DESENVOLVIMENTO DE UM APLICATIVO EM *ANDROID* PARA O SINDICATO DOS CONDUTORES EM TRANSPORTE PÚBLICO ALTERNATIVO DE PEDRAS DE FOGO - PB

ARAÚJO, Bruno Rodrigues de

RESUMO

A expansão dos smartphones nos tem possibilitado diversas transformações ao longo dos anos e em um espaço curto de tempo, têm surgido vários aplicativos que nos possibilitam fazer determinadas tarefas. Nesse contexto, tem surgido ferramentas que nos auxiliam na construção de aplicativos, como por exemplo, o Android Studio que nos possibilita criar aplicativos reais para o sistema operacional Android. É através dele que iremos criar a nossa aplicação. O trabalho tem como objetivo desenvolver um aplicativo para auxiliar os motoristas de transporte alternativo, visto que ainda é utilizado de uma forma manual, a realização do check-in, foi necessário a criação desse aplicativo em Android, utilizando a linguagem *Kotlin,* se comunicando com o Google Maps API, e através da localização do motorista e dentro da área determinada, poderá realizar o check-in.

Palavras chaves: Google Maps. Android. Firebase.

1. INTRODUÇÃO

Com o grande crescimento de *smartphones*, cada vez mais empresas têm se voltado para o mundo *mobile*, criando seus próprios aplicativos e investindo cada vez mais nessa área. Uma pesquisa sobre o uso de *smartphone* realizada pela empresa *Strategy Analytics* informou uma estimativa de que 3,85 bilhões de pessoas possuem um celular em 2021. "Nós estimamos que a base de usuários de *smartphones* cresceu dramaticamente, de 30 mil usuários em 1994 para 1 bilhão em 2012, e agora bateu o recorde com 3,85 bilhões em 2021", afirmou *Yiwen Wu*, analista sênior da empresa. "Com uma estimativa de 7,9 bilhões de pessoas no planeta em junho de 2021, isso significa que 50% de todo o mundo possui *smartphones*", acrescentou (TECMUNDO, 2021).

Alinhado com o avanço da tecnologia, as empresas e desenvolvedores têm criado diversos aplicativos com diferentes soluções para várias áreas, mas ainda assim, sentimos falta de um aplicativo em um contexto específico. Sendo assim, foi colocada em prática a ideia de um aplicativo para o transporte alternativo de Pedras de Fogo - PB.

O transporte alternativo nada mais é que um transporte de locomoção de uma cidade para outra em carros comuns. Nas cidades do interior da Paraíba existem esses transportes que levam as pessoas para a capital (João Pessoa - PB), que possui um local ou "ponto" na cidade, onde as pessoas que precisam se locomover para a capital pagam uma taxa para ir. Existem dois pontos onde ficam os carros, na cidade local (Pedras de Fogo - PB) e na cidade de destino (João Pessoa - PB). São dezenas de carros e quando o carro chega na capital, é necessário que uma pessoa anote o nome do motorista em um caderno, para que ele saiba qual a vez dele, já que são vários carros e geralmente só saem quando tem pessoas suficientes para a viagem.

Pensando nisso, foi criado um aplicativo *Alternative Check-In* para substituir essa forma manual de anotar o nome do motorista. O aplicativo feito em *Android*, funciona da seguinte forma, o motorista assim que chegar a cidade de destino, poderá fazer o *check-in* pelo aplicativo, o *check-in* verifica se o usuário está realmente no local, através da *API* do *Google Maps* e depois envia as informações do motorista para uma lista utilizando o *Realtime Database* do *Firebase*, onde irá mostrar a sua posição em tempo real.

2. ANDROID

O Android é o sistema operacional móvel mais utilizado do mundo, é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones e tablets, além de ter o código aberto (*open source*), isso facilita para desenvolvedores contribuir para a plataforma. Para trabalhar com desenvolvimento *Android* é necessário ter o *Android SDK* em sua máquina.

O *Android SDK* é o software utilizado para desenvolver aplicações no *Android*, que tem um emulador para simular o dispositivo, ferramentas utilitárias e uma *API* completa para a linguagem *Java*, com todas as classes necessárias para desenvolver as aplicações, (LECHETA, 2015). Utilizaremos o *Android Studio* que já vem com o *Android SDK* integrado com a linguagem de programação *Kotlin* para criarmos nossa aplicação.

O *Android Studio* é o ambiente de desenvolvimento integrado (*IDE*, na sigla em inglês) oficial para o desenvolvimento de aplicativos *Android*. O *Android* Studio possui diversos recursos para aumentar sua produtividade na criação de

aplicativos *Android,* dentre eles um sistema de compilação flexível baseado em *Gradle*, um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos *Android*, *Frameworks* e ferramentas de teste cheios de possibilidades *(ANDROID DEVELOPERS, 2021).*

KOTLIN

A linguagem de programação *Kotlin* foi desenvolvida pela *JetBrains*, a mesma criadora do *Android Studio*, e veio como uma alternativa ao *Java* para desenvolvimento de aplicativos *Android.* Em 2017, durante o evento *Google I/O,* foi feito o anúncio que *Kotlin* é oficialmente a linguagem para desenvolvimento de aplicativos *Android*. Podemos definir Kotlin como uma linguagem de programação pragmática que combina os paradigmas de Orientação a Objetos e Programação Funcional (RESENDE, 2018).

GOOGLE MAPS API

O *Google Maps API* (Interface de Programação de Aplicações) nos permite integrar a funcionalidade de mapas geográficos no desenvolvimento de um aplicativo *Android*. Os métodos da *API* fornecem localização através do *GPS*, recuperando latitude e longitude que serão usados para recuperar a localização atual do motorista, além de permitir a interação do motorista com o mapa.

FIREBASE

O *Firebase* é uma plataforma de desenvolvimento de aplicativos que ajuda você a criar e desenvolver aplicativos e jogos que os usuários adoram. Apoiado pelo *Google* e confiável por milhões de empresas em todo o mundo. (FIREBASE, 2022).

Segundo *Moroney* (2017), o objetivo é fornecer as ferramentas e a infraestrutura de que você precisa para criar ótimos aplicativos. É um aprimoramento, oferecendo serviços comuns que você pode precisar, como um *back-end* de banco de dados, autenticação segura, mensagens e muito mais. Isso evita a necessidade de criá-los por conta própria, permitindo que você se concentre no que torna seu aplicativo distinto. Além disso, possui tecnologias que você pode colocar em seu aplicativo e site que o ajudarão a expandir seus negócios por meio de referências, links e muito mais.

3. METODOLOGIA

A pesquisa sobre o determinado problema realizado nos leva a colher dados para chegarmos a uma solução. O objetivo da pesquisa foi desenvolver um aplicativo para fazer o *check-in* do motorista no local do destino através da sua localização.

REQUISITOS FUNCIONAIS

ID	NOME	DESCRIÇÃO	PRIORIDADE
RF01	Mostrar o mapa	O aplicativo deverá mostrar um mapa ao usuário	Essencial
RF02	Mostrar localização	O aplicativo deverá mostrar a localização do usuário	Essencial
RF03	Mostrar posição	O aplicativo deverá mostrar posição do usuário ao realizar check-in	Essencial
RF04	Mostrar posições de terceiros	O aplicativo deverá mostrar as posições de outros usuários que realizaram check-in	Essencial
RF05	Check-in	O sistema deve permitir realizar check-in do usuário	Essencial
RF06	Check-out	O sistema deve permitir realizar check-out do usuário	Essencial
RF07	Administrador	O sistema deve ter um usuário administrador	Essencial
RF08	Administrador cadastrar/editar/deletar	O sistema deve permitir que somente o administrador cadastre, edite e delete os motoristas	Essencial

REQUISITOS NÃO FUNCIONAIS

ID	NOME	DESCRIÇÃO	PRIORIDADE
RNF01	Desempenho	O sistema deve garantir desempenho satisfatório, com o intuito de justificar sua utilização em detrimento ao atendimento a outras plataformas do mercado..	Essencial
RNF02	Fluidez	O sistema irá garantir a rápida transição entre as telas, a partir de todos os recursos disponíveis.	Essencial
RNF03	Resiliência	O sistema possibilitará ajustes com rapidez de forma periódica e ocasional, evitando perda de disponibilidade.	Importante
RNF04	Tempo de resposta	O sistema garantirá rápida consulta ao banco de dados e disponibilização de dados após autenticação, além de boa comunicação com servidores.	Desejável
RNF05	Segurança	O sistema deverá garantir autenticidade, disponibilidade, integridade e restrições de acesso.	Essencial
RNF06	Restrição de acesso	O sistema assegurará que cada tipo de usuário (administrador, motorista) possua acesso exclusivo às suas respectivas áreas.	Essencial
RNF07	Autenticidade	O sistema garantirá acesso exclusivo dos usuários aos seus respectivos perfis, impedindo ao máximo acesso falsos ou não autorizados.	Importante
RNF08	Internet	O sistema deverá funcionar apenas com acesso a internet.	Essencial
RNF09	Sistema operacional	O sistema irá rodar no sistema operacional Android, a partir da versão 7.0	Essencial
RNF10	Localização	O sistema deverá funcionar apenas com a permissão da localização do usuário.	Essencial

4. DESENVOLVIMENTO

Inicialmente foram desenvolvidas as telas ou interface do usuário, para uma visão melhor do que está sendo aplicado.

FIGURA 1: Gradle - Dependência do Google Maps

```
//Maps
implementation 'com.google.android.gms:play-services-location:19.0.1'
implementation 'com.google.android.gms:play-services-maps:18.0.2'
```

Fonte: Próprio autor (2022)

Começando pela tela do mapa, foi necessário a inclusão da dependência do *Google Maps API* no gerenciador de dependências conhecido como *Gradle* no *Android Studio*, assim como mostra a imagem acima.

FIGURA 2: Gradle - Dependência do Google Maps

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyAE9Hcl2UEcj7AOZZZzoPOIi_Gk1RnzSY0" />
```

Fonte: Próprio autor (2022)

Sendo necessário a criação do projeto através do console *developers* do google, onde é gerado uma chave, e através da chave é que temos acesso ao *Google Maps API*, essa chave deve ser integrada ao projeto do Android no arquivo Manifest (arquivo onde possuem os requisitos do aplicativo).

Desta forma, como vemos na imagem agora podemos acessar o *Google Maps* e definir regra através do *Android*.

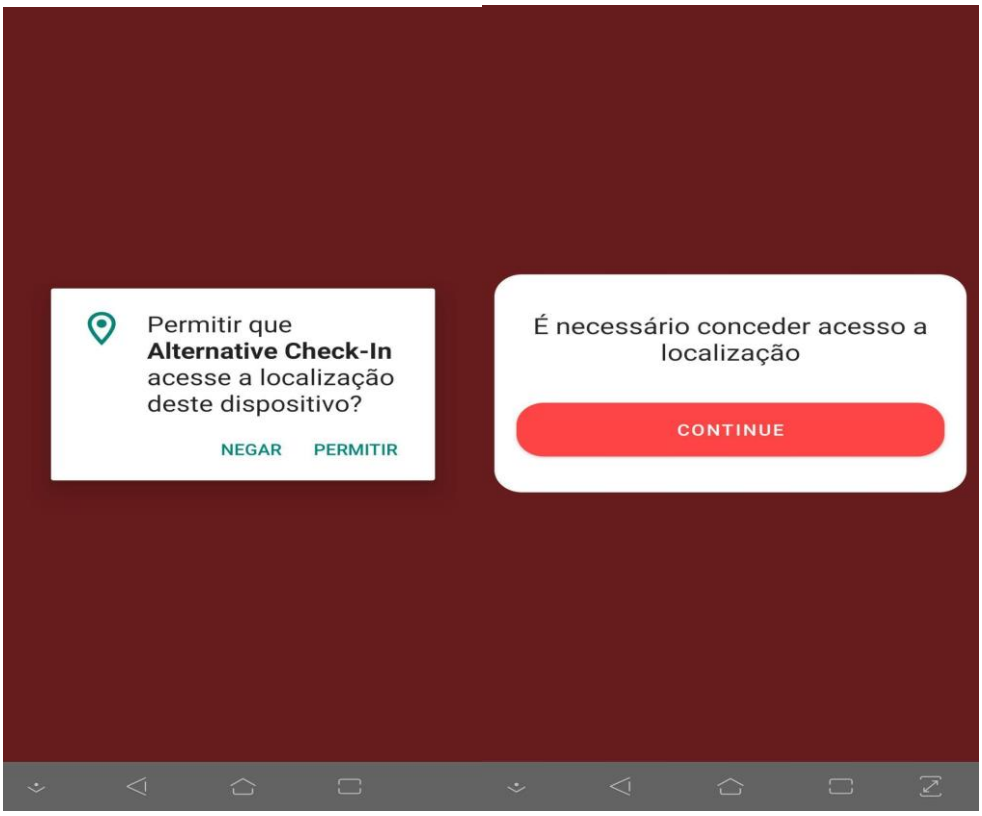
FIGURA 3: Arquivo Manifest - Permissões para obter localização

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Fonte: Próprio autor (2022)

Para obter a localização primeiramente precisamos obter a permissão do usuário para ter acesso a sua localização, também a permissão para acesso a internet, visto que o aplicativo precisará ter acesso a internet.

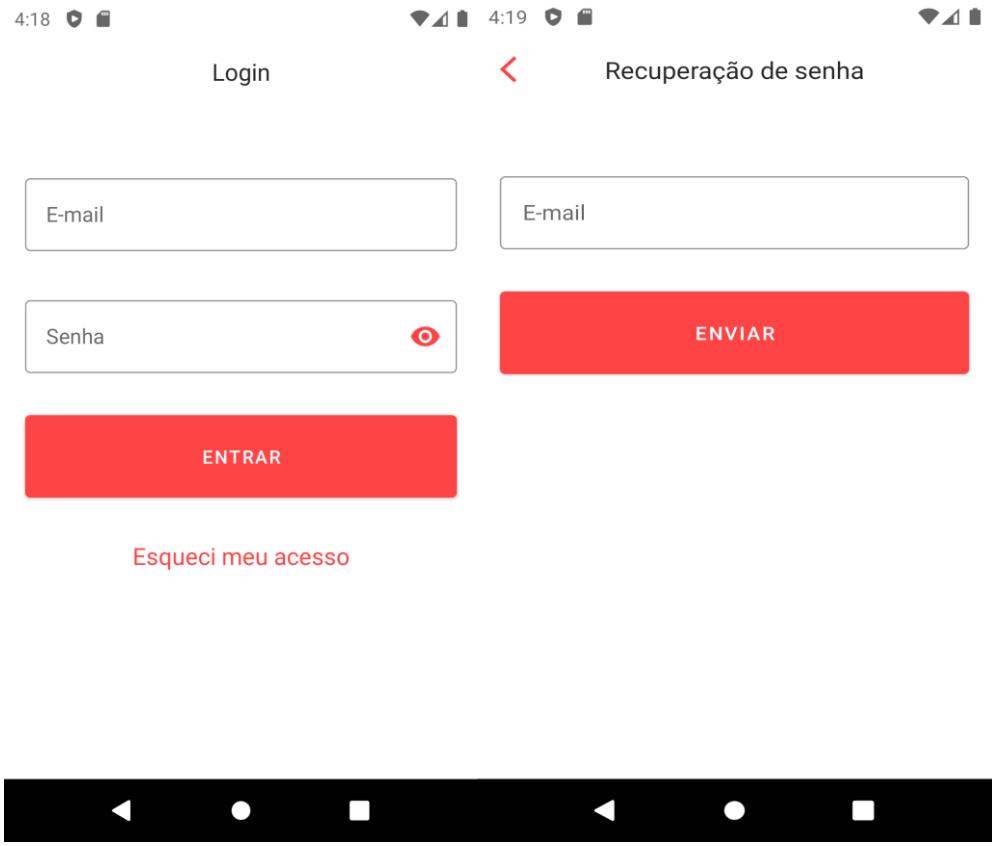
FIGURA 4: Dialog- Dialog para aceitar permissão



Fonte: Próprio autor (2022)

Na tela inicial solicitamos a permissão da localização, assim que o usuário permite, ele é direcionado para a tela de login, caso contrário é informado um *dialog* onde explica ao usuário que se faz necessário o acesso a sua localização.

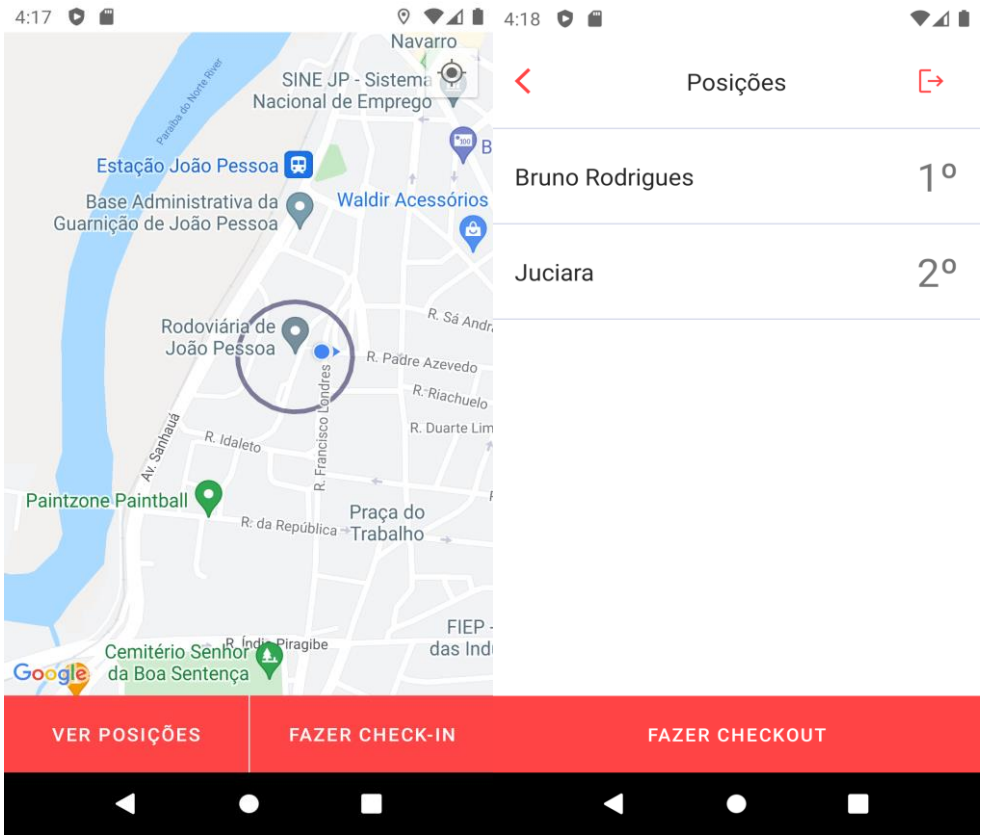
FIGURA 5: Telas - Tela de login e recuperação de senha



Fonte: Próprio autor (2022)

A tela de login do usuário é autenticado via Firebase Authentication, se caso o usuário não esteja cadastrado, é retornando uma mensagem avisando-o, senão o usuário é direcionado para tela de mapa.

FIGURA 6: Telas - Tela do Google Maps e Posições

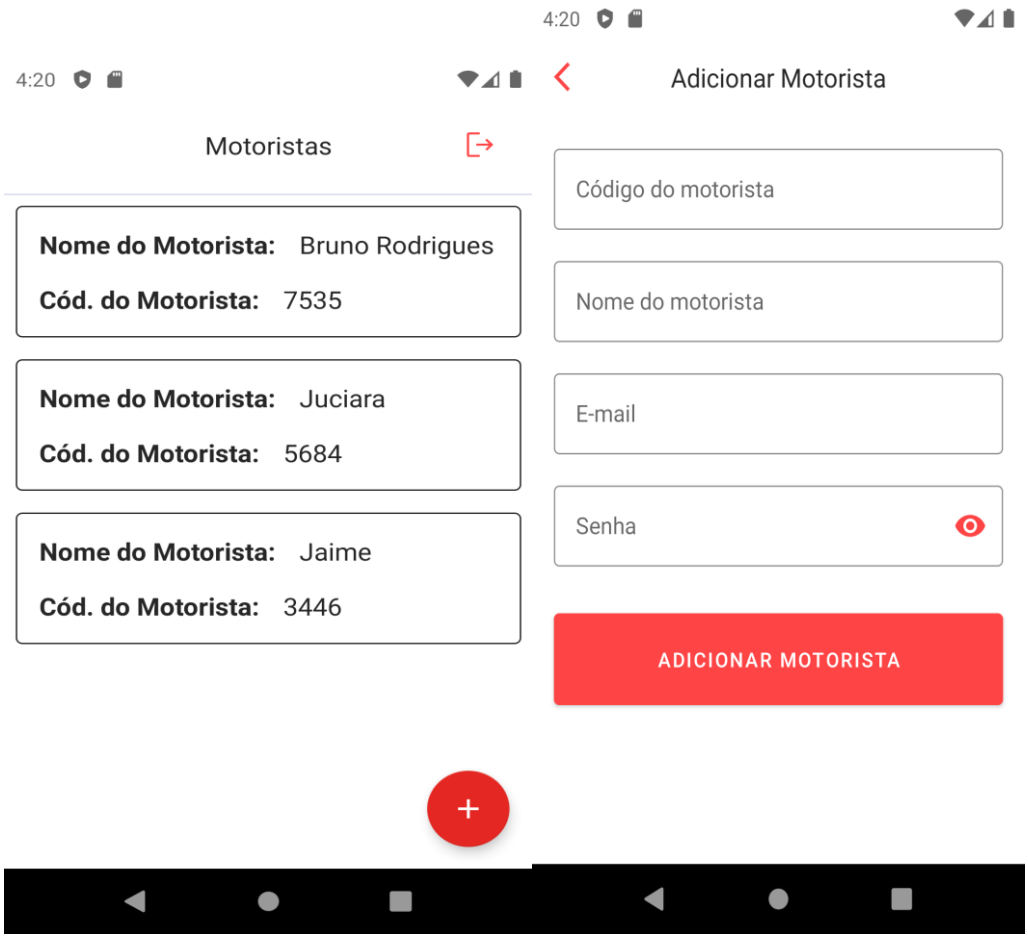


Fonte: Próprio autor (2022)

Na tela principal do motorista fica o mapa, onde mostra sua localização atual, e a opção de ver as posições de outros motoristas, o motorista só poderá fazer check-in na área determinada dentro do mapa, se ele não estiver naquela área, é exibido uma *dialog* informando, assim que ele faz o check-in, o motorista é direcionado para a tela de Posições, onde vai mostrar sua posição. Na tela de posição o usuário tem a opção de fazer o check-out, assim ele é retirado da lista.

Assim que o motorista faz o check-in, é enviado os dados para o Realtime Database do Firebase, uma ferramenta do Firebase onde mostra os dados em tempo real. Na tela de posições é onde pega esses dados enviados e lista cada motorista que fez o check-in. Quando o motorista faz o check-out, seu nome é excluído do database, sendo assim, a tela é atualizada e aquele motorista já não está mais na lista.

FIGURA 7: Telas - Tela para adicionar um novo motorista

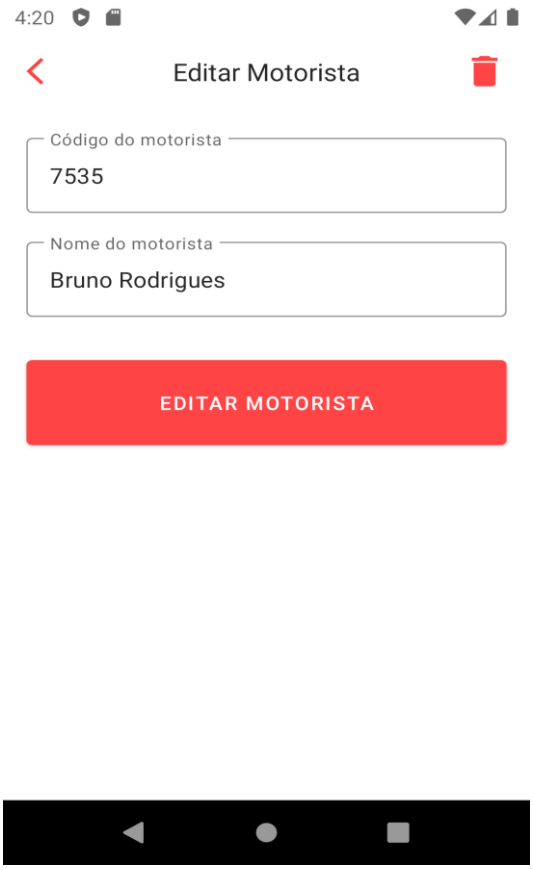


Fonte: Próprio autor (2022)

Por último temos a tela do administrador, onde tem a opção de cadastrar, editar e deletar um motorista. Foi um grande desafio já que tive que aprender como trabalhar com essas ferramentas do Firebase e fazer as integrações corretamente dos motoristas cadastrados. Na tela inicial, fica listado os motoristas que estão cadastrados,

No cadastramento, o usuário é integrado ao Realtime Database e Firebase Authentication, são feitas validações para os campos, o campo do código e nome, as validações são feitas no código, já o e-mail e a senha as validações são feitas através do Authentication, o mesmo retorna os possíveis erros e apenas listamos via código e transmitirmos para o administrador

FIGURA 8: Tela - Tela de editar motorista



Fonte: Próprio autor (2022)

Ao clicarmos em um motorista, é direcionado para a tela de edição, onde pode editá-lo como também deletá-lo.

5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo a criação de um aplicativo para transporte alternativo, utilizando o Android, Kotlin, Google Maps API e Firebase. Visto que, a forma como é feito o processo de incluir e verificar a posição do motorista, é de forma manual e que não garante uma forma eficaz e eficiente, foi necessário a criação do aplicativo, que garante que o motorista esteja no local, e garante a sua posição correta na lista, sem o devido esquecimento que poderia acontecer de forma manual.

O trabalho também fez com que o autor pudesse obter mais experiência na área, como também conhecimentos adquiridos ao decorrer do projeto, visto que foi necessário aprender algumas ferramentas, para que o aplicativo tivesse eficiência.

Sendo assim, foi possível aplicar os conhecimentos teóricos e tecnológicos adquiridos nas disciplinas do curso de sistemas para internet de uma forma integrada para poder desenvolver o aplicativo proposto.

6. TRABALHOS FUTUROS

O aplicativo até então desenvolvido já consegue atender aos usuários, mas necessita de algumas mudanças futuras, como interface, novas implementações, separação em dois aplicativos, um para o administrador e outro para o motorista, relatórios de check-in, quantidades de motoristas, suporte, avaliações, expansão para outras cidades, já que a maioria possui o mesmo ponto na capital. Entre outras funcionalidades também que podem surgir através de pesquisa ao decorrer do tempo.

REFERÊNCIAS

ANDROID DEVELOPERS. 2021. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em 17 de Abril de 2022.

FIREBASE, Firebase. 2022. <https://firebase.google.com/?hl=pt>. Acesso em 30 de Abril de 2022.

LECHETA, Ricardo R. **Google Android**: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 5. ed. São Paulo: Novatec Editora Ltda, 2015.

MORONEY, Laurence. **The Definitive Guide to Firebase**: Build Android Apps on Google's Mobile Platform. Apress; 1st ed. edição, 2017.

RESENDE, Kassiano. **Kotlin com Android**: Crie aplicativos de maneira fácil e divertida. Brasil: Casa do Código, 2018.

TECMUNDO, 2021. **Pesquisa estima que metade da população mundial tem smartphones.** Disponível em: <https://www.tecmundo.com.br/mercado/220009-pesquisa-estima-metade-populacao-mundial-tem-smartphones.htm#:~:text=A%20empresa%20de%20pesquisa%20e,cerca%20de%20 R%242%20 mil.> Acesso em 4 de Março de 2022.