

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO

Andrei Matiazi Schopf

**VENN - UMA PLATAFORMA DE TRANSPORTE COLETIVO POR  
DEMANDA: IMPLEMENTAÇÃO DO APLICATIVO PARA O USUÁRIO**

Santa Maria, RS  
2020

**Andrei Matiazi Schopf**

**VENN - UMA PLATAFORMA DE TRANSPORTE COLETIVO POR DEMANDA:  
IMPLEMENTAÇÃO DO APLICATIVO PARA O USUÁRIO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação, Área de Concentração em Área de concentração do CNPq, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro em Engenharia de Controle e Automação.**

ORIENTADORA: Prof.<sup>a</sup> Marcia Pasin

Santa Maria, RS  
2020

**Andrei Matiazi Schopf**

**VENN - UMA PLATAFORMA DE TRANSPORTE COLETIVO POR DEMANDA:  
IMPLEMENTAÇÃO DO APLICATIVO PARA O USUÁRIO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação, Área de Concentração em Área de concentração do CNPq, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro em Engenharia de Controle e Automação.**

**Aprovado em 4 de outubro de 2020:**

---

**Marcia Pasin, Dra. (UFSM)**  
(Presidenta/Orientadora)

---

**Claiton Moro Franchi, Dr. (UFSM)**

---

**Hélio Leães Hey, Dr. (UFSM)**

Santa Maria, RS  
2020

*Se falta de tempo realmente fosse uma justificativa para não realizar seus projetos, somente os desocupados teriam sucesso.*

*(Flávio Augusto da Silva)*

## RESUMO

### **VENN - UMA PLATAFORMA DE TRANSPORTE COLETIVO POR DEMANDA: IMPLEMENTAÇÃO DO APLICATIVO PARA O USUÁRIO**

AUTOR: Andrei Matiazi Schopf

ORIENTADORA: Marcia Pasin

De acordo com a pesquisa do IBGE, a quantidade de motoristas de aplicativo, taxistas e motoristas e trocadores de ônibus, aumentou 29,2% em 2018 e chegou a 3,6 milhões. Isso indica que a quantidade de pessoas que operam nessas posições conta com 810 mil pessoas a mais em relação a 2017. O aumento destes números evidencia também o aumento da quantidade de veículos em circulação. O excesso de veículos circulando nas ruas causa atrasos, engarrafamentos e aumenta a poluição. Medidas são necessárias para melhorar a mobilidade urbana e a qualidade de vida das pessoas. Com o objetivo de colaborar na melhoria de mobilidade urbana, este projeto propõe o desenvolvimento um sistema de transporte coletivo urbano compartilhado, denominado Venn. O serviço do Venn é oferecido por vans que percorrem rotas dinâmicas. Os usuários solicitam a jornada pelo aplicativo móvel, informando horário, origem e destino, e o sistema calcula as rotas com base nos veículos disponíveis. Em contraste com o serviço oferecido por veículos particulares, o uso de vans no Venn aumenta a capacidade da quantidade de pessoas que podem ser transportadas por veículos. No escopo deste trabalho, o enfoque é dado no desenvolvimento do aplicativo móvel para os usuários Venn, que é validado por prototipação. O sistema, quando estiver totalmente implantado, poderá colaborar na redução da necessidade de manter inúmeros veículos rodando sem passageiros, trabalhando com horários e paradas ineficientes. Como resultados esperados, busca-se colaborar na redução da quantidade de veículos em circulação nas vias e, conseqüentemente, impactar na redução de engarrafamentos.

**Palavras-chave:** Transporte Coletivo. Roteamento de Veículos. Desenvolvimento Mobile. Mobilidade Urbana.

## **ABSTRACT**

### **VENN: A COLLECTIVE TRANSPORT PLATFORM ON DEMAND**

**AUTHOR:** Andrei Matiazi Schopf

**ADVISOR:** Marcia Pasin

According to the IBGE-Brazil survey, the number of application drivers, taxi drivers, bus drivers and bus changers increased by 29.2 % in 2018 and reached 3.6 million. This indicates that the number of people operating in these positions has 810 thousand compared to 2017. The increasing in these numbers also shows the increasing in the number of vehicles in circulation. The excess of vehicles circulating on the streets causes delays, traffic jams and increases pollution. Measures are needed to improve urban mobility and people's quality of life. With the focus on improving mobility, this project proposes the development of a shared urban public transport system, called Venn. Venn's service is offered by vans that take dynamic routes. Users request their journeys through a mobile application, informing stating time, origin and destination, and the Venn system calculates routes based on available vehicles. In contrast to the service offered by private vehicles, the use of vans on Venn increases the capacity of the number of people that can be transported by vehicles. In the scope of this work, the focus is on the development of the mobile application for Venn users, which is validated by prototyping. Thus, the Venn system, when fully implemented, can help reduce the need to keep countless vehicles running without passengers, working with inefficient schedules and stops. As expected results, we seek to collaborate in reducing the number of vehicles on the road as well as to contribute to reduce traffic jams.

**Keywords:** Public Transport. Vehicle Routing. Mobile Development. Urban Mobility.

## LISTA DE FIGURAS

Figura 2.1 – Tela do aplicativo Buser. ....	13
Figura 2.2 – Tela do aplicativo Ubus. ....	15
Figura 2.3 – Tela do aplicativo CityBus 2.0. ....	16
Figura 3.1 – Exemplo com satélites para o funcionamento da trilateração. ....	20
Figura 3.2 – Distância entre dois pontos, sendo o receptor (R) e um satélite $n$ . ..	20
Figura 4.1 – Esquema simplificado da arquitetura do sistema Venn. ....	23
Figura 4.2 – Arquitetura com tecnologias de software do sistema Venn. ....	24
Figura 4.3 – Diagrama com esquema de preços Venn. ....	27
Figura 5.1 – Diagrama com a sequência de passos para a solicitação de embarque na Venn e atualização de rotas, e desembarque para finalização de ciclo de sequência de passos. ....	30
Figura 5.2 – Raio $\delta_o$ demonstrando quais passageiros seriam atendidos com a viagem indicada. ....	31
Figura 5.3 – Raio $\delta_d$ demonstrando quais passageiros que possivelmente seriam atendidos com a viagem indicada. ....	32
Figura 5.4 – Estrutura de pastas e arquivos. ....	34
Figura 5.5 – Adição da chave de API. ....	35
Figura 5.6 – Modelo de telas do aplicativo móvel para o usuário Venn. ....	36
Figura 5.7 – Modelo de telas com viagens e trajetos do aplicativo móvel para o usuário Venn. ....	37
Figura 5.8 – Modelo de tela do aplicativo do usuário Venn para pagamento do serviço. ....	39
Figura 5.9 – Tela de gerenciamento do Pagar.me. ....	40
Figura 5.10 – Exemplos de telas do aplicativo do usuário Venn. ....	41
Figura 5.11 – Telas do aplicativo do usuário Venn. ....	42
Figura 5.12 – Telas do aplicativo do usuário Venn. ....	43
Figura 5.13 – Telas do aplicativo do usuário Venn. ....	44

## LISTA DE TABELAS

Tabela 2.1 – Comparativo entre as soluções.....	17
---	----



## LISTA DE ABREVIATURAS E SIGLAS

<i>ANTP</i>	Associação Nacional de Transportes Públicos
<i>API</i>	Application Programming Interface
<i>BaaS</i>	Back-end as a Service
<i>BRT</i>	Transporte Rápido por Ônibus ou Bus Rapid Transit
<i>GPS</i>	Global Positioning System
<i>IBGE</i>	Instituto Brasileiro de Geografia e Estatística
<i>NTU</i>	Associação Nacional das Empresas de Transportes Urbanos
<i>PNAD</i>	Pesquisa Nacional por Amostra de Domicílios

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	MOTIVAÇÃO	9
1.2	OBJETIVOS	10
1.3	ESTRUTURA DESTE TEXTO	11
<b>2</b>	<b>REVISÃO DO ESTADO DA ARTE</b>	<b>12</b>
2.1	BUSER	12
2.2	UBUS	14
2.3	CITYBUS 2.0	14
2.4	COMPARAÇÃO ENTRE OS SISTEMAS DE TRANSPORTE DE PASSAGEIROS ALTERNATIVO	15
<b>3</b>	<b>TECNOLOGIAS E CONCEITOS</b>	<b>18</b>
3.1	REACT NATIVE	18
3.2	SISTEMA DE POSICIONAMENTO GLOBAL	19
3.3	TRILATERAÇÃO	19
3.4	GOOGLE MAPS API	20
3.5	FIREBASE	21
3.6	ADOBE XD E DESIGN UX	22
<b>4</b>	<b>O SISTEMA VENN</b>	<b>23</b>
4.1	MODELO DE NEGÓCIO	25
<b>4.1.1</b>	<b>Cenários</b>	<b>25</b>
4.1.1.1	<i>Cenário otimista</i>	25
4.1.1.2	<i>Cenário pessimista</i>	26
<b>4.1.2</b>	<b>Lucros</b>	<b>26</b>
4.2	ALGORITMOS PARA CALCULO DE ROTA DISPONÍVEIS PARA SEREM APLICADOS	28
<b>5</b>	<b>APLICATIVO MÓVEL PARA OS USUÁRIOS VENN</b>	<b>29</b>
5.1	FLUXO DE DADOS E VISÃO GERAL	29
5.2	INSTALAÇÃO DE SOFTWARE E CONFIGURAÇÃO DOS DISPOSITIVOS	32
5.3	MOCKUP DO APLICATIVO	35
5.4	VISUALIZAÇÃO DAS VIAGENS REALIZADAS	35
5.5	ATUALIZAÇÃO DOS VEÍCULOS QUE ESTÃO RODANDO NAS PROXIMIDADES	36
5.6	VERIFICAÇÃO DE LOGIN PARA SEGURANÇA	37
5.7	PAGAMENTO DIRETO ATRAVÉS DE CARTÃO DE CRÉDITO	38
5.8	TELAS DA APLICAÇÃO NATIVA EM DISPOSITIVO ANDROID	39
<b>6</b>	<b>CONCLUSÃO</b>	<b>45</b>
6.1	APONTAMENTOS FINAIS	45
6.2	TRABALHOS FUTUROS	46
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>47</b>

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

A dinâmica das cidades tem sofrido constantes evoluções, o que impacta na qualidade da mobilidade, principalmente nos grandes centros. Essas mudanças se fazem necessárias devido ao aumento da densidade populacional nos grandes centros.

Segundo as Estimativas de População (Agência IBGE Notícias, 2019), aproximadamente 31,7% dos brasileiros residem em 48 municípios com mais de 500 mil habitantes. Os municípios que mais crescem possuem entre 100 mil e 1 milhão de habitantes. Particularmente, municípios muito grandes, com mais de 1 milhão de habitantes, apresentam crescimento ligeiramente abaixo da média brasileira. Em municípios menores, a taxa média de crescimento é baixa.

Essa população, que não existia ou estava distribuída por cidades do interior há alguns anos, agora, necessita de meios de locomoção para se deslocar até seu trabalho, ter momentos de lazer e inúmeras outras atividades rotineiras. Assim como a população cresceu, os meios que garantem a mobilidade urbana devem se expandir, resultando em milhões de reais necessários de investimento (NTU, 2019).

A maior parte dos recursos destinados a melhora da mobilidade urbana são direcionados à recuperação e ao alargamento de vias. Essas medidas não resolvem o problema de uma forma geral, apenas postergam o problema desafogando o trânsito de veículos nas vias temporariamente. Todos os anos o número de veículos circulando aumenta, de forma que novamente as rodovias não deem vazão ao tráfego crescente (NTU, 2019).

Em contrapartida, os investimentos em mobilidade compartilhada, a criação de corredores para ônibus, os subsídios de passagens, a aquisição de novos veículos para a frota de serviço urbano e a construção de abrigos são historicamente consideravelmente menores do que os direcionados para fins que beneficiem unicamente o transporte individual realizado por meio de veículos de passeio e motocicletas.

As grandes soluções de mobilidade urbana não tratam os problemas atuais com a velocidade e eficiência necessárias, demorando muitos anos para serem finalizadas e para suprirem a demanda por mobilidade. Para suprir essa demanda, uma possibilidade é usar as soluções de transporte compartilhado por veículos de passeio. Ao contrário do transporte público, o transporte compartilhado não reduz drasticamente o número de veículos em circulação, mas colabora na redução número de veículos em circulação.

De acordo com a PNAD Contínua IBGE 2019 (Agência de Notícias IBGE, 2019),

a quantidade de motoristas de aplicativo, taxistas e motoristas e trocadores de ônibus, aumentou 29,2% em 2018 e chegou a 3,6 milhões, com 810 mil pessoas a mais em relação a 2017. O aumento destes números evidencia também o aumento da quantidade de veículos em circulação. Logo, o número de veículos em circulação continua aumentando a medida que os anos avançam.

O excesso de veículos circulando nas ruas causa atrasos, engarrafamentos, aumenta a poluição, e impactam negativamente a vida das pessoas. Cada vez mais medidas são necessárias para melhorar a mobilidade urbana e a qualidade de vida das pessoas. Esta solução precisa combinar custo adequado, e agilidade com redução da quantidade de veículos circulando nas ruas, de forma atraente para os usuários.

## 1.2 OBJETIVOS

Este projeto tem como finalidade o desenvolvimento de um aplicativo para clientes do sistema de transporte coletivo urbano, denominado Venn. O sistema usa vans ou micro-ônibus que circulam de forma definida pelo sistema, conforme a demanda de usuários e seus destinos definidos no momento da solicitação. O sistema não depende de locais fixos de embarque e de desembarque, frequentemente indesejáveis quando o passageiro deve caminhar para alcançá-los.

O sistema Venn, quando estiver totalmente implantado, não necessita manter inúmeros veículos rodando sem passageiros, trabalhando com horários e paradas ineficientes. O Venn visa reduzir os custos operacionais e possivelmente colaborar para minimizar congestionamentos. Este sistema deve retirar de circulação vários veículos, uma vez que a tendência é que exista um maior conforto e eficiência em relação às soluções tradicionais de transporte coletivo urbano. O transporte tradicional tem mais dificuldade em face de uma demanda variável nas regiões que atende, superlotando quando poucos veículos são disponibilizados.

O foco deste trabalho está no desenvolvimento do aplicativo móvel usado pelos usuários do sistema Venn. Para tanto, este texto detalha a arquitetura e implementação deste aplicativo. Para sua validação, foi construído um protótipo com tecnologias atuais. Finalmente, o modelo de negócios do sistema Venn é brevemente discutido dado que a proposta deve ser viável de implementação em um mundo real.

O aplicativo móvel desenvolvido possui usabilidade e tem fácil integração com diferentes sistemas (como o banco de dados e a plataforma para cadastramento de motoristas). Adicionalmente, como o sistema Venn não necessita de interferência direta do poder público, o sistema pode atingir grandes escalas. Outra vantagem é que, como o Venn opera sobre uma infraestrutura já disponível, as soluções de mobilidade

ocorrem de uma forma mais eficiente e rápida, quando comparada às grandes obras de metro ou Transporte Rápido por Ônibus (Bus Rapid Transit ou BRT). Como é sabido, as obras de metrô e de BRT demoram considerável tempo para serem concluídas e efetivamente serem utilizadas.

### 1.3 ESTRUTURA DESTE TEXTO

Este texto está organizado como segue. O Capítulo 2 apresenta os conceitos teóricos utilizados no desenvolvimento do projeto, além de uma revisão das soluções existentes no mercado, apontando os pontos fortes e fracos de cada uma delas. O Capítulo 3 apresenta as tecnologias e os conceitos necessários para a implementação e entendimento do aplicativo do usuário Venn. O Capítulo 4 apresenta o sistema Venn. Na sequência, o Capítulo 5 apresenta a arquitetura do sistema Venn do usuário (passageiro), com detalhamento de seus componentes, e detalhes de implementação do protótipo para o aplicativo do usuário Venn, como os métodos que foram escolhidos para a execução adequada do projeto. Finalmente, o Capítulo 6 apresenta as conclusões deste trabalho.

## 2 REVISÃO DO ESTADO DA ARTE

O Brasil reconheceu, em 2015, o transporte público como um direito social dos cidadãos brasileiros. Desde então era esperado que o setor passasse por mudanças que trouxessem mais qualidade ao serviço. Segundo a Associação Nacional de Transportes Públicos (ANTP, 2019), a manutenção de um padrão de satisfação do usuário que consiga realizar seus deslocamentos de forma confortável é fundamental para o funcionamento da cidade contemporânea.

No entanto, segundo dados divulgados em 2019 pela Associação Nacional das Empresas de Transportes Urbanos (NTU, 2019), 4.6 milhões de usuários por dia deixaram de utilizar esse tipo de transporte. Isso representa uma perda de 3.6 bilhões de reais por ano.

No Brasil, o transporte coletivo urbano é gerenciado pelas prefeituras dos municípios. Elas são responsáveis pelas licitações que selecionam as empresas que irão operar na cidade. Também possuem o dever de regulamentar e fiscalizar o serviço fornecido pelas empresas.

O sistema adotado no país consiste em ônibus que circulam em rotas fixas, de acordo com o plano de mobilidade da cidade. Nelas, há paradas fixas, com abrigos fornecidos pelas prefeituras. Embora haja uma tabela de horários mínima fornecida pelo município, as empresas são quem decidem os horários de acordo com seus próprios critérios. A tarifa é estabelecida por meio de contrato entre as prefeituras e as empresas. As alterações precisam ser justificadas e aprovadas pelo município.

Além do transporte coletivo urbano gerenciado pelas prefeituras dos municípios, existem outras alternativas que serão discutidas no texto que segue. Essas alternativas estão mais alinhadas ao escopo deste trabalho.

### 2.1 BUSER

O Buser (Buser, 2020) é uma plataforma de fretamento colaborativo criada em junho de 2017, com o objetivo de oferecer viagens mais baratas e com mais segurança e conforto para os usuários, que muitas vezes são tratados de forma desumana pelas empresas de viagem. O sistema funciona através da demanda por novas viagens. Assim que um número determinado de pessoas solicita a criação de uma nova rota, ela passa a ser disponibilizada pela empresa. Além disso, o Buser possui uma tarifa que varia conforme a lotação do ônibus, quanto mais próximo do número máximo de passageiros. Esse sistema é destinado a rotas intermunicipais, com destinos de início

e fim fixos.

Na Figura 2.1 está exemplificado a tela inicial do aplicativo Buser, nela pode ser visto o funcionamento do mesmo. Ele se baseia em rotas fixas, então só oferece a opção de encontrar as viagens que estão disponíveis.



Figura 2.1 – Tela do aplicativo Buser.

Fonte: Reprodução Buser

O sistema Buser foi desenvolvido por uma empresa brasileira de mesmo nome. Atualmente o serviço está disponível nos estados de Minas Gerais, São Paulo, Rio de Janeiro e Santa Catarina.

## 2.2 UBUS

Ubus (Ubus, 2020) oferece uma plataforma de mobilidade coletiva, com a finalidade de estabelecer a ponte entre as empresas operadoras do transporte coletivo e seus clientes. O Ubus utiliza um aplicativo mobile como a fonte de conexão entre o usuário e o sistema, para a solicitação de uma viagem ou verificação das viagens disponíveis. A plataforma possui três modalidades para realizar o atendimento ao cliente:

- Módulo Linha, com uma rota fixa determinada pelo operador,
- Módulo Fretamento, pré-determinado pelo cliente: itinerário, dia e horário.
- Módulo Porta a Porta, sob demanda solicitações de itinerário flexível para o operador e cliente.

A Figura 2.2 apresenta a tela de viagens do sistema, nela é possível consultar as viagens disponíveis além de agendar uma viagem. Todas as rotas são pré-definidas.

A Ubus opera atualmente na cidade de São Paulo, possui tecnologia desenvolvida pela própria empresa. A plataforma opera há mais de 6 meses na cidade.

## 2.3 CITYBUS 2.0

Entre todos os sistemas pesquisados, o CityBus 2.0 (CityBus 2.0, 2020) e o sistema mais semelhante ao Venn. O CityBus 2.0 possibilita fazer a solicitação da viagem via aplicativo de smartphone. As rotas do CityBus são construídas dinamicamente, assim como os pontos de embarque e desembarque são virtuais e próximos aos usuários. O CityBus dispõe de vans da própria empresa para realizar o transporte dos passageiros, além de utilizar um sistema importado de uma empresa dos EUA. O CityBus 2.0 ainda está em fase de testes, operando em uma região delimitada da cidade de Goiânia em Goiás.

A empresa estima que o tempo de espera, na região atendida, nunca passa de alguns minutos. Isso se deve a grande disponibilidade de veículos ofertada aos usuários, para garantir a oferta necessária que suprirá a demanda.

Conforme a Figura 2.3, o aplicativo trabalha com uma camada de mapa, possibilitando mostrar ao passageiro usuário o deslocamento em busca de outros passageiros em diferentes lugares da cidade. No aplicativo, é possível solicitar o destino da viagem e realizar o pagamento assim que a viagem for finalizada.





Figura 2.2 – Tela do aplicativo Ubus.

Fonte: Reprodução Ubus

O CityBus 2.0 foi desenvolvido por uma empresa residente nos Estados Unidos. Ele é operado pela HP transportes coletivos e atua em área central da cidade de Goiânia no estado de Goiás.

#### 2.4 COMPARAÇÃO ENTRE OS SISTEMAS DE TRANSPORTE DE PASSAGEIROS ALTERNATIVO

Embora os sistemas anteriormente relatados sejam semelhantes, eles possuem diferenças relevantes na sua estrutura quando comparado ao sistema desenvolvido neste trabalho. A Tabela 2.1 apresenta uma breve comparação dos sistemas relatados em relação ao Venn. Critérios observados incluem área de atuação, tipo de

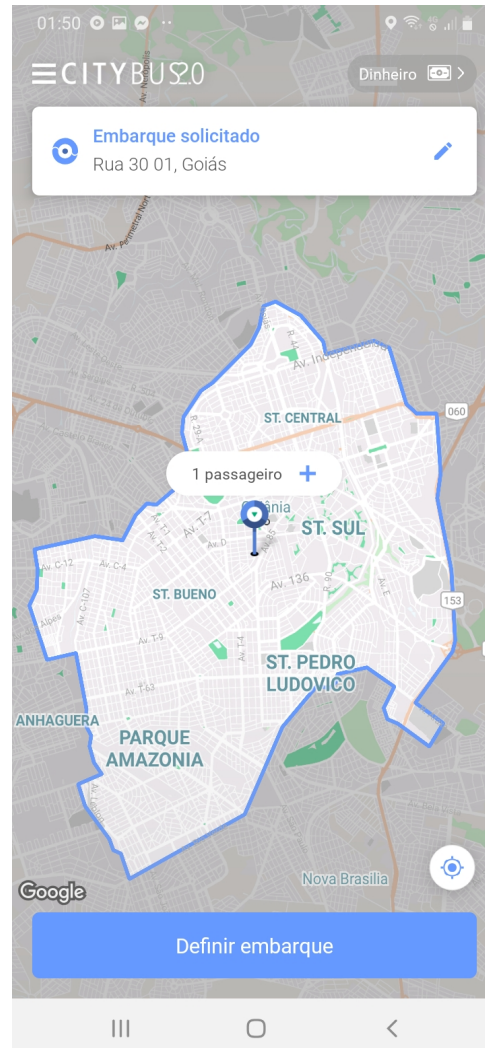


Figura 2.3 – Tela do aplicativo CityBus 2.0.

Fonte: Reprodução CityBus 2.0

veículos, tipo de paradas, tarifas e custo de implementação.

Em relação à área de atuação, tem-se a Buser realizando viagens intermunicipais e interestaduais. Enquanto Ubus, Citybus 2.0 e Venn realizando viagens intramunicipais. Nesse quesito, a Buser não seria um concorrente direto. Ubus e CityBus 2.0 atuam com veículos próprios, enquanto a Buser trabalha com um misto de fretamento e veículos próprios. Nesse aspecto, a Venn tem vantagens ao trabalhar com veículos pertencentes a parceiros, reduzindo em muito o custo de implementação da solução. As paradas da Buser e do CityBus 2.0 são pré-definidas, na Ubus as paradas são pré-definidas e dinâmicas. Já na Venn, as paradas são totalmente dinâmicas.

Finalmente, todas soluções pesquisadas possuem custo mais reduzido quando comparado ao uso de carros particulares ou de táxi convencionais. Além disso, Ubus e CityBus 2.0 possuem custo de implementação alto, enquanto na Buser e na Venn esse custo é baixo.

<b>Característica</b>	<b>Buser</b>	<b>Ubus</b>	<b>CityBus 2.0</b>	<b>Venn</b>
Área de atuação	Intermunicipal e interestadual	Intramunicipal	Intramunicipal	Intramunicipal
Veículos	Fretamento e veículos próprios	Veículos próprios	Veículos próprios	Parceiros
Paradas	Pré-definidas	Pré-definidas e dinâmicas	Pré-definidas	Dinâmicas
Tarifas	Mais baratos que ônibus tradicionais	Mais caros que transporte convencional	Mais caros que transporte convencional e mais barato que ride-sharing	Mais caros que transporte convencional e mais barato que ride-sharing
Custo de implementação	Baixo	Alto	Alto	Baixo

Tabela 2.1 – Comparativo entre as soluções.

### 3 TECNOLOGIAS E CONCEITOS

Neste capítulo, são apresentadas as tecnologias e os conceitos utilizados para o desenvolvimento do aplicativo do usuário da plataforma Venn. Inicialmente, é apresentado o React Native, criado pelo Facebook no ano de 2015. Obteve grande adesão da comunidade de desenvolvedores front-end devido a possibilidade de desenvolver aplicativos para Android e IOs de forma híbrida. Além disso, são tratados os conceitos e da utilização do GPS - Global Positioning System (Sistema de Posicionamento Global) nesse trabalho, e sobre o motivo dele ser escolhido em relação as outras tecnologias equivalentes. O Venn também precisa de tecnologias de visualização de mapas. Neste trabalho, foi utilizado o Google Maps API integrado ao React Native. Finalmente, são brevemente descritos o banco de dados Firebase, e as ferramentas Adobe XD e design UX, todos usados neste trabalho.

#### 3.1 REACT NATIVE

Neste trabalho, optou-se pela utilização do React Native (React Native, 2020) como forma de desenvolvimento mobile, principalmente pelas diversas funcionalidades e produtividade que ele fornece. Basicamente, React Native é um framework para JavaScript, desenvolvido pelo Facebook, no ano de 2015. Além de ser fácil de utilizar por se aproximar bastante a linguagens de desenvolvimento web, ele possibilita o desenvolvimento de aplicações mobile de forma híbrida, aproveitando quase 100% do código para desenvolver em ambos as plataformas Android e IOs.

Diferentemente de outros frameworks e bibliotecas que empacotam a aplicação e a executam em um browser para adquirir o aspecto de aplicativo mobile, o React Native converte todo o código para a linguagem nativa do sistema operacional. Além disso, também é possível realizar alterações rápidas no código enquanto o programa roda em um emulador ou celular, esse recurso é chamado de Hot Reloading. Ele permite ao aplicativo ser atualizado em menos de um minuto, sem necessitar que seja recompilado.

Apesar dos processos serem realizados de forma nativa, eles ainda necessitam que os componentes JavaScript sejam convertidos para a linguagem nativa do sistema operacional. Ao ser realizada ou solicitada uma ação aos sensores do dispositivo que está executando. Essa operação intermediária é chamada de bridge, por exemplo um evento de *Touch Event* leva cerca de alguns milissegundos desde o toque até o processamento final ocorrer.

O React Native pode ser integrado com diferentes APIs e recursos, além de possuir uma comunidade extremamente ativa facilitando assim o processo de correção de *bugs* e dificuldades que possam aparecer.

### 3.2 SISTEMA DE POSICIONAMENTO GLOBAL

A definição de posição dos veículos, bem como dos passageiros, requer a utilização de tecnologia que informe a latitude e longitude com certo grau de precisão. A tecnologia GPS é a mais difundida nos smartphones, sendo assim adequada para ser utilizada no protótipo.

O GPS (Global Positioning System ou Sistema de Posicionamento Global) (Escritório Nacional de Coordenação para Posicionamento, Navegação e Momento Baseados no Espaço - EUA, 2019) foi desenvolvido pelo Departamento de Defesa dos Estados Unidos. Na década de 80, foram realizados os primeiros testes com os receptores. Hoje, o sistema é conhecido e utilizado por toda a população mundial. Ele é suportado por um sistema de localização baseado em trilateração por satélite. São utilizados três satélites para retornar a posição e quatro para a altitude.

Existem inúmeras variações nos modelos de receptores GPS, no entanto neste trabalho foram utilizados apenas com os existentes nos celulares. Dessa forma, há uma ampla possibilidade, sabendo que há um GPS no bolso de cada pessoa que possui um smartphone.

### 3.3 TRILATERAÇÃO

A trilateração é um conceito matemático utilizado para definir a localização desconhecida de um objeto, dada a localização conhecida de três âncoras (LIU Z. YANG; JIAN, 2010).

Quando este procedimento usa mais de três âncoras, chama-se multilateração, tipicamente aplicada no GPS. Em um contexto com três âncoras, fazendo uso dos satélites GPS como pontos centrais de esferas é aferida a posição de um receptor, baseando-se unicamente na intersecção de três dessas esferas.

Um receptor GPS calcula a distância em função do tempo que um sinal emitido para um dos satélites leva para retornar a ele. Assim deve-se existir precisão de horários, tanto do satélite, como no dispositivo. Os satélites são equipados com relógios atômicos e são responsáveis pela atualização da hora dos dispositivos receptores, assegurando a qualidade da leitura.

A Figura 3.1 apresenta à esquerda um esquema com três satélites e um receptor R. Sabendo-se a posição dos satélites, e distância entre cada satélite e o receptor R, é possível calcular por trilateração a posição do receptor R.

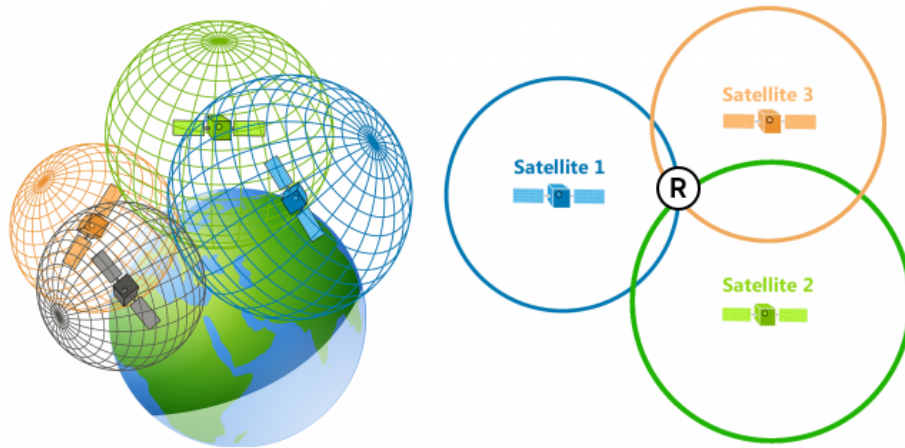


Figura 3.1 – Exemplo com satélites para o funcionamento da trilateração.

Fonte: [www.gisgeography.com](http://www.gisgeography.com)

Dada a distância entre cada Satélite e o receptor R, através da aplicação da fórmula da distância entre dois pontos, indicada na Figura 3.2, é possível descrever uma equação para cada Satélite.

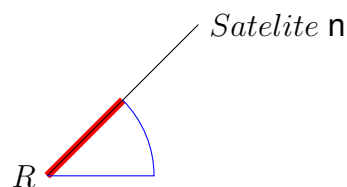


Figura 3.2 – Distância entre dois pontos, sendo o receptor (R) e um satélite  $n$ .

Aplicando-se uma combinação destas equações através do Teorema de Pitágoras, a posição do receptor R é aferida. Como o próprio nome já explica, a trilateração usa três satélites para calcular a posição de um receptor. Se forem usados quatro satélites, a altitude também pode ser calculada.

### 3.4 GOOGLE MAPS API

O Google Maps API (Google Maps API, 2020) é uma ferramenta pública e gratuita para geração e manipulação de mapas. Entretanto, em sistema sujeitos à muitas requisições, deve ser utilizada a versão paga da API. O Google Maps API requer uma

chave de autenticação. A chave é gratuita com limitação até 25000 requisições mensais, que é um valor bem satisfatório para a maioria dos usos. De qualquer forma, é possível adquirir um número maior de requisições quando pagas.

API suporta a adição de mapas as telas da plataforma, possibilitando a visualização da localização de um usuário. Além disso, ela tem funções que permitem a manipulação de objetos e dados que necessitam ser processados para a utilização dos algoritmos do sistema Venn.

### 3.5 FIREBASE

Neste projeto, a utilização de um banco de dados é imprescindível, tanto para armazenamento dos dados gerados pelos usuários, como processamento em back-end. Para este projeto ainda não é necessário o uso de tecnologias que permitam uma escalabilidade grande, sendo possível a utilização de ferramentas como o Firebase (Firebase, 2020) para a construção do protótipo.

O Firebase é um Back-end as a Service (Baas) para aplicações web e mobile. Ele foi lançado em 2004 e sofreu várias evoluções ao longo dos anos. Ele possui inúmeras funcionalidades que facilitam o processo de desenvolvimento e ajudam na identificação de falhas com os usuários. Ele possui serviços divididos em quatro categorias: Analytics, Develop, Grow, Earn.

Com o suporte do Analytics, existe a possibilidade de analisar o comportamento dos usuários, acompanhar possíveis falhas no sistema e outras funcionalidades. No Develop há disposição da ferramenta Realtime Database, que se trata de um banco de dados em tempo real. Ele possibilita o armazenamento e sincronismo dos dados entre usuários e dispositivos em tempo real através da utilização de um banco de dados NoSQL hospedado na nuvem. Outra ferramenta pertencente a categoria Develop é o Authentication, que oferece serviços de autenticação através de e-mail/senha, provedores de terceiros, como o Google ou Facebook, login anônimo, confirmação via SMS ou o uso direto do seu sistema de contas. Assim não se necessita construir todo o sistema de autenticação, o que atrasa o desenvolvimento.

Há também o serviço AdMob pertencente a categoria Earn, que é uma plataforma de publicidade usada para gerar receita através da apresentação de banners ou vídeos de empresas que anunciam no Google Ads. Os retornos das propagandas são relativos ao número de impressões e ao tempo de tela médio dos aplicativos.

### 3.6 ADOBE XD E DESIGN UX

No processo inicial de desenvolvimento da interface com o usuário, foi utilizado o Adobe XD (Adobe, 2019). Ele é uma ferramenta de desenvolvimento de interfaces se baseando em construir uma experiência de usuário que não seja difícil de utilizar, para isso existe um campo de estudo chamado de design UX, que trabalha para mapear a jornada do usuário a partir do contato com produtos.

Para desenvolver a interface podem ser utilizadas pesquisas de campo através da experimentação do usuário ou da utilização de conceitos já validados por pesquisas executadas. A utilização de pesquisas executadas é bastante eficaz, uma vez que não são inseridas preferências pessoais a avaliação do design.



## 4 O SISTEMA VENN

Este Capítulo apresenta o sistema Venn. O sistema Venn é composto pelos seguintes módulos, conforme Figura 4.1:

- aplicativo móvel utilizado para realizar o interfaceamento com o usuário,
- base de dados compartilhada,
- aplicativo para cadastramento e gerenciamento de viagens pelo motoristas, e
- cálculo de rotas, executados pelo servidor de aplicação.

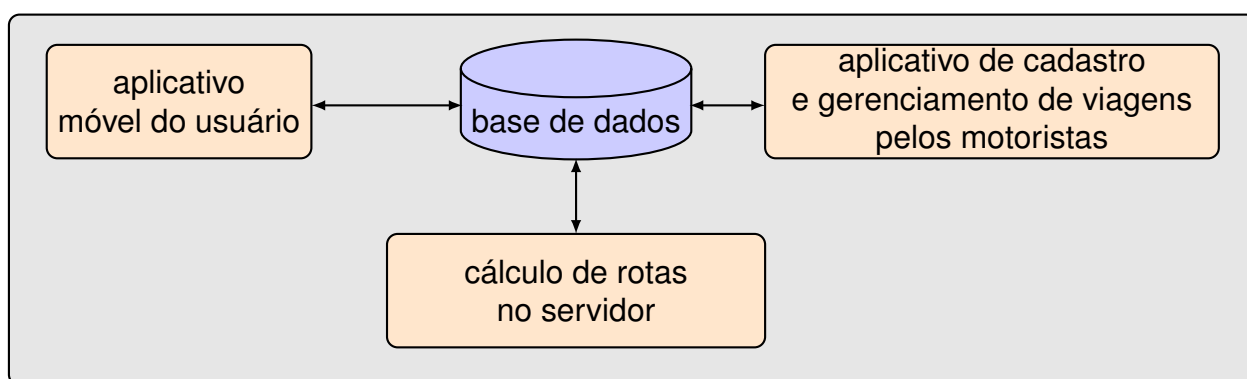


Figura 4.1 – Esquema simplificado da arquitetura do sistema Venn.

O sistema Venn foi projetado com base nos quatro módulos principais, para garantir um funcionamento completo. Uma representação destes módulos e respectivas tecnologias é representado na Figura 4.2.

O aplicativo Venn (Figura 4.2 (1)) é responsável pelo *front-end* do sistema, que se trata da interface de contato com o usuário que é transportado. Os dados das viagens, usuários e motoristas são armazenados no banco de dados (Figura 4.2 (2)).

O motorista tem suma importância no projeto, pois ele será o responsável por fornecer as viagens necessárias ao devido funcionamento do sistema. Como exemplificado, Figura 4.2 (3), o motorista deve comunicar no seu aplicativo ele quando está disponível, enviando seus dados de localização e se ele já está realizando um trajeto no momento. Após o envio dessas informações, o aplicativo (Figura 4.2(3)) define as rotas e passageiros que o motorista irá receber.

Para a construção do aplicativo, existem diversas opções de tecnologia que podem ser utilizadas. Na parte dedicada ao aplicativo, foi escolhido o React Native devido à possibilidade de construção de um aplicativo nativo, tanto para iOS quanto para Android realizando apenas alguns ajustes. O desenvolvimento focou na construção para a plataforma Android para a finalização do protótipo.

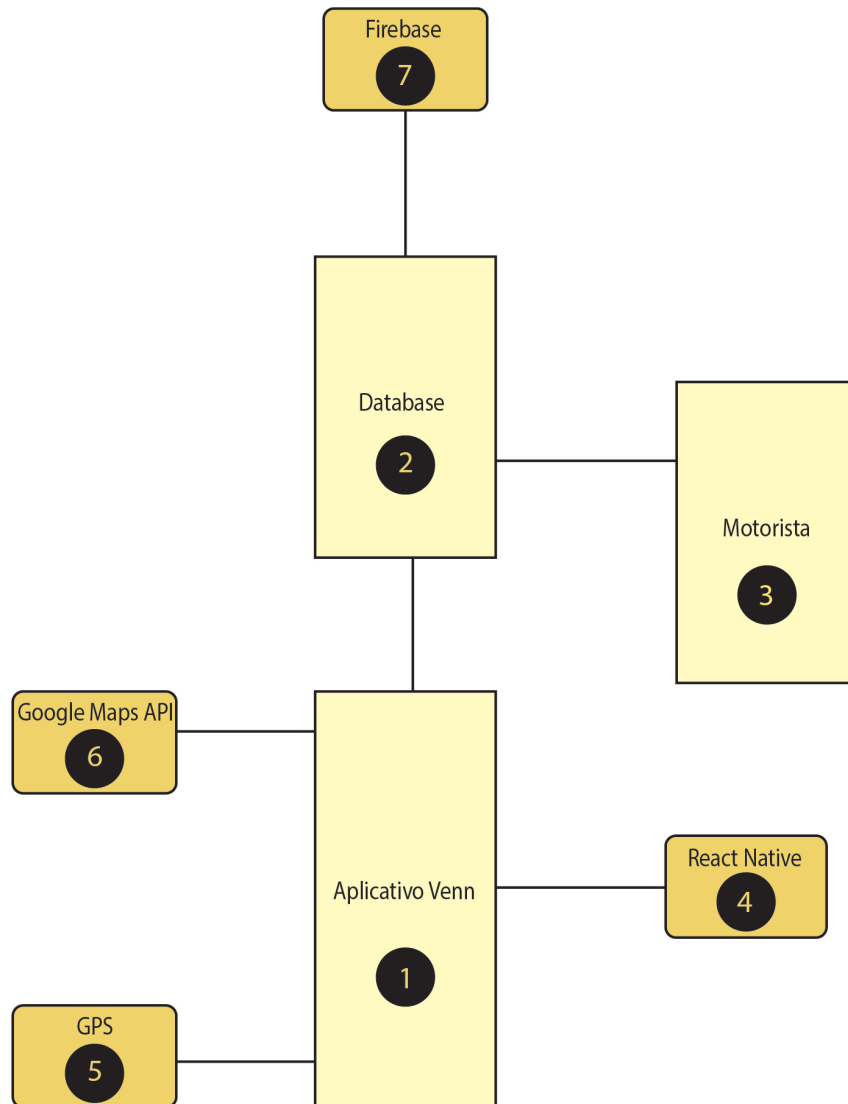


Figura 4.2 – Arquitetura com tecnologias de software do sistema Venn.

Na Figura 4.2, está representado nos números 1-3, os três módulos necessários: Database, aplicativo Motorista, e aplicativo usuário Venn. Também está exemplificado em Figura 4.2 (4-7), onde cada tecnologia citada anteriormente atua.

Além do framework React Native, foram usadas outras APIs que fornecem conexão a serviços não nativos da plataforma. Destaca-se a Google Maps API e o GPS.

A camada de mapas do aplicativo desenvolvido neste trabalho é oferecida pela Google Maps API, que possui fácil configuração e a possibilidade de ser utilizada de forma gratuita para testes e nos iniciais do aplicativo, onde não há a necessidade de infraestrutura que suporte tráfego muito alto de dados e serviços.

O GPS executa a trilateração da posição do usuário tornando possível definir a localização do usuário e, possivelmente, a origem da viagem. A partir da posição de origem, quando o destino é informado, pode-se calcular o tempo e percurso da viagem. Para a utilização do GPS do celular é necessário a utilização de ferramentas

que facilitem o interfaceamento e a utilização dos dados e localização. Neste projeto foi inserida a biblioteca React Native Geocoding, com a finalidade de obter esses dados.

#### 4.1 MODELO DE NEGÓCIO

O modelo de negócio adotado no Venn é baseado em modelos já validados no mercado. Em linhas gerais, a taxa aplicada será calculada com base na distância percorrida, tempo de viagem e demanda do serviço. Além disso, poderá haver variação do valor se o usuário for adepto de planos disponibilizados no aplicativo.

Outros aplicativos apresentados no projeto, como CityBus e Ubus, são sistemas com modelo de negócio já sólidos. Além disso, outros aplicativos que oferecem serviços de carona, como Uber, também funcionam de forma análoga a utilizada neste trabalho. Em síntese, no modelo, será aplicado uma taxa de uso ao motorista que disponibiliza seu veículo para transporte.

##### 4.1.1 Cenários

Para a plataforma Venn, foram analisados dois cenários descritos no texto que segue. Uma previsão de cenários mais completa demanda a utilização de diversos dados coletados em um extenso período de tempo. Após a coleta dos dados, é necessário o processamento dos mesmos, e posterior validação. Tendo em vista a complexidade desta análise, neste trabalho, foram utilizadas formas mais simples para análise. Assim fica a cargo de trabalhos futuros a realização de um estudo mais detalhado sobre o modelo do negócio.

###### 4.1.1.1 *Cenário otimista*

No cenário otimista, a Venn consegue inúmeras vans e micro-ônibus para trabalharem como parceiros. Com alta disponibilidade de veículos, aumenta a possibilidade de garantir uma oferta eficiente de veículos, suprimindo a demanda de viagens solicitadas.

Outra vantagem desse cenário seria a maior adoção por parte dos usuários, devido à menor tarifa que estaria sendo cobrada do serviço. Uma boa visão inicial por parte de seus clientes garante uma reputação incrível a empresa.

#### 4.1.1.2 *Cenário pessimista*

O cenário pessimista enfoca situações onde interesse por vans e micro-ônibus na plataforma é baixo. Neste cenário, a empresa sofre com um deficit de parceiros, o que compromete o fornecimento dos serviços.

Diferente da imagem eficiente que existiria no cenário otimista, aqui acontece uma imagem prejudicada pelos preços mais altos e demora na realização das viagens. Assim seria necessário um maior investimento inicial na tentativa de reduzir o valor da tarifa, ao menos até atrair novos parceiros para realizarem as viagens.

#### 4.1.2 **Lucros**

Os lucros da plataforma Venn, como mencionados anteriormente, ocorrem através de uma **taxa fixa de uso paga pelos motoristas** e um **adicional relacionado com a duração das viagens**. A taxa fixa dos motoristas depende do tempo de permanência dos motoristas no aplicativo. Quanto mais tempo o motorista está associado à plataforma, menor é a taxa.

O valor adicional da duração das viagens é proporcional ao valor pago pelas viagens de cada usuário. O valor da viagem varia de acordo com a duração da viagem e a quantidade de passageiros no veículo. Quanto mais veículos disponíveis, menor é o custo da viagem. Para o Venn, esse valor foi definido como sendo 17% do preço total da viagem de cada passageiro.

O motorista recebe 83% do valor total da viagem. A taxa fixa cobrada ao usuário é de R\$ 3,50 até 30 min, sendo acrescida de R\$ 1,00 a cada 10 minutos extras de viagens ou de R\$ 0,60 a cada quilometro extra rodado para caso a viagem ultrapasse 10 quilômetros. Os preços variáveis seguem a lógica da Figura 4.3. Com a formação de grupos, o valor decrementa em 5% a cada pessoa inclusa.

Quanto a tributação, esse tipo de sistema segue um modelo igual aos veículos de transporte privado como Uber e 99. Dessa forma é visível de que esses custos não serão elevados para a Venn.

O usuário tem a oportunidade de adquirir planos. Com estas opções, o usuário assina o serviço por um período de tempo e o valor é reduzido, já que aumenta a segurança dos recebíveis. Além disso, a assinatura com fidelidade garante ao usuário descontos e assim, vantagens. Por exemplo, no caso de ser assinado o plano de 1 mês ilimitado, o valor final proposto poderia ser de R\$ 149,90. Esse valor se for considerado um mês com 22 dias úteis e duas viagens por dia, para um custo unitário de viagem de R\$ 3,40. Para um usuário que utilize todos os dias úteis, a diferença nesse plano fica em R\$ 4,40, além disso usuário ainda poderá utilizar nos finais de

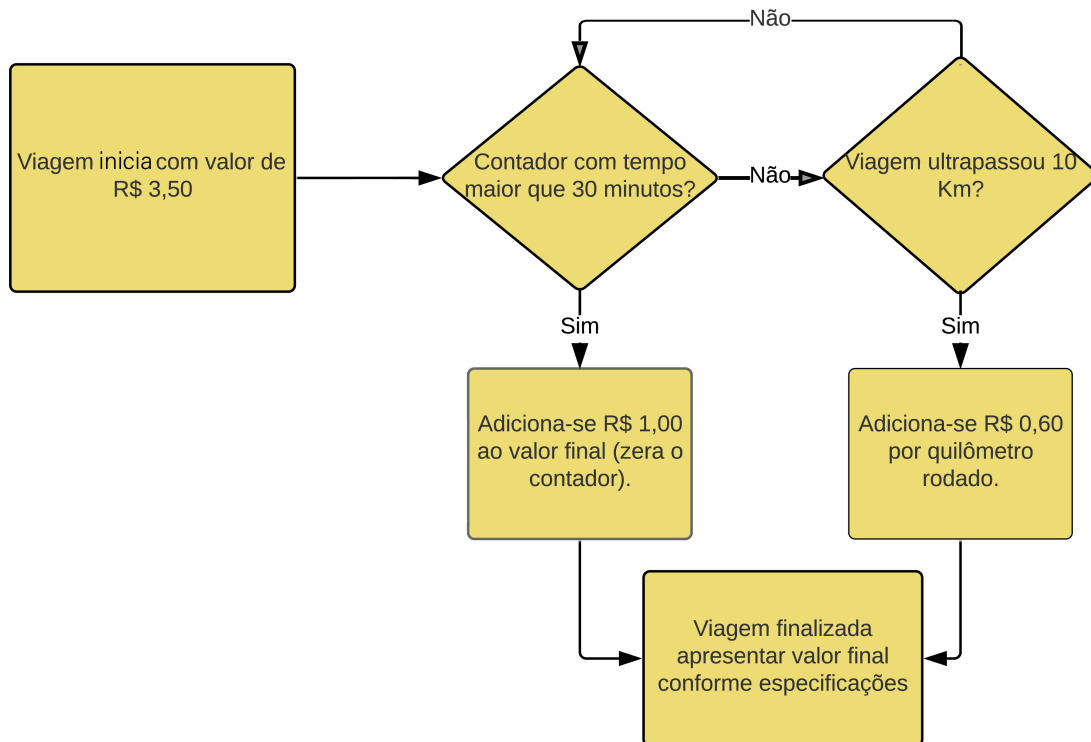


Figura 4.3 – Diagrama com esquema de preços Venn.

semana e feriados sem acréscimo nenhum.

Os veículos que podem atender os usuários do sistema Venn tendem a permanecer parados enquanto não estão realizando uma viagem previamente marcada. Devido a isso, o sistema proporciona redução do tempo ocioso ao proprietário que ficaria com o veículo sem uso durante um período de tempo. Ademais, o sistema possibilita que o veículo não circule em dias que há outras viagens marcadas, o que dá flexibilidade ao processo.

Os outros aplicativos semelhantes no mercado, não oferecem os mesmos serviços que o sistema Venn e por isso não seriam concorrentes diretos. Além disso, Ubus e CityBus, que possuem o modelo de funcionamento mais semelhante, não possuem a opção de planos. Ela pode ser atrativa ao motorista que pode ter uma segurança maior de que o sistema continuará por necessitar de corridas e conseqüentemente o veículo dele irá rodar.

Destaca-se que, conforme (LIGHTFOOT; LIMITED, 2011), inicialmente será oferecida uma estrutura simples de tarifas e preços, que pode ser revisada com base em fatores como a experiência de uso e comportamento dos clientes.

De fato, a tarifa cobrada aos clientes deve absorver as taxas administrativas, a taxa de inscrição do motorista, a cobrança mensal ou de anuidade, além de tarifas de tempo e distância, relativas a cada viagem, e outros elementos como seguro de viagem, pagamento de combustível, manutenção do veículo e outros encargos. Enfim, o sistema Venn precisa ser sustentável.

Finalmente, também podem ser aplicadas multas e itens reembolsáveis, que precisam ser definidas na cláusula de contratação de serviço pelos motoristas/proprietários de vans.

#### 4.2 ALGORITMOS PARA CALCULO DE ROTA DISPONÍVEIS PARA SEREM APLICADOS

Um artigo publicado na revista *Proceedings of the National Academy of Sciences* (ALONSO-MORA SAMITHA SAMARANAYAKE, 2017) aponta que 3 mil veículos com capacidade para 4 pessoas poderiam transportar 94% da demanda da cidade de Nova Iorque, nos Estados Unidos. Da mesma forma, 2 mil veículos com capacidade para 10 pessoas poderiam atender 95% da mesma demanda.

O estudo apresenta uma simulação utilizando um algoritmo. Nele, é possível realocar carros ociosos. Isso permite um sistema 20% mais rápido, já que sugere rotas com maior número de passageiros.

Um dos requisitos importantes ao servidor Venn é a implementação de um algoritmo otimizado o compartilhamento de caronas para diferentes demandas. A otimização do esquema de viagens proporcionará maior capacidade de atendimentos, tendo em vista que menores distâncias poderão ser utilizadas, evitando que um passageiro fique muito tempo esperando chegar ao seu destino.

O algoritmo para o cálculo do compartilhamento de caronas foge do escopo deste projeto. Uma recente revisão da literatura deste assunto é apresentada em (MOURAD; PUCHINGER; CHU, 2019).

O compartilhamento de caronas com o enfoque no perfil dos usuários (AYDIN; GOKASAR; KALAN, 2020), ocorre com base em características como sexo, idade, emprego e sua vontade de conhecer novas pessoas. Nesse algoritmo, além do destino próximo, são analisados os fatores sociais para o *match* entre passageiros e motoristas.

Quando forem realizados os testes em trabalhos futuros, uma conclusão do algoritmo mais adequado possibilitará uma redução nos custos computacionais bem como operacionais para os veículos inclusos no sistema.

O problema enunciado para as rotas de viagens se assemelha à solução do "Problema do Caixeiro Viajante com Caronas Múltiplas (PCV-MCA)"(ARAÚJO, 2016). No trabalho, é feita uma comparação entre os algoritmos: exato, evolucionários e colônia de formigas.

Então os algoritmos anteriormente descritos buscam solucionar este problema que pode ser adaptado para a solução das múltiplas caronas do sistema Venn.

## 5 APLICATIVO MÓVEL PARA OS USUÁRIOS VENN

Este presente trabalho foca na construção da interface com o usuário do sistema Venn. A interface deve garantir fluidez e facilidade na utilização do mesmo. Sendo assim, há um número limitado de telas, para evitar confusões ou complexidades desnecessárias ao funcionamento do projeto.

Os requisitos são necessários para o aplicativo dos usuários incluem:

- visualização das viagens realizadas,
- atualização dos veículos que estão rodando nas proximidades,
- verificação de login para segurança, e
- pagamento direto através de cartão de crédito

Esses requisitos garantem uma experiência mais automatizada e segura ao usuário, e serão enfocados neste texto nas seções 5.4-5.7.

O sistema Venn utiliza um aplicativo mobile, responsável por fornecer o meio necessário para realizar a solicitação de uma nova viagem. Também é necessário a utilização de uma tecnologia que realizará os cálculos da rota necessária.

Assim que uma trajeto for solicitado por um usuário através do aplicativo, a rota é calculada estimando o tempo que ela deve esperar e o tempo total do trajeto. Existem soluções semelhantes a essas, no entanto elas utilizam veículos próprios ou são destinadas a viagens intermunicipais. A solução proposta possibilitará que diversas pessoas ou empresas que possuem um veículo adequado possam operar no sistema.

### 5.1 FLUXO DE DADOS E VISÃO GERAL

Tomando como base a tarefa de solicitação de uma viagem na plataforma Venn, o diagrama apresentado na Figura 5.1 inicia a sequência de passos quando um usuário solicita o embarque.

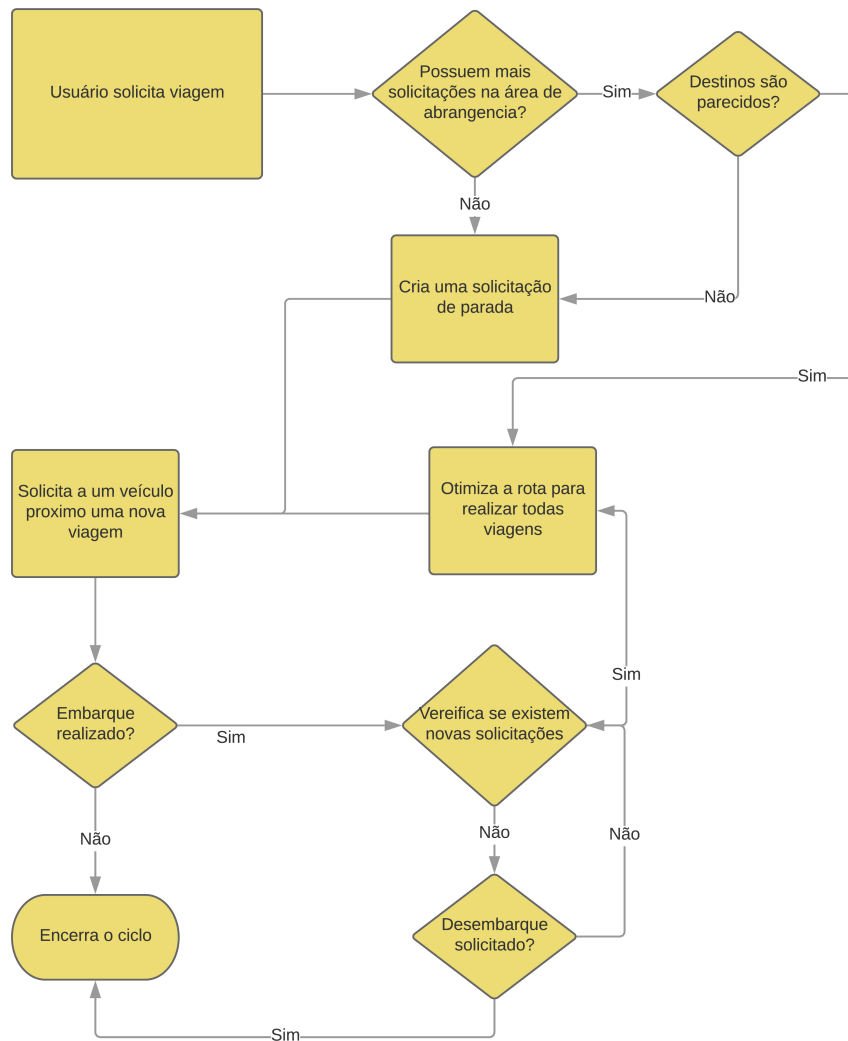


Figura 5.1 – Diagrama com a sequência de passos para a solicitação de embarque na Venn e atualização de rotas, e desembarque para finalização de ciclo de sequência de passos.



Quando a solicitação de um embarque ocorre por um passageiro, o sistema Venn verifica se há mais solicitações na área limitada - atendidas ou não - por uma circunferência de raio  $R$ , conforme a Figura 5.2, se os destinos coincidem com o mesmo bairro. Um veículo em curso pode atender uma solicitação. As solicitações são avaliadas e resolvidas pelo algoritmo de compartilhamento de viagens, executado pelo servidor Venn, que foge do escopo deste trabalho.

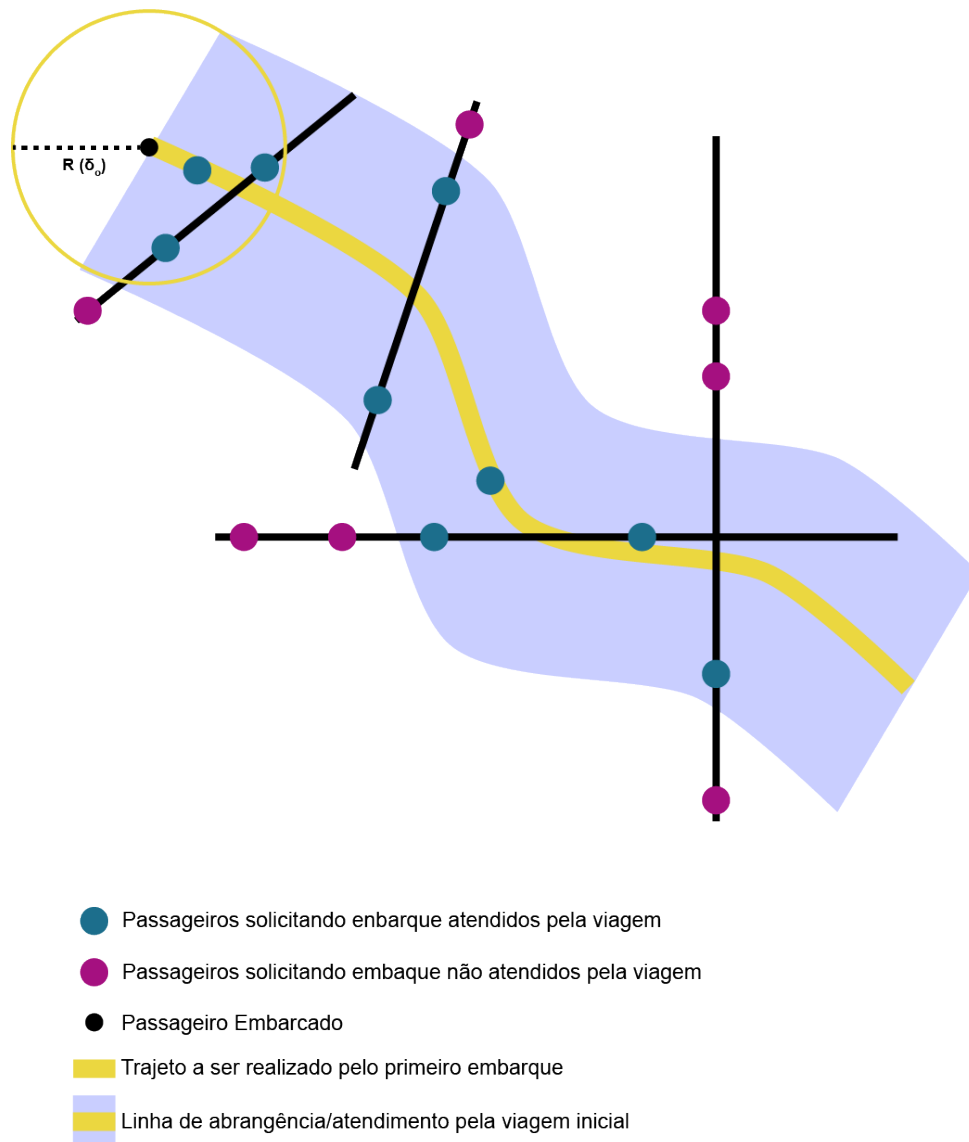


Figura 5.2 – Raio  $\delta_o$  demonstrando quais passageiros seriam atendidos com a viagem indicada.

Se não houver essa coincidência, a viagem é privada. Essa situação é descrita pela Figura 5.3.

Neste trabalho, foram usados os valores 500 m e 1,5 Km (i.e.,  $500 < r < 1,5km$ ), para os limites inferior e superior, respectivamente.

Conforme o processamento desses dados, o sistema gera uma rota e solicita a um veículo para realizá-la. Após o embarque do usuário, o sistema ainda verifica se

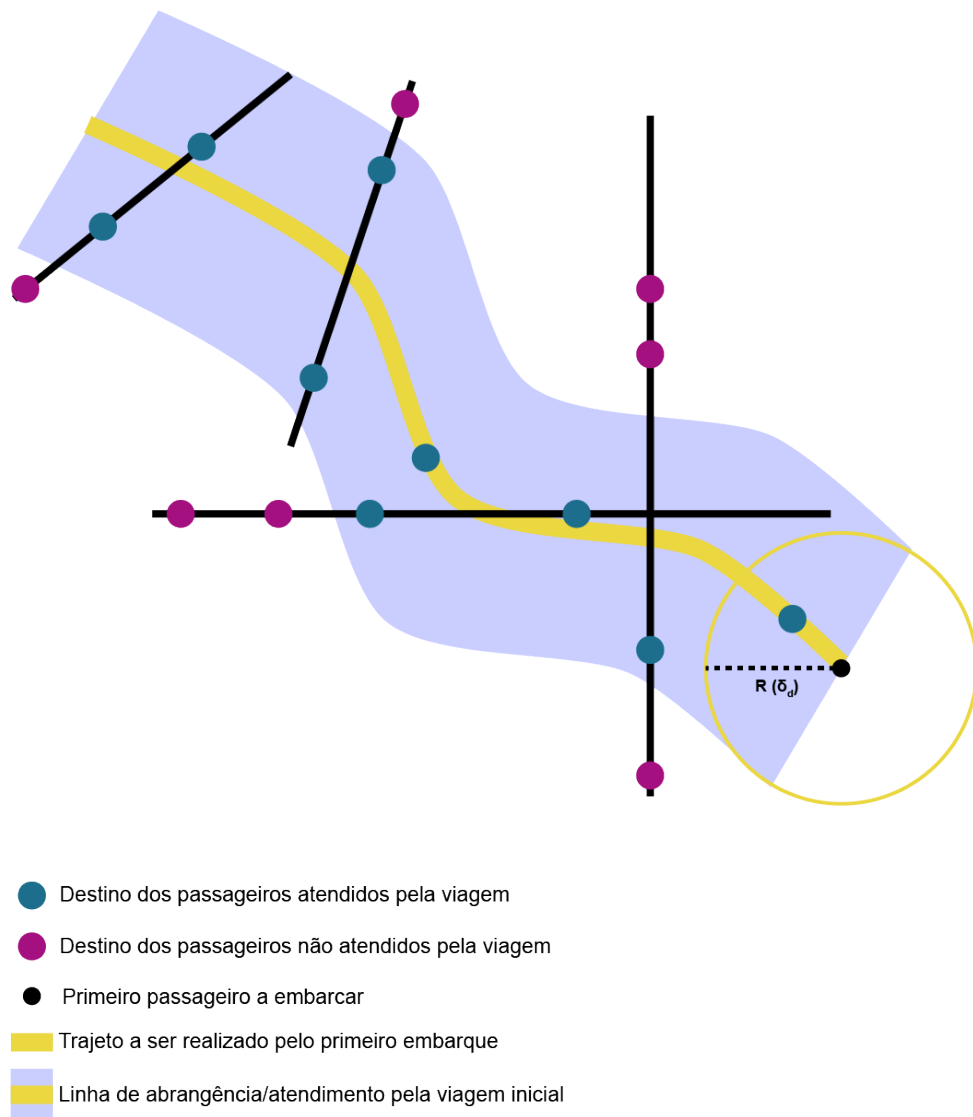


Figura 5.3 – Raio  $\delta_d$  demonstrando quais passageiros que possivelmente seriam atendidos com a viagem indicada.

existe alguma outra solicitação de viagem nas imediações. O ciclo se encerra quando os passageiros desembarcam.

## 5.2 INSTALAÇÃO DE SOFTWARE E CONFIGURAÇÃO DOS DISPOSITIVOS

O primeiro passo para inicializar o desenvolvimento do aplicativo é a instalação do Node.js, que pode ser definido como um ambiente de execução Javascript *server-side*. O Node.js suporta a execução do programa no ambiente de desenvolvimento. O download pode ser realizado diretamente através do site da organização disponível em <<http://nodejs.org>>.

O próximo pacote a ser instalado é a CLI do React Native. Esse processo deve

ser realizado pelo terminal de comando do ambiente de desenvolvimento. O comando para inicializar a instalação da CLI é dado por:

```
npm install -g react-native-cli [] .
```

Foi utilizado o gerenciador de pacotes *npm* para realizar a instalação e o *-g* é utilizado para instalar o pacote de forma global no projeto.

Com as tecnologias básicas já instaladas, a partir de agora é necessário inicializar a aplicação do React Native para inicializar o desenvolvimento do aplicativo. Esse procedimento também é realizado através do terminal de comando executando-se:

```
react-native init venn,
```

sendo que o Venn trata-se do nome que deve ser dado ao projeto. No momento de geração da aplicação, a versão corrente do React Native era 0.61.4, que ainda não é considerada uma versão estável podendo apresentar bugs durante o processo de desenvolvimento. No entanto, esses problemas não afetam na qualidade da aplicação final.

Após a criação da estrutura base, pode ser iniciado o desenvolvimento através da estrutura base gerada. Nesse momento, é seguida uma arquitetura de pastas conforme apresenta a Figura 5.4. A pasta *android* possui configurações referentes a execução e geração da aplicação para a plataforma Android. Existem configurações que estão disponíveis apenas em um dos sistemas operacionais. Da mesma forma, na pasta *iOS*, se encontram as configurações referentes ao sistema operacional *iOS*. Chaves específicas de cada sistema operacional devem ser inseridas na sua respectiva pasta.

Dentro da pasta *src* existe o arquivo *index.js* para a configuração de rotas dentro da aplicação. As rotas são responsáveis pela definição de quais telas o usuário deve ser direcionado ao realizar uma ação na aplicação. No entanto, antes da definição das rotas é necessário a instalação de um pacote que realiza o processo de navegação pelas páginas do aplicativo.

O pacote instalado para gerir as rotas da aplicação é o React Navigation, que deve ser acompanhado do React Navigation Drawer e do React Navigation Stack. Esse pacote possui fácil instalação basta realizar em sequência os comandos:

```
npm install react-navigation,  
npm install react-navigation-drawer,  
npm install react-navigation-stack.
```

Neste trabalho, foram usadas as seguintes versões para a implementação do protótipo do cliente Venn: *react-navigation* versão 4.0.10, *react-navigation-drawer* versão 2.3.3 e *react-navigation-stack* versão 1.10.3. As duas últimas são aplicações de código aberto mantidas pela comunidade de desenvolvedores React Native.

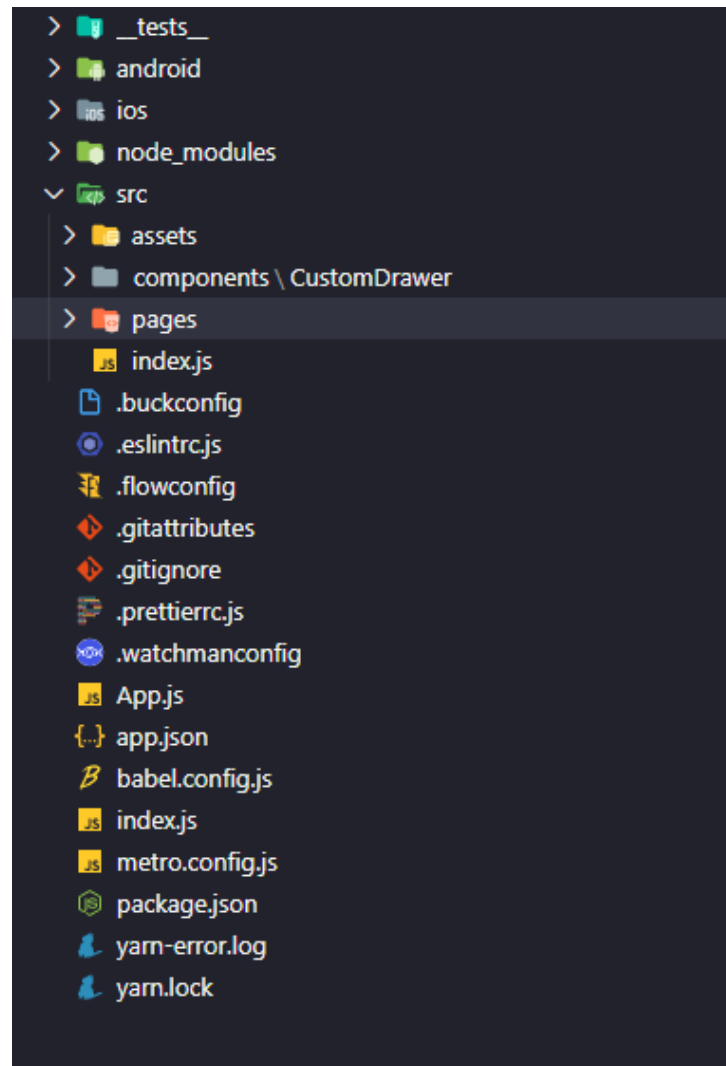


Figura 5.4 – Estrutura de pastas e arquivos.

Fonte: Autor

Na sequência, pode ser adicionado o mapa a tela inicial do aplicativo. Como apresentado anteriormente foi utilizado o Google Maps API. O Google Maps API requer a configuração da chave de API na Google Cloud Platform. Com a chave de utilização disponível, é necessário incluir a biblioteca React Native Maps na aplicação, a mesma é realizada com a execução do comando:

```
npm install react-native-maps.
```

Na sequência, a chave de API deve ser inserida ao arquivo que está localizado na pasta exemplificado pela Figura 5.5. Com a chave inserida o mapa já pode ser utilizado na aplicação.

Além dessas ferramentas, neste trabalho foi utilizado o suporte d banco de dados *firebase* para armazenamento e processamento de dados em backend.

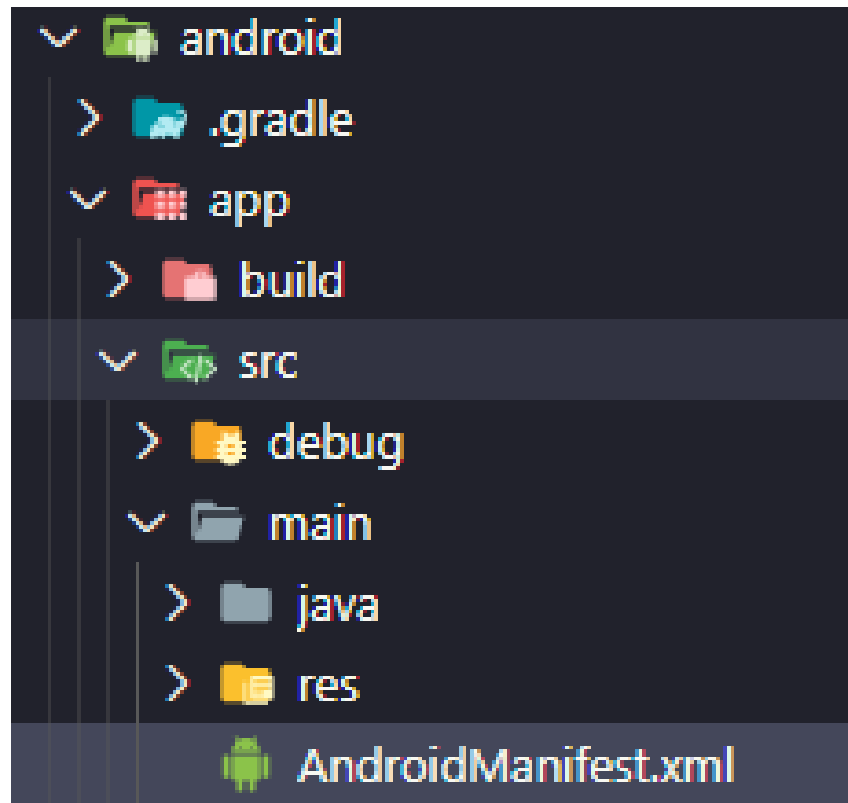


Figura 5.5 – Adição da chave de API.

### 5.3 MOCKUP DO APLICATIVO

O primeiro passo no desenvolvimento do aplicativo foi o projeto e a criação de uma identidade visual que garantisse simplicidade aliada a uma usabilidade satisfatória. Assim, se fez necessário a utilização de um software de design gráfico, neste caso o Adobe XD. Cada uma das telas de base foram criadas a partir da utilização deste software.

A primeira tela projetada foi a splash-screen. Conforme mostra a Figura 5.6(a), a tela possui estilo simples, mas descreve exatamente o que a aplicação é destinada a realizar. Tem a função de impedir que o usuário encerre a abertura antes da utilização do aplicativo.

### 5.4 VISUALIZAÇÃO DAS VIAGENS REALIZADAS

No aplicativo, é interessante que se visualize um histórico das viagens realizadas para possibilitar verificações futuras a respeito do que foi realizado. Essa função também auxilia na comparação de gastos, para o caso de adoção do sistema Venn em relação aos concorrentes.

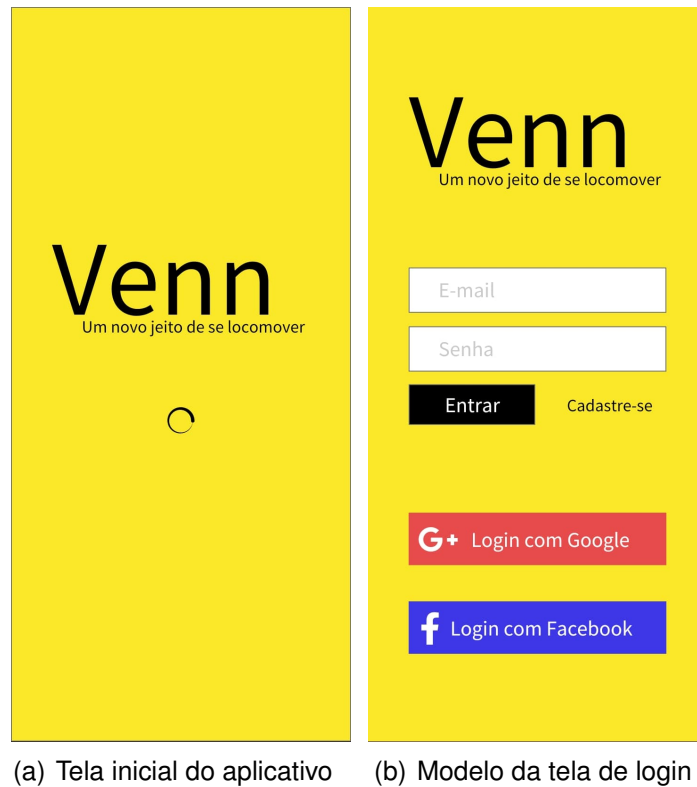


Figura 5.6 – Modelo de telas do aplicativo móvel para o usuário Venn.

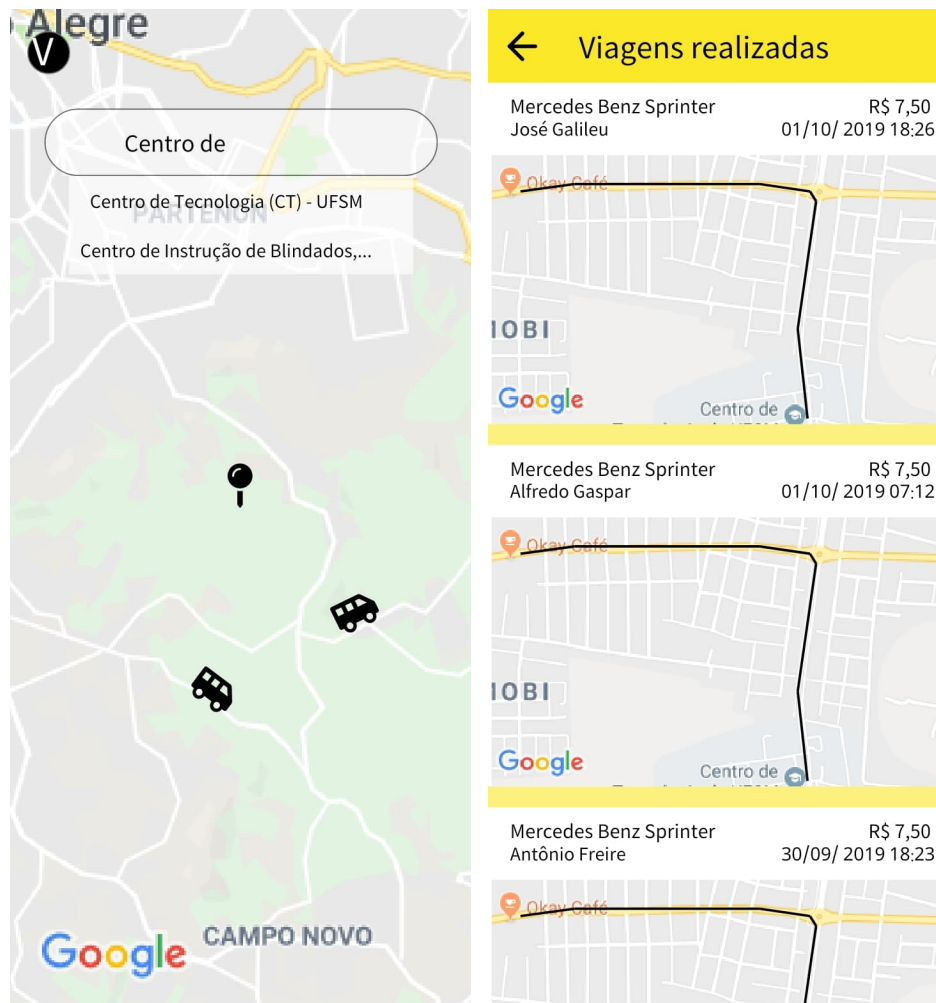
Um histórico de viagens se faz necessário para permitir que o usuário identifique quando realizou suas viagens, seus gastos estimados com o serviço, além de permitir que ele conteste dados caso ocorra algum erro no sistema e ele seja cobrado indevidamente. Na Figura 5.7(b) estão organizadas as informações referentes a três viagens fictícias realizadas, com indicação dos trajetos.

Nesta tela são apresentadas as informações de tipo de veículo que foi utilizado, custo total da viagem, Motorista que realizou e data e hora que a viagem foi realizada. No mesmo bloco ainda é apresentado o local de início e fim da viagem com o auxílio de um mapa.

## 5.5 ATUALIZAÇÃO DOS VEÍCULOS QUE ESTÃO RODANDO NAS PROXIMIDADES

Outra função é a visualização dos veículos nas proximidades. Está se faz necessária no momento em que usuários desejam ver qual modalidade de transporte melhor irá suprir as suas necessidades.

Esta função precisa do suporte do servidor Venn para ser implementada pois o servidor Venn mantém o controle do veículos que estão nas proximidades. Então, esta função será realizada em trabalhos futuros.



(a) Modelo da tela inicial de viagens do aplicativo (b) Modelo da tela com conjunto de viagens realizadas

Figura 5.7 – Modelo de telas com viagens e trajetos do aplicativo móvel para o usuário Venn.

## 5.6 VERIFICAÇÃO DE LOGIN PARA SEGURANÇA

Também é necessário fazer validações de identidade. São fundamentais para garantir a segurança dos usuários e também dos motoristas da plataforma. Essa validação pode ser feita pela exigência de um login e ou validação de documentos e antecedentes.

Assim que a splash-screen carrega a próxima tela é a de login, mostrada na Figura 5.6 (b). Esta é exibida uma única vez na abertura do aplicativo, ou em situações que o usuário limpe os dados da aplicação realizando assim logout. Para a construção e disposição dos itens nela são seguidos conceitos de design UX, prezando pela melhor usabilidade possível ao usuário (COYLE, 2016).

A disposição das informações foi colocada uma abaixo da outra na vertical, mantendo o fluxo de visualização dos dados de cima para baixo somente. foram

adicionadas formas para entrar através de redes sociais evitando a necessidade de digitar e lembrar mais de uma senha.

A próxima tela seguindo a sequência de abertura é a tela inicial conforme a 5.7(a). Esta deve conter um mapa para permitir que o usuário se localize, para isso foi definido que seria utilizado o Google Maps que possui a maior disponibilidade de locais para serem buscados.

Um campo de inserção de texto deve ser utilizado nesta tela, com a finalidade de buscar e escolher o local de destino desejado. Ele foi colocado na parte de cima sendo um dos principais pontos de visualização, essa pode ser confirmada em aplicativos como o Google Maps que tem uma usabilidade impecável.

## 5.7 PAGAMENTO DIRETO ATRAVÉS DE CARTÃO DE CRÉDITO

Uma automação que facilita o processo é o pagamento através do aplicativo, como é utilizado nos sistemas de ride-sharing. Essa adoção reduz o tempo em que o usuário leva para pagar pela viagem, além de facilitar para casos em que se necessitaria de troco ou que não é seguro utilizar dinheiro em mãos.

Com todas as funções inseridas é necessário que exista um controle das formas de pagamento, assim optou-se por desenvolver uma tela especialmente com essa função como está apresentada na Figura 5.8. Nela estão organizadas, no topo, as informações referentes aos planos de pagamento, com o intuito de aumentar a visibilidade dos mesmos tendo em vista que eles são o objetivo principal de fonte de receita da aplicação.

Abaixo dos planos encontra-se os cartões cadastrados e também um campo para adicionar novos cartões. Um campo dedicado à adição de saldo pré-pago foi incluído para permitir ao usuário controle máximo de gastos com o transporte, podendo este ser incluído também por boleto.

A API utilizada para o funcionamento desse sistema é a Pagarme<sup>1</sup>. Nela, é possível realizar todos os tipos de pagamento de forma integrada. Tanto pré-pago, quanto cobranças mensais e pagamentos pontuais com cartão de crédito. Ela é equipada com ferramentas anti-fraudes, como por exemplo: um teste de faturamento do valor no cartão de crédito, para garantir que é válido e com saldo. Por meio dela, também é possível gerar notas fiscais. Na Figura 5.9, é apresentado um exemplo de painel de controle dos lançamentos com valores de teste.

---

<sup>1</sup><https://docs.pagar.me>





Figura 5.8 – Modelo de tela do aplicativo do usuário Venn para pagamento do serviço.

## 5.8 TELAS DA APLICAÇÃO NATIVA EM DISPOSITIVO ANDROID

Nesta seção são apresentadas telas da aplicação desenvolvida. Elas foram baseadas nos Mockups construídos na seção anterior, não foi possível copiar com total fidelidade o design inicial devido a limitações tecnológicas. Apesar disso elas mantiveram os aspectos de usabilidade requisitados.

O desenvolvimento iniciou com a criação da aplicação React Native, a qual veio totalmente sem nenhuma característica gráfica, basicamente um aplicativo em branco apenas com os dados de compilação. Foi necessário também seguir o processo de instalação das bibliotecas utilizadas. Neste momento, foi iniciado o desenvolvimento da tela inicial do aplicativo, Figura 5.10(a), como fundo é utilizado o componente de mapa adicionado pela API do Google Maps. Além do mapa há um elemento para realizar a abertura e o fechamento do menu lateral, esse processo utiliza a biblioteca React Navigation instalada na configuração. No momento em que o botão é clicado ele muda o estado do *toggle* do menu lateral, permitindo realizar a navegação entre as diferentes telas que estão listadas nele. O último elemento inserido foi o campo de

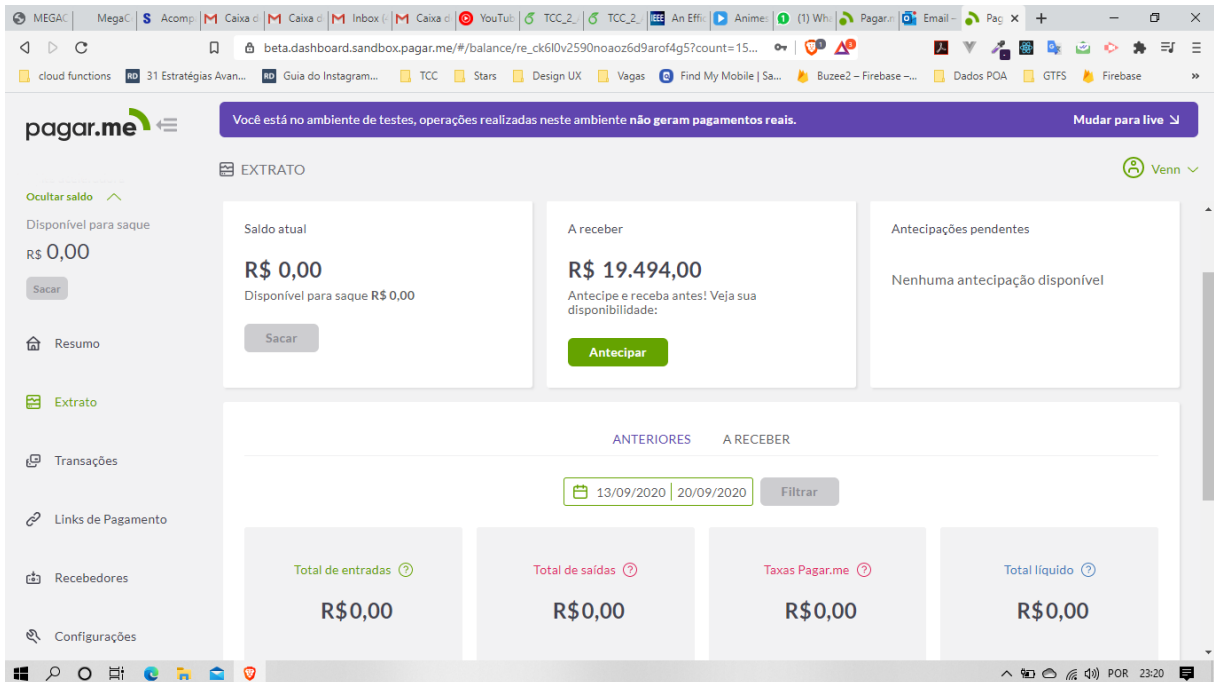


Figura 5.9 – Tela de gerenciamento do Pagar.me.

busca de destino, neste utilizamos o elemento *TextInput* do React Native, quando este campo for integrado ao backend que será realizado em trabalhos futuros ocorrerá a sugestão de destinos como estava exemplificado na tela do mockup.

Ao alterar o estado do *toggle* do menu lateral ele é aberto sendo possível visualizar os itens inseridos nele. O menu foi configurado para empurrar a tela principal para o lado quando ele for aberto como pode ser visto na Figura 5.10(b). Bem no topo existe a foto de perfil do usuário. Neste caso, é usada uma foto genérica devido ao fato do aplicativo não estar conectado ao backend. Logo abaixo está disposta a marca, identificando a missão da marca.

A partir do menu lateral, o usuário seleciona diferentes telas que estejam disponíveis para acesso, entre elas está a de Viagens Realizadas que ao ser selecionada no menu drawer o usuário é direcionado à tela conforme a Figura 5.11(a). Nesta estão representadas 3 viagens fictícias, utilizadas para o teste da interface. Cada bloco de viagem possui informações do veículo, motorista, custo e data e hora de realização da viagem. Também existe um mapa colocado em cada um dos blocos, quando conectado ao backend a rota deve ser traçada nele. Essa tela possui todas as informações necessárias para uma verificação de cobranças indevidas ou para revisar os valores investidos em transporte.

Além de visualizar as viagens realizadas é preciso uma forma de visualizar as opções de pagamentos, assinar um plano ou adicionar saldo a conta. Assim foi desenvolvido o frontend conforme a Figura 5.11(b). Na figura estão presentes todos os elementos requisitados no mockup. Para esta foi utilizada a mesma barra superior

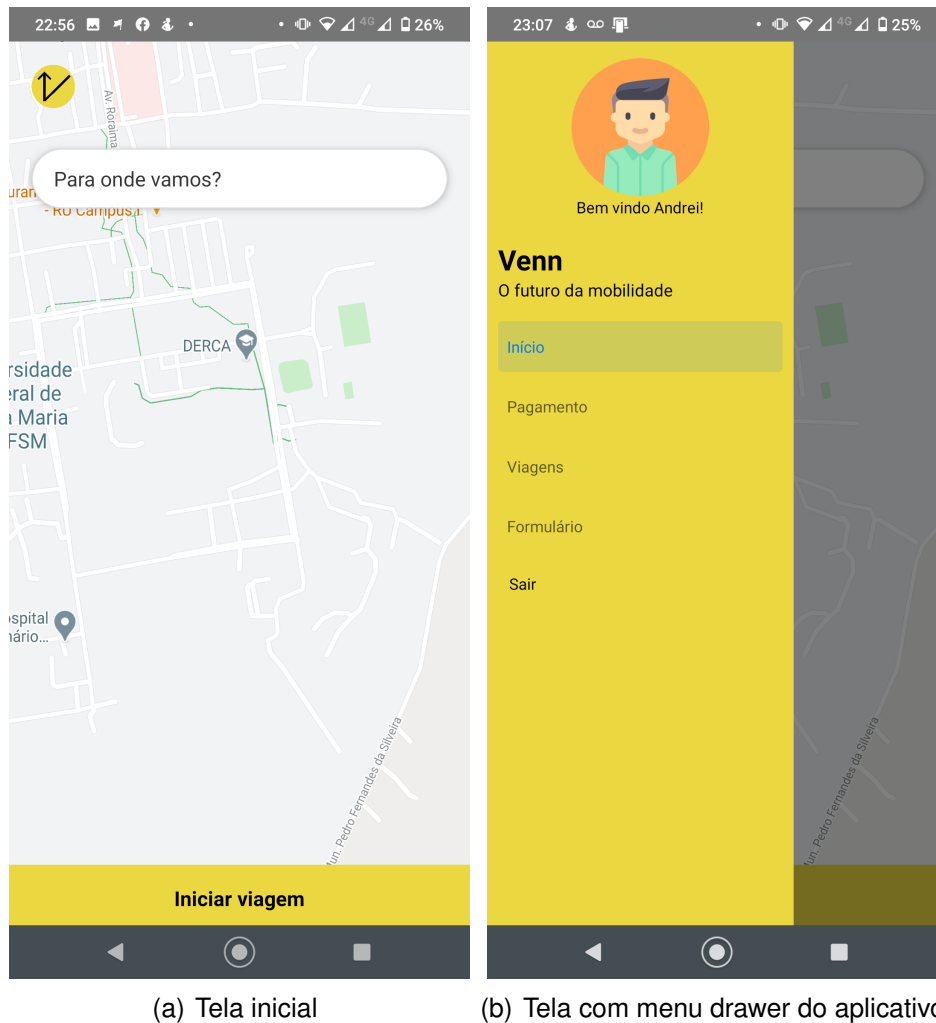


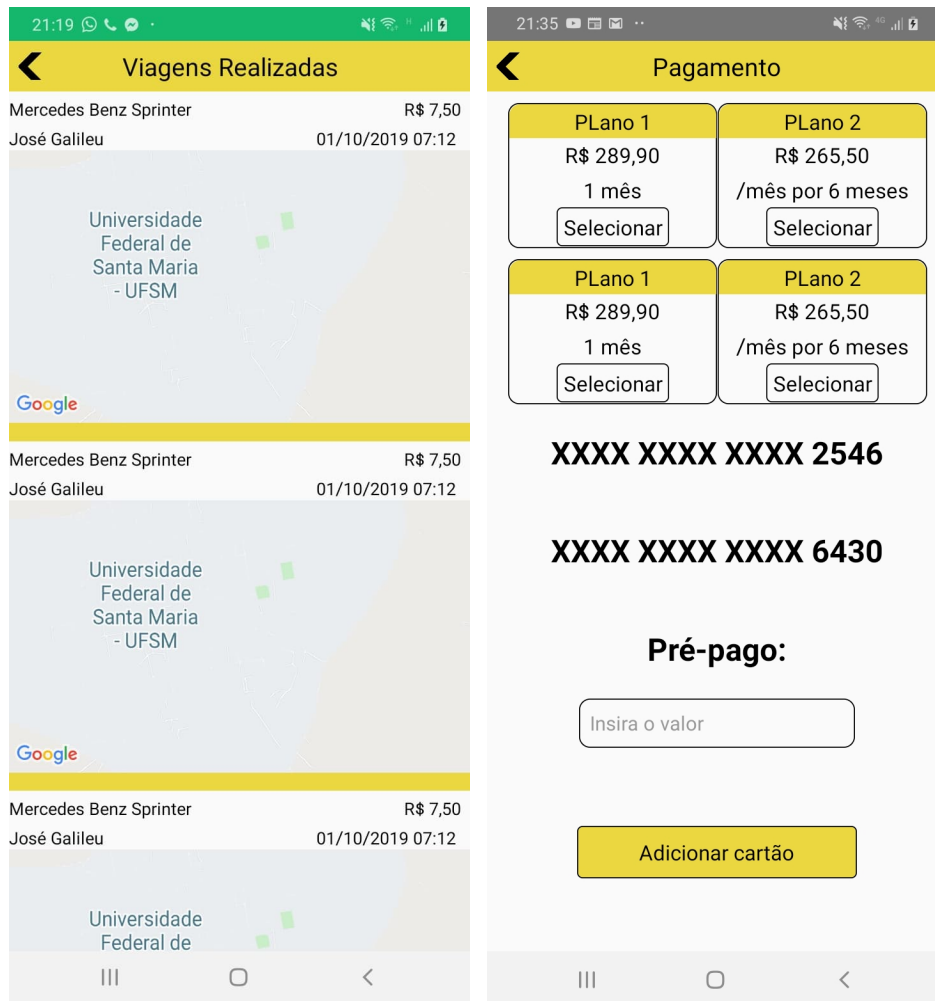
Figura 5.10 – Exemplos de telas do aplicativo do usuário Venn.

que a incluída na tela anterior, temos um bloco com planos para assinatura. Em cada linha do bloco de planos foi necessário transformar o elemento *View* em uma *row*, assim se utilizou 2 blocos para a criação dos planos, um para cada linha. Cada um dos planos possui um botão para realizar a escolha e direcionamento do mesmo.

Abaixo do modal de planos estão os cartões de créditos cadastrados com apenas os últimos números visíveis. Também é possível incluir crédito pré-pago através do elemento *TextInput*. Na parte mais baixa está um botão para adicionar um novo cartão.

Com a conexão e configuração do aplicativo com o banco de dados Firebase, foi possível a realização de autenticação na aplicação. O próprio banco de dados possui uma api de fácil utilização, realizando a persistência de estado do token de autenticação, permitindo acesso offline ao app.

Diante da necessidade atual de autenticação foram desenvolvidas duas telas, com essa finalidade na 5.12(a) temos a tela de Login e na 5.12(b) está a tela para criar um novo usuário. As duas telas seguiram padrões de usabilidade que garantem maior

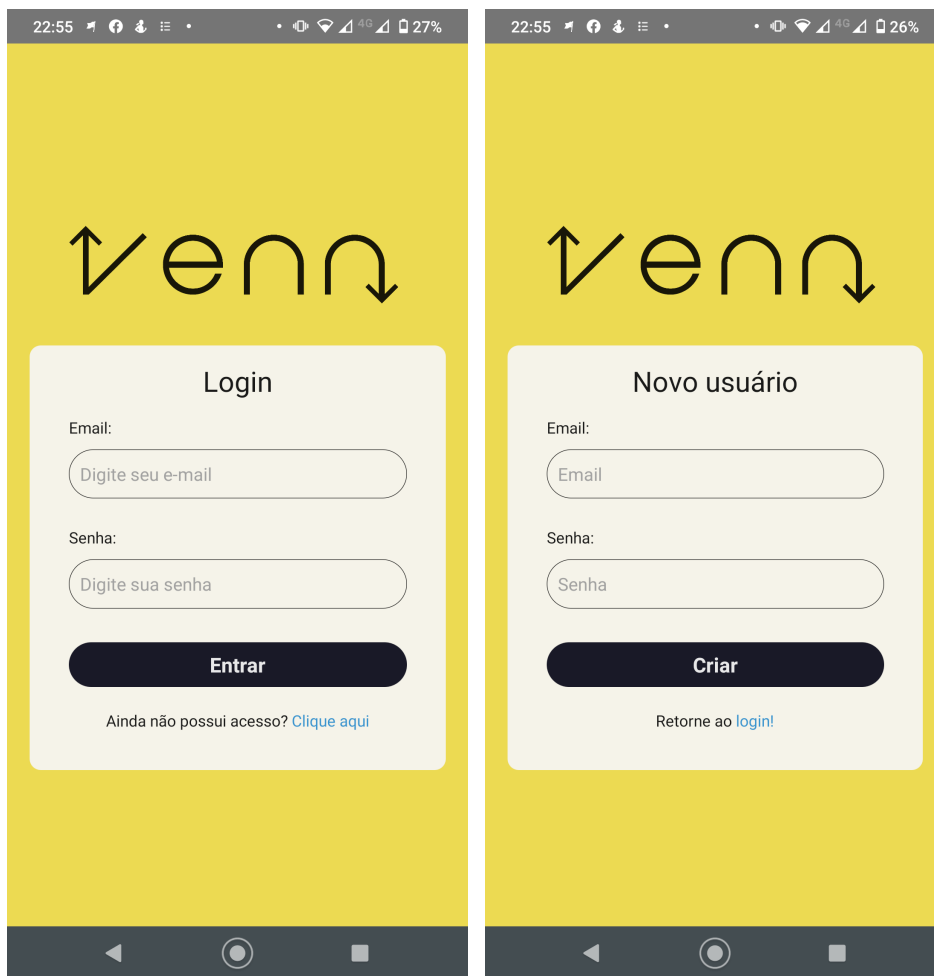


(a) Tela de Viagens Realizadas (b) Tela com menu drawer do aplicativo

Figura 5.11 – Telas do aplicativo do usuário Venn.

facilidade no acesso da aplicação.

A criação de um novo usuário deve ser verificada contra possíveis fraudes e também para garantir a segurança dos passageiros e motoristas. Com essa necessidade de uma certa segurança, após a criação de um novo usuário são solicitados dados pessoais que possam confirmar a identidade do usuário. Essa funcionalidade 5.13, bloqueia o acesso em caso de inconsistência nos dados ou de não preenchimento.



(a) Tela de Login

(b) Tela de novo usuário

Figura 5.12 – Telas do aplicativo do usuário Venn.

22:56

Para continuar você precisa preencher alguns dados

Nome \*

Telefone \*

CPF \*

Endereço

Entrar

(a) Tela de cadastro de dados pessoais

Figura 5.13 – Telas do aplicativo do usuário Venn.

## 6 CONCLUSÃO

### 6.1 APONTAMENTOS FINAIS

Este projeto apresentou uma plataforma para oferecer uma nova modalidade de transporte coletivo, com o intuito de reduzir os custos de transporte, bem como trazer mais qualidade ao mesmo. Esta plataforma une a capacidade de um transporte de pessoas em maior escala, com a comodidade de alguma privacidade oferecida pelos veículos de passeio. De forma mais específica, este relatório apresenta o processo de modelagem e construção do *frontend* do aplicativo Venn. São utilizadas técnicas de desenvolvimento de interfaces focando em uma melhor experiência do usuário.

A utilização do Adobe XD permitiu otimização do tempo necessário para a criação dos esboços iniciais de interface. Também permitiu fazer alguns testes de interação, os quais mostraram resultados satisfatórios, uma vez que para solicitar uma viagem foram necessários apenas três cliques. A utilização de técnicas de design UX melhorou muito o resultado final da aplicação

Após finalizados os esboços de interface partiu-se para a construção da aplicação nativa. O React Native mostrou-se uma escolha extremamente satisfatória, uma vez que foi possível entregar uma boa fidelidade ao projeto de design realizado no Adobe XD.

O desenvolvimento do aplicativo utilizou ferramentas modernas de programação, adquiridas principalmente por experiências anteriores. A utilização da experiência e de vivências com a startup pela qual fui sócio, permitiu a previsão de erros. Evitando com que eles ocorressem e fossem vistos somente em uma possível inserção do aplicativo no mercado sendo testado pelos usuários.

A utilização do Google Maps API foi satisfatória para a criação da interface de localização com o usuário. Com ela foi possível adicionar uma mapa as telas inicial e múltiplos, conforme o tamanho do histórico, a tela de viagens realizadas.

O GPS - Global Positioning System, permitiu encontrar atual do usuário e também permitir a definição dos destinos quando o backend estiver integrado. A precisão de localização varia conforme o modelo de celular, assim não foi possível definir um número exato.

O aplicativo foi integrado ao banco de dados. Dessa forma, é possível criar usuários e armazenar os seus dados. Também foi integrado a plataforma Pagarme, que possibilitou a realização de pagamentos com segurança.

Nesse trabalho também foram levantadas soluções para o algoritmo que vai direcionar os veículos. Nelas, são estabelecidos critérios para definir os passageiros

que irão juntos em determinada corrida.

Também foram realizados testes controlados com o aplicativo simulando o uso pelo usuário final. Esses testes mostraram que todas as funções até agora aplicadas estão funcionais.

Existe uma forte tendência de adesão por parte dos usuários para sistemas de ride-sharing. Logo, o sistema proposto tem grandes expectativas de receber um bom retorno financeiro, considerando o atual cenário de mobilidade brasileiro.

## 6.2 TRABALHOS FUTUROS

Como o trabalho não está totalmente concluído são necessárias novas ferramentas e integrações. Estima-se que o custo final desse desenvolvimento fique baixo, pois ele se baseia no desenvolvimento somente pelo executor que resulta no custo de horas apenas. O tempo para conclusão tende a se estender com apenas um executor do projeto, ficando em aproximadamente seis meses com dedicação exclusiva. Assim como trabalhos futuros, destaca-se:

- Desenvolvimento do algoritmo de cálculo de rotas otimizado. Podem ser utilizados como base os algoritmos para a solução do problema do caixeiro viajante, citados no trabalho;
- Criação da interface do motorista;
- Implantação da integração do aplicativo do motorista com o aplicativo do usuário, o consolidação com o servidor Venn.
- Teste da eficiência do modelo de preços criado a partir de simulações com diferentes cenários e demandas.



## REFERÊNCIAS BIBLIOGRÁFICAS

Adobe. **Recursos do XD**. São Paulo, 2019. Acesso em 01 mar. 2020. Disponível em: <<https://www.adobe.com/br/products/xd/details.html>>.

Agência de Notícias IBGE. **Número de pessoas que trabalham em veículos cresce 29,2%, maior alta da série**. 2019. Acesso em 11 fev. 2020. Disponível em: <<https://agenciadenoticias.ibge.gov.br>>.

Agência IBGE Notícias. **Um em cada três brasileiros mora em 48 municípios com mais de 500 mil habitantes**. 2019. Acesso em 10 fev. 2020. Disponível em: <<https://agenciadenoticias.ibge.gov.br>>.

ALONSO-MORA SAMITHA SAMARANAYAKE, A. W. E. F. D. R. J. High-capacity vehicle pooling and ride assignment. In: **Proceedings of the National Academy of Sciences**. [S.l.: s.n.], 2017. v. 114 (3), p. 462–467.

ANTP. **Revista dos Transportes Públicos**. São Paulo, 2019. Acesso em 25 fev. 2020. Disponível em: <<http://files.antp.org.br/2019/12/18/rtp153-e.pdf>>.

ARAÚJO, G. **Algoritmos Meta-heurísticos Para a Solução do Problema do Caixa-veículo Viajante com Múltiplas Caronas**. 2016. 73 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Natal, 2016.

AYDIN, O. F.; GOKASAR, I.; KALAN, O. Matching algorithm for improving ride-sharing by incorporating route splits and social factors. In: **PLoS ONE 15(3)**. [s.n.], 2020. Disponível em: <<https://doi.org/10.1371/journal.pone.0229674>>.

Buser. 2020. Acesso em 11 fev. 2020. Disponível em: <<http://www.buser.com.br>>.

CityBus 2.0. 2020. Acesso em 25 fev. 2020. Disponível em: <<https://citybusbr.com/>>.

COYLE, A. **Design Better Forms**. 2016.

Escritório Nacional de Coordenação para Posicionamento, Navegação e Momento Baseados no Espaço - EUA. **GPS**. 2019. Acesso em 25 fev. 2020. Disponível em: <<https://www.gps.gov/>>.

Firebase. 2020. Acesso em 25 fev. 2020. Disponível em: <<https://firebase.google.com/?hl=pt-br>>.

Google Maps API. 2020. Acesso em 25 fev. 2020. Disponível em: <<https://developers.google.com/maps/documentation?hl=pt-br>>.

LIGHTFOOT, G.; LIMITED, M. Deliverable 7.5 - guidelines for business plan to start up car-sharing. In: . [S.l.: s.n.], 2011.

LIU Z. YANG, X. W. Y.; JIAN, L. Location, localization, and localizability. **Journal of Computer Science and Technology**, v. 25, n. 2, p. 274–297, 2010.

MOURAD, A.; PUCHINGER, J.; CHU, C. A survey of models and algorithms for optimizing shared mobility. In: **Transportation Research Part B: Methodological**. [S.l.]: Elsevier, 2019. v. 123, p. 323–346.

NTU. **Anuário NTU**. Brasília, 2019. Acesso em 25 fev. 2020. Disponível em: <<http://www.ntu.org.br/novo/upload/Publicacao/Pub637020043450950070.pdf>>.

React Native. 2020. Acesso em 25 fev. 2020. Disponível em: <<https://reactnative.dev>>.

Ubus. 2020. Acesso em 25 fev. 2020. Disponível em: <<https://www.ubus.com.br/>>.