
**University Bus: Aplicativo Android para
auxiliar no controle do transporte
universitário**

Eugênio Lopes Arantes Júnior

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 30/11/2021

Assinatura: _____

University Bus: Aplicativo Android para auxiliar no controle do transporte universitário

Eugênio Lopes Arantes Júnior

Willyan Michel Ferreira

Monografia apresentada ao Centro Universitário de Formiga UNIFOR/MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof. Willyan Michel Ferreira

Unifor-MG - Formiga
11/2021

*A Deus, a minha família,
e a todos que contribuíram
para a minha formação*

Agradecimentos

É com imensa alegria e felicidade que neste momento único de minha vida, venho prestar os meus agradecimentos, por todos momentos vivenciados e por todos conhecimentos adquiridos ao longo desta jornada tão prazerosa.

Agradeço aos meus pais Eugênio e Maria do Carmo, a minha irmã Thuanny, e ao meu irmão Heitor, por todo o incentivo, apoio, amor e carinho que me deram, vocês são meu porto seguro. Também agradeço a minha namorada Bianca, que ao meu lado, sempre esteve me apoiando com palavras de incentivo e carinho.

Quero expressar minha gratidão também a toda minha família, meus tios e tias, primos e meu avô Antônio, onde cada um de vocês, tiveram participação muito importante para minha formação. Também agradeço a todos os meus colegas e amigos, que estiveram presentes em cada momento, superando juntos todas as batalhas ao longo de nossa formação, e também a todo corpo docente, e em especial, meu orientador Willyan, por toda confiança e experiência concedida.

Enfim, posso dizer que estou muito grato a Deus por ter me proporcionado todos estes momentos, de muitas alegrias, conhecimentos e aprendizados. A todos, minha eterna gratidão!

*”Cada sonho que você deixa pra trás, é um
pedaço do seu futuro que deixa de existir.”*

Steve Jobs

Sumário

Lista de Figuras	ii
1 Introdução	1
1.1 Objetivos do Trabalho	2
1.1.1 Objetivos Específicos	2
1.2 Justificativa	2
1.3 Estrutura da Monografia	3
2 Referencial Teórico	5
2.1 Computação Urbana	5
2.2 Android	5
2.3 SQLite	6
2.4 Firebase	7
3 Estudo da Arte	9
3.1 Considerações iniciais	9
3.2 Aplicativo para atender requisições de deslocamento solicitadas por passageiros .	9
3.3 Blumenbus: aplicativo móvel para consulta de informações do transporte público de Blumenau	10
3.4 Aplicativo de gestão de transporte rodoviário	12
3.5 Considerações finais	12
4 Metodologia e Desenvolvimento	15
4.1 Levantamento de requisitos	15
4.2 Diagrama de caso de uso	15
4.3 Arquitetura do aplicativo	16
4.3.1 Camada Model	17
4.3.2 Camada View	17
4.3.3 Camada Controller	17

4.4	Criação do Banco de dados	17
5	Resultados	23
5.1	Ícone do Aplicativo	23
5.2	Tela Inicial	23
5.3	Cadastro de usuário	24
5.3.1	Tela Cadastrar Motorista	24
5.3.2	Tela Cadastrar Estudante	25
5.4	Vincular à uma turma	26
5.4.1	Tela Procurar turmas existentes	26
5.4.2	Tela de espera	27
5.4.3	Tela criar nova turma	28
5.5	Tela principal	29
5.6	Tela de ida	30
5.6.1	Tela de ida do motorista	30
5.6.2	Tela de ida do estudante	31
5.7	Tela de volta	32
5.7.1	Tela de volta do motorista	32
5.7.2	Tela começar nova viagem	33
5.7.3	Tela de volta do estudante	34
5.8	Tela listagem diária	35
5.8.1	Tela visualizar situação no cenário de ida	36
5.8.2	Tela visualizar situação no cenário de volta	37
5.9	Tela de notificações	37
5.10	Tela dados pessoais	38
5.10.1	Tela meus dados	39
5.10.2	Tela dados da turma	40
5.10.3	Tela participantes da turma	40
5.10.4	Tela sair da turma	41
6	Conclusão	43
6.1	Trabalhos futuros	43

Lista de Figuras

3.1	Tela inicial do aplicativo	10
3.2	Tela dos favoritos	11
3.3	Tela das Linhas	11
3.4	Diagrama de caso de uso - Blumenbus	11
4.1	Diagrama de caso de uso - University Bus	16
4.2	Estrutura geral do banco de dados	18
4.3	Nó dos administradores	18
4.4	Nó contador do identificador dos usuários	19
4.5	Nó contador das solicitações	19
4.6	Nó contador das turmas	19
4.7	Nó que armazena as informações dos estudantes	20
4.8	Nó que armazena as informações dos motoristas	21
4.9	Nó que armazena as turmas	21
4.10	Diagrama de entidade relacionamento	22
5.1	Ícone do University Bus	23
5.2	Tela inicial do aplicativo	24
5.3	Tela de cadastro do motorista	25
5.4	Tela de cadastro do estudante	25
5.5	Tela Vincular à uma turma	26
5.6	Tela de procurar turmas existentes	27
5.7	Tela de confirmar o pedido	27
5.8	Tela de espera	28
5.9	Tela cadastrar nova turma	29
5.10	Tela principal do estudante	30
5.11	Tela principal do motorista	30
5.12	Tela de ida do motorista	31
5.13	Lista de falta	31

5.14	Lista no ponto de ônibus	31
5.15	Tela de ida do estudante	32
5.16	Tela de volta do motorista	33
5.17	Listagem chegaram no ônibus	33
5.18	Listagem não voltam de ônibus	33
5.19	Tela começar nova viagem	34
5.20	Tela de volta do estudante	35
5.21	Tela listagem diária	36
5.22	Tela da situação no cenário de ida	36
5.23	Tela da situação no cenário de volta	37
5.24	Tela de notificações	38
5.25	Tela dados pessoais	39
5.26	Tela meus dados motorista	40
5.27	Tela meus dados estudante	40
5.28	Tela dados da turma	40
5.29	Tela participantes da turma	41
5.30	Tela sair da turma	42

Lista de Siglas

IBGE - *Instituto Brasileiro de Geografia e Estatística*

SDK - *Software Development Kit*

API - *Application Programming Interface*

IDE - *Integrated Development Environment*

JDK - *Java Development Kit*

SQL - *Structured Query Language*

JSON - *JavaScript Object Notation*

MVC - *Model-View-Controller*

Resumo

JÚNIOR, E. L. A. *University Bus: Aplicativo android para auxiliar no controle do transporte universitário*. Monografia (Graduação) — Centro Universitário de Formiga – Unifor-MG – Formiga, 2021.

Este projeto foi idealizado primeiramente em razão das necessidades encontradas em relação a organização e comunicação entre estudantes e motoristas, no âmbito do transporte universitário. E também devido ao evidente crescimento tecnológico, e ao constante crescimento da utilização dos smartphones por meio das pessoas. Assim sendo, o objetivo deste projeto é desenvolver um aplicativo, que promova um ambiente onde será possível pelos estudantes, marcarem diariamente a sua frequência, e possibilitando a todos participantes, sendo estes, os estudantes e o motorista, a visualizarem tais indicações. Também será possível algumas marcações por meio do motorista, sendo elas: a saída da cidade origem, e no cenário de volta, o alerta para retornar e a saída da universidade. Contudo, este trabalho visa desta maneira, contribuir em uma melhor organização do transporte universitário e a facilitar a comunicação entre os motoristas e estudantes.

Palavras-chave: Transporte universitário, Migração pendular, Aplicação Android, Firebase

Abstract

JÚNIOR, E. L. A. *University Bus: Android application to help control university transport*. Monography (University Graduate) — Centro Universitário de Formiga – Unifor-MG – Formiga-MG, 2021.

This project was conceived primarily because of the needs found in relation to organization and communication between students and drivers, in the context of university transport. And also due to the evident technological growth, and the constant growth in the use of smartphones by people. Therefore, the objective of this project is to develop an application that promotes an environment where it will be possible for students to mark their frequency daily, and enabling all participants, being these, the students and the driver, to view such indications. It will also be possible to make some appointments through the driver, namely: leaving the city of origin, and in the back scenario, the warning to return and leaving the university. However, this work aims in this way, to contribute to a better organization of university transport and to facilitate communication between drivers and students.

Keywords: University Transport, Commuting, Android Application, Firebase

Introdução

Nas últimas décadas, o Brasil assistiu a um notável processo de crescimento do número de estudantes no ensino superior. No começo dos anos noventa do século passado, somavam-se 1.540.080 estudantes matriculados no ensino superior no Brasil (NEVES, 2012). Esse número saltou para 2.694.245 de estudantes em 2000 e para 6.379.299 em 2011. Outro estudo revela que até o ano de 2018, este número aumentou para 8,4 milhões. Já o total de concluintes do ano de 2018 subiu 5,1% em relação ao ano de 2017, passando de 1.199.769 para 1.264.288 (ABRES, 2019).

Para prática desses estudos, 7,4 milhões de pessoas locomovem-se de sua cidade para a cidade vizinha. Este deslocamento é bem comum também no âmbito estudantil e pode-se dizer que um dos fatores que propiciam essa migração pendular, ou seja, esse movimento diário destes estudantes para a prática de seus estudos, se deve ao fato deles não poderem continuar estudando em seus municípios de origem residencial e pela falta de oportunidade se movimentam para cidades vizinhas buscando uma formação acadêmica superior ou preparação profissional não estabelecendo residência no lugar escolhido (FRANCELLINO, 2020).

Assim sendo, os universitários recorrem a um meio de transporte para se locomover, porém este é um fator que gera conflitos constantemente. Visto que, de cidade a cidade há uma variação na organização destes meios de transportes, o que propicia então adversidades, bem como problemas de comunicação, como a falta de padronização de horários e controle de frequência dos universitários.

Contudo, como em determinadas áreas do meio de transporte a utilização de aplicativos de serviços está cada vez mais presente no dia a dia da sociedade. Como por exemplo significativo são os aplicativos de transporte remunerado de passageiros, como o Uber e vários outros em todo o mundo, o qual vem se tornando cada vez mais essencial em certas cidades e em determinados horários (ROCHA, 2018).

Uma alternativa que contribuiria para a solução do problema sobre a migração pendular estudantil seria o desenvolvimento de um aplicativo que auxiliasse com um ambiente de comunicação entre passageiros e motoristas, para promover e facilitar a troca de informações entre os mesmos. Visando assim, resolver os pontos de conflito como foi visto anteriormente, possibilitando então evitar possíveis adversidades e transtornos entre ambas as partes.

1.1 Objetivos do Trabalho

Este trabalho tem como objetivo geral, desenvolver um aplicativo móvel, para auxiliar na organização e comunicação entre estudantes e motoristas do ensino superior.

1.1.1 Objetivos Específicos

Os objetivos específicos deste trabalho são conceder acesso a um aplicativo móvel que permita realizar o cadastro de usuários, subdivididos entre estudantes e motoristas, onde estes, terão a possibilidade de cadastrar uma nova turma, ou escolher participar de alguma turma já existente.

Os estudantes terão possibilidade de marcar sua frequência e situação diária, tanto no cenário de partida da cidade origem, quanto na saída da universidade.

Já os motoristas terão acesso ao controle de frequência do momento atual dos estudantes, para saber se a turma está preparada e pronta para a partida. Ele também tem acesso para marcar a saída tanto da cidade origem, quanto a saída da universidade, no cenário da volta a cidade origem.

1.2 Justificativa

A migração pendular se refere aos deslocamentos diários dos indivíduos para realizar ações de sua vida cotidiana como: trabalhar, estudar, lazer entre outros.

Fatores que podem ser considerados como fundamentais para a realização dos deslocamentos da população são o avanço do meio de transporte e a disseminação do uso do automóvel, bem como a descentralização das atividades econômicas, o crescimento dos centros urbanos e a ocupação, expansão e adensamento populacional das áreas periféricas. Estes favoreceram o crescimento considerável das cidades e como consequência houve uma multiplicação dos fluxos, cada vez mais longos e complicados que ultrapassavam os limites municipais (OLIVEIRA, 2011).

Entretanto, é notório dizer que a migração pendular para os estudos no ensino superior vem seguindo em constante crescimento. Visto que, de acordo com (FRANCELLINO, 2020) em todo o Brasil, 7,4 milhões de pessoas se deslocam para cidades vizinhas de onde moram para trabalhar ou estudar, correspondendo a 6,7% da população que estuda e/ou trabalha.

Essas informações são baseadas no Censo de 2010, sobre concentrações urbanas e arranjos populacionais. Esse percentual sobe para 10,6% quando se considera as regiões metropolitanas, pois de um modo geral, os fluxos associados a esse tipo de deslocamentos se concentram nas principais aglomerações urbanas do país (FRANCELLINO, 2020).

Diante disso, e levando em conta que a utilização do uso da internet está crescendo constantemente no Brasil, podemos observar que em 2019, 82,7% dos domicílios brasileiros possuíam

serviço de internet, sendo este uso por pessoas de 10 anos ou mais de idade. A utilização da Internet subiu então de 74,7%, em 2018, para 78,3%, em 2019 (IBGE, 2019).

No entanto, destaca-se que esta utilização, foi de 98,1% sendo através de aparelhos celulares no ano de 2018, aumentando para 98,6% no ano de 2019 (IBGE, 2019).

Portanto, de posse do levantamento do índice de pessoas que fazem migração pendular para estudar, e do crescimento constante da utilização da internet no Brasil através dos aparelhos celulares, pode-se enaltecer a implementação do aplicativo que será desenvolvido para o controle do transporte estudantil das universidades.

1.3 Estrutura da Monografia

A estrutura deste projeto se dispõe da seguinte forma: No Capítulo 1 foi descrito a introdução, juntamente com os objetivos do trabalho. O Capítulo 2 apresenta o referencial teórico, com as tecnologias e conceitos utilizados para o desenvolvimento do aplicativo. No Capítulo 3 é apresentado o estudo da arte, onde se mostra alguns trabalhos relacionados a este. No Capítulo 4 é demonstrado o desenvolvimento e as metodologias que foram utilizadas. Já no Capítulo 5 são mostrados os resultados, por fim, no Capítulo 6 é apresentada a conclusão do trabalho, juntamente com os trabalhos futuros.

Referencial Teórico

Neste capítulo serão apresentados os conceitos, ferramentas e tecnologias que foram utilizadas para o planejamento e desenvolvimento deste projeto.

2.1 Computação Urbana

A computação urbana é um campo no qual as ciências da computação encontram quesitos que tem relações à cidade, como por exemplo transporte, engenharia civil, meio ambiente, economia, ecologia e sociologia no contexto dos espaços urbanos, e auxilia a entender o comportamento humano nas cidades (REIS; DUARTE; PONCIANO, 2018). Diante disso, temos que o presente trabalho tem relações com a computação urbana, visto que ele envolve questões como a mobilidade urbana, e o transporte público estudantil.

Outro fator importante é que a crescente urbanização resulta no aumento da mobilidade urbana, fazendo assim com que seu grau de complexidade se eleve. E ainda sobre esta questão temos que apenas alternativas que incluem o planejamento urbano e de transportes seriam eficazes para reduzir problemas ambientais e aumentar a qualidade e eficiências dos transportes urbanos (REIS; DUARTE; PONCIANO, 2018).

Contudo, com a computação urbana, seria possível suavizar alguns destes problemas, sendo um deles o meio de transporte, que com o auxílio da tecnologia, seria possível contribuir em uma melhor organização e comunicação entre os indivíduos. Sendo esta realizada através de um aplicativo, desenvolvido para o sistema operacional mais utilizado para dispositivos móveis, que é o *Android*.

2.2 Android

O *Android* é um sistema operacional para dispositivos móveis mais utilizado no mundo, com mais de dois bilhões de dispositivos ativos, principalmente em *smartphones* e *tablets*. Entretanto

o *Android* está também presente em outros dispositivos como: automóveis, televisores, relógios, entre outros (GLAUBER, 2019).

No entanto, pode-se dizer que o *Android* não é apenas um sistema operacional, mas sim um conjunto completo de *software* para dispositivos móveis que inclui: um sistema operacional que tem como base o *kernel* do *Linux*, que é o responsável pelo gerenciamento de processos, *drivers*, memória e energia. Tem também um *middleware*, que por sua vez, controla a interação entre os aplicativos instalados no aparelho, facilitando então a comunicação entre eles. E as aplicações-chave que são programas comuns, como discador, navegador, contatos, mensagens, dentre outros (GLAUBER, 2019)

Referindo sobre o desenvolvimento em *Android*, se faz necessário a utilização do *Android SDK (Software Development Kit)*, que é um software utilizado para o desenvolvimento das aplicações *Android*. Contendo neste, um emulador para simular o celular, ferramentas utilitárias e uma *API (Application Programming Interface)* completa com todas as classes necessárias para desenvolver as aplicações, para a linguagem *Java*, que é a linguagem de programação padrão para desenvolvimento (LECHETA, 2013).

Contudo, para um melhor desempenho e execução, é aconselhável a utilização da plataforma do *Android Studio*, que é o *IDE (Integrated Development Environment)* padrão para desenvolvimento de aplicações *Android*. Entretanto já é incluso nele o *Android SDK*, que foi citado anteriormente, bem como também o *Open JDK (Java Development Kit)* que possui as bibliotecas e ferramentas do *Java*, para serem utilizadas no desenvolvimento das aplicações (GLAUBER, 2019).

2.3 SQLite

É um banco de dados *Open Source*, que é utilizado no *Android*, suportando os padrões dos bancos de dados relacionais, bem como também a sintaxe *SQL (Structured Query Language)*. O *SQLite* é simples de ser utilizado, pois não requer nenhuma configuração inicial, é preciso somente descrever a instrução *SQL* para gerar o banco de dados e assim ele é automaticamente criado (VOGEL, 2011).

Sobre alguns tipos de dados que o *SQLite* suporta, temos: *Text*, sendo este similar ao tipo *String* em *Java*, o *Integer*, que é semelhante ao *Long* em *Java*, e o *Real*, que em *Java* se assemelha ao tipo *Double*. Para armazenamento de outros tipos de dados que não forem estes apresentados, os mesmos terão que ser convertidos para algum destes (VOGEL, 2011).

Já abordando sobre a maneira como uma tabela é criada, temos o comando *CREATE TABLE* da linguagem *SQL*. Sobre a parte de manipulação dos dados esta é realizada através dos comandos: *INSERT*, *UPDATE* e *DELETE*, e para uso de consultas são realizadas com o comando *SELECT* (CORDEIRO, 2017).

Contudo, visto que o *SQLite* é um banco de dados local, e que para atender necessidades como realizar comunicação através de dados armazenados em nuvem, se faz necessário a utilização do banco de dados *Realtime Database, do Firebase*.

2.4 Firebase

É uma plataforma do *Google* de alta performance para armazenamento e sincronização dos dados na nuvem, e em tempo real. Contribuindo assim para a criação de grandes aplicativos, possibilitando a eles crescimento e ganho em seus projetos e negócios (SILVA, 2018).

Para SILVA (2018), o que diferencia o Firebase das demais bases de dados é sua infraestrutura, onde cada base de dados é armazenada no estilo de árvore de objeto *JSON (JavaScript Object Notation)*, proporcionando desta maneira flexibilidade para todos os tipos de dados.

Dentre tantos serviços disponíveis do *Google Firebase*, abordando sobre seu serviço de banco de dados, temos o *Realtime Database*, sendo ele baseado em *NoSQL* e hospedado na nuvem. Seus dados são armazenados como *JSON* (SILVA, 2018) como havia sido citado anteriormente, sendo estes sincronizados em tempo real com todos seus usuários, destaca-se também o fato de mesmo quando não há conexão de rede através de um cache local, ele ainda continua disponível.

No entanto, outro quesito importante, é sobre como são suas definições para estruturar os dados, e a forma com que os usuários interagem com o banco, tendo estes, permissões personalizadas para determinadas funções. Para isso, são utilizadas algumas regras de linguagens baseadas em expressão, sendo chamadas de *Firebase Realtime Database Security Rules* (SILVA, 2018).

Estudo da Arte

3.1 Considerações iniciais

Este capítulo aborda sobre trabalhos que relacionam-se com as tecnologias e preceitos deste projeto.

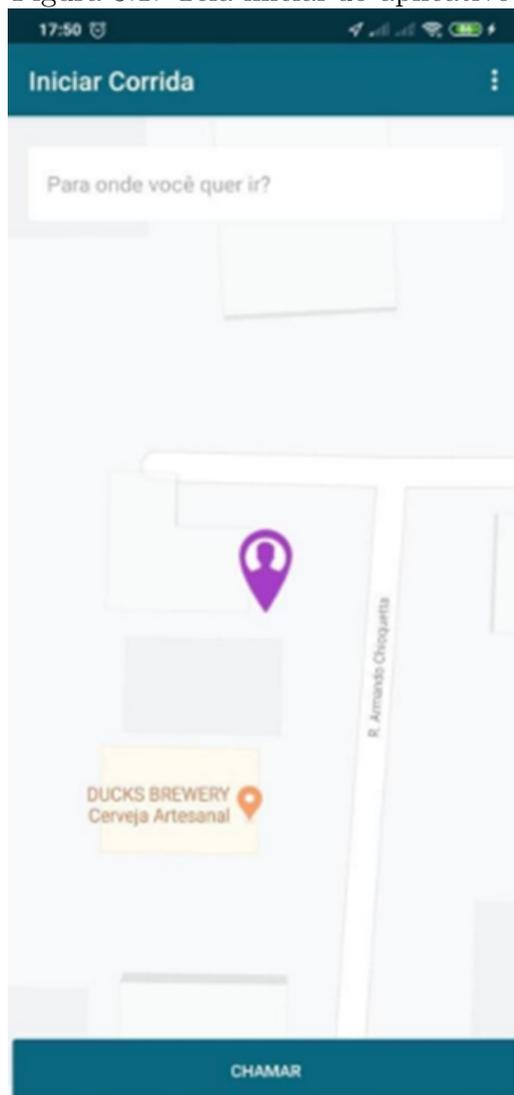
3.2 Aplicativo para atender requisições de deslocamento solicitadas por passageiros

Este aplicativo *Android* foi desenvolvido visando facilitar a busca por transporte, permitindo ao passageiro fazer requisições solicitando transporte para determinado local. Sendo assim os motoristas que estiverem disponíveis recebem uma requisição e o primeiro que aceitar fica responsável por ir ao local do passageiro, buscá-lo e levá-lo ao seu destino.

Diante disso, temos que os usuários que possuem acesso ao sistema pertencem a duas categorias: passageiros ou motoristas. Sendo o passageiro o criador das corridas, e o motorista sendo o responsável por realizá-las (SANTOS, 2020).

Na Figura 3.1 é possível ver a tela inicial do aplicativo em questão, que para seu desenvolvimento foram utilizadas algumas tecnologias, dentre elas, a linguagem *Java* e *Android*, e também o uso do banco de dados *Firebase Realtime Database*.

Figura 3.1: Tela inicial do aplicativo



Fonte : SANTOS (2020)

Por fim, os objetivos propostos foram realizados e o aplicativo foi desenvolvido. Contudo, foi permitindo então desta maneira o controle de requisições das corridas por parte do motorista, da solicitação pelo passageiro até a finalização da corrida (SANTOS, 2020).

3.3 Blumenbus: aplicativo móvel para consulta de informações do transporte público de Blumenau

O sistema de transporte público brasileiro sempre recebeu, e continua recebendo um grande número de reclamações devido à baixa qualidade do serviço prestado. Portanto, diante deste cenário, este trabalho apresenta o desenvolvimento de um aplicativo que auxilie os usuários do transporte coletivo de Blumenau a realizarem a consulta de informações referentes ao serviço, sendo estas: linhas de ônibus disponíveis, seus horários de partida, pontos de saída e trajeto realizado.

Conforme mostrado na Figura 3.2, nela é possível visualizar uma listagem das linhas favoritas, e na Figura 3.3 são apresentadas todas as linhas de forma geral. O aplicativo, foi

desenvolvido com as seguintes tecnologias: utilização da biblioteca *Ionic* sobre a plataforma *Cordova*, na linguagem de programação *JavaScript* através do *superset TypeScript* e *framework Angular*, e com a utilização do serviço de *backend* do *Firebase*, utilizando banco de dados não relacional (KOEPESEL, 2018).

Figura 3.2: Tela dos favoritos



Fonte : KOEPESEL (2018)

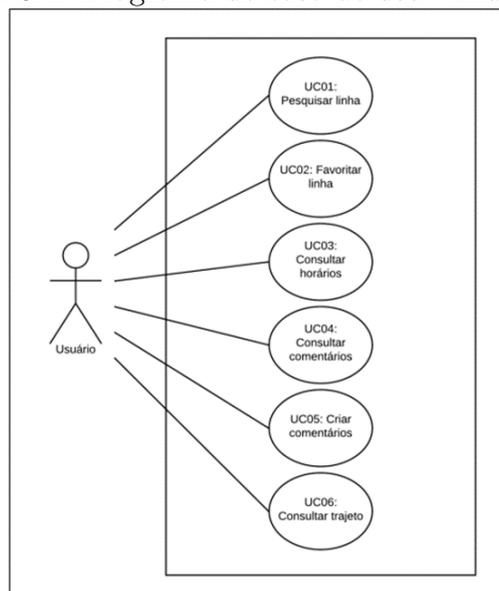
Figura 3.3: Tela das Linhas



Fonte : KOEPESEL (2018)

Conforme é apresentado na Figura 3.4, os casos de uso deste aplicativo são: pesquisar linha, favoritar linha, consultar horários, consultar comentários, criar comentários e consultar trajeto. Sendo cada um destes, funcionalidades que o aplicativo fornece ao usuário.

Figura 3.4: Diagrama de caso de uso - Blumenbus



Fonte : KOEPESEL (2018)

Contudo, pode-se afirmar que o objetivo deste trabalho foi alcançado, pois ao utilizar o aplicativo desenvolvido é possível consultar informações referentes às linhas de ônibus do trans-

porte público de Blumenau/SC, previamente cadastradas na base de dados, foi implementado também uma interface para colaboração entre os usuários do aplicativo, para assim ter a possibilidade dos mesmos poderem realizar comentários sobre determinada linha de ônibus e também consultar comentários realizados por outros usuários da aplicação (KOEPSEL, 2018).

3.4 Aplicativo de gestão de transporte rodoviário

Em um mundo globalizado, é cada vez maior a necessidade de ampliar o atendimento e reduzir os custos. Nesse contexto, as empresas buscam implementar inovações para que possam manter sua competitividade. Na medida em que a falta de comunicação na relação entre o caminhoneiro e a empresa constitui um problema no âmbito de uma empresa que possui sede no município, foi desenvolvido então este aplicativo programado em Java, para gerenciar o transporte de cargas da empresa (SALATINO et al., 2019).

Para o desenvolvimento foram realizadas pesquisas bibliográficas através de sites, livros e entrevistas com empresas e motoristas da região para compreender as principais dificuldades que as empresas enfrentam no âmbito do gerenciamento do transporte de cargas e desenvolver o conteúdo do aplicativo, que logo foi programado no *Android Studio*, um *software* que desenvolve aplicações para *Android* por meio da linguagem de programação *Java*, juntamente com o *Firebase* o qual foi necessário para atribuir o banco de dados ao protótipo (SALATINO et al., 2019).

Sobre as funcionalidades do sistema, temos que para cada tipo de usuário, seja ele motorista ou o gerente da empresa, é realizado um cadastro do mesmo. Abordando as funcionalidades que o gerente tem no aplicativo, são essas: Escolher o motorista para viagem; escolher carga, data de carregamento, chegada, peso, local do carregamento e entrega; excluir ou adicionar motoristas ao cadastro; acompanhar onde está o motorista; acessar o relatório que o motorista envia, se necessário; ver todas as viagens que ocorreram e estão ocorrendo; ter acesso a lista de gastos por viagem e gastos gerais. Já falando das funcionalidades da parte do motorista, são: Receber avisos com todas as informações sobre a carga que deve transportar; poder alterar os seus dados, bem como do caminhão que utiliza; o trajeto que deve seguir; um campo para relatar todos os problemas com a viagem; espaço para relatar gastos com diesel e outros eventuais custos (SALATINO et al., 2019).

Conclui-se então que este projeto desenvolveu um aplicativo com funcionalidades tanto para o motorista quanto para o gerente. A utilização deste aplicativo pelos motoristas e empresários permite que sejam evitadas falhas de comunicação entre os mesmos, permitindo também um controle mais amplo do trajeto realizado pelo motorista, a prevenção de pequenos problemas e consequentemente a economia de tempo nas viagens o que implica numa maior circulação das cargas (SALATINO et al., 2019).

3.5 Considerações finais

Diante dos trabalhos que foram apresentados, foi possível perceber algumas necessidades existentes, em relação a organização sobre os meios de transporte. E assim como foi desen-

volvido nos aplicativos apresentados, este presente trabalho também auxilia na comunicação e organização entre os usuários, porém este com foco no âmbito do transporte universitário.

Metodologia e Desenvolvimento

Neste capítulo serão apresentadas as etapas que foram seguidas para o desenvolvimento deste trabalho, sendo estas: levantamento de requisitos, diagrama de caso de uso, arquitetura do aplicativo e a criação do banco de dados.

4.1 Levantamento de requisitos

Para uma melhor definição das funcionalidades do *software*, foram denotados alguns requisitos, que serão discorridos abaixo:

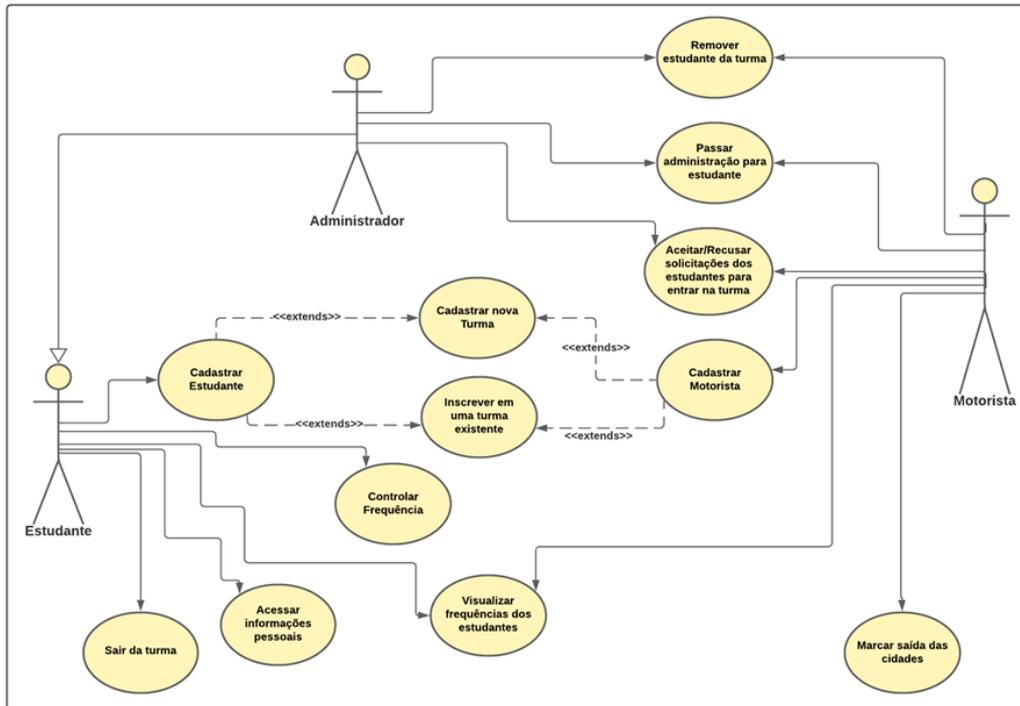
- Cadastro de estudantes: Realizar cadastro do nome, celular, cidade, curso e período.
- Cadastro de motoristas: Realizar cadastro do nome, celular e cidade.
- Cadastro de turma: Possibilitar os usuários cadastrar uma turma, com o nome, cidade e quantidade de vagas.
- Busca de turmas existentes: Proporcionar os usuários encontrar uma turma já existente e solicitar permissão para participar.
- Controle de frequência: Permitir com que os estudantes indiquem sua frequência, e o motorista visualize.
- Informações do perfil do usuário: Exibir as informações pessoais do usuário, e também possibilitar sua saída da turma.

4.2 Diagrama de caso de uso

A função dos diagramas de caso de uso é auxiliar na identificação das funcionalidades de um sistema, e visualização dos relacionamentos entre os atores e os casos de uso do *software* (SANTOS et al., 2016).

Diante disso, conforme demonstrado na Figura 4.1, foi produzido um diagrama de caso de uso, para contribuir na organização e modelagem dos comportamentos do *software*.

Figura 4.1: Diagrama de caso de uso - University Bus



Fonte : Autoria própria

Ainda conforme a Figura 4.1, é possível perceber que o aplicativo possuirá três atores: estudante, administrador e motorista, no qual cada um tem suas determinadas atividades. Sobre o ator estudante, este é capaz de realizar primeiramente o seu cadastro, cadastrar uma nova turma ou se inscrever em alguma já existente. Ele também consegue controlar sua frequência diária e visualizá-la, acessar os dados pessoais, e sair da turma vinculada.

Sobre o administrador, este é uma generalização do ator estudante, estendendo todas as suas atividades, porém com mais algumas adicionais, como: remover um estudante da turma, passar a administração para outro estudante e aceitar ou recusar solicitações para participar da turma.

Já o ator motorista, possui capacidade de realizar também o seu cadastro e vincular a uma turma, podendo escolher entre cadastrar uma nova turma, ou participar de alguma existente, ele também possui acesso as frequências dos estudantes e poderá indicar quando estiver saindo das cidades, no cenário de ida e de volta. Além de possuir algumas outras atividades em comum à um administrador, sendo estas: remover um estudante da turma, passar a administração da turma para um estudante e também aceitar ou recusar as solicitações dos estudantes, para participar de sua determinada turma.

4.3 Arquitetura do aplicativo

Adaptado aos recursos nativos do *Android*, o *software* foi desenvolvido com a implementação do modelo *MVC* (*Model-View-Controller*), que neste trabalho foi dividido em pastas, organizando os *models* em uma pasta chamada “model”, as *views* na pasta “layout” e os *controllers* na pasta “controller”.

4.3.1 Camada Model

A camada model é designada a gerenciar tarefas relacionadas aos dados, como: validação, estado de sessão e controle e estrutura de fonte de dados. Com a utilização deste modelo é possível notar uma considerável redução em relação a complexidade do código a ser desenvolvido (ROCHA, 2018).

4.3.2 Camada View

Esta camada é responsável pelo gerenciamento de todos os elementos gráficos da interface do usuário, auxiliando o desenvolvedor a ter uma melhor organização quando for necessário fazer alterações e mudanças dos elementos gráficos (ROCHA, 2018).

Neste projeto, a arquitetura do modelo *view* foi adaptada às classes com comportamento de *activity*, sendo estas as responsáveis pela exibição dos dados na interface gráfica.

4.3.3 Camada Controller

Nesta camada, sua tarefa é manusear eventos, podendo estes ser acionados por um usuário que interage com o *software*, ou mesmo por um processo do sistema. Sendo sua função definida por preparar os dados para uma resposta e estabelecer o formato dessa resposta, para logo gerar a visualização (ROCHA, 2018).

Na arquitetura do presente trabalho, o modelo *controller* foi implementado nas classes que são responsáveis pela busca e processamento dos dados a serem exibidos nas *views*, onde cada classe faz referência à sua respectiva *activity*.

4.4 Criação do Banco de dados

Para este projeto foi utilizado o *Firebase Realtime Database*, que é um banco de dados hospedado na nuvem, onde seus dados são gravados com o formato *JSON (JavaScript Object Notation)*, permitindo assim serem disponibilizados de forma sincronizada em tempo real com os usuários conectados a ele (FIREBASE, 2021).

Na Figura 4.2 é apresentada a estrutura geral dos nós que compõem o banco de dados deste projeto.

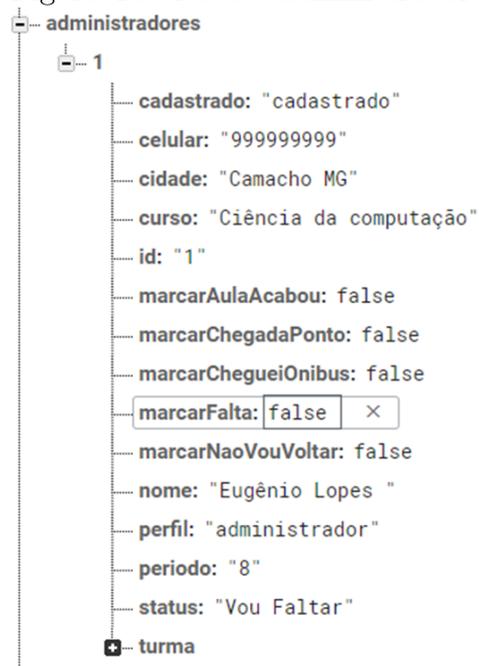
Figura 4.2: Estrutura geral do banco de dados



Fonte : Autoria própria

Conforme a Figura 4.3, será mostrado o nó administradores, que guarda todas as informações dos administradores do *software*.

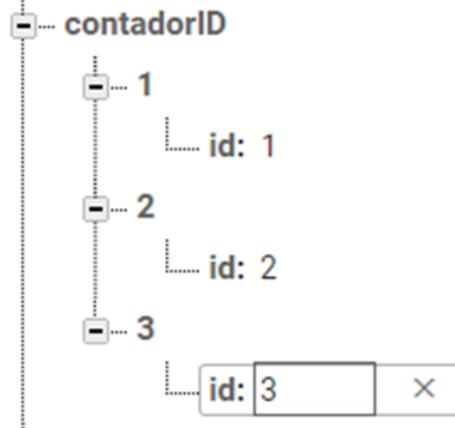
Figura 4.3: Nó dos administradores



Fonte : Autoria própria

Já na Figura 4.4, será apresentado o nó que é usado para controle do identificador (ID), dos usuários cadastrados.

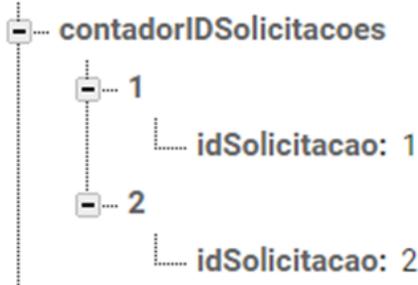
Figura 4.4: Nó contador do identificador dos usuários



Fonte : Autoria própria

Em sequência, na Figura 4.5 será mostrado o nó que faz armazenamento para controle do contador das solicitações enviadas.

Figura 4.5: Nó contador das solicitações



Fonte : Autoria própria

Como nó destinado ao controle e armazenamento do identificador das turmas, acompanhe na Figura 4.6

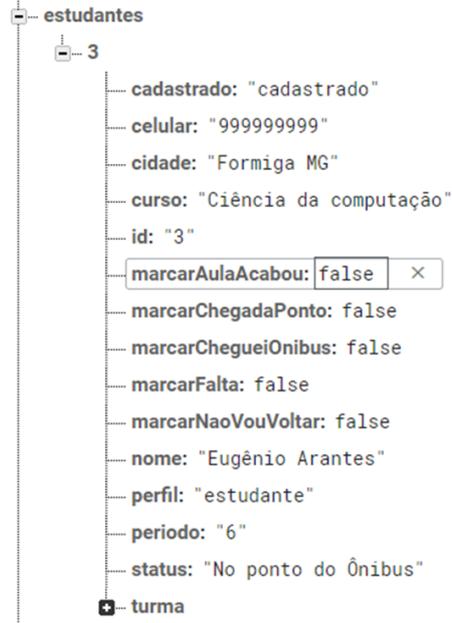
Figura 4.6: Nó contador das turmas



Fonte : Autoria própria

Na Figura 4.7 será evidenciado o nó que armazena todas informações dos estudantes.

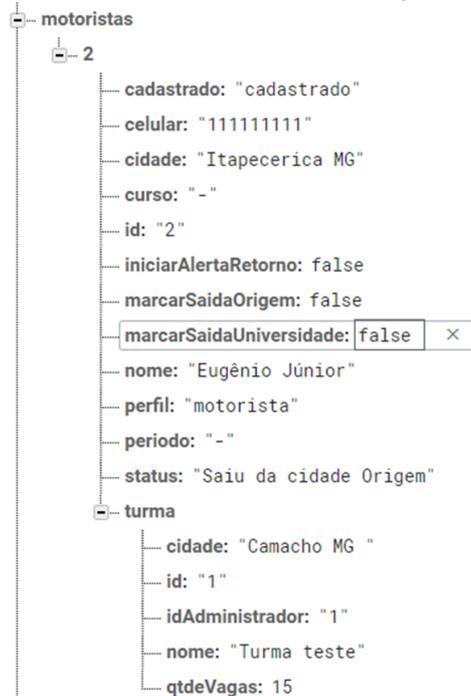
Figura 4.7: Nó que armazena as informações dos estudantes



Fonte : Autoria própria

Na Figura 4.8 é apresentado o nó que traz consigo as informações dos motoristas do aplicativo.

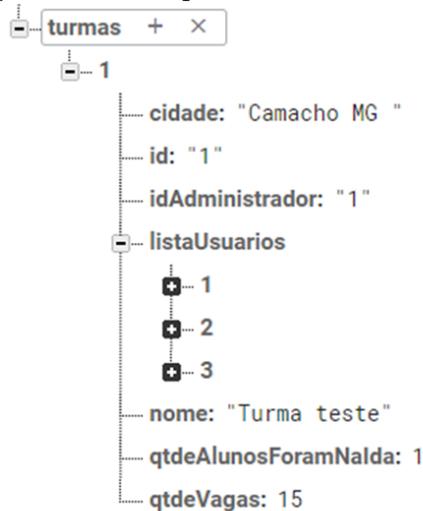
Figura 4.8: Nó que armazena as informações dos motoristas



Fonte : Autoria própria

Por fim, o nó turmas, guarda todas informações da turma cadastrada, conforme é mostrado na Figura 4.9.

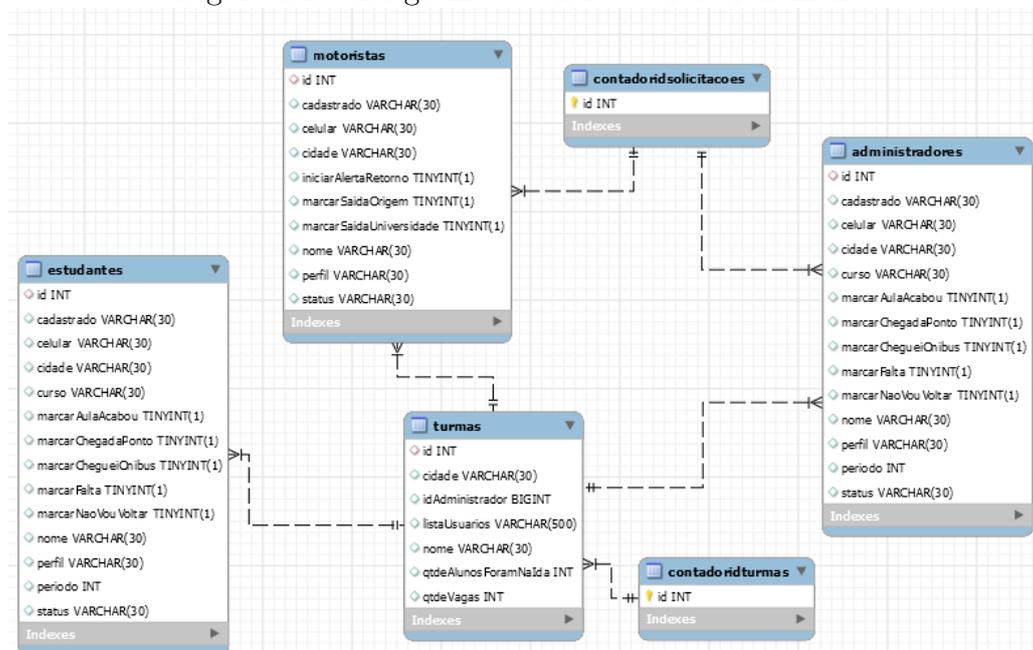
Figura 4.9: Nó que armazena as turmas



Fonte : Autoria própria

Para a modelagem e criação deste banco de dados, utilizou-se o modelo de entidade relacionamento. Conforme mostrado na Figura 4.10

Figura 4.10: Diagrama de entidade relacionamento



Fonte : Autoria própria

Resultados

Neste capítulo serão apresentados os resultados obtidos ao final do desenvolvimento do aplicativo University Bus, detalhando o seu funcionamento.

5.1 Ícone do Aplicativo

O ícone do aplicativo foi reproduzido com o aspecto de um ônibus, para assim transmitir a ideia geral do *software*, como mostrado na Figura 5.1

Figura 5.1: Ícone do University Bus



Fonte : Autoria própria

5.2 Tela Inicial

Na Figura 5.2 é apresentado a tela inicial do aplicativo onde ele terá duas opções, escolher o perfil de motorista ou de estudante.

Figura 5.2: Tela inicial do aplicativo



Fonte : Autoria própria

5.3 Cadastro de usuário

Após o usuário selecionar qual perfil irá seguir, o mesmo será redirecionado a tela de cadastro de usuário.

5.3.1 Tela Cadastrar Motorista

Visto que o usuário selecionou o perfil de motorista, lhe será apresentado a tela com os campos necessários para o seu cadastro, conforme mostrado na Figura 5.3

Figura 5.3: Tela de cadastro do motorista

The screenshot shows a mobile application interface for driver registration. At the top, there is a dark blue header with the title 'CADASTRO MOTORISTA'. Below the header, the form is organized into several sections. The first section is labeled 'Nome:' and contains the text 'Eugênio'. The second section is labeled 'Celular:' and contains the text '111111111'. The third section is labeled 'Cidade:' and contains the text 'Camacho MG'. At the bottom of the form, there is a dark blue button with the text 'CADASTRAR'. The status bar at the top of the phone shows a battery level of 66% and the time 22:08.

Fonte : Autoria própria

5.3.2 Tela Cadastrar Estudante

Se o usuário selecionou o perfil de estudante, será então apresentado a tela para o cadastro de estudante, com os seus respectivos campos a serem cadastrados, conforme apresentado na Figura 5.4

Figura 5.4: Tela de cadastro do estudante

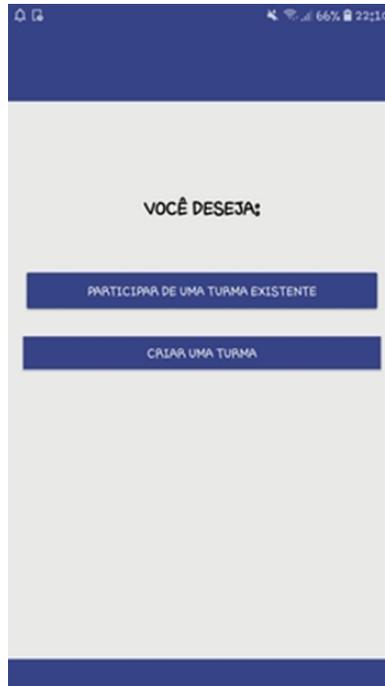
The screenshot shows a mobile application interface for student registration. At the top, there is a dark blue header with the title 'CADASTRO ESTUDANTE'. Below the header, the form is organized into several sections. The first section is labeled 'Nome:' and contains the text 'Eugênio'. The second section is labeled 'Celular:' and contains the text '999999999'. The third section is labeled 'Cidade:' and contains the text 'Formiga MG'. The fourth section is labeled 'Curso:' and contains the text 'Ciência da computação'. The fifth section is labeled 'Período:' and contains the text '8'. At the bottom of the form, there is a dark blue button with the text 'CADASTRAR'. The status bar at the top of the phone shows a battery level of 67% and the time 22:07.

Fonte : Autoria própria

5.4 Vincular à uma turma

Na Figura 5.5 é apresentada a tela onde o usuário terá que escolher entre as opções de: participar de uma turma já existente, ou a opção de criar uma nova turma.

Figura 5.5: Tela Vincular à uma turma



Fonte : Autoria própria

5.4.1 Tela Procurar turmas existentes

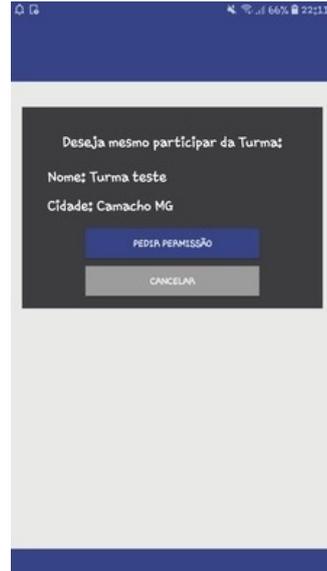
Conforme a Figura 5.6, dado que o usuário selecionou a opção de participar de uma turma existente, o mesmo será redirecionado para a tela onde buscará através de um código a sua turma desejada. O código a ser inserido para a busca da turma, poderá ser obtido através de sua disponibilidade pelo administrador, motorista, ou algum participante já cadastrado desta determinada turma.

Figura 5.6: Tela de procurar turmas existentes



Fonte : Autoria própria

Figura 5.7: Tela de confirmar o pedido



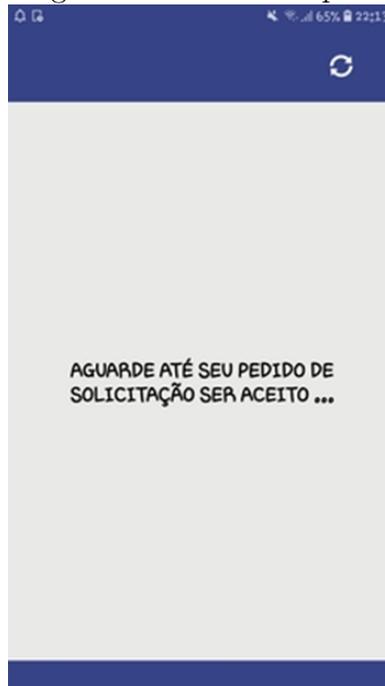
Fonte : Autoria própria

De posse do código, é possível então enviar uma solicitação pedindo permissão para participar, ao administrador ou motorista da turma em questão, conforme a Figura 5.7.

5.4.2 Tela de espera

Após o envio da solicitação de permissão, o usuário será redirecionado para a tela de espera, onde ele ficará à espera de uma resposta, de aprovação ou negação para participar da turma solicitada. Para atualizar a página, e verificar se houve resposta em relação a solicitação, é utilizado o botão no canto superior direito, conforme mostrado na Figura 5.8.

Figura 5.8: Tela de espera



Fonte : Autoria própria

5.4.3 Tela criar nova turma

Levando em conta que o usuário selecionou a opção de criar uma turma, logo ele será redirecionado a tela com os respectivos campos necessários para realizar o cadastro da nova turma, como é mostrado na Figura 5.9. E após o cadastro, se o perfil deste usuário for de um estudante, o mesmo será atribuído como administrador desta turma.

Figura 5.9: Tela cadastrar nova turma

Nome da Turma:
Turma teste

Cidade:
Camacho MG

Vagas:
15

PRÓXIMO

Fonte : Autoria própria

5.5 Tela principal

Após o usuário ter realizado o cadastro de uma nova turma, ou ter sua solicitação aprovada para participar de uma turma existente, o mesmo será redirecionado à tela principal do aplicativo.

Onde nela, o usuário terá a possibilidade de visualizar o seu status atual, bem como também acessar algumas informações sobre a viagem, como: ida, volta e a listagem diária, que se encontram nos seus respectivos botões apresentados na Figura 5.10 e na Figura 5.11. O usuário também terá acesso aos seus dados pessoais no botão encontrado no canto superior esquerdo, e poderá atualizar a página, utilizando o botão de cor branca no canto superior direito.

Na Figura 5.10 é retratada uma tela de um perfil de estudante, já na Figura 5.11 sendo a tela de um perfil de motorista ou administrador, onde nela se encontra um botão de um sino amarelo, no canto superior direito da tela, com função de redirecionar para a tela onde serão apresentadas as notificações recebidas dos usuários que solicitarem participação na turma.

Figura 5.10: Tela principal do estudante



Fonte : Autoria própria

Figura 5.11: Tela principal do motorista



Fonte : Autoria própria

5.6 Tela de ida

É na tela de ida, que serão tratados os dados e informações referentes ao deslocamento inicial dos estudantes e do motorista, com destino a universidade.

5.6.1 Tela de ida do motorista

Nesta tela o motorista possui acesso a informações atuais, como: a listagem de alunos que irão faltar, conforme apresentado na Figura 5.13, e também a listagem dos alunos que estão no ponto de ônibus, esta mostrada na Figura 5.14. Obtendo assim então a frequência diária de sua turma, em relação a saída da cidade origem.

Ainda nesta tela, contém também um botão para o motorista marcar a sua saída da cidade, rumo a universidade, conforme é evidenciado na Figura 5.12.

Figura 5.12: Tela de ida do motorista



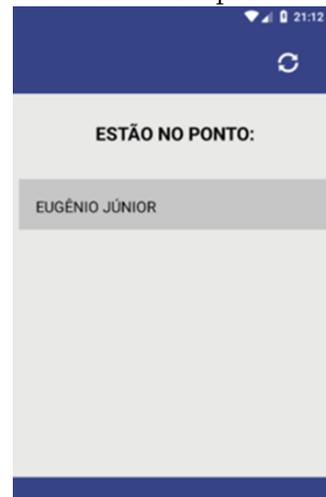
Fonte : Autoria própria

Figura 5.13: Lista de falta



Fonte : Autoria própria

Figura 5.14: Lista no ponto de ônibus



Fonte : Autoria própria

5.6.2 Tela de ida do estudante

É nesta tela onde o estudante tem a possibilidade de marcar a sua frequência diária no cenário de ida, contendo as possibilidades de: marcar falta ou marcar a chegada no ponto de ônibus.

Ainda nesta tela, conforme é possível notar na Figura 5.15, o estudante terá a possibilidade também de desmarcar a opção selecionada, bastando clicar no botão "limpar", no canto superior direito da tela.

Figura 5.15: Tela de ida do estudante



Fonte : Autoria própria

5.7 Tela de volta

Na tela de volta, serão organizadas as informações referentes a saída da universidade, com destino à cidade origem.

5.7.1 Tela de volta do motorista

Conforme apresentado na Figura 5.16, nesta tela serão apresentadas as informações e as ações que o motorista tem em relação ao cenário de volta.

Sendo as informações: uma listagem dos estudantes que já chegaram no ônibus, conforme mostrado na Figura 5.17, e também outra listagem que apresenta os estudantes que não irão voltar no ônibus, esta apresentada na Figura 5.18.

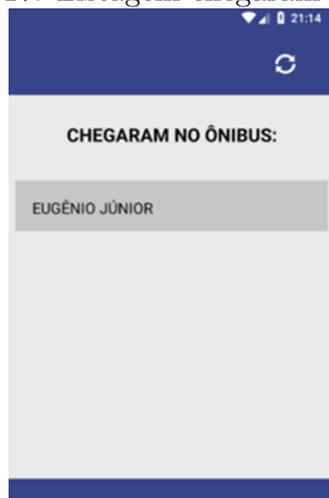
Já as ações que o motorista possui, são: marcar através do botão alerta de retorno, sua pretensão para a saída, e no outro botão marcar realmente a sua partida da universidade, rumo à cidade origem.

Figura 5.16: Tela de volta do motorista



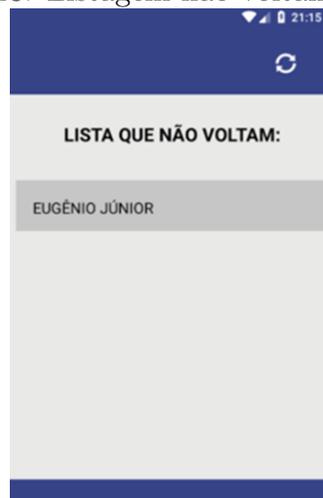
Fonte : Autoria própria

Figura 5.17: Listagem chegaram no ônibus



Fonte : Autoria própria

Figura 5.18: Listagem não voltam de ônibus



Fonte : Autoria própria

5.7.2 Tela começar nova viagem

Conforme apresentado na Figura 5.19, esta tela será apresentada ao motorista sempre quando o mesmo, marcar saída da universidade, surgindo então a necessidade de iniciar uma nova viagem.

Sendo assim, levando em conta que o motorista marcou a opção de começar uma nova viagem, logo então, os status de cada estudante desta turma serão apagados e restaurados, fazendo com que tudo fique pronto para uma nova viagem ser iniciada.

Figura 5.19: Tela começar nova viagem



Fonte : Autoria própria

5.7.3 Tela de volta do estudante

Na tela de volta do estudante, agora no cenário de volta, o mesmo possui também algumas ações, como: marcar que não irá voltar no ônibus ou marcar que sua aula terminou, e posteriormente indicar sua chegada no ônibus. Confira a seguir na Figura 5.20.

Figura 5.20: Tela de volta do estudante



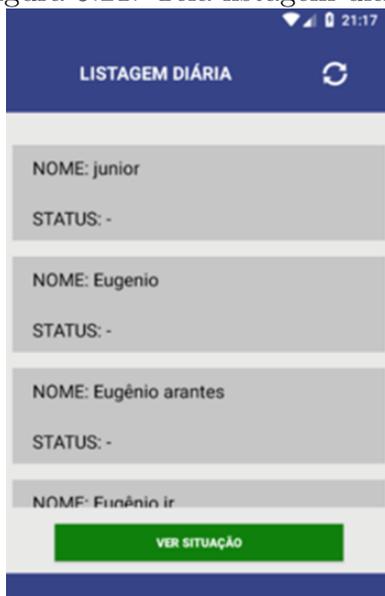
Fonte : Autoria própria

5.8 Tela listagem diária

Considerando que o usuário acessou o botão “lista diária”, encontrado na tela principal do aplicativo, o mesmo será redirecionado para a presente tela, conforme apresentada na Figura 5.21, onde nela o usuário poderá visualizar de maneira geral e atual, o status de todos os estudantes de sua turma.

Se encontra também nesta tela dois botões, sendo um deles o de atualizar os dados presentes na tela, ficando no canto superior direito, e o outro botão chamado “ver situação”, localizado na parte inferior central da tela, onde neste o usuário será redirecionado a outra tela que irá detalhar qual é a situação atual da turma.

Figura 5.21: Tela listagem diária

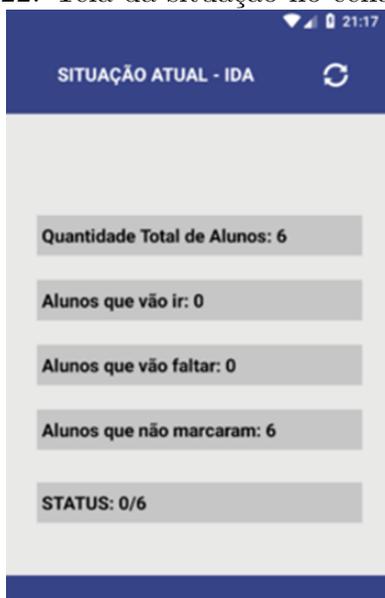


Fonte : Autoria própria

5.8.1 Tela visualizar situação no cenário de ida

Conforme apresenta na Figura 5.22, é possível nesta tela, visualizar algumas informações a respeito do cenário de ida, como: a quantidade total de estudantes que a turma tem, os estudantes que marcaram presença ou falta, e também os que ainda não marcaram nenhuma opção.

Figura 5.22: Tela da situação no cenário de ida



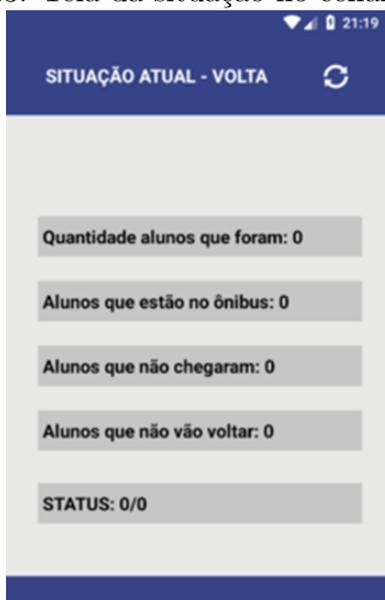
Fonte : Autoria própria

Por fim é apresentado o status, indicando quantos estudantes marcaram que estarão presentes e a quantidade total de estudantes da turma, assim então representando a situação atual da turma.

5.8.2 Tela visualizar situação no cenário de volta

Assim como mostrado na tela de situação do cenário de ida, neste presente cenário de volta, também é possível visualizar alguns dados a seu respeito, sendo estes: a quantidade de estudantes que foram para a universidade no ônibus, bem como os estudantes que já chegaram no ônibus e estão preparados para retornar à cidade origem, é apresentado também os estudantes que ainda não marcaram sua chegada no ônibus e os estudantes que marcaram que não irão voltar. Conforme apresentado na Figura 5.23.

Figura 5.23: Tela da situação no cenário de volta



Fonte : Autoria própria

Por fim, também é apresentado o status atual do cenário de volta, sendo este expressado da seguinte maneira: captura-se a quantidade de estudantes que estão na universidade, e exibe quantos destes, marcaram chegada e já estão no ônibus, prontos para o retorno.

5.9 Tela de notificações

Esta tela será visível somente para o administrador e o motorista da turma, onde nela serão exibidas as solicitações enviadas pelos usuários que pediram permissão para participar da turma em questão.

Ficará sobre o controle do administrador ou do motorista da turma, aceitar ou rejeitar então estas solicitações.

Ainda nesta tela, é apresentado um botão que está localizado no canto superior direito, contendo neste o objetivo de atualizar os dados da tela em questão. Conforme a Figura 5.24

Figura 5.24: Tela de notificações



Fonte : Autoria própria

5.10 Tela dados pessoais

De acordo com a Figura 5.25, nesta tela são apresentadas algumas informações pessoais do usuário, como os seus próprios dados cadastrais, os dados da turma em que o mesmo participa, a lista dos participantes de sua turma, e ainda a opção de sair da turma.

Figura 5.25: Tela dados pessoais



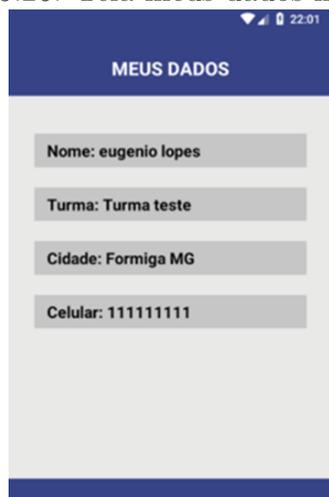
Fonte : Autoria própria

5.10.1 Tela meus dados

Ao selecionar o botão “meus dados”, o usuário é redirecionado ao painel onde será exibido os seus dados cadastrais.

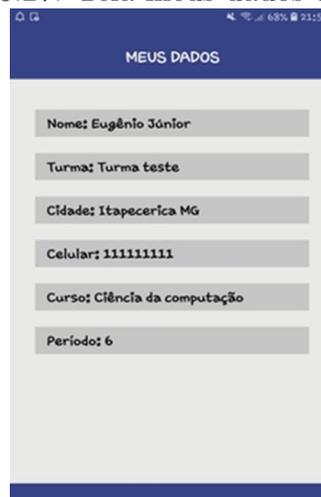
Logo, sendo identificado que o usuário é um motorista, será exibido os seus respectivos dados, conforme apresentado na Figura 5.26. De outro modo, se for identificado que o usuário é um estudante, serão apresentados seus dados conforme evidenciado na Figura 5.27.

Figura 5.26: Tela meus dados motorista



Fonte : Autoria própria

Figura 5.27: Tela meus dados estudante



Fonte : Autoria própria

5.10.2 Tela dados da turma

Nesta tela são mostradas as informações da turma, como o seu código, nome, cidade, coordenador que é o administrador da turma, o motorista e a quantidade de vagas. Conforme apresentado na Figura 5.28

Figura 5.28: Tela dados da turma



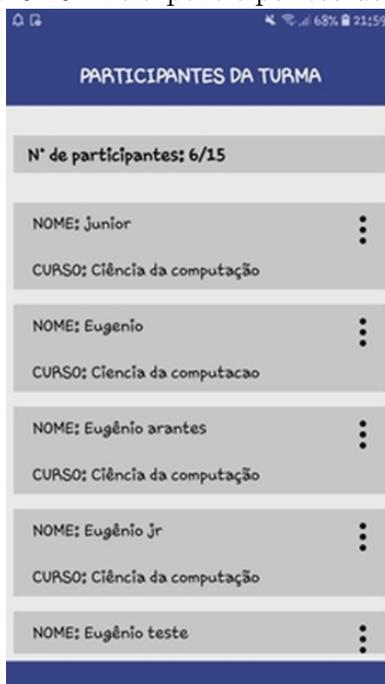
Fonte : Autoria própria

5.10.3 Tela participantes da turma

Esta tela apresenta a listagem de todos os usuários participantes da turma, onde conforme na Figura 5.29, é possível ver que para cada participante, é apresentado o seu respectivo nome e curso.

Ainda nesta tela, visto que foi identificado que o usuário for um motorista ou um administrador, será mostrado também um botão de opções, fixado no canto superior direito de cada bloco de um usuário da lista, onde neste serão apresentadas as opções de remover determinado usuário da turma, não sendo este um administrador, e também terá a opção de poder passar a coordenação e administração para este determinado usuário selecionado.

Figura 5.29: Tela participantes da turma



Fonte : Autoria própria

5.10.4 Tela sair da turma

Dado que o usuário selecionou o botão “sair da turma”, na tela “dados pessoais”, logo então o mesmo será redirecionado para a presente tela, apresentada na Figura 5.30.

Onde o usuário terá possibilidade de sair da turma, se caso ele não for o administrador da mesma, pois neste caso ele terá primeiro que passar a administração para um outro usuário conforme foi explicado o procedimento na tela “participantes da turma”, antes enfim sair da turma.

Figura 5.30: Tela sair da turma



Fonte : Autoria própria

Conclusão

Devido ao notório crescimento tecnológico que vivemos, este projeto foi idealizado visando promover uma aplicação que possibilitasse aos universitários e motoristas, comunicarem-se de maneira mais simples e eficiente por meio deste aplicativo, evitando assim possíveis problemas de organização entre os mesmos.

Tudo isso foi possível, após ser realizado um planejamento e levantamento dos requisitos deste aplicativo, que desta maneira foram encontradas algumas necessidades, que logo se originaram em funcionalidades, sendo estas implementadas e desenvolvidas posteriormente.

Sobre o desenvolvimento deste *software*, o mesmo foi realizado com o padrão *MVC*, que favoreceu em uma melhor organização e gerenciamento das conexões entre todos os processos na camada *controller*, até as interfaces localizadas na camada *view* do aplicativo.

Contudo, ao final do desenvolvimento deste projeto, foi possível perceber que diante das funcionalidades desenvolvidas, o objetivo final foi alcançado, no qual este era desenvolver um ambiente para auxiliar na organização do transporte universitário e facilitar a comunicação entre os estudantes e motoristas.

6.1 Trabalhos futuros

Como trabalhos futuros para serem desenvolvidos, algumas novas funcionalidades foram listadas:

- Implementação da *API* do *Google maps*, permitindo o acesso à localização dos usuários do aplicativo.
- Desenvolvimento das notificações *push*, para informar aos usuários sobre possíveis alertas.
- Pagamento e controle das mensalidades dos estudantes, pelo aplicativo.

Referências Bibliográficas

ABRES. *Estatísticas: Ensino Superior*. 2019. Disponível em: <<https://abres.org.br/estatisticas/>>.

CORDEIRO, F. *Guardando Dados com SQLite*. 2017. Disponível em: <<https://www.androidpro.com.br/blog/armazenamento-de-dados/sqlite/>>.

FIREBASE. *Firestore Realtime Database*. 2021. Disponível em: <<https://firebase.google.com/docs/database/>>.

FRANCELLINO, S. M. Rebello de L. Migração pendular de estudantes universitários na região de aquidauana-mato grosso do sul-brasil. *Trayectorias Humanas Trascontinentales*, Université de Limoges, n. 6, 2020.

GLAUBER, N. *Dominando o Android com Kotlin*. [S.l.]: Novatec, 2019.

IBGE, E. *USO DE INTERNET, TELEVISÃO E CELULAR NO BRASIL*. 2019. Disponível em: <<https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html>>.

KOEPSSEL, M. Z. Blumenbus: Aplicativo móvel para consulta de informações do transporte público de blumenau. 2018.

LECHETA, R. R. *Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. [S.l.]: Novatec, 2013.

NEVES, C. E. B. Ensino superior no brasil: expansão, diversificação e inclusão. In: *Trabalho apresentado no Congresso da LASA (Associação de Estudos Latino Americanos), São Francisco, Califórnia*. [S.l.: s.n.], 2012.

OLIVEIRA, P. A. B. *Análise da mobilidade pendular na região metropolitana de Belo Horizonte em 2001-2002*. 2011. Disponível em: <http://www.biblioteca.pucminas.br/teses/TratInfEspacial_OliveiraPAB_1.pdf>.

REIS, R. B.; DUARTE, V. R.; PONCIANO, L. Nossavan: Proposta e implementação de um aplicativo para usuários de transportes escolares. 2018.

ROCHA, J. G. Arquitetura em camadas com uso do paradigma mvc e processo unificado na programação de software orientado a objetos. *Tecnologias em Projeção*, v. 9, n. 1, p. 31–49, 2018.

ROCHA, P. S. B. Aplicativos de transporte de passageiros como uber e 99 em belo horizonte: uso e características dessas plataformas. *Gestão de Trânsito-Unisul Virtual*, 2018.

SALATINO, A. T. et al. Aplicativo de gestão de transporte rodoviário. *Anais da Feira de Ensino, Pesquisa e Extensão do Instituto Federal Catarinense Campus Fraiburgo-e-ISSN 2594-5572*, v. 1, n. 1, 2019.

SANTOS, A. F. dos et al. Identificação do escopo de software a partir da análise de requisitos utilizando a uml. 2016.

SANTOS, F. G. d. C. B. d. Aplicativo para atender requisições de deslocamento solicitadas por passageiros. Universidade Tecnológica Federal do Paraná, 2020.

SILVA, W. C. S. d. Aplicações móveis nativas com react native e firebase: um estudo de caso. Universidade Federal do Maranhão, 2018.

VOGEL, L. *Android SQLite Database - Tutoria*. 2011. Disponível em: <<http://www.vogella.de/articles/AndroidSQLite/article.html>>.