

DOCUMENTAÇÃO TÉCNICA DO PROJETO APLICADO: APLICATIVO DE CARONA SOLIDÁRIA

Projeto Aplicado I

Equipe do Projeto:
Fabiano Carcuchinski Haag
Luana Regina Coelho
Sidnei Avelino da Silva Junior
Marco Aurélio Virtuoso

Professor Responsável:
Iskailer Inaian Rodrigues

Florianópolis - SC
2025

SUMÁRIO




1. EQUIPE E PLANEJAMENTO DAS ATIVIDADES	PÁG. 3
1.1. Nome da Equipe	
1.2. Integrantes e Funções	
1.3. Cronograma do Projeto	
2. PROBLEMA ESCOLHIDO E DESCRIÇÃO	PÁG. 5
2.1. Contextualização	
2.2. Objetivo	
2.3. Justificativa	
2.4. Impacto Esperado	
3. NECESSIDADES DO CLIENTE/USUÁRIO E VALIDAÇÃO	PÁG. 6
3.1. Perfil do Cliente/Usuário	
3.2. Necessidades identificadas e validações	
4. TECNOLOGIAS ESCOLHIDAS E JUSTIFICATIVA	PÁG. 7
4.1. Estrutura	
5. DIAGRAMA E DESCRIÇÃO DA SOLUÇÃO PROPOSTA	PÁG. 10
5.1. Descrição Geral da Solução.....	pág. 10
5.2. Diagrama de Arquitetura.....	pág. 11
5.3. Protótipo da interface	pág. 12
5.4 Repositório Github.....	pág. 16
6. CONCLUSÃO E PRÓXIMOS PASSOS	PÁG. 17
6.1. Conclusão e próximos passos	
6.2 Balanço Final do Projeto (Resultados e Aprendizados)	

1. Equipe e Planejamento das Atividades





1.1 Equipe: “Aplicativo de Carona Solidária”



- **Luana Regina Coelho - Gerente de Projeto e Analista de Sistemas**
 - Responsabilidades: Liderar o planejamento, monitorar o cronograma, garantir a comunicação da equipe, gerenciar a documentação do projeto e traduzir os requisitos do cliente (UnISENAI PR) em especificações técnicas.
- **Fabiano Carcuchinski Haag - Desenvolvedor Back-end e Arquiteto de Banco de Dados.**
 - Responsabilidades: Projetar e implementar a arquitetura do servidor, desenvolver a API, gerenciar o banco de dados e garantir a segurança e a integridade das informações pessoais dos alunos.
- **Sidnei Avelino da Silva Junior - Desenvolvedor Front-end.**
 - Responsabilidades: Desenvolver a interface do aplicativo com base no design de UI/UX, garantir que o aplicativo funcione em várias plataformas e integrar a interface com as funcionalidades do back-end (APIs).
- **Marco Aurélio Virtuoso - Designer de UI/UX (Interface de Usuário e Experiência do Usuário) e Analista de Qualidade (QA)**
 - Responsabilidades: Criar uma interface agradável e intuitiva, desenvolver *wireframes* e protótipos, planejar e executar todos os testes (funcionais, de usabilidade e de segurança) para garantir a qualidade do produto final.

1.2 Integrantes e Funções

 Nome	 Função	 Principais responsabilidades
Fabiano Carcuchinski Haag	Engenharia	Prototipagem
Sidnei Avelino da Silva Junior	Engenharia	Prototipagem
Luana Regina Coelho	Gerenciamento de projetos	Planejamento de projeto
Marco Aurélio Virtuoso	Design	Especificações

1.3 Cronograma do Projeto

 Responsável	 Descrição	 Data Limite	 Finalizado
Todos	Iniciação e Planejamento. Definição de escopo e requisitos	19/08/2025	Sim
Sidnei, Fabiano	Análise e Design. Modelagem do banco	02/09/2025	Sim

 Responsável	Tt Descrição	 Data Limite	Tt Finalizado
	de dados. Criação de diagramas e fluxos esperados. Design de interfaces de usuário.		
Todos	Configuração e Prototipagem. Configuração dos ambientes de desenvolvimento e controle de versão. Validação inicial.	09/09/2025	Sim
Sidnei e Fabiano	Desenvolvimento. Implementação do banco de dados, implementação das regras de negócio e validações de segurança. Codificação das telas principais do aplicativo e componentes de interface.	14/10/2025	Sim
Todos	Integração. Integração do front-end, que é a parte visível e interativa do aplicativo, com a API (back-end, que, nos bastidores, cuida da lógica, dos dados e do servidor). Integração com API do Google Maps para visualização de rotas. Implementação de funcionalidades de segurança.	28/10/2025	Sim
Luana	Acompanhamento das etapas de desenvolvimento, gestão de prazos, validações, comunicação com professor para alinhamentos e monitoramento de check-points.	08/11/2025	Sim
Luana e Marco Aurélio	Testes e validação. Realização de testes unitários e de integração. Execução de testes com um grupo de alunos.	08/11/2025	Sim
Sidnei e Fabiano	Testes e validação. Correção de bugs e ajustes com base no feedback.	08/11/2025	Sim
Todos	Refinamento e Documentação. Ajustes finais de interface. Elaboração da documentação técnica e manual do usuário.	22/11/2025	Sim
Todos	Apresentação e encerramento. Simulação, apresentação final do projeto. Entrega da documentação e código-fonte final.	24/11/2025	Sim

2. Problema Escolhido e Descrição

Demanda da indústria: UniSENAI PR: Aplicativo de Carona Solidária

2.1 Contextualização

Alunos do UniSENAI PR, especialmente as mulheres, enfrentam um desafio significativo de segurança ao retornarem para casa após as aulas, principalmente no período noturno. A percepção de vulnerabilidade e o medo de assédio ou violência no trajeto são preocupações reais para a comunidade acadêmica. Essa insegurança gera consequências diretas na vida acadêmica das estudantes, que muitas vezes se veem forçadas a sair mais cedo de suas atividades ou, em casos extremos, a trancar suas matrículas para evitar o risco. O projeto surge neste contexto como uma solução tecnológica voltada para um problema social e de segurança específico da instituição.

2.2 Objetivo

Desenvolver um aplicativo de carona solidária, seguro e exclusivo para os alunos e funcionários do UniSENAI PR. O objetivo principal é oferecer uma alternativa de transporte confiável que minimize os riscos associados ao deslocamento noturno, com foco especial em criar um ambiente seguro para as alunas, e que seja integrado a ferramentas de geolocalização como o Google Maps.

2.3. Justificativa

A criação de um aplicativo de carona solidária se justifica pela necessidade urgente de prover uma solução que aumente a segurança e a tranquilidade dos estudantes. Atualmente, a falta de alternativas de transporte seguras impacta negativamente na permanência e no bem-estar das alunas. O projeto visa diminuir os riscos de assédio e violência, fortalecendo o senso de comunidade e apoio mútuo entre os membros do UniSENAI do Paraná. Ao restringir o uso apenas a estudantes e funcionários, o aplicativo cria um ambiente de confiança, diferentemente de outras soluções de transporte abertas ao público geral. A solução proposta atende a um requisito fundamental de segurança, permitindo que as alunas possam se dedicar integralmente aos seus estudos sem o medo constante associado ao seu deslocamento.

2.4 Impacto Esperado

A implementação bem-sucedida do aplicativo de carona solidária deverá gerar os seguintes impactos positivos:

- **Aumento da Segurança:** Minimizar os riscos de assédio e violência durante o trajeto de ida e volta da instituição, oferecendo uma alternativa de deslocamento segura e confiável.
- **Melhora na Qualidade de Vida:** Promover mais tranquilidade e bem-estar para os alunos, especialmente para as mulheres, reduzindo o estresse e a ansiedade relacionados ao transporte.

- **Apoio à Permanência Estudantil:** Contribuir para a redução da evasão acadêmica, ao eliminar um dos fatores de risco que levam alunas a trancarem suas matrículas.
- **Facilitação do Deslocamento:** Auxiliar de forma prática e eficiente o deslocamento de toda a comunidade acadêmica, otimizando rotas e horários.
- **Fortalecimento da Comunidade:** Incentivar a colaboração e a solidariedade entre alunos e funcionários, criando uma rede de apoio mútua.

3. Necessidades do Cliente/Usuário e Validação

3.1 O que é esperado:

Esta seção detalha as necessidades primordiais identificadas junto à comunidade do UniSENAI PR, que serviram como base para a definição das funcionalidades do Aplicativo de Carona Solidária. A validação foi realizada por meio de uma abordagem mista, combinando pesquisas quantitativas e qualitativas para garantir que a solução proposta esteja alinhada com as expectativas e demandas reais dos futuros usuários.

3.2 Necessidades identificadas e validações:

- **Perfil do Cliente/Usuário:** O público-alvo do projeto são os alunos e funcionários da instituição UniSENAI PR. Este grupo é caracterizado por indivíduos que buscam soluções de transporte econômicas e, prioritariamente, seguras, especialmente para deslocamentos em horários de baixa movimentação, como o noturno. O foco principal recai sobre as estudantes do sexo feminino, que expressam maior preocupação com a segurança no trajeto de volta para casa.
- **Necessidades Identificadas:**
 - **Necessidade 1:** Necessidade 1: Garantia de um ambiente exclusivo e seguro.
 - * Descrição: Os usuários precisam ter a certeza de que todas as pessoas no aplicativo (motoristas e passageiros) são, de fato, membros da comunidade acadêmica do UniSENAI PR (alunos ou funcionários). Essa exclusividade é o principal fator para a construção da confiança.
 - **Método de Validação:** Foi realizada uma pesquisa online com 120 estudantes do período noturno. Os resultados apontaram que **94%** dos entrevistados consideram "essencial" que o aplicativo seja restrito apenas a membros com vínculo comprovado com a instituição para que se sintam seguros ao utilizá-lo.
 - **Necessidade 2:** Necessidade 2: Opção de carona exclusiva para mulheres.
 - * Descrição: Devido ao problema central de insegurança que afeta principalmente o público feminino, identificou-se a necessidade de uma funcionalidade que permita às mulheres (motoristas e passageiras) optar por viagens compartilhadas apenas com outras mulheres.
 - **Método de Validação:** Conduzimos entrevistas em grupo com 15 alunas. Dessas, 100% afirmaram que a opção de carona "só para mulheres"

aumentaria drasticamente sua sensação de segurança e seria um fator decisivo para a adesão ao aplicativo.

- **Necessidade 3:** Necessidade 3: Monitoramento da viagem e botão de emergência.
 - * Descrição: Os usuários desejam ter recursos de segurança ativos durante o trajeto. Isso inclui a possibilidade de compartilhar a rota em tempo real com contatos de confiança e ter acesso a um "botão de pânico" para acionar ajuda rapidamente em caso de emergência.
 - **Método de Validação:** Em um workshop de ideação com potenciais usuários, a funcionalidade de "compartilhamento de rota" foi a segunda mais votada como item de segurança indispensável. Adicionalmente, uma análise de aplicativos de mobilidade bem-sucedidos no mercado mostrou que recursos de segurança ativos são um padrão da indústria e uma expectativa dos usuários.
- **Necessidade 4:** Necessidade 4: Sistema de avaliação e reputação dos usuários.
 - * Descrição: Para fortalecer o senso de comunidade e confiança mútua, os usuários precisam de um mecanismo para avaliar uns aos outros após cada carona. Um sistema de reputação visível a todos ajuda na escolha de com quem compartilhar a viagem.
 - **Método de Validação:** A pesquisa online inicial também questionou sobre a importância de um sistema de avaliação. 88% dos respondentes indicaram que se sentiriam mais confortáveis em aceitar ou oferecer uma carona a usuários com boas avaliações.
- **Necessidade 5:** Necessidade 5: Interface intuitiva e processo de combinação simplificado.
 - * Descrição: O aplicativo precisa ser fácil de usar, permitindo que os usuários ofereçam ou encontrem uma carona de forma rápida e eficiente. Um processo complicado desestimula o uso contínuo da plataforma.
 - **Método de Validação:** Foram realizados testes de usabilidade com protótipos de baixa fidelidade baseados em fluxos de aplicativos de mercado. A observação direta de 10 usuários demonstrou que um fluxo com no máximo 4 etapas para solicitar ou oferecer uma carona obteve a maior taxa de sucesso e satisfação.

4. Tecnologias Escolhidas e Justificativa

4.1 Estrutura:

- **Frontend:** React Native

Justificativa:

- Multiplataforma (iOS & Android): Você escreve um único código-base (em JavaScript/TypeScript) que funciona tanto para Android quanto para iOS.

- Performance Nativa: O React Native compila para componentes de UI nativos, oferecendo uma experiência de usuário fluida e responsiva, essencial para mapas e interações em tempo real.

- Ecossistema Maduro: Possui uma vasta biblioteca de componentes prontos, especialmente o react-native-maps (para a integração com Google Maps) e bibliotecas para geolocalização, que são o coração deste aplicativo.

- Time de Desenvolvimento: É mais fácil encontrar desenvolvedores com conhecimento em React/JavaScript do que em duas linguagens nativas distintas.

- **Backend:** Node.js (especificamente com o framework NestJS)

Justificativa:

- Comunicação em Tempo Real: A principal função do app (acompanhar a carona no mapa) exige comunicação em tempo real. O Node.js é imbatível nisso, pois é "orientado a eventos" (event-driven) e se integra perfeitamente com WebSockets (via bibliotecas como Socket.io). Isso permite que o servidor envie a localização do motorista para o passageiro (e vice-versa) instantaneamente, sem que o app precise ficar perguntando "onde ele está?" a cada segundo.

- Performance (I/O): É excelente para lidar com muitas conexões simultâneas (vários alunos usando o app ao mesmo tempo) e operações de entrada/saída (I/O), como consultar o banco de dados e enviar notificações.

- Linguagem Unificada: Se o frontend for em React Native (JavaScript/TypeScript), usar NestJS (TypeScript) no backend permite que a equipe utilize uma única linguagem em todo o projeto, simplificando o desenvolvimento (Full-stack JS/TS).

- Por que NestJS? Enquanto o Node.js puro é muito flexível, o NestJS impõe uma arquitetura organizada e escalável (similar ao Spring Boot), o que é vital para um projeto que lida com a segurança dos alunos.

- **Banco de Dados:** PostgreSQL (com extensão PostGis)

Justificativa:

- Extensão PostGIS: Este é o principal motivo. O PostGIS é uma extensão para o PostgreSQL que o transforma em um banco de dados geoespacial de alta performance. Ele permite que você armazene dados de localização (latitude/longitude) e execute consultas complexas de forma muito eficiente, como:

"Encontrar todos os motoristas disponíveis dentro de um raio de 2km da aluna X."

"Calcular a rota mais próxima entre o motorista Y e o passageiro Z."

- Confiabilidade (ACID): Sendo um banco de dados relacional (SQL), o PostgreSQL oferece transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade). Isso é fundamental para garantir que operações críticas (como confirmar uma carona, verificar um usuário) sejam concluídas com segurança e sem perda de dados.

- Segurança: Possui recursos robustos de segurança e gerenciamento de permissões, essenciais para proteger os dados dos alunos.

- **Hospedagem/Deploy:** AWS (Amazon Web Services)

Justificativa:

- Para um aplicativo que depende de disponibilidade e processamento em tempo real, uma infraestrutura em nuvem robusta é necessária.

- Ecossistema Completo: A AWS oferece todos os serviços necessários para este aplicativo em um só lugar, permitindo que ele escale conforme o uso:

- Amazon RDS: Para hospedar o banco de dados PostgreSQL (com PostGIS) de forma gerenciada, segura e com backups automáticos.

- Amazon EC2 ou ECS/EKS: Para hospedar o backend Node.js. O EKS (Kubernetes) ou ECS (Containers) permite que o backend escale automaticamente em horários de pico (ex: fim da aula).

- Amazon S3: Para armazenar arquivos estáticos, como fotos de perfil dos alunos e documentos de verificação dos motoristas.

- Amazon SNS: Para enviar notificações push (ex: "Sua carona chegou!").

- Confiabilidade: É a plataforma de nuvem líder de mercado, garantindo que o aplicativo esteja disponível quando os alunos mais precisarem (especialmente à noite).

- Integração com Google Maps: Embora o Google Maps seja um serviço do Google, ele funciona perfeitamente em qualquer provedor de nuvem. A infraestrutura da AWS é mais do que capaz de lidar com as chamadas de API necessárias.

- **Outras Ferramentas:** GitHub, Figma e Draw.io

GitHub (Controle de Versão):

- Indispensável. O GitHub é usado para armazenar o código-fonte (frontend e backend). Ele permite que múltiplos desenvolvedores trabalhem juntos, rastreiem cada mudança, revertam erros e gerenciem diferentes versões do aplicativo (desenvolvimento, testes, produção). É a base da colaboração no desenvolvimento.

Draw.io (Fluxogramas):

- O Draw.io auxilia na elaboração de fluxogramas para mapeamento do fluxo do usuário e motorista. Antes de escrever qualquer código, você precisa entender como um usuário (seja

um passageiro ou um motorista) navegará pelo aplicativo. O draw.io será utilizado para criar fluxogramas que detalham cada etapa. Também pode ser utilizado para criar diagramas de arquitetura do sistema.

Figma (Design de UI/UX):

- Antes de programar, é preciso desenhar. O Figma é a ferramenta líder para design de interface (UI) e experiência do usuário (UX). A equipe de design usará o Figma para criar "protótipos" de alta fidelidade de todas as telas do aplicativo (login, mapa, perfil, etc.). Isso permite testar o fluxo de uso e garantir que o app seja intuitivo e seguro antes de escrever o código.

5. Diagrama e Descrição da Solução Proposta

5.1 Descrição Geral da Solução

A solução proposta é um aplicativo móvel de carona solidária, de uso exclusivo para a comunidade acadêmica (alunos e funcionários) do UniSENAI PR. O objetivo central é resolver o problema de insegurança no deslocamento noturno, oferecendo uma plataforma de transporte confiável, acessível e que fortalece o senso de comunidade.

Diferente de aplicativos de mercado, nossa solução foca na confiança e segurança, traduzindo as necessidades validadas dos usuários em funcionalidades centrais:

- **Para atender à Necessidade 1 (Ambiente Exclusivo e Seguro):** A solução implementará um sistema de cadastro robusto onde o Backend (Node.js) fará a validação do vínculo do usuário (via e-mail institucional ou matrícula), possivelmente consultando sistemas internos do UniSENAI, garantindo que apenas membros da comunidade acadêmica tenham acesso.
- **Para atender à Necessidade 2 (Carona Exclusiva para Mulheres):** O aplicativo (React Native) terá um filtro de fácil acesso na interface de busca, permitindo que tanto motoristas quanto passageiras optem por viagens "apenas para mulheres". O Backend aplicará essa regra de negócio no momento de combinar as caronas.
- **Para atender à Necessidade 3 (Monitoramento e Emergência):** Esta é uma funcionalidade central. O Backend (Node.js com WebSockets) gerenciará a comunicação em tempo real, permitindo o compartilhamento da rota ao vivo. O Banco de Dados (PostgreSQL com PostGIS) armazenará e consultará eficientemente os dados de geolocalização. O aplicativo também terá um "botão de emergência" visível para situações de pânico.
- **Para atender à Necessidade 4 (Sistema de Avaliação):** Após cada viagem, o aplicativo solicitará que motorista e passageiro se avaliem mutuamente. Esses dados serão armazenados no PostgreSQL e usados pelo backend para construir um sistema de reputação visível no perfil do usuário, fortalecendo a confiança da comunidade.

- **Para atender à Necessidade 5 (Interface Intuitiva):** O design da interface no Figma e a implementação no React Native seguirão os protótipos validados, garantindo um fluxo simplificado (máximo de 4 etapas) para oferecer ou encontrar uma carona, o que é crucial para a adoção e uso contínuo.

5.2 Diagrama de Arquitetura

A arquitetura do sistema foi desenhada em nuvem (AWS) para garantir alta disponibilidade, segurança e escalabilidade, suportando as demandas de geolocalização e comunicação em tempo real.

O fluxo básico é: O usuário interage com o **Aplicativo Móvel (React Native)**. O aplicativo se comunica com o **Backend (Node.js/NestJS)** através de uma API REST e WebSockets. O Backend, por sua vez, executa a lógica de negócio e consulta o **Banco de Dados (PostgreSQL com PostGIS)** para dados persistentes e consultas geoespaciais. O Backend também se comunica com APIs externas, como o **Google Maps** (para rotas) e o **Sistema Acadêmico do UniSENAI** (para validação de usuários).

- **Frontend (Aplicativo):** Desenvolvido em **React Native**, responsável pela interface do usuário e experiência de navegação.
- **Backend (Servidor):** Desenvolvido em **Node.js (NestJS)**, responsável pela lógica de negócios, gerenciamento de usuários, combinação de caronas e segurança da API.
- **Banco de Dados: PostgreSQL com a extensão PostGIS**, hospedado na AWS (ex: RDS), para armazenamento de dados de usuários, viagens, avaliações e consultas geoespaciais complexas.
- **APIs Externas: Google Maps API (rotas e mapas) e API do UniSENAI PR (validação de cadastro).**

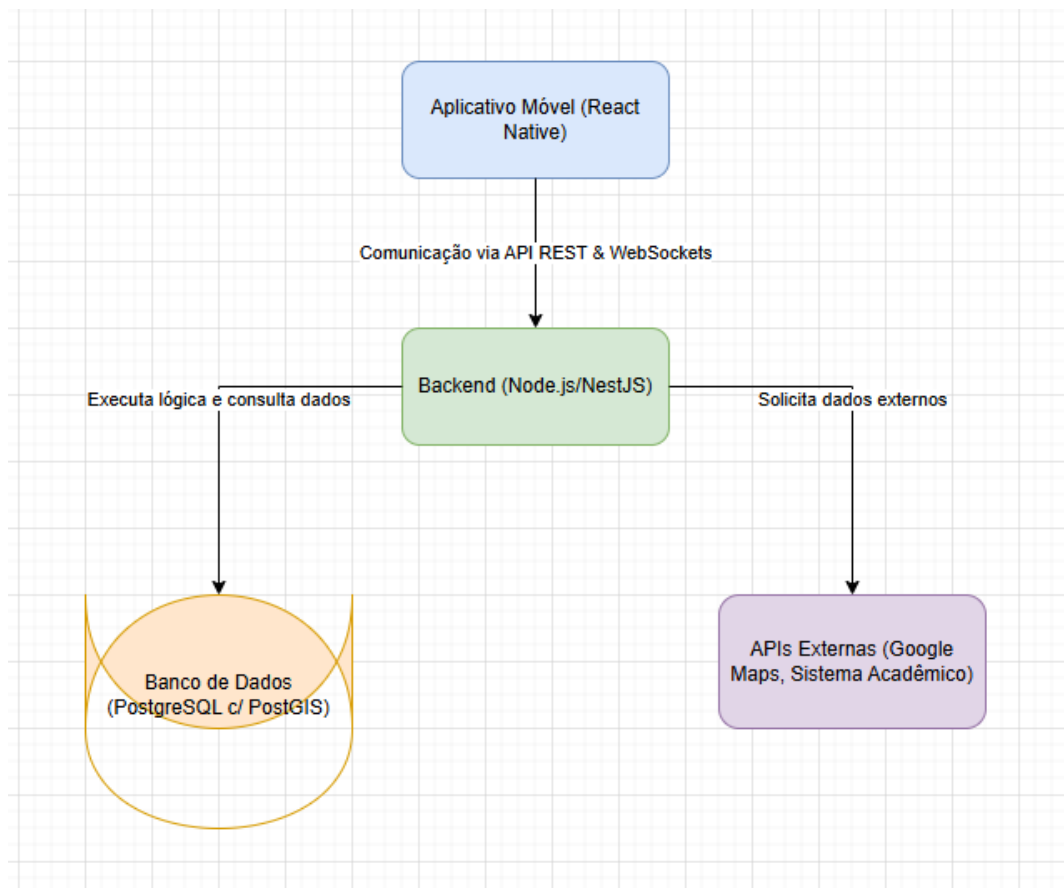


Figura 1 - Diagrama de arquitetura do sistema criado no Draw.io para representar a comunicação entre o Backend, o Aplicativo móvel, o banco de dados e as APIs externas.

5.3 Protótipo da Interface

O protótipo de alta fidelidade foi desenvolvido na plataforma **Figma** com foco em uma experiência de usuário limpa, segura e que atende diretamente às necessidades validadas.

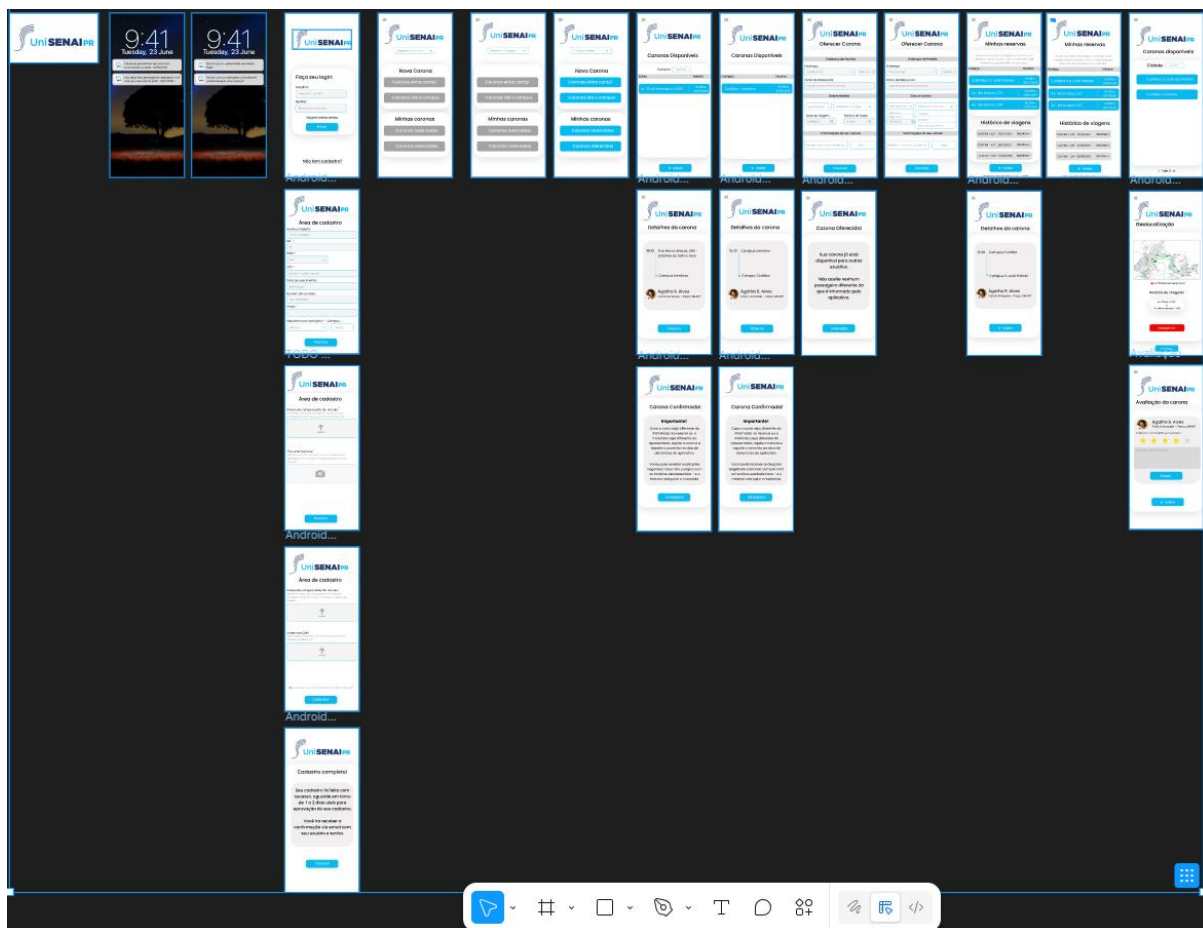


Figura 2 – Visão geral das telas de interface desenvolvidas na plataforma Figma.

A solução desenvolvida pode ser acessada no seguinte link:

<https://www.figma.com/proto/NMveD2c0gdckYWAZosQR33/Prototipo-UniSenaiPR?node-id=0-1&t=kzXzJDTHR5ztsql-1>

Telas Principais da Solução:



Faça seu login:

Usuário:

Senha:

[Esqueci minha senha](#)

Entrar

[Não tem cadastro?](#)



Área de cadastro

Nome completo: *

RG: *

Sexo: *

sexo
▼

CPF: *

Data de nascimento: *

Numero de contato: *

Email: *

Selecione sua instituição: * Campus: *

Instituição
▼
campus
▼

Proximo



Área de cadastro

Anexe seu comprovante de vínculo: *

Usaremos sua identidade para fins de segurança e verificação, ela ira única e exclusivamente para isso.




Tire uma foto sua: *

Usaremos sua foto para por no perfil e verificar sua identidade, além de servir de segurança para outros usuarios.



Proximo


Figura 3- Fluxo de cadastro e validação institucional. Justificativa: Atende à Necessidade 1 (Ambiente Exclusivo e Seguro).



Caronas

Oferecer carona

powered by UniRides



Oferecer Carona

Endereço de Partida

Endereço:

Ponto de Referencia:

Data e Horário

Sexo Motorista
▼


Selecione o Campus
▼

Data da viagem

Horário de saída

Informações do seu veículo

Concluir



Oferecer Carona

Endereço de Partida

Endereço:

Ponto de Referencia:

Data e Horário

Sexo Motorista
▼

Selecione o Campus
▼

Feminino

Masculino

Curitiba

Londrina

São José dos Pinhais

Informações do seu veículo

Concluir

Figura 4 - Tela principal de busca de carona. Justificativa: Atende à Necessidade 5 (Interface Intuitiva) e inclui o filtro para a Necessidade 2 (Carona Exclusiva para Mulheres).

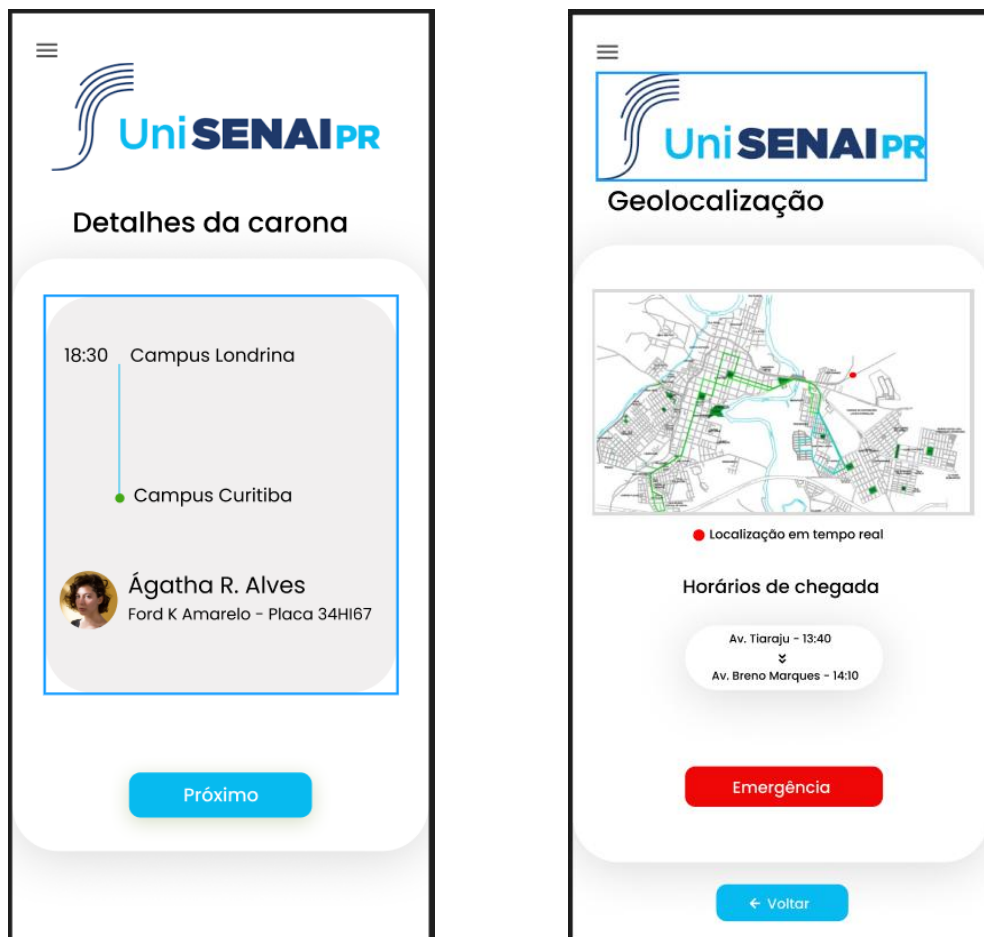


Figura 5 - Tela de monitoramento da viagem em tempo real. Justificativa: Atende à Necessidade 3 (Monitoramento e Botão de Emergência)



Figura 6 - Tela de avaliação mútua. Justificativa: Atende à Necessidade 4 (Sistema de Avaliação e Reputação).

5.4 Repositório Github

Foi criado um repositório na plataforma Github para compartilhamento de arquivos e controle de versões. Nele é disponibilizada a documentação completa do projeto, diagramas, o protótipo desenvolvido na plataforma Figma (interativa) e instruções de instalação. O link para acesso ao repositório é apresentado abaixo:

https://github.com/sidneisilvajunior/ProjetoAplicadoI_ADS_2025

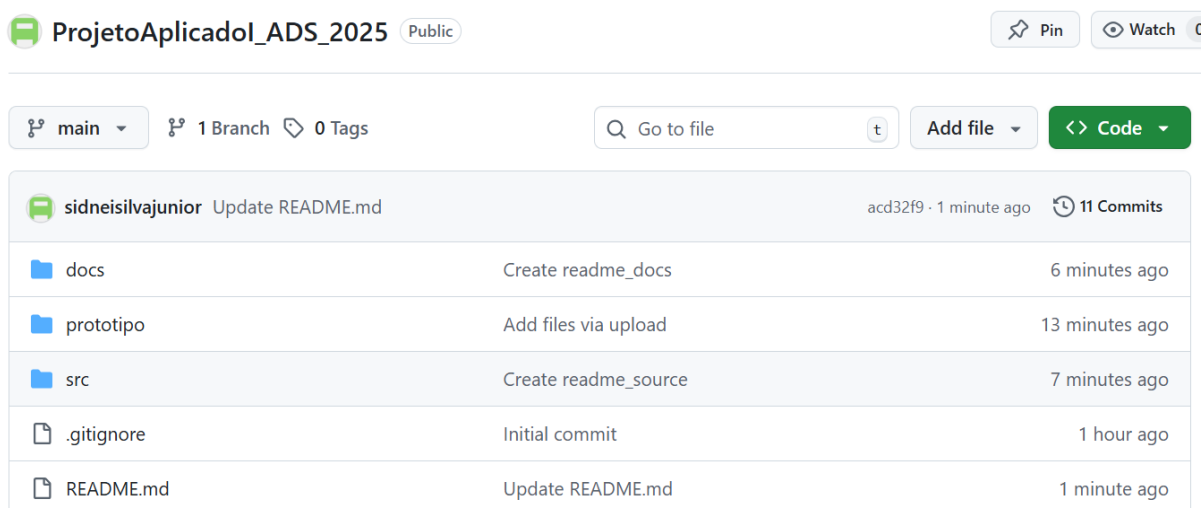


Figura 7 – Estrutura do diretório compartilhado no Github.

6. Conclusão e Próximos Passos

6.1 Conclusão e próximos passos

A validação das necessidades do cliente reforçou a direção do projeto e trouxe clareza sobre as funcionalidades que devem ser priorizadas para atender ao objetivo central: aumentar a segurança dos estudantes do UniSENAI PR.

- **Influência na Solução:** As necessidades validadas serão traduzidas diretamente em funcionalidades no *backlog* de desenvolvimento.
 - A **Necessidade 1** exigirá um sistema de cadastro robusto, possivelmente integrado à base de dados da instituição para validação automática do vínculo do usuário à instituição UNISENAI.
 - As **Necessidades 2, 3 e 4** impactam diretamente o design da interface (UI) e a lógica de negócio no back-end, que deverá comportar filtros de gênero, geolocalização em tempo real e um sistema de pontuação baseado nas avaliações de usuários.
 - A **Necessidade 5** orientará o trabalho da equipe de UI/UX (Marco Aurélio) e Front-end (Sidnei) para garantir uma boa experiência do usuário na utilização do App.
- **Ajustes no Projeto:**
 - **Cronograma:** O prazo para a etapa de "Desenvolvimento", com data limite em 14/10/2025, será mantido, mas as tarefas terão suas prioridades atualizadas para focar nos recursos de segurança. A implementação da validação de usuários com o sistema do UniSENAI do Paraná será detalhada como uma tarefa crítica na etapa de "Integração" (28/10/2025).
 - **Escopo Técnico:** A complexidade da integração com os sistemas do UniSENAI PR para verificação de usuários será investigada pelo Fabiano (Back-end), podendo exigir um planejamento técnico mais detalhado para garantir a segurança dos dados.

Com as necessidades devidamente validadas, o projeto avança para a fase de detalhamento técnico e desenvolvimento com maior segurança de que o produto final atenderá às expectativas reais da comunidade acadêmica.

6.2 Balanço Final do Projeto (Resultados e Aprendizados)

Esta seção complementa a análise anterior, focando nos resultados globais do projeto conforme as exigências da entrega final (4/4).

Resultados Alcançados

O projeto atingiu com sucesso seu objetivo principal para esta Unidade Curricular: transformar um problema social complexo – a insegurança dos estudantes do UniSENAI PR em uma solução técnica completa, validada e pronta para o desenvolvimento. Os principais resultados entregues foram:

- **Solução Validada:** A validação rigorosa das necessidades (Seção 3) garantiu que o produto está alinhado com as expectativas reais dos usuários, especialmente em relação à segurança e à funcionalidade de carona exclusiva para mulheres.
- **Arquitetura Definida:** Foi projetada uma arquitetura de sistema (Seção 5.2), baseada nas tecnologias da Seção 4 (React Native, Node.js, PostGIS), capaz de suportar as funcionalidades críticas de geolocalização e segurança em tempo real.

- **Protótipo de Alta Fidelidade:** Foi entregue um protótipo interativo no Figma (Seção 5.3) que demonstra um fluxo de usuário simples e intuitivo, traduzindo visualmente como as necessidades de segurança serão atendidas.
- **Repositório Organizado:** Todo o código-fonte, diagramas e a documentação técnica foram disponibilizados no GitHub, garantindo a continuidade do projeto.

Dificuldades e Aprendizados

- **Principais Dificuldades:**
 - **Complexidade de Integração:** O maior desafio técnico identificado foi a integração com os sistemas acadêmicos do UniSENAI PR. Garantir a validação segura e automática dos usuários exige um planejamento de segurança de dados (LGPD) e esforço técnico que não puderam ser subestimados.
 - **Gestão de Escopo:** O feedback dos usuários gerou muitas ideias valiosas. Manter o foco nas funcionalidades essenciais (MVP) de segurança, sem se desviar para funcionalidades secundárias, foi um desafio constante de gerenciamento de projeto.
- **Principais Aprendizados:**
 - **Validação Dita a Tecnologia:** O aprendizado mais significativo foi como a Seção 3 (Necessidades) ditou diretamente a Seção 4 (Tecnologias). A "Necessidade 3: Monitoramento", por exemplo, tornou a escolha do PostGIS (com Node.js) uma decisão técnica clara, em vez de uma simples preferência.
 - **A Importância do Protótipo:** Testar o protótipo do Figma (Seção 5.3) revelou falhas de usabilidade que foram corrigidas antes que uma única linha de código fosse escrita, economizando dezenas de horas de retrabalho de desenvolvimento.

Possíveis Melhorias Futuras (Próximas UCs)

Este projeto estabelece uma base sólida para as próximas UCs de Projeto Aplicado. Os próximos passos recomendados são:

- 1 **Desenvolvimento do MVP (Minimum Viable Product):** Iniciar a codificação (Frontend e Backend) das funcionalidades centrais já prototipadas: cadastro de usuário (com validação), filtro de gênero e o ciclo de "oferecer/procurar/avaliar" carona.
- 2 **Prova de Conceito (PoC) da Integração:** Dedicar uma sprint ou UC específica para resolver o desafio da integração com o sistema acadêmico do UniSENAI PR, criando uma prova de conceito segura.
- 3 **Testes Beta:** Executar o plano de testes com um grupo de alunos reais, conforme planejado, para coletar feedback sobre o desempenho e a segurança do aplicativo em um cenário de uso real.
- 4 **Evolução:** Após o MVP, implementar funcionalidades secundárias como otimização de rotas para múltiplos passageiros e um chat interno seguro.