



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

FGA Pool: Um aplicativo móvel gamificado voltado para o compartilhamento de caronas na FGA

Autor: Vinícius Borges de Almeida
Orientador: Professor Doutor Wander Cleber Pereira

Brasília, DF
2023



Vinícius Borges de Almeida

FGA Pool: Um aplicativo móvel gamificado voltado para o compartilhamento de caronas na FGA

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Professor Doutor Wander Cleber Pereira

Brasília, DF

2023

Vinícius Borges de Almeida

FGA Pool: Um aplicativo móvel gamificado voltado para o compartilhamento de caronas na FGA/ Vinícius Borges de Almeida. – Brasília, DF, 2023-57 p. : il. (algumas color.) ; 30 cm.

Orientador: Professor Doutor Wander Cleber Pereira

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2023.

1. Mobilidade compartilhada. 2. Gamificação. I. Professor Doutor Wander Cleber Pereira. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. FGA Pool: Um aplicativo móvel gamificado voltado para o compartilhamento de caronas na FGA

CDU 02:141:005.6

Vinícius Borges de Almeida

FGA Pool: Um aplicativo móvel gamificado voltado para o compartilhamento de caronas na FGA

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 23 de Julho de 2023:

**Professor Doutor Wander Cleber
Pereira**
Orientador

Prof.^a Dr.^a Elaine Venson
Convidado 1

**Prof. Dr. John Lenon Cardoso
Gardenghi**
Convidado 2

Brasília, DF
2023

Resumo

Diante da crescente populacional no DF e, conseqüentemente, do aumento de veículos, há congestionamentos constantes, falta de lugares para estacionar e aumento da poluição. Para mitigar esses problemas, existem modos de mobilidade compartilhada, nos quais visam uma maior otimização do número de carros e passageiros. Têm maior flexibilidade no horário de partida, custo-benefício e menor quantidade de poluição produzida por pessoa. E, para fornecer uma contribuição para o assunto, este trabalho propõe um aplicativo móvel, visando um maior engajamento entre o público da FGA. A solução será implementada junto às 'técnicas de gamificação'. O objetivo é uma entrega de maior valor ao usuário e ao produto.

Palavras-chaves: Mobilidade Compartilhada. Aplicativo Móvel. Gamificação.

Abstract

In view of the growing population in the Federal District and, consequently, the increase of vehicles, there are heavy traffic, lack of parking spaces and increased pollution. To solve this problem, there are modes of shared mobility, which aims at a greater optimization between the number of cars and passengers. It has greater flexibility in the departure time, cost-benefit and less amount of pollution produced per person. And, to provide a contribution to these issues, this work proposes a mobile application for greater engagement among the FGA audience. The solution will be implemented together with gamification techniques. The goal is to deliver greater value to the user and the product.

Key-words: Shared-mobility. Mobile app. Gamification.

Lista de ilustrações

Figura 1 – Alunos Unb - Gama (Autoria própria)	14
Figura 2 – Roubo a Transeuntes - Gama (Autoria própria)	15
Figura 3 – Shared-mobility Umbrella.(MACHADO et al., 2018)	18
Figura 4 – Ocatalisys Framework.(CHOU, 2020)	23
Figura 5 – Left Brain vs Right Brain. (CHOU, 2020)	24
Figura 6 – White Hat vs. Black Hat. (CHOU, 2020)	25
Figura 7 – Octalisys Dashboard Strategy. (CHOU, 2020)	26
Figura 8 – Resultado Pesquisa(BESSA, 2020)	29
Figura 9 – Características dos Métodos de Priorização(KHAN et al., 2015)	34
Figura 10 – Ciclo Scrum Solo(PAGOTTO et al., 2016)	36
Figura 11 – Telas Protótipo(Autoria Própria)	37
Figura 12 – Telas Protótipo(Autoria Própria)	38
Figura 13 – Telas Protótipo(Autoria Própria)	38
Figura 14 – Telas Protótipo(Autoria Própria)	39
Figura 15 – Estrutura Banco de Dados(Autoria Própria)	40
Figura 16 – Estrutura rides(Autoria Própria)	41
Figura 17 – Estrutura rides(Autoria Própria)	41
Figura 18 – Estrutura ride_request(Autoria Própria)	42
Figura 19 – Estrutura resquest_response(Autoria Própria)	42
Figura 20 – Arquitetura MVC(VASILYIY, 2015)	43
Figura 21 – Organização dos Arquivos(Autoria Própria)	43
Figura 22 – Telas Aplicativo(Autoria Própria)	46
Figura 23 – Telas Aplicativo(Autoria Própria)	47
Figura 24 – Telas Aplicativo(Autoria Própria)	48
Figura 25 – Telas Aplicativo(Autoria Própria)	49
Figura 26 – Telas Aplicativo(Autoria Própria)	50
Figura 27 – Telas Aplicativo(Autoria Própria)	51
Figura 28 – Telas Aplicativo(Autoria Própria)	52

Lista de tabelas

Tabela 1 – Requisitos não Funcionais (Autoria própria)	31
Tabela 2 – Requisitos Funcionais Adaptado de (BESSA, 2020)	33
Tabela 3 – Priorização dos Requisitos Funcionais (Autoria própria)	35

Lista de abreviaturas e siglas

FGA	Faculdade do Gama - UnB
GPS	Global Positioning System
DF	Distrito Federal
UID	Identificador Único do Usuário

Sumário

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.2	Justificativa	14
1.3	Problema de Pesquisa	15
1.4	Objetivos	15
1.4.1	Objetivo geral	16
1.4.2	Objetivo específico	16
1.5	Organização do Trabalho	16
2	REFERENCIAL TEÓRICO	17
2.1	Shared-mobility	17
2.2	Shared-mobility e Suas Modalidades	17
2.2.1	Car-sharing	19
2.2.1.1	Station-based Round-trip	19
2.2.1.2	Station-based One-Way	19
2.2.1.3	Free-Floating One Way	19
2.2.2	Personal Vehicle Sharing	19
2.2.2.1	Peer-to-Peer	20
2.2.2.2	Fractional Ownership	20
2.2.3	Bike-sharing	20
2.2.4	Ride-sharing	20
2.2.4.1	Car-pooling	21
2.2.4.2	Van-pooling	21
2.2.5	On-demand Ride Services	21
2.2.5.1	Ride-sourcing	22
2.2.5.2	Ride-splitting	22
2.3	Gamificação	22
2.3.1	Framework de Gamificação	22
2.3.1.1	Octalysis Framework	22
2.3.1.2	Motivações Lado Esquerdo Cerebral (Tendência Extrínseca) vs. Lado Direito Cerebral (Tendência Intrínseca)	24
2.3.1.3	White Hat vs. Black Hat	25
2.3.1.4	Octalysis Strategy DashBoard	26
2.4	Aplicativos Existentes	26
3	METODOLOGIA	29

3.1	Escopo e Definições do Aplicativo	29
3.1.1	Modalidade do Compartilhamento de Mobilidade	29
3.1.2	Elementos de Gamificação	30
3.1.3	Requisitos do Sistema	31
3.1.3.1	Requisitos Não Funcionais	31
3.1.3.2	Requisitos Funcionais	31
3.1.3.3	Priorização de Requisitos	33
3.2	Desenvolvimento do Aplicativo	35
3.2.1	Método de Desenvolvimento do Aplicativo	36
3.2.1.1	Scrum Solo	36
3.2.1.2	Kanban	36
3.2.2	Prototipação	37
3.2.3	Tecnologias	39
3.2.3.1	Sistema operacional	39
3.2.3.2	Ambiente de Desenvolvimento Integrado(IDE) e Linguagem	39
3.2.3.3	Serviços em Nuvem	40
3.2.3.4	Versionamento de Código e Repositório	40
3.2.4	Banco de Dados	40
3.2.5	Arquitetura	42
3.2.6	Regras de Gamificação	44
3.2.6.1	Avaliação	44
3.2.6.2	Distribuição de pontos	44
3.2.6.3	Níveis	44
3.2.6.4	LeaderBoard	44
4	FGA POOL	45
4.1	Login	46
4.2	Cadastro	47
4.3	Cadastro Carona	48
4.4	Solicitar Carona	49
4.5	Iniciar Corrida	50
4.6	Outras Funcionalidades	51
4.7	Perfil	52
5	CONSIDERAÇÕES FINAIS	53
5.1	Trabalhos e Aplicações Futuras	53
	REFERÊNCIAS	55

1 Introdução

Neste capítulo, é apresentado o trabalho de conclusão de curso e a contextualização acerca dos assuntos que irão ser abordados ao decorrer do trabalho: Seção 1.1, contextualização; seção 1.2, justificativa; seção 1.3, problemas de pesquisa; seção 1.4, objetivos; seção 1.4.1, objetivos gerais; seção 1.4.2, objetivos específicos; e, por fim, a organização do trabalho na seção 1.5.

1.1 Contextualização

Devido ao crescimento populacional e econômico, a infraestrutura de grandes centros urbanos não conseguiu acompanhá-los. Como consequência, geram-se mais engarrafamentos, principalmente em horários de grande movimentação de veículos, ocasionando gastos excessivos de combustível e perda de tempo no trânsito (LIU; ZHANG; YANG, 2019). Este fato tem impactos financeiros e na saúde da população em geral devido a grande emissão de poluentes.

Brasília, a capital do país, não ficou de fora dessa crescente populacional. A cidade tinha sido planejada para 500 mil habitantes e, em 2019, contou com uma estimativa de 3.015.268 habitantes e 1.812.473 veículos (IBGE, 2019).

Tendo em vista que muitos veículos não estão sendo utilizados em sua capacidade de passageiros máxima, uma possibilidade para mitigar esse problema seria a utilização dos serviços de shared-mobility (mobilidade compartilhada). Estes se baseiam na utilização de um mesmo veículo (qualquer tipo) por várias pessoas para chegar a um destino comum. Pode ser considerada uma abordagem promissora para redução da utilização de carros em uma cidade, tendo benefícios tanto para o motorista, por conta da redução de consumo de gasolina, quanto para a cidade, com a redução de poluição e tráfego (CICI et al., 2014).

Dentro da mobilidade compartilhada, existem ramificações, como o carpooling, sharing táxis e também serviços pagos como o Uber, 99, Lyft. Denominados 'real time', 'on demand' ou 'peer to peer ride-sharing', esses serviços possuem características próprias e aplicações a diferentes grupos de usuários.

A partir disso e tendo em foco os discentes e docentes da Faculdade UnB Gama, como ter um engajamento da comunidade com um possível aplicativo de shared-mobility?

Com a evolução e crescimento de aparelhos móveis, é necessário considerar a comunicação entre pessoa e aparelho eletrônico (TóTH; TóVöLGYI, 2016), podendo trazer o tópico de gamificação para conseguir uma boa experiência e resultados que sejam significativos aos usuários (TóTH; TóVöLGYI, 2016). Dessa forma, as pessoas participam de

inúmeras interações online, sejam elas em fóruns, redes sociais, aplicativos de mensagem instantânea. Podem haver várias motivações para as interações, entre elas: conhecer novas pessoas, se divertir ou se atualizar na sua área profissional.

Com diferentes utilizações no mundo, os jogos, sejam de tabuleiro ou digital, passaram por anos de aperfeiçoamento nos quesitos de motivação e engajamento do seu público. Utilizando vários componentes, como significância, habilidades e recompensas, algumas dessas técnicas, quando utilizadas em um contexto fora de jogos, funcionam como catalisadores para tornar algo mais atraente, influenciando o comportamento do usuário e métodos de interação social (TóTH; TóVöLGYI, 2016).

1.2 Justificativa

O crescimento populacional do DF conta com uma taxa maior que a média nacional entre os anos de 2018 e 2019 (IBGE, 2019). Devido a este crescimento, também há o aumento de veículos particulares em todas as áreas do DF. Segundo o boletim da Universidade de Brasília (UnB) em números, sendo o mais recente publicado relativo ao ano de 2016, a entrada de estudantes comparada ao ano de 2009 é aproximadamente o dobro por semestre. Em consequência, a quantidade de alunos regulares no Campus UnB - Gama no ano de 2016 é mais que o triplo ao ano de 2009, totalizando 2293 alunos (DPO, 2019).

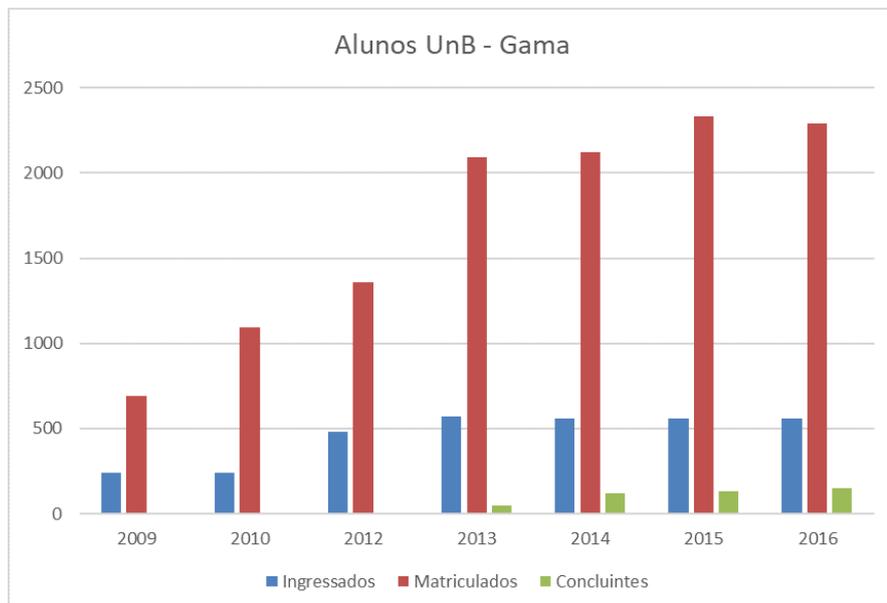


Figura 1 – Alunos Unb - Gama (Autoria própria)

O gráfico a seguir mostra análises feitas sobre roubo de pedestres na Região Administrativa II - Gama, dados divulgados pela Secretaria de Estado e Segurança Pública. Os dados foram coletados entre os anos de 2014 a 2019, de Janeiro a Dezembro de cada ano (SSP/DF, 2019).

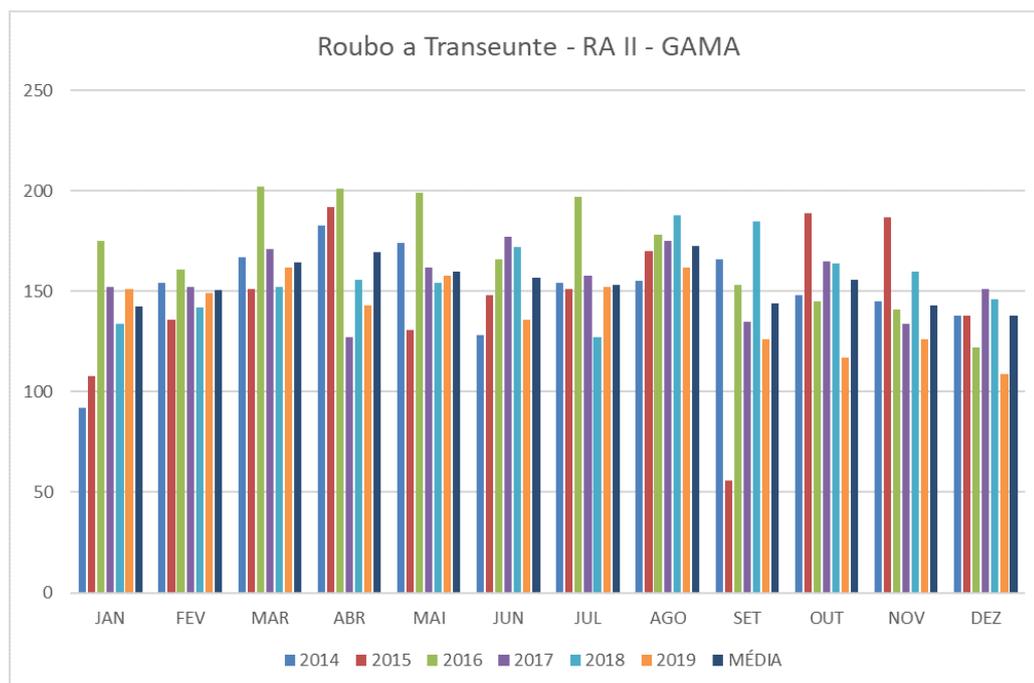


Figura 2 – Roubo a Transeuntes - Gama (Autoria própria)

Foi verificado um aumento da média de roubos nos meses de período letivo da UnB - faculdade do Gama. Além disso, houveram inúmeros depoimentos de roubo nas regiões adjacentes à faculdade, relatados pelos estudantes em redes sociais.

Portanto, o aumento populacional crescente, o aumento de veículos particulares, os problemas relacionados ao transporte público, a quantidade crescente de alunos regulares e a falta de segurança nas áreas ao redor do campus evidenciam a necessidade de uma política de transporte alternativo à comunidade acadêmica.

1.3 Problema de Pesquisa

Dado o contexto acima, o presente trabalho tem como problema principal a seguinte questão: Como desenvolver um aplicativo móvel gamificado de shared-mobility para a comunidade da Faculdade do Gama como alternativa ao transporte público?

1.4 Objetivos

Este trabalho possui um objetivo geral e, a partir dele, foram definidos objetivos específicos para que possa ser alcançado o que foi proposto para mitigar a problemática.

1.4.1 Objetivo geral

Desenvolver um aplicativo móvel gamificado na plataforma Android para o compartilhamento de mobilidade para a comunidade acadêmica da UnB - campus Gama.

1.4.2 Objetivo específico

- Definir uma modalidade de shared-mobility que se acomode melhor à comunidade;
- Definir técnicas de gamificação a serem utilizados pelo aplicativo;
- Extrair requisitos do aplicativo junto à comunidade;
- Desenvolver o aplicativo abordando os tópicos acima a partir de uma metodologia.

1.5 Organização do Trabalho

Os tópicos deste trabalho estão organizados da seguinte forma:

- Capítulo 2: Referencial Teórico
- Capítulo 3: Metodologia
- Capítulo 4: FGA Pool
- Capítulo 5: Considerações Finais

2 Referencial Teórico

Neste capítulo, será apresentada a pesquisa sobre os fundamentos abordados para o andamento deste trabalho. Na seção 2.1, *shared-mobility*, serão apresentados seus conceitos e tópicos principais. Na seção 2.2, tipos de *shared-mobility*. Na seção 2.3, serão abordados conceitos gerais de gamificação. Na seção 2.4, aplicativos existentes.

2.1 Shared-mobility

A partir do século 21, houve uma grande preocupação com o meio ambiente e com os problemas derivados pelo uso indiscriminado de recursos naturais (MACHADO *et al.*, 2018). Com isso, surgiram serviços alternativos focados nessa problemática, como a mentalidade de aluguéis em pequeno prazo, sejam eles de moradia, veículos, espaço de trabalho, entre vários outros. Nesse contexto, surge o conceito de economia compartilhada como um novo paradigma que provém acessibilidade a serviços e produtos em detrimento da propriedade individual (MACHADO *et al.*, 2018).

Com isso, diversas soluções se popularizaram, como aplicativos de aluguéis de imóveis por temporada (Airbnb, Hoteis.com, Booking.com, Trivago) e de transporte (Uber, 99, Cabify). Mobilidade compartilhada faz parte desse segmento de economia compartilhada e tem um grande potencial em áreas urbanas quando integrados aos sistemas de transporte públicos existentes (MACHADO *et al.*, 2018).

Mobilidade compartilhada tem o potencial de reduzir congestionamentos de carros e a procura por estacionamento, dando início a uma possível redução de veículos (MACHADO *et al.*, 2018) e uma otimização pela quantidade de vagas disponível em um determinado tipo de veículo. Dentro desse leque de mobilidade compartilhada, existem modos de passeio em diferentes tipos de veículos. A escolha desse modo pelo público-alvo é crucial para que haja resultados esperados na solução proposta deste trabalho.

2.2 Shared-mobility e Suas Modalidades

O mercado de mobilidade pessoal está mudando rapidamente por conta de modificações estruturais nos segmentos social e cultural, assim como avanços tecnológicos tal como smartphones, Internet das coisas (IoT), Big Data, computação em nuvem, processamento de dados e a ampliação de conectividade de dados.

Mobilidade compartilhada é um conceito inovador de transporte, no qual pode ser considerado como um leque de serviços de mobilidade que permite o planejamento e

reserva, informação em tempo real e pagamento justo pelo serviço utilizado em uma única interface de usuário (MACHADO et al., 2018).

Em termos de mobilidade urbana, a mudança de ter um veículo para uso de serviços de mobilidade, muitas vezes promovido em conceitos de sustentabilidade, é acessível na maioria das cidades ao redor do mundo pelos sistemas de mobilidade compartilhada (MACHADO et al., 2018). Uma pesquisa feita nos países baixos com usuários de car-sharing apresentou 30% a menos de proprietários de veículos entre os que praticam essa modalidade e 15% a 20% de milhas economizadas (SANTOS, 2018). A pesquisa também apresentou que, graças à redução de proprietários de veículos particulares, os que adotaram essa modalidade por pessoa emitiram menos gás carbônico por ano. Ou seja, adotando tais modalidades que englobam o compartilhamento é perceptível a redução de custos anuais e impactos relacionados ao meio ambiente. Essas modalidades fazem o uso de um veículo (carro, bicicleta, van, lambreta, etc.) de uma forma inovadora em questão de estratégia, permitindo aos usuários o acesso a estes transportes por um curto prazo.

Em uma visão macro, mobilidade compartilhada pode ser definida como uma alternativa de transporte que visa maximizar o uso de recursos de mobilidade que a sociedade consegue arcar, priorizando sua utilização em detrimento de possuir um veículo. O objetivo é fornecer uma vasta gama de opções de mobilidade, assim, pode-se aumentar a multimodalidade e reduzir custos com transporte (MACHADO et al., 2018).

De acordo com (SHAHEEN et al., 2017), mobilidade compartilhada é um 'guarda-chuva', termo que se refere à variedade de formas inovadoras de modos de transporte com finalidades diversas, modelos de negócios e impactos sobre comportamentos de viagem. Na figura abaixo, é mostrado o esquema das modalidades existentes de mobilidade compartilhada:

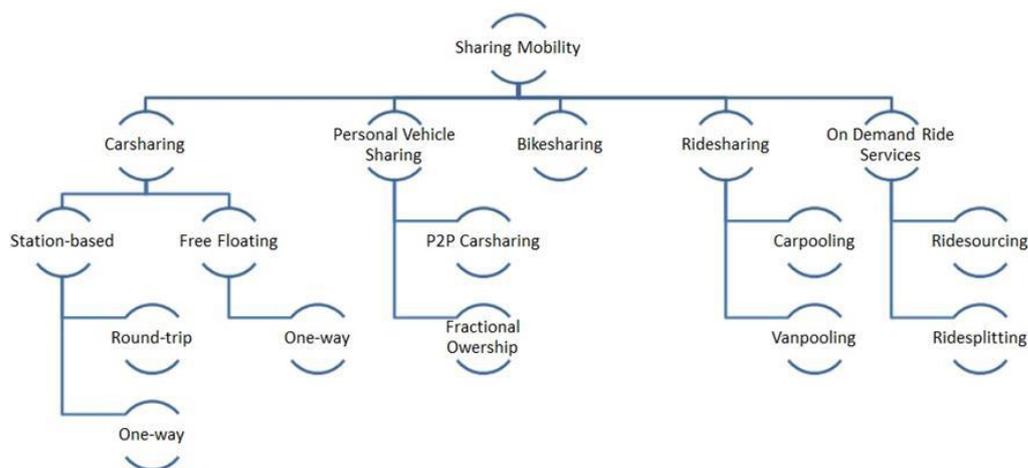


Figura 3 – Shared-mobility Umbrella.(MACHADO et al., 2018)

2.2.1 Car-sharing

É um modo de transporte no qual várias pessoas usam o mesmo veículo. Os utilizadores se beneficiam do veículo sem ter as responsabilidades de possuir um e de arcar com seus custos. Alguns benefícios desse modo são: a redução da frequência de utilização de carro, redução de milhas/quilômetros andados, cancelamento ou atraso na apropriação de um carro (NAMAZU et al., 2018). Existem duas categorias de operação dentro de carsharing que dependem do tipo, função e público-alvo, como demonstrado na figura 3, são eles;

2.2.1.1 Station-based Round-trip

É um serviço de duas vias, onde os clientes pegam o veículo em uma estação designada e o retornam para o mesmo lugar após sua utilização. O custo é de utilização por hora, podendo incluir quilômetros grátis ou custo por quilômetro andado.

2.2.1.2 Station-based One-Way

É similar ao round-trip, mas o cliente não precisa necessariamente voltar para a mesma estação, ele pode deixar em qualquer uma das estações espalhadas pela cidade. É um modo mais flexível e adaptável às necessidades do usuário. Normalmente, é cobrado por minuto, as distâncias são menores e não requer reserva. Porém, este modo tem um problema de proporção de carros disponíveis nas estações, o que pode acarretar em uma frota maior do que se necessita e carros parados sem utilização.

2.2.1.3 Free-Floating One Way

Este modo disponibiliza carros para serem pegos e deixados em qualquer área dentro do perímetro designado. Sem estações fixas, o usuário pode sair desse perímetro quando estiver dirigindo, porém, o carro deve ser retornado para a área do serviço quando for finalizar a sua utilização. Este serviço pode ser comparado aos de patinetes motorizados disponibilizados recentemente em Brasília, onde o veículo é encontrado através de GPS (Global Positioning System) por um aplicativo e é ativado através de smartphones.

2.2.2 Personal Vehicle Sharing

Compartilhamento de veículo pessoal é um sistema onde os proprietários desses veículos os alugam para outras pessoas por um período de tempo. Geralmente, companhias desse tipo de serviço intermediam a transação entre proprietário e cliente, fornecendo os recursos necessários para ambas as partes para que ocorra a transação. Assim, providenciam suporte ao consumidor de forma segura e tecnológica (SHAHEEN; MALLERY; KINGSLEY, 2012).

2.2.2.1 Peer-to-Peer

Este modo emprega carros particulares que são disponibilizados para o uso compartilhado por um indivíduo ou membros da companhia. A empresa intermedia o procedimento de aluguel, atribui cronogramas para a disponibilidade dos veículos, podendo confirmar ou recusar propostas. O local de pegar e deixar o veículo é acordado entre o proprietário do veículo e o locatário. Gastos com seguro são cobertos pela companhia na hora da locação. Para que exista manutenção desse serviço, a companhia paga uma taxa do valor da locação. Esta modalidade promove o compartilhamento do que já tem dono e é pouco utilizada comparada às companhias que possuem frotas próprias (MACHADO et al., 2018).

2.2.2.2 Fractional Ownership

É um modelo de compartilhamento da propriedade de um veículo entre um grupo pequeno de pessoas, no qual cada indivíduo desse grupo arca com parte dos custos para que tenha acesso a este serviço. Normalmente, este serviço é firmado por contrato, que pode variar de indivíduo para indivíduo dependendo de suas necessidades. Assim, podem acessar certos tipos de veículos que normalmente não conseguiriam comprar. É um modo menos flexível comparado aos de car-sharing por conta da utilização ser feita por acordo entre os usuários.

2.2.3 Bike-sharing

O compartilhamento de bicicletas é similar ao de car-sharing, onde usuários podem acessar esse serviço quando precisam. As bicicletas ficam em estações, normalmente não vigiadas, e o serviço é igual aos de car-sharing: station-based one-way (as bicicletas podem ser devolvidas em qualquer estação), station-based round-trip (devem ser retornadas para a mesma estação), free-floating one-way (podem ser deixadas dentro do perímetro do serviço). No último, pode ser acessado por aplicativos móveis juntamente com GPS.

Estudos mostram que os benefícios desse sistema são numerosos, como por exemplo: evitar estacionamentos, manutenções, conexão conveniente com o transporte público, redução de tempo entre viagens e custos em centros urbanos, contribuição para a saúde e abertura de portas para mais interação social e lazer (MACHADO et al., 2018).

2.2.4 Ride-sharing

Compartilhamento de passeio é um modo que reúne vários passageiros em um mesmo veículo, sendo que estes possuem destino ou caminho similar e com horário de partida também similar, geralmente um carro ou uma van. Tornou-se uma estratégia poderosa para redução de congestionamentos ao ver que este modo tem o potencial de

diminuir a quantidade de veículos nas estradas por encorajar o compartilhamento do mesmo veículo (MACHADO et al., 2018).

Nos anos recentes, a facilidade em conseguir o compartilhamento de passeio trazido por tecnologias de comunicação, tal como a internet, aplicações móveis e veículos conectados, possibilitou às pessoas o gerenciamento do uso dos seus carros de melhor forma, porque motoristas compartilham em tempo real informações de tráfego. Essa nova geração de plataformas de caronas teve um aumento massivo de motoristas e passageiros. De acordo com (MACHADO et al., 2018), ride-sharing é uma nova rede de serviço de transporte ponto a ponto baseada em uma plataforma dinâmica, na qual motoristas oferecem seus veículos para transportar passageiros que tenham o mesmo destino ou caminho similar.

2.2.4.1 Car-pooling

Car-pooling é um modo informal de compartilhamento de carona, no qual atende pelo menos três passageiros que contribuem de uma quantia para ajudar os custos do motorista. O motorista deixa disponível seu veículo para passageiros que têm seus destinos na mesma rota. Este modo é geralmente utilizado por pessoas que têm uma rotina similar, seja ir para escola, universidade ou trabalho, e é praticado em sua maioria informalmente, sem que tenha uma plataforma centralizada (MACHADO et al., 2018). Existem também incentivos praticados por governos e negócios locais, como estacionamentos, espaço dedicado para pegar e deixar passageiros, descontos e menor tempo de espera em pedágios.

2.2.4.2 Van-pooling

Tipicamente, é composto de sete a quinze pessoas, sendo utilizado regularmente uma van ou veículo de mesmas proporções. Os custos de operação são divididos pelos passageiros e podem compartilhar a responsabilidade de dirigir também. Van-pools geralmente têm um coordenador e um coordenador alternativo (SHAHEEN; COHEN; ZOHDY, 2016). Os seus benefícios são confiabilidade, interação social, redução de estresse, redução de custos, menor agressão ao meio ambiente e mais estacionamentos.

2.2.5 On-demand Ride Services

Serviços de passeio sob demanda são formas inovadoras para pessoas terem acesso a opções de transporte prontamente e quando solicitado através de seus smartphones. Tem como foco principal a minimização do tempo de transporte e custo (MACHADO et al., 2018). O maior motivo de sua popularidade é a conveniência de ser pego e deixado onde quiser, sendo de fácil agendamento, informações automáticas de destino, pagamento online e custo benefício (GUPTA; BURIRO; CRISPO, 2019).

2.2.5.1 Ride-sourcing

É uma solução efetiva de sistemas interativos de compartilhamento de viagem através de serviços de passeio pagos, onde passageiros usam veículos particulares para passeios sob demanda. Nesse sistema, o serviço cobra custo de combustível, depreciação do automóvel, a comissão do motorista, taxa da plataforma e taxas estaduais para a regularização do serviço. É oferecida uma frota de veículos particulares para o serviço de transporte nessas plataformas, que é ininterrupto, personalizado e altamente flexível.

2.2.5.2 Ride-splitting

É uma variação do modelo de ride-sourcing, onde passageiros com rotas similares ou sobrepostas dividem o transporte e os custos. Este modo pode facilitar o serviço com veículos que tenham mais assentos e redução de custos. Muitas companhias de serviço on-demand fornecem ride-sourcing e ride-splitting como opção de transporte. São utilizados pelas plataformas mais populares atualmente, sendo elas: uber, 99, cabify, entre outras.

2.3 Gamificação

Quando o termo gamificação foi introduzido no começo dos anos 2000, não houve uma grande repercussão por ser uma diferente figura na indústria, pelo contrário, conferências colocaram-no no mapa para uma audiência maior na segunda metade de 2010 (GROH, 2012). De acordo com (DETERDING et al., 2011), gamificação é definida como o uso de elementos de design de jogos em contextos diferentes de jogos. Já para (HUOTARI; HAMARI, 2012), gamificação se refere ao processo de melhoria de um serviço com acessibilidade para experiências divertidas a fim de apoiar o valor de criação geral de um usuário. Para (ZICHERMANN; CUNNINGHAM, 2011) é usar pensamento e mecânica de jogos para resolver problemas e engajar um público.

Mesmo com todas essas definições de diferentes autores, pode-se perceber algo em comum, que é o processo de transformação de um contexto, utilizando características de jogos que agreguem valor tanto ao usuário quanto ao produto.

2.3.1 Framework de Gamificação

2.3.1.1 Octalysis Framework

É um framework completo para analisar e construir estratégias em volta dos vários sistemas que fazem com que os jogos sejam tão atraentes (CHOU, 2015).

Na maioria dos jogos de sucesso, eles tendem a ter certos core drives (unidades motivacionais), que fazem com que o usuário caminhe para uma variedade de decisões e atividades. Diferentes técnicas tendem a guiar os usuários diferentemente; alguns por

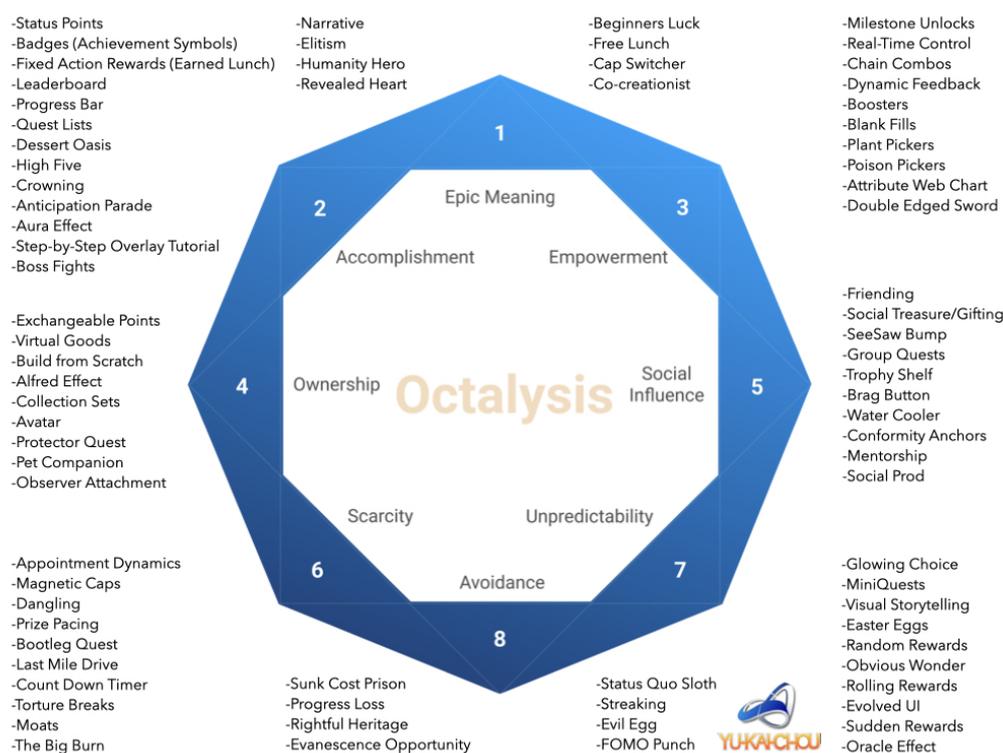


Figura 4 – Ocatalisys Framework.(CHOU, 2020)

inspiração e empoderamento, outros por manipulação e obsessão (CHOU, 2015). O resultado desses core drives deram origem ao framework, que é representado por um octógono e possui em cada um dos seus lados uma motivação, totalizando oito core drives.

- **Core Drive 1 - Significado Épico e Chamado:** Quando uma pessoa acredita que ela está fazendo algo maior que ela e/ou que ela foi a escolhida para realizar aquela ação.
- **Core Drive 2 - Desenvolvimento e Realização:** É a motivação de estar fazendo progresso, desenvolvendo habilidades, alcançando maestria e, eventualmente, superando desafios. O desafio em si nessa questão é bastante importante, pois badge (distintivo) ou troféu sem um desafio não tem significado algum (CHOU, 2015). Esta unidade apresenta a maioria dos PBLs (pontos, badges, placares).
- **Core Drive 3 - Empoderamento de Criatividade e Feedback:** É expressado quando usuários estão ocupados em um processo de criação, onde estão sempre descobrindo coisas novas e tentando novas combinações. Não só se expressam, mas também querem um feedback da sua criatividade.
- **Core Drive 4 - Propriedade e Posse:** É quando usuários estão motivados por sentirem que possuem ou controlam algo. Há um maior desejo de acumular riquezas. Este core lida com a maioria dos itens virtuais ou moedas virtuais dentro do sistema.

- **Core Drive 5 - Influência social e Pertencimento:** Este core incorpora todos os elementos sociais que motivam as pessoas, incluindo: mentoria, aceitação social, feedback social, companheirismo, competição e inveja.
- **Core Drive 6 - Escassez e Impaciência:** É a motivação de querer algo simplesmente porque é extremamente raro, exclusivo ou imediatamente inatingível.
- **Core Drive 7 - Imprevisibilidade e Curiosidade:** É a motivação de estar constantemente engajado, pois não se sabe o que pode acontecer.
- **Core Drive 8 - Perda e Rejeição:** É a motivação de evitar que alguma coisa negativa aconteça.

2.3.1.2 Motivações Lado Esquerdo Cerebral (Tendência Extrínseca) vs. Lado Direito Cerebral (Tendência Intrínseca)

O porquê de tudo que o usuário faz está baseado em uma ou mais unidades motivacionais dessas oito. Quando não tem nenhuma dessas unidades por trás de uma ação desejada, significa que não existe motivação e nenhuma ação é tomada (CHOU, 2015). Ou seja, cada uma dessas unidades motivacionais possuem uma natureza por trás e o posicionamento delas no framework determina a natureza dessa motivação.

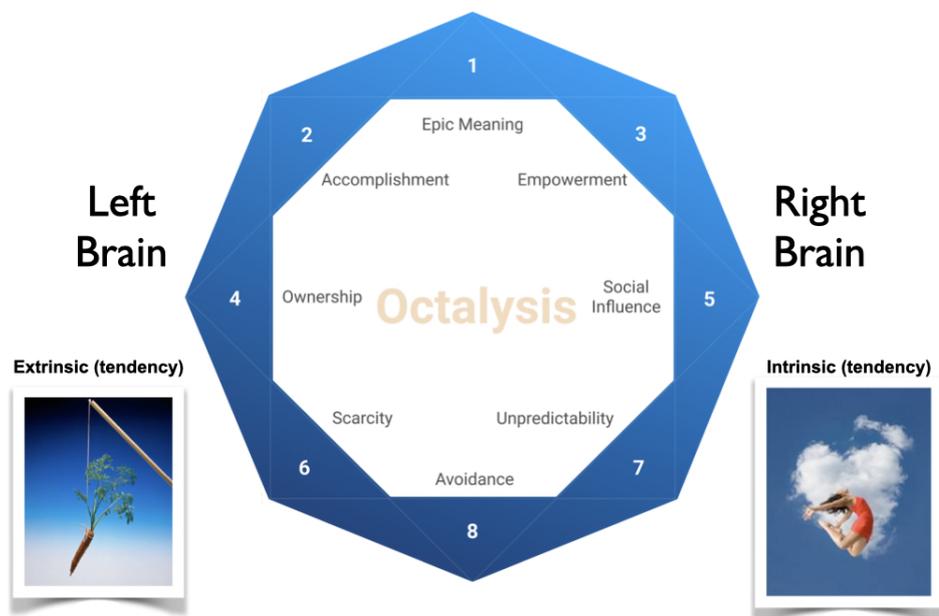


Figura 5 – Left Brain vs Right Brain. (CHOU, 2020)

O framework é organizado para que as unidades motivacionais que focam na criatividade, auto expressão e dinâmicas sociais fiquem do lado direito do octógono, parte

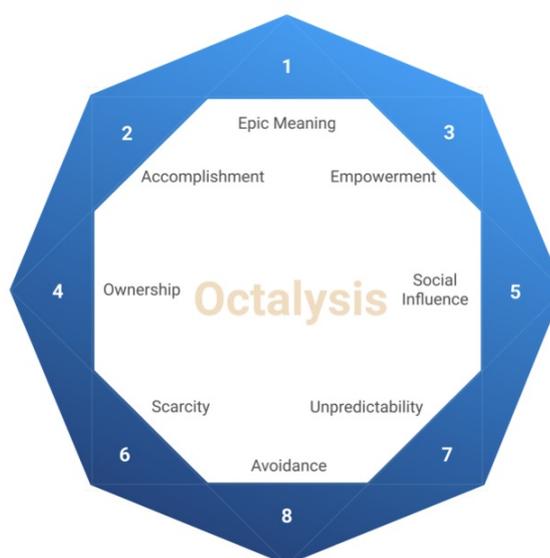
que é chamada de lado direito cerebral. E as unidades ligadas à lógica, raciocínio analítico e propriedade estão dispostas do lado esquerdo do octógono, denominadas unidades motivacionais do lado esquerdo cerebral.

Interessantemente, unidades motivacionais do lado esquerdo contam com motivação extrínseca, isto é, o usuário é motivado porque ele quer obter algo, um objetivo, um item ou algo que o usuário não pode obter. Em contrapartida, o lado direito está majoritariamente associado a motivações intrínsecas, o que significa que o usuário não precisa de um objetivo ou recompensa para usar sua criatividade, passar o tempo com os amigos ou sentir o suspense da imprevisibilidade, sendo a atividade por si só recompensadora (CHOU, 2015).

2.3.1.3 White Hat vs. Black Hat

Mais um fator a ser observado no framework é que as unidades motivacionais superiores são consideradas muito positivas enquanto as inferiores são mais negativas. Ou seja, técnicas que utilizam mais as unidades motivacionais superiores são chamadas de white hat gamification enquanto as que utilizam as de baixo são chamadas de black hat gamification.

White Hat Gamification



Black Hat Gamification

Figura 6 – White Hat vs. Black Hat. (CHOU, 2020)

Segundo (CHOU, 2015), se algo é engajador, permite que o usuário expresse sua criatividade, sinta-se sucedido através da maestria de habilidade e dá um senso maior de significado, ou seja, faz com que ele se sinta bem e poderoso. Em contrapartida, se ele faz algo que nunca sabe o que pode acontecer em seguida, está sempre com medo de perder algo ou pode se perguntar o porquê de estar se esforçando para conseguir algo que ele não

pode conseguir. A experiência muitas vezes vai deixá-lo com um sentimento ruim. Mesmo que esses motivadores estejam na parte inferior do octógono, não significa necessariamente que eles sejam ruins. Assim como outros motivadores, eles também podem ser utilizados para bons resultados (CHOU, 2015).

2.3.1.4 Octalysis Strategy DashBoard

Este tópico aborda brevemente como implementar o octalysis framework do início do projeto. No início de cada projeto, as primeiras coisas a serem definidas são: métricas de negócio, que guiam para o objetivo do jogo; usuários, que guiam para os jogadores; ações desejadas, que guiam para etapas vencedoras; mecânicas de feedback, que guiam para gatilhos; e incentivos, que guiam para recompensas (CHOU, 2015).

Octalysis Strategy Dashboard

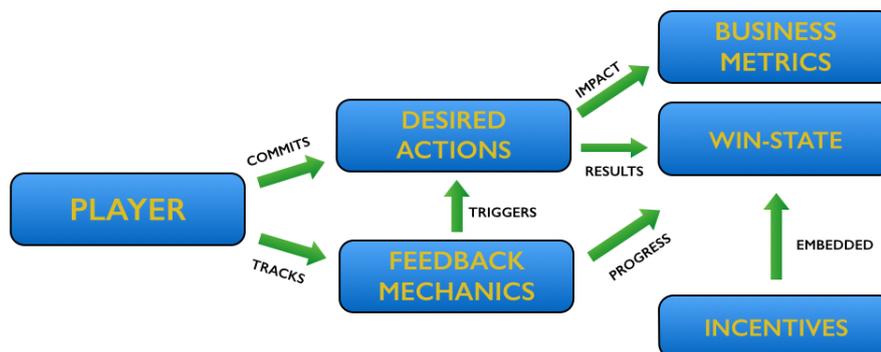


Figura 7 – Octalysis Dashboard Strategy. (CHOU, 2020)

O dashboard não foi feito para ser tão compreensivo ou estático quanto um plano de negócio, mas ele deve fornecer o mínimo de informações críticas que ajude a executar o projeto de gamificação (CHOU, 2015).

2.4 Aplicativos Existentes

Atualmente, há aplicativos semelhantes ao proposto por este trabalho. Neste tópico, serão abordados os principais aplicativos existentes relacionados à mobilidade compartilhada, assim como suas funcionalidades básicas padrão.

As aplicações de compartilhamento de mobilidade estão se popularizando cada dia mais, seja para uso diário, para eventos ou até mesmo viagens interestaduais. Os principais utilizados pelo Brasil são o Uber, BlaBlaCar e WazeCarpool.

Pela Uber, existiu uma modalidade de seus serviços para o compartilhamento de uma corrida por demanda de passageiros desconhecidos. Conhecido como UberPool e, posteriormente, chamado de Uber Juntos, ele funciona da seguinte forma: quando solicitado, verificava passageiros perto um do outro com rotas parecidas e apontava um local de encontro que não devia ser mais que poucos minutos a pé. Quando em corrida, podia-se ver em tempo real o deslocamento, preço, informações do motorista, passageiros e método de pagamento. Após o uso, ambos, motorista e passageiro, se avaliavam. É uma alternativa que pode aproveitar uma viagem mais barata que as outras modalidades, tornando-a eficiente e acessível. Devido a pandemia de 2020 do coronavírus, para ajudar a desacelerar o contágio do vírus, a Uber decidiu descontinuar esse serviço no Brasil e até então não voltou.

O WazeCarpool é um aplicativo de caronas compartilhadas da Waze e, diferente do UberPool, ele não é por demanda. Os motoristas ofertavam a carona, podendo ser agendadas durante toda a semana, com seus horários, pontos de encontro, rotas e seus preços. Após solicitada a carona, o passageiro entrava em contato para tirar dúvidas relativas ao ponto de encontro e de descida. Após solicitação aceita, o motorista ia de encontro com o passageiro e o deixava no local acordado entre as partes. O pagamento era feito via aplicativo e repassado pelo GooglePay para o motorista. No final, ambos se avaliavam. Não era permitido mais de duas caronas diárias. Também por conta da pandemia e da falta de demanda, foi descontinuado em 2022.

O BlaBlaCar se diferencia dos demais por ser um aplicativo de caronas interestaduais. Funcionando como o WazeCarpool, é oferecido a carona com seu preço fixo pela distância, tempo e outros passageiros. Também possui informações de rota e ponto de encontro. O passageiro solicita a carona e, quando aceito, o passageiro vai de encontro com o motorista para continuar a viagem. Após o pagamento, ambos se avaliam. O pagamento é feito por dinheiro ou PIX.

Como visto nos aplicativos existentes, eles possuem basicamente a mesma estrutura: oferta de carona, informações sobre o motorista, pagamento e avaliação.

3 Metodologia

A metodologia do presente trabalho se dividiu em duas partes: a primeira, relacionada com o escopo do trabalho, a definição de modalidade de compartilhamento de mobilidade e elementos de gamificação e requisitos do sistema; a segunda, em relação às fases do desenvolvimento da aplicação e suas tecnologias.

3.1 Escopo e Definições do Aplicativo

Neste tópico, é abordado o escopo e as definições do aplicativo. Na seção 3.1.1, Modalidade do Compartilhamento de Mobilidade; na seção 3.1.2, Elementos de Gamificação; e, por último, na seção 3.1.3, Requisitos do Sistema.

3.1.1 Modalidade do Compartilhamento de Mobilidade

Como pode ser observado na imagem 8, são respostas referentes ao tipo de transporte utilizado para a locomoção dos alunos, sendo os mais relevantes: carro com 41,9%, transporte público com 73,1% e carona com 22,6%. O aluno poderia marcar mais de uma opção.

Diante disso, pode-se escolher a modalidade do tipo de mobilidade compartilhada para atender adequadamente às necessidades dos alunos. Como visto anteriormente, car-pooling é a modalidade de compartilhamento de carona na qual atende até 4 passageiros, que contribuem com uma quantia para ajudar o motorista nos custos e que seus destinos estejam na mesma rota. É utilizado para pessoas que possuem a mesma rotina, seja trabalho, faculdade ou escola. A partir disso, a modalidade escolhida para o avanço do projeto foi o car-pooling.

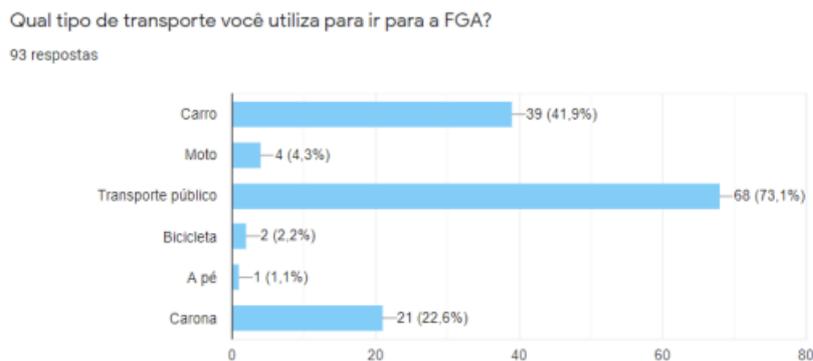


Figura 8 – Resultado Pesquisa (BESSA, 2020)

3.1.2 Elementos de Gamificação

A partir do comportamento direcionado para a aplicação, pode-se escolher core drives e, a partir deles, listar elementos de jogos para a implementação. Seguindo o pressuposto de colaboração entre alunos e sustentabilidade, como visto na introdução e no referencial teórico, os possíveis core drives podem ser aplicados como "Significado Épico Chamado", "Desenvolvimento Realização", "Empoderamento Feedback" e "Influência Social Pertencimento" (BESSA, 2020). Como são motivadores positivos, pode-se ver a aplicação do "White Hat Gamification", parte superior do Ocatalysis Framework.

A partir dos core drives escolhidos, as seguintes técnicas de jogos podem ser aplicadas:

- Pontos de Status
- Badges - Brasões de determinadas conquistas
- Leaderboard - Lista com os maiores pontuadores
- Barra de Progresso
- Quest List - Listas de desafios
- Narrativa
- Herói da Humanidade
- Free Lunch - Ganhar algo de graça
- Milestone Unlocks - recompensas de progressão
- Feedback
- Fazer Amizade
- Social Treasure - Itens obtidos através de outros players
- Itens Virtuais
- Pontos de Troca
- Coleção de Conjuntos
- Avatar
- Prateleira de Troféus

3.1.3 Requisitos do Sistema

Os requisitos de software são funcionalidades que o sistema deve realizar, ou seja, serviços oferecidos e restrições do sistema. Isto reflete na necessidade das partes interessadas sobre o sistema. (SOMMERVILLE, 2011) Os tópicos seguintes descrevem as funcionalidades do sistema proposto e etapas anteriores para o levantamento dos requisitos.

3.1.3.1 Requisitos Não Funcionais

São requisitos que não interferem diretamente quando o produto está em desenvolvimento e podem estar associados com as propriedades de um sistema. (SOMMERVILLE, 2011) Dito isto, a tabela 1 corresponde aos requisitos não funcionais propostos na aplicação:

Requisito	Descrição
Desempenho	O sistema deve de forma quase instantânea responder às interações do usuário.
Disponibilidade	O sistema deve estar disponível para o usuário vinte quatro horas por sete dias da semana.
Segurança	O sistema deve disponibilizar informações importantes apenas quando o usuário for permitido.
Usabilidade	O sistema deve ser familiar para os usuários de Android, não necessitando de tutoriais para o uso.
Compatibilidade	O sistema deve ser compatível com aparelhos móveis com o S.O Android 7.0 e superior

Tabela 1 – Requisitos não Funcionais (Autoria própria)

3.1.3.2 Requisitos Funcionais

"A elicitação e análise de requisitos podem envolver diversos tipos de pessoas em uma organização. Um stakeholder do sistema é quem tem alguma influência direta ou indireta sobre os requisitos do sistema."(SOMMERVILLE, 2011)

Para elicitar os requisitos do sistema, foi utilizado a técnica de entrevista fechada, feita através de um questionário na plataforma Google Forms por (BESSA, 2020). Assim, foi compartilhado pelas redes sociais dos alunos da FGA, que são os usuários finais, para que pudessem colaborar. Abaixo, corresponde às perguntas do questionário:

- Qual tipo de transporte você utiliza para ir para a FGA?
 - Carro
 - Moto

- Transporte público
 - Bicicleta
 - A pé
 - Carona
- Como motorista, você costuma oferecer caronas? Se sim, por que? Se não, o que te motivaria a começar a oferecer?
 - Como passageiro, você utiliza serviços de caronas? Se sim, por que? Se não, o que te motivaria a começar a utilizar?
 - Um aplicativo ou sistema de caronas dentro da faculdade seria útil? Tornaria a vida mais fácil? Por que?
 - Você conhece algum aplicativo ou sistema de caronas? Ou conhece alguém próximo que utilize?
 - Quais funcionalidades seriam importantes nesse possível aplicativo ou sistema?
 - Como motorista ou passageiro, você se sentiria seguro/confortável em dividir carona com um aluno desconhecido? Em caso de resposta negativa, explique quais medidas poderiam ser feitas para inverter essa situação.

A partir das respostas do questionário, foram retirados possíveis requisitos dos sistema e, posteriormente, comparados aos requisitos retirados de outra técnica, entrevista com grupo focal também realizada por (BESSA, 2020). Comparando os requisitos das duas técnicas, foi constatado que eles continham basicamente os mesmos requisitos, o que deu origem aos requisitos finais elicitados, como pode ser observado na tabela 2. A pesquisa obteve noventa e três respostas dos alunos das cinco engenharias da faculdade.

Categoria	Requisito
Carona [C]	[C01] - Visualizar caronas disponíveis no dia
	[C02] - Indicar e visualizar horário de saída da carona
	[C03] - Indicar e visualizar local de partida
	[C04] - Indicar e visualizar local de destino
	[C05] - Indicar e visualizar trajeto da carona
	[C06] - Escolher quantidade de assentos no carro
	[C07] - Receber lembrete de carona
	[C08] - Indicar e visualizar pagamento voluntário ou compartilhado
	[C09] - Visualizar histórico de caronas
	[C10] - Indicar e visualizar tempo deslocamento
	[C11] - Agendar/programar carona
	[C12] - Adicionar amigos à rede
	[C13] - Mostrar as caronas que o passageiro está vinculado
	[C14] - Pesquisar caronas pelo seu destino
	[C15] - Escolher quilometragem máxima de desvio de rota
Segurança [S]	[S01] - Entrar na aplicação por e-mail
	[S02] - Visualizar informações do motorista e dos passageiros
	[S03] - Verificar se é aluno (Verificação através do email institucional)
	[S04] - Escolher o sexo do motorista/passageiro
	[S05] - Compartilhar localização em tempo real
	[S06] - Enviar informações sobre a carona para uma pessoa de confiança
	[S07] - Visualizar pessoas na mesma carona
Comunicação [CO]	[CO01] - Iniciar uma conversa entre o passageiro e o motorista
	[CO02] - Bloquear usuário
Pagamento [P]	[P01] - Cartão de crédito ou débito
	[P02] - Indicar pagamento em dinheiro ou PIX
Gamificação [G]	[G01] - Avaliar motorista e passageiro
	[G02] - Acumular pontos por carona
	[G03] - Ranking de motoristas que mais ofereceram caronas
	[G04] - Subir de nível de acordo com pontuação

Tabela 2 – Requisitos Funcionais Adaptado de (BESSA, 2020)

3.1.3.3 Priorização de Requisitos

Como todos os requisitos não podem ser desenvolvidos em um único incremento, seja por limitação de recursos em termo de tempo, orçamento ou outros recursos que estão inclusos durante o desenvolvimento de um software, é preciso a priorização dos requisitos do software para que consiga a satisfação do usuário final (KHAN et al., 2015).

Um estudo foi realizado por (KHAN et al., 2015) para comparar métodos de priorização de requisitos, dentre eles: Hundred Dollar, Analytic Hierarchy Process (ANP), Hierarchy AHP, Minimal Spanning Tree, Planning Game, Numerical assignment, MoS-coW, Simple Ranking, Bubble Sort e Binary Search Tree. É um projeto pequeno e com

escopo bem definido. O autor faz parte dos stakeholders e levou em consideração os métodos que levam menos tempo, com menos dificuldade, pouco esforço, porém, que ainda mantém consistência. Na imagem 9, pode ser vista essa comparação:

Evaluation Criteria	Simple Ranking	MoScoW	100 dollar	AHP
Ratio Scale Information			Yes	Yes
High Confidence from User	Yes	Yes	Yes	
Consistent	Yes	Yes	Yes	Yes
Low difficulty	Yes	Yes	Yes	
Low effort	Yes	Yes	Yes	
Able to handle large number of alternatives		Yes		

Figura 9 – Características dos Métodos de Priorização(KHAN et al., 2015)

A técnica de priorização escolhida para este trabalho foi a MoScoW, uma técnica nominal que se utiliza de categorias para classificar os requisitos de acordo com sua importância. Uma das mais fáceis de se usar, utilizada por analistas e stakeholders para priorizar os requisitos de forma colaborativa (HUDAIB et al., 2018). A técnica consiste em classificar os requisitos em quatro categorias:

- **M - Must have.** Nessa categoria, os requisitos devem estar no projeto. Uma falha de entregar esses requisitos pode acarretar em falha no projeto.
- **S - Should have.** Uma funcionalidade de alta prioridade, mas que não é crítica para o lançamento do produto. Ainda sim, é considerada importante e de alto valor para os usuários finais.
- **C - Could have.** Requisitos desejáveis, porém, não necessários.
- **W - Won't have.** Requisitos que não serão implementados no atual desenvolvimento, mas poderiam ser incluídos posteriormente.

Categoria	Requisito	Prioridade
Carona [C]	[C01] - Visualizar caronas disponíveis no dia	M
	[C02] - Indicar e visualizar horário de saída da carona	M
	[C03] - Indicar e visualizar local de partida	M
	[C04] - Indicar e visualizar local de destino	M
	[C05] - Indicar e visualizar trajeto da carona	M
	[C06] - Escolher quantidade de assentos no carro	M
	[C07] - Receber lembrete de carona	S
	[C08] - Indicar e visualizar pagamento voluntário ou compartilhado	W
	[C09] - Visualizar histórico de caronas	S
	[C10] - Indicar e visualizar tempo deslocamento	W
	[C11] - Agendar/programar carona	M
	[C12] - Adicionar amigos à rede	C
	[C13] - Mostrar as caronas que o passageiro está vinculado	M
	[C14] - Pesquisar caronas pelo seu destino	M
	[C15] - Escolher quilometragem máxima de desvio de rota	W
Segurança [S]	[S01] - Entrar na aplicação por e-mail	M
	[S02] - Visualizar informações do motorista e dos passageiros	M
	[S03] - Verificar se é aluno (Verificação através do email institucional)	S
	[S04] - Escolher o sexo do motorista/passageiro	M
	[S05] - Compartilhar localização em tempo real	S
	[S06] - Enviar informações sobre a carona para uma pessoa de confiança	S
	[S07] - Visualizar pessoas na mesma carona	M
Comunicação [CO]	[CO01] - Iniciar uma conversa entre o passageiro e o motorista	M
	[CO02] - Bloquear usuário	W
Pagamento [P]	[P01] - Cartão de crédito ou débito	W
	[P02] - Indicar pagamento em dinheiro ou PIX	W
Gamificação [G]	[G01] - Avaliar motorista e passageiro	W
	[G02] - Acumular pontos por carona	S
	[G03] - Ranking de motoristas que mais ofereceram caronas	W
	[G04] - Subir de nível de acordo com pontuação	W

Tabela 3 – Priorização dos Requisitos Funcionais (Autoria própria)

3.2 Desenvolvimento do Aplicativo

Nesse tópico, é abordado todas as fases do desenvolvimento deste projeto. Na seção 3.2.1, Método de Desenvolvimento do Aplicativo; na seção 3.2.2, Prototipação; na seção 3.2.3, Tecnologias; na seção 3.2.4, Banco de Dados; na seção 3.2.5, Arquitetura; e, por

último, na seção 3.2.6, Regras de Gamificação.

3.2.1 Método de Desenvolvimento do Aplicativo

O método de desenvolvimento do aplicativo será baseado na adaptação do framework ágil 'Scrum', 'Scrum Solo' e contará com a utilização da metodologia Kanban para organizar e manter o rastreamento dos requisitos que foram desenvolvidos.

3.2.1.1 Scrum Solo

O Scrum Solo vem como uma variante do Scrum para o desenvolvimento individual.



Figura 10 – Ciclo Scrum Solo (PAGOTTO et al., 2016)

Como pode ser visto na imagem acima, o Product Backlog e o Sprint Backlog ocorrem de forma igual em ambos frameworks. Comparado ao Scrum, o Scrum Solo não possui reuniões diárias e as Sprints são reduzidas a uma semana. No final de cada Sprint, deve ser entregue um incremento de valor ao produto. Quando necessário, haverá reuniões para validação do produto com as partes interessadas (PAGOTTO et al., 2016)

Neste trabalho, foram utilizadas algumas das atividades propostas pelo framework Scrum Solo: Product Backlog, Sprint Backlog, Sprint, Protótipo do Produto e Parte do Produto desenvolvido.

3.2.1.2 Kanban

O kanban para este projeto traz um incremento de melhoria no processo de desenvolvimento do software. Do japonês, 'kanban' significa "cartão de visualização" e, na abordagem de desenvolvimento de software, significa que somente as funcionalidades requisitadas pelo cliente serão desenvolvidas. Este último elimina funcionalidades onde se-

riam interessantes apenas para os desenvolvedores, não requeridas pelo cliente e, portanto, talvez nunca seriam usadas (RAJU; KRISHNEGOWDA, 2013).

O kanban utilizado para o projeto foi o GitHub. O kanban limita o número de funcionalidades em estado de “em desenvolvimento”, onde outras funcionalidades só serão adicionadas caso as atuais sejam desenvolvidas. Isto ajuda a balancear em geral as atividades de desenvolvimento por eliminar o “entupimento” no ciclo de desenvolvimento (RAJU; KRISHNEGOWDA, 2013).

3.2.2 Prototipação

É uma visão inicial de um projeto, utilizado para experimentar opções disponíveis no produto. Permite aos usuários verificarem se realmente o produto vai atendê-los. Através dele, pode-se verificar pontos fortes e fracos e, assim, propor novos requisitos ao sistema. Além disso, ele pode mostrar erros e omissões nos requisitos levantados (SOMMERVILLE, 2011).

Os protótipos foram desenvolvidos em 2020 utilizando a ferramenta figma, uma ferramenta online de design, que é também de baixa fidelidade aos desenhados no papel.

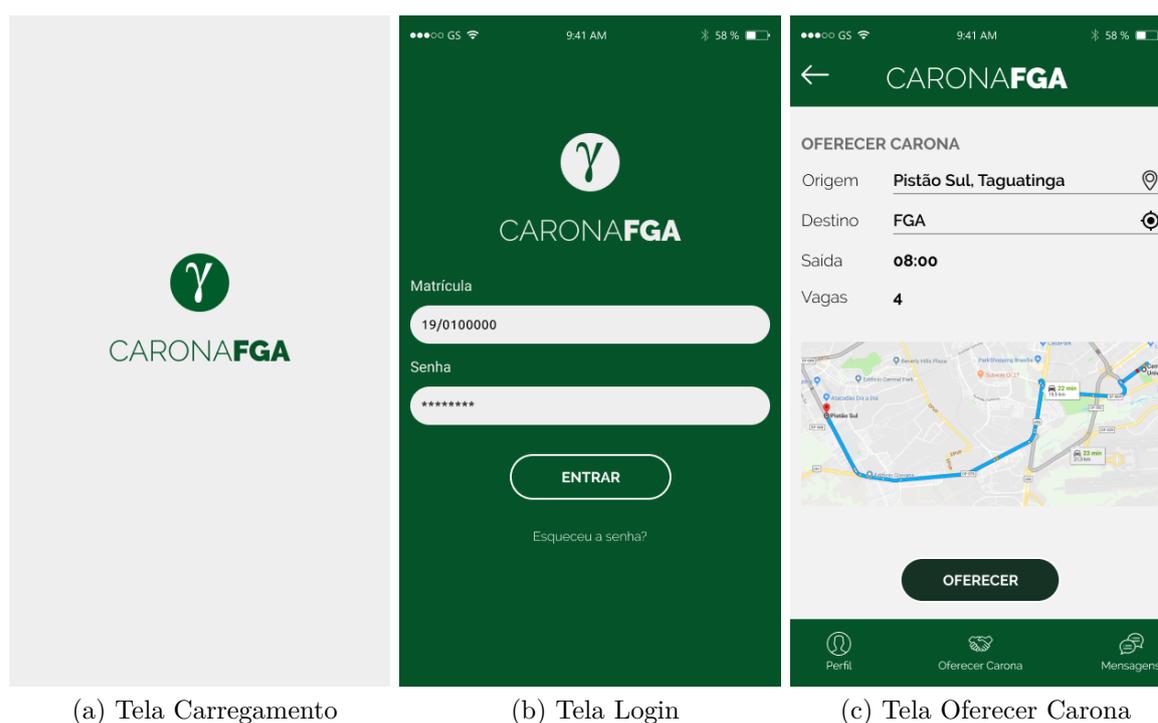
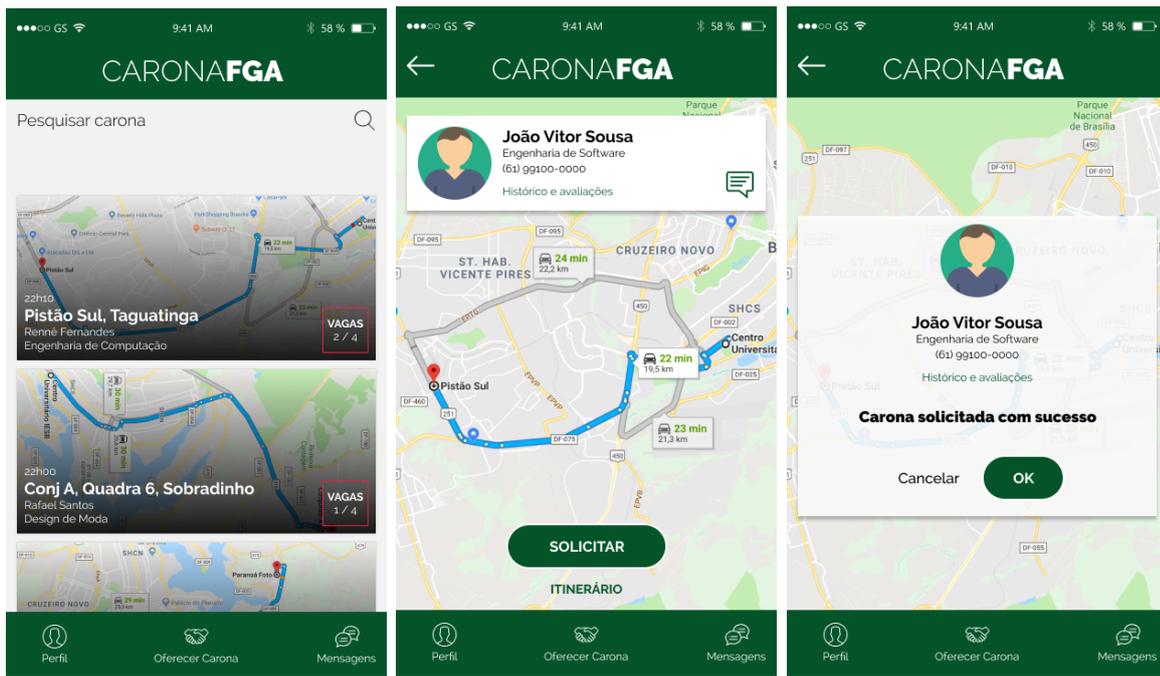


Figura 11 – Telas Protótipo(Autoria Própria)



(a) Tela Caronas

(b) Tela Solicitar Carona

(c) Tela Carona Solicitada

Figura 12 – Telas Protótipo(Autoria Própria)



(a) Tela Solicitar Carona

(b) Tela Carona Aceita

Figura 13 – Telas Protótipo(Autoria Própria)

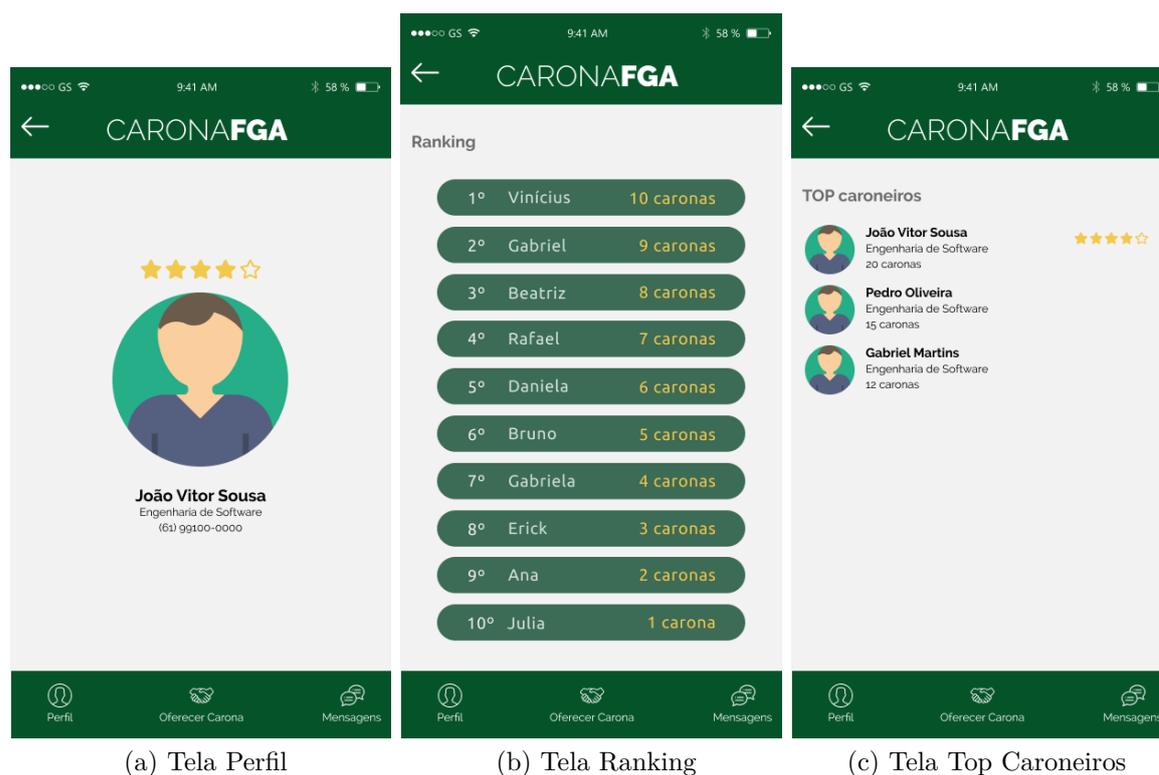


Figura 14 – Telas Protótipo(Autoria Própria)

3.2.3 Tecnologias

Neste tópico, é abordado as tecnologias utilizadas para o desenvolvimento do projeto. Este estará disponível para download em sua página de repositório no GitHub.

3.2.3.1 Sistema operacional

O sistema operacional utilizado para rodar a aplicação foi o Android e foi baseado em Linux e em diversas arquiteturas de processadores. Linux conta com 3.3 bilhões de usuários no mundo em 2023, o que faz com que ele tenha mais de 70% de quota no mercado de sistema operacional em aparelhos móveis (RUBY, 2023).

3.2.3.2 Ambiente de Desenvolvimento Integrado(IDE) e Linguagem

O IDE utilizado para o desenvolvimento foi o Android Studio Dolphin 2021.3.1, disponibilizado pelo Google. Ele possui integração com o GitHub para melhor controle de versionamento, automação de build, editor de código inteligente, emulador com recursos completos e outros. A linguagem escolhida para o desenvolvimento foi Kotlin, linguagem recomendada pelo Google para desenvolvimento de apps nativos. Uma linguagem de multiplataforma, concisa, estaticamente tipada e que possui interoperabilidade com Java. O aplicativo foi desenvolvido 100% com Kotlin.

3.2.3.3 Serviços em Nuvem

Para serviços em nuvem, foi utilizado o Firebase, plataforma de integração com aplicativos de diversas linguagens. Possui diversas integrações: de banco de dados, mensagens, autenticação, análise de uso, aprendizado de máquina e outros. Para este trabalho, foi utilizado o Realtime Database e a autenticação foi feita pela plataforma do Google Accounts. O primeiro, banco de dados NoSQL JSON em tempo real em nuvem; o segundo, autenticação de usuário através da conta do Google presente em todo smartphone Android.

3.2.3.4 Versionamento de Código e Repositório

Para o versionamento de código, foi utilizado o Git, sistema de controle de versões distribuído. Foi usado em conjunto com GitHub para o armazenamento do código.

3.2.4 Banco de Dados

Para este projeto, o objetivo foi a facilidade de integração com aplicativos de diferentes plataformas e a atualização em tempo real de todos aparelhos conectados a ele. O banco de dados escolhido foi o Realtime Database da Firebase. Esse BD é NoSQL em nuvem. Ele usa um método para armazenar dados como uma grande árvore em JSON. Ou seja, se ele estiver conectado em locais diferentes, assim que houver uma atualização no banco, ele é atualizado para os clientes conectados. Também funciona de forma offline (NAMEE et al., 2022).

Ele simplifica o desenvolvimento backend. O Firebase Realtime Database requer somente código que altere o banco de dados do lado do cliente, não necessitando de código do lado do servidor (OHYVER et al., 2019).

A partir disso, a estrutura do BD possui quatro nós raízes: `signup_info`, `rides`, `ride_request` e `request_response`. É mostrado na imagem 15:



Figura 15 – Estrutura Banco de Dados(Autoria Própria)

- **signup_info**

Dentro deste nó, existe outro nó que faz referência ao identificador único do usuário (UID). Nele, fica armazenado as informações do usuário, como pode ser visto na imagem 16:



Figura 16 – Estrutura rides(Autoria Própria)

- **rides**

Dentro deste nó, existe outro nó, também com o UID, da carona. Dentro dele, estão as informações da carona, como mostra a imagem 17:

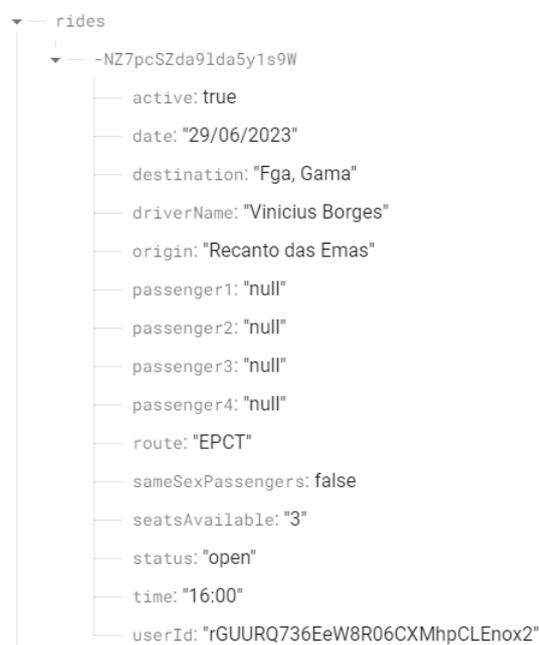


Figura 17 – Estrutura rides(Autoria Própria)

- `ride_request`

Neste nó, há outro nó com a UID dos objetos e suas informações relevantes para fazer a solicitação de uma carona, visto na imagem 18:

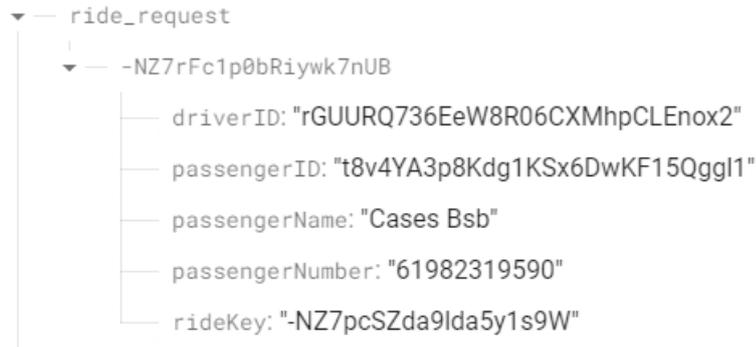


Figura 18 – Estrutura `ride_request`(Autoria Própria)

- `request_response`

Este nó também possui UID para cada objeto registrado e, dentro dele, as informações referentes à resposta da solicitação de carona, como mostra a imagem 19:

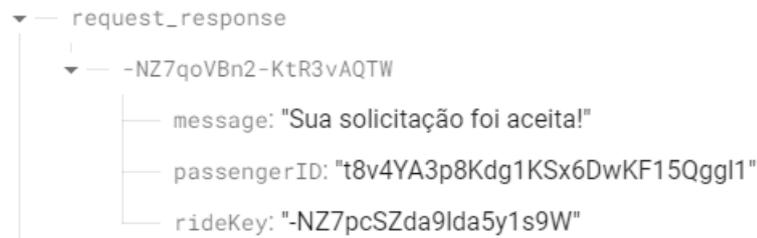


Figura 19 – Estrutura `request_response`(Autoria Própria)

3.2.5 Arquitetura

Para este trabalho, a arquitetura escolhida foi a arquitetura MVC (Model View Controller). A Model é um conjunto de entidades que representam conhecimento, informação e também define as regras para a manipulação de dados. A Model não deve saber de outros componentes, mas pode enviar notificações sobre o estado de seus objetos. A View é responsável por prover uma representação visual da Model. E, por último, a Controller é o ponto de entrada principal, a que manipula a View e transcreve as ações do usuário para a Model (DAOUDI et al., 2019). Essas interações podem ser vistas na imagem 20:

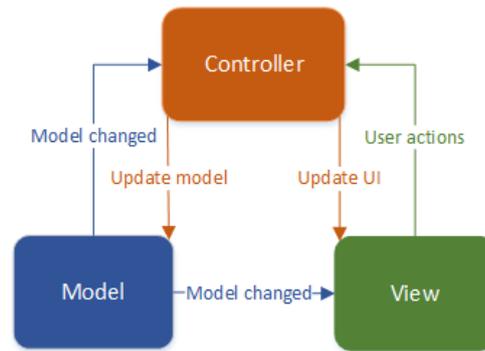


Figura 20 – Arquitetura MVC(VASILYIY, 2015)

No projeto do aplicativo, temos cada elemento dessa arquitetura bem definida, sendo a View os layouts em XML; as Controllers sendo as Fragments e Activities; e a Model sendo o RealTime Database da Firebase. Na imagem 21, pode-se ver essa categorização. Em verde, os componentes da Controller e, em roxo, os layouts das Views:

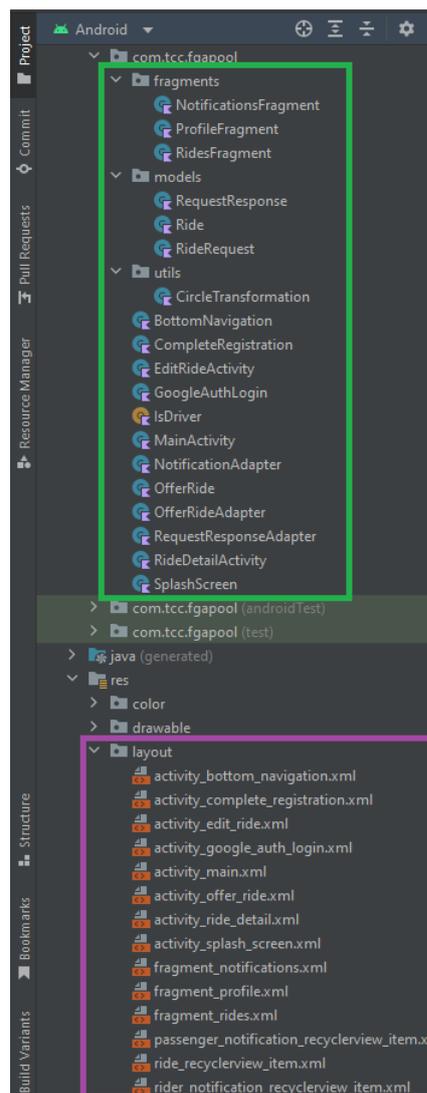


Figura 21 – Organização dos Arquivos(Autoria Própria)

3.2.6 Regras de Gamificação

Para o início deste projeto, foram escolhidas algumas das técnicas de gamificação. Neste tópico, será abordado o funcionamento delas e a distribuição de pontuação.

3.2.6.1 Avaliação

Os passageiros avaliarão a carona entre 1 e 5 estrelas e os motoristas avaliarão os passageiros entre 1 e 5 estrelas. Será mostrado a média das pontuações em seus perfis e comentários.

3.2.6.2 Distribuição de pontos

- O motorista recebe 5 pontos por passageiro em uma carona completada
- O motorista recebe 2 pontos por carona agendada
- O motorista recebe 2 pontos por passageiro avaliado
- O passageiro recebe 10 pontos por carona completada
- O passageiro recebe 5 pontos por carona avaliada

3.2.6.3 Níveis

Em cada nível, será incrementado um valor de 20 pontos em relação ao nível anterior.

- **Nível 1** - 0 a 10 pontos
- **Nível 2** - 11 a 30 pontos
- **Nível 3** - 31 a 50 pontos
- **Nível 4** - 51 a 70 pontos

E assim por diante.

3.2.6.4 LeaderBoard

Passageiro e motorista terão suas respectivas leaderboards, que serão mostradas na tela os dez com maior pontuação em ordem decrescente.

4 FGA Pool

Neste tópico, será abordado tudo que diz respeito ao que foi desenvolvido na aplicação, os desafios iniciais e as mudanças que ocorreram ao longo do desenvolvimento. O início do desenvolvimento foi de 20 de agosto até 6 de outubro de 2021 e, posteriormente, retomado em 03 de janeiro e foi até 07 de junho de 2023, totalizando aproximadamente 7 meses de desenvolvimento. O projeto contou com 55 commits e está disponível em seu repositório do GitHub¹.

Inicialmente, a ideia era utilizar o Google Maps para mostrar o trajeto da carona. Porém, para pesquisar os locais em que precisa ser colocado a origem e destino, era necessário utilizar a API do Google Places, que pega as coordenadas dos lugares para transpor no mapa. A utilização desta API ficou inviável, pois era de alto valor e, na categoria Free, era liberado apenas uma requisição por dia. Após isso, foi retirado do código as implementações relativas ao Google Maps e foi remodelado para inserir as informações manualmente. Outro ponto foi a correção de um bug, que, a partir de quatro caronas, o menu de editar e deletar não aparecia mais do lado do carona. Isto fez com que tivesse uma Sprint apenas para a correção dele, reestruturando todo o código.

De acordo com a tabela 3, as seguintes funcionalidades foram desenvolvidas:

[C01] - Visualizar caronas disponíveis no dia; [C02] - Indicar e visualizar horário de saída da carona; [C03] - Indicar e visualizar local de partida; [C04] - Indicar e visualizar local de destino; [C05] - Indicar e visualizar trajeto da carona; [C11] - Agendar/programar carona; [C14] - Pesquisar caronas pelo seu destino; [S01] - Entrar na aplicação por e-mail; [S02] - Visualizar informações do motorista e dos passageiros; [S04] - Escolher o sexo do motorista/passageiro; [S07] - Visualizar pessoas na mesma carona; [CO01] - Iniciar uma conversa entre o passageiro e o motorista;

Abaixo, é mostrada a aplicação em sua última versão até o momento.

¹ Disponível em: <https://github.com/ViBorges/FGAPool>

4.1 Login

A primeira tela é a de carregamento da aplicação; a segunda, a de Login; e a terceira, após clicar em “Entrar com o Google”, é onde tem as opções de escolha de conta.

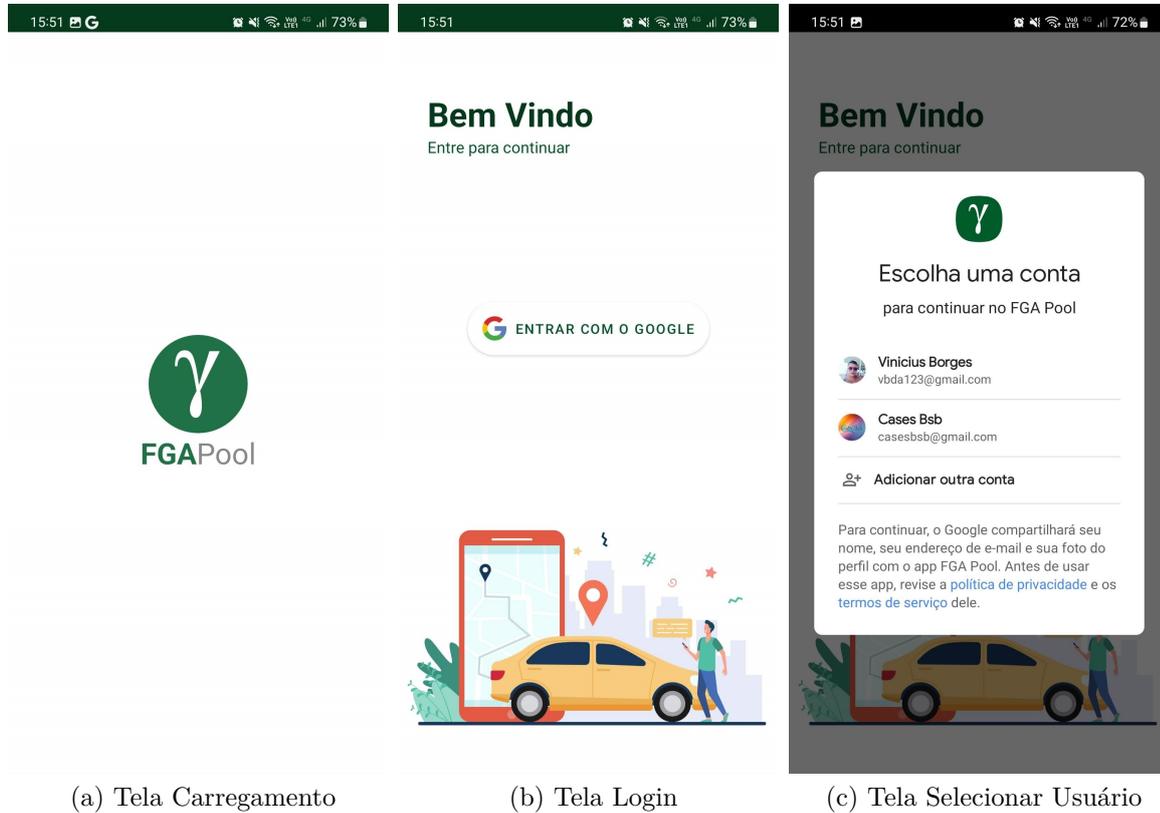
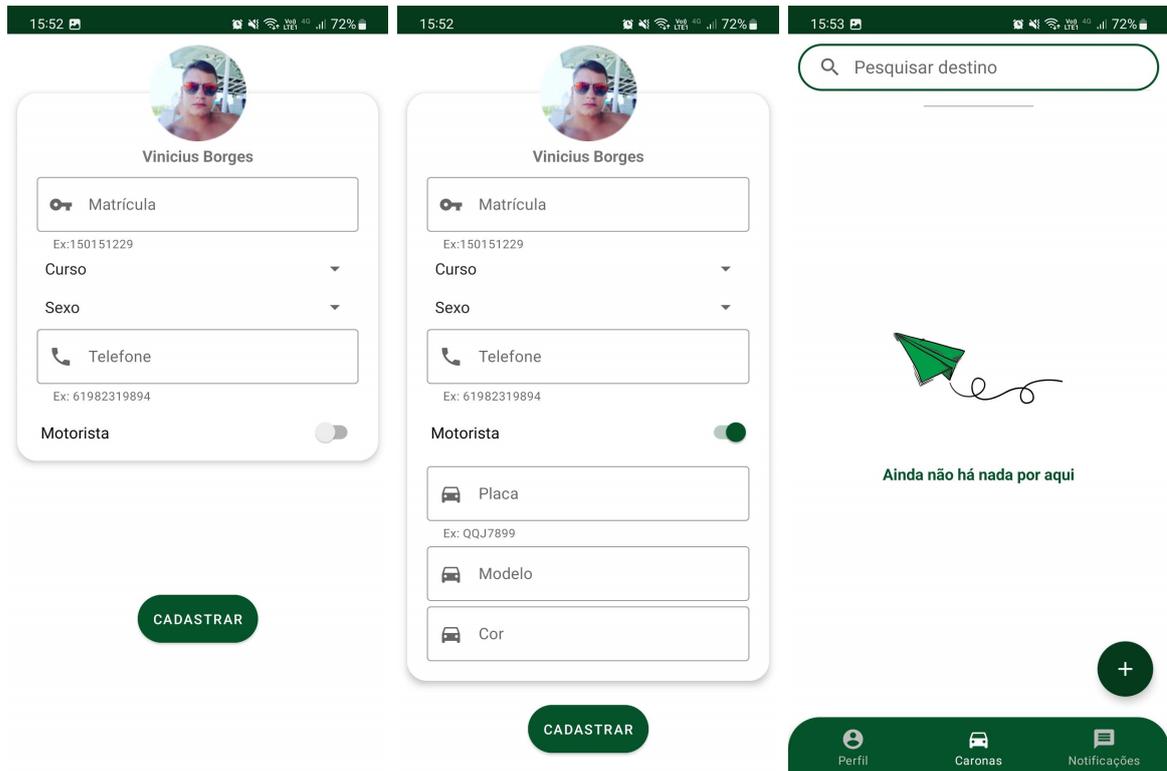


Figura 22 – Telas Aplicativo (Autoria Própria)

4.2 Cadastro

Após realizar o login, caso seja o primeiro acesso, irá aparecer a primeira tela de cadastro e é diferente caso escolha ser passageiro (primeira) ou caso escolha ser motorista (segunda). A terceira é a tela inicial do aplicativo na visão do motorista.



(a) Tela Cadastro Passageiro

(b) Tela Cadastro Motorista

(c) Tela Principal

Figura 23 – Telas Aplicativo(Autoria Própria)

4.3 Cadastro Carona

Na tela inicial do motorista, após pressionar o botão de adicionar, aparecerá a primeira tela, que é a de cadastro de uma carona. Depois de cadastrado, na segunda tela, a carona irá aparecer para todos os passageiros; sendo, para o motorista, apenas suas caronas cadastradas. No terceiro, caso o motorista deseje editar ou deletar a carona, deve apertar o três pontinhos da carona.

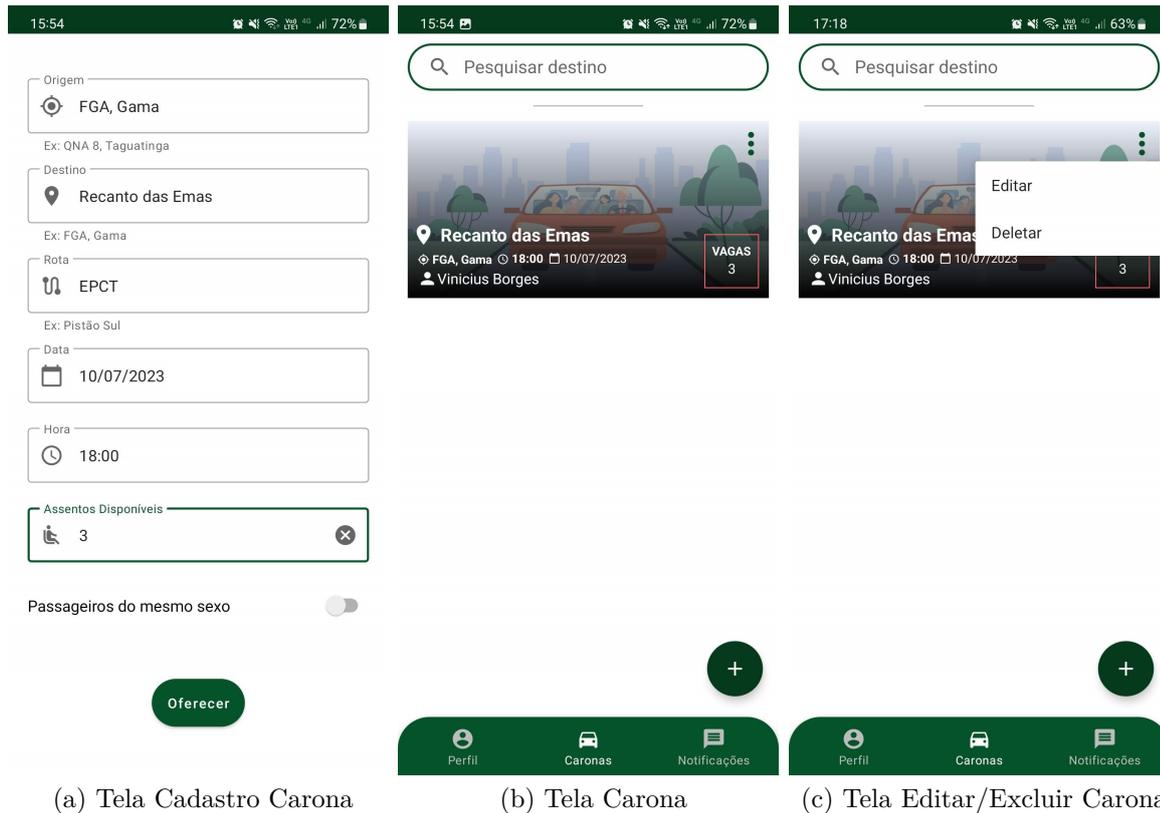
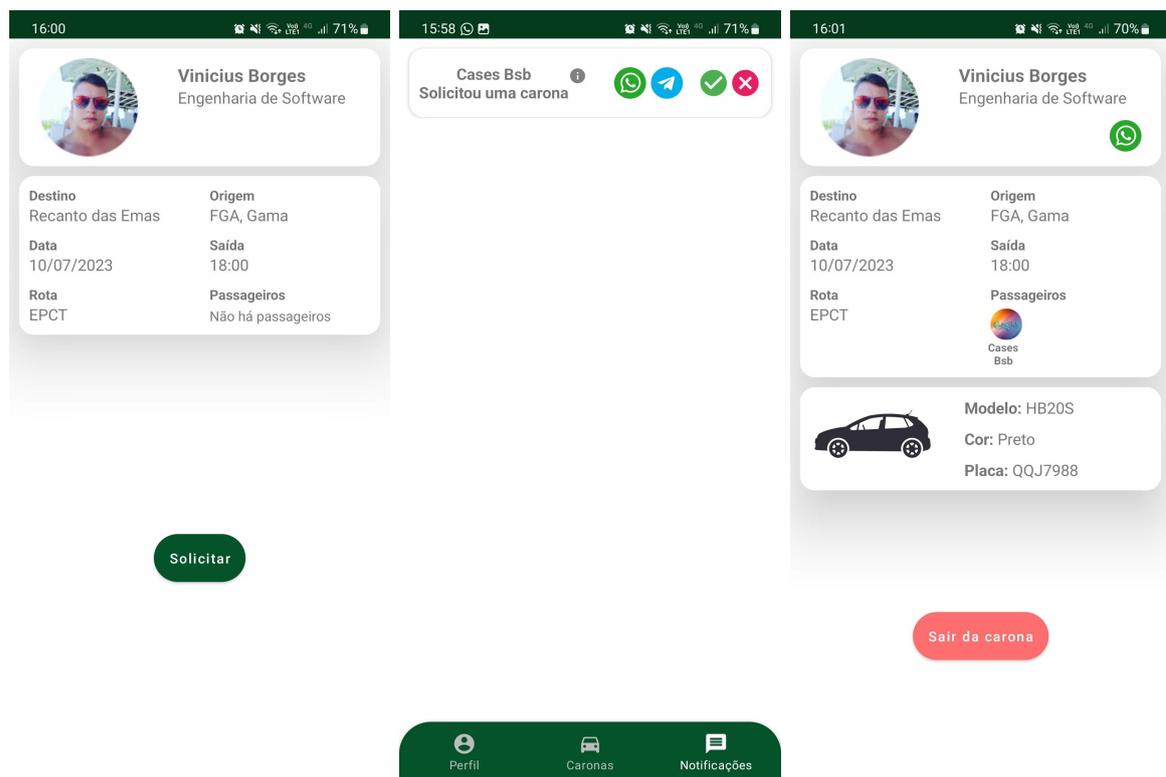


Figura 24 – Telas Aplicativo(Autoria Própria)

4.4 Solicitar Carona

O passageiro, ao clicar na carona, irá abrir outra tela com detalhes da carona, podendo solicitá-la como na primeira tela. Após a solicitação do passageiro, irá surgir uma notificação para o motorista, como na segunda tela. Ele pode aceitar ou não a solicitação, assim como canais para se comunicar com o solicitante. Após o motorista aceitar a solicitação, informações como carro e whatsapp do motorista irão aparecer para o passageiro. Também tem a opção de sair da carona, como visto na terceira tela.



(a) Tela Detalhe Carona

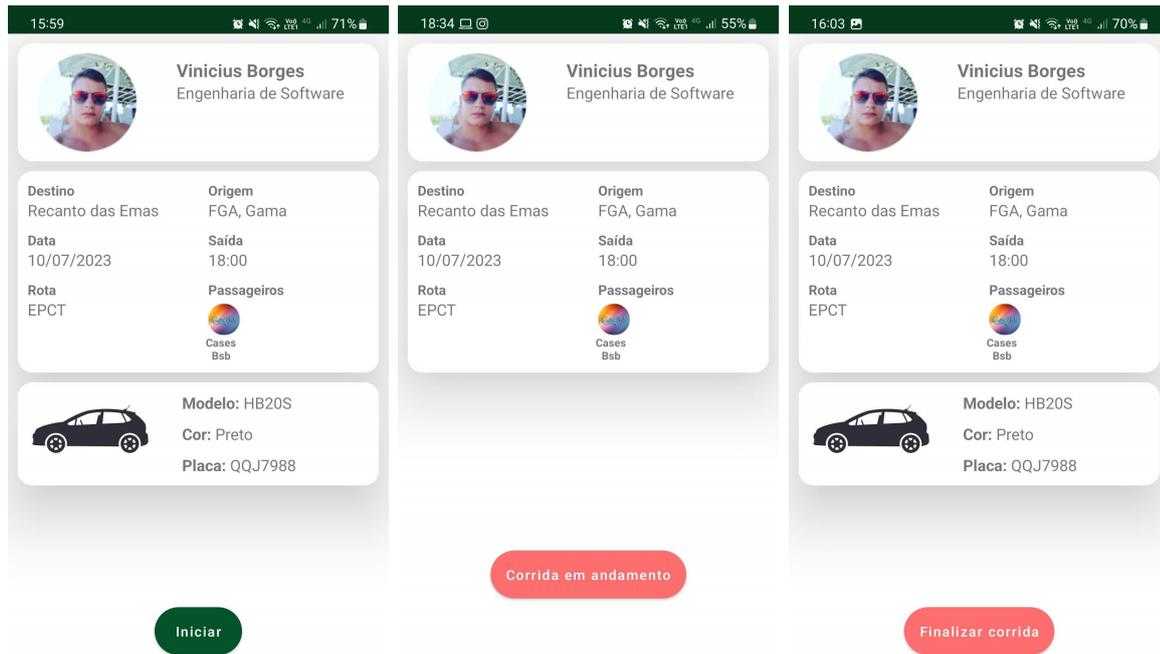
(b) Tela Notificações

(c) Tela Carona Aceita

Figura 25 – Telas Aplicativo(Autoria Própria)

4.5 Iniciar Corrida

Na primeira tela, se houver pelo menos um passageiro, será possível iniciar a corrida. Após iniciar a corrida, aparecerá para o passageiro que a corrida está em andamento (segunda tela) e, para o motorista, a opção de finalizar a corrida (terceira tela). Após finalizar a corrida, ela ficará salva no banco de dados para futuras consultas e/ou novas funcionalidades.



(a) Tela Detalhe Carona:
Motorista

(b) Tela Detalhe Carona:
Passageiro

(c) Tela Detalhe Carona:
Motorista

Figura 26 – Telas Aplicativo(Autoria Própria)

4.6 Outras Funcionalidades

Tanto para o passageiro quanto para o motorista será possível pesquisar e filtrar caronas por destino (primeira e segunda tela). Para o passageiro, tem a opção de ativar “Minhas Caronas” abaixo do campo de pesquisa. Feito isso, serão filtradas as caronas em que o passageiro está vinculado (terceira tela).

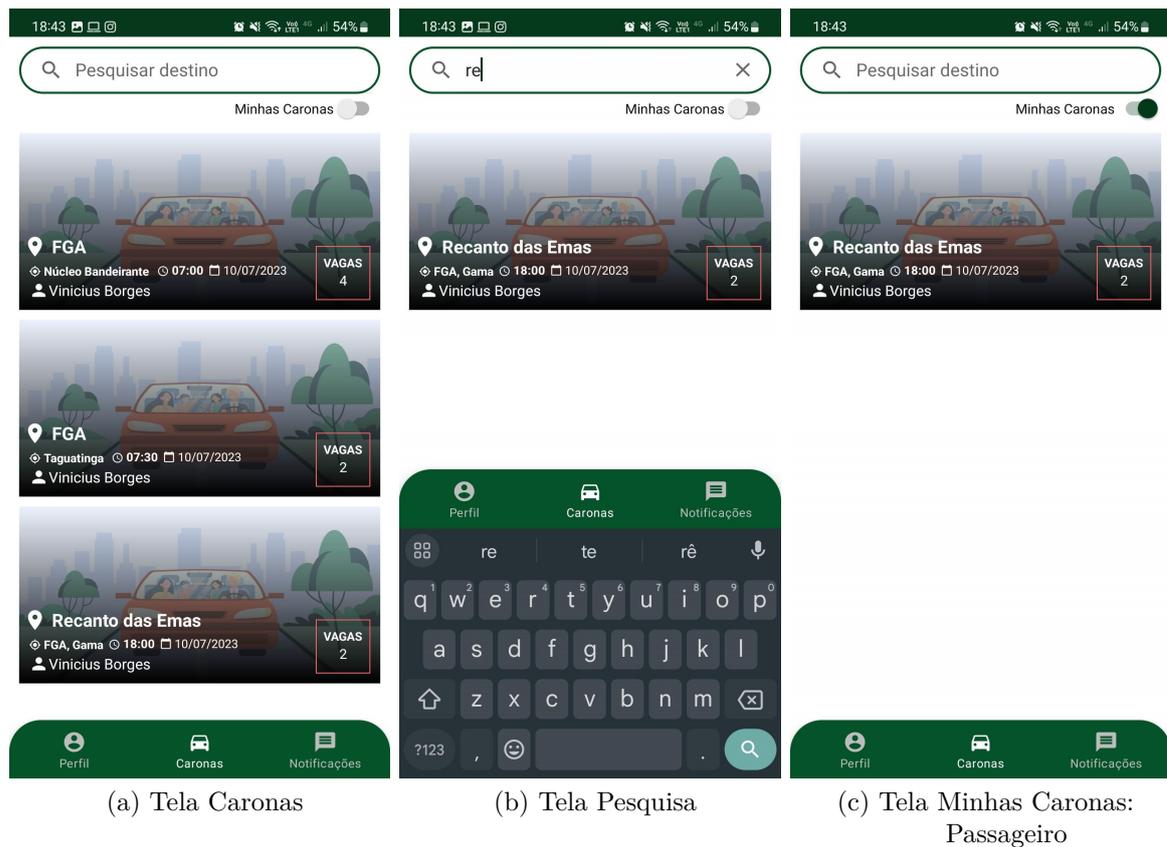


Figura 27 – Telas Aplicativo(Autoria Própria)

4.7 Perfil

Como pode ser visto nas telas, o perfil com foto é associado à conta da Google e mostra nome, curso e se é passageiro ou motorista. Há espaço para futuras atualizações associadas à gamificação. E, por último, o botão de logout da aplicação.

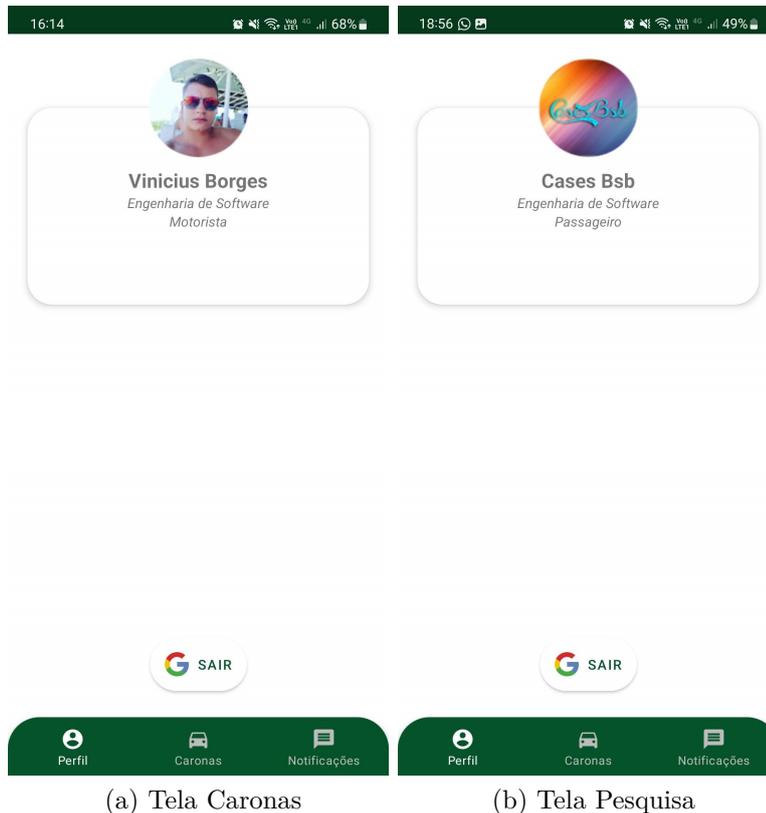


Figura 28 – Telas Aplicativo (Autoria Própria)

5 Considerações Finais

A proposta do trabalho foi voltada para o desenvolvimento de um app mobile que ajudasse no compartilhamento de mobilidade na Faculdade do Gama (FGA) e contou com o auxílio da gamificação para tornar o aplicativo mais atraente. Visto que o tempo para desenvolver o core da aplicação foi relativamente longo, alguns dos requisitos ficaram de fora dessa release inicial, deixando uma abertura para trabalhos futuros. Nos estudos levantados, foram vistos o crescimento populacional no DF e a frota de veículos pessoais, que acarretam em gastos excessivos de combustíveis e emissão de CO₂ por pessoa. Também foram levadas em conta a crescente quantidade de alunos na FGA e falta de segurança aos redores da faculdade, o que motivou a desenvolver uma aplicação com potencial para reduzir esses efeitos. Dentre as pesquisas, foi necessário uma visão geral sobre mobilidade compartilhada na literatura e suas principais modalidades para que, depois da pesquisa, o aluno pudesse escolher a que mais se encaixa em seu contexto. Foi uma pesquisa de extrema importância, principalmente para ouvir as inseguranças dos alunos sobre o compartilhamento de carona e, a partir disso, saná-las através de implementações. Para tornar a aplicação mais atraente, foram realizadas pesquisas em torno do termo "gamificação", o que retornou principalmente o trabalho de (CHOU, 2015), que mostra seus core drives, principais técnicas de jogos utilizadas em cada uma dessas cores e também guia o usuário para algum tipo de comportamento desejado.

O desenvolvimento foi um desafio por se tratar de linguagem e tecnologia novas para o autor, mas que foram de extrema importância para conseguir completar essa primeira release da aplicação. Serviu de aprendizado durante os 7 meses de desenvolvimento, o que permitiu ao autor aprimorar o conhecimento em desenvolvimento Android nativo. Como não foi possível o desenvolvimento de todos requisitos e a aplicação das técnicas de gamificação escolhidas devido à falta de tempo para o desenvolvimento completo da aplicação, fica a oportunidade para dar continuidade ao projeto.

5.1 Trabalhos e Aplicações Futuras

O software ainda pode ser melhorado de várias formas, como aplicação de testes, melhoria de código, aplicação de padrões de projeto e outras partes que compõem a engenharia de software. Em especial sobre as funcionalidades listadas neste trabalho que não foram desenvolvidas, segue os tópicos a serem desenvolvidos em trabalhos futuros:

- [C07] - Receber lembrete de carona
- [C08] - Indicar e visualizar pagamento voluntário ou compartilhado

- [C09] - Visualizar histórico de caronas
- [C10] - Indicar e visualizar tempo deslocamento
- [C12] - Adicionar amigos à rede
- [C15] - Escolher quilometragem máxima de desvio de rota
- [S03] - Verificar se é aluno (Verificação através do email institucional)
- [S05] - Compartilhar localização em tempo real
- [S06] - Enviar informações sobre a carona para uma pessoa de confiança
- [CO02] - Bloquear usuário
- [P01] - Pagamento por cartão de crédito ou débito
- [P02] - Indicar pagamento em dinheiro ou PIX
- [G01] - Avaliar motorista e passageiro
- [G02] - Acumular pontos por carona
- [G03] - Ranking de motoristas que mais ofereceram caronas
- [G04] - Subir de nível de acordo com pontuação

Referências

- BESSA, C. F. D. d. O. *Desenvolvimento de software gamificado para fomentar a mobilidade compartilhada em ambiente universitário*. 2020. Disponível em: <<https://bdm.unb.br/handle/10483/26359>>. Citado 7 vezes nas páginas 6, 7, 29, 30, 31, 32 e 33.
- CHOU, Y.-k. Actionable Gamification. p. 501, 2015. Citado 6 vezes nas páginas 22, 23, 24, 25, 26 e 53.
- CHOU, Y.-k. *The Octalysis Framework for Gamification Behavioral Design*. 2020. <<https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>>. [Online; accessed 7-May-2020]. Citado 5 vezes nas páginas 6, 23, 24, 25 e 26.
- CICI, B. et al. Assessing the Potential of Ride-Sharing Using Mobile and Social Data. *arXiv:1305.3876 [physics]*, mar. 2014. ArXiv: 1305.3876. Disponível em: <<http://arxiv.org/abs/1305.3876>>. Citado na página 13.
- DAOUDI, A. et al. An exploratory study of MVC-based architectural patterns in Android apps. In: . [S.l.: s.n.], 2019. p. 1711–1720. ISBN 978-1-4503-5933-7. Citado na página 42.
- DETERDING, S. et al. From game design elements to gamefulness: defining "gamification". In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. [S.l.: s.n.], 2011. p. 9–15. Citado na página 22.
- DPO. *Folder UnB em Números*. 2019. <http://www.dpo.unb.br/index.php?option=com_content&view=article&id=47&Itemid=872>. [Online; accessed 07-December-2019]. Citado na página 14.
- GROH, F. Gamification: State of the Art Definition and Utilization. p. 8, 2012. Citado na página 22.
- GUPTA, S.; BURIRO, A.; CRISPO, B. DriverAuth: Behavioral biometric-based driver authentication mechanism for on-demand ride and ridesharing infrastructure. *ICT Express*, v. 5, n. 1, p. 16–20, 2019. Publisher: Elsevier. Citado na página 21.
- HUDAIB, A. et al. Requirements Prioritization Techniques Comparison. *Modern Applied Science*, v. 12, jan. 2018. Citado na página 34.
- HUOTARI, K.; HAMARI, J. Defining gamification: a service marketing perspective. In: *Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12*. Tampere, Finland: ACM Press, 2012. p. 17. ISBN 978-1-4503-1637-8. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2393132.2393137>>. Citado na página 22.
- IBGE. *Panorama DF*. 2019. <<https://cidades.ibge.gov.br/brasil/df/panorama>>. [Online; accessed 29-November-2019]. Citado 2 vezes nas páginas 13 e 14.

- KHAN, J. et al. Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique. *International Journal of Modern Education and Computer Science*, v. 7, p. 53–59, nov. 2015. Citado 3 vezes nas páginas 6, 33 e 34.
- LIU, K.; ZHANG, J.; YANG, Q. Bus Pooling: A Large-Scale Bus Ridesharing Service. *IEEE Access*, v. 7, p. 74248–74262, 2019. ISSN 2169-3536. Conference Name: IEEE Access. Citado na página 13.
- MACHADO, C. A. S. et al. An Overview of Shared Mobility. *Sustainability*, v. 10, n. 12, p. 4342, nov. 2018. ISSN 2071-1050. Publisher: MDPI AG. Disponível em: <<https://doaj.org>>. Citado 5 vezes nas páginas 6, 17, 18, 20 e 21.
- NAMAZU, M. et al. Is carsharing for everyone? Understanding the diffusion of carsharing services. *Transport Policy*, v. 63, p. 189–199, 2018. Publisher: Elsevier. Citado na página 19.
- NAMEE, K. et al. Development a Teleconsultation Platform for Outpatients during the COVID-19 Pandemic based on Cloud Firestore and Realtime Databases. In: *2022 14th Biomedical Engineering International Conference (BMEiCON)*. [S.l.: s.n.], 2022. p. 1–5. ISSN: 2334-3052. Citado na página 40.
- OHYVER, M. et al. The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test. *Procedia Computer Science*, v. 157, p. 396–405, jan. 2019. ISSN 1877-0509. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050919311500>>. Citado na página 40.
- PAGOTTO, T. et al. Scrum solo: Software process for individual development. In: *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.: s.n.], 2016. p. 1–6. Citado 2 vezes nas páginas 6 e 36.
- RAJU, H. K.; KRISHNEGOWDA, Y. T. Kanban Pull and Flow — A transparent workflow for improved quality and productivity in software developmet. In: *Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2013)*. [S.l.: s.n.], 2013. p. 44–51. Citado na página 37.
- RUBY, D. *Android Statistics In 2023*. 2023. <<https://www.demandsage.com/android-statistics/>>. [Online; accessed 5-July-2023]. Citado na página 39.
- SANTOS, G. Sustainability and Shared Mobility Models. *Sustainability*, v. 10, n. 9, p. 3194, set. 2018. ISSN 2071-1050. Publisher: MDPI AG. Disponível em: <<https://doaj.org>>. Citado na página 18.
- SHAHEEN, S. et al. *Travel behavior: Shared mobility and transportation equity*. [S.l.], 2017. Citado na página 18.
- SHAHEEN, S.; COHEN, A.; ZOHDY, I. *Shared mobility: current practices and guiding principles*. [S.l.], 2016. Citado na página 21.
- SHAHEEN, S. A.; MALLERY, M. A.; KINGSLEY, K. J. Personal vehicle sharing services in North America. *Research in Transportation Business & Management*, v. 3, p. 71–81, 2012. Publisher: Elsevier. Citado na página 19.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. ed. São Paulo: Pearson Prentice Hall, 2011. OCLC: 940079598. ISBN 978-85-7936-108-1. Citado 2 vezes nas páginas 31 e 37.

SSP/DF. *Dados Criminais*. 2019. <<http://www.ssp.df.gov.br/dados-por-regiao-administrativa/>>. [Online; accessed 10-December-2019]. Citado na página 14.

TÓTH, ; TÓVÖLGYI, S. The introduction of gamification: A review paper about the applied gamification in the smartphone applications. In: *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. [S.l.]: IEEE, 2016. p. 000213–000218. Citado 2 vezes nas páginas 13 e 14.

VASILIIY. *MVC Architectural Pattern in Android*. 2015. <<https://www.techyourchance.com/mvc-android-1/>>. [Online; accessed 10-July-2023]. Citado 2 vezes nas páginas 6 e 43.

ZICHERMANN, G.; CUNNINGHAM, C. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. [S.l.]: "O'Reilly Media, Inc.", 2011. Google-Books-ID: zZcpuMRpAB8C. ISBN 978-1-4493-1539-9. Citado na página 22.