

VOU DE VAN: UM APLICATIVO MÓVEL PARA GERENCIAMENTO DE VANS ESCOLARES EM REACT NATIVE

VOU DE VAN: A MOBILE APP FOR MANAGEMENT OF SCHOOL VANS IN REACT NATIVE

Augusto Kalel Stocco Silva
ORCID: 0000-0003-0927-7340
UNASP – Hortolândia/SP
augustokalel2@gmail.com

Gabriel de Almeida de Oliveira
ORCID: 0000-0003-3320-8230
UNASP – Hortolândia/SP
gabriel-ao@hotmail.com

Orientador: Gregory Acacio Seibert Oliveira
ORCID: 0000-0002-6370-7797
UNASP – Hortolândia/SP
gregory.oliveira@unasp.edu.br

RESUMO: Estudantes que iniciam a vida universitária tem diversos desafios pela frente, dentre eles está localizar um serviço de transporte de van que atendam suas necessidades, a divulgação deste tipo de serviço encontra-se na maioria dos casos fragmentadas em diversas plataformas dificultando a localização dessas informações pelos estudantes. Pensando nessa dificuldade o presente artigo irá abordar uma solução para essa problemática, que será um protótipo de aplicativo, que visa facilitar a busca de quem precisa desses serviços, quanto de quem faz divulgação, reunindo em uma única plataforma dados relevantes, como localizar a van mais próximo do seu ponto de início e final, valor da mensalidade, serviços disponíveis, entre outras informações que serão abordadas ao decorrer desse artigo. Foi feita também uma pesquisa quantitativa que visa levantar a aceitação das funcionalidades e da interface, perante possíveis usuários do aplicativo. Durante esse estudo foi abordado principalmente o algoritmo responsável pela busca de vans com a linguagem Java script, utilizando como framework o React Native e o banco de dados não relacional Fire Base para a simulação da busca de vans. Foi levado em consideração o desenvolvimento do protótipo diretamente na linguagem nativa do Android, porem seria necessário maior estudo e principalmente tempo para apresentar os resultados desejados, optando assim pelas tecnologias abordadas acima. Por fim um pequeno protótipo visual foi criado a fim de entender como o aplicativo poderia vir a ficar na versão final. Como objetivos futuros, é pretendido finalizar o produto, lançando-o no mercado.

ABSTRACT: Students who start university life have several challenges ahead, among them is locating a van transport service that meets their needs, the dissemination of this type of service is in most cases fragmented in various platforms making it difficult to find this information by students. With this difficulty in mind, this article will address a solution to this problem, which will be an application prototype, which aims to facilitate the search for those who need these services, as well as those who disseminate, gathering on a single platform relevant data, such as locating the van. closest to your starting and ending point, monthly

fee, available services, and other information that will be covered throughout this article. It was also made a quantitative research that aims to raise the acceptance of features and interface, before possible users of the application. During this study we mainly approached the algorithm responsible for the search of vans with the Java script language, using as framework React Native and the non-relational database Fire Base for the simulation of the search of vans. Considering the development of the prototype directly in the native language of Android was taken into account, but it would take more study and especially time to present the desired results, thus opting for the technologies discussed above. Finally a small visual prototype was created in order to understand what the application might look like in the final version. As future goals, it is intended to finalize the product by launching it on the market.

PALAVRAS-CHAVE: Aplicativo. React Native. Serviço de transporte.

KEYWORDS: Application. React Native. Transportation service.

1 INTRODUÇÃO

Com o crescimento populacional em regiões urbanas, e o aumento da demanda por veículos (ANFAVEA, 2018), começaram a surgir problemas como aumento do congestionamento por conta da maior frota de automóveis nas ruas (FOLHA, 2018). Uma boa solução para esses problemas seria o uso de transportes públicos e do compartilhamento de veículos. Em especial o transporte universitário de vans é uma ótima opção quando falamos em maneiras de locomoção alternativas, afinal esse meio de transporte fornece mais conforto e segurança, já que o estudante embarca e desembarca em seu ponto de preferência.

Contudo encontrar esses fornecedores é uma tarefa árdua, a informação se encontra espalhada em diversas plataformas como o Facebook, cartazes, OLX entre outros, dificultando a comunicação tanto de quem precisa do transporte e de quem anuncia o serviço. Sendo assim muitos estudantes acabam não conseguindo entrar em contato com esses profissionais, e prestadores de serviço não conseguem adaptar suas rotas em regiões que necessitam de seus serviços.

Levando em conta as situações levantadas anteriormente, esse artigo vai abordar o uma possível solução a partir de uma aplicação mobile, visando facilitar a vida de quem precisa de transporte. A solução será focada nos universitários, com funcionalidades de busca por condutores de van sempre focando na praticidade e facilidade, facilitando a busca desse tipo de serviço além de fornecer mais opções de escolha para os universitários e conseqüentemente diminuindo os veículos nas ruas e facilitando a vida de quem estuda.

Nessa pesquisa será feita a construção de um protótipo do aplicativo voltado ao estudante, que utilizará como principais ferramentas o *REACT Native*, *Firebase*, *Expo.io* e

o *Node.js*, com funcionalidades que iram facilitar a busca e a visualização das rotas, exibindo valores e outros dados relevantes como serviços extras (ar-condicionado, wifi), auxiliando toda essa comunidade, já que toda a informação ficaria centralizada em uma única plataforma que poderia ser acessada através de smartphones.

2 METODOLOGIA

Está pesquisa tem foco quantitativo que terá como objetivo comprovar numericamente se o aplicativo terá impacto significativo com o usuário final (MINDMINERS, 2019), foi obtido aproximadamente cerca de 90 opiniões distintas, com público alvo sendo estudantes universitários, com intuito de auxiliar pessoas que desejam contratar um serviço de van particular.

2 DESENVOLVIMENTO

Foi utilizado como IDE o Visual Studio Code que é disponibilizada gratuitamente pela empresa Microsoft e com código livre (*open source*), essa ferramenta é excepcional para se trabalhar pois é leve e multiplataforma, suportando diversas linguagens de programação como C#, Ruby e React (DEV MEDIA, 2016). Além dessas vantagens, por se tratar de uma IDE desenvolvida e mantida pela Microsoft, temos todo o suporte da mesma sendo atualizada mensalmente além de possuir uma grande gama de extensões (VISUAL, 2019), a tornando uma ótima opção de escolha. A figura 7 exibe o ambiente de trabalho do VS Code, sendo utilizado para produção da aplicação abordada nesse artigo.

O React-Native tem como principais vantagens o uso multiplataforma de seu código, ou seja basta aprender uma vez como a framework funciona, então será capaz de desenvolver aplicativos para Android e IOS, utilizando respectivamente as funções nativas de cada uma das plataformas (REACT, 2019), por esse motivo foi o framework escolhido para desenvolver o protótipo, podendo assim desenvolver nossa ideia nos dois principais sistemas operacionais mobile do mercado.

Foi utilizado o Firebase que é um banco de dados não relacional hospedado na nuvem, sua principal vantagem é a sincronização em tempo real em diversas plataformas (FIREBASE, 2019).

A codificação na figura 8 realiza a conexão do React-Native com o banco de dados Firebase, como exposto na variável “config” é passado o valor da apikey, o domínio, o URL

do banco de dados e o ID do projeto. Dessa forma, o aplicativo é autorizado e tem concedida o acesso ao banco de dados.

Figura 5: *Conexão DO Firebase-React*

```
const config = {
  apiKey: "Chave da api disponibilizada ",
  authDomain: "vanuniversitariafirebase.firebaseio.com",
  databaseURL: "https://vanuniversitariafirebase.firebaseio.com",
  projectId: "vanuniversitariafirebase",
};
firebase.initializeApp(config);

export default class App extends React.Component {
  state = {
    isLoadingComplete: false,
  };
};
```

Fonte: autoria própria.

Se os dados estiverem corretos, o comando `InitializeApp(config)` vai realizar a conexão no banco de dados. O `Export default class App extends React.Component` exporta os dados para a comunicação.

4 RESULTADOS OBTIDOS

Para começar com a programação em *React-Native* é necessário importar todos os componentes que serão usados, dessa forma poderemos trabalhar com todas as bibliotecas e conexões necessárias como demonstrado na Figura 9.

Figura 6: *Bibliotecas importadas.*

```
import React, {Component} from 'react';
import firebase from 'firebase';

import {
  Image,
  Platform,
  ScrollView,
  StyleSheet,
  Text,
  Button,
  View,
  TextInput,
  FlatList
} from 'react-native';

import VanCard from '../components/VanCard';
```

Fonte: autoria própria.

O seguinte trecho do código tem como foco a busca da van escolar de acordo com o local que o usuário precisa desembarcar, com isso o aplicativo exibirá os motoristas disponíveis para aquela rota ou a rota mais próxima ao destino solicitado, disponibilizando assim, o telefone para contato do condutor. Como se pode ver na figura 10 é feita uma busca no banco de dados do *Firebase*, procurando na referência de “vans” ordenando pelo destino.

Figura 7: *Lógica busca de van.*

```
buscarVan = () => {  
  //Executar o Query e retornar os motoristas  
  
  const component = this; //Bind do contexto this  
  const chegada = component.state.chegada.toString();// Assegurar que a Query é uma string  
  
  //Cria uma referencia do Database e ordena por "destino"  
  firebase.database().ref("vans").orderByChild("destino").equalTo(chegada).once("value", function (response) {  
    if (response.exists()) {  
      const vanResult = Object.values(response.val()); //Vetor resposta do firebase  
    }  
  });  
}
```

Fonte: autoria própria.

O método *vansComp* irá gerar a lista de vans que são compatíveis com o que o usuário precisa, buscando cada item possível no *Firebase*, dessa forma, se houver uma van compatível a lista é exibida ao usuário com os condutores que fazem a rota, e mesmo que não seja encontrado precisamente com a busca rota, o aplicativo fara a exibição de outros contatos que podem vir a atender a necessidade do usuário. Na figura 11 comentamos essas funções para que o entendimento das funções seja de fácil compreensão.

Figura 8: *Gerar lista.*

```
let vansComp = []; //Cria uma lista de vans compatíveis com o campo de busca de Embarque  
  
const partida = component.state.embarque.toString(); //Converte o valor do input em string  
vanResult.map((van) => { //a função map executa a função para cada item da resposta do firebase  
  if(van.partida.includes(partida)){ //verifica se cada item possui o bairro em questão no trajeto  
    vanResult.splice(van,1); //Retira a(s) van que deu Match da lista das resposta para colocar em destaque  
    vansComp.push(van); //Adiciona, se positivo a van na lista de vans compatíveis  
  }  
});
```

Fonte: autoria própria.

Se os dados que o usuário digitar forem localizados no banco de dados, esses dados caíram na condição *IF* para encontrar a van que ele precisa, se não houver um condutor com essa rota, a condição passa para um *ELSE* e com isso é feita uma busca de possíveis rotas para atendê-lo, a seguir na figura 12 temos a foto do código e como o aplicativo se comportaria.

Figura 9: Rotas alternativas.

```

const vansSugestao = vanResult.slice();

if(vansComp.length != 0){ //Houve match

  if(vansSugestao.length !=0){
    component.setState({
      vanMatch: vansComp,
      outrasVans: vansSugestao,
      matchCard: true,
      outrasCard:true,
      dontFind: false});
  }

  else{
    component.setState({
      vanMatch: vansComp,
      outrasVans: vansSugestao,
      matchCard: true,
      outrasCard:false,
      dontFind: false});
  }
}

```

Busque por uma Van

Nova América

UNICAMP

BUSCAR VAN

Motorista: Carlos
Van saindo de: Nova América
Para: UNICAMP
Preço: 140
Lugares: 7

(19)99678-7668

(19)99350-0978

Outras vans indo para UNICAMP

Motorista: Carlos
Van saindo de: Nova América
Para: UNICAMP
Preço: 140
Lugares: 7

(19)99678-7668

Buscar Van
 Perfil

Fonte: autoria própria.

Caso não houver nenhuma dessas opções, uma terceira condição é colocada, ela mostra que no momento não há condutores que atendam a rota, ou seja, retornando apenas um erro, como demonstrado na figura 13.

Figura 10: Sem vans disponíveis.

```

else{ //Não Houve match mas tem outras vansF
  component.setState({matchCard: false,
    outrasCard: true,
    dontFind: true,
    outrasVans: vansSugestao
  })
}

} //Se o valor não existir no banco

else{
  component.setState({matchCard: false,
    outrasCard: false,
    dontFind: true})
}

```

Busque por uma Van

Parque Ortolandia

São Leopoldo Mandic

BUSCAR VAN

Não encontramos nenhuma van :/

Fonte: autoria própria.

Nesta etapa, foi trabalhado os padrões de tela com a classe “Render()”. Em “style” aqui é trabalhado tudo relacionado ao visual, nas tags como “TextInpu” será extraído os

dados informados pelo usuário, tratando os valores de embarque e chegada, sendo assim o aplicativo possuirá os dados do bairro e do local de estudo, podendo então analisar e fazer a combinação das informações. Além dessa *tags* utilizamos um botão para servir como o gatilho para que a ação quando o botão for pressionado seja executada, enviando assim os dados pelo comando `onPress={this.buscarVan}` para verificar se destino que ele necessita é atendida por um condutor. Na figura 14 podemos visualizar o código e como ele seria apresentado ao usuário.

Figura 11: Tela de busca.



Fonte: autoria própria.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como principal foco o desenvolvimento de um algoritmo que fosse capaz de fazer de forma objetiva a busca de rotas capazes de atender de forma rápida e precisa a necessidade de locomoção de universitários que precisam de transporte para sua universidade, que foi executada de maneira satisfatória. Divulgamos uma pesquisa, com o intuito de observar a aceitação de possíveis funcionalidades, como resultado tivemos todas as informações necessárias para desenvolver o protótipo real da aplicação e a se haveria aceitação pelo público alvo.

Foi desenvolvido o *core* da aplicação, responsável pelo encontro do condutor e do aluno, foi utilizado técnicas em programação e no banco de dados para que fosse possível essa interação, possibilitando buscas precisas entre os endereços buscados e possíveis rotas para caso não seja localizado um endereço preciso.

Por fim por questões técnicas, não foi possível desenvolver a versão completa do que originalmente era pretendido apresentar, porém com os resultados obtidos nesta pesquisa, é pretendido futuramente, a conclusão do aplicativo, disponibilizando comercialmente para uso na sociedade.

REFERÊNCIAS

AFAVEA. Anfavea ajusta projeções para a indústria automobilística em 2018. **Base de dados** [anfavea.com.br](http://www.anfavea.com.br) 2017. Disponível em: http://www.anfavea.com.br/docs/04.10.18_Press_Resultados_Setembro.pdf Acesso em: 22 out. 2019.

DEV MEDIA. Introdução ao Visual Studio Code. **Base de dados** [devmedia.com.br](https://www.devmedia.com.br) 2016. Disponível em: <https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418> Acesso em: 22 out. 2019

FIREBASE. Firebase Realtime Database. **Base de dados** [firebase.google.com](https://firebase.google.com/docs/database) 2019. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br> Acesso em: 22 out. 2019

FOLHA DE SÃO PAULO. Lentidão média piora nas principais vias da cidade de São Paulo. **Base de dados** [folha.uol.com.br](https://www1.folha.uol.com.br) 2018. Disponível em: <https://www1.folha.uol.com.br/cotidiano/2018/08/lentidao-media-piora-nas-principais-vias-da-cidade-de-sao-paulo.shtml> Acesso em: 22 out. 2019.

MINDMINERS. Pesquisa qualitativa e quantitativa: qual é a melhor opção?. In: Mindminers. **Base de dados** <https://mindminers.com> 2016. Disponível em: <https://mindminers.com/blog/pesquisa-qualitativa-quantitativa/> Acesso em: 30 nov. 2019

REACT NATIVE. Build native mobile apps using JavaScript and React. **Base de dados** facebook.github.io 2019. Disponível em: <https://facebook.github.io/react-native> Acesso em: 22 out. 2019

VISUAL STUDIO CODE. Setting up Visual Studio Code. In: Visual Studio Code. **Base de dados** <https://code.visualstudio.com> 2019. Disponível em: <https://code.visualstudio.com/docs/setup/setup-overview> Acesso em: 22 out. 2019