar documentos	
ID	UC 15
Descrição	Este caso de uso tem por objetivo permitir o envio da CNH e foto de perfil.
Ator Primário	Usuário do sistema (motorista)
Pré-condição	O usuário precisa estar autenticado no sistema e ter conta com tipo “Motorista”.
Cenário Principal	1. O use case inicia quando o motorista realiza o primeiro login no sistema.

	2. O sistema identifica que o envio de documentos obrigatórios ainda não foi realizado. 3. O sistema exibe uma tela com informações solicitando o upload da CNH e da foto de perfil. 4. O sistema solicita permissão para acessar os arquivos do dispositivo. 5. O usuário concede permissão de acesso. 6. O sistema exibe a galeria de arquivos do dispositivo. 7. O usuário seleciona os arquivos correspondentes à CNH e à foto de perfil. 8. O usuário confirma a seleção clicando na opção “Enviar documentos”. 9. O sistema realiza o upload dos arquivos e valida o formato (ex: JPEG, PNG, PDF). 10. O sistema exibe uma mensagem de confirmação de envio bem-sucedido. 11. O sistema redireciona o usuário para a página principal do aplicativo.
Pós-condição	Os documentos são armazenados com o perfil do motorista e no próximo acesso não será solicitado o reenvio.
Cenário Alternativo	4a – O usuário nega a permissão de acesso aos arquivos 4a.1 O sistema exibe uma mensagem explicando que o envio dos documentos é obrigatório. 4a.2 O sistema oferece a opção de reabrir a solicitação de permissão ou encerrar a sessão. 9a – O usuário envia arquivos inválidos (formato não aceito ou imagem corrompida) 9a.1 O sistema exibe uma mensagem de erro informando o problema. 9a.2 O sistema solicita que o usuário envie novamente os arquivos corretos.

Fonte: Autores (2025)

3.7 Matriz de Rastreabilidade

Segundo Sommerville (2011), a matriz de rastreabilidade é uma ferramenta para o processo de engenharia de requisitos, utilizada para estabelecer relações entre os diversos elementos de um sistema, como requisitos, testes, artefatos de design e casos de uso. A proposta dessa matriz é fornecer uma visão estruturada que permita o acompanhamento da evolução e do atendimento dos requisitos ao longo de todo o ciclo de vida do projeto.

A matriz de rastreabilidade facilita o gerenciamento de requisitos, garantindo que todas as necessidades especificadas inicialmente sejam devidamente implementadas e testadas. Essa ferramenta torna-se estratégica para assegurar que alterações no sistema possam ser controladas e avaliadas com clareza quanto aos impactos que geram.

A importância da matriz reside na sua capacidade de revelar lacunas, conflitos ou requisitos não implementados, de acordo com Pressman (2016). Ela auxilia no

controle de mudanças, permite auditorias mais eficientes e garante a cobertura completa dos testes, promovendo maior segurança e confiabilidade no produto desenvolvido. Dessa forma, a matriz de rastreabilidade não apenas serve como uma ferramenta de documentação e controle, mas também como um mecanismo preventivo, evitando retrabalho, facilitando a manutenção e assegurando o alinhamento contínuo entre os objetivos do sistema e a solução envolvida.

A seguir, são apresentadas duas matrizes de rastreabilidade elaboradas a partir dos documentos de requisitos funcionais e dos casos de uso deste projeto.

Quadro 7 – Requisitos x Regras do negócio

	RN01	RN02	RN03	RN04	RN05	RN06	RN07	RN08	RN09	RN10
RF01	X	X								
RF02	X									
RF03					X					
RF04						X				
RF05	X	X		X						
RF06	X	X		X						
RF07							X			
RF08	X	X								
RF09			X							
RF10				X						
RF11				X						
RF12								X		
RF13									X	
RF14										X
RF15				X						
RF16		X								

Fonte: Autores (2025)

Quadro 8 – Requisitos x Regras Sistêmicas

	UC 01	UC 02	UC 03	UC 04	UC 05	UC 06	UC 07	UC 08	UC 09	UC 10	UC 11	UC 12	UC 13	UC 14	UC 15
RF01	X														
RF02	X														
RF03		X													
RF04			X												
RF05				X											
RF06					X										
RF07						X									
RF08							X								
RF09								X							
RF10									X						
RF11										X					
RF12											X				
RF13												X			
RF14													X		
RF15														X	
RF16															X

Fonte: Autores (2025)

4 Ferramentas e Métodos

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade. Além disso, essas ferramentas têm documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e

implementarem as soluções. A escolha dessas ferramentas também foi influenciada pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas. As licenças das ferramentas são de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais.

Para o banco de dados, optou-se pelo MongoDB, um sistema NoSQL amplamente reconhecido por sua flexibilidade na modelagem de dados em documentos no formato BSON (uma extensão do JSON). Segundo Chodorow (2019), o MongoDB é particularmente adequado para aplicações que lidam com dados semi-estruturados e em constante evolução, uma vez que dispensa esquemas fixos e possibilita alterações no formato dos dados sem impacto estrutural significativo. Isso o torna ideal para o contexto do sistema de fretes, onde informações sobre cargas, motoristas e rotas podem variar em estrutura e volume.

A decisão por um banco de dados não relacional também se alinha às demandas modernas de aplicações com alto volume de transações e dados heterogêneos. Conforme apontado por Kristina Chodorow (2013), bancos NoSQL, como o MongoDB, oferecem alta disponibilidade, escalabilidade horizontal e desempenho otimizado, especialmente em sistemas distribuídos. O mecanismo de sharding e os índices eficientes do MongoDB permitem distribuir dados entre diferentes servidores e realizar consultas de forma rápida e consistente, reduzindo a latência e garantindo o desempenho do sistema.

No caso específico deste projeto, a estrutura orientada a documentos do MongoDB traz vantagens práticas, como a fácil integração com APIs REST e a agilidade no processamento de dados relacionados à geolocalização, status de cargas e interações entre motoristas e clientes. Isso assegura não apenas a confiabilidade do armazenamento, mas também a escalabilidade necessária para atender ao crescimento futuro da aplicação.

Como framework de desenvolvimento, optou-se pelo React Native, uma tecnologia desenvolvida pelo Facebook, voltada para a criação de aplicativos móveis multiplataforma com uso de JavaScript e React. (Facebook, 2025). Segundo Akhter e Roy (2020), o React Native possibilita a geração de aplicativos tanto para Android quanto para iOS a partir de uma única base de código, o que reduz significativamente o esforço de desenvolvimento e os custos de manutenção, eliminando a duplicação de lógica entre plataformas.

No contexto do sistema de fretes proposto, o React Native se mostrou vantajoso por oferecer recursos como hot-reloading, permitindo que alterações no código sejam visualizadas em tempo real, o que acelera o ciclo de testes e ajustes da interface. A necessidade de uma aplicação responsiva e intuitiva é plenamente atendida com esse framework, cuja ampla comunidade ativa fornece uma grande variedade de bibliotecas modernas, documentação e exemplos, facilitando a implementação de funcionalidades complexas com menor esforço técnico.

Como ferramenta complementar ao desenvolvimento em React Native, o projeto adota o Expo, um framework e plataforma de build que facilita significativamente a criação, teste e deploy de aplicativos móveis. O Expo elimina a complexidade tradicional do setup de ambientes nativos, oferecendo um ecossistema completo que permite aos desenvolvedores criar aplicações móveis utilizando apenas JavaScript, sem a necessidade de configurar Xcode, Android Studio ou outros ambientes nativos.

A eficiência do Expo reside no seu ambiente de desenvolvimento integrado e na automação de tarefas comuns, como a compilação, emulação e deploy. O sistema de build do Expo permite testar rapidamente funcionalidades da aplicação diretamente em dispositivos físicos ou emuladores, por meio do aplicativo Expo Go, otimizando o ciclo de desenvolvimento (Expo Documentation, 2025). Além disso, sua biblioteca oferece suporte nativo a funcionalidades cruciais como geolocalização, câmera, notificações, armazenamento local e outras APIs móveis, sem que o desenvolvedor precise lidar com configurações específicas de cada sistema operacional.

Para o sistema de fretes desenvolvido neste projeto, o Expo se mostra uma ferramenta estratégica. O aplicativo exige recursos como geolocalização em tempo real e notificações internas, que são nativamente suportadas e facilmente implementadas via bibliotecas do próprio Expo. Sua capacidade de deploy rápido, inclusive em ambientes de teste ou produção, permite iterar versões com agilidade, contribuindo para um fluxo de desenvolvimento contínuo e eficiente.

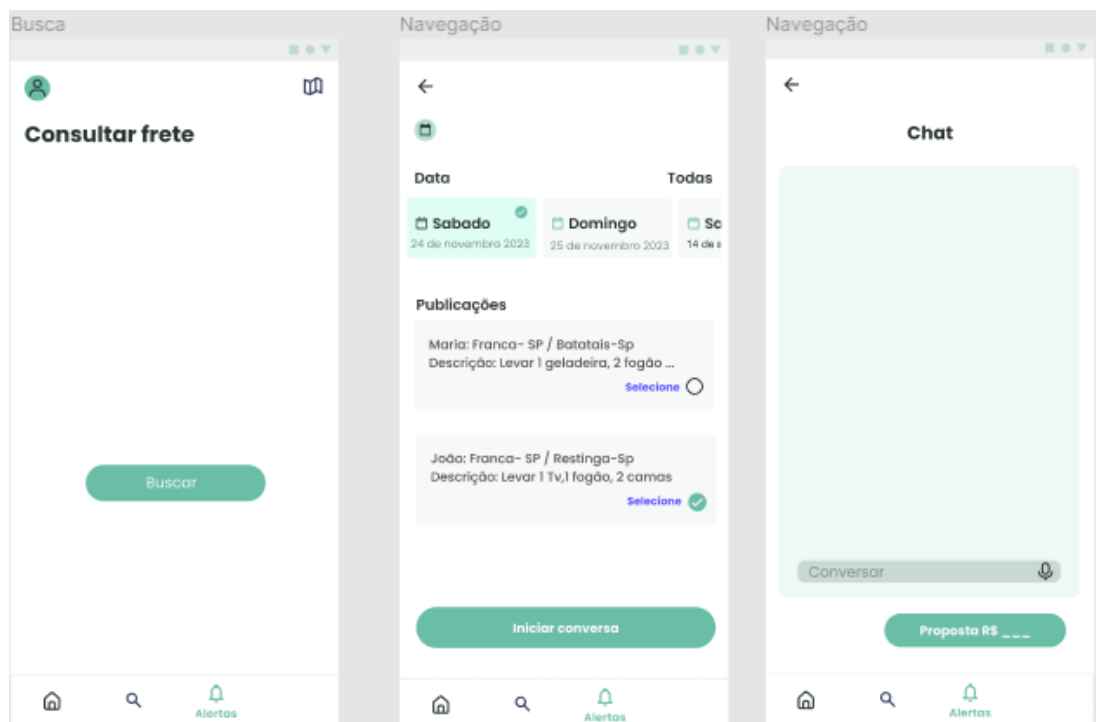
5 Desenvolvimento

No início do desenvolvimento deste projeto, foi realizada prototipação de telas com objetivo de definir o design e a funcionalidade das telas que compõem a aplicação. Para isso, utilizou-se a ferramenta Figma, amplamente reconhecida no

mercado por sua capacidade de criar protótipos interativos. Esses protótipos permitiram visualizar de forma clara e antecipada a experiência do usuário final, facilitando ajustes de interface e usabilidade ainda na etapa inicial do projeto.

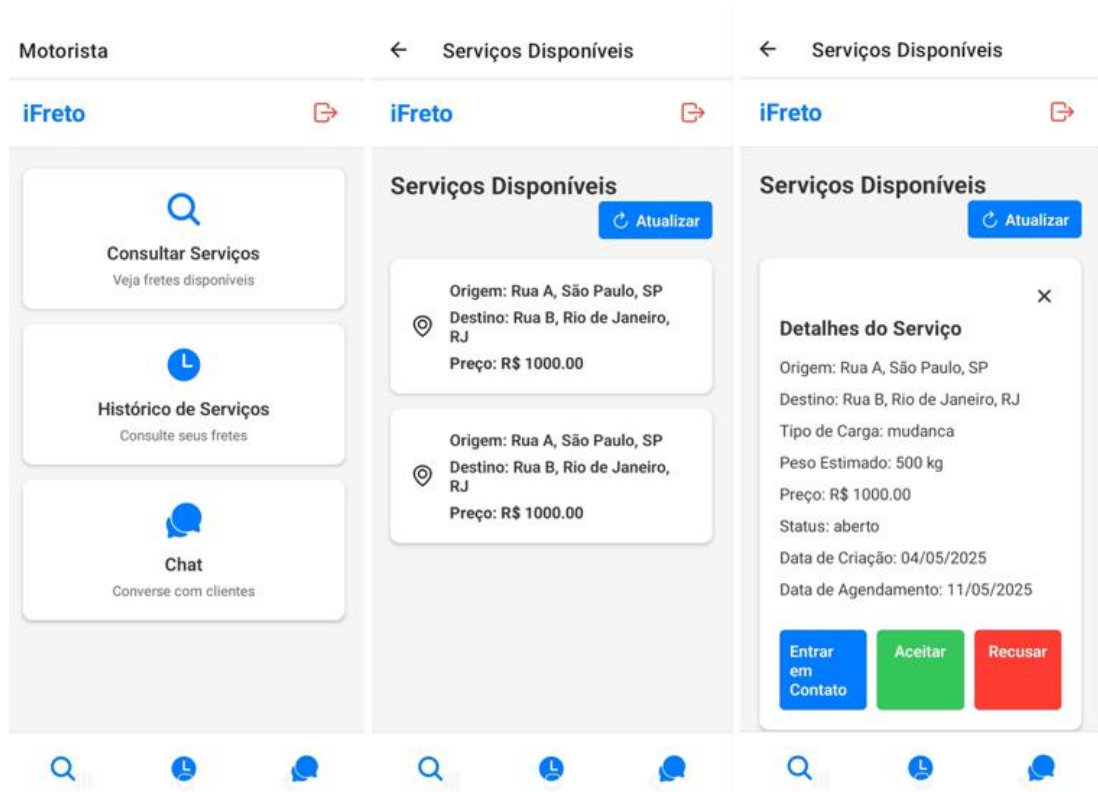
Os protótipos desenvolvidos contemplaram as principais interfaces voltadas tanto para motoristas quanto para clientes, com foco em acessibilidade, navegação intuitiva e responsividade em dispositivos móveis. A figura 5 apresenta uma das telas centrais do aplicativo, que exibe os detalhes de um frete disponível para o motorista. Nessa interface, o motorista pode visualizar informações como origem e destino da carga, tipo da carga e a data.

Figura 5 – Prototipação consultar fretes



Fonte: Autores (2025)

Como resultado da implementação, apresentamos na figura 6 uma prévia da funcionalidade de consultar fretes desta seção de motoristas. Ainda não foi implementado filtro de busca, entretanto este será o design final que será utilizado no aplicativo. Por questões de usabilidade alteramos a forma com que são exibidos os cards dos serviços disponíveis, tornando-os dinâmicos para expandir os detalhes com opções de tratativas para o frete como “Entrar em Contato”, “Aceitar” e “Recusar”.

Figura 6 – Implementação final funcionalidade consultar fretes

Fonte: Autores (2025)

Nesta seção apresentamos algumas das funcionalidades já implementadas no aplicativo de fretes, na camada de backend. Dividimos o projeto em pacotes para ordenar o código fonte por domínios úteis como "middleware", "models", "routes" e "service". No pacote "middleware" temos a função "authMiddleware" ilustrada na figura 7, que é uma camada de proteção que atua sobre as rotas do backend, garantindo que apenas usuários autenticados tenham acesso a determinados recursos da aplicação. Esta função é integrada ao pipeline do Express, sendo executada antes de controladores que exigem autenticação. O middleware realiza as seguintes operações: extração do token, validação do token, decodificação do token e tratamento de erros.

Figura 7 – AuthMiddleware

```
middleware > (); auth.js > authMiddleware
1  import jwt from 'jsonwebtoken';
2
3  export default function authMiddleware(req, res, next) {
4    const authHeader = req.headers.authorization;
5    if (!authHeader || !authHeader.startsWith('Bearer ')) {
6      return res.status(401).json({ erro: 'Token não fornecido' });
7    }
8
9    const token = authHeader.split(' ')[1];
10
11    try {
12      const decoded = jwt.verify(token, process.env.JWT_SECRET || 'sua_chave_secreta');
13      req.user = { id: decoded.id, tipo: decoded.tipo };
14      next();
15    } catch (error) {}
16    res.status(401).json({ erro: 'Token inválido' });
17  }
18 }
```

Fonte: Autores (2025)

No pacote “models” criamos os “schemas” para gerar os documentos que serão armazenados no banco de dados. Estes documentos recebem um conjunto de atributos e valores, semelhante ao formato JSON padrão. Na figura 8 abaixo ilustramos o “usuarioSchema” que é uma abstração de um usuário na aplicação. Para evitar a criação de esquemas separados para motoristas e clientes, utilizamos de apenas um objeto com a diferenciação de um atributo enum que é o tipo.

Figura 8 – UsuarioSchema

```

models > (); Usuarios.js > [x] usuarioSchema
1  import { Schema, model } from 'mongoose';
2
3  const usuarioSchema = new Schema({
4    nome: { required: true, trim: true, },
5    email: { type: String, required: true, unique: true, lowercase: true, trim: true, },
6    senha: { type: String, required: true, },
7    tipo: { type: String, enum: ['motorista', 'cliente'], required: true, },
8    telefone: { type: String, required: true, },
9    endereco: { cidade: String, estado: String, },
10   motoristaDetalhes: {
11     cnh: {
12       type: String,
13       required: function() { return this.tipo === 'motorista'; },
14     },
15     tipoVeiculo: {
16       type: String,
17       enum: ['caminhonete', 'van', 'caminhao', 'outro'],
18       required: function() { return this.tipo === 'motorista'; },
19     },
20   },
21   criadoEm: {
22     type: Date,
23     default: Date.now,
24   },
25 });
26
27 export default model('Usuario', usuarioSchema);

```

Fonte: Autores (2025)

No pacote “routes” definimos as rotas da aplicação separando por camadas as responsabilidades como “servico.js” e “usuários.js”. Nesta seção a camada de rotas do backend organiza os pontos de entrada da API, encaminhando requisições para os serviços responsáveis. Utilizando o framework Express, o arquivo “servico.js” define rotas relacionadas ao gerenciamento de fretes, como criação, aceite, início e cancelamento. Cada rota está protegida pelo “authMiddleware”, garantindo que apenas usuários autenticados possam acessá-las. As ações específicas são delegadas ao módulo “servicoService.js”, promovendo separação de responsabilidades e melhor organização do código, facilitando manutenção e escalabilidade da aplicação. A figura 9 ilustra o “servico.js”.

Figura 9 – Servico.js

```
routes > (); servicos.js > [⌕] default
1  import express from 'express';
2  import * as servico from '../service/servicoService.js';
3  import authMiddleware from '../middleware/auth.js';
4
5  const router = express.Router();
6
7  router.get('/buscar', authMiddleware, servico.listarServicos);
8  router.post('/criar', authMiddleware, servico.criarServico);
9  router.post('/:id/aceitar', authMiddleware, servico.aceitarServico);
10 router.post('/:id/rejeitar', authMiddleware, servico.rejeitarServico);
11 router.post('/:id/iniciar', authMiddleware, servico.iniciarServico);
12 router.post('/:id/concluir', authMiddleware, servico.concluirServico);
13 router.post('/:id/cancelar', authMiddleware, servico.cancelarServico);
14
15 export default router;
```

Fonte: Autores (2025)

No pacote “service” temos a camada de serviço que é responsável por implementar a lógica de negócio da aplicação, sendo um intermediário entre as rotas e o acesso direto ao banco de dados. Ela centraliza funcionalidades específicas, facilitando a reutilização de código, a manutenção e os testes unitários. Na figura 10 “usuarioService.js” apresentamos o método para cadastrar usuários que implementa o processo de criação de contas na aplicação, validando os dados enviados com a biblioteca “Joi” e verificando se o e-mail já está cadastrado. Em seguida, a senha é criptografada com bcrypt para garantir a segurança. O usuário é então criado e salvo no banco de dados MongoDB, e um token JWT é gerado para autenticação nas próximas requisições. Caso o tipo de usuário seja “motorista”, dados adicionais são incluídos. A resposta final pode ser uma mensagem de sucesso com o token ou um erro, caso ocorra alguma falha.

Figura 10 – UsuarioService.js

```

service > 0; usuarioService.js > [0] cadastrarUsuario
38
39 // Cadastro de usuário
40 export const cadastrarUsuario = async (req, res) => {
41   try {
42     // Validar dados
43     const { error } = cadastroSchema.validate(req.body, { abortEarly: false });
44     if (error) return res.status(400).json({ erro: error.details.map(detail => detail.message) });
45
46     const { nome, email, senha, tipo, telefone, endereco, motoristaDetalhes } = req.body;
47
48     // Verificar se o e-mail já existe
49     const usuarioExistente = await Usuario.findOne({ email });
50     if (usuarioExistente) return res.status(400).json({ erro: 'E-mail já cadastrado' });
51
52     // Hash da senha
53     const senhaHash = await bcrypt.hash(senha, 10);
54
55     // Criar usuário
56     const usuario = new Usuario({
57       nome, email, senha: senhaHash, tipo, telefone, endereco,
58       motoristaDetalhes: tipo === 'motorista' ? motoristaDetalhes : undefined,
59     });
60
61     await usuario.save();
62
63     // Gerar token JWT
64     const token = jwt.sign({ id: usuario._id, tipo: usuario.tipo }, process.env.JWT_SECRET, { expiresIn: '1h' });
65
66     res.status(201).json({ mensagem: 'Usuário criado', token });
67   } catch (erro) {
68     res.status(500).json({ erro: 'Erro ao cadastrar usuário' });
69   }
70 };

```

Fonte: Autores (2025)

6 Resultados e Discussão

Este capítulo dedica-se a apresentar as conquistas e as análises aprofundadas do projeto APP IFRETE. Esta seção é fundamental para demonstrar o que foi alcançado com a entrega do Produto Mínimo Viável (MVP), detalhando a validação parcial da solução desenvolvida e os retornos obtidos junto aos usuários-alvo.

Aqui, será realizada uma análise crítica do progresso em relação aos objetivos propostos, abrangendo:

A Análise de Portabilidade, que avalia a capacidade do sistema de ser executado em diferentes ambientes computacionais, bem como a infraestrutura necessária e a acessibilidade para os usuários.

Os Cálculos das Métricas de desenvolvimento, oferecendo uma avaliação quantitativa do esforço empregado, da cobertura funcional e da complexidade do código.

A Proposta Comercial do aplicativo, delineando as estratégias para monetização e sustentabilidade do negócio.

Este capítulo, portanto, consolida os achados práticos e as avaliações estratégicas que evidenciam o potencial do APP IFRETE para otimizar o transporte de cargas urbanas e contribuir para a redução da ociosidade de caminhoneiros, conforme os objetivos definidos na introdução do trabalho

6.1 Análise de Portabilidade

A análise de portabilidade avalia a capacidade do sistema de ser executado em diferentes ambientes computacionais, com foco na infraestrutura necessária, na acessibilidade por parte dos usuários e na viabilidade de implantação. No contexto do aplicativo de fretes desenvolvido, a portabilidade foi planejada considerando sua execução exclusiva em dispositivos móveis, com uso de ferramentas gratuitas e acessíveis, tanto no desenvolvimento quanto na execução. A escolha por tecnologias open-source e multiplataforma garante flexibilidade na implantação e facilidade de manutenção.

A aplicação foi construída com ferramentas que não exigem infraestrutura robusta ou investimentos em licenciamento. O backend utiliza Node.js com Express, operando localmente ou em servidores gratuitos como Railway ou Render para testes. O banco de dados é o MongoDB Compass para gerenciamento. O frontend é baseado em React Native, com suporte do Expo Go, que dispensa emuladores ou builds nativos, facilitando testes em dispositivos reais de forma simples e gratuita.

A equipe de desenvolvimento pode ser enxuta, composta por estudantes ou desenvolvedores com conhecimento básico em JavaScript, React Native, Node.js e MongoDB. Como todas as tecnologias utilizadas são amplamente documentadas e contam com grandes comunidades de apoio, o aprendizado e a solução de problemas são facilitados, sem necessidade de profissionais altamente especializados.

O aplicativo funciona a partir de dispositivos Android 8.0 (Oreo) ou iOS 12, e requer apenas a instalação do aplicativo Expo Go para testes ou uso direto durante o desenvolvimento. O consumo de armazenamento é leve, em torno de 100MB, e o sistema exige conexão à internet para comunicação com o backend.

O público-alvo, composto por motoristas e clientes, pode utilizar smartphone com configurações acessíveis:

- Processador quad-core 1.4GHz
- 2GB de RAM

- Andoid ou iOS
- Armazenamento interno com ao menos 200MB livres
- Expo Go instalado via Play Store ou App Store
- Conexão com internet móvel ou Wi-Fi

A aplicação pode ser hospedada localmente ou em serviços gratuitos com as seguintes configurações mínimas:

- Node.js instalado (versão LTS)
- 2CPU, 2GB de RAM, armazenamento de 100GB
- MongoDB Server Community local
- Acesso via terminal e navegador
- Ambiente de execução configurado com ferramentas como VSCode e Postman

6.2 Métricas

A avaliação métrica deste projeto baseou-se em uma abordagem prática e simplificada, adequada ao contexto acadêmico e ao escopo de um aplicativo mobile. Essa estratégia foi adotada por permitir uma análise objetiva sem recorrer a metodologias complexas como Use Case Points (UCP), que exigem parâmetros nem sempre disponíveis com precisão em projetos de pequena escala.

Inicialmente contabilizou-se um total de 15 casos de uso implementados, cobrindo os 16 requisitos funcionais documentados no início do projeto. Com isso obteve-se uma cobertura funcional de 100%, assegurando que todas as funcionalidades previstas foram abordadas na aplicação.

Para estimar o esforço de desenvolvimento, considerou-se uma média de 7 horas por caso de uso, totalizando aproximadamente 105 horas de trabalho efetivo (7 horas x 15 casos de uso). Esse valor inclui atividade como análise de requisitos, codificação, testes e validação de cada funcionalidade. Com base em uma taxa estimada de R\$25,00 por hora de trabalho, o custo de desenvolvimento estimado do sistema foi de R\$ 2.625,00 (25 reais/hora * 105 horas).

Quanto a complexidade do código, adotou-se uma estimativa qualitativa com base na complexidade ciclomática. Funcionalidades como validação de documentos via OCR e uso de geolocalização apresentam lógicas de controle mais elaboradas, classificadas como complexidade média a alta. Para o conjunto do sistema, a média

de complexidade situou-se entre 5 e 6, considerando também funcionalidades CRUD e autenticação.

6.3 Proposta Comercial

A presente proposta tem como objetivo apresentar uma solução tecnológica que conecta motoristas de frete a clientes com demandas de transporte urbano, como mudanças ou entregas. O aplicativo foi concebido com foco em acessibilidade, escalabilidade e usabilidade, utilizando ferramentas gratuitas e amplamente suportadas pela comunidade de desenvolvimento.

6.3.1 Solução proposta

A solução consiste em um aplicativo mobile, desenvolvido com React Native e Expo GO, com backend construído em Node.js e banco de dados MongoDB Community Edition. A proposta visa criar um ecossistema digital que permita a publicação, aceite e acompanhamento de serviços de frete, com validação segura de motoristas, comunicação entre usuários e histórico de serviços prestados.

6.3.2 Visão geral da solução

O sistema foi desenvolvido com foco em dispositivos móveis, oferecendo uma interface intuitiva e leve. Sua arquitetura contempla:

- Cadastro e autenticação de usuários (motoristas e clientes)
- Publicação e consulta de fretes
- Confirmação de aceite de serviços
- Acompanhamento com base em geolocalização
- Sistema interno de avaliação de motoristas
- Upload e validação de documentos (CNH e foto de perfil)
- Histórico de serviços
- Notificações internas
- Comunicação com cliente via canal externo (WhatsApp)

6.3.3 Escopo da solução

A solução conta com os seguintes módulos e entregas técnicas:

- Aplicativo mobile multiplataforma (iOS e Android)
- Backend com autenticação JWT e middleware de segurança
- Banco de dados não relacional com MongoDB Community
- Cadastro e login de motoristas e clientes
- CRUD completo de fretes
- Canal de contato via integração externa com WhatsApp
- Upload de documentos com validação básica
- Integração com APIs de localização e navegação
- Painel administrativo simplificado (em planejamento para versão futura)

6.3.4 Prazos

O projeto seguiu o seguinte cronograma:

Início do projeto: 03/02/2025

Encerramento: 90 dias úteis

6.3.5 Investimento

A proposta comercial da aplicação de fretes visa oferecer uma solução completa para conectar motoristas autônomos e clientes com demandas locais, como mudanças residenciais e entregas urbanas. O investimento estimado nesta fase inicial está segmentado em dois blocos, custo de desenvolvimento e custo operacional para entrada no mercado.

Com base em 105 horas estimadas de trabalho distribuídas entre dois desenvolvedores, o valor do desenvolvimento da aplicação, utilizando ferramentas gratuitas como Node.js, React Native e MongoDB Community, é de:

- R\$ 2.625,00 podendo ser dividido em até 3 vezes sem juros.
- Parcelamento superior a 3 vezes, receberá taxa de juros de 7% em cima do valor de cada mensalidade.

Além do desenvolvimento técnico, um lançamento funcional exige investimentos complementares para que a aplicação se torne viável comercialmente. Elencamos na tabela 1 os principais itens para que a aplicação entre em produção.

Tabela 1 – Custos operacionais para entrada no mercado

Artefato	Item	Estimativa em R\$
Hospedagem Servidor	VPS	30,00 ~ 50,00 / mês
Banco de dados	MongoDB Atlas	50,00 ~ 160,00 / mês
Publicação	Google Play Store	129,00 / pag. único
Publicação	App Store (iOS)	~ 500,00 / ano
Campanha marketing	Ads, mídia local, parcerias	500,00 ~ 1200,00 / inv. único
Suporte técnico e administração	Responsável técnico	~ 400,00 / mês (até 10 horas de suporte)

Fonte – Autores (2025)

Para viabilizar a sustentabilidade da aplicação e garantir retorno financeiro ao empreendedor, foram previstas as seguintes estratégias, com base no modelo de negócios proposto:

- Comissão por frete intermediado diretamente na plataforma, com taxa fixa por cada frete concluído (ex.: 10%).
- Plano de fidelização para motoristas, versão gratuita com anúncios e versão paga sem anúncios com funcionalidades extras a combinar.
- Venda de espaço publicitário, com anúncios regionais no app para oficinas, postos, seguradoras e outros serviços úteis aos motoristas.
- Plano empresarial para empresas com demandas de fretes, com pagamento mensal.

Com uma base inicial de 100 usuários ativos (50 motoristas e 50 clientes), considerando 2 fretes semanais por motorista com comissão de R\$ 5,00 por frete, a plataforma já poderia gerar aproximadamente R\$ 2.000,00 mensais em comissões.

Considerações finais

O desenvolvimento do aplicativo de fretes teve como objetivo principal criar uma solução digital acessível e funcional para conectar motoristas de caminhão a clientes com demandas urbanas, como entregas e mudanças. A proposta visa atender a um problema real de logística local, oferecendo uma ferramenta que facilite a intermediação de serviços de transporte de forma segura, eficiente e transparente.

Durante a execução do projeto, foi possível implementar uma base sólida composta por backend em Node.js com integração ao MondoDB, além de uma interface mobile desenvolvida em React Native com apoio da plataforma Expo. Entre os recursos desenvolvidos estão: cadastro e autenticação de usuários, publicação e

aceite de fretes, sistema de avaliação de motoristas, notificações internas e históricos de serviços. Também foi estabelecida uma estrutura de middleware para autenticação via JWT e rotas protegidas, garantindo maior segurança aos usuários.

Os principais desafios enfrentados incluíram a implementação de validações mais avançadas, como a leitura da CNH via OCR, e a limitação de tempo para entregar todas as funcionalidades previstas inicialmente. Além disso, como o projeto foi desenvolvido por uma equipe enxuta de dois integrantes, foi necessário priorizar tarefas críticas e adotar soluções escaláveis com ferramentas gratuitas e de fácil manutenção.

O projeto, no entanto, abre espaço para diversas evoluções futuras. Entre as melhorias previstas estão a implementação completa da geolocalização com rastreamento em tempo real, envio automatizado de documentos com validação visual, integração direta com o WhatsApp para comunicação entre cliente e motorista, painel administrativo com estatísticas de uso, e a monetização por meio de anúncios e comissão por serviço intermediado. A proposta comercial já delineada também mostra viabilidade para transformar o aplicativo em um produto sustentável, com planos de expansão e fidelização de usuários.

Concluimos, portanto, que o projeto atendeu aos objetivos propostos em sua fase inicial, com alto potencial de escalabilidade e continuidade, configurando-se como uma solução promissora para o setor de fretes urbanos.

Referências

AKHTER, M.; ROY, S. Cross-Platform Mobile App Development with React Native. *International Journal of Advanced Computer Science and Applications*, v. 11, n. 6, 2020.

ANDERSON, David J. Kanban: Mudança Evolucionária de Sucesso para seu Negócio de Tecnologia. Tradução: Rafael Santos. São Paulo: Novatec, 2011.

CASTRO, L. F.; MOREIRA, F. R. Engenharia de Software: Práticas e Fundamentos para Desenvolvimento Ágil. Rio de Janeiro: Ciência Moderna, 2022.

CHODOROW, Kristina. MongoDB: The Definitive Guide. 3. ed. Sebastopol: O'Reilly Media, 2019.

CHODOROW, Kristina. Scaling MongoDB. Sebastopol: O'Reilly Media, 2013.

Confederação Nacional do Transporte (CNT). Perfil dos caminhoneiros. Brasília: CNT, 2019. Disponível em: <https://observatoriotransporte.cnt.org.br/estudo/perfil-dos-caminhoneiros-2019/>. Acesso em: 27 maio 2025.

DUMAS, M.; LA ROSA, M.; MENDLING, J.; REIJERS, H. A. Fundamentals of Business Process Management. Cham: Springer, 2018.

EXPO. Expo Documentation – API reference & Guides. 2025. Disponível em: <https://docs.expo.dev/>. Acesso em: 20 jun. 2025.

FACEBOOK. React Native – Create native apps for Android and iOS using React. Disponível em: <https://reactnative.dev/>. Acesso em: 20 jun. 2025.

GARCÍA, M.; KACZMAREK, T. Business Rule Management Systems and Their Role in Modern Organizations. Journal of Business Process Management, v. 28, n. 4, 2020.

GUEDES, G. T. A. UML 2 – Uma Abordagem Prática. 3. ed. São Paulo: Novatec Editora, 2018.

HILL, Charles W. L.; JONES, Gareth R. Strategic Management: An Integrated Approach. 10. ed. Mason, OH: Cengage Learning, 2012.

IBM. O que são regras de negócios?. Disponível em: <https://www.ibm.com/br-pt/topics/business-rules>. Acesso em: 19 jun. 2025.

JACOBSON, Ivar; CHRISTERSON, Magnus; JONSSON, Patrik; ÖVERGAARD, Gunnar. Object-Oriented Software Engineering: A Use Case Driven Approach. Reading, MA: Addison-Wesley, 1992.

KERZNER, Harold. Gestão de Projetos: As melhores práticas. 3. ed. Porto Alegre: Bookman, 2017.

KOTLER, Philip; KELLER, Kevin Lane. Administração de marketing. 14. ed. São Paulo: Pearson Prentice Hall, 2012.

Object Management Group (OMG). Business Process Model and Notation (BPMN) Version 2.0. 2011.

OLIVEIRA, Edi Carlos de (Org.). Administração: Técnicas e ferramentas para gestão organizacional. Paraná: Atena Editora, 2024.

OSTERWALDER, Alexander; PIGNEUR, Yves. Business Model Generation: Inovação Em Modelos De Negócios. 1. ed. Rio de Janeiro: Alta Books, 2011

PMI – Project Management Institute. Guia PMBOK® – Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 7. ed. Pennsylvania: Project Management Institute, 2021.

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

RUBINOFF, G.; SHARMA, R. Business Rules for the Modern Enterprise: A Framework for Operational Efficiency. Cham: Springer, 2021.

SOMMAVILLA, R.; PERES, F. Qualidade de Software: Princípios e Práticas. Florianópolis: Visual Books, 2019.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Addison Wesley, 2011.

WIEGERS, Karl E.; BEATTY, Joy. Software Requirements. 3rd ed. Redmond: Microsoft Press, 2013.

Apêndice 1 – Levantamento Inicial de Requisitos

Perguntas Realizadas (baseadas em conversas informais com motoristas)

1. Quais são as maiores dificuldades que você enfrenta para conseguir fretes atualmente?
2. Como você costuma ser contatado por pessoas ou empresas que precisam de transporte?
3. Você já ficou parado ou teve que se deslocar até outro local esperando por carga? Como isso afetou seu dia?
4. Que tipo de informações você gostaria de visualizar antes de aceitar um frete?
5. Você se sentiria confortável em ser avaliado por contratantes? E em avaliar também?
6. Um aplicativo que conecte você diretamente com clientes te ajudaria? O que ele deveria ter para ser útil?
7. Você costuma usar WhatsApp para negociações? Isso funciona bem?
8. Você se sentiria mais seguro se houvesse algum tipo de verificação de identidade no aplicativo?

Respostas Representativas

“Muitas vezes eu fico no posto esperando alguém aparecer, e nem sempre tem carga. É tempo e diesel perdido.”

“A gente se fala muito no boca a boca ou por WhatsApp, mas é difícil saber quem precisa de frete.”

“Já fui até outra cidade só pra tentar pegar uma carga de volta, mas cheguei lá e não tinha nada.”

“Antes de pegar um frete, preciso saber se compensa: peso, destino, contato confiável.”

“WhatsApp é o que mais uso. Se o aplicativo já abrir direto uma conversa, facilita demais.”

“Se tiver um sistema de avaliação, ajuda a saber quem é sério, tanto motorista quanto cliente.”

“Ter uma forma de ver se a pessoa é mesmo quem diz ser ajuda a evitar dor de cabeça.”

Histórias de Usuário (User Stories)

- Como contratante, quero publicar uma demanda de frete com origem, destino e tipo de carga, para que motoristas interessados possam visualizar e entrar em contato.
- Como motorista, quero ver uma lista de fretes disponíveis próximos de mim, para escolher qual se encaixa melhor na minha rota e tipo de caminhão.
- Como motorista, quero entrar em contato rapidamente com o contratante via WhatsApp, para negociar o serviço diretamente.
- Como contratante, quero avaliar o serviço do motorista após a entrega, para ajudar outros usuários a tomarem decisões futuras.
- Como usuário, quero que o sistema verifique a identidade dos motoristas, para garantir maior segurança nas negociações.
- Como administrador, quero acompanhar as demandas cadastradas e os usuários ativos, para monitorar a saúde do sistema.