



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS PIRIPIRI
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

RAUAN DOS SANTOS BANDEIRA

**DESENVOLVIMENTO E AVALIAÇÃO DE UMA SOLUÇÃO SAAS *MULTI-TENANT*
PARA GESTÃO DO TRANSPORTE UNIVERSITÁRIO INTERMUNICIPAL**

**PIRIPIRI/PI
2026**

RAUAN DOS SANTOS BANDEIRA

DESENVOLVIMENTO E AVALIAÇÃO DE UMA SOLUÇÃO SAAS *MULTI-TENANT*
PARA GESTÃO DO TRANSPORTE UNIVERSITÁRIO INTERMUNICIPAL

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Piripiri.

Orientador(a): Prof. Me. Jeferson do Nascimento Soares

DESENVOLVIMENTO E AVALIAÇÃO DE UMA SOLUÇÃO SAAS *MULTI-TENANT* PARA GESTÃO DO TRANSPORTE UNIVERSITÁRIO INTERMUNICIPAL

Rauan dos Santos Bandeira¹

Prof. Me. Jeferson do Nascimento Soares²

RESUMO

Este estudo foi conduzido a partir do serviço de transporte universitário intermunicipal da rota entre Campo Maior e Piripiri, no Piauí. No contexto investigado, reservas, confirmações de presença, acompanhamento de pagamentos e organização das viagens eram realizados com apoio de listas em aplicativos de mensagens e controles dispersos, o que dificultava o controle de capacidade e a consolidação das informações operacionais. O objetivo foi desenvolver e avaliar uma solução de *software* como serviço com arquitetura *multi-tenant* para apoiar a gestão desse serviço. A solução foi materializada em um protótipo funcional composto por uma interface de programação de aplicações e uma interface *web* responsiva, projetada prioritariamente para dispositivos móveis; sua arquitetura foi organizada em camadas, inspirada em princípios da *Clean Architecture*, com regras de negócio centralizadas no servidor. A verificação técnica combinou testes unitários automatizados e implantação experimental em nuvem, abrangendo 13 módulos funcionais, 93 pontos de acesso da interface e 458 testes. A avaliação exploratória de usabilidade recebeu 19 respostas ao questionário: a análise quantitativa considerou as 19 respostas, enquanto a qualitativa examinou 18 conjuntos de respostas abertas. Os fluxos centrais foram implementados e verificados, e as médias dos itens variaram entre 4,37 e 4,89 em escala Likert de cinco pontos, fornecendo indícios iniciais favoráveis de usabilidade e utilidade percebidas. A principal limitação está na amostra pequena, não probabilística e desequilibrada entre os perfis de uso, o que impede generalizar os resultados da avaliação.

Palavras-chave: transporte universitário; *software* como serviço; desenvolvimento de *software*; arquitetura *multi-tenant*; avaliação exploratória de usabilidade.

ABSTRACT

This study was conducted in the context of the intercity university transportation service operating between Campo Maior and Piripiri, in the state of Piauí. In the investigated setting, bookings, attendance confirmations, payment tracking, and trip organization relied on lists in messaging applications and scattered records, which hindered capacity control and the consolidation of operational information. The objective was to develop and evaluate a multi-tenant software-as-a-service solution to support the management of this service. The solution was implemented as a functional prototype comprising an application programming interface and a responsive web interface designed primarily for mobile devices; its architecture was organized in layers, inspired by *Clean Architecture* principles, with business rules centralized on the server. Technical verification combined automated unit tests and an experimental cloud deployment, covering 13 functional modules, 93 interface endpoints, and 458 tests. The exploratory usability

¹Discente do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFPI – Campus Piripiri. E-mail: capir.2024116tads0010@aluno.ifpi.edu.br.

²Docente orientador do IFPI – Campus Piripiri.

evaluation received 19 questionnaire responses: the quantitative analysis considered all 19 responses, while the qualitative analysis examined 18 sets of open-ended responses. The core flows were implemented and verified, and item means ranged from 4.37 to 4.89 on a five-point Likert scale, providing initial favorable evidence of perceived usability and usefulness. The main limitation is the small, non-probabilistic sample and its imbalance across user profiles, which prevents generalization of the evaluation results.

Keywords: intercity university transportation; software as a service; software development; multi-tenancy; exploratory usability evaluation.

Data de aprovação: 16/08/2026

1 INTRODUÇÃO

O deslocamento entre municípios faz parte da rotina de muitos universitários brasileiros, sobretudo onde a oferta de cursos presenciais se concentra em cidades-polo. A pesquisa *Regiões de Influência das Cidades 2018* (REGIC 2018), publicada pelo IBGE em 2020, indica que estudantes que frequentam o ensino superior fora do município de origem percorrem, em média, 92 quilômetros para acessar os cursos, além de apontar cidades intermediárias com forte atratividade educacional (IBGE, 2020). Segundo o Censo Demográfico de 2022 (IBGE, 2025), 27,3% dos estudantes de graduação frequentam instituição situada em município diferente daquele em que residem; em 2024, o Censo da Educação Superior registrou 5.037.875 matrículas presenciais em 2.444 instituições (INEP, 2025). Nesse cenário, o transporte universitário intermunicipal, seja custeado pelos próprios estudantes, ofertado por pequenas empresas ou organizado por prefeituras, torna-se condição concreta de acesso e permanência no ensino superior (Silva, 2022).

Quando esse transporte é instável, caro ou marcado por falhas de comunicação, os impactos recaem diretamente sobre os estudantes. A incerteza sobre horários, vagas, confirmação da viagem e custos pode gerar atrasos, faltas, desgaste diário e risco à permanência acadêmica, especialmente para quem depende do deslocamento para manter vínculo com a instituição de ensino. Estudos sobre transporte universitário apontam que a oferta regular desse serviço contribui para reduzir barreiras financeiras e favorecer a permanência no ensino superior (Silva, 2022); de forma semelhante, experiências com aplicativos de ônibus universitários indicam ganhos de previsibilidade e acesso às informações de deslocamento (Bizerril, 2024).

Além do impacto sobre os alunos, a organização manual do serviço também constitui problema de gestão para o prestador. Em muitas localidades, agendamento, confirmação de presença e cobrança são controlados por listas em aplicativos de mensagens e anotações avulsas. Esse modo de operação gera duplicidade de registros,

falhas de comunicação, dificuldade para acompanhar a situação operacional dos pagamentos, pouca visibilidade sobre vagas disponíveis e incerteza sobre a arrecadação mínima necessária para confirmar a viagem. A ausência de sistemas informatizados compromete a eficiência operacional e a tomada de decisão (Reis et al., 2021), fazendo com que decisões como confirmar, remanejar ou cancelar uma viagem sejam tomadas com pouca informação, em prejuízo tanto dos responsáveis quanto dos estudantes.

Há iniciativas que enfrentam partes desse problema. O PickBus propõe uma solução para transporte fretado de passageiros, com foco em eventos e deslocamentos coletivos (Pretto et al., 2021); Paiva (2023) apresenta um sistema para gerenciamento de transportes em contexto acadêmico-institucional; e Bizerril (2024) analisa o aplicativo Intercampi, voltado à consulta de horários e rotas de ônibus universitários. Essas contribuições mostram a relevância de soluções digitais para o transporte coletivo e universitário, mas não tratam, de forma integrada, do recorte específico de organizações que operam viagens universitárias intermunicipais recorrentes com reservas, capacidade, diferentes perfis de acesso, isolamento entre organizações e controle operacional de pagamentos.

A relevância do problema e do estudo apoia-se em quatro aspectos: trata-se de uma demanda recorrente que afeta a permanência de estudantes no ensino superior, sobretudo em contextos nos quais a oferta educacional se organiza em centralidades regionais (IBGE, 2020; Silva, 2022); entre os trabalhos analisados, o recorte específico do transporte universitário intermunicipal aparece pouco explorado, pois os estudos identificados tratam do fretamento eventual, de frotas institucionais, de ônibus universitários intercampi ou do transporte escolar municipal; o estudo documenta decisões de arquitetura aplicáveis a outros domínios de gestão compartilhada de recursos; e o protótipo resultante constitui base concreta para evolução futura.

Diante dessa lacuna, este estudo busca responder a duas questões de pesquisa. A primeira, relacionada ao desenvolvimento do protótipo, é: como estruturar a arquitetura de um protótipo *Software as a Service* (SaaS) *multi-tenant* para apoiar a gestão operacional do transporte universitário intermunicipal? A segunda, relacionada à avaliação exploratória, é: em que medida esse protótipo apresenta indícios iniciais de utilidade percebida e usabilidade junto aos perfis de usuários envolvidos no serviço?

Como resposta a esse problema, este trabalho desenvolve e avalia uma aplicação *web* para a gestão do transporte universitário intermunicipal. Suas funções abrangem o cadastro de viagens, motoristas e veículos pelas organizações de transporte; a consulta de viagens disponíveis e a reserva de lugares pelos estudantes; e o acompanhamento centralizado das listas de passageiros, da situação operacional dos pagamentos e da estimativa de arrecadação de cada viagem pelos responsáveis.

A plataforma foi concebida no modelo SaaS, em que uma única aplicação, acessada pela internet, atende a diversas organizações de forma independente, mantendo os dados de cada organização logicamente isolados dos dados das demais, abordagem conhecida como *multi-tenant*.

Este artigo tem como objetivo desenvolver e avaliar uma solução SaaS *multi-tenant* para a gestão do transporte universitário intermunicipal. De forma específica, o estudo buscou levantar e documentar requisitos a partir do funcionamento real do serviço, modelar o sistema e a sua base de dados, implementar um protótipo funcional, verificar tecnicamente as funcionalidades por meio de testes automatizados e implantação experimental e realizar uma avaliação exploratória de usabilidade com usuários.

Após esta introdução, a fundamentação teórica (Seção 2) parte dos trabalhos relacionados e reúne os conceitos necessários à compreensão da solução. Seguem-se o percurso metodológico (Seção 3), os resultados acompanhados da discussão (Seção 4) e as considerações finais com os trabalhos futuros (Seção 5).

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção reúne os fundamentos necessários para contextualizar o problema e orientar o desenvolvimento e a avaliação da solução proposta. Para isso, são apresentados os trabalhos relacionados, os conceitos de engenharia de *software* e modelagem, os princípios arquiteturais adotados, as características de aplicações SaaS *multi-tenant* e os conceitos empregados na avaliação exploratória de usabilidade.

2.1 TRABALHOS RELACIONADOS

Os trabalhos relacionados foram selecionados por meio de um levantamento bibliográfico exploratório e não sistemático, realizado a partir de pesquisas temáticas no Google Scholar e de consultas a repositórios e periódicos acadêmicos. Foram priorizados trabalhos com proximidade ao transporte universitário, ao transporte fretado, à gestão de frotas ou à informatização do transporte escolar. Por não se tratar de uma revisão sistemática, o conjunto analisado não pretende representar de forma exaustiva toda a produção científica da área, e as comparações apresentadas a seguir referem-se apenas aos estudos examinados.

Os estudos examinados registram problemas de fragmentação das informações, dependência de controles manuais e dificuldade de acesso a dados operacionais em diferentes modalidades de transporte coletivo. Nesse cenário, as soluções existentes concentram-se em partes específicas do problema, como fretamento eventual, gestão de frota institucional, modelagem do transporte escolar ou consulta de horários e rotas.

O PickBus, proposto por Pretto et al. (2021), é uma solução multiplataforma para gerenciamento de transporte fretado de passageiros, com foco em eventos e deslocamentos coletivos; o produto foi avaliado com 34 usuários parceiros, por meio de questionário *online*, e seus resultados evidenciam a carência de ferramentas digitais específicas para o setor. Paiva (2023) apresenta um sistema para modernizar a solicitação e o gerenciamento de transportes em uma instituição acadêmica, com cadastro de veículos, motoristas e setores, fluxo de aprovação e relatórios, verificado por testes internos durante o desenvolvimento, sem avaliação com usuários finais relatada. Já Reis et al. (2021) apresenta a modelagem de um *software* para o transporte escolar público municipal, com ênfase no levantamento de requisitos e na UML, evidenciando que a ausência de sistemas informatizados compromete a eficiência operacional e a tomada de decisão.

Bizerril (2024) analisa o aplicativo Intercampi, desenvolvido no contexto da UNILAB, com foco em facilitar o acesso a horários e rotas de ônibus universitários. O estudo destaca a substituição de tabelas impressas e arquivos PDF por uma solução móvel com acesso às informações de transporte, inclusive em situações sem conexão, e avalia a percepção dos usuários a partir de dados da Google Play e análise de sentimentos dos comentários. Embora se aproxime mais diretamente do domínio universitário, o Intercampi concentra-se na consulta de informações e na melhoria da previsibilidade do deslocamento, sem tratar reservas em viagens recorrentes, capacidade, gestão por múltiplas organizações ou controle operacional de pagamentos.

A lacuna identificada está na ausência, entre os trabalhos analisados, de uma solução que reúna o transporte universitário intermunicipal recorrente, a gestão por múltiplas organizações, as reservas com controle de capacidade e o acompanhamento operacional de pagamentos. A proposta deste estudo aborda esse recorte por meio de uma arquitetura SaaS *multi-tenant*; o preço registrado em cada inscrição permite estimar a arrecadação e apoiar a avaliação de viabilidade da viagem. A Seção 4.8 retoma essa comparação após a apresentação dos resultados.

2.2 ENGENHARIA DE SOFTWARE E MODELAGEM

Requisitos de *software* descrevem as funções, restrições e qualidades esperadas de um sistema. A literatura distingue requisitos funcionais, voltados ao comportamento do sistema, e requisitos não funcionais, associados a atributos como desempenho e segurança (Sommerville, 2011; Pressman; Maxim, 2016). Sua elicitação pode recorrer à observação do contexto de uso, a entrevistas e à análise de documentos e fluxos de trabalho, de modo a compreender as necessidades dos interessados antes do projeto da solução (Sommerville, 2011).

A modelagem emprega abstrações gráficas, como diagramas de casos de uso e de classes da *Unified Modeling Language* (UML), para representar atores, respon-

sabilidades e decisões de projeto. A verificação por testes executa o sistema sob condições controladas para revelar defeitos (Pressman; Maxim, 2016); no nível unitário, partes isoladas do código são exercitadas com apoio de padrões como *Arrange-Act-Assert* (AAA) e de *test doubles* (*mocks*).

No projeto de bancos de dados relacionais, o modelo entidade-relacionamento descreve conceitualmente entidades, atributos e relacionamentos (Heuser, 2009). No nível lógico, os dados são organizados em tabelas associadas por chaves primárias e estrangeiras (Elmasri; Navathe, 2018). Normalização, restrições de integridade e propriedades ACID contribuem para reduzir redundâncias e preservar a consistência. Esses fundamentos orientam o modelo de dados da Seção 4.1, inclusive o vínculo dos recursos sujeitos ao isolamento com a respectiva organização (Seção 2.4).

2.3 ARQUITETURA EM CAMADAS, DDD SELETIVO E REST

A organização em camadas separa responsabilidades de apresentação, aplicação, domínio e infraestrutura. Inspirada em princípios da *Clean Architecture*, essa divisão mantém as regras centrais de negócio independentes de detalhes de interface, persistência e *frameworks*, favorecendo testes e mudanças tecnológicas (Martin, 2017).

O *Domain-Driven Design* (DDD) foi considerado de forma seletiva, sobretudo por meio de entidades e *value objects*; seus padrões estratégicos não integram o escopo desta solução (Evans, 2003). Para a comunicação entre cliente e servidor, o *Representational State Transfer* (REST) estabelece restrições como separação cliente-servidor, ausência de estado de sessão no servidor e interface uniforme baseada em recursos manipulados por métodos HTTP (Fielding, 2000).

2.4 SAAS, MULTI-TENANCY E SEGURANÇA

Software as a Service (SaaS) é um modelo de entrega no qual a aplicação executa na infraestrutura do provedor e é disponibilizada pela rede, sob demanda (Mell; Grance, 2011). Em plataformas *multi-tenant*, uma mesma instância atende a múltiplas organizações, com isolamento lógico de dados. Esse isolamento pode ocorrer por bancos ou esquemas separados ou, como neste trabalho, pelo vínculo dos recursos à organização e pela filtragem aplicada nos mecanismos de autorização.

Em APIs REST sem estado de sessão, a segurança pode envolver *tokens* assinados no padrão *JSON Web Token* (JWT) (Jones; Bradley; Sakimura, 2015), senhas armazenadas como *hash* adaptativo com *salt* (Provos; Mazières, 1999) e controle de acesso baseado em papéis (RBAC) (National Institute of Standards and Technology, s.d.; Joint Task Force, 2020). As diretrizes do OWASP orientam a prevenção de falhas como *Broken Access Control* e referências diretas inseguras a objetos (IDOR) (Owasp Foundation, 2021); o tratamento de dados pessoais também deve observar

os princípios da Lei Geral de Proteção de Dados (Brasil, 2018).

2.5 AVALIAÇÃO DE USABILIDADE

Usabilidade é o grau em que usuários específicos alcançam objetivos com eficácia, eficiência e satisfação em determinado contexto de uso (ISO, 2018). Na abordagem de Nielsen, facilidade de aprendizado, eficiência, capacidade de memorização, erros e satisfação são atributos de usabilidade; as heurísticas, por sua vez, são princípios empregados na inspeção de interfaces, como visibilidade do estado do sistema, consistência e prevenção de erros (Nielsen, 1993). Atributos percebidos em um questionário e conformidade com heurísticas não são, portanto, medidas equivalentes. Em protótipos, avaliações exploratórias permitem identificar problemas iniciais de interação sem sustentar generalizações sobre toda a população de usuários.

Questionários pós-uso podem registrar a percepção dos participantes após tarefas representativas por meio de escalas de concordância do tipo Likert (Likert, 1932). O *System Usability Scale* (SUS) constitui um instrumento consolidado desse formato (Brooke, 1996), enquanto o *Technology Acceptance Model* (TAM) relaciona utilidade percebida e facilidade de uso percebida à intenção de adoção (Davis, 1989). Esses conceitos fundamentam a avaliação exploratória descrita na Seção 3.5, sem caracterizá-la como aplicação integral do SUS ou do TAM.

3 METODOLOGIA

Este trabalho caracteriza-se como desenvolvimento tecnológico aplicado, de caráter exploratório, conduzido no contexto delimitado do transporte universitário intermunicipal da rota Campo Maior-PI–Piripiri-PI. Embora parta de uma situação real específica, não foi adotado protocolo formal de estudo de caso. O percurso metodológico foi organizado em quatro etapas distintas: levantamento do contexto, desenvolvimento da solução, verificação técnica e avaliação com usuários. O Quadro 1 explicita como os procedimentos e as evidências de cada etapa respondem às duas questões de pesquisa; os artefatos e resultados correspondentes são apresentados na Seção 4.

3.1 ALINHAMENTO ENTRE QUESTÕES DE PESQUISA E PROCEDIMENTOS

Quadro 1 – Alinhamento entre questões de pesquisa, procedimentos, evidências e limitações.

Questão de pesquisa	Procedimentos	Artefatos ou dados produzidos	Forma de análise	Limitações
Questão arquitetural: como estruturar a solução SaaS <i>multi-tenant</i> ?	Observação do contexto e conversas informais; documentação dos requisitos; modelagem; implementação; testes unitários; implantação experimental.	Fluxos observados; requisitos; diagramas; código e protótipo; resultados dos testes; ambiente experimental.	Rastreabilidade entre problema, requisitos e artefatos; verificação das regras e dos fluxos no ambiente local e implantado.	Levantamento situado, sem protocolo formal de estudo de caso; testes concentrados no servidor e sem banco real, concorrência ou E2E; implantação não institucional.
Questão de usabilidade: quais os indícios iniciais de utilidade percebida e usabilidade?	Tarefas por perfil; questionário <i>online</i> ; itens em escala Likert; questões abertas.	Conclusão autodeclarada das tarefas; 19 conjuntos de itens fechados; 18 conjuntos de respostas abertas analisáveis.	Estatística descritiva dos itens fechados; leitura qualitativa exploratória das respostas abertas.	Amostra pequena, não probabilística e desequilibrada; tarefas não observadas; instrumento próprio e não padronizado.

Fonte: Elaborado pelo autor (2026).

3.2 LEVANTAMENTO DO CONTEXTO

A etapa inicial consistiu na observação direta/participante do serviço de transporte universitário na rota Campo Maior-PI–Piripiri-PI, utilizada regularmente pelo autor por aproximadamente três meses. Nesse período, observaram-se os fluxos práticos de reserva, confirmação de presença, pagamento, cancelamento e organização das viagens. A observação foi complementada por conversas informais, sem gravação, transcrição ou roteiro estruturado, com o prestador responsável pelo serviço em uma ou duas ocasiões e com cerca de quatro estudantes passageiros da rota.

Esse levantamento produziu uma descrição situada dos fluxos e problemas operacionais usada como entrada para a documentação dos requisitos. Seu propósito foi compreender o contexto investigado, sem representar estatisticamente o transporte universitário intermunicipal nem caracterizar um estudo de caso formal.

3.3 DESENVOLVIMENTO DA SOLUÇÃO

O desenvolvimento da solução foi organizado em três atividades complementares: levantamento e documentação dos requisitos, modelagem da solução e implementação do protótipo com definição das principais decisões tecnológicas. Essa

organização permitiu relacionar as necessidades observadas no serviço aos modelos produzidos e, posteriormente, aos componentes efetivamente implementados.

3.3.1 Levantamento e documentação dos requisitos

Os requisitos foram derivados da observação do contexto, das conversas informais, da análise do fluxo real do serviço e da documentação do projeto. A organização dos achados separou necessidades funcionais, ligadas às ações esperadas do sistema, e requisitos não funcionais, ligados a qualidades como segurança, manutenção, consistência e uso em dispositivos móveis.

3.3.2 Modelagem da solução

A modelagem foi elaborada a partir dos requisitos levantados e buscou transformar o funcionamento observado do serviço em uma estrutura de sistema. Foram utilizados diagramas UML de casos de uso para representar atores e funcionalidades, modelo de domínio para organizar as principais responsabilidades do sistema e um esquema lógico-relacional para orientar a estrutura do banco de dados. Essa etapa serviu como base para as decisões arquiteturais e para a implementação do protótipo.

3.3.3 Implementação e decisões tecnológicas

A implementação ocorreu de forma incremental a partir dos requisitos e modelos produzidos. As decisões tecnológicas consideraram as qualidades esperadas do protótipo, a experiência prévia do autor e a compatibilidade com a arquitetura definida. Não foi realizado *benchmarking* comparativo entre tecnologias. O Quadro 2 registra as justificativas efetivamente adotadas.

Quadro 2 – Justificativas das principais decisões tecnológicas do projeto.

Tecnologia ou decisão	Relação com o projeto	Justificativa
TypeScript	Manutenção e evolução (RNF06)	Tipagem estática e familiaridade do autor com a linguagem, favorecendo contratos explícitos entre módulos e cliente.
NestJS (NESTJS, 2024)	Padronização da API e manutenção (RNF01 e RNF06)	Organização modular e suporte à injeção de dependências, úteis para separar responsabilidades e isolar dependências nos testes unitários.
PostgreSQL	Integridade e persistência relacional (RNF08 e RNF09)	Restrições relacionais e suporte a transações nas operações críticas.
Prisma ORM (PRISMA, 2024)	Integridade, evolução do esquema e persistência (RNF08 e RNF09)	Definição declarativa do <i>schema</i> , migrações versionadas e integração tipada com TypeScript.
React, TanStack Start e TanStack Router (TANSTACK, 2024)	Interface responsiva e navegação (RNF10)	Construção da aplicação <i>web</i> responsiva e organização do roteamento por contextos de acesso.
Docker e Docker Compose	Ambiente de desenvolvimento	Repetibilidade da execução local dos serviços necessários ao desenvolvimento.
Cloudflare Workers, Render e Neon	Implantação experimental	Seleção pragmática orientada pela compatibilidade, respectivamente, com o <i>build</i> do cliente, a execução persistente da API e o PostgreSQL gerenciado, além da disponibilidade necessária ao experimento. Não houve <i>benchmarking</i> formal nem comparação sistemática entre provedores.

Fonte: Elaborado pelo autor (2026).

O uso de inteligência artificial ocorreu como desenvolvimento assistido, iterativo e supervisionado. No servidor, o apoio concentrou-se em estruturas repetitivas entre módulos, objetos de transferência de dados e testes unitários. No cliente, o Lovable apoiou a geração inicial de telas e componentes, o Claude Design auxiliou no refinamento visual e o Claude Code apoiou componentes, integração com a API e correções. Escopo, requisitos, arquitetura, regras de negócio e aceitação das alterações permaneceram sob decisão do autor.

A incorporação das sugestões não foi automática: o material produzido foi revisado no contexto do projeto e submetido aos procedimentos comprovadamente utilizados, como execução local, testes unitários automatizados no servidor, testes manuais dos fluxos principais no cliente e verificação funcional no ambiente experimental. O apoio automatizado também introduz riscos de duplicação de estruturas, inconsistência entre contratos ou regras e acúmulo de dívida técnica; por isso, revisão humana e verificação iterativa acompanharam a incorporação das alterações. Essas ferramentas foram tratadas como apoio ao desenvolvimento, e não como metodologia autônoma de programação.

3.4 VERIFICAÇÃO TÉCNICA

A verificação técnica foi conduzida separadamente da avaliação com usuários. No servidor, foram executados testes unitários automatizados com Jest e *ts-jest*, priorizando regras consideradas críticas, como autorização, inscrição, cancelamento, capacidade, controle operacional de pagamento e isolamento lógico. As dependências foram substituídas por *mocks* manuais, de modo que essa etapa verificou casos de uso e regras de domínio sem acionar banco real, servidor HTTP ou fluxos ponta a ponta.

Também foram realizados execução local, testes manuais dos fluxos principais no cliente e implantação experimental em nuvem. No ambiente implantado, verificaram-se a comunicação entre cliente, API e banco, a configuração das dependências externas e a execução funcional dos fluxos centrais. A topologia produzida e os resultados quantitativos dos testes são apresentados na Seção 4.6. Essa verificação não incluiu testes de integração com banco real, concorrência, auditoria de segurança, medição formal de disponibilidade ou testes E2E.

3.5 AVALIAÇÃO COM USUÁRIOS

Para obter indícios iniciais sobre a experiência de uso do protótipo, conduziu-se uma avaliação exploratória de usabilidade aplicada à versão implantada em nuvem (Seção 4.6). A avaliação concentrou-se na realização autodeclarada de tarefas representativas e na percepção registrada pelos respondentes quanto à facilidade de uso, à clareza das informações, à utilidade percebida e à satisfação geral. Não se pretendeu, nessa etapa, verificar tecnicamente a arquitetura ou a implementação da solução, nem utilizar o formulário para o levantamento de requisitos.

A coleta ocorreu entre 3 e 9 de julho de 2026, com amostra não probabilística, selecionada por conveniência. Foram recebidas 19 respostas ao questionário. Os respondentes foram orientados a acessar o protótipo por seus próprios dispositivos. A execução das tarefas não foi observada diretamente pelo pesquisador; sua conclusão foi autodeclarada no questionário.

Utilizou-se um questionário *online* e autoaplicado, estruturado em três blocos. O primeiro caracterizou o respondente quanto ao vínculo com o ensino superior, à experiência prévia com transporte intermunicipal para estudo e ao perfil utilizado durante a avaliação. O segundo apresentou um roteiro de tarefas específico para cada perfil, acompanhado de itens para autodeclaração de conclusão. O terceiro reuniu itens fechados de percepção e questões abertas. O instrumento não coletou nome nem endereço de e-mail.

Para o perfil Passageiro, as tarefas consistiram em localizar uma viagem disponível, consultar seus detalhes, realizar uma inscrição e posteriormente consultá-la

ou cancelá-la. Para o perfil Administrador, incluíram o acesso à área administrativa, a consulta ao painel da organização, a criação ou edição de uma viagem, a visualização dos passageiros inscritos, o cadastro ou a seleção de um veículo e a associação de motorista e veículo à viagem. Para o perfil Motorista, envolveram o acesso às viagens atribuídas, a consulta dos passageiros, a alteração do estado da viagem e a confirmação da presença e da situação de pagamento de cada passageiro.

Os itens fechados compreenderam oito afirmativas relacionadas à facilidade de uso, clareza das informações, simplicidade do fluxo, funcionamento no dispositivo, organização em comparação com listas manuais, intenção de uso, utilidade percebida e satisfação geral. As respostas foram registradas em escala Likert de cinco pontos (Likert, 1932), variando de “discordo totalmente” (1) a “concordo totalmente” (5). O instrumento foi elaborado especificamente para o escopo do protótipo e inspirado no formato de questionários pós-uso, como o SUS (Brooke, 1996), sem sua aplicação integral e sem cálculo de escore SUS. Os itens relacionados à utilidade percebida e à intenção de uso basearam-se em aspectos do TAM (Davis, 1989), sem aplicação formal do modelo. As questões abertas abordaram aspectos considerados úteis, dificuldades encontradas, erros percebidos, sugestões de melhoria e comparação com a organização realizada por WhatsApp e listas manuais. Não foi conduzida inspeção por especialistas nem avaliação pelas heurísticas de Nielsen; os itens registraram percepções sobre atributos do uso, utilidade e intenção, e não conformidade heurística.

A análise quantitativa considerou as 19 respostas e empregou estatística descritiva, com frequências, médias e percentuais de concordância. A análise qualitativa considerou 18 conjuntos de respostas abertas, examinados por leitura exploratória para identificar dificuldades recorrentes, aspectos valorizados nas respostas e sugestões de evolução do protótipo. Em uma das respostas, os campos abertos não respondiam adequadamente aos respectivos enunciados e foram tratados como dados qualitativos ausentes. Os itens fechados dessa resposta foram mantidos na análise quantitativa por apresentarem preenchimento completo e sem inconsistências internas (Seção 4.7).

4 RESULTADOS E DISCUSSÃO

Esta seção apresenta e discute os resultados obtidos nas etapas descritas na metodologia. Inicialmente, são examinadas a cobertura dos requisitos e a modelagem resultante; em seguida, são apresentadas a arquitetura, as regras operacionais, os mecanismos de isolamento e controle de acesso, a aplicação cliente, a implantação experimental, a verificação técnica e os resultados da avaliação exploratória de usabilidade.

4.1 COBERTURA DOS REQUISITOS E MODELAGEM RESULTANTE

O levantamento descrito na metodologia resultou em 26 requisitos funcionais (RF01–RF26) e 12 requisitos não funcionais (RNF01–RNF12). O Quadro 3 apresenta a rastreabilidade dos requisitos funcionais. Dos 26 requisitos, 18 foram implementados diretamente e três de forma simplificada, totalizando 21 requisitos com alguma materialização no protótipo; quatro permaneceram como evolução futura e um foi apenas modelado. A cobertura alcançada concentra-se no núcleo operacional investigado: cadastros, viagens, reservas, presença, capacidade e acompanhamento operacional de pagamentos. Essa contagem expressa cobertura funcional do protótipo, não adoção em operação real nem validação empresarial.

Quadro 3 – Rastreabilidade dos requisitos funcionais da solução proposta.

Código	Necessidade atendida	Situação no protótipo
RF01	Cadastrar passageiros e demais usuários com dados básicos de identificação.	Implementado
RF02	Cadastrar organizações de transporte e seus responsáveis administrativos.	Implementado
RF03	Permitir o cadastro do perfil de motorista.	Implementado
RF04	Agendar viagens com data, horário, turno, origem e destino.	Implementado
RF05	Cancelar viagens pela organização responsável.	Implementado
RF06	Classificar viagens por turno, como manhã, tarde ou noite.	Implementado
RF07	Permitir que passageiros façam inscrição em viagens disponíveis.	Implementado
RF08	Permitir cancelamento de inscrição dentro de prazo definido.	Implementado
RF09	Permitir escolha entre ida, volta ou ida e volta.	Implementado
RF10	Registrar a forma de pagamento informada pelo passageiro.	Implementado
RF11	Exibir a lista de passageiros inscritos em uma viagem.	Implementado
RF12	Confirmar presença no momento do embarque.	Implementado
RF13	Permitir que o motorista consulte passageiros da viagem.	Implementado
RF14	Permitir que o motorista marque a viagem como iniciada ou concluída.	Implementado
RF15	Gerar relatórios consolidados de viagens realizadas.	Evolução futura
RF16	Exibir histórico consolidado de viagens da organização.	Evolução futura
RF17	Enviar lembretes sobre inscrições e viagens próximas.	Evolução futura
RF18	Enviar avisos de confirmação ou cancelamento.	Evolução futura
RF19	Estimar a arrecadação operacional de cada viagem.	Implementado
RF20	Cadastrar veículos usados no transporte.	Implementado
RF21	Vincular motorista e veículo a uma viagem.	Implementado
RF22	Impedir inscrições acima da capacidade máxima definida.	Implementado

continua na próxima página

Quadro 3 – continuação

Código	Necessidade atendida	Situação no protótipo
RF23	Cadastrar pontos de embarque pela organização.	Implementado de forma simplificada
RF24	Permitir seleção de ponto de embarque na inscrição.	Implementado de forma simplificada
RF25	Exibir os pontos de embarque ao motorista.	Implementado de forma simplificada
RF26	Registrar ações relevantes em trilha de auditoria.	Modelado, não persistido

Fonte: Elaborado pelo autor (2026), a partir da documentação de requisitos.

Os requisitos não funcionais foram tratados como qualidades esperadas do protótipo, e não como resultados certificados. O Quadro 4 separa mecanismos implementados, evidências indiretas e atributos não medidos. Desempenho, disponibilidade e escalabilidade não foram submetidos a medição formal; segurança, manutenção e isolamento foram examinados apenas até o limite das estruturas implementadas, dos testes unitários e da verificação funcional descrita neste artigo.

Quadro 4 – Rastreabilidade dos requisitos não funcionais da solução proposta.

Código	Qualidade esperada	Tratamento no protótipo
RNF01	API organizada de forma padronizada e previsível.	Contratos REST/JSON e configuração de especificação OpenAPI.
RNF02	Respostas em até 2 s nas consultas principais.	Meta de projeto, sem medição formal.
RNF03	Disponibilidade mínima de 95%.	Meta de projeto, sem medição formal.
RNF04	Segurança e privacidade no acesso aos dados.	Evidência de implementação: autenticação, papéis e proteção de senhas; sem auditoria independente ou teste de invasão.
RNF05	Isolamento lógico entre organizações.	Evidência em filtros por organização, autorização e testes unitários; sem teste integrado com banco real ou validação independente.
RNF06	Facilidade de manutenção e evolução.	Separação modular e testes como evidências indiretas; sem métricas de manutenibilidade ou avaliação longitudinal.
RNF07	Comunicação por dados estruturados.	Atendido por contratos REST/JSON.

Quadro 4 – continuação

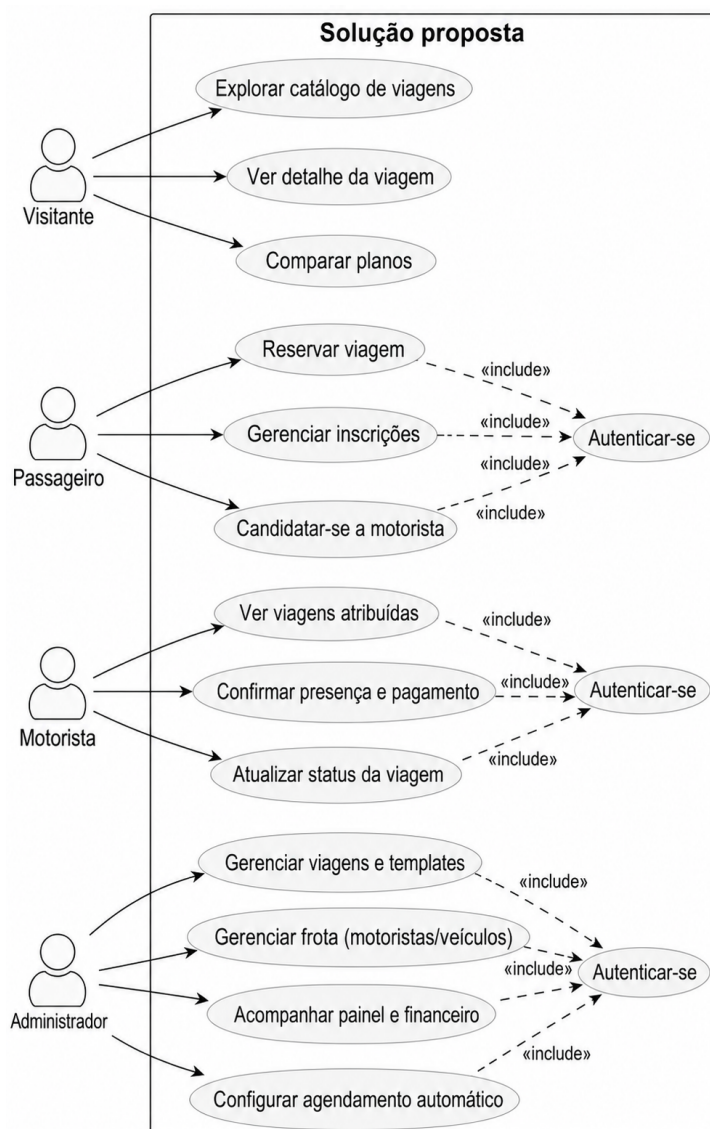
Código	Qualidade esperada	Tratamento no protótipo
RNF08	Integridade dos dados nas operações críticas.	Transações e validações de duplicidade implementadas; sem teste integrado de concorrência.
RNF09	Persistência em banco relacional confiável.	Atendido com PostgreSQL e migrações versionadas.
RNF10	Interface em português e adequada ao uso em celular.	Implementado na aplicação cliente responsiva <i>mobile-first</i> , em português.
RNF11	Continuidade da sessão do usuário autenticado.	Implementado por renovação automática de sessão; verificado de forma funcional.
RNF12	Consistência de horários exibidos ao usuário.	Verificado de forma funcional, por padronização do armazenamento e conversão para o horário local na exibição.

Fonte: Elaborado pelo autor (2026), a partir da documentação de requisitos.

A partir desses requisitos, a modelagem resultante definiu os atores, os principais casos de uso, o modelo de domínio e a estrutura relacional dos dados. Com exceção do Visitante, único ator não autenticado, que explora o catálogo público de viagens e organizações, os atores representam diferentes contextos de atuação de usuários autenticados e não correspondem diretamente aos papéis do controle de acesso. Todo usuário autenticado é uma instância da entidade base *User*. Passageiro é a denominação funcional atribuída ao *User* comum que consulta viagens, realiza inscrições e gerencia suas reservas; não se trata de um papel formal, pois a enumeração de papéis do controle de acesso contempla apenas *ADMIN* e *DRIVER*. O perfil de Motorista corresponde a um *User* que possui também um registro relacionado na entidade *Driver*, com informações adicionais necessárias à operação das viagens (como número e categorias da CNH), e que atua sob o papel *DRIVER* no âmbito de uma organização. O Administrador é identificado por seu vínculo administrativo com uma organização, materializado por um registro de participação (*OrganizationMembership*) com o papel *ADMIN*, a partir do qual gerencia rotas, viagens, motoristas, veículos, plano e métricas dessa organização. Embora não constituam uma hierarquia de papéis, os atores expressam uma progressão de responsabilidade, do acesso público do Visitante à gestão completa do Administrador.

A Figura 1 consolida essa visão nos casos de uso essenciais do sistema, agrupados por ator. Sua função no texto é dupla: tornar visível a progressão de responsabilidade entre os atores, do acesso público do Visitante à gestão completa do Administrador, e explicitar a fronteira de autenticação que separa o Visitante dos demais contextos de atuação.

Figura 1 – Diagrama de casos de uso da solução proposta.



Fonte: Elaborado pelo autor (2026).

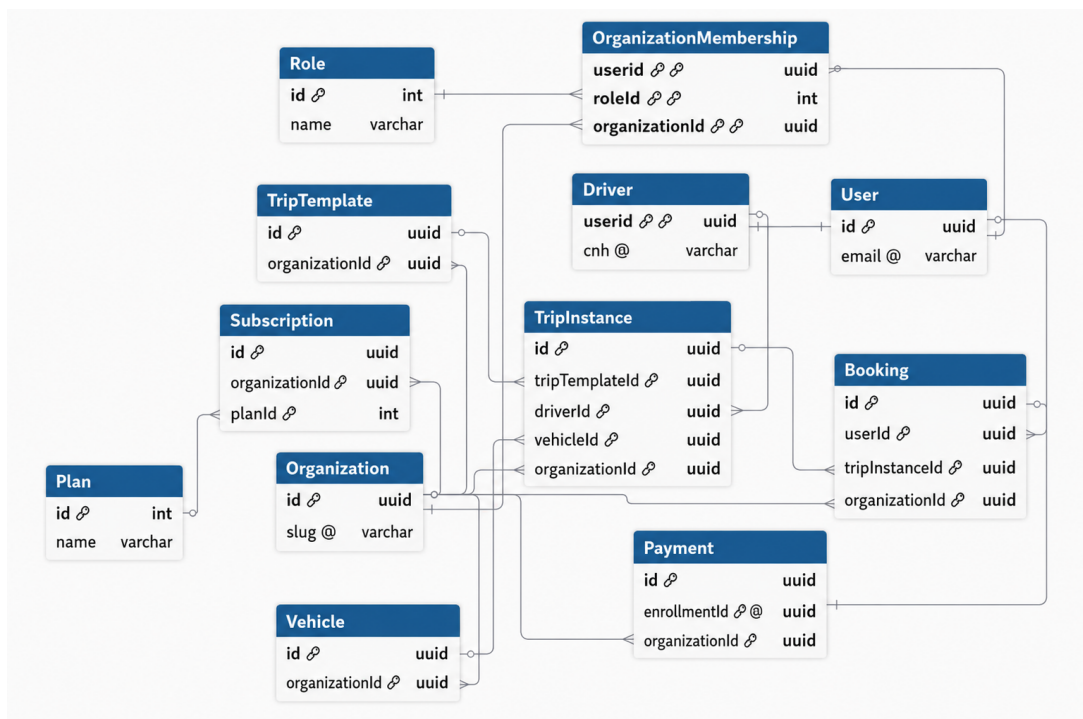
No diagrama, os casos de uso do Visitante (explorar o catálogo, ver detalhes e comparar planos) dispensam autenticação, decisão que materializa o catálogo público de viagens como porta de entrada do sistema; já as operações do Passageiro incluem obrigatoriamente a autenticação (relação *include*). A autenticação constitui a fronteira que separa o Visitante dos usuários autenticados: as operações de Passageiro exigem apenas um *User* autenticado, enquanto as verificações de papel do RBAC (ADMIN e DRIVER), detalhadas na Seção 4.4, incidem sobre os casos de uso de operação e de gestão. Motorista e Administrador concentram, respectivamente, os casos de uso operacionais (viagens atribuídas, presença e pagamento) e os de gestão (viagens, frota, painel de pagamentos e agendamento automático).

O modelo de domínio incorpora conceitos de modelagem orientada ao domínio, especialmente entidades, *value objects* e a organização das regras em casos de uso,

ao estruturar regras e dados em torno dos conceitos centrais do problema. As catorze entidades foram agrupadas em cinco responsabilidades: identidade e acesso; frota; viagens; reservas e pagamentos; e planos e assinaturas. Essa divisão aproxima o sistema da rotina observada no transporte: quem usa, quem administra, quais veículos e motoristas operam, como as viagens são criadas, como as reservas são controladas e como o pagamento é acompanhado de forma operacional. O cliente *web* não define regras próprias de negócio; ele apresenta dados, coleta ações do usuário e delega as decisões à API.

O modelo de dados foi elaborado segundo a abordagem de projeto conceitual e lógico descrita na Seção 2.2. A Figura 2 apresenta um esquema relacional simplificado derivado do modelo implementado no Prisma. A representação seleciona apenas as tabelas correspondentes às entidades centrais e explicita atributos, tipos, chaves e relacionamentos relevantes para compreender a estrutura lógica do banco e o desenho *multi-tenant*.

Figura 2 – Esquema relacional simplificado da solução proposta.



Fonte: Elaborado pelo autor (2026).

Nota: o símbolo de chave identifica uma chave primária (PK), e o símbolo de elo identifica uma chave estrangeira (FK). A presença conjunta dos dois símbolos indica uma FK que também participa de uma chave primária composta. O símbolo @ indica uma restrição de unicidade (UNIQUE); os conectores representam relacionamentos e suas cardinalidades.

Entre as relações selecionadas para o esquema, destacam-se o identificador da organização, usado para separar logicamente os dados de cada *tenant* (Seção 2.4), e a distinção entre a configuração recorrente de uma rota e as viagens realizadas em datas específicas, da qual derivam as regras de agendamento, capacidade e cancela-

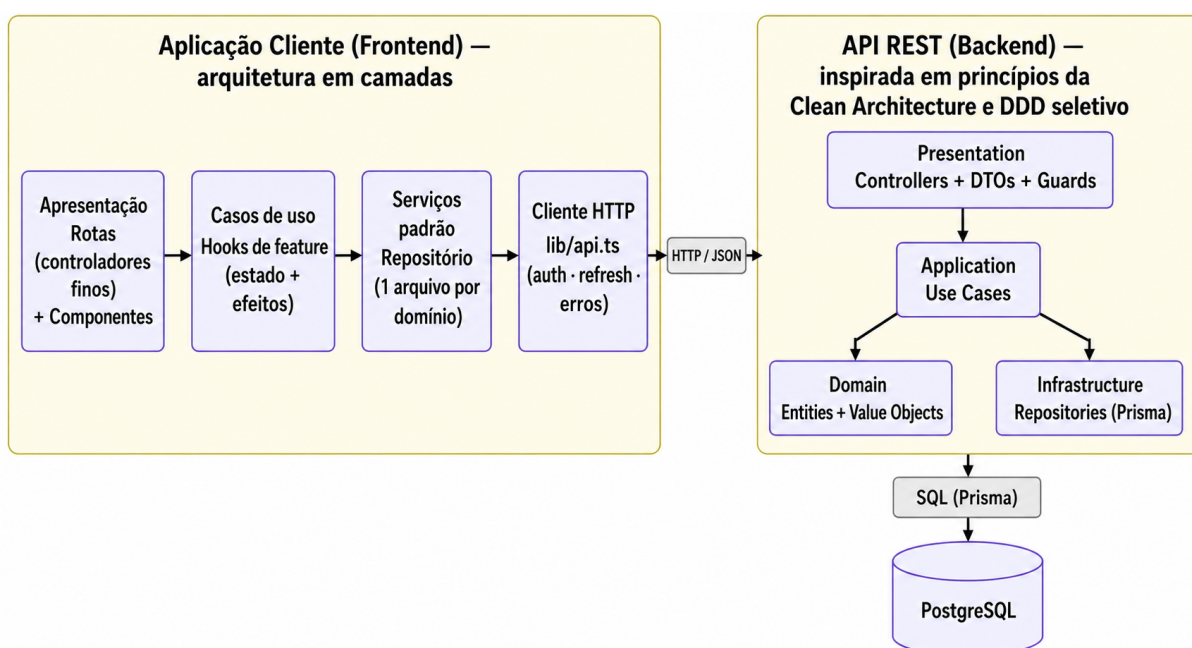
mento (Seção 4.3). O registro de pagamento vincula-se a exatamente uma inscrição (entidade `Enrollment`) por meio da chave estrangeira `enrollmentId` sob restrição de unicidade, o que caracteriza uma relação um-para-um; o esquema não define uma entidade de reserva distinta, de modo que a própria inscrição cumpre esse papel.

4.2 ARQUITETURA E COBERTURA DOS FLUXOS

Em alto nível, o resultado arquitetural compõe-se de três elementos: aplicação cliente, API REST e banco de dados relacional. O cliente comunica-se com a API por HTTP/JSON, e somente a API acessa o banco. Essa fronteira mantém no servidor as decisões sobre autorização, disponibilidade, preço, capacidade, cancelamento e pagamento, evitando que diferentes interfaces definam regras divergentes.

A Figura 3 representa o caminho de uma requisição da aplicação cliente até a persistência. A separação entre entrada, casos de uso, regras de domínio e acesso aos dados materializa a arquitetura em camadas descrita na fundamentação (Martin, 2017). Trata-se de evidência da estrutura implementada e de sua correspondência com os requisitos, não de medição de manutenibilidade ou escalabilidade.

Figura 3 – Visão geral da aplicação cliente, da API REST em camadas e da persistência.



Fonte: Elaborado pelo autor (2026).

A separação de responsabilidades também foi aplicada ao cliente, que concentra apresentação e interação e delega decisões de negócio à API. Essa organização favorece a localização das responsabilidades no código, mas seu efeito sobre esforço de mudança, defeitos ou tempo de manutenção não foi medido.

4.3 REGRAS DE RESERVA, CAPACIDADE E PAGAMENTO

A cobertura funcional foi materializada em operações públicas, de passageiro, de motorista e de administração. O catálogo de viagens e organizações pode ser consultado sem autenticação, enquanto reservas e operações de gestão exigem identidade e papel compatíveis. A API expõe 93 *endpoints* REST, número mantido como descrição do escopo implementado, não como contribuição do estudo.

As regras centrais aparecem no ciclo da viagem e da reserva. Rotas recorrentes são separadas das viagens em datas específicas; a ativação depende da atribuição de motorista e veículo; e rotinas automáticas podem gerar novas viagens e cancelar as que vencem sem atingir a receita mínima. Ao solicitar uma reserva, a API verifica disponibilidade e capacidade, calcula o preço no servidor e aplica o prazo de cancelamento de 30 minutos antes da partida. Essas regras cobrem diretamente RF04–RF10, RF19, RF21 e RF22.

A criação da reserva e do respectivo registro operacional de pagamento é executada em uma única transação, coordenada por uma camada de *Unit of Work*: a reserva e o pagamento (registrado com situação pendente) são confirmados ou revertidos em conjunto, de modo que a falha de uma escrita descarta a outra. Nas operações críticas, adotou-se o nível de isolamento *Serializable*, e a verificação de capacidade e de duplicidade ocorre dentro da própria transação: a contagem de reservas ativas da viagem é comparada à sua capacidade máxima antes da gravação, e a duplicidade é barrada tanto por consulta prévia quanto por restrição de unicidade no banco. Como transações *Serializable* concorrentes podem disputar a última vaga e falhar por conflito de serialização, esses conflitos são tratados por repetição automática: a transação é reexecutada em até três tentativas ao detectar o erro correspondente do Prisma (P2034), sem espera progressiva entre as tentativas. A propagação do cliente transacional para os repositórios apoia-se em *AsyncLocalStorage*, o que mantém as escritas de uma mesma operação vinculadas à transação em curso. O pagamento registra método, valor e situação para acompanhamento operacional e estimativa de arrecadação, mas não movimenta recursos financeiros nem se integra a *gateway* externo. Esses mecanismos procuram reduzir inconsistências em reservas simultâneas; contudo, sua eficácia sob concorrência real não foi avaliada por testes integrados com banco real neste estudo, de modo que o trabalho não afirma a prevenção definitiva de *overbooking*.

4.4 ISOLAMENTO ENTRE ORGANIZAÇÕES E CONTROLE DE ACESSO

O modelo de identidade parte de uma entidade base, *User*, que representa qualquer usuário autenticado. Sobre ela, os papéis do controle de acesso restringem-se a *ADMIN* e *DRIVER* e são atribuídos a um usuário no âmbito de uma organização

por meio de um registro de participação (`OrganizationMembership`) que associa `User`, `Role` e `Organization`. Não há um papel de passageiro: o usuário comum que consulta viagens e gerencia inscrições é um `User` autenticado sem papel administrativo ou de motorista, ao passo que o perfil de motorista combina o papel `DRIVER` a um registro relacionado em `Driver`, que reúne os dados específicos da atuação. Sobre esse modelo, o controle de acesso combina autenticação por JWT, verificação de papéis e conferência da organização associada à requisição. Cada recurso sujeito ao escopo organizacional mantém o identificador correspondente, e os casos de uso e consultas aplicam a organização obtida do contexto autenticado. Operações sobre recursos específicos também verificam sua pertença à organização solicitante. No cliente, a restrição de telas orienta a navegação, mas não substitui a autorização aplicada pela API. Quanto às credenciais, as senhas são persistidas apenas na forma de *hashes* gerados com *bcrypt*, com fator de custo 10, sem armazenamento do valor original; durante a autenticação, a senha informada é comparada ao *hash* persistido pela própria função de verificação da biblioteca.

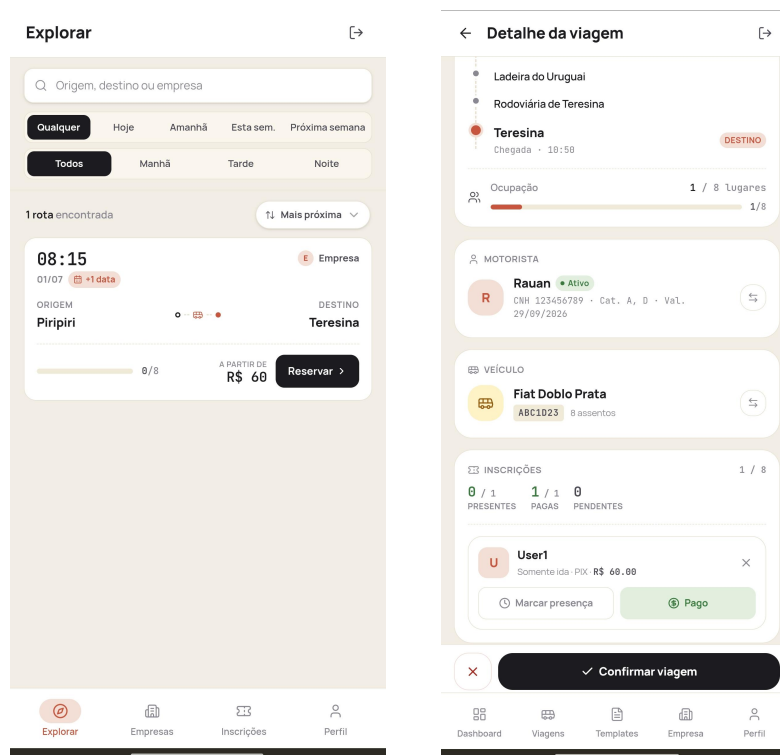
Esses mecanismos e os testes unitários de autorização constituem a evidência disponível de isolamento lógico. Não houve ensaio integrado de acesso cruzado entre organizações com banco real, auditoria independente ou teste de invasão. O banco preserva relações e restrições, mas não aplica política nativa por organização, como *Row Level Security*. Portanto, o artigo registra implementação de mecanismos de isolamento, e não medição conclusiva de sua eficácia em ambiente operacional.

4.5 MATERIALIZAÇÃO DOS REQUISITOS NA APLICAÇÃO CLIENTE

A aplicação cliente materializa os requisitos em uma interface *web* responsiva organizada por contexto de acesso. O fluxo público permite localizar e consultar viagens; o passageiro realiza e acompanha reservas; o administrador gerencia viagens, frota, agendamento e indicadores; e o motorista consulta viagens atribuídas, passageiros, presença e situação do pagamento. A interface apresenta e coleta informações, enquanto disponibilidade, preço, capacidade e criação do registro operacional de pagamento permanecem sob decisão da API. No cliente, a interface reutiliza componentes e mantém padrões visuais consistentes entre as telas; esses padrões, contudo, ainda não estão formalizados, e sua consolidação e padronização permanecem como evolução futura.

A Figura 4 ilustra dois resultados diretamente ligados aos requisitos: o catálogo público para localização de viagens e a tela operacional para confirmação de presença e acompanhamento da situação de pagamento por passageiro. A disposição responsiva prioriza o uso em telas de celular durante a consulta e o embarque.

Figura 4 – Catálogo público de viagens (à esquerda) e detalhe operacional da viagem (à direita).



Fonte: Elaborado pelo autor (2026).

Complementarmente, a Figura 5 apresenta o painel administrativo como resultado da integração entre requisitos e regras. A arrecadação estimada, com valores confirmados e pendentes, materializa RF19; os indicadores de inscritos e vagas refletem o controle de capacidade de RF22; e o alerta de viagem sem inscritos antecipa o risco de cancelamento por receita mínima. O painel consolida informações antes dispersas, embora sua utilidade para decisões empresariais não tenha sido validada em operação real.

Figura 5 – Painel administrativo da organização, com indicadores operacionais e de pagamento.



Fonte: Elaborado pelo autor (2026).

4.6 AMBIENTES DE EXECUÇÃO, IMPLANTAÇÃO EXPERIMENTAL E VERIFICAÇÃO

No ambiente local, a aplicação cliente era executada pelo servidor de desenvolvimento do Vite, enquanto a API NestJS e o PostgreSQL 17 podiam ser executados por Docker Compose. Esse ambiente apoiou a implementação, a execução dos testes unitários do servidor e os testes manuais dos fluxos do cliente. Ele permaneceu separado da implantação em nuvem e não constituiu ambiente de produção.

A implantação experimental em nuvem resultou em um ambiente *online* composto por aplicação cliente na Cloudflare Workers, API REST no Render e PostgreSQL gerenciado no Neon. Sua finalidade foi conferir a comunicação entre os artefatos, a configuração das dependências externas e a execução dos fluxos principais fora do ambiente local. A seleção dos três serviços foi pragmática, baseada na compatibilidade com os artefatos existentes e na disponibilidade necessária à verificação e à

avaliação exploratória; não houve *benchmarking* formal nem comparação sistemática de desempenho, escalabilidade ou custo entre provedores.

Quadro 5 – Classificação dos componentes da implantação experimental.

Componente	Configuração encontrada	Classificação	Papel na implantação
Aplicação cliente	Cloudflare Workers, com entrada de servidor do TanStack Start configurada pelo Wrangler	Aplicação <i>web</i> com execução de borda em modelo <i>serverless</i>	Entrega da interface e processamento previsto pelo <i>build</i> do cliente.
API REST	Render Web Service, executando o JavaScript compilado da aplicação NestJS como processo HTTP	Serviço Node.js persistente; não <i>serverless</i>	Autenticação, autorização, casos de uso, regras de negócio e acesso aos dados.
Banco de dados	PostgreSQL gerenciado no Neon; conexão direta para migrações e conexão por meio de <i>pool</i> de conexões no tempo de execução	Banco relacional gerenciado e persistente	Estado compartilhado da aplicação e integridade relacional.
Rotinas agendadas	<i>jobs</i> do @nestjs/schedule executados dentro do processo da API	Processamento interno à API; não são <i>workers</i> independentes	Geração recorrente e cancelamento automático de viagens.

Fonte: Elaborado pelo autor (2026), com base nas configurações dos projetos.

A passagem para a nuvem exigiu a geração do Prisma Client durante o *build*, a execução do servidor pelo JavaScript compilado, a configuração da URL pública da API no cliente, o uso do *pool* de conexões do Neon no servidor e o controle de CORS por origem autorizada. No ambiente público, a geração da especificação OpenAPI e a exposição da interface Swagger UI foram desabilitadas por configuração; a autorização efetiva permaneceu sob responsabilidade da API. O ambiente serviu também de base para a avaliação exploratória de usabilidade (Seção 4.7), permitindo o acesso pelos dispositivos dos respondentes. Não houve, no escopo deste estudo, uma implantação separada em produção real: a disponibilidade pública do protótipo, assim como o uso de comandos de construção e execução destinados à produção, não altera sua classificação como ambiente experimental, sem operação institucional definitiva.

Não foram realizados testes de carga, de estresse ou de longa duração, nem medições formais de vazão, latência, elasticidade ou limites de concorrência. Portanto, a implantação não demonstra escalabilidade. Como características favoráveis a uma evolução futura, observam-se a separação entre cliente, API e banco e o atendimento

HTTP sem estado (*stateless*) quanto à sessão do usuário, pois a autenticação utiliza JWT e o estado compartilhado permanece no banco. Essas propriedades reduzem o acoplamento entre os componentes e podem facilitar a replicação do atendimento HTTP, mas constituem apenas potencial arquitetural, não evidência de capacidade.

Os limites também são relevantes. O PostgreSQL permanece como recurso central compartilhado; cada processo da API cria seu próprio *pool* de conexões por meio do adaptador PostgreSQL do Prisma, sem tamanho explicitamente configurado no projeto, de modo que novas réplicas podem aumentar a pressão sobre o limite de conexões do banco. Os *jobs* periódicos executam no mesmo serviço da API e usam proteção apenas dentro de cada processo, sem bloqueio distribuído, o que exige coordenação adicional caso existam múltiplas réplicas. A capacidade efetiva do Web Service da API, o comportamento sob requisições concorrentes e a atomicidade das reservas em situação de disputa não foram medidos. Assim, embora o cliente utilize execução de borda em modelo *serverless*, a solução completa não é integralmente *serverless*: ela inclui uma API persistente, um banco relacional com estado e rotinas agendadas acopladas ao processo da API.

A verificação técnica do servidor executou 458 testes unitários distribuídos em 57 suítes. A seleção priorizou autorização, reserva, cancelamento, capacidade, controle operacional de pagamento e isolamento na camada de aplicação, relacionando os testes às regras centrais do problema. Dependências externas foram substituídas por *mocks* manuais; assim, o resultado demonstra a execução esperada dos casos isolados testados, mas não o comportamento com banco real, migrações, requisições concorrentes ou fluxos ponta a ponta. No cliente, a verificação permaneceu predominantemente manual, tanto no ambiente local quanto na implantação experimental. Permanecem necessários testes integrados para autenticação, acesso cruzado entre organizações, criação simultânea de reservas, registro de pagamento e limite de capacidade, além de testes E2E e testes automatizados do cliente.

4.7 RESULTADOS DA AVALIAÇÃO EXPLORATÓRIA DE USABILIDADE

A coleta ocorreu entre 3 e 9 de julho de 2026, por amostragem não probabilística, por conveniência. Foram recebidas 19 respostas ao questionário. Em uma delas, as respostas abertas foram desconsideradas por apresentarem conteúdo incompatível com as perguntas formuladas, enquanto os itens fechados, considerados coerentes, foram mantidos. Dessa forma, a análise quantitativa considerou as 19 respostas, e a análise qualitativa considerou 18 conjuntos de respostas abertas.

As respostas ficaram assim distribuídas: catorze relativas ao perfil Passageiro, quatro ao perfil Administrador e uma ao perfil Motorista. Treze respostas indicaram vínculo com o ensino superior, e catorze registraram experiência prévia com transporte intermunicipal para estudar, o que aproxima os dados do público-alvo, mas torna

os resultados mais representativos do fluxo Passageiro. As percepções dos perfis Administrador (quatro respostas) e Motorista (uma resposta) devem ser interpretadas apenas como sinais iniciais, devido ao número reduzido de respostas desses grupos. Os registros de execução do roteiro correspondem à conclusão declarada em cada resposta e não resultam de observação direta pelo pesquisador.

A Tabela 1 apresenta os oito itens de percepção do instrumento, identificados de I1 a I8, com a média das respostas e o percentual de concordância (soma das respostas “concordo parcialmente” e “concordo totalmente”). Ela permite localizar, de um lado, os itens em que a percepção foi unânime e, de outro, aqueles que concentraram as ressalvas dos participantes.

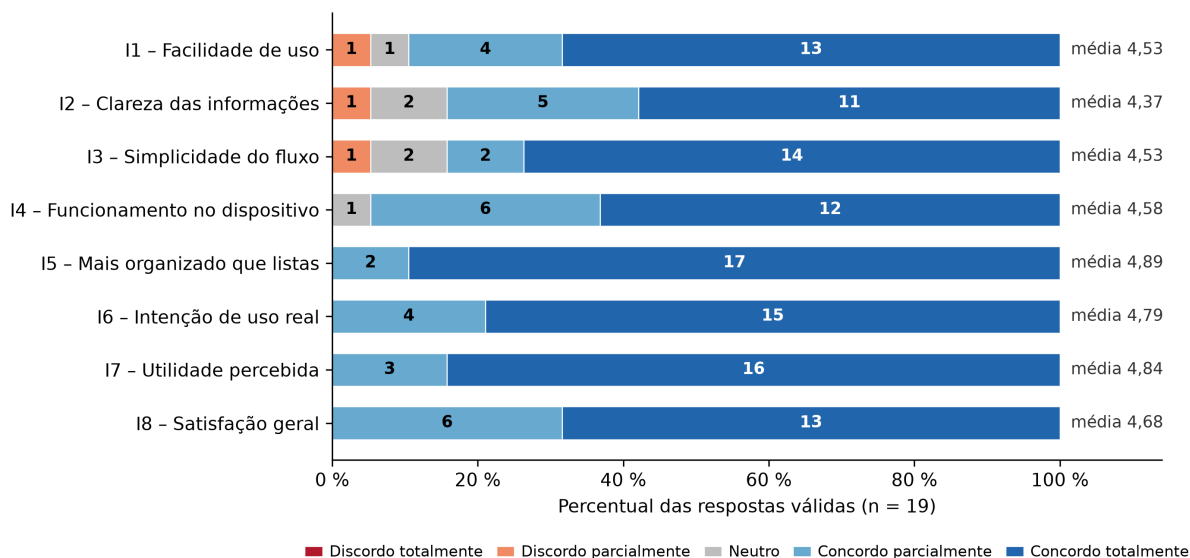
Tabela 1 – Itens de percepção sobre uso, utilidade, intenção e satisfação: média e concordância (n = 19).

Item	Afirmativa avaliada	Média (1–5)	Concordância
I1	O sistema foi fácil de usar	4,53	89,5%
I2	As informações apresentadas estavam claras	4,37	84,2%
I3	O fluxo testado foi simples de entender	4,53	84,2%
I4	A interface funcionou bem no dispositivo utilizado	4,58	94,7%
I5	O sistema parece mais organizado do que listas em WhatsApp ou controles manuais	4,89	100%
I6	Eu usaria uma solução como essa em um cenário real	4,79	100%
I7	O sistema parece útil para gestão de transporte universitário	4,84	100%
I8	Minha satisfação geral com o sistema foi boa	4,68	100%

Fonte: Elaborado pelo autor (2026), a partir das respostas ao questionário.

Como se observa na Tabela 1, todos os itens registraram média igual ou superior a 4,37, com mediana 5. A média simples entre os oito itens foi de 4,65 e serve apenas como síntese descritiva, pois o conjunto não constitui escala validada. I1–I4 registram percepções sobre facilidade, clareza, fluxo e funcionamento no dispositivo; I8 registra satisfação geral. I5–I7 tratam, respectivamente, de organização em comparação com listas manuais, intenção declarada de uso e utilidade percebida, construtos relacionados, mas distintos dos atributos de usabilidade. I5–I8 alcançaram 100% de concordância, e 17 das 19 respostas marcaram concordância total em I5. Esses resultados não representam avaliação pelas heurísticas de Nielsen. O Gráfico 1 detalha a distribuição em cada ponto da escala.

Gráfico 1 – Distribuição das respostas aos itens de percepção (I1–I8), em escala Likert de cinco pontos.



Fonte: Elaborado pelo autor (2026), a partir das respostas ao questionário.

O Gráfico 1 mostra ausência de discordância em I4–I8, com um único registro neutro em I4. As respostas menos favoráveis concentram-se em I1–I3: uma resposta do perfil Administrador registrou discordância parcial nos três itens, enquanto os registros neutros apareceram em outra resposta desse perfil e na resposta do perfil Motorista. As respostas relativas ao perfil Passageiro foram positivas nesses itens. A distribuição aponta onde surgiram ressalvas de interação, mas o número reduzido de respostas de Administrador e Motorista e a ausência de observação das tarefas impedem atribuir essas diferenças à complexidade das áreas ou generalizá-las aos perfis.

As respostas abertas reforçam essa leitura. Entre os aspectos mais úteis, os participantes destacaram a centralização das informações e a organização das viagens, a facilidade de reservar e cancelar, os filtros de busca do catálogo, o registro de formas de pagamento e a automação do agendamento recorrente, apontada por um participante do perfil administrativo como forma de evitar tarefas repetitivas. Entre as dificuldades, concentradas na área administrativa, foram citadas a quantidade de campos e informações em tela, a navegação pouco intuitiva dessa área, mensagens de erro genéricas (do tipo “erro ao realizar”, sem indicar causa ou correção), a ausência de um atalho para retornar à tela inicial, a possibilidade de confusão entre os perfis de passageiro e motorista e a nomenclatura das paradas, interpretada por participantes do perfil administrativo como pontos de descanso ao longo do trajeto, com a sugestão de distingui-las explicitamente como paradas de embarque e de desembarque. Nas respostas abertas analisadas, não foram relatados defeitos funcionais que impedis-

sem a conclusão declarada das tarefas; o único comportamento indesejado registrado refere-se à especificidade das mensagens de validação.

As sugestões de melhoria convergem, em parte, com requisitos já mapeados como evolução futura: notificações de confirmação e lembrete antes da partida correspondem aos requisitos RF17–RF18 (Quadro 3), e o pedido de painéis mais ricos remete aos relatórios consolidados (RF15–RF16). Surgiram ainda sugestões novas, incorporadas ao roteiro de evolução: ampliação do prazo de cancelamento, verificação de identidade de motoristas e organizações como mecanismo de confiança, versão otimizada para *desktop* na área administrativa, personalização de tema e ofertas de última hora para vagas liberadas por desistência. A pergunta de comparação direta com WhatsApp e listas manuais recebeu 15 preenchimentos; um não respondia ao enunciado e foi tratado como dado qualitativo ausente. Nas 14 respostas analisáveis, todas expressaram avaliação favorável à solução, principalmente por organização, centralização e rapidez. Essa unanimidade descreve somente as respostas analisadas e não representa comparação observacional de tempo, erro ou produtividade.

Cumpra delimitar o alcance desses resultados. A amostra é pequena, não probabilística, selecionada por conveniência e desequilibrada entre perfis, com catorze respostas relativas ao perfil Passageiro, quatro ao perfil Administrador e uma ao perfil Motorista. A execução das tarefas foi autodeclarada e não observada diretamente. O instrumento é próprio e não padronizado; o SUS não foi aplicado integralmente nem teve seu escore calculado (Brooke, 1996), e não houve inspeção heurística. Os resultados devem ser lidos separadamente como percepções exploratórias sobre uso, satisfação, utilidade e intenção declarada, sem generalização para os três perfis e sem equivalência a um escore padronizado.

4.8 DISCUSSÃO E COMPARAÇÃO COM TRABALHOS RELACIONADOS

Os resultados respondem à questão arquitetural pela rastreabilidade entre requisitos, regras e artefatos: 21 dos 26 requisitos funcionais tiveram implementação direta ou simplificada, incluindo reserva, capacidade, presença, pagamento operacional e estimativa de arrecadação. A verificação técnica acrescentou evidência unitária sobre as regras priorizadas, e a implantação experimental demonstrou a comunicação funcional entre cliente, API e banco fora do ambiente local. A contribuição discutida não está na quantidade de bibliotecas, módulos ou *endpoints*, mas na integração dessas funções no recorte investigado. O Quadro 6 situa esse resultado diante dos estudos examinados.

Quadro 6 – Comparação entre a solução proposta e os trabalhos relacionados.

Critério	PickBus (Pretto et al., 2021)	Paiva (Paiva, 2023)	Intercampi (Bizerril, 2024)	Reis et al. (Reis et al., 2021)	Solução proposta
Domínio de aplicação	Fretamento eventual de passageiros	Transporte institucional acadêmico	Ônibus universitários intercampi	Transporte escolar público municipal	Transporte universitário intermunicipal
Natureza do trabalho	Implementação multiplataforma	Implementação de sistema de gestão	Aplicativo móvel e análise de uso	Modelagem (requisitos e UML)	Implementação (API REST e cliente <i>web</i>)
Modelo SaaS <i>multi-tenant</i>	Não relatado	Não relatado	Não relatado	Não relatado	Sim
Controle operacional de pagamento	Não relatado	Não relatado	Não relatado	Não relatado	Sim (situação, valor por inscrição e estimativa de arrecadação)
Avaliação de viabilidade da viagem	Não relatado	Não relatado	Não relatado	Não relatado	Sim (receita mínima e auto-cancelamento)
Testes automatizados relatados	Não relatado	Não relatado	Não relatado	Não relatado	458 testes unitários
Avaliação com usuários	Sim (34 usuários parceiros)	Testes internos, sem usuários finais	Sim (notas e comentários da Google Play)	Não relatado	Sim (exploratória; 19 respostas consideradas na análise quantitativa)
Implantação relatada	Não relatado	Não relatado	Sim (Android/Google Play)	Não relatado	Experimental, em nuvem

Fonte: Elaborado pelo autor (2026), com base nos trabalhos citados.

Como mostra o Quadro 6, a distinção da solução proposta, no conjunto analisado, está no recorte do deslocamento intermunicipal recorrente e na integração entre reserva, capacidade, operação da viagem e acompanhamento de pagamentos. A receita mínima e o cancelamento automático relacionam ocupação e viabilidade operacional. A modelagem de Reis et al. (2021) concentra-se em requisitos e diagramas, sem relatar implantação ou verificação técnica, enquanto este estudo combina desenvolvimento prototipal, testes unitários, implantação experimental e avaliação exploratória. A comparação não permite afirmar superioridade geral sobre as demais soluções, pois os estudos têm objetivos, contextos e métodos de avaliação distintos.

Quanto à avaliação, o PickBus relata participação de usuários parceiros, enquanto o Intercampi analisa notas e comentários públicos da Google Play. Neste estudo, as tarefas foram autodeclaradas e as respostas concentraram-se no perfil Passa-

geiro; por isso, os resultados representam indícios exploratórios, não equivalentes aos procedimentos dos outros trabalhos. Também é necessário separar os achados: I1–I4 e I8 registram percepções ligadas ao uso e à satisfação, enquanto I5–I7 tratam de organização percebida, intenção e utilidade. Nenhum desses resultados corresponde a inspeção pelas heurísticas de Nielsen.

Escalabilidade, segurança, manutenção e isolamento não foram medidos além das evidências apresentadas. Para escalabilidade, existem separação dos componentes e atendimento HTTP sem estado, mas não houve teste de carga. Para segurança, foram implementados autenticação, papéis e filtros por organização, sem auditoria ou teste de invasão. Para manutenção, a separação modular e os testes são evidências indiretas, sem métricas ou acompanhamento longitudinal. Para isolamento, há verificações na camada de aplicação e testes unitários, mas não testes integrados de acesso cruzado com banco real. O pagamento permanece simulado e sem movimentação financeira.

Também não houve validação empresarial. Nenhuma organização adotou o protótipo em operação regular, e não foi conduzido piloto com viagens reais, análise de custos, adequação aos processos de uma empresa ou avaliação de viabilidade do negócio. As quatro respostas relativas ao perfil Administrador registram percepções individuais sobre o uso e não substituem evidência de adoção organizacional. Essas limitações mantêm a solução no estágio de protótipo funcional e orientam testes integrados, avaliação em contexto operacional e validação empresarial como trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

Este trabalho desenvolveu e avaliou uma solução SaaS *multi-tenant* para a gestão do transporte universitário intermunicipal, materializada em um protótipo funcional composto por uma API REST e uma interface *web* responsiva. A rastreabilidade dos requisitos mostra cobertura do núcleo operacional, com regras de reserva, capacidade, cancelamento, pagamento operacional e isolamento concentradas na API. Os testes unitários e a implantação experimental forneceram evidências técnicas limitadas ao escopo verificado. A avaliação exploratória registrou percepções favoráveis sobre uso, satisfação, utilidade e intenção, com médias entre 4,37 e 4,89, mas não constituiu escala padronizada, inspeção heurística ou validação empresarial. Assim, a solução deve ser entendida como protótipo funcional e base para evolução futura, e não como produto acabado; no estágio atual, constitui, no máximo, um protótipo apto a testes de validação de adoção operacional em cenário real.

Como trabalhos futuros, destacam-se: testes de integração com banco real, fluxos E2E e verificação concorrente; proteção adicional no banco, como *Row Level*

Security ou estratégia equivalente; integração com *gateway* de pagamento real; persistência de auditoria e observabilidade; fortalecimento do armazenamento de sessão; notificações por e-mail/SMS com processamento assíncrono; entidade dedicada de paradas com precificação por trecho; testes automatizados no cliente; melhorias da área administrativa, como mensagens de erro específicas, simplificação de fluxos e visão otimizada para *desktop*; a formalização e o aprimoramento dos padrões visuais e dos componentes reutilizáveis da interface; a validação com gestores e empresas de transporte, por meio de entrevistas semiestruturadas, estudo-piloto com viagens reais e análise de aderência aos processos organizacionais; e uma avaliação de usabilidade formal, com instrumento padronizado e amostra ampliada dos perfis administrativo e operacional, antes de eventual adoção em contexto real.

REFERÊNCIAS

- BIZERRIL, F. E. A. *Aplicativo Intercampi: desenvolvimento, operacionalização e relevância para a comunidade acadêmica*. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Energias) — Universidade da Integração Internacional da Lusofonia Afro-Brasileira, Redenção, 2024.
- Brasil. *Lei nº 13.709, de 14 de agosto de 2018 — Lei Geral de Proteção de Dados Pessoais (LGPD)*. 2018. Acesso em: 15 jun. 2026. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>.
- BROOKE, J. SUS: a “quick and dirty” usability scale. In: JORDAN, P. W. et al. (Ed.). *Usability Evaluation in Industry*. London: Taylor & Francis, 1996. p. 189–194.
- DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, v. 13, n. 3, p. 319–340, 1989. Acesso em: 6 jul. 2026. Disponível em: <<https://doi.org/10.2307/249008>>.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 7. ed. São Paulo: Pearson, 2018.
- EVANS, E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison-Wesley, 2003.
- FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000. Acesso em: 3 jul. 2026. Disponível em: <https://roy.gbiv.com/pubs/dissertation/fielding_dissertation.pdf>.
- HEUSER, C. A. *Projeto de Banco de Dados*. 6. ed. Porto Alegre: Bookman, 2009.
- IBGE. *Regiões de influência das cidades: 2018*. Rio de Janeiro: IBGE, Coordenação de Geografia, 2020. Instituto Brasileiro de Geografia e Estatística. Acesso em: 9 jul. 2026. ISBN 978-65-87201-04-7. Disponível em: <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv101728.pdf>>.

IBGE. *Censo 2022: como a população se desloca para estudar e trabalhar*. 2025. Instituto Brasileiro de Geografia e Estatística. Publicado em: 10 nov. 2025. Acesso em: 12 jan. 2026. Disponível em: <<https://educa.ibge.gov.br/jovens/materias-especiais/23064-censo-2022-como-a-populacao-se-desloca-para-estudar-e-trabalhar.html>>.

INEP. *Censo da Educação Superior 2024: notas estatísticas*. Brasília, DF, 2025. Acesso em: 12 jan. 2026. Disponível em: <<https://www.gov.br/inep/pt-br/centrais-de-contenido/acervo-linha-editorial/publicacoes-institucionais/estatisticas-e-indicadores-educacionais/censo-escolar-da-educacao-superior-2024-notas-estatisticas>>.

ISO. *ISO 9241-11:2018 — Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. 2018. International Standard. Acesso em: 6 jul. 2026. Disponível em: <<https://www.iso.org/standard/63500.html>>.

Joint Task Force. *Security and Privacy Controls for Information Systems and Organizations*. Gaithersburg, 2020. Acesso em: 3 jul. 2026. Disponível em: <<https://doi.org/10.6028/NIST.SP.800-53r5>>.

JONES, M. B.; BRADLEY, J.; SAKIMURA, N. *RFC 7519: JSON Web Token (JWT)*. 2015. Internet Engineering Task Force (IETF). Acesso em: 3 jul. 2026. Disponível em: <<https://www.rfc-editor.org/rfc/rfc7519>>.

LIKERT, R. A technique for the measurement of attitudes. *Archives of Psychology*, v. 22, n. 140, p. 5–55, 1932.

MARTIN, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Boston: Prentice Hall, 2017.

MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. [S.l.], 2011. Acesso em: 3 jul. 2026. Disponível em: <<https://doi.org/10.6028/NIST.SP.800-145>>.

National Institute of Standards and Technology. *Role-Based Access Control (RBAC)*. s.d. Glossário CSRC. Acesso em: 3 jul. 2026. Disponível em: <https://csrc.nist.gov/glossary/term/role_based_access_control>.

NESTJS. *NestJS — A Progressive Node.js Framework*. 2024. Acesso em: 15 jun. 2026. Disponível em: <<https://nestjs.com>>.

NIELSEN, J. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1993.

Owasp Foundation. *OWASP Top 10:2021 — The Ten Most Critical Web Application Security Risks*. 2021. Acesso em: 3 jul. 2026. Disponível em: <<https://owasp.org/Top10/>>.

PAIVA, A. D. d. *Sistema para Gerenciamento de Transportes*. Trabalho de Conclusão de Curso (Bacharelado em Tecnologia da Informação) — Universidade Federal Rural do Semi-Árido, Pau dos Ferros, 2023. Acesso em: 29 jan. 2026. Disponível em: <<https://repositorio.ufersa.edu.br/handle/prefix/11169>>.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: McGraw-Hill / AMGH, 2016.

PRETTO, F. et al. Pickbus: uma solução para serviços de transporte fretado de passageiros. *Revista Destaques Acadêmicos*, Lajeado, v. 13, n. 1, p. 167–187, 2021. ISSN 2176-3070. Acesso em: 29 jan. 2026. Disponível em: <<http://dx.doi.org/10.22410/issn.2176-3070.v13i1a2021.2857>>.

PRISMA. *Prisma — Next-generation Node.js and TypeScript ORM*. 2024. Acesso em: 15 jun. 2026. Disponível em: <<https://www.prisma.io>>.

PROVOS, N.; MAZIÈRES, D. *A Future-Adaptable Password Scheme (bcrypt)*. 1999. USENIX Annual Technical Conference. Acesso em: 15 jun. 2026. Disponível em: <<https://www.usenix.org/legacy/events/usenix99/provos.html>>.

REIS, A. L. d. et al. Modelagem de software para atender o sistema público de transporte escolar municipal. *Brazilian Journal of Development*, v. 7, n. 12, p. 111756–111774, 2021. ISSN 2525-8761. Acesso em: 29 jan. 2026. Disponível em: <<https://doi.org/10.34117/bjdv7n12-117>>.

SILVA, L. T. d. A. *Transporte universitário como política de acesso e permanência no Ensino Superior: análise situacional sobre o programa Passe Livre Universitário implementado no município de Itaocara – Rio de Janeiro (2013–2020)*. Dissertação (Mestrado em Políticas Sociais) — Universidade Estadual do Norte Fluminense Darcy Ribeiro, Campos dos Goytacazes, 2022.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

TANSTACK. *TanStack Router — Type-safe file-based routing for React*. 2024. Acesso em: 30 jun. 2026. Disponível em: <<https://tanstack.com/router>>.