

Plataforma para Gerenciamento e Visualização de Transporte Coletivo Privado

Lucas Lutz¹, Felipe Martin Sampaio¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Farroupilha

Farroupilha, RS - Brasil

contato.lucaslutz@gmail.com, felipe.sampaio@farroupilha.ifrs.edu.br

Abstract. *The use of private collective transportation is quite common, especially in business environments. At the same time, the increasing availability of rapid access to information, combined with the daily use of mobile devices and the evolution of resources and connectivity offered by these devices, creates a favorable scenario for the development of solutions capable of leveraging this abundance of data to enhance communication and operational efficiency. In this context, this study explores the use of technology to enable more efficient management of private collective transportation through three separate applications: one designed for the chartering company, another for the drivers, and a third for the passengers, each offering specific functionalities for management, operation, and monitoring, respectively.*

Keywords: *Management, Intelligent Transportation Systems, Transportation, Urban Mobility*

Resumo. *O uso do transporte coletivo privado é bastante comum, sobretudo no contexto empresarial. Paralelamente, o acesso cada vez mais rápido à informação, aliado ao uso cotidiano de dispositivos móveis e à evolução dos recursos e da conectividade disponíveis nesses equipamentos, cria um cenário favorável ao desenvolvimento de soluções capazes de utilizar a alta disponibilidade de dados para aumentar a eficiência na comunicação e nas operações diárias. Nesse contexto, insere-se o presente trabalho, que explora o uso da tecnologia para viabilizar uma gestão mais eficiente do transporte coletivo privado através de três aplicações separadas: uma voltada à empresa fretadora, outra destinada aos motoristas e, por fim, uma terceira para os passageiros, cada qual com funcionalidades específicas para gestão, operação e acompanhamento, respectivamente.*

Palavras-chave: *Gerenciamento, Sistemas Inteligentes de Transporte, Transporte, Mobilidade Urbana*

1. Introdução

Ao longo das últimas décadas, o processo de urbanização se intensificou juntamente com o crescimento populacional, levando à expansão dos centros urbanos. Essa expansão tornou o deslocamento uma necessidade essencial para grande parte da população. Cada vez mais pessoas passaram a depender de meios de transporte para

acessar trabalho, educação, saúde e lazer. (Pradhan, Arvin e Nair, 2021).

O processo de industrialização dos últimos dois séculos possibilitou a produção em massa de automóveis e consolidou um modelo de deslocamento baseado no transporte individual. Contudo, a infraestrutura viária urbana não acompanhou esse crescimento, resultando em congestionamentos, aumento da poluição e desafios logísticos (Gao, 2022). No Brasil, em 2024, havia cerca de 124 milhões de veículos em circulação, sendo 73,9% automóveis e motocicletas, o que evidencia a dimensão do problema da mobilidade urbana (SENATRAN, 2024).

Diante desse cenário de intensa frota de veículos individuais que, na maioria dos casos, transportam apenas até quatro pessoas por vez, é necessário buscar soluções que reduzam a quantidade de automóveis em circulação. Junto ao transporte coletivo público, o transporte coletivo privado surge como uma alternativa para amenizar a problemática do volume de veículos em circulação, visto que, através de vans e ônibus, é possível comportar vários passageiros por veículo, oferecendo flexibilidade para deslocamento e economia para os seus utilizadores. Usualmente este tipo de solução demanda gestão de veículos, rotas e passageiros, além de exigir uma logística eficiente para garantir que o transporte seja o mais ágil possível. (Sun, 2025).

Do ponto de vista corporativo e ambiental, uma gestão deficitária de transporte coletivo privado ocasiona o desperdício de recursos quando uma rota é ineficiente ou quando o veículo é conduzido até um passageiro que não utilizará o transporte no momento em questão, já que o deslocamento desnecessário aumenta o consumo de combustível e o desgaste do veículo. Um dos desafios que pode prejudicar a gestão de empresas fretadoras é administrar viagens à medida que sua quantidade cresce, podendo gerar um aumento no custo operacional que será repassado aos utilizadores do transporte, impactando negativamente na satisfação com o serviço ofertado. Por sua vez, as empresas de pequeno porte podem perder sua competitividade devido à falta de um sistema eficiente e que viabilize um transporte ágil e confiável, prejudicando a adesão ao transporte coletivo privado. (Sun, 2025).

Uma gestão eficiente não se resume apenas à operação dos veículos, mas também à experiência do utilizador. Um usuário descontente com o serviço de transporte pode buscar o transporte individual como alternativa para locomoção, aumentando o volume do trânsito. Um dos principais fatores que influenciam a adesão ao transporte coletivo é a previsibilidade do serviço. Em uma análise da Associação Nacional das Empresas de Transportes Urbanos NTU (NTU, 2023), 24% dos entrevistados afirmaram que o maior incentivo para o uso de transporte coletivo público é de esperar por menor período de tempo nos pontos e paradas de ônibus. Além disso, a partir da mesma pesquisa foi possível concluir que: “A comunicação em tempo real por meio de equipamentos nos pontos de embarque e outros recursos (por exemplo, aplicativos) pode contribuir para atenuar a percepção negativa sobre o tema. Se existem canais de comunicação confiáveis sobre o tempo de espera, os usuários podem programar suas viagens de forma mais cômoda, minimizando a percepção negativa em relação ao tempo de espera pelos ônibus” (Moura, 2023, apud NTU, 2023).

Para tornar essa comunicação mais eficaz, tecnologias inovadoras têm sido implementadas no setor de transportes. O advento dos Sistemas Inteligentes de Transporte (SIT) permite diversas melhorias para a mobilidade urbana através do uso de sensores, geolocalização e conectividade presentes em dispositivos móveis e *IoT* (Internet of Things). Segundo Figueiredo (2005): “*De uma forma simplificada pode*

dizer-se que os SIT utilizam as tecnologias de informação, comunicação e controlo para melhorar o desempenho e a eficiência das operações da rede de transportes [...]”. A partir deste contexto é possível automatizar processos e acelerar a troca de informações através da comunicação instantânea, suprimindo os pontos elencados pela análise da pesquisa da Moovit supracitada. A ilustração abaixo demonstra o uso de SIT e seus principais elementos.

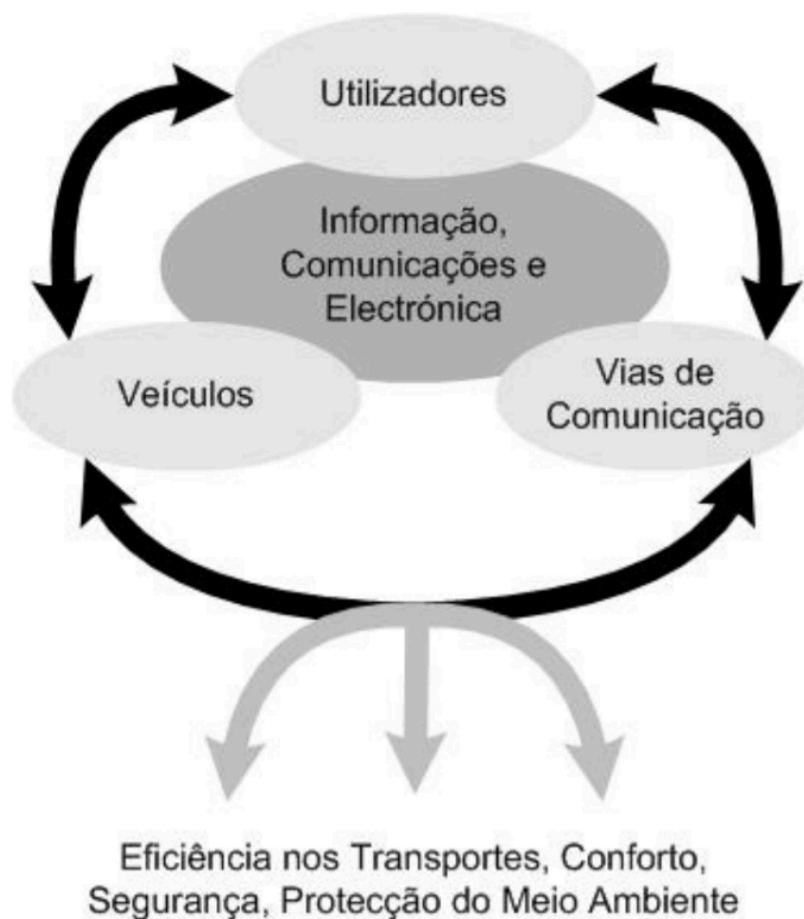


Figura 1. Sistemas Inteligentes de Transporte

Fonte: Figueiredo (2005).

Diante do exposto, o objetivo deste trabalho é desenvolver uma plataforma de código aberto voltada à comunicação entre passageiros e motoristas no contexto de serviços de transporte coletivo privado. Devido a essa característica, a solução pretende ampliar a integração com outros sistemas, incentivar a colaboração entre diferentes desenvolvedores e operadores do setor e promover o acesso inclusivo a tecnologias de mobilidade, favorecendo a adaptabilidade e a evolução contínua da aplicação.

O presente trabalho propõe o desenvolvimento de aplicações para a gerência do transporte coletivo privado. Partindo da premissa do uso de SIT, os principais recursos utilizados serão dispositivos móveis baseados em sistema operacional Android e iOS, conexão com a internet e geolocalização em tempo real. Para isso, contempla-se o desenvolvimento de uma plataforma tecnológica que inclui um aplicativo móvel para que passageiros possam acompanhar, em tempo real, a localização da van; uma

ferramenta para que motoristas recebam informações sobre os pontos de embarque e desembarque; e um sistema web para que as empresas possam gerenciar rotas, motoristas e passageiros. Adicionalmente, o sistema prevê a integração de geolocalização via sistema de posicionamento global (GPS) para previsão precisa de chegada aos pontos de parada.

A escolha deste tema surgiu do interesse pessoal do autor pela área de mobilidade e pela vivência como usuário de transporte coletivo privado. Sendo passageiro, foi possível observar de forma prática alguns desafios enfrentados no dia a dia, especialmente relacionados à previsibilidade dos horários e à eficiência das rotas. Conversas com motoristas e outros passageiros reforçaram a percepção pessoal do autor, de que um sistema integrado, que unisse comunicação em tempo real, geolocalização e gestão centralizada, poderia trazer melhorias significativas para a operação e para a experiência de todos os envolvidos.

O presente trabalho está organizado da seguinte forma: a seção 2 apresenta a fundamentação teórica; a seção 3 descreve os procedimentos metodológicos utilizados; a seção 4 apresenta a proposta da solução e suas funcionalidades; a seção 5 indica a implementação; a seção 6 indica os resultados obtidos; por fim, a seção 7 apresenta as considerações finais do trabalho.

2. Referencial Teórico

2.1. Sistemas Inteligentes de Transporte (SIT)

Os sistemas inteligentes de transporte estão presentes em diversas esferas, sendo utilizados desde o gerenciamento de frotas de ônibus urbanos, controle de tráfego em tempo real, até o rastreamento de veículos e passageiros. Com as funcionalidades supracitadas e escopo abrangente, tornam-se ferramentas essenciais para a modernização da mobilidade urbana. (Figueiredo, 2005).

O surgimento dos Sistemas Inteligentes de Transporte (SIT) como conhecidos hoje ocorreu em etapas: preparação (1930-1980), estudo de viabilidade (1980-1995) e desenvolvimento de sistemas (1995-atualmente). Como marco inicial, destaca-se o surgimento dos primeiros semáforos, em 1928, nos Estados Unidos da América. (Figueiredo, 2005).

A primeira etapa foi caracterizada pela transição entre sistemáticas manuais e mecânicas para eletrônicas em meio a um aumento exponencial na quantidade de veículos em circulação nas maiores capitais do mundo e de avanços significativos na área de microprocessadores. Nesse contexto, surgiram os primeiros sistemas automatizados, como os semáforos eletrônicos e os sistemas de posicionamento global (GPS), originalmente desenvolvidos para fins militares, mas posteriormente incorporados ao transporte civil. A segunda etapa teve as parcerias entre governos, empresas do ramo privado e centros de pesquisa como característica principal, com o objetivo de explorar as premissas e a viabilidade do desenvolvimento e fabricação de veículos autônomos e transportes automatizados. A etapa atual, iniciada a partir de meados da década de 1990, é marcada pelo desenvolvimento e implementação concreta de soluções baseadas em SIT, aplicadas em contextos urbanos reais e alinhadas a políticas públicas de mobilidade e segurança viária. (Figueiredo, 2005).

Dada sua ampla adesão, aplicabilidade em diferentes modais e a avançada integração entre sistemas, os Sistemas Inteligentes de Transporte (SIT) têm se

expandido para além dos modelos tradicionais, como o transporte público, ganhando espaço também no setor de transporte coletivo privado. Aplicações recentes, como as descritas na seção 2.2. têm demonstrado a viabilidade da adoção desses sistemas em soluções regionais e segmentadas.

2.2. Revisão da Literatura

O funcionamento eficaz dos SIT exige um ecossistema tecnológico baseado na integração entre sensores, sistemas de posicionamento por satélite (GPS), redes de comunicação (como internet e bluetooth), além de sistemas capazes de processar grandes volumes de dados (Big Data) e de comunicar informações em tempo real. Embora nem todas essas tecnologias sejam utilizadas em todas as aplicações, sua combinação, em diferentes níveis, permite construir soluções adaptáveis a diferentes contextos, como o transporte coletivo privado. Com uma ampla gama de tecnologias à disposição, a escolha dos recursos tecnológicos depende das características da operação, das restrições do ambiente e das demandas de comunicação.

Diversas soluções já exploram essa combinação tecnológica em diferentes partes do mundo, oferecendo exemplos concretos da aplicação de SIT na mobilidade urbana. No contexto internacional, pode-se citar o uso do software Moovit (MOOVIT, 2025). Esta aplicação tem como objetivo principal fornecer informações acerca de linhas de transportes públicos, como trens, metrô e ônibus. De forma semelhante, o Citymapper (CITYMAPPER, 2025) expande esse escopo permitindo a integração entre diversas modalidades de transporte, o que possibilita ao usuário o planejamento de viagens com mais de uma modalidade de transporte através de uma única aplicação, trazendo uma experiência transparente e integrada. Em âmbito nacional, a aplicação Murbi (MURBI, 2025) foi desenvolvida em Caxias do Sul/RS e voltada ao transporte coletivo privado. Ela fornece uma plataforma web para empresa fretadora, um aplicativo mobile para o motorista e outro para o passageiro. Entre suas funcionalidades estão o planejamento de rotas e horários e a visualização destes em tempo real através de geolocalização.

O presente trabalho se inspira na metodologia de funcionamento do Murbi que, assim como os demais sistemas supracitados, é um software proprietário. Nesse contexto, propõe-se o desenvolvimento de uma aplicação de código aberto, com o objetivo de ampliar a integração com outros sistemas, incentivar a colaboração entre diferentes atores sociais e oferecer maior flexibilidade e adaptabilidade. Nesse cenário, um modelo baseado em software open source se apresenta como alternativa viável, permitindo que qualquer operador, independentemente do porte, possa adotar, adaptar e evoluir a solução, tornando o acesso a tecnologias de mobilidade mais inclusivo e sustentável. Conforme Levin e Prietula (2014), a colaboração aberta característica do software livre demonstra capacidade de gerar soluções robustas e continuamente aprimoradas por diferentes atores, evidenciando seu potencial como modelo inclusivo e sustentável de inovação.

A seguir, apresentam-se os procedimentos metodológicos adotados para a condução da pesquisa e estruturação do desenvolvimento da aplicação, contemplando as etapas, técnicas e ferramentas utilizadas.

3. Procedimentos Metodológicos

Com base na fundamentação descrita na seção anterior, a metodologia adotada para o desenvolvimento da aplicação foi estruturada em etapas que detalham o levantamento

de requisitos, a modelagem do sistema e as técnicas e ferramentas utilizadas para implementação e validação da solução.

Trata-se de uma pesquisa aplicada, de caráter exploratório e descritivo, com abordagem qualitativa e quantitativa. Essa classificação justifica-se pelo objetivo de propor uma solução tecnológica concreta, analisando práticas existentes e avaliando a viabilidade da implementação em ambiente controlado

3.1. Levantamento de Requisitos

O processo foi iniciado pelo levantamento de requisitos funcionais, conduzido de forma exploratória, a partir da observação de fluxos de trabalho característicos do transporte coletivo privado no município de Farroupilha/RS. Embora inspirados em práticas adotadas por empresas reais, os requisitos foram redigidos de maneira genérica, buscando contemplar situações comuns a empresas fretadoras, motoristas e passageiros, sem referência direta a organizações específicas.

Requisitos funcionais:

- Permitir cadastro e autenticação de usuários (empresa fretadora, motorista e passageiro).
- Disponibilizar mapas para consulta de rotas, veículos e passageiros, com atualização em tempo real.
- Oferecer visualização de rotas por status (ativas e inativas).
- Permitir o gerenciamento de registros de vans, motoristas, passageiros e rotas.
- Possibilitar a associação de pontos de embarque e destino a cada passageiro.
- Gerar, automaticamente, o traçado das rotas conforme confirmações de presença.
- Enviar notificações aos passageiros sobre início de rotas.
- Exibir estimativas de chegada do transporte ao ponto de embarque.
- Permitir ao motorista iniciar e finalizar rotas.

Histórias de Usuários:

A partir dos requisitos descritos acima, foram elaboradas histórias de usuários, representando as necessidades práticas do sistema de maneira estruturada e centrada nos diferentes perfis de usuário. As histórias de usuários estão descritas abaixo.

Empresa fretadora:

- “Como empresa fretadora, desejo realizar cadastro e *login* na plataforma.”
- “Como empresa fretadora, desejo visualizar as rotas de transporte por meio de mapas, com informações como nome dos passageiros, motorista e van utilizada. A localização da van deve ser atualizada em tempo real.”
- “Como empresa fretadora, desejo filtrar rotas por status: ativas (em andamento) e inativas (existentes, mas não em operação).”
- “Como empresa fretadora, desejo adicionar, editar e excluir registros de vans.”

- “Como empresa fretadora, desejo adicionar, editar e excluir registros de motoristas.”
- “Como empresa fretadora, desejo adicionar, editar e excluir rotas.”
- “Como empresa fretadora, desejo associar pontos de embarque e destino a cada passageiro, dentro de uma rota.”

Passageiro:

- “Como passageiro, desejo realizar cadastro e *login* na aplicação.”
- “Como passageiro, desejo visualizar, em tempo real, a localização da van responsável pela minha rota.”
- “Como passageiro, desejo receber uma notificação quando uma rota à qual estou vinculado for iniciada.”
- “Como passageiro, desejo receber uma estimativa de chegada do transporte ao endereço do meu embarque.”

Motorista:

- “Como motorista, desejo realizar cadastro e *login* na aplicação.”
- “Como motorista, desejo visualizar, em um mapa, todos os passageiros da rota em andamento.”
- “Como motorista, desejo que a aplicação gere, automaticamente, o traçado da rota com base apenas nos passageiros que confirmaram presença para o dia em questão.”
- “Como motorista, desejo iniciar e finalizar rotas.”

Esses requisitos serviram de base para a idealização da diagramação e seleção das tecnologias apresentadas na próxima seção.

3.2. Modelagem do Sistema

Baseado nos requisitos e histórias de usuários descritos acima, com a finalidade de ilustrar o funcionamento do sistema proposto, elaborou-se o diagrama de casos de uso. O diagrama está representado na Figura 2 e contempla três atores principais: Administrador, Passageiro e Motorista.

O Administrador representa o responsável pela empresa fretadora, sendo o usuário da interface web destinada à administração do sistema. O Motorista, por sua vez, utiliza a aplicação para acessar informações referentes às rotas e aos passageiros. Já o Passageiro utiliza o sistema para confirmar sua presença no transporte e acompanhar, em tempo real, sua localização.

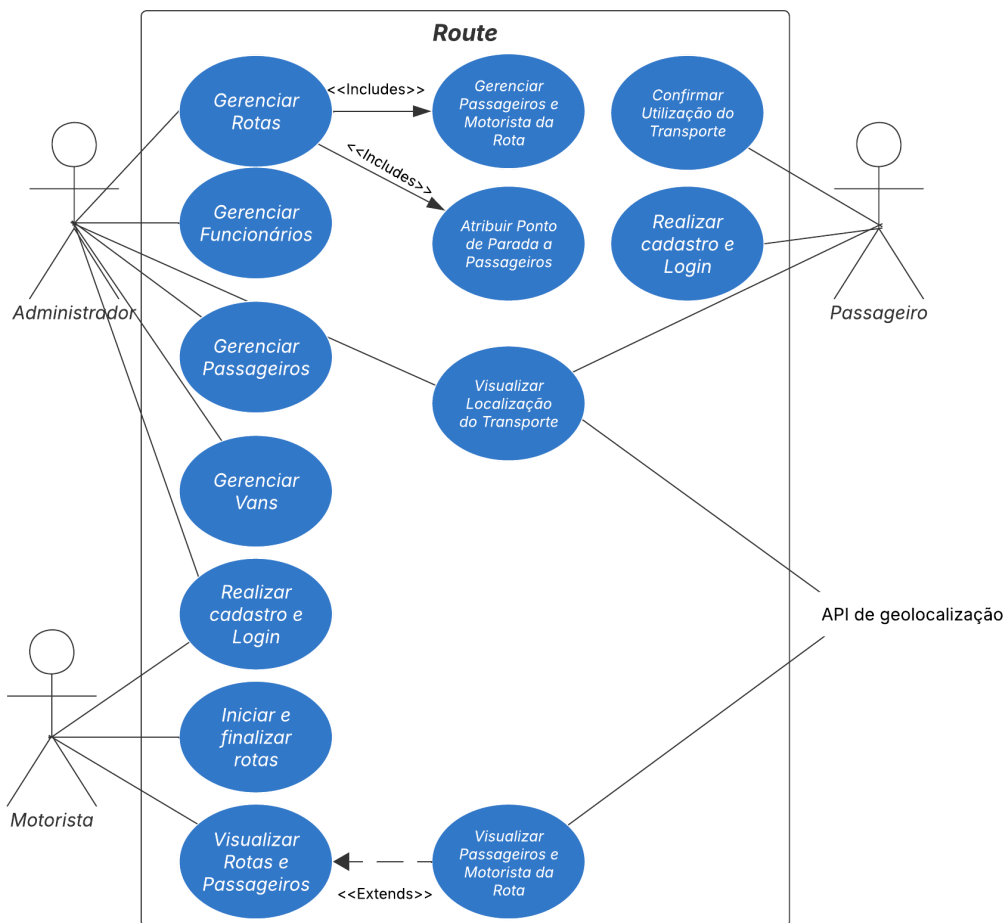


Figura 2. Casos de Uso
Desenvolvido com LucidChart (www.lucidchart.com)
Fonte: Autoria Própria.

3.3. Critérios de Avaliação

A validação dos requisitos foi realizada em ambiente controlado, de forma experimental. Para isso, cada requisito funcional identificado na etapa de levantamento foi testado diretamente na aplicação em execução local. A verificação consistiu em observar se a funcionalidade estava presente, se seu comportamento correspondia ao esperado e se a experiência de uso era fluida, sem falhas de navegação ou inconsistências. Essa forma de validação, embora não envolva usuários finais, permitiu assegurar que a aplicação atendeu aos objetivos inicialmente definidos e que os fluxos principais estavam coerentes com as histórias de usuário levantadas.

Os critérios de sucesso do desenvolvimento foram definidos a partir de percepções como: Consistência da atualização em tempo real, facilidade de uso das interfaces e estabilidade durante os testes. Esses parâmetros orientaram a priorização de recursos e a validação da solução.

A partir dessa metodologia, foi possível estruturar o trabalho de forma integrada: o levantamento de requisitos orientou a modelagem do sistema, que, por sua vez, serviu

de base para a implementação. Por fim, a etapa de validação permitiu verificar se a solução desenvolvida atendia aos objetivos definidos, garantindo coerência em todas as fases do processo.

4. Proposta da Solução

A plataforma proposta é composta por três módulos interligados: (1 - Route Passageiro) um aplicativo móvel destinado aos passageiros, (2 - Route Motorista) um aplicativo móvel para os motoristas, e (3 - Route Web) um painel administrativo para as empresas de transporte. O objetivo é permitir comunicação em tempo real e gerenciamento de passageiros, motoristas, rotas e motoristas.

No módulo para passageiros, é possível acompanhar a localização do veículo em tempo real, confirmar o uso do transporte no dia e visualizar dados do motorista. No módulo do motorista, são exibidos os pontos de parada com base nas confirmações dos passageiros, auxiliando na visualização do trajeto. No painel web, a empresa pode cadastrar usuários, motoristas, rotas e consultar o deslocamento dos veículos.

A proposta visa resolver problemas recorrentes enfrentados no transporte de passageiros sob demanda, especialmente em contextos empresariais ou escolares, onde o controle sobre rotas, presença de usuários e localização dos veículos tende a ser feito de forma manual, imprecisa ou descentralizada. A ausência de uma solução integrada pode acarretar atrasos, falta de previsibilidade, insegurança por parte dos passageiros e dificuldade de gestão por parte das empresas operadoras.

Dessa forma, a solução proposta centraliza todas as funcionalidades essenciais em uma única aplicação digital, aproveitando recursos em tempo real fornecidos por tecnologias já presentes no mercado. A comunicação entre os três módulos ocorre por meio de uma infraestrutura em nuvem, garantindo sincronização de dados e escalabilidade.

Entre as principais funcionalidades da solução, destacam-se:

- Sincronização em tempo real entre motoristas, passageiros e gestores;
- Gestão autônoma e segura de presença por parte dos passageiros;
- Visualização georreferenciada do percurso e da posição atual da van;
- Cadastro e edição simplificados de rotas, usuários e veículos por parte da empresa;
- Interface intuitiva e compatível com dispositivos móveis e navegadores modernos, garantindo acessibilidade e boa experiência de uso.

A proposta também considera a privacidade e segurança dos dados dos usuários, utilizando autenticação via Firebase e regras de acesso personalizadas por tipo de usuário (passageiro, motorista, empresa).

4.1. Diagramas de navegação

Com o propósito de representar graficamente o fluxo de funcionalidades disponíveis em cada aplicação do sistema, foram desenvolvidos três diagramas de navegação distintos: um para a interface web administrativa (Figura 3), outro para o aplicativo destinado aos motoristas (Figura 4) e um terceiro voltado aos passageiros (Figura 5). Tais diagramas têm como objetivo evidenciar a sequência e a interligação

entre as telas e ações principais de cada sistema, facilitando a compreensão da estrutura de navegação e da lógica de uso.

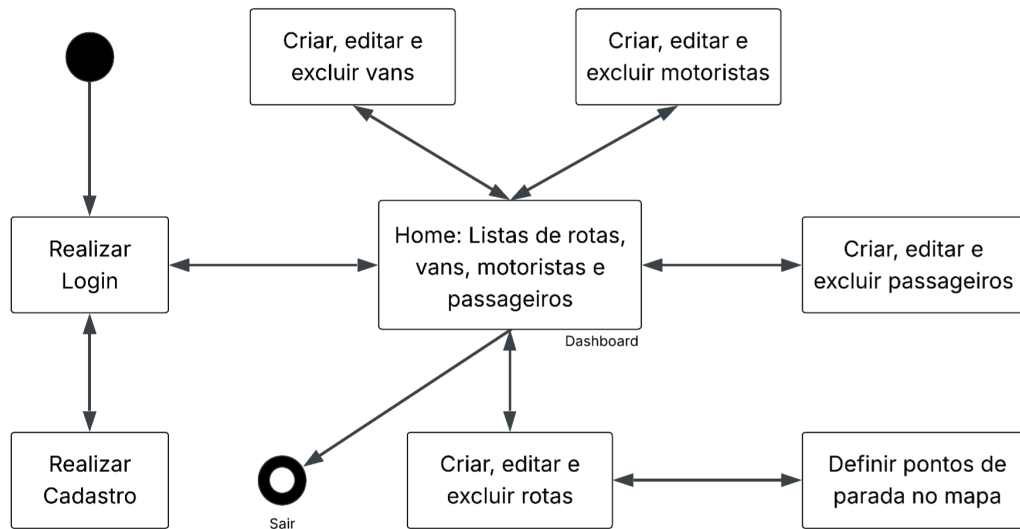


Figura 3. Diagrama de navegação - Plataforma Web

Fonte: Autoria Própria.

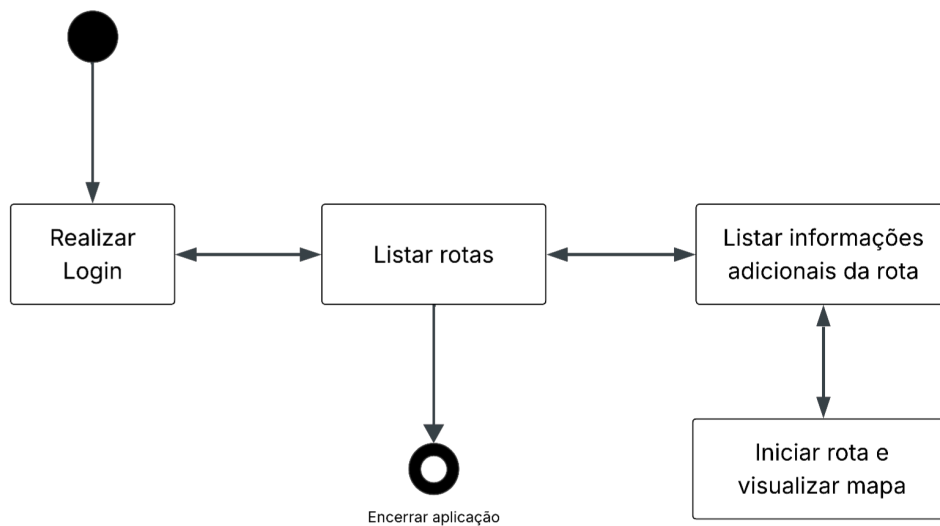


Figura 4. Diagrama de navegação - Aplicação para motorista

Fonte: Autoria Própria.

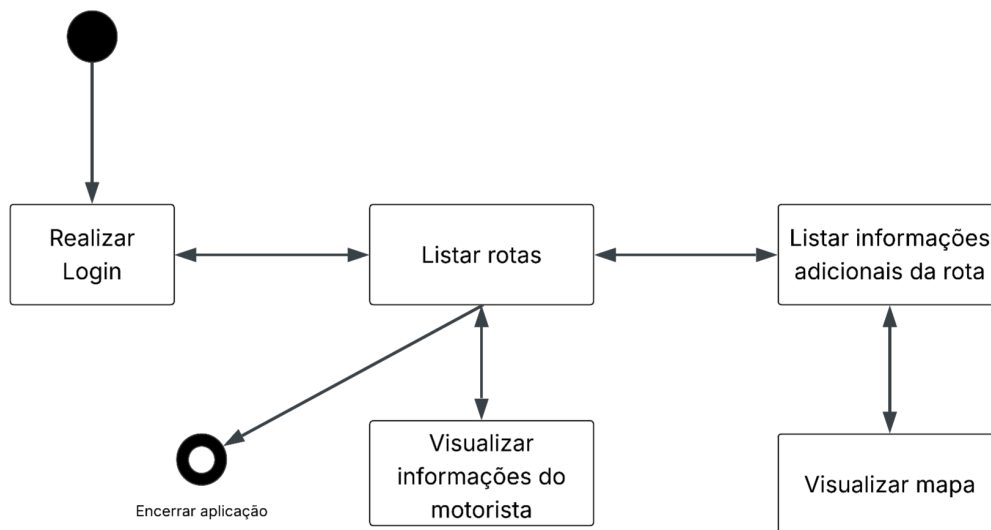


Figura 5. Diagrama de navegação - Aplicação para passageiro

Fonte: Autoria Própria.

5. Implementação

A implementação da solução proposta seguiu as diretrizes estabelecidas na etapa de levantamento de requisitos e modelagem da arquitetura. A construção dos três módulos da plataforma foi conduzida de forma iterativa, priorizando funcionalidades essenciais para o uso prático da aplicação por parte da empresa parceira.

A seguir, descrevem-se as principais etapas de implementação e os aspectos técnicos relevantes de cada módulo.

5.1. Arquitetura

A arquitetura das aplicações foi projetada com foco na modularidade, reutilização de código e separação clara de responsabilidades. Apesar das diferenças naturais entre os ambientes web e mobile, ambas as soluções adotaram padrões de organização e desenvolvimento que promovem fácil manutenção, escalabilidade e coesão entre os módulos do sistema.

Tanto a plataforma web quanto as aplicações mobile foram idealizados baseando-se na separação de componentes visuais, lógica de negócio, serviços externos e tipagem de dados. Essa abordagem não apenas facilitou a manutenção e evolução das funcionalidades ao longo do desenvolvimento, como também permitiu maior alinhamento entre as equipes responsáveis por diferentes plataformas.

Além disso, ambas as arquiteturas foram construídas sobre a ideia de integração com serviços externos por meio de SDKs (kit de desenvolvimento de software - Do inglês, Software Development Kit) e APIs (Interface de Programação de Aplicações - Do inglês, Application Programming Interface) RESTful. Essa centralização dos serviços facilitou o compartilhamento de regras de negócio e estratégias de autenticação, bem como a manipulação de dados em tempo real em todas as aplicações.

5.2. Tecnologias Utilizadas

As tecnologias utilizadas no desenvolvimento das aplicações foram escolhidas com base em critérios de escalabilidade, desempenho, integração com serviços externos e facilidade de manutenção. Tanto a plataforma web quanto as aplicações mobile compartilham uma base conceitual semelhante, centrada na biblioteca *React*, o que permitiu o reaproveitamento de lógica de negócio, padronização de componentes e uma curva de aprendizado uniforme.

Além disso, serviços como Firebase e Google Maps foram adotados em todas as plataformas, viabilizando a implementação de funcionalidades em tempo real, autenticação segura, manipulação geoespacial de dados e integração com SDKs modernos. O uso de TypeScript também foi comum a todas as aplicações, contribuindo para a segurança e previsibilidade do código, tanto no front-end quanto em chamadas a serviços externos.

Dessa forma, a arquitetura do projeto priorizou o uso de tecnologias amplamente consolidadas no ecossistema TypeScript, com foco em interoperabilidade. A seguir, são descritas as tecnologias específicas utilizadas em cada uma das aplicações desenvolvidas.

5.2.1. Autenticação e Autorização de Usuários

O sistema RouteWeb adota o Firebase Authentication como base para o gerenciamento de usuários, aliado ao uso de *Custom Claims* para controle de papéis e ao Firestore para persistência dos dados relacionados. Essa arquitetura é organizada em duas camadas: o Firebase Client SDK, utilizado no *frontend* para autenticação inicial, e o Firebase Admin SDK, empregado no *backend* para operações administrativas.

O processo de criação de usuários inicia-se com o cadastro do administrador da empresa, realizado no *frontend*, por meio do método *createUserWithEmailAndPassword()* do Firebase. Após a criação da conta, os dados são enviados ao backend, que registra as informações no Firestore e vincula o usuário à respectiva empresa. A partir desse ponto, o administrador tem a capacidade de cadastrar motoristas e passageiros, cujos perfis são armazenados em subcoleções específicas da empresa no banco de dados, conforme ilustra a Figura 6.

```

// Cria o documento da empresa no Firestore
const empresaRef = await db.collection("empresas").add({
  nome: empresaNome,
  telefone: empresaTelefone,
  criadaEm: new Date(),
  email: email
});
const empresaId = empresaRef.id;

// Cria o documento do administrador da empresa no Firestore
await db.collection("usuarios").doc(uid).set({
  email,
  empresaId,
  criadoEm: new Date(),
  role: 'admin'
});

//Configura Custom Claims com o papel de administrador do sistema
await auth.setCustomUserClaims(uid, {
  empresaId,
  role: 'admin'
});

```

Figura 6. Trecho de código referente à criação de administrador de empresa.

Fonte: Aatoria Própria.

O mecanismo de Custom Claims desempenha papel central no controle de acesso. Cada usuário recebe atributos adicionais no token de autenticação, como o campo *role* (administrador, motorista ou passageiro) e o identificador da empresa (*empresaId*). Esses atributos possibilitam um controle granular de permissões, assegurando que cada usuário acesse apenas os recursos condizentes com seu papel e vinculados à sua organização. Além disso, como as claims são validadas no servidor e incorporadas ao token JSON Web Token (JWT), conforme Figura 7, obtém-se maior segurança, desempenho e flexibilidade.

```

try {
  //Autenticação com o Firebase Authentication
  const userCredential = await signInWithEmailAndPassword(auth, email, password);
  const user = userCredential.user;

  // Aguarda a atualização dos claims
  await user.getIdToken(true);

  // Busca os claims do usuário
  const tokenResult = await user.getIdTokenResult();
  const claims = tokenResult.claims;

  // Valida se usuário está vinculado a alguma empresa
  if (!claims.empresaId) {
    throw new Error("Usuário não está vinculado a uma empresa.");
  }
  router.push("/dashboard");
}

```

Figura 7. Trecho de código referente à autenticação de administrador de empresa.

Fonte: Aatoria Própria.

A estrutura de dados do Firestore reflete essa organização hierárquica, com coleções principais para empresas e usuários administradores, e subcoleções específicas para motoristas e passageiros vinculados a cada empresa. Esse modelo favorece o isolamento entre organizações (multi-tenancy), evitando que usuários de diferentes empresas tenham acesso a informações alheias.

Do ponto de vista de segurança, todas as rotas de API protegidas exigem autenticação e verificação dos papéis definidos em Custom Claims. Um *middleware* especializado garante que motoristas, passageiros e administradores tenham acesso apenas às funcionalidades destinadas ao seu perfil. É possível visualizar o código do *middleware* através da Figura 8.

```
// Verificar o token no Firebase Admin
const decodedToken = await auth.verifyIdToken(token);
if (!decodedToken) {
  return {
    success: false,
    error: 'Token inválido ou expirado',
    status: 401
  };
}
// Extrair informações do usuário
const userInfo = {
  uid: decodedToken.uid,
  email: decodedToken.email,
  role: decodedToken.role || 'motorista',
  empresaId: decodedToken.empresaId
};
return {
  success: true,
  user: userInfo
};
} catch (error) {
  if (error.code === 'auth/id-token-expired') {
    return {
      success: false,
      error: 'Token expirado',
      status: 401
    };
  }

  if (error.code === 'auth/id-token-revoked') {
    return {
      success: false,
      error: 'Token revogado',
      status: 401
    };
  }
}
```

Figura 8. Trecho de código referente ao middleware de autenticação com token JWT.

Fonte: Autoria Própria.

Essa abordagem resulta em um sistema de autenticação e autorização robusto, capaz de atender às diferentes necessidades de administradores, motoristas e passageiros, mantendo integridade, segurança e eficiência operacional.

5.2.2. Armazenamento de dados

Com o intuito de representar a estrutura de armazenamento de dados do sistema, elaborou-se uma ilustração (Figura 9) que distingue os dois tipos de banco de dados presentes: o Cloud Firestore (GOOGLE INC, 2025), utilizado para armazenar dados consistentes, e o Realtime Database (GOOGLE INC, 2025), destinado ao gerenciamento de informações em tempo real. Na representação gráfica, os retângulos simbolizam as coleções de dados, enquanto os elementos posicionados à sua direita correspondem aos pares chave-valor contidos em cada documento dessas coleções. Tal organização visa evidenciar a lógica de persistência e acesso às informações, conforme os diferentes requisitos de consistência e atualização dinâmica do sistema.

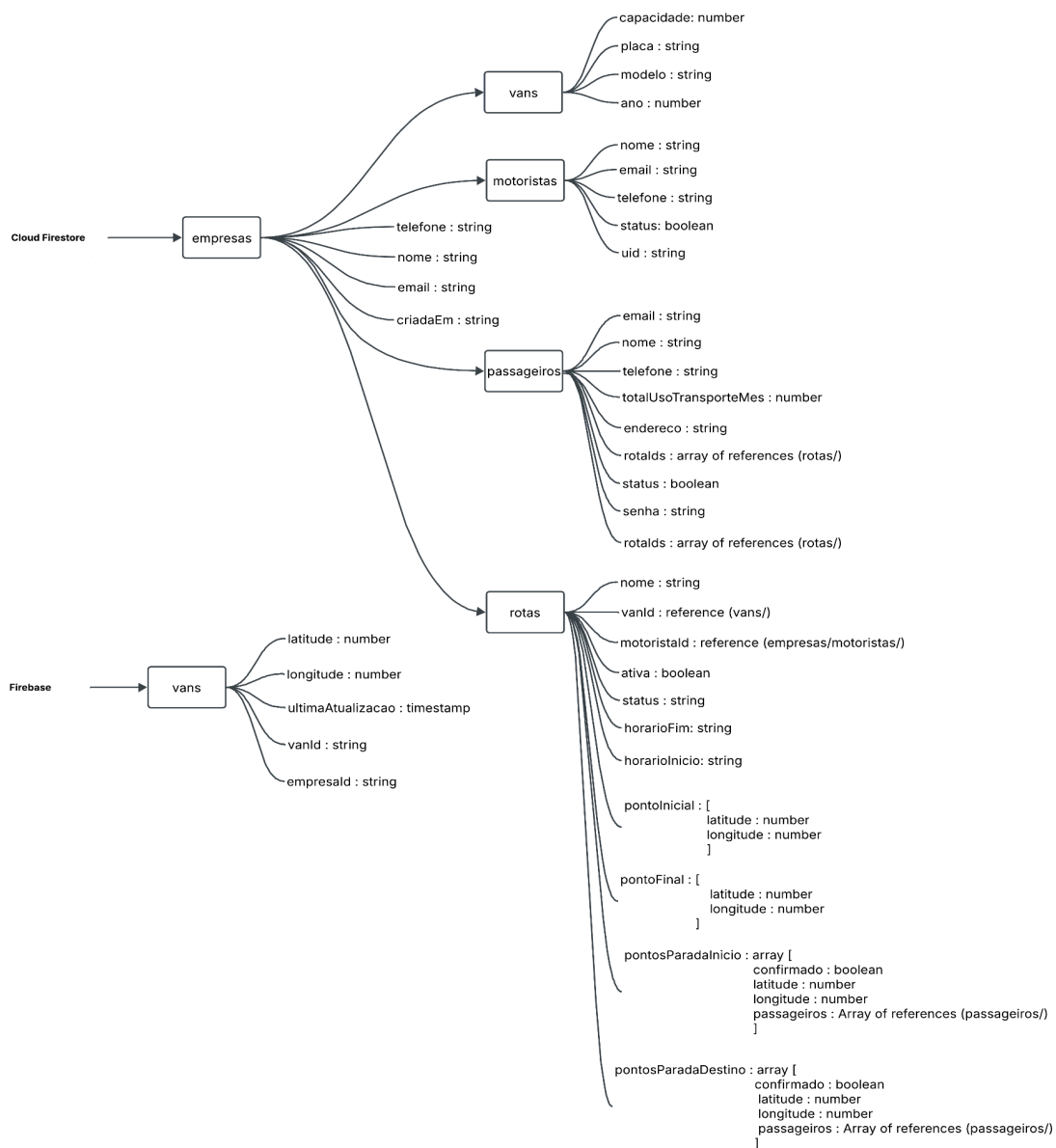


Figura 9. Estruturação de armazenamento de dados
Desenvolvido com LucidChart (www.lucidchart.com)

Fonte: Autoria Própria.

5.2.3. Aplicação Web

O desenvolvimento da plataforma web foi realizado com o uso do *framework* Next.js (NEXT.JS, 2025), versão 15.2.0. Por ser baseado na biblioteca React, ele permite construir aplicações de alto desempenho e escaláveis. O Next.js possui diversas funcionalidades que facilitam tanto o processo de desenvolvimento quanto o de manutenção das aplicações, além de possuir compatibilidade com bibliotecas amplamente utilizadas atualmente, como TailwindCSS (TAILWIND LABS INC, 2025). Aliando essas características ao fato do Next.js ser compatível com serviços do Firebase (GOOGLE INC, 2025), que fornece autenticação, banco de dados em tempo real e envio de notificações, o *framework* se mostra uma opção viável e consistente para ser base do presente projeto.

Cabe destacar suas principais características, sendo a principal delas a otimização automática de imagens, fontes e scripts, o que traz um desempenho satisfatório em aplicações web complexas. Ele também possui a capacidade de realizar a renderização do conteúdo de páginas web no lado do servidor, permitindo a consolidação de front-end e back-end em uma única base de código. Dessa forma, reduz-se a necessidade de criar *APIs* externas separadas para comunicação entre camadas, uma vez que o Next.js oferece suporte nativo à criação de rotas internas (API Routes). O Next.js também suporta componentes assíncronos do React, permitindo uma busca de dados integrada, considerando tanto dados presentes no servidor quanto dados presentes no cliente. Outra funcionalidade relevante deste *framework* é a criação de sistema de rotas similar à estruturação de arquivos, já familiar para usuários e desenvolvedores, que torna a organização dos projetos intuitiva. Por fim, o Next.js também oferece suporte nativo à ferramenta ESLint (OPENJS FOUNDATION, 2025), permitindo uma qualidade e manutenção de código consistentes.

A linguagem de programação adotada no projeto foi o TypeScript (MICROSOFT, 2025), que é baseado em JavaScript (MOZILLA FOUNDATION, 2025) e adiciona o suporte à tipagem estática para esta linguagem. Por permitir a definição explícita de tipos de dados, é possível detectar erros durante a fase de desenvolvimento, fornecendo previsibilidade e segurança no que diz respeito à manipulação de dados. O TypeScript é nativamente compatível com Next.js, sendo uma das principais linguagens utilizadas junto a este *framework*. Por ser desenvolvido e mantido pela Microsoft, atualmente há um amplo suporte e uso desta linguagem pela comunidade de desenvolvimento. Além disso, a linguagem é integrada aos principais editores de código atuais, permitindo usufruir de recursos como auto-complete e verificação de erros em tempo real durante o desenvolvimento.

Para a construção das interfaces das aplicações foi utilizada a biblioteca React 18.2.0 (META PLATFORMS INC, 2025). Sua principal característica é permitir o desenvolvimento de interfaces com componentes, o que permite uma ótima reutilização e promove organização no código-fonte. Aliado ao fato do React combinar lógica e estrutura visual dentro de um mesmo componente, é possível padronizar e escalar a estilização das aplicações de forma rápida. Outra característica do React é que ele pode ser integrado com frameworks como Next.js e serviços como Firebase, e ambos os exemplos citados possuem *SDK* (software development kit) específicos para esta biblioteca.

Para a estilização das interfaces de usuário, aliado ao React, foi utilizada a biblioteca TailwindCSS em sua versão 3.4.1. Por fornecer classes prontas e utilitárias, permite realizar personalizações diretamente no código HTML de forma fácil, prática e legível. Além disso, para garantir consistência na aplicação de estilos e evitar conflitos entre classes CSS, foram utilizadas as bibliotecas clsx e tailwind-merge, responsáveis por compor e unificar classes de forma otimizada (CLSX, 2025; TAILWIND MERGE, 2025).

Para as funcionalidades de back-end foi utilizada a plataforma Firebase, desenvolvida pela Google. Entre elas estão: autenticação de usuários, banco de dados em tempo real, armazenamento de arquivos, notificações e gerenciamento de dados. Por se tratar de uma plataforma *BaaS* (Backend-as-a-Service), fornece infraestrutura em nuvem para processar e armazenar o back-end dos projetos, além de disponibilizar uma API própria para a interação e manipulação de dados de forma segura. O Firebase Authentication oferece autenticação para os usuários das aplicações e é integrável com React. Por sua vez, os serviços Cloud Firestore e Realtime Database são responsáveis pela manipulação de dados dinâmicos em tempo real.

Para a exibição e manipulação de dados geoespaciais, o principal recurso utilizado foi a plataforma Google Maps. Acessível por meio de API, foi utilizada para a criação de mapas dinâmicos diretamente em componente React, além de sugerir rotas para o deslocamento veicular. Por ser desenvolvida pela mesma empresa responsável pelo Firebase, a Google, é altamente integrável para o desenvolvimento de aplicações que fazem uso de dados em tempo real. Por fim, também possui *SDK* próprio para a linguagem JavaScript, tornando viável o desenvolvimento por meio de TypeScript.

O gerenciamento de estado da plataforma web foi realizado por meio de *hooks* customizados baseados em React, como useRotas, useVanPosition, useGoogleMaps, useGooglePlaces, useEmpresaId e useFirebase. Esses *hooks* encapsulam lógicas específicas e promovem reutilização de código, facilitando a manutenção e clareza do projeto.

Definidas as tecnologias, a arquitetura da plataforma web foi idealizada com foco na organização e escalabilidade, aproveitando os recursos oferecidos pelo framework Next.js. Essa organização foi viabilizada por meio de uma estrutura de pastas clara e pela adoção do novo sistema de roteamento App Router, introduzido nas versões mais recentes do framework.

A aplicação possui a seguinte estrutura de diretórios, conforme a convenção do Next.js:

- **/app:** definição das rotas e páginas da aplicação;
- **/components:** componentes reutilizáveis e responsáveis pela padronização da interface, como caixas de alerta, botões e outros elementos visuais;
- **/hooks:** hooks personalizados do React, utilizados para abstrair lógicas específicas, como autenticação e manipulação de mapas;
- **/services:** módulo responsável pela comunicação com APIs de serviços externos, como Firebase e Google Maps;
- **/types:** definições de tipos e interfaces TypeScript utilizadas na modelagem de dados;

- **/lib:** abriga funções utilitárias, como validadores, constantes de configuração e formataadores;
- **/styles:** arquivos de configuração e estilização utilizando Tailwind CSS.

A organização interna do código contemplou a definição de interfaces específicas para entidades como usuários, motoristas, passageiros, rotas e veículos, em conformidade com os modelos desenvolvidos na etapa de diagramação. Essa padronização com tipos tipados contribuiu para a prevenção de erros e para o consumo eficiente desses dados por outros módulos da aplicação.

Outro aspecto relevante na arquitetura é a separação entre o processamento do lado do cliente e do servidor, característica nativa do Next.js. No cliente, são processadas funcionalidades como visualização de mapas em tempo real, localização de veículos e confirmação de presença de passageiros, o que proporciona uma interface mais responsiva e reduz a carga no servidor.

Já o processamento do lado do servidor inclui tarefas críticas como autenticação de usuários por meio do Firebase Authentication e validação de regras de negócio. Esse modelo contribui para a segurança da aplicação, uma vez que limita a exposição de dados sensíveis e reduz a superfície de ataque.

A estrutura de roteamento dinâmico do App Router foi essencial para a organização de recursos hierárquicos, como evidenciado na estrutura `/app/api/empresas/[id]/rotas/[rotaId]`. Essa configuração permitiu a criação de endpoints RESTful parametrizados, otimizando o gerenciamento de rotas e motoristas por empresa.

Além disso, foi utilizado o conceito de `intercepting routes` para exibição de formulários modais sem alteração da URL principal, como no caso da rota `/app/dashboard/rotas/@modal/(.)criar/page.tsx`.

A gestão de estado da aplicação foi implementada com hooks personalizados como `useRotas`, `useVanPosition`, `useGoogleMaps`, `useGooglePlaces`, `useEmpresaId` e `useFirebase`, os quais encapsulam lógica de negócio reutilizável e comunicação com APIs externas.

A integração com serviços externos como Firebase e Google Maps foi realizada por meio de abstrações modulares. Essa organização permitiu a aplicação de estratégias de fallback e tratamento de exceções, mantendo a resiliência da aplicação mesmo em cenários de instabilidade dos serviços.

5.2.4. Aplicações Mobile

O desenvolvimento mobile foi facilitado com o uso do Expo (EXPO, 2025), uma ferramenta *framework* que simplifica o uso do React Native ao oferecer bibliotecas nativas já pré-configuradas, ambiente de desenvolvimento integrado e suporte a builds nativos sem a necessidade de configuração manual de ambientes. Um dos principais benefícios do Expo é permitir o desenvolvimento de aplicações para iOS mesmo em ambientes que não utilizam o ecossistema Apple, o que seria uma limitação no desenvolvimento com React Native puro.

Para gerenciar a navegação entre telas nas aplicações *mobile*, foi utilizada a biblioteca React Navigation (REACT NAVIGATION, 2025), que oferece suporte a diversos padrões de navegação como `stack`, `tabs` e `drawer`, mantendo uma experiência fluida e próxima da nativa em ambas as plataformas.

A exibição de mapas nas aplicações *mobile* foi implementada com a biblioteca React Native Maps (AIRBNB, 2025), que integra os mapas do Google Maps diretamente em componentes do React Native, permitindo visualizações e interações geoespaciais semelhantes às da plataforma web.

O gerenciamento de estado nas aplicações *mobile* foi realizado com o uso da Context API do React, que permitiu compartilhar dados de autenticação e contexto entre componentes. Além disso, foram criados hooks personalizados para encapsular lógica de localização, autenticação e controle de rotas, promovendo organização e reutilização de código.

A arquitetura das aplicações *mobile*, voltadas para motoristas e passageiros, foi estruturada considerando os recursos oferecidos pelo React Native e o ecossistema Expo. A ênfase esteve na clareza da estrutura, separação de responsabilidades e reutilização de lógica entre telas e funcionalidades.

As aplicações adotam a seguinte estrutura de diretórios no padrão `/src`:

- **/src/components**: componentes reutilizáveis responsáveis pela padronização da interface, como cards de rota, botões de confirmação e elementos visuais;
- **/src/hooks**: hooks personalizados do React, utilizados para encapsular lógicas como gerenciamento de rotas, localização e interação com Firebase;
- **/src/screens**: telas organizadas por funcionalidade (Login, Home, Detalhes da Rota, Mapa, etc.);
- **/src/context**: gerenciamento de estado global via Context API, com destaque para autenticação de usuários;
- **/src/config**: arquivos de configuração para integração com Firebase e Google Maps;
- **/src/types**: definições de tipos e interfaces TypeScript utilizadas na modelagem dos dados;
- **/src/styles**: arquivos de estilização com uso do StyleSheet do React Native.

Assim como na plataforma web, o projeto foi implementado com interfaces tipadas para representar entidades como usuários, rotas, pontos de parada e veículos. Essa modelagem padronizada contribuiu para a consistência dos dados e facilitou a manutenção.

A arquitetura também reforça a separação entre lógica de negócio e apresentação. A lógica foi encapsulada em hooks reutilizáveis, enquanto a interface foi desenvolvida com componentes modulares e reutilizáveis, garantindo coerência visual e eficiência no desenvolvimento.

Entre as funcionalidades das aplicações *mobile* destacam-se:

- Visualização de mapas em tempo real com React Native Maps;
- Rastreamento da localização dos veículos via Firebase Realtime Database;
- Confirmação de presença dos passageiros nos pontos de parada.

Essas funcionalidades são processadas localmente no dispositivo móvel, proporcionando uma experiência mais fluida e com suporte offline em caso de perda temporária de conexão.

A comunicação com serviços externos foi realizada por meio de SDKs nativos fornecidos pelo Firebase e pelo Google Maps. A utilização do Expo facilitou essa integração ao fornecer uma camada de abstração sobre APIs específicas de cada plataforma (Android e iOS), eliminando, inclusive, a necessidade de acesso a hardware da Apple para desenvolver para iOS, que se mostra uma vantagem significativa em ambientes de desenvolvimento multiplataforma.

5.3. Desenvolvimento da Aplicação Web

A plataforma web foi desenvolvida com o framework Next.js, utilizando renderização híbrida (client-side e server-side) conforme a necessidade de cada funcionalidade. As rotas foram definidas utilizando o sistema de arquivos da pasta /app, e os endpoints internos foram estruturados na forma de rotas API, como em /app/api/empresas/[id]/rotas/[rotaId].

Foram implementadas funcionalidades como:

- Cadastro e edição de motoristas, passageiros, rotas e empresas;
- Visualização da rota em tempo real por meio da API do Google Maps;
- Reordenação de pontos de parada com drag-and-drop;
- Gerenciamento de acessos por meio de autenticação com Firebase Authentication;
- Exibição de dados dinâmicos, como a posição da van e o status dos passageiros confirmados.

A autenticação foi controlada por middleware no lado do servidor, que verifica tokens do Firebase antes de liberar o acesso às páginas protegidas. Hooks personalizados foram criados para encapsular lógicas como busca de rotas, integração com serviços externos e gerenciamento de estado local.

5.4. Desenvolvimento dos Aplicativos Mobile

Os aplicativos mobile foram implementados com React Native e Expo, permitindo compatibilidade com dispositivos Android e iOS a partir de uma única base de código. As aplicações para motoristas e passageiros compartilham parte da estrutura, mas apresentam interfaces e funcionalidades específicas para cada público.

As principais funcionalidades do aplicativo do motorista incluem:

- Visualização da lista de passageiros com confirmação de presença;
- Acesso aos pontos de parada organizados conforme os passageiros confirmados;
- Rastreamento em tempo real da posição do veículo, atualizada via Realtime Database.

Já o aplicativo do passageiro contempla:

- Confirmação de presença no dia da viagem;
- Visualização da rota e localização da van em tempo real;
- Acesso aos dados do motorista responsável pela rota.
- A navegação entre as telas foi implementada com a biblioteca React Navigation.

A comunicação com os serviços externos, como Firebase e Google Maps, foi realizada com SDKs nativos integrados ao Expo, o que simplificou a configuração dos ambientes e permitiu o desenvolvimento para iOS sem necessidade de hardware Apple.

6. Resultados do Desenvolvimento

Nesta seção apresentam-se os resultados obtidos com a implementação da plataforma, considerando a plataforma web voltada à empresa fretadora e os aplicativos móveis destinados a motoristas e passageiros. Todas as funcionalidades principais descritas no levantamento de requisitos foram testadas, com destaque para autenticação de usuários, visualização de rotas, confirmação de presença e sincronização em tempo real.

6.1. Autenticação e Cadastro

O processo de autenticação mostrou-se funcional em todos os perfis de usuário (empresa, motorista e passageiro). O controle de acesso foi validado por meio do uso de roles configurados no Firebase, garantindo que cada usuário tivesse acesso apenas às funcionalidades específicas de seu papel. A Figura 10 apresenta exemplos de usuários cadastrados, juntamente com suas informações básicas e respectivas roles.

Nome	Role	Email	UID	Data de Criação	Último Acesso	Custom Claims
Lucas Lutz	ADMINISTRADOR	lucaslutz98@gmail.com	dTr23uClZKgrehNNwjnW2B7bn BS2	16/03/2025	10/11/2025	{ "empresaId": "HvbN1e0y36ykkJYQchUy", "role": "admin" }
José	MOTORISTA	jose@motorista.com	qPPMPAj8NCbW1is2GsauKFG5S ID3	10/11/2025		{ "role": "motorista", "empresaId": "HvbN1e0y36ykkJYQchUy" }
Lucas	PASSAGEIRO	lucas@passageiro.com	DfCo9dzmyyPGYAeWYMxPFWz AVOS2	10/11/2025		{ "role": "passageiro", "empresaId": "HvbN1e0y36ykkJYQchUy" }

Figura 10. Usuários do sistema e suas respectivas roles

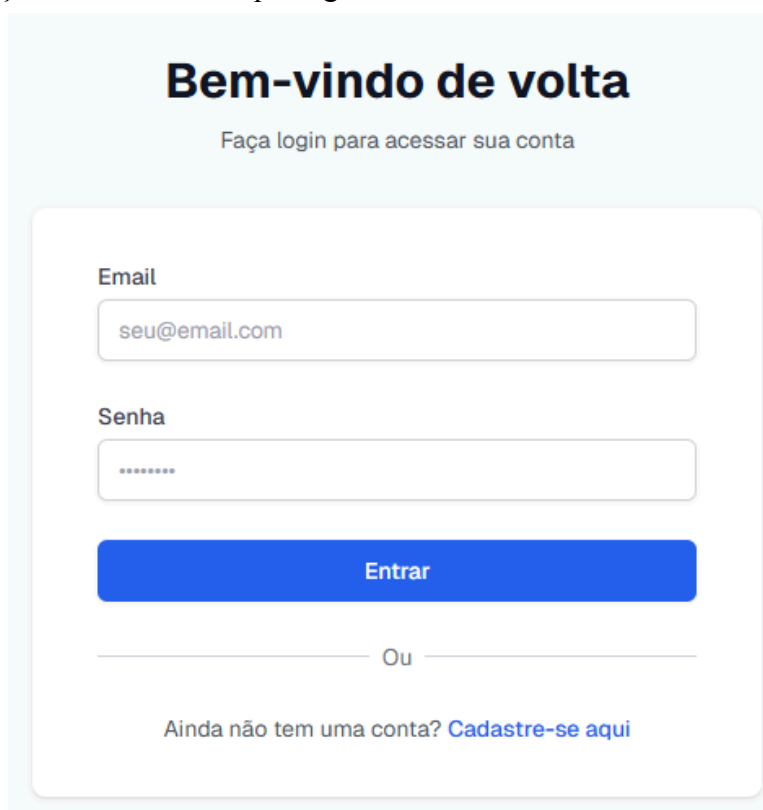
Fonte: Autoria Própria.

6.2. Plataforma web

A plataforma web possibilitou o cadastro, edição e exclusão de motoristas, passageiros, rotas e veículos. Além disso, permitiu a visualização em tempo real da posição das vans, a gestão de rotas ativas e inativas e a reordenação de pontos de parada de forma prática. Essa centralização contribuiu para uma visão global do sistema e melhor controle das operações.

A seguir são apresentadas as principais telas da aplicação web desenvolvida, utilizadas pela empresa para o gerenciamento de rotas, motoristas, vans e passageiros.

A Figura 11 apresenta o login, que é a tela inicial do sistema, onde é necessário informar o e-mail e a senha para autenticação do usuário. Segue-se o mesmo padrão para as aplicações de motorista e passageiro.



A imagem mostra a tela de login da aplicação web. No topo, há o título "Bem-vindo de volta" em uma fonte grande e escura, seguido pelo subtítulo "Faça login para acessar sua conta" em uma fonte menor e cinza. Abaixo, há um formulário com dois campos de entrada: "Email" com o texto "seu@email.com" e "Senha" com pontos para ocultar o conteúdo. Um botão azul com o texto "Entrar" está posicionado abaixo dos campos. Abaixo do botão, há o texto "Ou" flutuando entre duas linhas horizontais. No rodapé do formulário, há o link "Ainda não tem uma conta? Cadastre-se aqui" em azul.

Figura 11. Tela de Login da aplicação web

Fonte: Autoria Própria.

A Figura 12 evidencia o painel de gerência de rotas, o qual permite à empresa visualizar todas as rotas cadastradas, exibindo o trajeto no mapa juntamente com os pontos de embarque e desembarque dos passageiros e a localização em tempo real da van. Além disso, a interface permite otimizar a rota, gerando automaticamente o traçado com base no ponto inicial, nos pontos de parada e no ponto final, calculando também a distância total percorrida.

Rotas Vans Motoristas Passageiros

Gerenciar Rotas

[Nova Rota](#)

Lista de Rotas

Rota Noite Farroupilha [Editar](#) [Excluir](#)

Motorista: Francisco
 Van: abc-1234
 Horário: 19:00 - 21:00
 Status: Pendente
 Pontos de Parada - Ida: 2/2
 Pontos de Parada - Volta: 2/2
 Total de Passageiros: 2

Rota Manhã Farroupilha [Editar](#) [Excluir](#)

Motorista: Francisco
 Van: cba-4321
 Horário: 09:00 - 10:30
 Status: Concluída
 Pontos de Parada - Ida: 2/2
 Pontos de Parada - Volta: 2/2
 Total de Passageiros: 2

Rota Tarde Farroupilha ● Em Execução [Editar](#) [Excluir](#)

Motorista: José
 Van: abc-1234
 Horário: 13:00 - 14:00
 Status: Em Andamento
 Pontos de Parada - Ida: 2/2
 Pontos de Parada - Volta: 2/2
 Total de Passageiros: 2

Visualização no Mapa

[Otimizar Rota](#)

Google Atalhos do teclado Dados cartográficos ©2025 Termos Informar erro no mapa

Localização da Van em Tempo Real

Van: **abc-1234** Última atualização: 10:41:25

Coordenadas: -29.215956, -51.334605

Rota Otimizada

Distância total:	Duração estimada:
27.3 km	43 min
<small>(27.341 metros)</small>	<small>(42 minutos e 42 segundos)</small>

Informação: Rota otimizada considerando apenas passageiros confirmados (6 pontos)

Figura 12. Tela de gerência de rotas (dashboard principal)

Fonte: Autoria Própria.

A Figura 13 ilustra a tela de criação e edição de rotas, na qual é possível cadastrar ou alterar informações como o motorista responsável pela rota, a van utilizada, os horários de início e término, o status da rota e os pontos de embarque e desembarque dos passageiros.

Editar Rota

Nome
Rota Tarde Farroupilha

Motorista
José

Van
abc-1234 - Sprinter

Horário de Início
13:00

Horário de Fim
14:00

Status
Em Andamento

Ponto de Embarque Único?
Não - Cada passageiro tem seu destino

Cancelar Definir Pontos no Mapa Atualizar

Selecione um passageiro

Cancelar Salvar Rota

Pontos da Rota

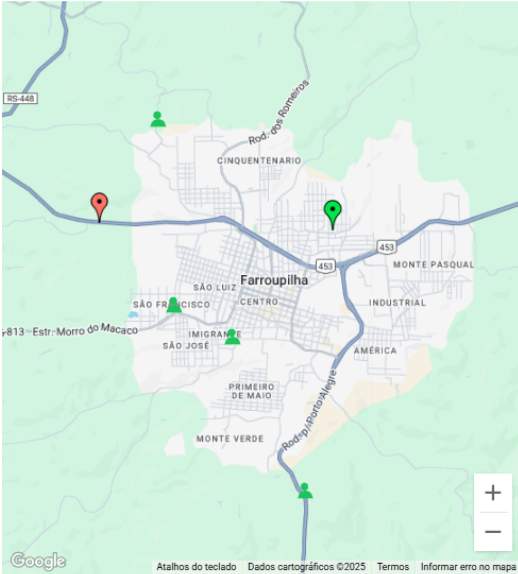
Ponto Inicial da Rota Remove
-29.216017, -51.334989
✓ Editar Ponto Inicial

Ponto Final da Rota Remove
-29.214812, -51.376854
✓ Editar Ponto Final

Passageiros da Rota

Thiago Remove
farroupilha - RS
✓ Editar Embarque ✓ Editar Destino

Mateus Remove
Farroupilha - RS
✓ Editar Embarque ✓ Editar Destino



Mapa de Farroupilha, RS, mostrando a rota planejada com pontos de embarque e destino. O mapa inclui rotas locais como Rod. dos Formosos e Rod. Farroupilha, e áreas como Cinquentenário, Centro, Industrial, América, Monte Pasqual, São José, Imigrante, São Francisco, São Luiz, Primeiro de Maio, Monte Verde e Estr. Morro do Macaco. Há marcadores de pontos de embarque e destino em verde e um marcador de ponto inicial em vermelho.

Figura 13. Tela de criação e edição de rotas

Fonte: Autoria Própria.

A Figura 14 apresenta o painel de gerenciamento de passageiros, onde é possível cadastrar, editar e excluir registros. São exibidas informações como nome, e-mail, telefone, endereço e status (ativo ou inativo).

Passageiros						Novo Passageiro
NOME	EMAIL	TELEFONE	ENDEREÇO	STATUS	AÇÕES	
Thiago	thiago@passageiro.com	52424252	farroupilha - RS	ativo	Editar	Excluir
Mateus	mateus@passageiro.com	252524242	Farroupilha - RS	ativo	Editar	Excluir
Lucas	lucas@passageiro.com	123456798	Farroupilha - RS	ativo	Editar	Excluir

Figura 14. Tela de gerência de passageiros (dashboard principal)

Fonte: Autorial Própria.

A Figura 15 mostra o painel de gerenciamento de vans, no qual são listadas as vans cadastradas com informações como placa, modelo e capacidade de passageiros, além das opções para edição e exclusão de registros.

Vans				Nova Van
PLACA	MODELO	CAPACIDADE	AÇÕES	
cba-4321	Sprinter	10 passageiros	Editar	Excluir
abc-1234	Sprinter	10 passageiros	Editar	Excluir

Figura 15. Tela de gerência de vans (dashboard principal)

Fonte: Autorial Própria.

Por fim, a Figura 16 apresenta o painel de gerenciamento de motoristas, exibindo dados como nome, número da CNH, telefone, e-mail e status de atividade, com a possibilidade de criar, editar ou remover registros.

Motoristas						Novo Motorista
NOME	CNH	TELEFONE	EMAIL	STATUS	AÇÕES	
Francisco	123456789	123456789	francisco@motorista.com	ativo	Editar	Excluir
José	123456789	12345679	jose@motorista.com	ativo	Editar	Excluir

Figura 16. Tela de gerência de motoristas (dashboard principal)

Fonte: Autorial Própria.

6.3. Aplicativo para Passageiros

O aplicativo destinado aos passageiros demonstrou funcionamento consistente em relação às principais funcionalidades: confirmação de presença, acompanhamento da van em tempo real e consulta aos dados do motorista responsável pela rota. As

notificações de alerta sobre o início da rota e a solicitação de confirmação também foram exibidas corretamente aos usuários. Entretanto, por limitação da versão básica do Expo Go, as notificações são apresentadas apenas quando o aplicativo está aberto. A implementação atual permite evoluções futuras, como a integração de notificações push por meio do Firebase, o que poderá aprimorar a experiência dos usuários ao fornecer notificações mesmo sem a aplicação estar em execução.

A seguir são apresentadas as principais telas da aplicação móvel destinada aos passageiros. O aplicativo foi desenvolvido com foco na usabilidade, permitindo ao usuário acompanhar suas rotas, receber notificações e visualizar o trajeto em tempo real.

A Figura 17 apresenta a tela de permissões exibida ao iniciar o aplicativo pela primeira vez. Nela, o sistema solicita ao usuário autorização para o uso de notificações e serviços de localização, essenciais para o funcionamento da aplicação. Ao selecionar a opção “Permitir”, é aberta a janela de controle padrão do sistema operacional (Android ou iOS), solicitando a concessão das permissões correspondentes.

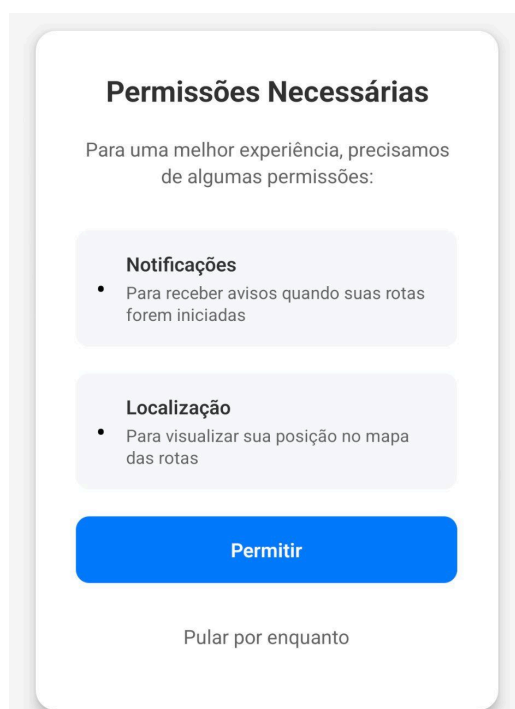


Figura 17. Tela de permissões necessárias para uso da aplicação

Fonte: Autoria Própria.

A Figura 18 exhibe a tela do menu principal, apresentada logo após o processo de login (o qual possui interface idêntica à descrita na Figura 11). Nessa tela, são listadas as rotas às quais o passageiro logado está associado, com informações sobre o horário de início e término, motorista responsável e status (ativa, pendente ou finalizada). Quando a rota está ativa, o sistema habilita o botão “Visualizar no mapa”, que permite acompanhar o deslocamento do transporte em tempo real.

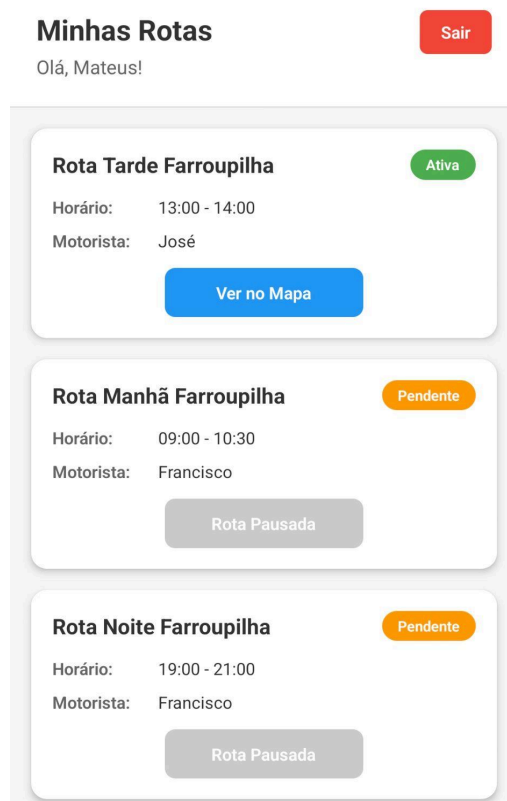


Figura 18. Tela do menu principal do passageiro

Fonte: Autoria Própria.

A Figura 19 apresenta a tela de mapa da rota, que exibe a localização em tempo real da van, os pontos de embarque e desembarque e as informações principais da rota, como nome, horário, motorista e status. Além disso, a interface conta com uma legenda que identifica os ícones utilizados no mapa, permitindo ao usuário distinguir visualmente sua posição, a posição da van e os demais pontos de parada do trajeto.

[← Voltar](#) **Mapa da Rota**

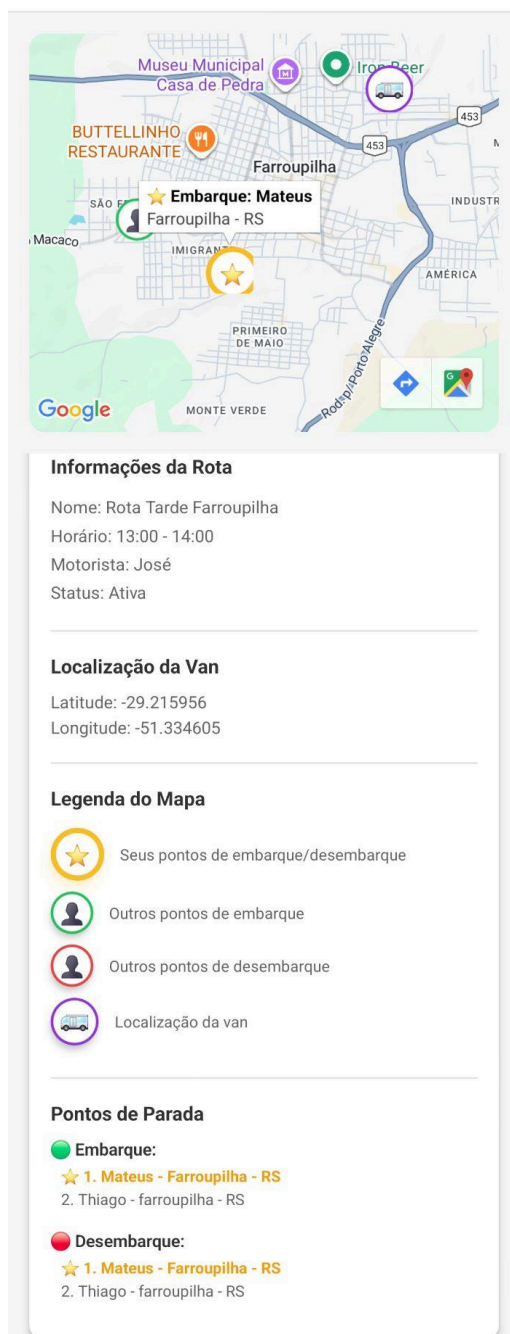


Figura 19. Mapa da rota com localização e pontos de embarque e desembarque.

Fonte: Autoria Própria.

A Figura 20 exemplifica uma notificação recebida pelo passageiro quando uma rota associada a ele é iniciada pelo motorista. Essa funcionalidade visa manter o usuário informado em tempo real sobre o status da viagem, reforçando a integração entre as aplicações de motorista e passageiro.

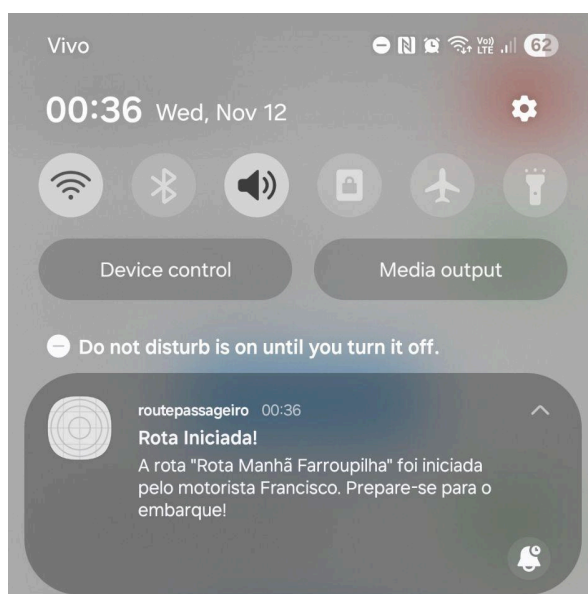


Figura 20. Notificação de início de rota recebida pelo passageiro

Fonte: Autoria Própria.

6.4. Aplicativo para Motoristas

No aplicativo voltado aos motoristas, as funcionalidades de iniciar e finalizar rotas, visualizar passageiros confirmados e acessar pontos de parada organizados foram validadas com êxito. Assim como na versão destinada aos passageiros, o sistema de notificações é exibido apenas quando o aplicativo está aberto, podendo ser aprimorado futuramente com o uso de notificações *push* integradas ao Firebase.

A seguir são apresentadas as principais telas da aplicação móvel desenvolvida para os motoristas.

A Figura 21 apresenta a tela inicial da aplicação, exibida logo após o login do motorista. Nela são listadas as rotas sob sua responsabilidade, apresentadas em formato de cartões com informações como nome da rota, status (ativa ou inativa), horários de início e término, além da quantidade de pontos de parada. Cada rota conta com um botão “Ver detalhes”, que direciona o motorista para a tela de visualização da rota, apresentada na Figura 22.

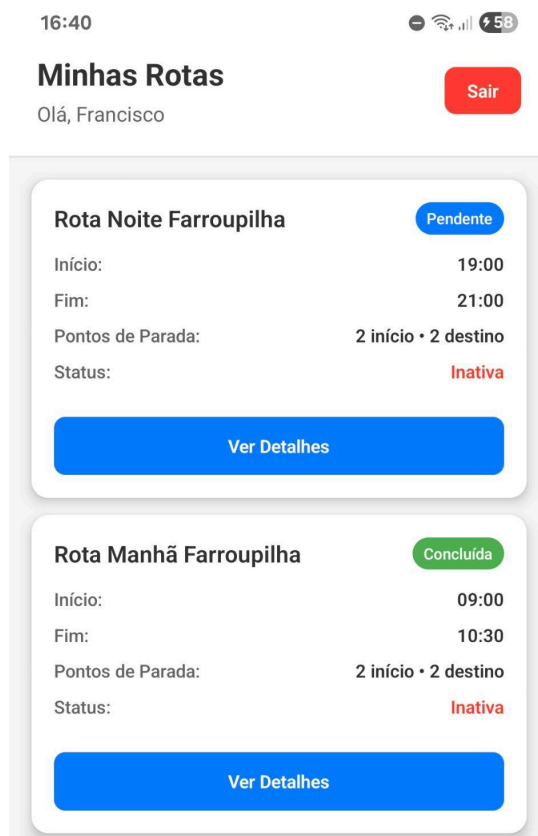


Figura 21. Tela inicial do aplicativo de motorista

Fonte: Autoria Própria.

A Figura 22 mostra a tela de detalhes da rota, onde é apresentado o mapa com a localização atual do motorista e os pontos de parada correspondentes ao trajeto. O aplicativo também oferece a opção de exibir ou ocultar uma sugestão de traçado entre os pontos da rota, auxiliando na navegação. Nessa tela está disponível o botão “Iniciar rota”, que permite ao motorista dar início ao percurso programado.

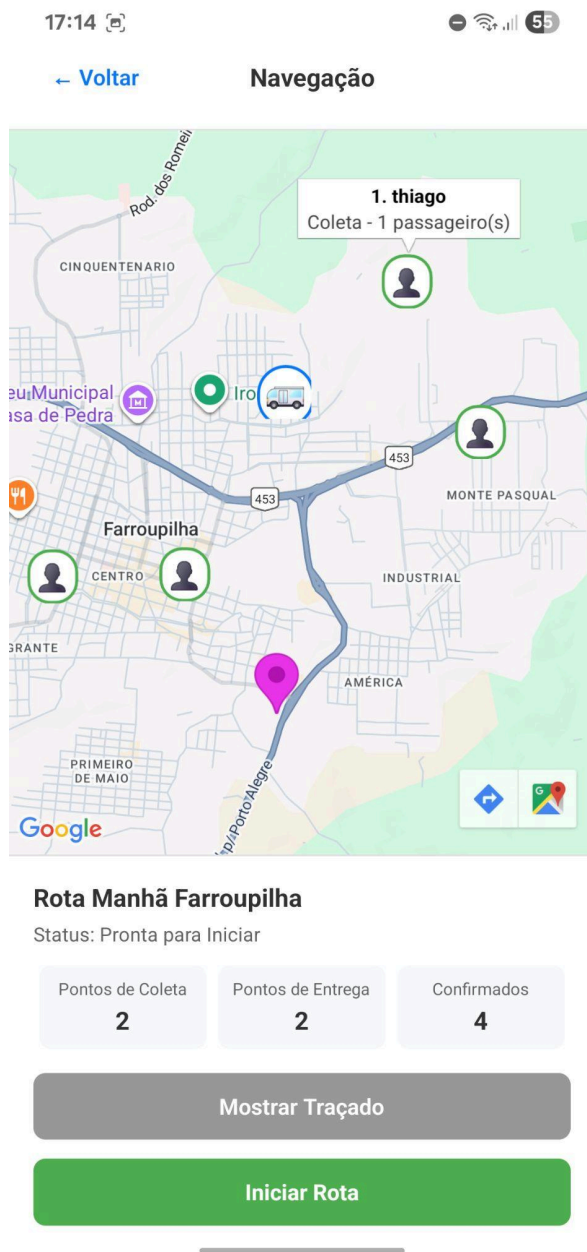


Figura 22. Mapa da rota e detalhes do trajeto

Fonte: Autoria Própria.

A Figura 23 apresenta a tela exibida após o início da rota. Assim como a anterior, ela contém o mapa da rota, porém destaca o próximo ponto que o motorista deve alcançar. Também estão disponíveis as opções de “Finalizar rota” e de ocultar o traçado sugerido. Essa interface foi desenvolvida para oferecer ao motorista uma experiência de acompanhamento mais intuitiva durante a execução do trajeto.

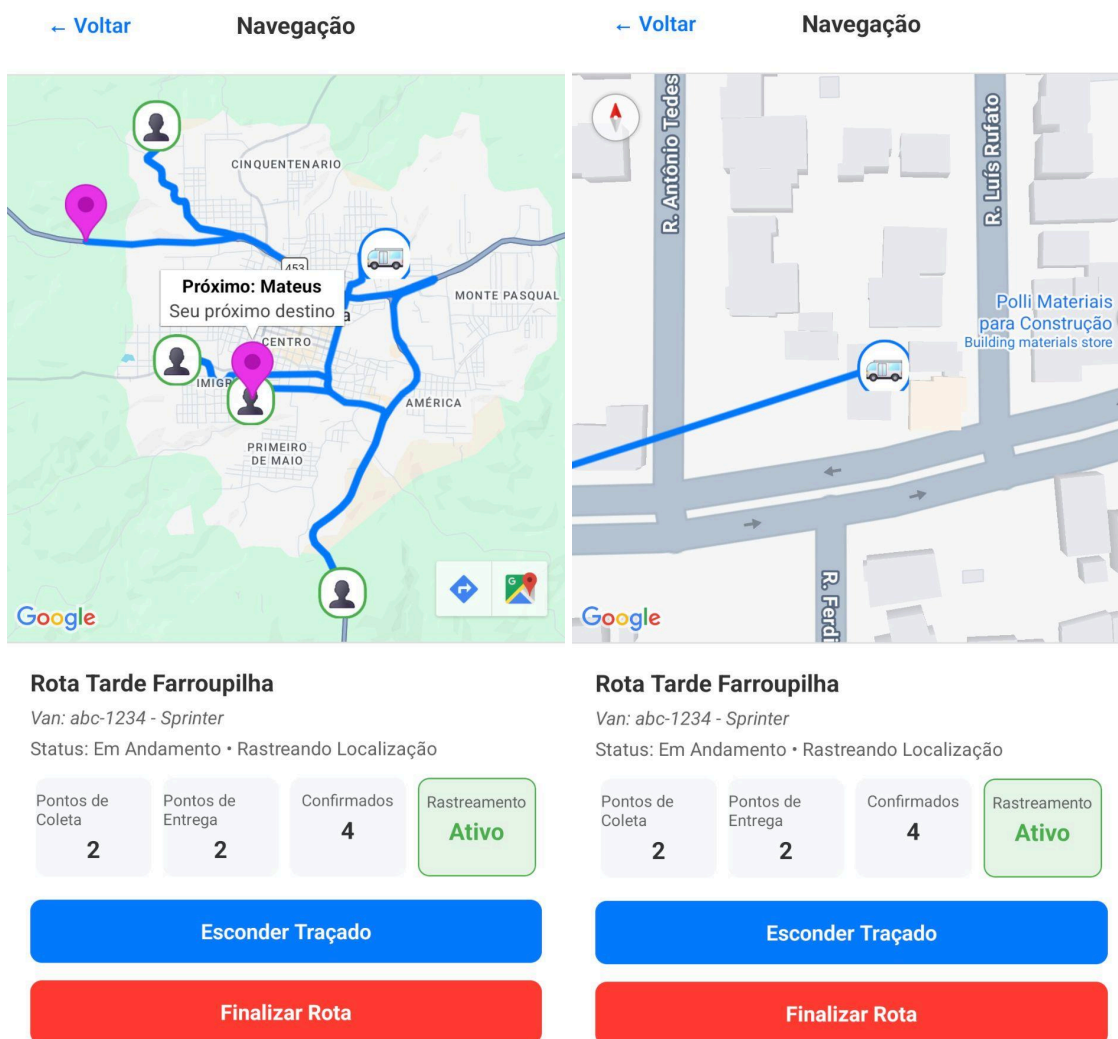


Figura 23. Tela de rota em andamento com próximo ponto de parada

Fonte: Autoria Própria.

Além disso, identificou-se a possibilidade de expansão da aplicação com a adição de um módulo de navegação “turn-by-turn”, semelhante ao oferecido por aplicativos como Waze e Google Maps. Tal funcionalidade não foi implementada, pois a API do Google Maps utilizada no projeto não fornece suporte nativo a esse tipo de navegação. No entanto, por se tratar de uma aplicação de código aberto, o sistema está preparado para futuras integrações com outras APIs de navegação que viabilizem uma experiência mais completa e interativa para o motorista.

6.5. Sincronização em Tempo Real e Experiência de Uso

A integração entre os três módulos demonstrou desempenho satisfatório. Alterações de rota, atualização de posição dos veículos e confirmações de presença refletiram-se em tempo real tanto na plataforma web quanto nos aplicativos móveis, confirmando a eficácia do uso conjunto do Firestore e Realtime Database.

A interface construída com React, React Native e TailwindCSS apresentou boa responsividade e compatibilidade com diferentes dispositivos. A experiência do usuário se mostrou satisfatória, principalmente em relação à visualização do mapa em tempo

real e à simplicidade no cadastro e confirmação de presença. Dessa forma, os resultados obtidos confirmam o funcionamento das principais funcionalidades propostas.

7. Considerações Finais

Este trabalho teve como proposta o desenvolvimento de uma plataforma *open source* para transporte coletivo privado, composta por três módulos interligados: uma plataforma web para a empresa fretadora, um aplicativo móvel para passageiros e outro para motoristas. A solução buscou integrar, em tempo real, informações de rotas, passageiros e veículos, com o objetivo de aprimorar a comunicação entre os atores envolvidos e permitir maior transparência no processo que envolve a logística deste tipo de fretamento.

Os resultados obtidos demonstraram que a plataforma contempla funcionalidades essenciais para o gerenciamento de transporte fretado, incluindo autenticação de usuários, cadastro e edição de rotas, visualização da posição das vans em tempo real, confirmação de presença dos passageiros e visualização destas por meio dos mapas das rotas. A integração entre os três módulos mostrou-se consistente e capaz de proporcionar uma experiência unificada.

Ainda que os testes tenham ocorrido em ambiente controlado, a plataforma tem a capacidade de melhorar a comunicação entre gestores, motoristas e passageiros, além de contribuir para a redução de custos operacionais.

Em relação às limitações, algumas funcionalidades não foram implementadas nesta versão. O envio de notificações push sem a aplicação estar em primeiro plano nos dispositivos móveis, por exemplo, permanece como um recurso a ser desenvolvido em trabalhos futuros. Além disso, a navegação no aplicativo do motorista, embora funcional na exibição dos pontos de parada e da rota, não é totalmente intuitiva. Para aprimoramento, sugere-se a implementação de um sistema de navegação turn-by-turn, o qual não é disponibilizado de forma nativa pela API do Google Maps, mas que poderia fornecer uma experiência mais clara e agradável durante a condução.

Devido ao caráter *open source* da plataforma, a disponibilização do código-fonte em repositório público possibilita que pesquisadores, empresas e desenvolvedores possam não apenas utilizar, mas também adaptar e expandir a solução, fomentando trabalhos colaborativos e assegurando a continuidade da evolução da ferramenta. Nesse sentido, por ser um sistema de código aberto e com alto potencial de integração a demais soluções, abre-se espaço para que futuros trabalhos possam complementar o sistema com novas funcionalidades, como integração com meios de pagamento para facilitar a gestão financeira, extensão da plataforma para cenários de uso público ou comunitário e implementação de relatórios de consumo de combustível estimado, tempo médio de viagem e histórico de presença. Ressalta-se que tais possibilidades não compõem as premissas originais deste trabalho, mas representam caminhos viáveis para evolução futura da plataforma a partir de iniciativas da comunidade ou de outras pesquisas.

Dessa forma, entende-se que a plataforma proposta possui potencial de oferecer uma solução tecnológica viável e escalável para o transporte coletivo privado. Com os aprimoramentos futuros indicados, espera-se que a ferramenta possa ser aplicada em cenários reais, trazendo benefícios tanto para empresas fretadoras quanto para os passageiros atendidos.

Referências

- AIRBNB (2025). “React Native Maps”. Disponível em: <https://www.npmjs.com/package/react-native-maps>. Acesso em: 20 Ago, 2025.
- Associação Nacional das Empresas de Transporte (2023). “Como anda o transporte coletivo brasileiro?”. Disponível em: <https://www.ntu.org.br/novo/ckfinder/userfiles/files/Sociedade61.pdf>. Acesso em: 28 Mar, 2025.
- CITYMAPPER (2025). “Citymapper”. Disponível em: <https://citymapper.com/?lang=pt-br>. Acesso em: 28 Mar, 2025.
- EXPO (2025). “Introduction”. Disponível em: <https://docs.expo.dev/get-started/introduction/>. Acesso em: 11 Mai, 2025.
- Figueiredo, L. M. B. (2005) “Sistemas Inteligentes de Transporte”. Faculdade de Engenharia, Universidade do Porto, Porto. 241 p. Disponível em: http://ave.dee.isep.ipp.pt/~gris/_private/Teses/Thesis_PhD_Lino2005.pdf. Acesso em: 20 mar, 2025.
- Firebase (2025). “Firebase: Build apps and grow businesses”. Disponível em: <https://firebase.google.com/products-build?hl=pt-br>. Acesso em: 11 Mai, 2025.
- Firebase Firestore (2025). “Firestore NoSQL cloud database”. Disponível em: <https://firebase.google.com/products/firestore?hl=pt-br>. Acesso em: 11 Mai, 2025.
- Firebase Realtime Database (2025). “Real-time database by Firebase”. Disponível em: <https://firebase.google.com/products/realtime-database?hl=pt-br>. Acesso em: 11 Mai, 2025.
- Gao, Y. and Zhu, J. (2022) “Characteristics, impacts and trends of urban transportation”, Encyclopedia, Basel, v. 2, n. 2, p. 1168–1182, June 15. Disponível em: <https://www.mdpi.com/2673-8392/2/2/78>. Acesso em: 20 Mar, 2025. DOI: <https://doi.org/10.3390/encyclopedia2020078>.
- Google Maps Platform (2025). “Google Maps Platform: Build with comprehensive location data”. Disponível em: <https://mapsplatform.google.com/>. Acesso em: 11 Mai, 2025.

Levin, S. S. and Prietula, M. J. (2014) “Open Collaboration for Innovation: Principles and Performance”, *Organization Science*, v. 25, n. 5, p. 1414–1433. Disponível em: <https://arxiv.org/abs/1406.7541>. Acesso em: 18 Ago, 2025. DOI: <https://doi.org/10.48550/arXiv.1406.7541>.

META (2025). “React: The library for web and native UI”. Disponível em: <https://react.dev/>. Acesso em: 11 Mai, 2025.

MICROSOFT (2025). “TypeScript: JavaScript with syntax for types”. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 11 Mai, 2025.

MINISTÉRIO DOS TRANSPORTES (2024). “Frota de veículos 2024”. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-Senatran/frota-de-veiculos-2024>. Acesso em: 28 mar, 2025.

MOOVIT (2024). “Índice sobre o Transporte Público”. Disponível em: https://moovitapp.com/insights/pt-br/Moovit_Insights_%C3%8Dndice_sobre_o_Transporte_P%C3%BAblico-countries. Acesso em: 28 Mar, 2025.

MOOVIT (2025). “Aplicativo Moovit”. Disponível em: <https://moovit.com/pt/moovit/>. Acesso em: 28 Mar, 2025.

MOZILLA FOUNDATION (2025). “JavaScript”. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 14 Jun, 2025.

MURBI (2025). A revolução digital do transporte fretado. Disponível em: <https://murbi.com.br/>. Acesso em: 28 Mar, 2025.

Next.js (2025). “Next.js: The React framework”. Disponível em: <https://nextjs.org/>. Acesso em: 11 Mai, 2025.

Next.js (2025). “Next.js showcase projects”. Disponível em: <https://nextjs.org/showcase>. Acesso em: 11 mai, 2025.

Openjs Foundation (2025). “ESLint”. Disponível em: <https://eslint.org/>. Acesso em: 11 Mai, 2025.

Pradhan, R. P., Arvin, M. B. and Nair, M. (2021) “Urbanization, transportation infrastructure, ICT, and economic growth: A temporal causal analysis”, *Cities*, v. 115. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0264275121001116>. Acesso em: 20 Mar, 2025. DOI: <https://doi.org/10.1016/j.cities.2021.103213>.

React Navigation (2025). “React Navigation”. Disponível em: <https://reactnavigation.org/>. Acesso em: 11 Mai, 2025.

Sun, X. and Zu, Y. (2025) “Research on Fleet Size of Demand Response Shuttle Bus Based on Minimum Cost Method”, *Applied Sciences*, Basel, v. 15, n. 10, p. 5350, 10 maio. Disponível em: <https://www.mdpi.com/2076-3417/15/10/5350>. Acesso em: 12 Out, 2025. DOI: <https://doi.org/10.3390/app15105350>.

Tailwindcss (2025). “Tailwind CSS: Rapidly build modern websites”. Disponível em: <https://tailwindcss.com/>. Acesso em: 11 Mai, 2025.

Tailwind Merge (2025). “CLSX”. Disponível em: <https://www.npmjs.com/package/clsx>. Acesso em: 11 Mai, 2025.