



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS PICOS
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JEAN CARLOS RODRIGUES SOUSA

VIABUS: PLATAFORMA PARA GESTÃO INTEGRADA E DIGITALIZAÇÃO DO
TRANSPORTE RODOVIÁRIO ALTERNATIVO DE PASSAGEIROS

PICOS, PIAUÍ
2025

JEAN CARLOS RODRIGUES SOUSA

VIABUS: PLATAFORMA PARA GESTÃO INTEGRADA E DIGITALIZAÇÃO DO
TRANSPORTE RODOVIÁRIO ALTERNATIVO DE PASSAGEIROS

Trabalho de Conclusão de Curso (Relatório Técnico de Software) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos.

Orientador: Prof. M^e. Aislan Rafael Rodrigues de Sousa

PICOS, PIAUÍ
2025

Dedico este trabalho à minha família, cujo apoio e incentivo tornaram possível esta
nova jornada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder força e perseverança ao longo desta caminhada.

Ao meu orientador, Prof. Me. Aislan Rafael Rodrigues de Sousa, pelo apoio, paciência e orientação fundamentais para a realização deste trabalho.

Aos meus pais, Josué José Filho e Maria de Fátima Rodrigues, por todo o apoio, suporte e incentivo ao longo da minha jornada acadêmica.

Aos professores do curso de Análise e Desenvolvimento de Sistemas do IFPI – Campus Picos, por compartilharem conhecimento e contribuírem para minha formação profissional.

E, com imensa gratidão, à minha namorada Karielly de Carvalho, por estar ao meu lado, apoiando-me em todos os momentos.

A todos que, de alguma forma, fizeram parte desta jornada, meu sincero agradecimento.

"Software é uma grande combinação de arte e engenharia."
Bill Gates

RESUMO

O setor de transporte rodoviário alternativo de passageiros carece de soluções tecnológicas acessíveis, resultando em uma gestão predominantemente manual, fragmentada e ineficiente. Este trabalho aborda este problema através do projeto e desenvolvimento de um protótipo funcional de uma plataforma *web*, denominada ViaBus, no modelo *Software as a Service* (SaaS). A metodologia adotada partiu de uma pesquisa de mercado qualitativa para o levantamento de requisitos, seguida pela modelagem de uma arquitetura de *software multi-tenant* e a implementação do protótipo com as tecnologias NestJS para o *backend* e Next.js para o *frontend*. Como resultado, foi entregue uma aplicação funcional que centraliza o gerenciamento de frotas, motoristas, rotas e a venda de passagens, com a verificação de requisitos confirmando a implementação do núcleo operacional. Adicionalmente, uma avaliação heurística da interface indicou uma base de usabilidade sólida, com forte aderência a princípios de consistência e clareza, embora tenha apontado oportunidades de melhoria na prevenção de erros. Conclui-se que a abordagem SaaS com foco na simplicidade é uma solução tecnicamente viável para o nicho de mercado estudado, e que o protótipo ViaBus serve como uma prova de conceito eficaz dessa proposta, com suas limitações e oportunidades de trabalhos futuros devidamente documentadas.

Palavras-chave: Transporte de Passageiros; Software como Serviço; Sistema de Gestão; Arquitetura de Software.

ABSTRACT

The alternative road passenger transport sector lacks accessible technological solutions, resulting in predominantly manual, fragmented, and inefficient management. This work addresses this problem through the design and development of a functional prototype of a web platform, named ViaBus, based on the Software as a Service (SaaS) model. The adopted methodology started with a qualitative market research for requirements gathering, followed by the modeling of a multi-tenant software architecture and the implementation of the prototype using NestJS for the backend and Next.js for the frontend. As a result, a functional application was delivered that centralizes the management of fleets, drivers, routes, and ticket sales, with the requirements verification confirming the implementation of the operational core. Additionally, a heuristic evaluation of the interface indicated a solid usability foundation, with strong adherence to principles of consistency and clarity, although it pointed out opportunities for improvement in error prevention. It is concluded that the SaaS approach focused on simplicity is a technically feasible solution for the studied market niche, and that the ViaBus prototype serves as an effective proof of concept for this proposal, with its limitations and opportunities for future work duly documented.

Keywords: Passenger Transportation; Software as a Service; Management System; Software Architecture.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão de alto nível da solução.	26
Figura 2 – Agregado organizacional - estrutura multiempresa.	28
Figura 3 – Agregado de configuração de rotas, paradas e horários.	28
Figura 4 – Agregado de recursos operacionais - frota e motoristas.	29
Figura 5 – Agregado de operação - viagens, recursos e vendas.	30
Figura 6 – Organização modular do backend ViaBus.	30
Figura 7 – Pipeline de processamento de requisições.	31
Figura 8 – Diagrama de sequência - fluxo de autenticação.	32
Figura 9 – Arquitetura multi-tenant: uma única aplicação com dados logicamente isolados por empresa no banco de dados.	33
Figura 10 – Arquitetura em camadas do frontend e fluxo de dados principal.	35
Figura 11 – Tela de login do sistema ViaBus.	38
Figura 12 – Formulário de criação de empresa.	38
Figura 13 – Painel de controle principal com métricas e navegação.	39
Figura 14 – Listagem de paradas com funcionalidades de busca e filtros.	40
Figura 15 – Formulário de cadastro de parada com mapa interativo.	40
Figura 16 – Listagem de rotas com funcionalidades de busca e filtros.	41
Figura 17 – Interface de criação de rota com seleção de paradas.	41
Figura 18 – Listagem de veículos com status operacional.	42
Figura 19 – Interface de gerenciamento de motoristas.	42
Figura 20 – Processo de agendamento de viagem.	43
Figura 21 – Seleção de rota no wizard de passagens.	44
Figura 22 – Seleção de data e horário.	44
Figura 23 – Cadastro de informações do passageiro.	45
Figura 24 – Seleção de locais de embarque e desembarque.	45
Figura 25 – Confirmação final e geração da passagem.	46
Figura 26 – Resultado da pergunta sobre empresa ou serviço.	60
Figura 27 – Resultado da pergunta sobre tamanho da frota de veículos.	60
Figura 28 – Resultado da pergunta sobre número de rotas operadas.	60
Figura 29 – Resultado da pergunta sobre tipo de operação.	61
Figura 30 – Resultado da pergunta sobre gerenciamento atual de passagens.	61
Figura 31 – Resultado da pergunta sobre maiores desafios operacionais.	61
Figura 32 – Resultado da pergunta sobre valor de um sistema único.	62
Figura 33 – Resultado da pergunta sobre funcionalidades mais importantes.	62
Figura 34 – Resultado da pergunta sobre aplicativo como diferencial.	62
Figura 35 – Resultado da pergunta sobre modelo de parceria.	63

Figura 36 – Resultado da pergunta sobre interesse em testar a plataforma. . . .	63
Figura 37 – Diagrama de classes do domínio ViaBus.	64

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais do sistema ViaBus	23
Tabela 2 – Requisitos não funcionais do sistema ViaBus	23
Tabela 3 – Tecnologias utilizadas no Backend	24
Tabela 4 – Tecnologias utilizadas no Frontend	25
Tabela 5 – Infraestrutura e Ferramentas de Suporte	25
Tabela 6 – Principais entidades do domínio.	27
Tabela 7 – Verificação dos Requisitos Funcionais (RF)	47
Tabela 8 – Verificação dos Requisitos Não Funcionais (RNF)	48

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ANTT	Agência Nacional de Transportes Terrestres
CAPEX	Capital Expenditure
CI/CD	Continuous Integration/Continuous Delivery
CNH	Carteira Nacional de Habilitação
CNPJ	Cadastro Nacional da Pessoa Jurídica
CPF	Cadastro de Pessoas Físicas
CRUD	Create, Read, Update, Delete
DTO	Data Transfer Object
GCP	Google Cloud Platform
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IBGE	Instituto Brasileiro de Geografia e Estatística
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
JWT	JSON Web Token
ORM	Object-Relational Mapping
OPEX	Operational Expenditure
OSM	OpenStreetMap
PME	Pequenas e Médias Empresas
RF	Requisito Funcional
RNF	Requisito Não Funcional
SaaS	Software as a Service
SSR	Server-Side Rendering

SGBD	Sistema de Gerenciamento de Banco de Dados
SLA	Service Level Agreement
SQL	Structured Query Language
SSL	Secure Sockets Layer
STRIP/PI	Sistema de Transporte Rodoviário Intermunicipal de Passageiros do Estado do Piauí
TCC	Trabalho de Conclusão de Curso
TLS	Transport Layer Security
UI	User Interface
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	16
1.2	Objetivos	16
1.2.1	Geral	16
1.2.2	Específicos	16
1.3	Metodologia	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Software as a Service (SaaS)	18
2.1.1	Arquitetura e Modelo de Negócio	18
2.1.2	Vantagens e Desafios para PMEs	19
2.2	Tecnologia e Inovação no Transporte Rodoviário	19
2.3	Arquitetura Tecnológica da Solução Proposta	20
2.3.1	Tecnologias do Backend	20
2.3.1.1	NestJS e Arquitetura Modular	20
2.3.1.2	TypeORM e Mapeamento Objeto-Relacional (ORM)	20
2.3.1.3	Autenticação com JWT e Passport.js	20
2.3.2	Tecnologias do Frontend	21
2.3.2.1	Next.js, SSR e App Router	21
2.3.2.2	React, TypeScript e Ecossistema de UI	21
3	REQUISITOS DO SISTEMA	22
3.1	Requisitos Funcionais	22
3.2	Requisitos Não Funcionais	22
4	TECNOLOGIAS ENVOLVIDAS	24
4.1	Tecnologias do Backend	24
4.2	Tecnologias do Frontend	24
4.3	Infraestrutura e Ferramentas de Suporte	25
5	ARQUITETURA E MODELAGEM DO SISTEMA	26
5.1	Arquitetura da Solução	26
5.2	Modelagem do Banco de Dados	26
5.2.1	Diagramas UML Fracionados por Agregado	27
5.3	Arquitetura do Backend	30
5.3.1	Organização Modular	30
5.3.2	Pipeline de Processamento	31

5.3.3	Fluxo de Autenticação	31
5.3.4	Arquitetura Multi-tenant	32
5.4	Arquitetura do Frontend	33
5.4.1	Stack Tecnológico e Organização	33
5.4.2	Arquitetura de Roteamento	34
5.4.3	Gerenciamento de Estado e Contextos	34
5.4.4	Sistema de Autenticação e Comunicação	34
5.4.5	Arquitetura de Componentes	35
5.4.6	Tipagem e Validação	35
5.5	Estratégia de Implantação	36
6	APRESENTAÇÃO DO PROTÓTIPO FUNCIONAL	37
6.1	Onboarding e Visão Geral do Sistema	37
6.1.0.1	Autenticação e Criação de Empresa	37
6.1.0.2	Painel de Controle (Dashboard)	39
6.2	Gerenciamento de Recursos Operacionais	39
6.2.0.1	Gestão de Paradas e Rotas	39
6.2.0.2	Gestão da Frota e de Motoristas	41
6.3	Execução das Operações Diárias	42
6.3.0.1	Agendamento de Viagens	42
6.3.0.2	Assistente de Venda de Passagens	43
7	RESULTADOS E DISCUSSÃO	47
7.1	Verificação de Requisitos	47
7.2	Avaliação Heurística da Interface	48
7.2.0.1	Pontos Fortes da Interface	48
7.2.0.2	Oportunidades de Melhoria	49
7.3	Discussão dos Resultados	50
8	CONSIDERAÇÕES FINAIS	52
8.1	Conclusão	52
8.2	Limitações do Trabalho	52
8.3	Trabalhos Futuros	52
	Referências	54
	APÊNDICE A – QUESTIONÁRIO DA PESQUISA DE MERCADO .	57
	APÊNDICE B – RESULTADOS DA PESQUISA DE MERCADO . .	60
	APÊNDICE C – DIAGRAMA DE CLASSE DO VIABUS	64

APÊNDICE D – REPOSITÓRIOS DO CÓDIGO-FONTE	65
--	-----------

1 INTRODUÇÃO

Em um país de dimensões continentais, o transporte rodoviário de passageiros funciona como um elemento estratégico para a integração socioeconômica, conectando municípios e garantindo a mobilidade da população (FGV Transportes, 2023). Enquanto o sistema interestadual é regulamentado em nível federal pela Agência Nacional de Transportes Terrestres (ANTT), conforme estabelecido em sua lei de criação (Brasil, 2001), uma vasta e heterogênea rede de transporte intermunicipal opera sob jurisdição estadual.

No estado do Piauí, esta modalidade é uma realidade consolidada, cuja organização é definida pela Lei Nº 8.562 de 2025, que dispõe sobre o Sistema de Transporte Rodoviário Intermunicipal de Passageiros do Estado do Piauí (STRIP/PI). A referida lei classifica o "serviço alternativo" como uma das categorias oficiais do sistema, designando a Secretaria dos Transportes (SETRANS) como o poder concedente (Estado do Piauí, 2025). Este serviço, prestado por veículos de menor porte, surge para suprir as lacunas deixadas pelo sistema convencional. A existência de uma legislação específica demonstra a relevância do setor, mas também evidencia a necessidade de ferramentas de gestão que se adaptem às suas particularidades operacionais.

Diante desse cenário, este trabalho propõe o desenvolvimento de uma plataforma baseada no modelo Software como Serviço (SaaS, *Software as a Service*) para a gestão integrada de empresas de transporte rodoviário alternativo. A escolha por este modelo se fundamenta na definição de Chong e Carraro (2006 apud Melo *et al.* (2007)) como um "*software* implementado como um serviço hospedado e acessado pela Internet", o que permite que empresas utilizem soluções baseadas em nuvem com custos operacionais reduzidos e alta escalabilidade — características especialmente vantajosas para o setor em foco.

Este cenário, onde um serviço regulamentado e essencial como o transporte alternativo ainda opera com uma gestão predominantemente analógica — fato constatado na pesquisa de mercado realizada para este trabalho (Apêndice B) —, evidencia os desafios para a modernização do setor. A ausência de sistemas de gestão integrados e a consequente dependência de processos manuais não apenas comprometem a eficiência operacional e a qualidade dos serviços prestados, mas também criam uma barreira para a inovação e o crescimento sustentável das empresas que atuam nesse segmento.

1.1 JUSTIFICATIVA

O transporte rodoviário alternativo de passageiros no Brasil, embora essencial para a conectividade regional, opera de forma majoritariamente analógica e fragmentada. Uma pesquisa de mercado realizada para este estudo (Apêndice B) revelou que a gestão de frotas e a venda de passagens ainda são fortemente dependentes de processos manuais, como o uso de cadernos de anotações e planilhas. Essa carência de ferramentas digitais integradas limita a eficiência e o potencial de crescimento do setor. Neste contexto, a adoção de plataformas SaaS surge como uma solução estratégica para transformar esse cenário, otimizando recursos e reduzindo custos operacionais. Estudos de mercado mais amplos corroboram essa visão, indicando que tecnologias no setor de transportes podem elevar a eficiência logística em até 15% (SETCEPAR, 2023).

Além disso, uma plataforma SaaS voltada para o transporte alternativo pode facilitar o acesso à inovação para pequenas e médias empresas, sem exigir altos investimentos. Esse modelo permite automação da bilhetagem, gestão integrada de rotas e melhor experiência para o passageiro. Essas soluções já demonstram impacto positivo em outros segmentos do transporte, aumentando a competitividade e garantindo serviços mais confiáveis (PrologApp, 2024b). Assim, a digitalização do setor não só fortalece as empresas, mas também melhora a mobilidade interurbana, tornando os serviços mais eficientes e acessíveis.

1.2 OBJETIVOS

1.2.1 Geral

Desenvolver um protótipo funcional de uma plataforma SaaS para a gestão integrada do transporte rodoviário alternativo de passageiros, verificando a implementação de suas funcionalidades essenciais e avaliando a usabilidade de sua interface a partir de princípios de design.

1.2.2 Específicos

- Realizar uma pesquisa de mercado para identificar as dores e os processos manuais de empresas do setor de transporte alternativo;
- Especificar os requisitos funcionais e não funcionais de um sistema de gestão com base nas necessidades levantadas na pesquisa;
- Desenvolver um protótipo funcional da plataforma ViaBus, implementando os módulos de gestão de rotas, paradas, veículos, motoristas e um fluxo para agendamento de passagens;

- Realizar uma verificação técnica do protótipo para analisar a aderência do software aos requisitos especificados e conduzir uma avaliação heurística para identificar possíveis melhorias de usabilidade na interface.

1.3 METODOLOGIA

A metodologia empregada para a concepção e desenvolvimento do sistema ViaBus foi estruturada em três etapas sequenciais: levantamento de requisitos, desenvolvimento do protótipo e avaliação da solução.

A primeira etapa, de levantamento de requisitos, utilizou uma abordagem mista, partindo da observação empírica do autor sobre os desafios do setor de transporte alternativo. Tais observações foram então validadas por meio de uma pesquisa de mercado qualitativa. Para tal, foi elaborado um questionário online (detalhado no Apêndice A) e aplicado junto a gestores de duas empresas do setor. As respostas, consolidadas na tabela do Apêndice B, foram essenciais para a especificação formal dos requisitos que nortearam o desenvolvimento.

A segunda etapa, de desenvolvimento do protótipo, seguiu um processo iterativo alinhado a práticas de prototipagem evolutiva, onde o software é construído e refinado em ciclos contínuos (Sommerville, 2011). Adotou-se a estratégia *Frontend-First*, que prioriza a construção da interface do usuário (UI) como guia para o desenvolvimento do sistema. Utilizando o framework Next.js e a biblioteca de componentes shadcn/ui, todas as telas da aplicação foram inicialmente desenvolvidas com dados estáticos (*mockados*), permitindo a validação dos fluxos de interação antes da implementação da lógica de negócio no backend.

Por fim, a terceira etapa consistiu na avaliação do protótipo. Devido à impossibilidade de realizar testes com usuários finais, optou-se por uma verificação interna em duas frentes. A primeira foi uma *Verificação de Requisitos*, na qual se analisou sistematicamente o atendimento aos requisitos funcionais e não funcionais. A segunda foi uma *Avaliação Heurística*, um método de inspeção consolidado para encontrar problemas de usabilidade em interfaces (Nielsen, 1994). Atuando como avaliador especialista, o autor inspecionou o sistema com base nas 10 heurísticas de usabilidade de Nielsen, cujos resultados detalhados são apresentados no Capítulo 7.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos essenciais que fundamentam o desenvolvimento deste trabalho. A primeira seção aborda o modelo *Software as a Service* (SaaS), detalhando sua definição, arquitetura, vantagens e desafios. A segunda seção explora a aplicação de tecnologias no setor de transporte rodoviário, conectando a teoria à prática observada. Por fim, a terceira seção aprofunda-se na arquitetura e nas tecnologias específicas escolhidas para a construção do protótipo ViaBus.

2.1 SOFTWARE AS A SERVICE (SAAS)

O modelo *Software as a Service* (SaaS), ou *Software* como Serviço, representa uma mudança de paradigma na forma como o *software* é distribuído e consumido. Diferentemente do modelo tradicional *on-premise*, onde o cliente adquire licenças e é responsável pela infraestrutura, no modelo SaaS o cliente paga uma assinatura periódica para acessar a aplicação pela internet, que é hospedada na nuvem pelo provedor do serviço (Moveideias, 2025).

Nesse modelo, toda a infraestrutura subjacente — servidores, armazenamento, redes e o próprio *software* — é gerenciada pelo provedor. Isso significa que o fornecedor é responsável pela manutenção, atualizações, segurança e disponibilidade da aplicação, permitindo que as empresas clientes foquem em suas atividades principais sem se preocupar com a complexidade da gestão de TI (Moveideias, 2025).

2.1.1 Arquitetura e Modelo de Negócio

A arquitetura mais comum em soluções SaaS é a *multi-tenancy* (multilocação), onde uma única instância da aplicação e da infraestrutura serve a múltiplos clientes (locatários ou *tenants*) (Frontegg, 2021). Embora compartilhem os mesmos recursos computacionais, os dados de cada cliente são mantidos isolados e seguros, garantindo a privacidade e a confidencialidade das informações. Essa abordagem permite que o provedor otimize os recursos e reduza os custos, o que se reflete em preços mais acessíveis para o cliente final.

O modelo de negócio é baseado em assinaturas, que podem variar em preço conforme o número de usuários, os recursos contratados ou o volume de uso. Essa flexibilidade oferece escalabilidade, permitindo que as empresas ajustem o serviço de acordo com seu crescimento e suas necessidades, pagando apenas pelo que utilizam (Prolog App, 2024a).

2.1.2 Vantagens e Desafios para PMEs

A adoção de plataformas SaaS oferece um conjunto significativo de vantagens estratégicas para as Pequenas e Médias Empresas (PMEs). A principal delas é a drástica redução de custos, pois o modelo de assinatura elimina a necessidade de altos investimentos de capital (CAPEX) em licenças e hardware, transformando-os em despesas operacionais (OPEX) previsíveis (Blog Praxio, 2021).

Além da otimização financeira, outros benefícios se destacam. O modelo introduz uma notável escalabilidade e flexibilidade, permitindo que as empresas aumentem ou diminuam facilmente a capacidade de uso do *software* para responder rapidamente às mudanças do mercado. Soma-se a isso a acessibilidade e mobilidade intrínsecas ao serviço, que, por ser acessado via internet, pode ser utilizado de qualquer lugar e em diferentes dispositivos — algo transformador para o setor de logística. Ao terceirizar a gestão da infraestrutura de TI, as empresas também ganham maior foco no *core business*, dedicando mais tempo e recursos à otimização de suas operações. Por fim, provedores de SaaS geralmente oferecem um nível de segurança e disponibilidade superior ao que uma PME poderia implementar por conta própria, garantido por Acordos de Nível de Serviço (SLAs) (Prolog App, 2024a; Blog Praxio, 2021).

Apesar das vantagens, o modelo também apresenta desafios, como a dependência de uma conexão estável com a internet e opções de personalização potencialmente mais limitadas. É justamente esse conjunto de benefícios, especialmente a redução de custos e a acessibilidade, que torna o modelo SaaS uma solução promissora para modernizar setores tradicionalmente analógicos, como o de transporte rodoviário.

2.2 TECNOLOGIA E INOVAÇÃO NO TRANSPORTE RODOVIÁRIO

O setor de transporte rodoviário de passageiros, apesar de sua importância socioeconômica, apresenta um ritmo lento na adoção de tecnologias digitais (SEST SENAT, 2021). A ausência de soluções tecnológicas integradas, especialmente no segmento alternativo, resulta em ineficiências como a falta de controle sobre os processos, a incapacidade de atender a picos de demanda e altos índices de reclamação de clientes (EnGeTec - Fatec Zona Leste, 2022). Essa carência é corroborada pela pesquisa de mercado conduzida para este trabalho (ver Apêndice B), que revelou que 100% dos gestores entrevistados ainda dependem de métodos como cadernos de anotações e mensagens de WhatsApp para gerenciar suas operações.

Plataformas SaaS surgem como uma solução promissora para democratizar o acesso a tecnologias avançadas neste setor. A análise de soluções comerciais existentes, como TOTVS, iTransport e Praxio, revela uma lacuna de mercado: as ferramentas ou são muito complexas e caras para PMEs, ou são focadas em nichos

específicos como o fretamento corporativo, não atendendo de forma integrada as necessidades do operador de transporte alternativo (TOTVS, 2025; iTransport, 2025; Praxio, 2025).

2.3 ARQUITETURA TECNOLÓGICA DA SOLUÇÃO PROPOSTA

2.3.1 Tecnologias do Backend

O *backend* é o alicerce da plataforma, responsável pela lógica de negócios, persistência de dados e segurança. A *stack* escolhida foi projetada para ser modular, robusta e escalável, utilizando tecnologias consolidadas no ecossistema TypeScript.

2.3.1.1 NestJS e Arquitetura Modular

O framework escolhido para o desenvolvimento do *backend* foi o NestJS. Trata-se de um *framework* Node.js progressivo, construído com e para o TypeScript, que utiliza uma arquitetura fortemente modular (NestJS, 2025). O pilar do NestJS é o conceito de Módulos, que organizam o código em blocos coesos e funcionais (e.g., um módulo para autenticação, outro para gestão de rotas). Cada módulo encapsula seus próprios *controllers*, *providers* (serviços) e pode importar ou exportar funcionalidades (NestJS Docs, 2025). Essa estrutura promove uma forte separação de responsabilidades, facilita a reutilização de código e a manutenção do sistema, sendo ideal para a plataforma proposta, onde funcionalidades complexas podem ser desenvolvidas como módulos independentes (Dev and Deliver, 2024).

2.3.1.2 TypeORM e Mapeamento Objeto-Relacional (ORM)

Para a camada de persistência de dados, a escolha foi o TypeORM. Um *Object-Relational Mapper* (ORM) é uma técnica que cria uma ponte entre o paradigma orientado a objetos da aplicação e o paradigma relacional dos bancos de dados, permitindo que os desenvolvedores manipulem o banco de dados através de objetos e classes, abstraindo a necessidade de escrever consultas SQL manualmente (LogRocket Blog, 2024). O TypeORM é um ORM maduro para o ecossistema TypeScript, que utiliza intensivamente decoradores para definir Entidades (classes que mapeiam para tabelas) de forma declarativa e fortemente tipada. Isso acelera o desenvolvimento e reduz a probabilidade de erros relacionados a dados (Alim, 2024).

2.3.1.3 Autenticação com JWT e Passport.js

A segurança da API é um requisito crítico. A estratégia de autenticação escolhida foi baseada em *JSON Web Tokens* (JWT), implementada com a biblioteca Passport.js. JWT é um padrão aberto (RFC 7519) para a criação de tokens de acesso

compactos e autossuficientes. Quando um cliente faz uma requisição a um recurso protegido, ele envia o JWT, e o servidor pode verificar a assinatura para autenticar o usuário sem precisar consultar um banco de dados de sessões, resultando em uma autenticação *stateless* ideal para APIs RESTful (Soshace, 2024). O Passport.js atua como um *middleware* de autenticação modular, e a estratégia *passport-jwt* é projetada especificamente para extrair e verificar a validade desses *tokens*, oferecendo uma solução robusta e padronizada para proteger as rotas da API (Passport.js, 2025).

2.3.2 Tecnologias do Frontend

A interface do usuário (*frontend*) é o principal ponto de contato com os gestores e passageiros. Sua arquitetura foi projetada com foco em performance e robustez, utilizando um ecossistema moderno baseado em React.

2.3.2.1 Next.js, SSR e App Router

O framework escolhido para o *frontend* é o Next.js. Uma de suas principais características é o suporte nativo à Renderização no Servidor (*Server-Side Rendering - SSR*), na qual a página HTML é gerada no servidor a cada requisição. Isso melhora o tempo de carregamento inicial da página e a otimização para motores de busca (SEO) (bloodturtle, 2025). Com a introdução do *App Router*, o Next.js adotou por padrão o uso de *React Server Components*, que permitem que a busca de dados e a renderização de componentes não interativos ocorram exclusivamente no servidor. Isso resulta em uma redução significativa da quantidade de JavaScript enviada ao cliente e otimiza a performance geral da aplicação (Next.js Docs, 2025).

2.3.2.2 React, TypeScript e Ecossistema de UI

A base da interface é o React, uma biblioteca JavaScript para a construção de interfaces baseada em um modelo de componentes reutilizáveis. Para garantir a robustez e a manutenibilidade, o React é utilizado com o TypeScript, um superconjunto do JavaScript que adiciona tipagem estática. O TypeScript permite a detecção de erros em tempo de compilação e serve como uma forma de documentação viva, tornando o código mais fácil de entender e facilitando a colaboração (DhiWise, 2024).

Para a estilização, foi adotado o Tailwind CSS, um framework *utility-first* que acelera o desenvolvimento e garante consistência visual (Ignatovich.dm, 2025). Complementando-o, a biblioteca de componentes *shadcn/ui* foi utilizada. Diferente de bibliotecas tradicionais, seus componentes são copiados para o código-fonte do projeto, dando ao desenvolvedor controle total sobre o código e permitindo personalizações profundas sem *vendor lock-in* (shadcn/ui, 2025).

3 REQUISITOS DO SISTEMA

A concepção do sistema ViaBus partiu de uma necessidade concreta, identificada na pesquisa de mercado (Apêndice B) realizada com gestores de empresas de transporte de passageiros. A pesquisa revelou um cenário operacional amplamente manual, onde ferramentas como *"cadernos de anotações"* e *"mensagens de WhatsApp"* são o padrão para tarefas críticas como a venda de passagens e a organização de listas de embarque. Essa dependência de métodos analógicos resulta nos principais desafios apontados pelos gestores: a dificuldade de *"controlar os assentos já vendidos"*, a *"perda de tempo organizando listas"* e a *"comunicação com os motoristas"*.

Diante desse diagnóstico, os requisitos do ViaBus foram definidos para atacar diretamente essas dores. Eles representam a digitalização e a centralização de um processo hoje fragmentado e suscetível a erros. Este capítulo detalha esses requisitos em duas perspectivas: os requisitos funcionais, que descrevem o que o sistema deve ser capaz de fazer para o usuário, e os requisitos não funcionais, que definem os atributos de qualidade e as condições sob as quais o sistema deve operar.

3.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais foram agrupados em categorias que espelham o fluxo de trabalho de uma empresa de transporte. Eles traduzem as necessidades operacionais em funcionalidades concretas, conforme detalhado na Tabela 1.

3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais definem os critérios de qualidade que garantem uma boa experiência de uso e a confiabilidade do sistema. Eles são apresentados na Tabela 2.

Tabela 1 – Requisitos funcionais do sistema ViaBus

ID	Descrição da Funcionalidade
Organização da Empresa e Recursos	
RF01	O gestor deve poder cadastrar e gerenciar os dados da sua frota de veículos.
RF02	O gestor deve poder cadastrar, consultar e atualizar as informações de seus motoristas, como dados de contato e validade da CNH.
RF03	O sistema deve permitir o cadastro de todos os pontos de parada (embarque/desembarque) utilizados pela empresa, incluindo sua localização.
RF04	O gestor deve poder criar e organizar as rotas (itinerários) da empresa, definindo a sequência de paradas e os horários de operação.
Operação e Venda de Passagens	
RF05	O sistema deve permitir o agendamento de viagens futuras, associando uma rota, um veículo e um motorista.
RF06	Um vendedor ou gestor deve poder vender passagens de forma rápida e guiada, selecionando a rota e a data da viagem.
RF07	Ao vender uma passagem, o sistema deve registrar as informações essenciais do passageiro.
RF08	O sistema não deve permitir, em hipótese alguma, a venda de uma passagem para um assento que já está ocupado em uma viagem (evitar overbooking).
RF09	O sistema deve gerar uma lista de passageiros de fácil consulta para cada viagem, que possa ser usada no momento do embarque.
Controle e Análise	
RF10	O gestor deve poder visualizar rapidamente a taxa de ocupação de cada viagem para tomar decisões.
RF11	O sistema deve fornecer relatórios simples de vendas, mostrando o faturamento por rota ou por período.

Tabela 2 – Requisitos não funcionais do sistema ViaBus

ID	Descrição da Funcionalidade
Usabilidade e Acesso	
RNF01	A interface do sistema deve ser clara, organizada e intuitiva, exigindo o mínimo de treinamento para um novo usuário.
RNF02	O sistema deve ser fácil de usar tanto em um computador no escritório quanto em um celular ou tablet na rodoviária.
Confiabilidade e Desempenho	
RNF03	O sistema deve ser rápido, com telas e informações carregando em poucos segundos.
RNF04	A plataforma deve estar disponível para uso a maior parte do tempo, sem quedas ou instabilidades frequentes.
Segurança	
RNF05	O acesso ao sistema deve ser protegido por login e senha.
RNF06	Os dados de uma empresa de transporte não podem, sob nenhuma circunstância, ser acessados pelos usuários de outra empresa.

4 TECNOLOGIAS ENVOLVIDAS

A construção do protótipo ViaBus foi fundamentada em um conjunto de tecnologias de mercado, selecionadas a partir de critérios objetivos como adequação aos requisitos do projeto, ecossistema e características técnicas específicas. Este capítulo apresenta as principais ferramentas, bibliotecas e frameworks utilizados, organizados em tabelas por categoria para facilitar a consulta e a compreensão da arquitetura da solução. Cada tabela detalha a aplicação específica da tecnologia no projeto e a justificativa técnica para sua escolha. Para consultar as versões exatas de cada dependência utilizada, veja o Apêndice D, que lista os repositórios do código-fonte do projeto.

4.1 TECNOLOGIAS DO BACKEND

As tecnologias do *backend* formam o alicerce da aplicação, responsáveis pela lógica de negócio, segurança e persistência dos dados. As principais escolhas estão detalhadas na Tabela 3.

Tabela 3 – Tecnologias utilizadas no Backend

Tecnologia	Aplicação no Projeto ViaBus	Justificativa da Escolha
NestJS	Framework principal para a construção da API RESTful, estruturando os módulos de negócio (rotas, motoristas, passagens, etc.).	A arquitetura modular e o sistema de injeção de dependência nativo facilitam a organização e a manutenção de um projeto com múltiplos domínios.
TypeORM	Mapeamento das classes do domínio para as tabelas do banco de dados PostgreSQL, abstraindo a escrita de consultas SQL.	Integração madura com o ecossistema TypeScript e suporte a <i>decorators</i> para a definição de entidades, o que aumenta a produtividade e a clareza do código.
Passport.js com JWT	Implementação da estratégia de autenticação e autorização, protegendo as rotas da API e gerenciando as sessões dos usuários.	Padrão de mercado para autenticação <i>stateless</i> em APIs. A modularidade do Passport.js permite uma implementação desacoplada e segura.

4.2 TECNOLOGIAS DO FRONTEND

O *frontend* é a camada de apresentação do sistema, com a qual o usuário interage diretamente. As tecnologias foram escolhidas com foco em performance e experiência de desenvolvimento, conforme a Tabela 4.

Tabela 4 – Tecnologias utilizadas no Frontend

Tecnologia	Aplicação no Projeto ViaBus	Justificativa da Escolha
Next.js	Framework principal para a construção da interface de usuário (UI), incluindo o painel de gestão e os fluxos de venda de passagens.	Suporte nativo a Server-Side Rendering (SSR) e Server Components, estratégias que melhoraram o tempo de carregamento inicial e a performance percebida pelo usuário.
React	Biblioteca base para a criação de toda a interface em um modelo de componentes reutilizáveis (botões, formulários, tabelas, etc.).	Ecossistema consolidado e vasta disponibilidade de bibliotecas, o que viabilizou a rápida prototipação da interface.
TypeScript	Utilizado em todo o desenvolvimento (backend e frontend) para adicionar tipagem estática ao JavaScript.	Aumenta a manutenibilidade do código e reduz a ocorrência de erros em tempo de execução, servindo como uma forma de documentação para a estrutura de dados.
Tailwind CSS	Framework CSS utilizado para a estilização de todos os componentes da interface.	A abordagem <i>utility-first</i> acelera o desenvolvimento da UI e facilita a manutenção de um design system consistente em toda a aplicação.

4.3 INFRAESTRUTURA E FERRAMENTAS DE SUPORTE

Este conjunto de ferramentas dá suporte ao desenvolvimento, à persistência dos dados e à implantação do sistema, garantindo consistência e qualidade ao longo do ciclo de vida do projeto (Tabela 5).

Tabela 5 – Infraestrutura e Ferramentas de Suporte

Tecnologia	Aplicação no Projeto ViaBus	Justificativa da Escolha
PostgreSQL	Sistema de Gerenciamento de Banco de Dados (SGBD) relacional para a persistência de todos os dados da aplicação.	SGBD de código aberto com funcionalidades avançadas, como suporte a dados geoespaciais (PostGIS), e alta compatibilidade com o TypeORM.
Docker	Containerização do backend, do frontend e do banco de dados para os ambientes de desenvolvimento e produção.	Garante a paridade entre os ambientes, simplifica a configuração do projeto e facilita a implantação em qualquer provedor de nuvem.
ESLint e Prettier	Ferramentas para análise estática e formatação automática do código-fonte.	Asseguram a padronização e a qualidade do código, independentemente do desenvolvedor, facilitando a leitura e a manutenção futura.

5 ARQUITETURA E MODELAGEM DO SISTEMA

5.1 ARQUITETURA DA SOLUÇÃO

Esta seção descreve a organização estrutural do sistema ViaBus, que provê funções de gestão de transporte para empresas: cadastro de empresas, usuários, motoristas, veículos, rotas, paradas, viagens e bilhetes. A solução adota uma arquitetura web em camadas, com separação clara entre apresentação (*frontend*), lógica de negócio e APIs (*backend*) e persistência (banco de dados).

No *frontend*, utiliza-se Next.js 15 com roteamento via *App Router*, estado de sessão via NextAuth (estratégia JWT) e componentes React tipados (TypeScript). O *frontend* consome APIs REST autenticadas, mantém contexto de usuário/empresa e incorpora mapas e edição geográfica com Leaflet/React-Leaflet para operações sobre rotas e paradas.

O *backend* é implementado com NestJS 11, estruturado por módulos de domínio (*auth, users, companies, drivers, vehicles, stops, routes, trips, tickets, etc.*). As regras de negócio são expostas por controladores REST, protegidos por autenticação JWT e autorização baseada em papéis. A multiempresa é tratada por *companyId* e por um interceptor que injeta o contexto de empresa a partir do token. A persistência usa TypeORM com PostgreSQL.

A Figura 1 ilustra essa arquitetura em camadas.

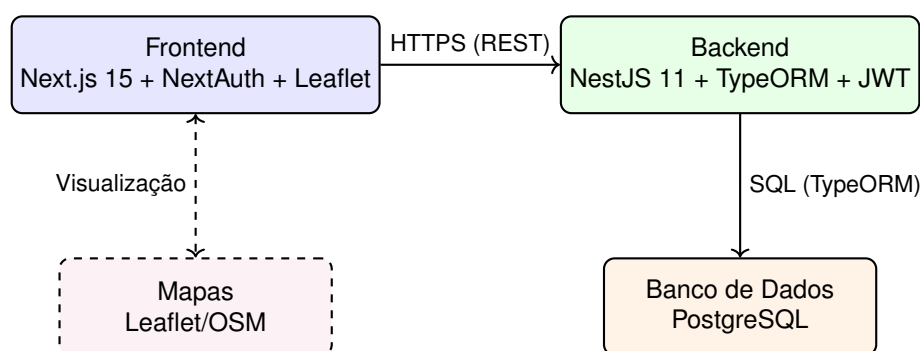


Figura 1 – Visão de alto nível da solução.

5.2 MODELAGEM DO BANCO DE DADOS

O modelo de dados foi projetado para refletir as agregações de negócio, garantindo a coesão e a integridade das entidades do domínio. A Tabela 6 sumariza as principais entidades e seus papéis. Para facilitar a análise, a modelagem será apresentada através de diagramas UML fracionados por agregado. A visão completa e detalhada de todos os relacionamentos do domínio está disponível para consulta no Apêndice C.

Entidade	Descrição resumida
companies	Empresa (razão social, nome fantasia, CNPJ, slug, e-mail, telefone, logo)
users	Usuário (nome, e-mail, senha, papel, status, telefone, foto, company_id)
drivers	Motorista (nome, CPF, CNH, categoria, validade, telefone, e-mail, data nascimento, contratação, status, contato emergência, endereço, observações, company_id)
vehicles	Veículo (placa, modelo, marca, ano, capacidade, categoria, conforto, tipo ônibus, data aquisição, hodômetro, manutenção, status, observações, company_id)
addresses	Endereço (CEP, logradouro, número, complemento, bairro, cidade, estado, longitude, latitude)
stops	Parada (nome, address_id, ativa, acessibilidade, abrigo, company_id)
routes	Rota (nome, descrição, ativa, duração estimada, distância, company_id)
route_stops	Associação rota-parada com ordem e horário de partida opcional
route_schedules	Agenda por dia da semana para a rota (0-6, domingo-sábado)
trips	Viagem (rota, horários partida/chegada, status, preço base, assentos total/disponível, auto-gerada, observações, company_id)
trip_vehicles	Associação viagem-veículo com motoristas primário/secundário
tickets	Bilhete (passageiro, documento, telefone, e-mail, assento, preço, status, pontos embarque/desembarque com coordenadas, observações, company_id)

Tabela 6 – Principais entidades do domínio.

5.2.1 Diagramas UML Fracionados por Agregado

Para facilitar a análise do domínio e a compreensão das decisões de modelagem, o esquema de dados foi decomposto em agregados lógicos. Um agregado é um cluster de objetos de domínio que podem ser tratados como uma única unidade, garantindo a consistência das regras de negócio. A seguir, cada agregado principal é detalhado, justificando-se as principais decisões de modelagem.

Agregado Organizacional: Empresas e Usuários

O pilar da arquitetura *multi-tenant* do sistema reside neste agregado. A entidade *Company* é a raiz de todos os dados operacionais. A decisão de incluir a chave estrangeira *companyId* na entidade *User* estabelece o vínculo fundamental que garante que cada usuário pertença inequivocamente a uma única empresa. Este design é a base para o filtro de isolamento de dados que será aplicado em todas as consultas subsequentes, conforme detalhado na seção de arquitetura do *backend*. Figura 2 ilustra este agregado.

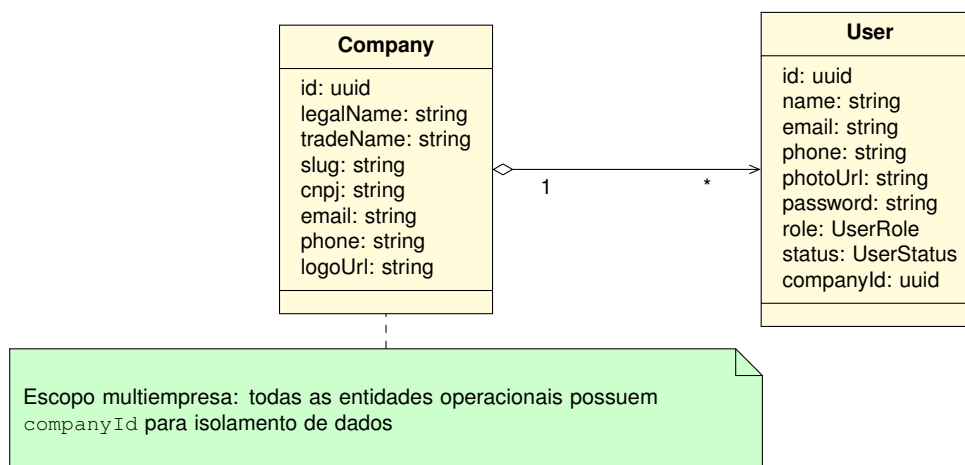


Figura 2 – Agregado organizacional - estrutura multiempresa.

Agregado de Configuração: Rotas e Paradas

Este agregado (Figura 3) modela a estrutura logística fundamental de uma empresa de transporte. A decisão mais crítica aqui foi a criação da entidade associativa `RouteStop`. Em vez de uma relação simples, a `RouteStop` foi necessária para resolver o relacionamento N:N entre `Route` e `Stop`, permitindo que uma mesma parada (ex: "Rodoviária de Picos") possa pertencer a múltiplas rotas. Mais importante, esta entidade armazena o atributo `order`, que é a regra de negócio central para definir a sequência do trajeto, transformando um conjunto de pontos em um itinerário ordenado. Adicionalmente, a entidade `RouteSchedule` foi criada para desacoplar a definição da rota de sua frequência operacional, permitindo que uma mesma rota tenha horários diferentes dependendo do dia da semana.

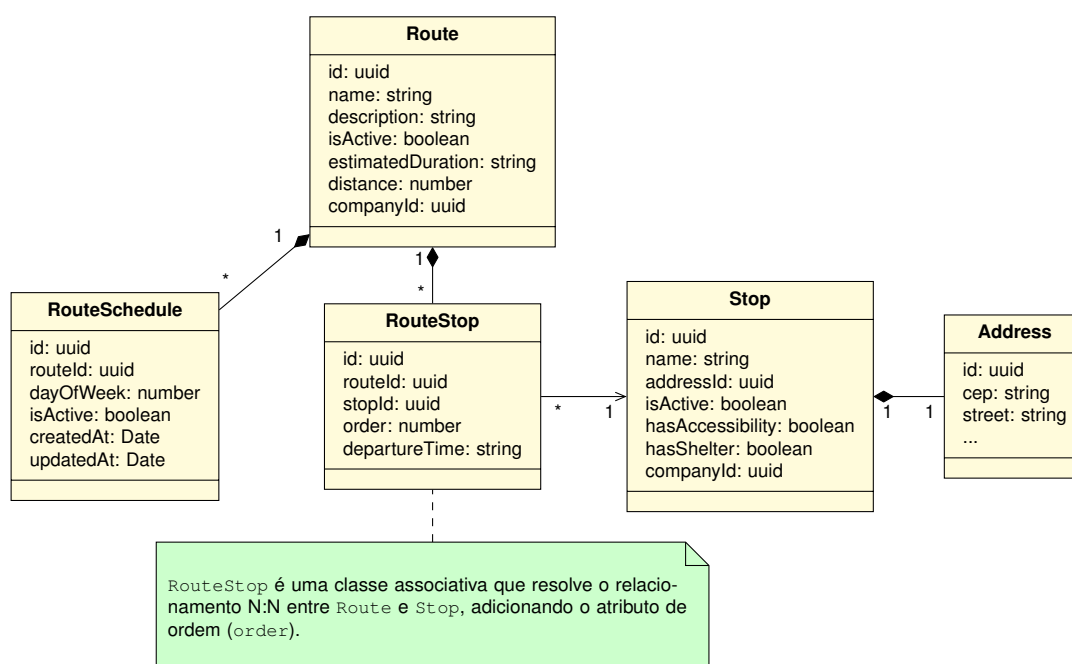


Figura 3 – Agregado de configuração de rotas, paradas e horários.

Agregado de Recursos Operacionais: Veículos e Motoristas

Este agregado (Figura 4) representa os ativos físicos e humanos da empresa. As entidades *Vehicle* e *Driver* são recursos independentes que são dinamicamente alocados às viagens. A modelagem através da entidade associativa *TripVehicle* foi uma decisão deliberada para conferir flexibilidade operacional. Em vez de vincular um motorista fixo a um veículo, a *TripVehicle* permite que a alocação seja feita por viagem, refletindo a realidade das operações de transporte. A inclusão de campos para motorista primário e secundário (*primaryDriverId*, *secondaryDriverId*) nesta entidade já prevê a complexidade de viagens longas ou que necessitam de revezamento, demonstrando a escalabilidade do modelo.

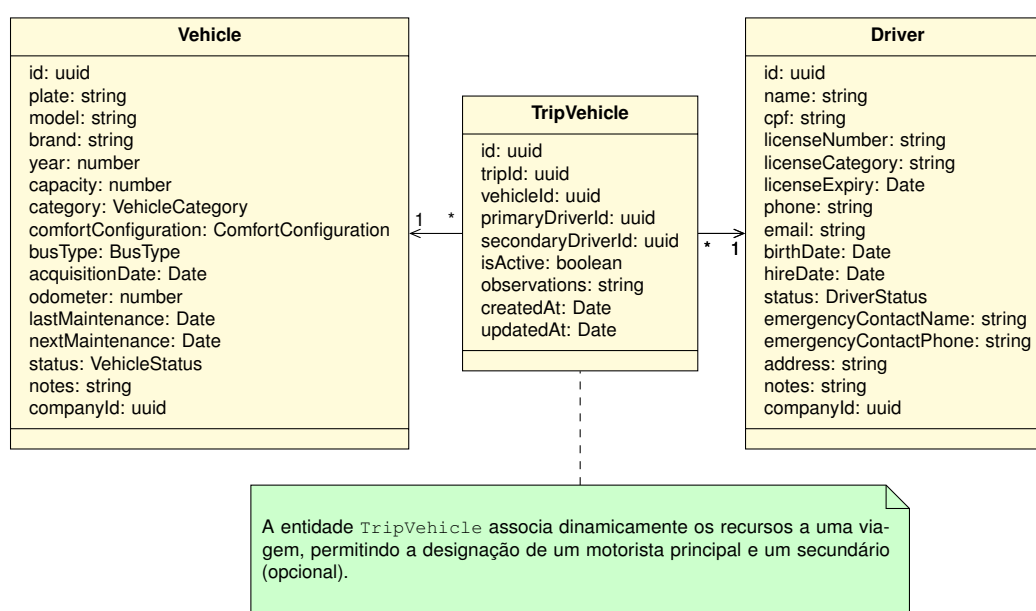


Figura 4 – Agregado de recursos operacionais - frota e motoristas.

Agregado de Operação e Vendas: Viagens e Bilhetes

Este é o agregado central que conecta a configuração (Rotas) com os recursos (Veículos/Motoristas) e a atividade comercial (Bilhetes), ilustrado na Figura 5. A entidade *Trip* é a instância de uma *Route* em uma data e horário específicos. Uma decisão de modelagem importante na entidade *Ticket* foi a inclusão de campos para registrar os pontos de embarque e desembarque (*boardingStopId*, *landingStopId*), permitindo a venda de bilhetes para trechos seccionados da viagem. Além disso, a flexibilidade foi aumentada ao permitir que os pontos de embarque/desembarque possam ser tanto uma parada pré-cadastrada quanto uma localização customizada (com descrição e coordenadas), atendendo à demanda do transporte alternativo onde pontos de encontro podem ser flexíveis.

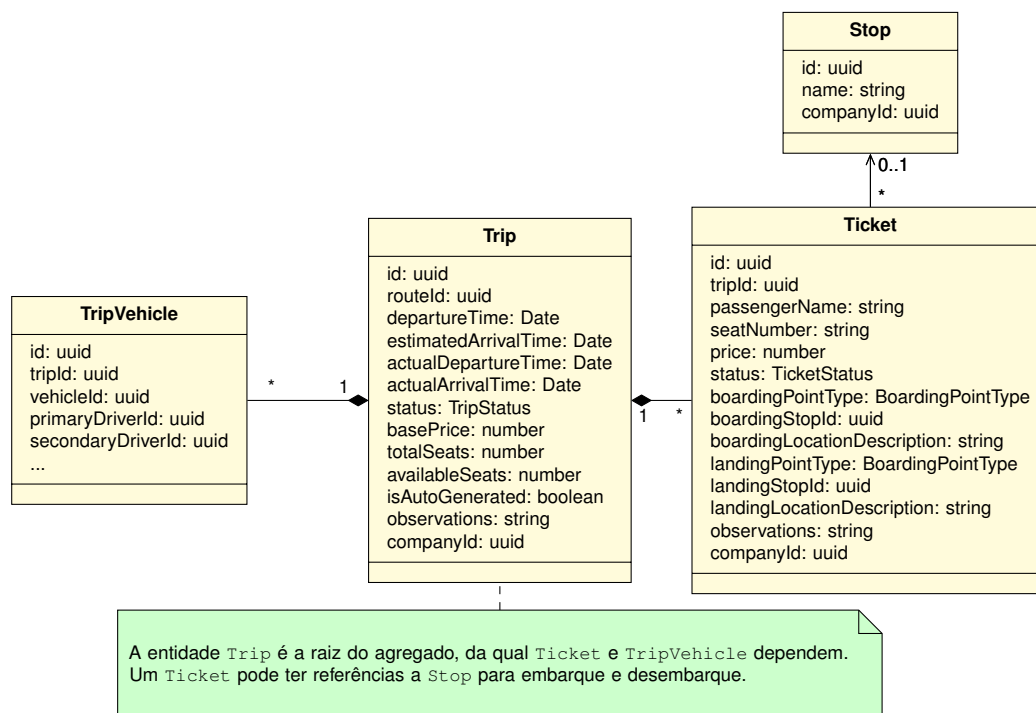


Figura 5 – Agregado de operação - viagens, recursos e vendas.

5.3 ARQUITETURA DO BACKEND

O *backend* implementa uma arquitetura modular baseada no *framework* NestJS 11, seguindo os princípios de *Separation of Concerns* e *Dependency Injection*. A aplicação é estruturada em 10 módulos de domínio, com camadas bem definidas de segurança, validação e persistência.

5.3.1 Organização Modular

O sistema está organizado em três grupos funcionais de módulos, conforme ilustrado na Figura 6.

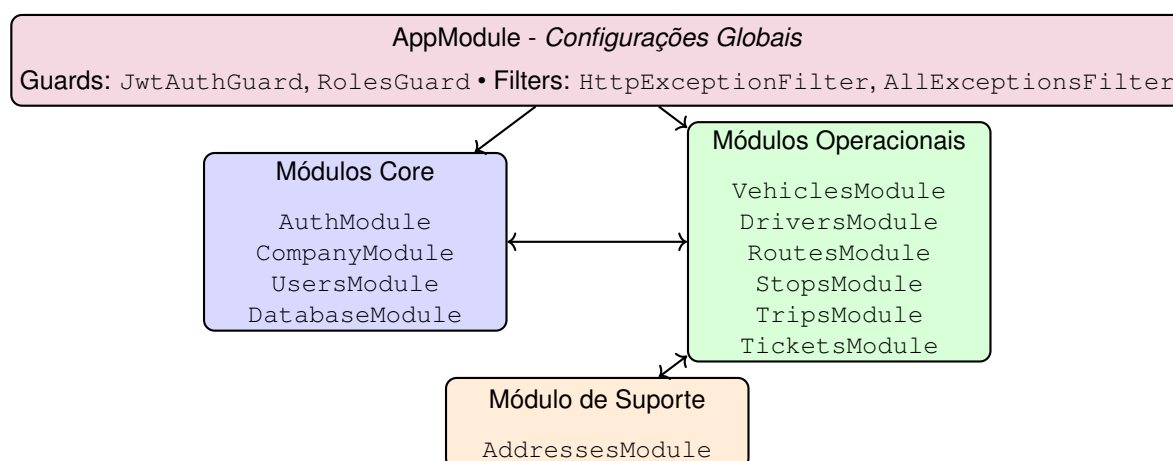


Figura 6 – Organização modular do backend ViaBus.

A organização modular agrupa os componentes por responsabilidade funcional. Os Módulos Core gerenciam autenticação, usuários, empresas e configuração de banco de dados. Os Módulos Operacionais implementam as regras de negócio específicas do domínio de transporte. O Módulo de Suporte fornece funcionalidades auxiliares como geocodificação. O `AppModule` centraliza as configurações globais de segurança e tratamento de erros, aplicadas a todos os módulos.

5.3.2 Pipeline de Processamento

Cada requisição HTTP passa por um *pipeline* estruturado de *middleware*, *guards*, *interceptors* e *filters*, garantindo segurança e consistência. A Figura 7 ilustra este pipeline.

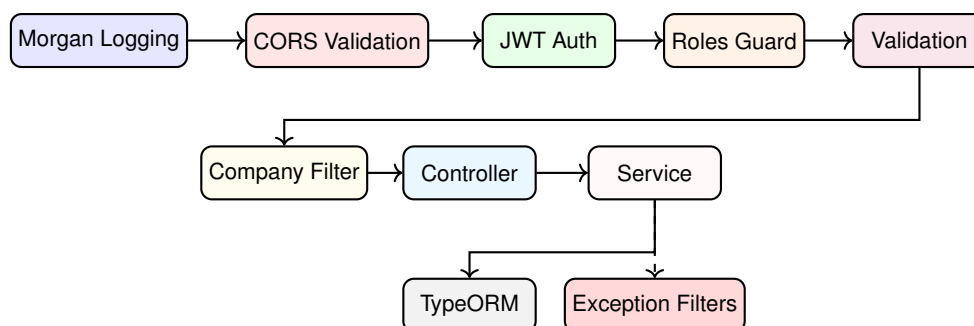


Figura 7 – Pipeline de processamento de requisições.

O *pipeline* garante que cada requisição passe por validações sequenciais obrigatórias. Inicialmente, o Morgan registra *logs* e o CORS valida origens permitidas. A autenticação via `JwtAuthGuard` verifica *tokens* válidos, seguida da autorização por `RolesGuard` que confirma permissões do usuário. O `ValidationPipe` valida DTOs de entrada. O `CompanyFilterInterceptor` injeta o contexto multiempresa antes do processamento pelo `Controller` e `Service`. Finalmente, o `TypeORM` persiste dados e os `Exception Filters` tratam erros de forma padronizada.

5.3.3 Fluxo de Autenticação

O sistema utiliza autenticação baseada em JWT com suporte a *multi-tenancy*. A Figura 8 detalha o processo completo.

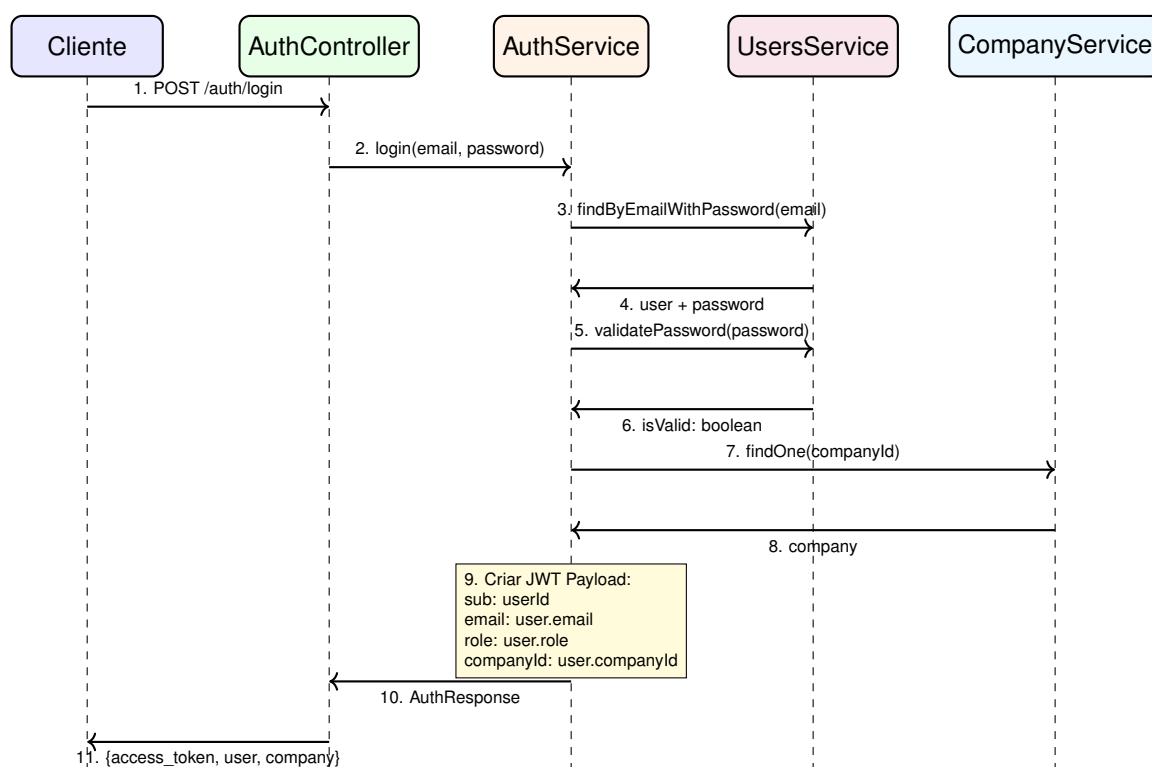


Figura 8 – Diagrama de sequência - fluxo de autenticação.

O fluxo de autenticação implementa um processo seguro e eficiente em 11 etapas. O cliente envia credenciais via POST para o `AuthController`, que delega ao `AuthService` a validação. O `UsersService` busca o usuário por email (incluindo senha hash), valida a senha fornecida e retorna o resultado. Confirmada a autenticação, o `AuthService` consulta os dados da empresa via `CompanyService` e cria o *payload* JWT contendo informações do usuário e contexto empresarial. A resposta final inclui o *token* de acesso, dados do usuário e informações da empresa, permitindo que o *frontend* mantenha o contexto de sessão multiempresa.

5.3.4 Arquitetura Multi-tenant

O sistema implementa isolamento de dados por empresa através de uma arquitetura *multi-tenant* baseada em filtros automáticos. A solução utiliza três componentes principais que trabalham em conjunto para garantir separação completa entre inquilinos:

- `BaseCompanyService<T>`: Classe abstrata que implementa operações CRUD com filtro automático por `companyId`
- `CompanyFilterInterceptor`: Interceptor que extrai o `companyId` do token JWT e o injeta no contexto da requisição
- `@CurrentUser`: Decorator que facilita o acesso aos dados do usuário autenticado

A implementação garante que todas as operações de banco de dados incluam automaticamente a cláusula `WHERE companyId = ?`, eliminando a possibilidade de vazamento de dados entre empresas. O padrão é aplicado consistentemente em todos os módulos operacionais (usuários, motoristas, veículos, rotas, paradas, viagens e passagens). A Figura 9 ilustra esta arquitetura.

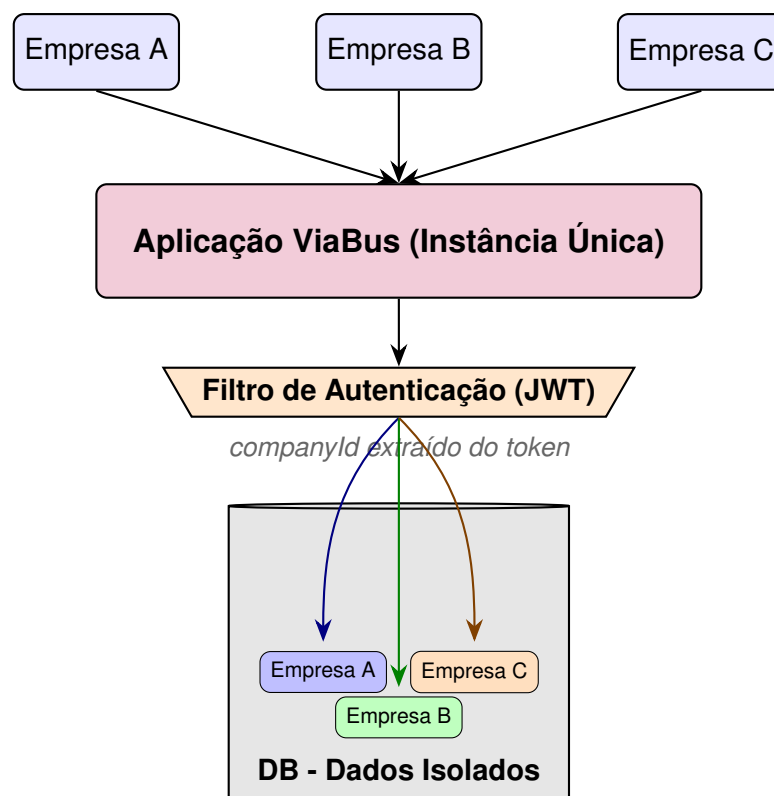


Figura 9 – Arquitetura multi-tenant: uma única aplicação com dados logicamente isolados por empresa no banco de dados.

5.4 ARQUITETURA DO FRONTEND

O *frontend* implementa uma arquitetura moderna baseada em Next.js 15 (React 19) com seu *App Router*, utilizando TypeScript para garantir uma tipagem estática completa. A aplicação foi estruturada seguindo padrões de arquitetura limpa e uma organização orientada a domínios de negócio, visando a manutenibilidade e a escalabilidade da interface.

5.4.1 Stack Tecnológico e Organização

Para garantir *performance*, uma boa experiência de usuário e manutenibilidade, o *frontend* utiliza um conjunto integrado de tecnologias. A base da aplicação é o Next.js 15, que, junto ao React 19, oferece uma plataforma sólida para a renderização da interface. A construção da UI segue um *design system* customizado, empregando componentes primitivos do Radix UI para assegurar a acessibilidade e o Tailwind CSS

para uma estilização *utility-first* ágil e consistente. A gestão de formulários é responsabilidade do React Hook Form combinado com o Zod para validação tipada de dados. A autenticação é gerenciada pelo NextAuth v4, enquanto as funcionalidades geográficas são implementadas com Leaflet. O estado global da aplicação, por sua vez, é controlado de forma eficiente pela Context API nativa do React.

5.4.2 Arquitetura de Roteamento

A aplicação utiliza o *App Router* do Next.js 15, que permite um roteamento dinâmico e baseado em arquivos. A estrutura de rotas foi projetada para suportar a natureza multiempresa do sistema, seguindo o padrão `/dashboard/[company]/recurso`. Essa abordagem garante o isolamento dos dados no nível da URL, onde cada empresa acessa seus próprios recursos, como `/motoristas` ou `/rotas`, dentro de seu próprio escopo. O sistema diferencia entre páginas públicas, como `/login` e `/criar-empresa`, e as rotas privadas do *dashboard*. A proteção de rotas é garantida por um componente de ordem superior (*wrapper*) chamado `ProtectedPage`, que valida a autenticação e as permissões do usuário antes de renderizar a página solicitada.

5.4.3 Gerenciamento de Estado e Contextos

O gerenciamento do estado global da aplicação é implementado através de uma hierarquia de *Contextos* React. Essa abordagem permite um compartilhamento de dados eficiente e desacoplado entre os componentes. No topo da hierarquia, o `SessionProvider` gerencia a sessão do NextAuth e os *tokens* JWT. Abaixo dele, o `ThemeProvider` controla o tema visual da aplicação (claro/escuro). Em seguida, o `AuthProvider` expõe os dados do usuário autenticado e da empresa associada. O `CompanyProvider` mantém o estado específico da empresa que está sendo visualizada no *dashboard*, enquanto o `SidebarProvider`, na camada mais interna, gerencia estados de UI, como a visibilidade do menu lateral.

5.4.4 Sistema de Autenticação e Comunicação

A comunicação com o *backend* é centralizada para garantir consistência e segurança. A autenticação integra o NextAuth com a API NestJS através de *tokens* JWT, e todas as requisições HTTP são gerenciadas por um serviço centralizado, o `api.service.ts`. Este serviço encapsula responsabilidades críticas: ele anexa automaticamente os *Bearer tokens* de autenticação em todas as chamadas a rotas protegidas, trata respostas de erro de forma padronizada (redirecionando para a página de *login* em caso de erro 401, por exemplo), processa as respostas da API para

extrair os dados e valida a sessão do usuário antes de cada requisição, otimizando a comunicação com o servidor.

5.4.5 Arquitetura de Componentes

O sistema de componentes foi projetado em camadas, seguindo uma lógica de composição que vai do mais simples ao mais complexo. Na base, estão os Primitivos de UI do Radix UI, que fornecem comportamento e acessibilidade sem estilização. Sobre eles, foi construído um *Design System* de componentes reutilizáveis e tipados (localizados em `components/ui/`), que definem a aparência visual da aplicação. Estes são, então, utilizados para compor os Componentes de Domínio, que são específicos de cada funcionalidade, como formulários de cadastro ou tabelas de dados. Por fim, as Páginas são montadas através da composição desses componentes, organizadas dentro de *Layouts* reutilizáveis que fornecem a estrutura de navegação e os provedores de contexto necessários.

5.4.6 Tipagem e Validação

O TypeScript é utilizado de forma extensiva para garantir a integridade dos dados em toda a aplicação. Um diretório central de tipos (*Types Directory*) contém as definições para cada entidade de negócio (`User`, `Company`, etc.), servindo como uma fonte única da verdade para a estrutura de dados. Essa tipagem se estende às respostas da API, garantindo que os dados recebidos do *backend* correspondam ao esperado. No lado do cliente, a validação de formulários é realizada com *Schemas Zod*, que asseguram a consistência dos dados antes do envio. Essa abordagem cria uma arquitetura com tipagem estática de ponta a ponta (*end-to-end*), desde o banco de dados até a interface do usuário. A Figura 10 ilustra esta arquitetura.

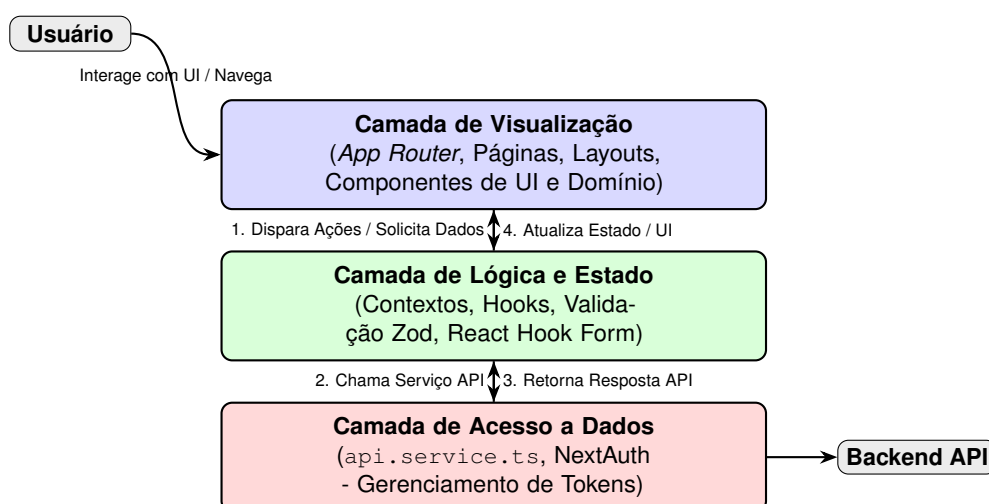


Figura 10 – Arquitetura em camadas do frontend e fluxo de dados principal.

5.5 ESTRATÉGIA DE IMPLANTAÇÃO

A implantação do sistema ViaBus foi planejada em uma arquitetura de nuvem, utilizando um conjunto de serviços da Google Cloud Platform (GCP) para automação e escalabilidade. A estratégia adotada consistiu em quatro etapas principais: containerização da aplicação, implantação dos serviços em ambiente *serverless*, gerenciamento do banco de dados e automação da entrega (CI/CD).

Para garantir a portabilidade entre os ambientes, a aplicação foi containerizada com Docker. Foram criados arquivos `Dockerfile` específicos para o *backend* (NestJS) e o *frontend* (Next.js), utilizando a abordagem de *multi-stage builds* para gerar imagens otimizadas e seguras. A orquestração do ambiente de desenvolvimento local foi realizada com o Docker Compose.

Os contêineres da aplicação foram implantados na nuvem como dois serviços independentes no Google Cloud Run, uma plataforma de computação *serverless*. Essa configuração permite que o *frontend* e o *backend* escalem de forma autônoma com base na demanda de requisições. A comunicação externa é garantida via HTTPS, com certificados TLS/SSL gerenciados automaticamente pela plataforma.

Para a persistência dos dados, foi utilizada uma instância gerenciada do PostgreSQL no serviço Google Cloud SQL. A comunicação entre a API executada no Cloud Run e o banco de dados ocorre em uma rede segura e privada, com criptografia dos dados em trânsito e em repouso. A responsabilidade por tarefas de manutenção, como backups e atualizações, é delegada ao próprio serviço da GCP.

Por fim, o ciclo de vida da aplicação foi automatizado com práticas de Integração e Entrega Contínuas (CI/CD). Foram estruturados ambientes isolados de desenvolvimento e produção na GCP. O gerenciamento de informações sensíveis, como chaves de API e credenciais de banco de dados, foi centralizado no Google Secret Manager. Um *pipeline* no Google Cloud Build foi configurado para automatizar o processo de construção das imagens Docker e a implantação de novas versões no Cloud Run a cada atualização no repositório principal do Git.

6 APRESENTAÇÃO DO PROTÓTIPO FUNCIONAL

Este capítulo apresenta a materialização do sistema ViaBus através de seu protótipo funcional de *frontend*. O objetivo aqui não é apenas exibir as telas desenvolvidas, mas demonstrar como as funcionalidades implementadas atendem diretamente aos Requisitos Funcionais (RF) definidos no Capítulo 3, solucionando os problemas operacionais identificados na pesquisa de mercado.

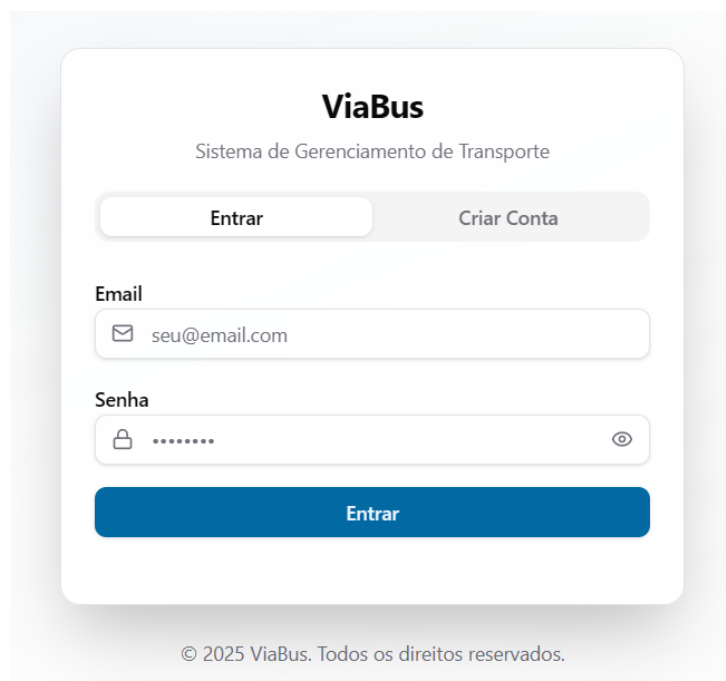
A interface foi desenvolvida seguindo os princípios de usabilidade e *design* responsivo (RNF01, RNF02), garantindo uma experiência de usuário clara e consistente em diferentes dispositivos. As seções a seguir estão organizadas de acordo com o fluxo de trabalho de um gestor: o *onboarding* inicial, o gerenciamento dos recursos da empresa e, por fim, a execução das operações diárias.

6.1 ONBOARDING E VISÃO GERAL DO SISTEMA

A primeira interação do usuário com o sistema envolve a autenticação e o acesso ao painel de controle, que fornece uma visão geral das operações.

6.1.0.1 Autenticação e Criação de Empresa

Para garantir a segurança e o isolamento dos dados (RNF05, RNF06), o sistema implementa um fluxo de autenticação via e-mail e senha, gerenciado pelo NextAuth, conforme visto na Figura 11. No primeiro acesso, o usuário é guiado por um formulário para cadastrar os dados de sua empresa (Figura 12), estabelecendo o ambiente de trabalho multi-tenant.



The login screen for ViaBus features a white card on a light gray background. At the top, the 'ViaBus' logo is centered, followed by the subtitle 'Sistema de Gerenciamento de Transporte'. Below this are two buttons: 'Entrar' (highlighted in blue) and 'Criar Conta' (gray). The 'Email' field contains 'seu@email.com' with an envelope icon. The 'Senha' field is masked with dots and includes a lock icon and a toggle eye icon. A large blue 'Entrar' button is at the bottom of the card. The footer shows '© 2025 ViaBus. Todos os direitos reservados.'

ViaBus
Sistema de Gerenciamento de Transporte

[Entrar](#) [Criar Conta](#)

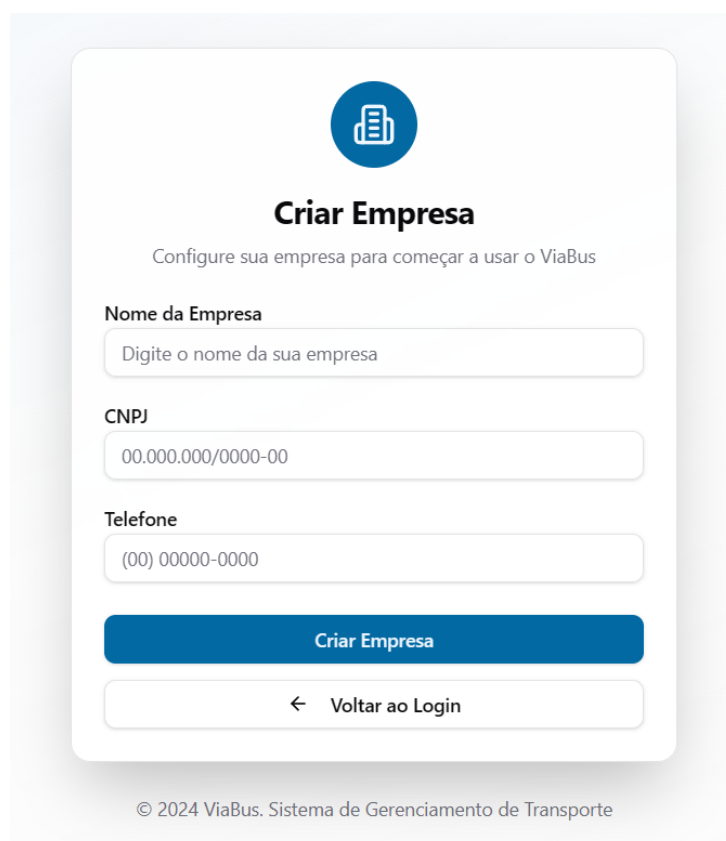
Email
seu@email.com

Senha
.....

[Entrar](#)

© 2025 ViaBus. Todos os direitos reservados.

Figura 11 – Tela de login do sistema ViaBus.



The company creation screen features a white card on a light gray background. At the top is a blue circular icon with a white building. Below it is the title 'Criar Empresa' and the subtitle 'Configure sua empresa para começar a usar o ViaBus'. The form includes three fields: 'Nome da Empresa' with placeholder 'Digite o nome da sua empresa', 'CNPJ' with placeholder '00.000.000/0000-00', and 'Telefone' with placeholder '(00) 00000-0000'. At the bottom are two buttons: a blue 'Criar Empresa' button and a white '← Voltar ao Login' button. The footer shows '© 2024 ViaBus. Sistema de Gerenciamento de Transporte'.



Criar Empresa
Configure sua empresa para começar a usar o ViaBus

Nome da Empresa
Digite o nome da sua empresa

CNPJ
00.000.000/0000-00

Telefone
(00) 00000-0000

[Criar Empresa](#)

[← Voltar ao Login](#)

© 2024 ViaBus. Sistema de Gerenciamento de Transporte

Figura 12 – Formulário de criação de empresa.

6.1.0.2 Painel de Controle (Dashboard)

Após o login, o gestor acessa o painel de controle (*dashboard*), que foi concebido para atender ao requisito de RF10 (visualizar rapidamente a ocupação e outras métricas). A interface, apresentada na Figura 13, serve como uma central de comando visual para o gestor.

Nesta fase do protótipo, o *layout* apresenta uma representação visual dos principais indicadores da operação. O *design* foi projetado para, em futuras iterações, exibir cartões com dados consolidados e atualizados em tempo real, como o número de viagens, passagens vendidas, e o status de veículos e motoristas. O objetivo destes componentes é oferecer ao gestor um panorama instantâneo para a tomada de decisões estratégicas.

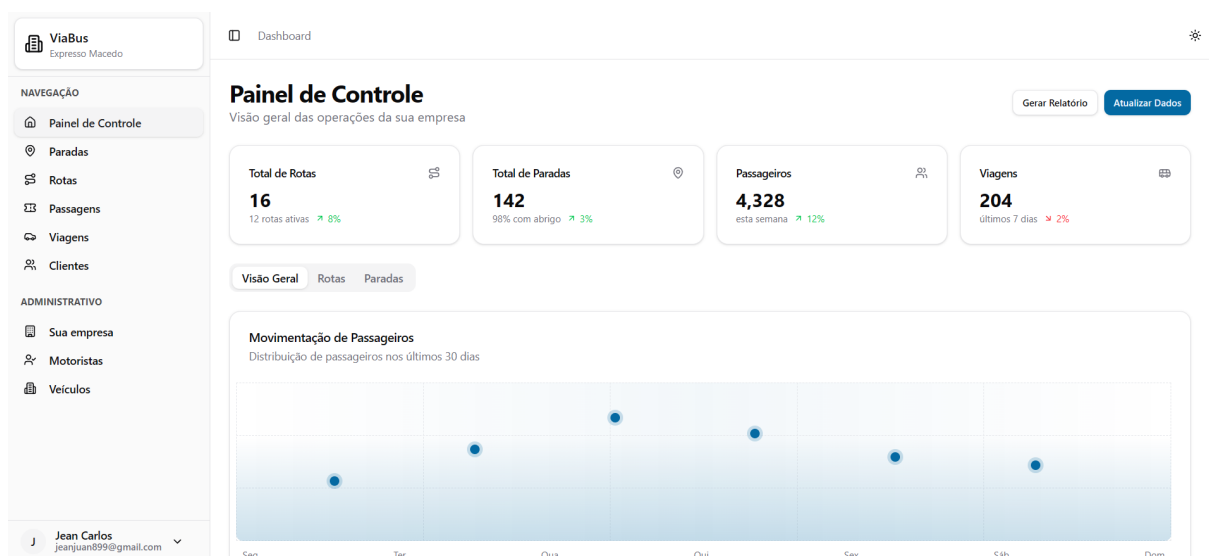


Figura 13 – Painel de controle principal com métricas e navegação.

6.2 GERENCIAMENTO DE RECURSOS OPERACIONAIS

Uma vez dentro do sistema, o gestor precisa configurar os recursos fundamentais de sua operação. Esta seção demonstra as interfaces criadas para o cadastro de paradas, rotas, veículos e motoristas.

6.2.0.1 Gestão de Paradas e Rotas

Para atender aos requisitos RF03 (cadastro de pontos de parada) e RF04 (organização de rotas), foram desenvolvidos módulos específicos. A Figura 14 exibe a tela de listagem de paradas, enquanto a Figura 15 demonstra o formulário de cadastro, que integra um mapa interativo para facilitar a definição precisa da localização. Em seguida, na interface de criação de rotas (Figura 17), o gestor pode selecionar essas paradas em uma sequência lógica para formar um itinerário.

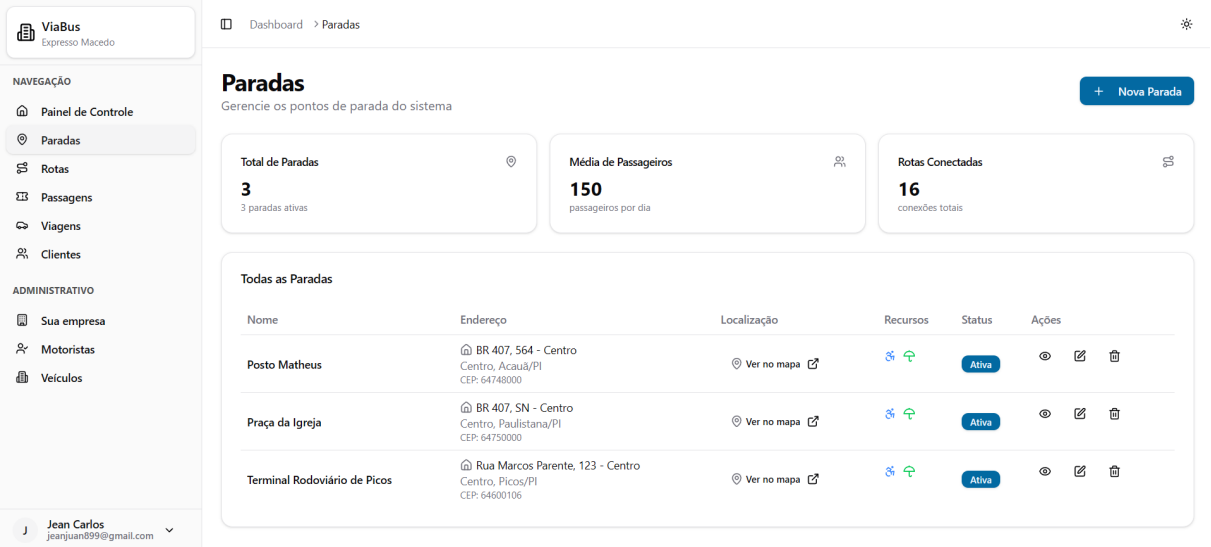


Figura 14 – Listagem de paradas com funcionalidades de busca e filtros.

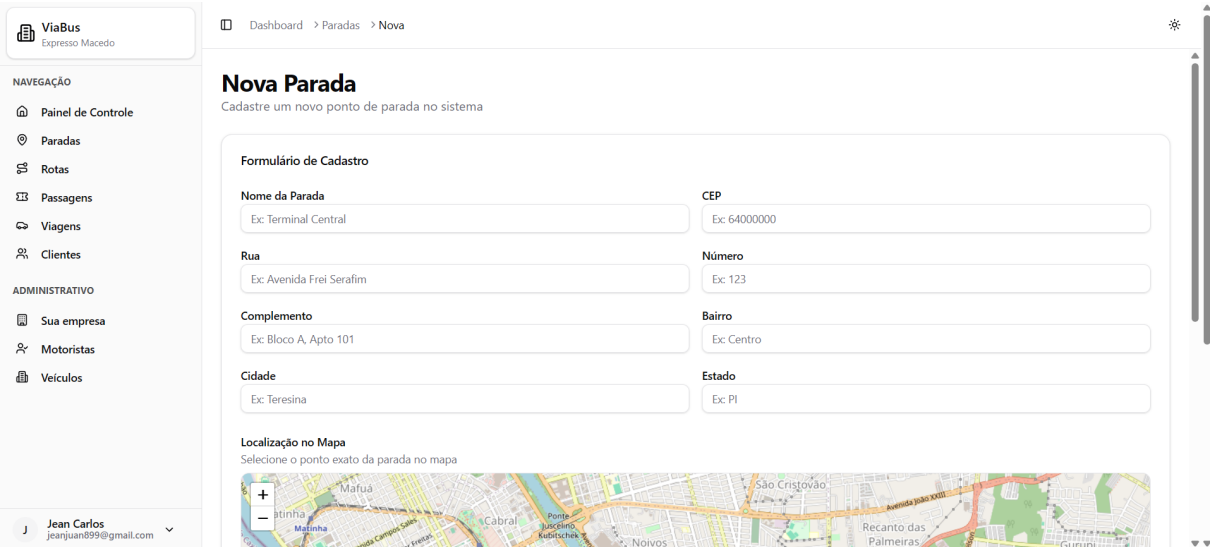


Figura 15 – Formulário de cadastro de parada com mapa interativo.

ViaBus
Expresso Macedo

Dashboard > Rotas

Rotas
Gerencie as rotas disponíveis no sistema

Total de Rotas
Rotas cadastradas no sistema
3

Rotas Ativas
Rotas em operação
3

Paradas Totais
Total de pontos de parada
7

Todas as Rotas

Nome	Descrição	Paradas	Duração	Distância	Status	Ações
Acauã - Picos	Linha dos domingos	3 paradas	2h 1min	166.9 km	Ativa	👁️ 📄 🗑️ Programação
Rota 02	Rota da semana	2 paradas	2h	166.6 km	Ativa	👁️ 📄 🗑️ Programação
Rota 02	Finalis de semana	2 paradas	2h 2min	167.9 km	Ativa	👁️ 📄 🗑️ Programação

Jean Carlos
jeanjuan899@gmail.com

Figura 16 – Listagem de rotas com funcionalidades de busca e filtros.

Criar Nova Rota

Informações da Rota

Nome da Rota
Ex: Centro - Aeroporto

Status
☒ Rota Ativa

Descrição
Descrição da rota...

Paradas da Rota 3 paradas

Buscar parada por nome ou endereço...

1ª Parada: **Posto Matheus**
BR 407, Centro
Acessível Abrigo
Horário de Partida (opcional) --:--

2ª Parada: **Praça da Igreja**
BR 407, Centro
Acessível Abrigo
Horário de Partida (opcional) --:--

3ª Parada: **Terminal Rodoviário de Picos**
Rua Marcos Parente, Centro
Acessível Abrigo

Figura 17 – Interface de criação de rota com seleção de paradas.

6.2.0.2 Gestão da Frota e de Motoristas

Atendendo aos requisitos RF01 (gerenciar frota) e RF02 (gerenciar motoristas), o sistema oferece interfaces dedicadas para o controle de veículos e condutores. As Figuras 18 e 19 mostram as telas de listagem, onde é possível consultar informações detalhadas, verificar o status operacional de cada recurso e realizar ações de gerenciamento.

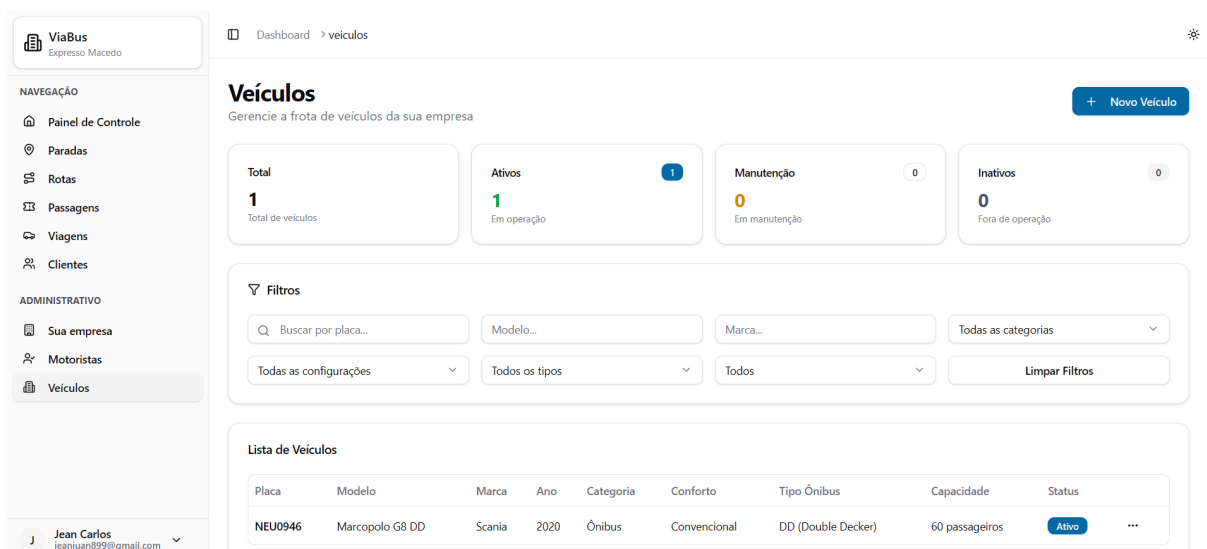


Figura 18 – Listagem de veículos com status operacional.

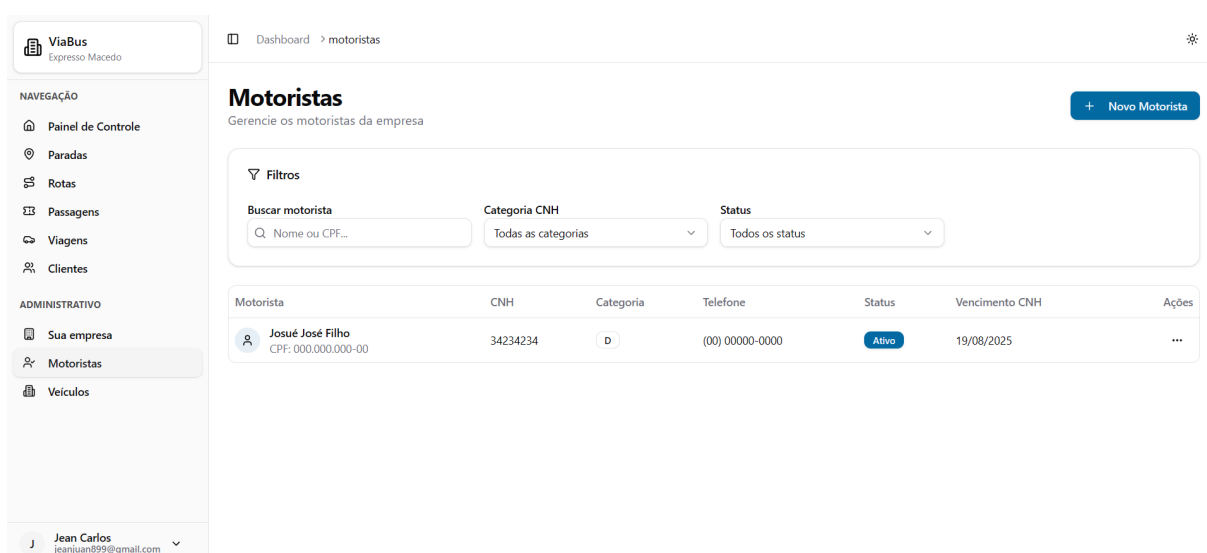


Figura 19 – Interface de gerenciamento de motoristas.

6.3 EXECUÇÃO DAS OPERAÇÕES DIÁRIAS

Com os recursos configurados, esta seção demonstra as funcionalidades centrais para a operação do dia a dia: o agendamento de viagens e a venda de passagens, que digitalizam o processo antes realizado em cadernos e aplicativos de mensagem.

6.3.0.1 Agendamento de Viagens

O requisito RF05 (agendamento de viagens futuras) é materializado na interface da Figura 20. Nela, o gestor pode criar uma nova viagem selecionando uma rota pré-

definida e associando os recursos necessários, como o veículo e o motorista que realizarão o trajeto.

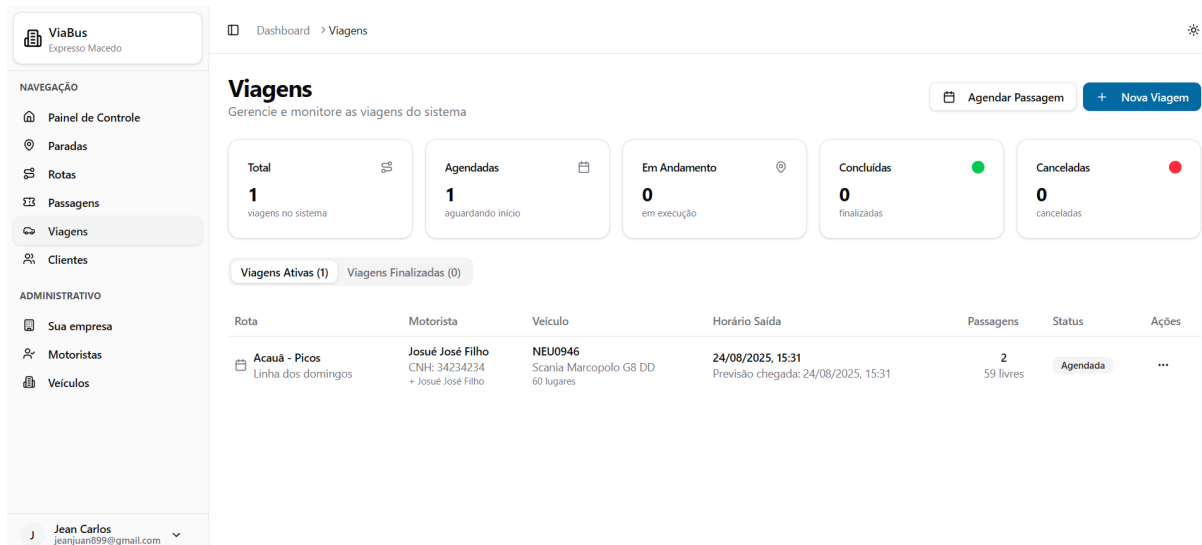


Figura 20 – Processo de agendamento de viagem.

6.3.0.2 Assistente de Venda de Passagens

O processo de venda de passagens foi projetado para ser rápido e à prova de erros, atendendo a um conjunto de requisitos essenciais. Para cumprir o RF06 (venda rápida e guiada), foi implementado um assistente (*wizard*) em cinco etapas. O fluxo se inicia com a seleção da rota e horário (Figuras 21 e 22). Em seguida, são coletadas as informações do passageiro, conforme o RF07 (Figura 23), e selecionados os locais de embarque e desembarque (Figura 24). Durante todo o processo, o sistema valida a disponibilidade de assentos em tempo real, satisfazendo o requisito crítico RF08 (evitar overbooking). A etapa final (Figura 25) resume a venda e gera o bilhete, que pode ser usado para compor a lista de passageiros do RF09.

Agendar Nova Passagem

Selecione a rota, data e preencha os dados do passageiro. O sistema criará automaticamente uma viagem se necessário.

Rota → Data → Passageiro → Locais

Selecionar Rota

Seleção de Rota

Acauã - Picos

Disponível

Linha dos domingos

Saída: 18:31

Disponibilidade

4 datas

31/08 07/09 14/09 +1 mais

Paradas

Terminal Rodoviário de Picos 10:00

Praça da Igreja 19:00

+1 paradas

Selecionar Rota

Figura 21 – Seleção de rota no wizard de passagens.

Agendar Nova Passagem

Selecione a rota, data e preencha os dados do passageiro. O sistema criará automaticamente uma viagem se necessário.

Rota → Data → Passageiro → Locais

Voltar

Seleção de Data

Acauã - Picos

Linha dos domingos

Horário de saída: 18:31

agosto 2025

Dom	Seg	Ter	Qua	Qui	Sex	Sáb
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Próximas datas disponíveis:

domingo, 31 de agosto 18:31

domingo, 7 de setembro 18:31

domingo, 14 de setembro 18:31

Figura 22 – Seleção de data e horário.

Agendar Nova Passagem ×

Selecione a rota, data e preencha os dados do passageiro. O sistema criará automaticamente uma viagem se necessário.

Rota → Data → **Passageiro** → Locais Voltar

👤

Dados do Passageiro

Informações Pessoais

Nome Completo *

Telefone *

CPF

Email

● Campos obrigatórios

● Nome completo

● Telefone

Continuar

Figura 23 – Cadastro de informações do passageiro.

Agendar Nova Passagem ×

Selecione a rota, data e preencha os dados do passageiro. O sistema criará automaticamente uma viagem se necessário.

Rota → Data → Passageiro → **Locais** Voltar

📍

Pontos de Embarque e Desembarque

● Ponto de Embarque

☒ Parada Oficial

☐ Local Especifico

Selecione a parada

● Ponto de Desembarque

☒ Parada Oficial

☐ Local Especifico

Selecione a parada

Preço (R\$)

Observações

Confirmar Agendamento

Figura 24 – Seleção de locais de embarque e desembarque.

Agendar Nova Passagem

Selecione a rota, data e preencha os dados do passageiro. O sistema criará automaticamente uma viagem se necessário.

Rota → Data → Passageiro → Locais

Voltar

Confirmação do Agendamento

Detalhes da Viagem

Data domingo, 31 de agosto de 2025	Horário 18:31
Rota Acauã - Picos	

Dados do Passageiro

Nome Jean Carlos	Telefone 89994077294
----------------------------	--------------------------------

Pontos de Embarque e Desembarque

Embarque Parada não selecionada	Desembarque Parada não selecionada
---	--

Valor e Observações

Preço
R\$ 60,00

Confirmar Agendamento

Figura 25 – Confirmação final e geração da passagem.

7 RESULTADOS E DISCUSSÃO

7.1 VERIFICAÇÃO DE REQUISITOS

A primeira etapa da avaliação do protótipo consistiu na verificação sistemática do grau de implementação dos Requisitos Funcionais (RF) e Não Funcionais (RNF), conforme definidos no Capítulo 3. Esta análise busca constatar de forma objetiva se as funcionalidades e características de qualidade planejadas foram efetivamente traduzidas no software desenvolvido. As Tabelas 7 e 8 apresentam o resultado consolidado desta verificação.

Tabela 7 – Verificação dos Requisitos Funcionais (RF)

ID	Status	Observações e Evidências
RF01	Implementado	O gestor pode cadastrar e gerenciar a frota de veículos através do módulo dedicado, conforme demonstrado na Figura 18.
RF02	Implementado	O módulo de motoristas permite o CRUD completo das informações dos condutores (Figura 19).
RF03	Implementado	O sistema permite o cadastro de pontos de parada, incluindo sua localização via mapa interativo (Figuras 14 e 15).
RF04	Implementado	A funcionalidade de criação e organização de rotas, com sequência de paradas, está implementada (Figura 17).
RF05	Implementado	O agendamento de viagens, associando rota, veículo e motorista, é funcional, como visto na Figura 20.
RF06	Implementado	O assistente (<i>wizard</i>) de venda de passagens guia o usuário pelo processo de forma rápida, conforme demonstrado a partir da Figura 21.
RF07	Implementado	O formulário de venda de passagens coleta e armazena as informações essenciais do passageiro (Figura 23).
RF08	Parcialmente - Implementado	O sistema controla o número total de assentos disponíveis, impedindo a venda acima da capacidade (<i>overbooking</i>). A funcionalidade de gerenciamento de assentos específicos (mapa de assentos) foi definida como uma melhoria futura.
RF09	Parcialmente - Implementado	A geração de listas de passageiros por viagem é funcional. No entanto, a opção de imprimir um manifesto e um módulo dedicado ao motorista para controle de embarque não foram contemplados no escopo deste protótipo.
RF10	Parcialmente - Implementado	O sistema exibe o número de assentos livres, permitindo ao gestor inferir a ocupação. A visualização direta da "taxa de ocupação" em formato de dashboard (Figura 13) é uma representação visual que, em futuras iterações, será conectada aos dados em tempo real.
RF11	Não Implementado	A geração de relatórios financeiros foi deixada fora do escopo do protótipo devido a limitações de tempo, sendo um requisito prioritário para versões futuras do sistema.

Tabela 8 – Verificação dos Requisitos Não Funcionais (RNF)

ID	Status	Observações e Evidências
RNF01	Implementado	A interface se mostrou clara e intuitiva nos principais fluxos de trabalho. A avaliação heurística, discutida na próxima seção, aponta pontos de melhoria pontuais.
RNF02	Implementado	O sistema foi desenvolvido com Tailwind CSS, garantindo a responsividade e a adaptação da interface a diferentes tamanhos de tela (desktop, tablet e mobile).
RNF03	Implementado	As tecnologias escolhidas (Next.js com SSR) resultaram em tempos de carregamento de página adequados para os fluxos de trabalho testados.
RNF04	Implementado	A arquitetura de implantação no Google Cloud Run foi projetada para alta disponibilidade, embora testes de estresse em larga escala não tenham sido parte do escopo.
RNF05	Implementado	O acesso é protegido por autenticação baseada em JWT, com rotas do backend e do frontend devidamente protegidas.
RNF06	Implementado	A arquitetura multi-tenant com filtro automático por <i>companyId</i> garante o isolamento lógico dos dados de cada empresa na camada de aplicação.

7.2 AVALIAÇÃO HEURÍSTICA DA INTERFACE

Após a verificação dos requisitos, foi realizada uma Avaliação Heurística no protótipo do ViaBus, com o objetivo de identificar possíveis problemas de usabilidade na interface. Este método consiste na inspeção da interface por um avaliador, que analisa com base em um conjunto de princípios e diretrizes de usabilidade reconhecidos, conhecidos como "heurísticas" (Nielsen, 1994). Para esta análise, foram utilizadas as 10 Heurísticas de Usabilidade de Jakob Nielsen, um dos conjuntos de princípios mais consolidados na área de Interação Humano-Computador.

A avaliação buscou identificar tanto os pontos fortes do *design* proposto quanto as oportunidades de melhoria. Os resultados são discutidos a seguir, agrupados pelas heurísticas mais relevantes para o contexto do protótipo.

7.2.0.1 Pontos Fortes da Interface

O design do protótipo demonstrou forte aderência a diversas heurísticas fundamentais, resultando em uma base de interação sólida.

- Heurística 1: Visibilidade do status do sistema. O sistema se comunica de forma eficaz com o usuário. Como visto na Figura 21, o assistente de venda de passagens exibe claramente em qual das cinco etapas o usuário se encontra. Além disso, o uso consistente de *toasts* de notificação (Figura 16, por exemplo) para confirmar ações como "Rota criada com sucesso!" fornece feedback imediato e claro, mantendo o usuário informado sobre o resultado de suas operações.

- Heurística 4: Consistência e padronização. A interface do ViaBus mantém um alto grau de consistência visual e de interação. A estrutura de navegação, com o menu lateral (*sidebar*) presente em todas as telas de gerenciamento (Figura 13), e o padrão de tabelas com filtros e botões de ação (Figuras 14 e 18) se repetem em todos os módulos. Essa padronização reduz a carga cognitiva do usuário, que não precisa reaprender a usar o sistema a cada nova tela.
- Heurística 8: Estética e design minimalista. A interface adota um design limpo e focado no conteúdo, evitando elementos visuais desnecessários que poderiam distrair o usuário. Os formulários, como o de criação de parada (Figura 15), são um bom exemplo: os campos são claramente rotulados e organizados, apresentando apenas as informações essenciais para a tarefa em questão.

7.2.0.2 Oportunidades de Melhoria

A avaliação também identificou pontos onde a interface pode ser aprimorada para evitar erros e reduzir a fricção do usuário.

- Heurística 5: Prevenção de erros. Embora o sistema valide os dados no *backend*, a interface poderia ser mais proativa na prevenção de erros. Por exemplo, ao confirmar um agendamento de passagem (Figura 20), o botão "Confirmar" não deve estar ativo se as informações como paradas de embarque e desembarque e preço total da passagem não forem informadas.
- Heurística 6: Reconhecimento em vez de memorização. Na interface de agendamento de viagem (Figura 20), os campos para associar veículo e motorista são campos de busca. Embora funcional, o gestor precisa memorizar a placa do veículo ou o nome do motorista para encontrá-los. Uma melhoria significativa seria substituir esses campos por um seletor (*dropdown*) ou uma busca com sugestões automáticas, que listaria os recursos disponíveis. Isso trocaria a necessidade de memorização pela simples tarefa de reconhecimento, tornando o processo mais rápido e menos propenso a erros.

Em suma, a avaliação heurística indica que o protótipo do ViaBus possui uma base de usabilidade sólida, com destaque para a clareza, consistência e design minimalista. As principais oportunidades de aprimoramento concentram-se na implementação de mecanismos para a prevenção de erros e na redução da carga de memorização do usuário em fluxos de trabalho específicos.

7.3 DISCUSSÃO DOS RESULTADOS

A análise dos resultados indica que o protótipo do ViaBus atingiu seu objetivo primário: materializar as funcionalidades essenciais para a digitalização da gestão de transporte alternativo. A verificação de requisitos demonstrou que o núcleo operacional do sistema — abrangendo o gerenciamento de recursos como frotas e motoristas, e a execução de operações como agendamento de viagens e vendas — foi implementado com sucesso. Contudo, a análise também revelou que funcionalidades analíticas avançadas, como a de relatórios, foram designadas como trabalhos futuros devido a limitações no escopo do projeto. Do ponto de vista da usabilidade, a avaliação heurística revelou uma base de interface sólida e consistente, mas também apontou para a necessidade de refinar mecanismos de prevenção de erros para garantir uma operação mais segura no dia a dia.

A principal contribuição do protótipo reside na sua capacidade de atacar diretamente os desafios operacionais revelados pela pesquisa de mercado (ver Apêndice B). A dependência de "cadernos de anotações" e processos manuais via WhatsApp, uma realidade para 100% dos gestores entrevistados, é diretamente substituída pelo assistente de venda de passagens (Figuras 21 a 25), que centraliza e digitaliza o processo. O requisito RF08 (Anti-Overbooking), que foi implementado, soluciona uma das maiores preocupações operacionais — a venda de assentos acima da capacidade —, um problema recorrente no controle manual. Da mesma forma, a geração automática da lista de passageiros (RF09) elimina a "perda de tempo organizando listas", um dos principais entraves à eficiência citados na pesquisa.

Apesar dos avanços, é crucial reconhecer as limitações funcionais do protótipo em seu estado atual. A não implementação do módulo de relatórios financeiros (RF11) é a lacuna mais significativa, pois impede que o gestor extraia inteligência de negócio a partir dos dados coletados, uma das vantagens centrais prometidas pelo modelo SaaS. Adicionalmente, a implementação parcial do RF09, que não inclui um manifesto de viagem ou módulo dedicado ao motorista, significa que a "comunicação com os motoristas" — outro desafio identificado na pesquisa — ainda não está totalmente digitalizada no protótipo, resolvendo apenas parte do problema de comunicação entre a gestão e a operação de embarque.

A avaliação heurística reforça que a usabilidade da plataforma é um ponto forte, especialmente sua consistência e clareza (aderência às Heurísticas 4 e 8 de Nielsen), o que é vital para um público-alvo que pode não ter familiaridade com sistemas complexos. Contudo, a falha identificada na "Prevenção de Erros" (Heurística 5) — a ausência de confirmação antes de ações destrutivas — representa um risco operacional real. Em um ambiente de produção, um clique acidental poderia levar à exclusão de uma rota inteira, demonstrando que a robustez de um sistema não depende apenas de sua funcionalidade, mas também de mecanismos que protejam o usuário de erros.

inadvertidos.

Finalmente, ao contrastar o protótipo com as soluções de mercado citadas na Fundamentação Teórica, fica evidente o nicho que o ViaBus busca ocupar. Comparado a sistemas de grande porte, ele é funcionalmente mais enxuto. No entanto, sua proposta de valor não reside na competição direta de funcionalidades, mas na simplicidade, usabilidade e baixo custo de entrada, preenchendo a lacuna de mercado para pequenas e médias empresas de transporte alternativo que consideram as soluções existentes complexas ou com um custo proibitivo.

8 CONSIDERAÇÕES FINAIS

Este trabalho de conclusão de curso abordou o desafio da gestão operacional no setor de transporte rodoviário alternativo de passageiros, um segmento caracterizado pela carência de soluções tecnológicas acessíveis. O problema central identificado foi a dependência de métodos manuais e fragmentados, que resultam em ineficiências e risco de erros. Diante deste cenário, o objetivo geral do projeto foi desenvolver um protótipo funcional de uma plataforma web, no modelo *Software as a Service* (SaaS), denominada ViaBus, para centralizar e digitalizar essa gestão.

8.1 CONCLUSÃO

Conclui-se que o objetivo geral do trabalho foi atingido, uma vez que o protótipo funcional foi desenvolvido e sua arquitetura, detalhada. A implementação materializou os requisitos essenciais levantados na pesquisa de mercado — como o gerenciamento de recursos e a execução de operações de venda —, demonstrando que a abordagem técnica proposta é uma solução plausível para o problema. As contribuições deste projeto são, portanto, de ordem prática e documental: o próprio protótipo funcional, que serve como prova de conceito, e a documentação de uma proposta de arquitetura para a construção de um sistema SaaS *multi-tenant* focado neste nicho de mercado.

8.2 LIMITAÇÕES DO TRABALHO

É fundamental reconhecer as limitações inerentes a este trabalho para contextualizar o alcance de suas conclusões. A principal limitação metodológica reside na pesquisa de mercado, conduzida com uma amostra de apenas dois gestores, o que não permite a generalização estatística dos achados. No que tange à validação, a utilização de uma Avaliação Heurística em vez de testes com usuários finais oferece conclusões de usabilidade de caráter preliminar. Por fim, o escopo do protótipo funcional foi deliberadamente focado na perspectiva do gestor, não incluindo o desenvolvimento de uma aplicação para o passageiro e motorista, funcionalidades apontadas como relevantes na pesquisa.

8.3 TRABALHOS FUTUROS

Com base nas limitações apontadas, delineia-se um caminho claro para a evolução do projeto. Recomenda-se, primeiramente, a validação da solução em maior escala, através de uma pesquisa de mercado quantitativa e da condução de testes de usabilidade com usuários reais. Em paralelo, o desenvolvimento do protótipo pode

avancar com a implementação de funcionalidades de relatório (RF11) e do manifesto de viagem para o motorista (RF09). A criação de um único e versátil aplicativo móvel que, por meio de perfis de usuário distintos, atenderia tanto ao passageiro — com funcionalidades para a compra de bilhetes e o rastreamento de viagens — quanto ao motorista — oferecendo ferramentas para o controle digital do embarque — e a integração com sistemas de pagamento online representam os passos subsequentes para transformar o protótipo em um produto de mercado.

REFERÊNCIAS

ALIM. **An overview of TypeORM.** [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://dev.to/alim1496/an-overview-of-typeorm-1cp2>. Citado na p. 20.

BLOG PRAXIO. **Vantagens de um software SaaS na gestão de transporte.** [S. l.: s. n.], 2021. Acesso em: 22 ago. 2025. Disponível em: <https://blog.praxio.com.br/vantagens-de-um-software-saas-na-gestao-de-transporte/>. Citado na p. 19.

BLOODTURTLE. **Understanding Server-Side Rendering in Next.js 15: Benefits and Drawbacks.** [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://medium.com/@bloodturtle/understanding-server-side-rendering-in-next-js-15-benefits-and-drawbacks-5f4a11346666>. Citado na p. 21.

BRASIL. **Lei Nº 10.233, de 5 de junho de 2001. Dispõe sobre a reestruturação dos transportes aquaviário e terrestre, cria o Conselho Nacional de Integração de Políticas de Transporte, a Agência Nacional de Transportes Terrestres, a Agência Nacional de Transportes Aquaviários e o Departamento Nacional de Infra-Estrutura de Transportes, e dá outras providências.** [S. l.: s. n.], 2001. Diário Oficial da União. Citado na p. 15.

DEV AND DELIVER. **A C-suite guide to mastering architecture in NestJS apps.** [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://devanddeliver.com/blog/backend/level-up-your-backend-a-c-suite-guide-to-mastering-modular-architecture-in-nest-js-applications>. Citado na p. 20.

DHIWISE. **10 Reasons Why You Should Use TypeScript With React.** [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://www.dhiwise.com/post/10-reasons-why-you-should-use-typescript-with-react>. Citado na p. 21.

ENGETEC - FATEC ZONA LESTE. **Impactos tecnológicos no transporte rodoviário de cargas no Brasil.** [S. l.: s. n.], 2022. Acesso em: 22 ago. 2025. Disponível em: https://www.fateczl.edu.br/engetec/engetec_2022/5_EnGeTec_paper_023.pdf. Citado na p. 19.

ESTADO DO PIAUÍ. **Lei Nº 8.562, de 07 de janeiro de 2025. Dispõe sobre o Sistema de Transporte Rodoviário Intermunicipal de Passageiros do Estado do Piauí – STRIP/PI.** [S. l.: s. n.], jan. 2025. Publicado no Diário Oficial do Estado do Piauí em 09 de janeiro de 2025. Disponível em: <https://www.legisweb.com.br/legislacao/?id=471810>. Citado na p. 15.

FGV TRANSPORTES. Desafios e perspectivas do transporte rodoviário intermunicipal e interestadual no Brasil. **Repositório Digital FGV**, 2023. Disponível em: <https://repositorio.fgv.br/items/f215c0df-2fbe-4c55-b86b-a9c11c99a67f>. Citado na p. 15.

FRONTEGG. **The Complete Guide to SaaS Multi-Tenant Architecture**. [S. l.: s. n.], 2021. Acesso em: 22 ago. 2025. Disponível em: <https://frontegg.com/wp-content/uploads/2021/11/The-complete-guide-to-multi-tenant-SaaS-architecture-Frontegg-.pdf>. Citado na p. 18.

IGNATOVICH.DM. **CSS Modules vs CSS-in-JS vs Tailwind CSS: A Comprehensive Comparison**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://medium.com/@ignatovich.dm/css-modules-vs-css-in-js-vs-tailwind-css-a-comprehensive-comparison-24e7cb6f48e9>. Citado na p. 21.

ITRANSPORT. **iTransport - Software para gestão de transporte e frota**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://itransport.com.br/>. Citado na p. 20.

LOGROCKET BLOG. **TypeORM: Object-relational mapping with Node.js**. [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://blog.logrocket.com/typeorm-object-relational-mapping-node-js/>. Citado na p. 20.

MELO, Cássio A; ARCOVERDE, D; MORAES, E; PIMENTEL, J; FREITAS, R. Software como serviço: Um modelo de negócio emergente. **Paper Centro de Informática–Universidade Federal de Pernambuco (UFPE)**, p. 27, 2007. Citado na p. 15.

MOVEIDEIAS. **Vantagens de usar software SaaS em logística**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://moveideias.com.br/vantagens-de-usar-software-saas-em-logistica/>. Citado na p. 18.

NESTJS. **NestJS - A progressive Node.js framework**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://nestjs.com/>. Citado na p. 20.

NESTJS DOCS. **Modules | NestJS - A progressive Node.js framework**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://docs.nestjs.com/modules>. Citado na p. 20.

NEXT.JS DOCS. **Server Components - Rendering - Next.js**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://nextjs.org/docs/14/app/building-your-application/rendering/server-components>. Citado na p. 21.

NIELSEN, Jakob. **Usability Engineering**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. ISBN 0125184069. Citado nas pp. 17, 48.

PASSPORT.JS. **passport-jwt**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://www.passportjs.org/packages/passport-jwt/>. Citado na p. 21.

PRAXIO. **Praxio Luna - Venda de Passagens**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Baseado na análise de mercado que descreve o produto. Citado na p. 20.

PROLOG APP. **Como uma solução SaaS impacta a gestão de frotas e transportes**. [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://www.prologapp.com/blog/saas-de-logistica/>. Citado nas pp. 18, 19.

PROLOGAPP. **SaaS de logística: como funciona e quais as vantagens?** Acesso em: 10 mar. 2025. 2024. Disponível em: <https://prologapp.com/blog/saas-de-logistica/>. Citado na p. 16.

SEST SENAT. **Relatório Final - Transformação Digital no Transporte**. [S. l.: s. n.], 2021. Acesso em: 22 ago. 2025. Disponível em: <https://publicador.sestsenat.org.br/arquivos/8d212be2-9b94-47bc-99c6-d0129b9be8d3.pdf>. Citado na p. 19.

SETCEPAR. **Digitalização reduz os custos nos transportes**. Acesso em: 10 mar. 2025. 2023. Disponível em: <https://setcepar.com.br/comunicacao/digitalizacao-reduz-os-custos-nos-transportes>. Citado na p. 16.

SHADCN/UI. **Introduction - shadcn/ui**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://ui.shadcn.com/docs>. Citado na p. 21.

SOMMERVILLE, I. **Software Engineering**. [S. l.]: Pearson, 2011. (International Computer Science Series). ISBN 9780137053469. Disponível em: <https://books.google.com.br/books?id=l0egcQAACAAJ>. Citado na p. 17.

SOSHACE. **Securing Node.js Applications with JWT and Passport.js**. [S. l.: s. n.], 2024. Acesso em: 22 ago. 2025. Disponível em: <https://soshace.com/securing-node-js-applications-with-jwt-and-passport-js/>. Citado na p. 21.

TOTVS. **Tudo sobre o TOTVS Prestadores de Serviços Transporte de Passageiros**. [S. l.: s. n.], 2025. Acesso em: 22 ago. 2025. Disponível em: <https://produtos.totvs.com/ficha-tecnica/tudo-sobre-o-totvs-prestadores-de-servicos-transporte-de-passageiros/>. Citado na p. 20.

APÊNDICE A – QUESTIONÁRIO DA PESQUISA DE MERCADO

Este apêndice apresenta a transcrição fiel do questionário, desenvolvido na plataforma Google Forms, que foi utilizado para o levantamento de requisitos do sistema ViaBus.

Título do Formulário: ViaBus: simplificando a gestão do seu transporte de passageiros (pesquisa de mercado)

PERGUNTAS

1. Em qual empresa ou serviço você atua? (Opcional)

Campo para resposta curta.

2. Qual o tamanho da frota de veículos?

Marcar apenas uma oval.

- 1 veículo
- 2 a 5 veículos
- 6 a 10 veículos
- 11 a 20 veículos
- Mais de 20 veículos

3. Quantas rotas (trechos) diferentes a empresa opera regularmente?

Marcar apenas uma oval.

- 1 a 2 rotas
- 3 a 5 rotas
- 6 a 10 rotas
- Mais de 10 rotas
- Não opero rotas fixas (apenas fretamento)

4. A operação é principalmente:

Marcar apenas uma oval.

- Intermunicipal (dentro do mesmo estado)
- Interestadual (entre estados diferentes)
- Ambos

5. Como é gerenciada a venda de passagens e a lista de passageiros hoje?

Marque todas que se aplicam.

- Caderno de anotações / Planilha em papel
- Planilhas no computador (Excel, Google Sheets)
- Mensagens diretas (WhatsApp, Telegram)
- Ligação telefônica
- Diretamente com o motorista, na hora do embarque
- Utilizo outro sistema/software
- Outro

6. Quais são os maiores desafios operacionais hoje para a empresa?

Marque todas que se aplicam.

- Realizar o embarque de passageiros com planilhas manuais.
- Controlar quais assentos já foram vendidos e quais estão livres.
- Perda de tempo organizando listas de passageiros e viagens.
- Prejuízo com assentos vazios (baixa ocupação)
- Comunicação com os motoristas sobre a rota e os passageiros.
- Receber os pagamentos (muito dinheiro em espécie, dificuldade com Pix, etc.).
- Outro

7. Quão valioso seria ter um sistema único para gerenciar rotas, horários, veículos, motoristas e passagens em um só lugar, acessível pelo celular ou computador?

Marcar apenas uma oval, em uma escala de 1 a 5.

- 1 2 3 4 5

8. Quais funcionalidades de um sistema como o ViaBus seriam mais importantes para o seu negócio?

Marque todas que se aplicam.

- Embarque facilitado através de um aplicativo/site, substituindo planilhas manuais.
- Painel para cadastrar e organizar minhas rotas, horários e preços.
- Possibilidade de agendamento externo (ex.: prefeituras às quais a empresa presta serviços).

- Venda de passagens online (com pagamento por Pix e cartão).
- Cadastro e gestão de motoristas e veículos.
- Relatórios financeiros (vendas por período, por rota, etc.).
- Outro

9. Se você pudesse oferecer aos seus passageiros um aplicativo para eles mesmos comprarem e acompanharem a viagem (ver horário, paradas, etc.), você acredita que isso seria um diferencial contra seus concorrentes?

Marcar apenas uma oval.

- Sim, com certeza seria um grande diferencial.
- Talvez, poderia atrair mais clientes.
- Não, acho que não faria diferença.
- Tenho dúvidas se meus clientes usariam.

10. Qual modelo de parceria você consideraria mais justo para usar um sistema completo como este?

Marcar apenas uma oval.

- Uma pequena taxa fixa por mês.
- Uma pequena porcentagem (%) sobre cada passagem vendida pelo sistema.
- Um modelo misto (taxa mensal baixa + porcentagem pequena).
- Não tenho certeza.

11. Gostaria de receber um convite para testar a plataforma ViaBus em primeira mão e sem custos quando estiver disponível?

Marcar apenas uma oval.

- Sim, tenho interesse!
- Não, obrigado.
- Talvez, gostaria de mais informações antes.

APÊNDICE B – RESULTADOS DA PESQUISA DE MERCADO

Este apêndice apresenta os resultados visuais coletados através da pesquisa de mercado realizada com dois gestores de empresas de transporte de passageiros.

Em qual empresa ou serviço você atua? (Opcional)

2 respostas

Expresso Macedo

Marcio e Regis Transportes e Serviços Ltda

Figura 26 – Resultado da pergunta sobre empresa ou serviço.

Qual o tamanho da frota de veículos?

2 respostas

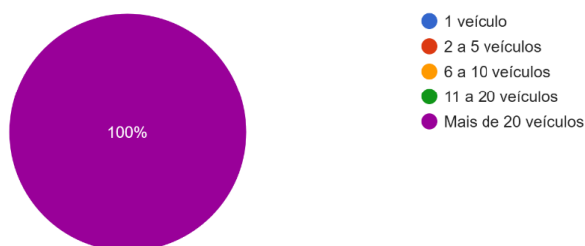


Figura 27 – Resultado da pergunta sobre tamanho da frota de veículos.

Quantas rotas (trechos) diferentes a empresa opera regularmente?

2 respostas

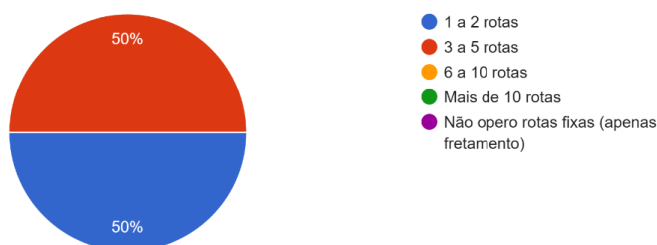


Figura 28 – Resultado da pergunta sobre número de rotas operadas.

A operação é principalmente:

2 respostas

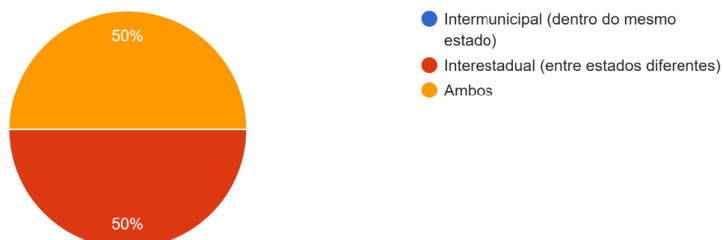


Figura 29 – Resultado da pergunta sobre tipo de operação.

Como é gerenciada a venda de passagens e a lista de passageiros hoje?

2 respostas

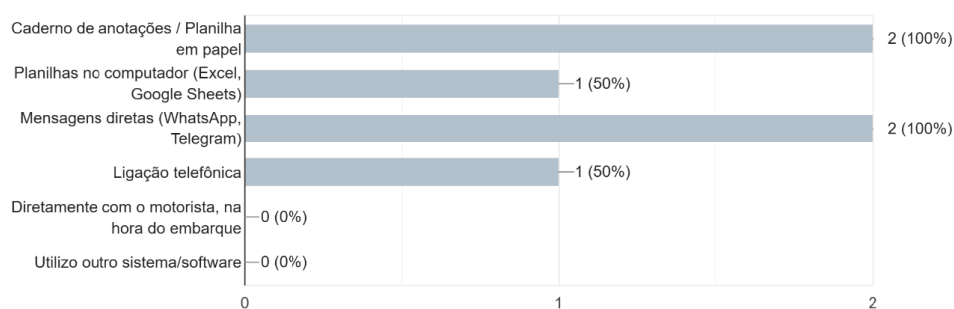


Figura 30 – Resultado da pergunta sobre gerenciamento atual de passagens.

Quais são os maiores desafios operacionais hoje para a empresa?

2 respostas

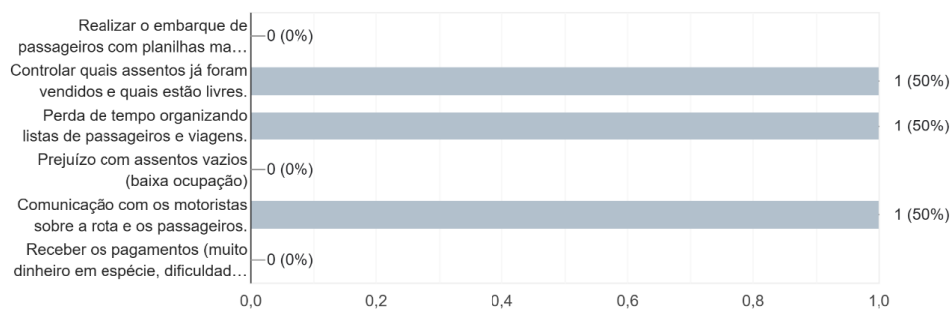


Figura 31 – Resultado da pergunta sobre maiores desafios operacionais.

Quão valioso seria ter um sistema único para gerenciar rotas, horários, veículos, motoristas e passagens em um só lugar, acessível pelo celular ...Marcar apenas uma oval, em uma escala de 1 a 5)
2 respostas

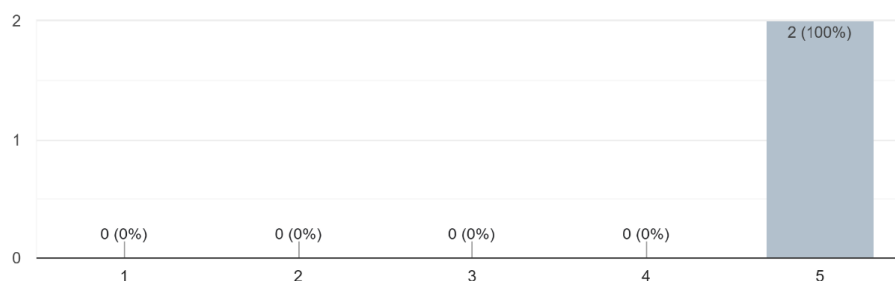


Figura 32 – Resultado da pergunta sobre valor de um sistema único.

Quais funcionalidades de um sistema como o ViaBus seriam mais importantes para o seu negócio?
2 respostas

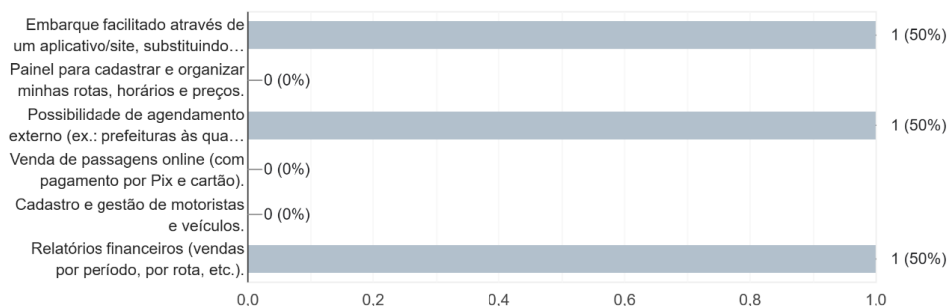


Figura 33 – Resultado da pergunta sobre funcionalidades mais importantes.

Se você pudesse oferecer aos seus passageiros um aplicativo para eles mesmos comprarem e acompanharem a viagem (ver horário, paradas, etc....)ra seus concorrentes? (Marcar apenas uma oval)
2 respostas

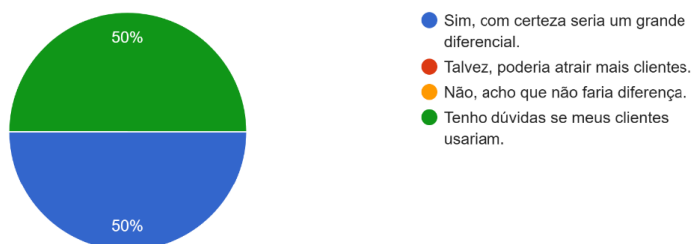


Figura 34 – Resultado da pergunta sobre aplicativo como diferencial.

Qual modelo de parceria você consideraria mais justo para usar um sistema completo como este?
2 respostas



Figura 35 – Resultado da pergunta sobre modelo de parceria.

Gostaria de receber um convite para testar a plataforma ViaBus em primeira mão e sem custos quando estiver disponível?
2 respostas

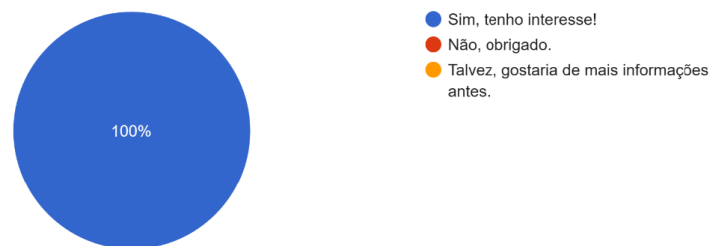


Figura 36 – Resultado da pergunta sobre interesse em testar a plataforma.

APÊNDICE C – DIAGRAMA DE CLASSE DO VIABUS

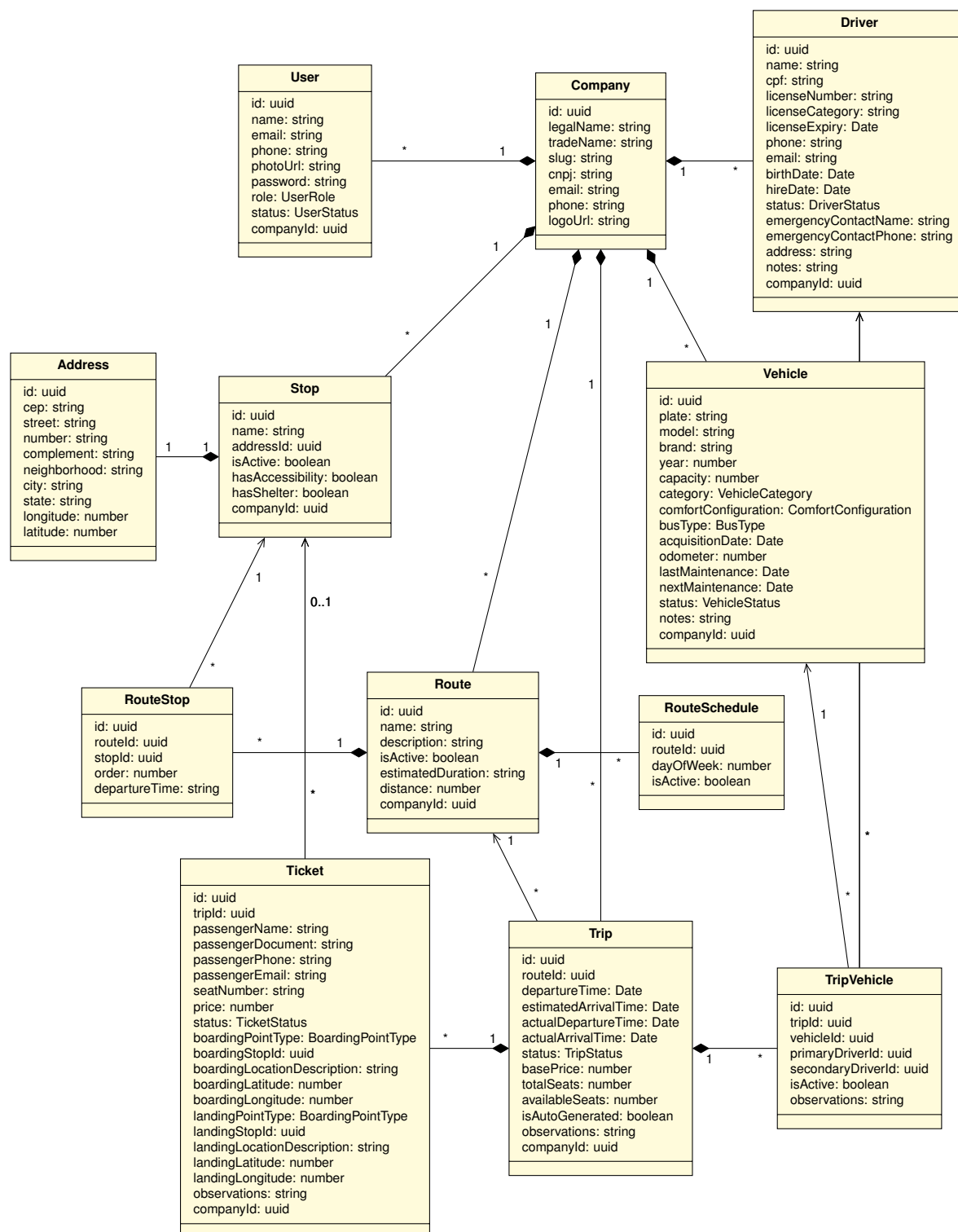


Figura 37 – Diagrama de classes do domínio ViaBus.

APÊNDICE D – REPOSITÓRIOS DO CÓDIGO-FONTE

O código-fonte completo do protótipo do sistema ViaBus, desenvolvido como parte deste trabalho, está publicamente disponível para consulta e análise nos seguintes repositórios da plataforma GitHub:

- **Repositório do Frontend (Interface do Usuário):**

<https://github.com/JeanCarlos899/viabus-front-end>

- **Repositório do Backend (API e Lógica de Negócio):**

<https://github.com/JeanCarlos899/viabus-back-end>