



**ANTONIO MENEGHETTI FACULDADE - AMF  
SISTEMAS DE INFORMAÇÃO**

**GUILHERME MÜLLER SAGRILO RODRIGUES**

**APLICAÇÃO MÓVEL PARA RECOMENDAÇÃO DE PRODUTOS DE  
SUPERMERCADOS**

**RESTINGA SÊCA, RS**

**2018**

**GUILHERME MÜLLER SAGRILO RODRIGUES**

**APLICAÇÃO MÓVEL PARA RECOMENDAÇÃO DE PRODUTOS DE  
SUPERMERCADOS**

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação, Antonio Meneghetti Faculdade – AMF, apresentado como requisito parcial para obtenção do grau de bacharel de Sistemas de Informação.

Orientador: Profº Ms. João Otávio Cadó de Matos

**RESTINGA SÊCA, RS**

**2018**

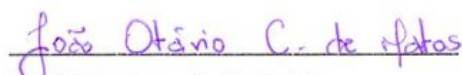
FACULDADE ANTONIO MENEGHETTI

Guilherme Muller Sagrilo Rodrigues

APLICAÇÃO MÓVEL PARA RECOMENDAÇÃO DE PRODUTOS DE  
SUPERMERCADO.

Trabalho de Conclusão de Curso-Monografia, apresentado como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

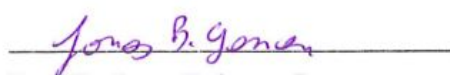
Orientador: Prof. Ms. João Cadó de Matos



Prof. Ms. João Cadó de Matos

Orientador do Trabalho de Conclusão de Curso

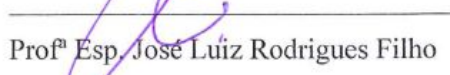
Antonio Meneghetti Faculdade



Prof<sup>o</sup> Dr Jonas Bulegon Gassen

Membro da Banca Examinadora

Antonio Meneghetti Faculdade



Prof<sup>o</sup> Esp. José Luiz Rodrigues Filho

Membro da Banca Examinadora

Antonio Meneghetti Faculdade

**Restinga Sêca, RS, 28 de novembro de 2018.**

## **AGRADECIMENTOS**

À todos que direta ou indiretamente fizeram parte da minha formação,  
o meu muito obrigado.

“Os que são loucos o suficiente para pensarem que podem mudar o mundo,  
são os que fazem”

Steve Jobs

## RESUMO

O presente trabalho descreve o desenvolvimento de uma aplicação móvel para as plataformas android e iOS, utilizando o *framework* React Native, a linguagem de programação JavaScript e a API do Google Maps. O objetivo do trabalho é o desenvolvimento de uma sistema de recomendação de produtos de supermercados baseado em filtros e na localização do usuário. Além do React Native, foram utilizadas algumas bibliotecas para complementar e auxiliar o desenvolvedor, como as bibliotecas Axios, React Native Maps e React Navigation. Entende-se que uma solução como essa tem a intenção de contribuir para que os usuários economizem tempo e dinheiro durante suas compras em supermercados. Diante desse cenário, espera-se também que o uso da aplicação contribua para uma redução dos preços dos produtos ofertados, estimulando a livre concorrência entre os estabelecimentos. A metodologia de pesquisa utilizada foi a de uma pesquisa quantitativa de caráter experimental, de forma que o autor pudesse, de forma prática, explanar a solução do problema de pesquisa. O resultado do trabalho constitui-se em um protótipo do aplicativo, apresentando algumas funcionalidades.

**Palavras-chave:** Sistema de Recomendação, React Native, API do Google Maps.

## ABSTRACT

This work describes the development of a mobile application for the android and iOS platforms, using the React Native framework, the JavaScript programming language and the Google Maps API. The objective of the work is the development of a supermarket products recommendation system based on filters and the user's location. In addition to the React Native, some libraries were used to complement and assist the developer, as the libraries Axios, React Native Maps and React Navigation. It is understood that such a solution is intended to help users save time and money during their grocery shopping. Given this scenario, it is also expected that the use of the application contributes to a reduction in the prices of the products offered, stimulating free competition between establishments. The research methodology used was that of a quantitative research of experimental character, so that the author could, in a practical way, explain the solution of the research problem. The result of the work is a prototype of the application, presenting some functionalities.

**Keywords:** Recommendation System, React Native, Google Maps API.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1.</b> Exemplo de Código escrito em React	15
<b>Figura 2.</b> Comparação de renderização React e React Native	16
<b>Figura 3.</b> Exemplo de código React Native	17
<b>Figura 4.</b> Exemplo de código para criar um mapa do Google	20
<b>Figura 5.</b> Resultado do exemplo de mapa do Google	21
<b>Figura 6.</b> Página inicial do VSCode	22
<b>Figura 7.</b> Tela de acesso ao aplicativo	26
<b>Figura 8.</b> Página inicial do aplicativo com o loader	27
<b>Figura 9.</b> Primeira aba do aplicativo	28
<b>Figura 10.</b> Segunda aba do aplicativo	29
<b>Figura 11.</b> Tela de perfil do usuário com informações padrão	30
<b>Figura 12.</b> Tela de perfil do usuário com informações modificadas	31
<b>Figura 13.</b> Resultado da consulta de produtos	32



## LISTA DE TABELAS

<b>Tabela 1.</b> Comparativo Ferramentas Mobile	17
<b>Tabela 2.</b> Ranking de linguagens de programação mais utilizadas na plataforma do Github	19

## LISTA DE ABREVIATURAS

JS - *JavaScript*

RN - *React Native*

IDE - *Integrated Development Environment ou Ambiente de desenvolvimento integrado*

iOS - *iPhone Operational System ou Sistema Operacional iPhone*

SDK - *Software Development Kit ou Kit de Desenvolvimento de Software*

HTTP - *Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto*

GUI - *Graphical User Interface ou Interface Gráfica do Usuário*

API - *Application Programming Interface ou Interface de programação de aplicativos*

## SUMÁRIO

<b>RESUMO</b>	<b>5</b>
<b>ABSTRACT</b>	<b>6</b>
<b>LISTA DE ILUSTRAÇÕES</b>	<b>7</b>
<b>LISTA DE TABELAS</b>	<b>8</b>
<b>LISTA DE ABREVIATURAS</b>	<b>9</b>
<b>SUMÁRIO</b>	<b>10</b>
<b>1 INTRODUÇÃO</b>	<b>12</b>
1.1 OBJETIVOS	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivos Específicos	13
1.2 JUSTIFICATIVA	13
<b>2 ABORDAGEM TEÓRICA</b>	<b>15</b>
2.1 REACT	15
2.2 REACT NATIVE	16
2.3 NODE.JS	18
2.4 JAVASCRIPT	18
2.5 API DO GOOGLE MAPS	19
2.6 VISUAL STUDIO CODE	21
2.7 TRABALHOS RELACIONADOS	22
<b>3 METODOLOGIA</b>	<b>22</b>
<b>4 DESENVOLVIMENTO DO PROJETO</b>	<b>24</b>
<b>5 RESULTADOS</b>	<b>25</b>
5.1 TELA DE ACESSO	25
5.2 TELA INICIAL	27
5.3 TELA DE FILTROS	29
5.4 TELA DE PERFIL DO USUÁRIO	30
5.5 RESULTADO DA CONSULTA	32
<b>6 CONSIDERAÇÕES FINAIS</b>	<b>33</b>

	11
6.1 TRABALHOS FUTUROS	33
<b>REFERÊNCIAS</b>	<b>35</b>

# 1 INTRODUÇÃO

Atualmente, a tecnologia se faz presente em grande parte de nossas atividades diárias. Nesse cenário, o uso de aplicativos e de novas ferramentas tecnológicas tende a melhorar e baratear os produtos e serviços comercializados ao redor do mundo.

Podemos enxergar isso facilmente por meio das mudanças que inúmeras empresas vêm realizando em suas formas de comercialização ao longo dos últimos anos, migrando das lojas físicas para as lojas virtuais, transformando-se em poderosos e-commerces a fim de baratear custos e potencializar lucros.

Olhando nessa direção, hoje em dia, já encontramos diversos serviços de busca gratuitos disponíveis na internet, os quais possibilitam aos usuários a busca, a comparação e a seleção de produtos e serviços de acordo com um ou mais critérios como preço, localização e disponibilidade em estoque. Como exemplo, podemos citar o site BuscaPé, o qual agrupa organiza inúmeros produtos em um mesmo ambiente virtual, com o intuito de tornar o processo de compra muito mais rápido e fácil. (BUSCAPÉ, s/d)

Diante do exposto, é possível perceber que aqueles consumidores que fazem uso da tecnologia na hora de buscar por serviços e produtos, tendem a economizar tempo, dinheiro, e, além disso, encontrar exatamente aquilo que procuram, tendo em vista a maior variedade de itens comercializados nos e-commerces em comparação com as unidades físicas.

Ainda dentro desse ambiente de compra e venda de produtos, nos últimos anos, com o aumento notável do número de marcas e de estabelecimentos, é possível percebermos uma diferença significativa entre os preços de um mesmo produto em diferentes supermercados. Muitas vezes, supermercados que estão a apenas 100 metros de distância, chegam a apresentar uma diferença de até 70% no preço atribuído a uma mesma mercadoria (EXAME, 2017). Em tempos de crise econômica, e, levando em consideração que um terço do orçamento de uma família é gasto, geralmente, em supermercados (EXAME, 2017), essa diferença entre os preços dos produtos acaba sendo bastante determinante.

Muitas pessoas, no início do mês, pegam suas listas de compras e saem pela cidade em busca do lugar mais barato a se comprar. Entretanto, depois de algum tempo realizando essa tarefa, os consumidores deixam de procurar e acabam indo nos supermercados mais próximos de suas casas, principalmente em função da comodidade e do custo reduzido de tempo e de locomoção.

Diante do apresentado, torna-se visível que os supermercados carecem de ferramentas que facilitem a vida dos consumidores na hora de buscar pelos produtos que lhe oferecem o melhor custo benefício.

A partir desse cenário, o presente trabalho propõe-se a abordar o seguinte problema de pesquisa: como encontrar o produto mais interessante (preço e distância) dentre muitos supermercados, levando em consideração minha localização atual?

## **1.1 OBJETIVOS**

A presente seção destina-se à apresentação do objetivo geral e dos objetivos específicos do trabalho.

### **1.1.1 Objetivo Geral**

Conforme mencionado, o principal objetivo do trabalho consiste no desenvolvimento de uma aplicação móvel colaborativa para busca e cadastro de preços de produtos de supermercados, traçando um comparativo entre distância e preço para identificar a melhor opção de compra. Em outras palavras, a aplicação baseia-se em um sistema de recomendação de itens para usuários. A fim de alcançar o objetivo proposto, pretende-se utilizar o framework React Native, a linguagem de programação JavaScript e a API do Google Maps.

### **1.1.2 Objetivos Específicos**

- Identificar e compreender como funciona a construção de uma aplicativo utilizando a plataforma do React Native, bem como a sua arquitetura;
- Compreender e vislumbrar novas possibilidades de utilização da API do Google Maps;
- Compreender e vislumbrar novas possibilidades de utilização dos chamados sistemas de recomendação;
- Traçar um comparativo trazendo o produto com o melhor custo-benefício, levando em consideração o deslocamento, o valor e outras variáveis.

## **1.2 JUSTIFICATIVA**

Entende-se que a solução proposta neste trabalho tende a contribuir não apenas para que os consumidores economizem tempo e dinheiro, aumentando a qualidade de vida dos

mesmos, mas também para a redução dos preços dos produtos ofertados e, ainda, para o aumento das opções de produtos acessados pelos usuários. Nessa conjuntura, é inevitável que a solução proposta estimule o aumento da livre concorrência de mercado entre os estabelecimentos, trazendo visibilidade àqueles supermercados que oferecem produtos mais baratos e de melhor qualidade.

Além disso, o tema foi escolhido também com o propósito pessoal do autor de aprender novas tecnologias que não são empregadas no seu dia-a-dia de trabalho, objetivando aprimorar sua lógica de programação e se familiarizar com ferramentas cada vez mais utilizadas no desenvolvimento de sistemas web e móveis.

## 2 ABORDAGEM TEÓRICA

Neste item será apresentado toda a base teórica que dará uma maior perspectiva a essa pesquisa. Será exposta as tecnologias que foram empregadas no desenvolvimento da aplicação bem como seus conceitos.

No item 2.1 será apresentado o framework React, no item 2.2 mostramos o React Native, que é a plataforma base para o desenvolvimento da aplicação front-end. No item 2.3 apresenta-se a plataforma Node.Js no qual será escrito o back-end da aplicação. No item 2.4 exhibe a linguagem JavaScript, que é a linguagem que serve para o back-end bem como o front-end. No item 2.5 é explicado o funcionamento de uma API bem como a API do Google Maps que utilizaremos para a navegação de mapas. No item 2.6 é explanado o Visual Studio Code, IDE utilizada para desenvolvimento, por fim no item 2.7 são apresentados alguns trabalhos relacionados.

### 2.1 REACT

O React é um framework open-source de JavaScript para a criação de interfaces de usuário. Foi lançado pelo Instagram em 2013, ele facilita manipulação do DOM e por consequência disso também facilita a criação da GUI e na construção de componentes da web.

Com o React é possível criar componentes encapsulados que gerenciam o seu próprio estado, tornando mais fácil criar interfaces complexas.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="World!" />,
  mountNode
);
```

**Figura 1.** Exemplo de Código escrito em React.

Disponível em: [React](#)

A Figura 1 ilustra um exemplo de código feito em React.

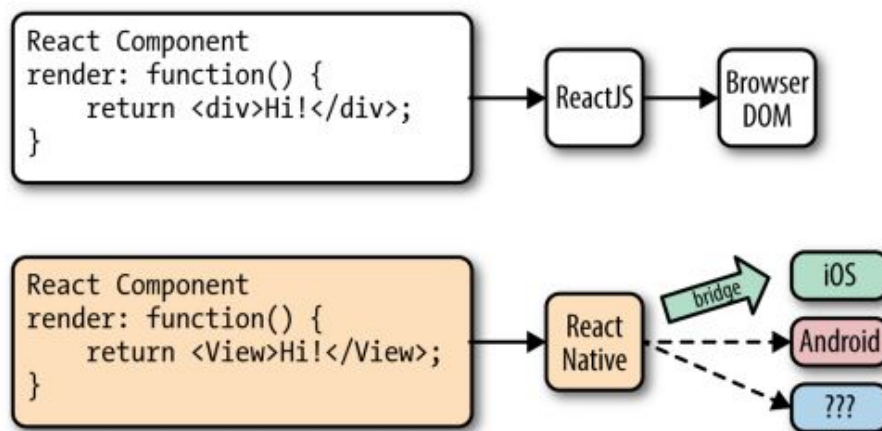


Uma das premissas do React é “Aprenda uma vez, escreva em qualquer lugar”, fazendo uma clara referência ao React Native que veremos a seguir.

## 2.2 REACT NATIVE

React Native é um framework, criado pelo Facebook, no qual possui várias ferramentas que facilitam o desenvolvimento. Seu objetivo principal é a criação a aplicações móveis nativas para os sistemas operacionais Android e iOS. O RN utiliza o React para estruturar as views da aplicação.

A figura 2 representa um modelo de como são renderizados os componentes criados em React e React Native.



**Figura 2.** Comparação de renderização React e React Native.  
Disponível em: [EISENMAN](#)

Uma característica muito forte deste framework é que pode ser escrito apenas um código em JavaScript para rodar em ambas plataformas, tornando o custo de produção muito abaixo dos demais métodos de desenvolvimento móvel disponíveis no mercado.

Abaixo um exemplo de código escrito com React Native.

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

class WhyReactNativeIsSoGreat extends Component {
  render() {
    return (
      <View>
        <Text>
          If you like React on the web, you'll like React Native.
        </Text>
      </View>
    );
  }
}
```

```

<Text>
  You just use native components like 'View' and 'Text',
  instead of web components like 'div' and 'span'.
</Text>
</View>
);
}
}

```

**Figura 3.** Exemplo de código React Native  
Disponível em: [React Native](#)

O RN permite que seja usado um *live reload* que em vez de recompilar, ele apenas recarrega o aplicativo, podendo ver as modificações em tempo real e instantaneamente.

Outra vantagem do RN é a possibilidade de utilizar componentes nativos escritos em Java, Objective-C ou Swift, juntamente com os componentes escritos em React.

A tabela 1 representa uma comparação direta entre as principais linguagens e frameworks de desenvolvimento móvel.

COMPARATIVO FERRAMENTAS MOBILE	WEB APPS	NATIVO	HÍBRIDO	PWA	FLUTTER	REACT NATIVE
	Apps contruídas para rodar no navegador	Utilizam linguagem nativa em forma de app.	WebView renderizada em um app nativo.	Versão mobile de aplicação web.	Versão multiplataforma da Google com Dart.	JavaScript controlando interface nativa.
<b>FACILIDADE DE CRIAÇÃO</b> Curva de aprendizado, tempo para masterizar	😊	😞	😊	😊	😊	😊
<b>CUSTO DE PRODUÇÃO</b> Valor agregado aos projetos	😊	😞	😊	😊	😊	😊
<b>LINGUAGENS</b> Facilidade de dominar as linguagens	😊	😞	😊	😊	😞	😊
<b>COMPATIBILIDADE</b> Compatibilidade com todos dispositivos	😊	😞	😊	😞	😊	😊
<b>INTERFACE</b> É possível obter interface nativa? Ou parecida?	😊	😊	😊	😊	😊	😊
<b>API'S NATIVAS</b> Integração com câmera, acelerômetro, etc	😞	😊	😊	😊	😊	😊
<b>APP STORE (PAY/PUSH)</b> Conseguimos colocar o app na store?	😞	😊	😊	😞	😊	😊
<b>PERFORMANCE</b> Não tem descrição, é performance mesmo	😞	😊	😞	😊	😊	😊

www.rocketseat.com.br Rocketseat

**Tabela 1.** Comparativo Ferramentas Mobile  
Disponível em: [RocketSeat](#)

Diferente de outras alternativas para aplicações mobile que utilizam webviews para renderizar a aplicação, o React Native invoca as APIs de renderização nativas em Objective C, para iOS, ou Java para Android. Portanto, a aplicação será renderizada utilizando componentes nativos da plataforma (EISENMAN, 2016).

### **2.3 NODE.JS**

Muitos desenvolvedores gostam de utilizar Javascript no front-end das aplicações. No nosso navegador manipulando as ações dos usuários, realizando validações, animações entre outras coisas.

O Node.JS é um processador que nasceu justamente para suprir com a necessidade de utilizar JavaScript no back-end das aplicações, rodando no lado do servidor, não apenas no lado do cliente.

Tendo em vista esse cenário, conseguimos desenvolver aplicações mais rápidas e escaláveis em um tempo menor de desenvolvimento. Por se tratar de uma tecnologia muito eficiente, ela é ideal para aplicações em tempo real, com troca intensa de dados através de dispositivos distribuídos.

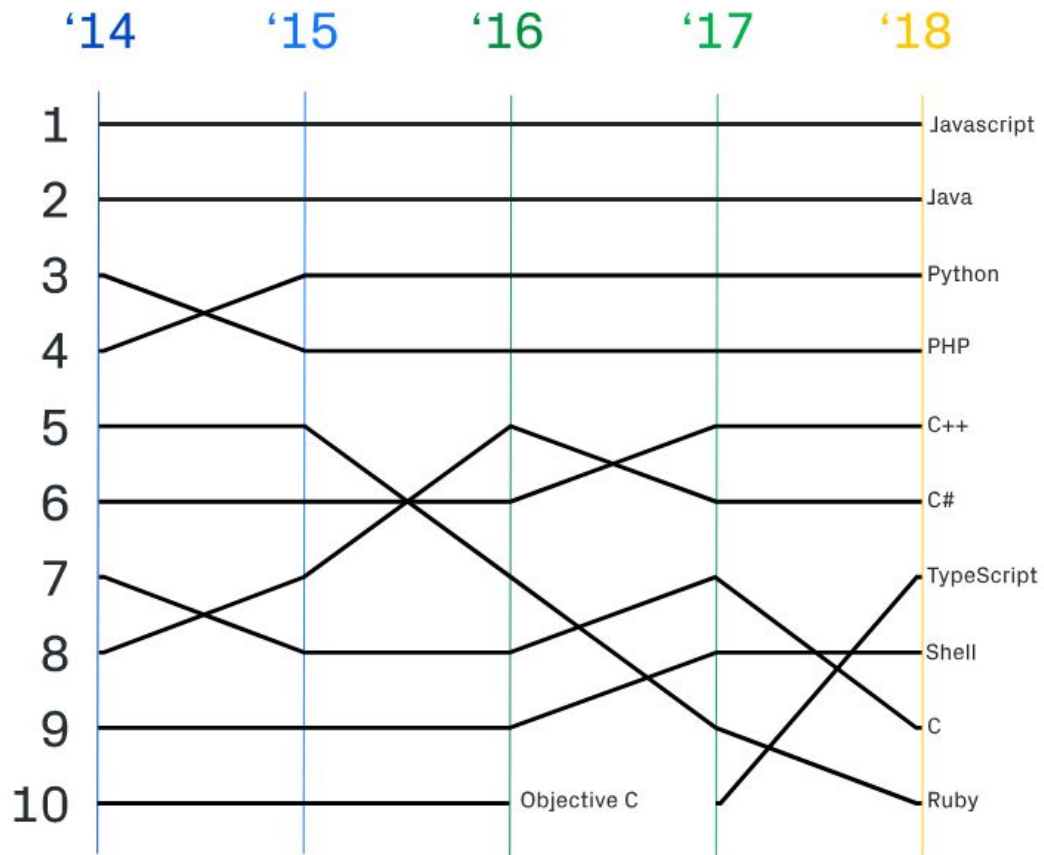
O Node.Js foi criado a partir do Engine V8 da Google, que é o motor de processamento usado no navegador Google Chrome, esse motor tem como responsabilidade ler e interpretar JavaScript.

### **2.4 JAVASCRIPT**

O JavaScript nasceu por volta de 1995, seu principal objetivo era manipular as validações de entradas que anteriormente eram feitas com linguagens no lado do servidor como Java e PHP. O JavaScript fez a atual geração de aplicativos baseados na web possível em navegadores de e-mail, mapas e redes sociais, e também é usado em locais mais tradicionais por fornecer diversas formas de interatividade (HAVERBEK, 2014).

Hoje em dia o JavaScript é usado em grande escala por diversas empresas de tecnologia para desenvolver seus produtos.

A tabela 2 representa um ranking de linguagens mais utilizadas dentro do Github.



**Tabela 2.** Ranking de linguagens de programação mais utilizadas na plataforma do Github. Disponível em: [Github](#)

Como podemos observar na tabela 2, desde 2014 o JavaScript se mantém na primeira posição das linguagens mais utilizadas dentro da plataforma do Github.

Apesar de ser uma linguagem antiga, o JavaScript é bastante requisitado e parte desse sucesso se deve a sua simplicidade (SALLES, 2018).

## 2.5 API DO GOOGLE MAPS

A API JavaScript do Google Maps permite personalizar mapas com seu próprio conteúdo e imagens para exibição em páginas da Web e dispositivos móveis (GOOGLE, 2018).

Para o nosso projeto, a API do Google Maps é ideal, pois ela apresenta quatro tipos básicos de mapas (roteiro, satélite, híbrido e terreno) que você pode modificar usando camadas e estilos, controles e eventos, além de vários serviços e bibliotecas (GOOGLE, 2018).

A figura 4 mostra o código de como é feito um simples *Hello World* para demonstrar o funcionamento da API e na outra figura 5 temos o resultado do código.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="/maps/documentation/javascript/cgc/demos.css">
  </head>
  <body>
    <div id="map"></div>
    <script>
      function initMap() {
        var myLatLng = {lat: -25.363, lng: 131.044};
        var map = new
google.maps.Map(document.getElementById('map'), {
          center: myLatLng,
          zoom: 4
        });
        var marker = new google.maps.Marker({
          map: map,
          position: myLatLng,
          title: 'Hello World!'
        });
      }
    </script>
    <script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callb
ack=initMap"
      async defer></script>
  </body>
</html>
```

**Figura 4.** Exemplo de código para criar um mapa do Google.  
Disponível em: [Google](#)

Como podemos ver na figura 4, para API funcionar e fazermos todas a requisições, precisamos informar uma chave de acesso que conseguimos acessando o console de desenvolvedores do Google em [www.console.developers.google.com](http://www.console.developers.google.com), após adquirir a chave de acesso podemos realizar todas as funcionalidades desejadas na aplicação.



**Figura 5.** Resultado do exemplo de mapa do Google.  
Disponível em: [Google](#)

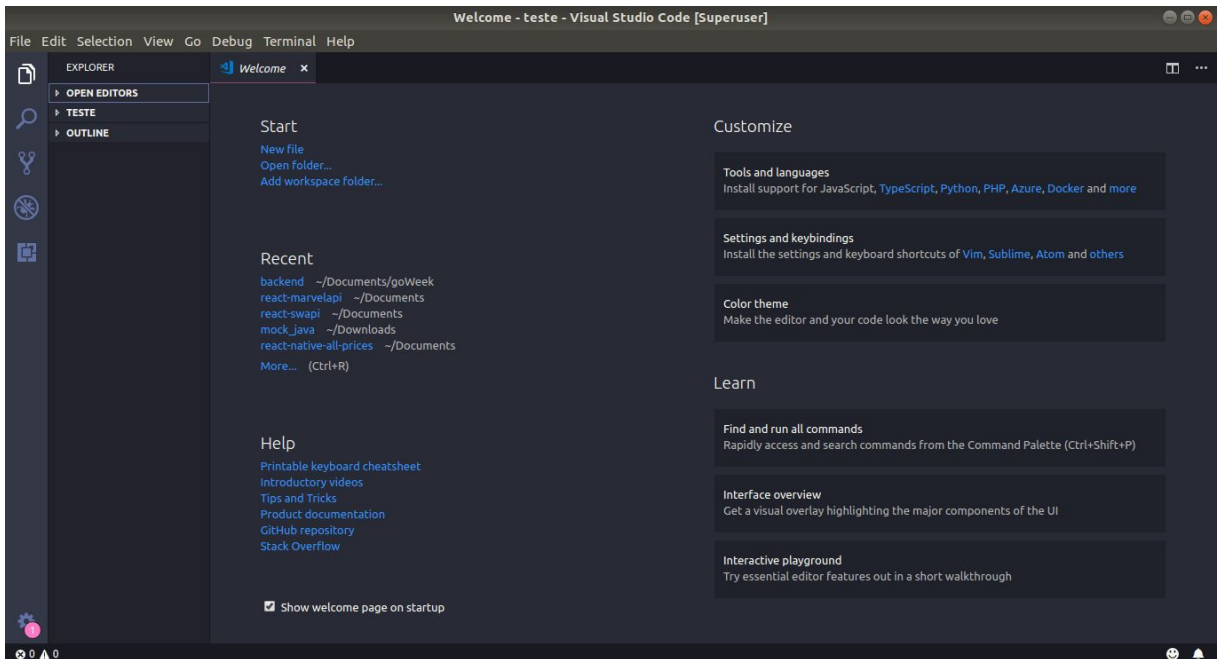
## 2.6 VISUAL STUDIO CODE

O Visual Studio Code, popularmente conhecido como VSCode, é uma IDE *open-source* da Microsoft. Essa IDE possui muitas funcionalidades que ajudam no desenvolvimento.

O Visual Studio Code é um editor de texto multiplataforma disponibilizado pela Microsoft para o desenvolvimento de aplicações web, com suporte principalmente a ASP.NET 5 e Node.js (DIONISIO, 2016).

VSCode tem cara de editor como no Sublime Text sem perder funcionalidades de IDE como debug integrado com breakpoints, integração nativa com Git e autocomplete muito rápido (FERNANDES, 2018).

A figura 6 ilustra a última versão da ferramenta.



**Figura 6.** Página inicial do VSCoDe

Fonte: Autor

## 2.7 TRABALHOS RELACIONADOS

Nesta seção serão abordados alguns trabalhos relacionados ao projeto proposto, os quais auxiliaram no desenvolvimento deste trabalho.

O primeiro aplicativo que encontramos semelhança com o nosso é o PagPouco, no qual faz a comparação de preços de supermercado, porém esse aplicativo está limitado para a região sul de São Paulo e Santos (EXAME, 2016).

O segundo aplicativos que encontramos é o ComboBox, ele é baseado no conceito da economia colaborativa, em que usuários podem postar atualizações de preço e compartilhar com amigos. A diferença entre esse aplicativo e o nosso, é que ele não possui a visualização no mapa para identificar os supermercados.

## 3 METODOLOGIA

No primeiro momento, foi realizado um estudo sobre aplicativos com funcionalidades parecidas e a viabilidade da aplicação. Depois, foi feito um levantamento sobre as tecnologias que fundamentam a criação de aplicações móveis, dentro desse contexto, faz-se um estudo sobre os padrões de desenvolvimento do JavaScript usados pelos frameworks e bibliotecas.

Este trabalho fez uso da metodologia qualitativa do tipo experimental, pois de acordo com Minayo (2001) a pesquisa qualitativa não se baseia no critério numérico para garantir sua representatividade. A amostra adequada é a que possibilita abranger a totalidade do problema investigado em sua múltiplas abordagens.

Quanto aos objetivos, trata-se de uma pesquisa experimental, onde “determinamos um objeto de estudo, selecionamos as variáveis que seriam capazes de influenciá-lo, definimos as formas de controle e de observação dos efeitos que a variável produz no objeto” (PRODANOV; FREITAS, 2013).



## 4 DESENVOLVIMENTO DO PROJETO

Para alcançar o objetivo proposto e solucionar o problema de pesquisa foi desenvolvido um aplicativo para Android, o qual pode ser compilado também para a plataforma iOS. Ao aplicativo, de-se o nome de *allPrices* (“todos os preços” na tradução livre para o português).

Para o desenvolvimento do projeto, a principal linguagem de programação utilizada foi o JavaScript, juntamente com o *framework* React Native. Para garantir a usabilidade e o desempenho da aplicação, adotou-se um estilo de arquitetura recomendado pelo *framework*.

Já para a exibição do mapa, optou-se pela utilização da API do Google Maps, a qual possui muitas facilidades para o desenvolvedor. Além de uma documentação completa, possui suporte para diversas linguagens de programação, incluindo JavaScript. As API’s do Google Maps utilizadas foram a Maps SDK para Android e a Places API. Segundo o Google, com a Places API, possuímos acesso a dados públicos de mais de 100 milhões de estabelecimentos.

No desenvolvimento do aplicativo em si, foram utilizadas algumas bibliotecas de terceiros para facilitar o desenvolvimento. A biblioteca AXIOS, por exemplo, foi utilizada para realizar requisições HTTP e fazer as buscas em geral da aplicação. A biblioteca React Native Maps, por sua vez, foi usada para se ter total controle e manipulação do mapa da aplicação. Para a navegação do usuário, optou-se pela biblioteca React Navigation. Por fim, para a padronização de código e desenvolvimento, usou-se a ferramenta ESLint.

## **5 RESULTADOS**

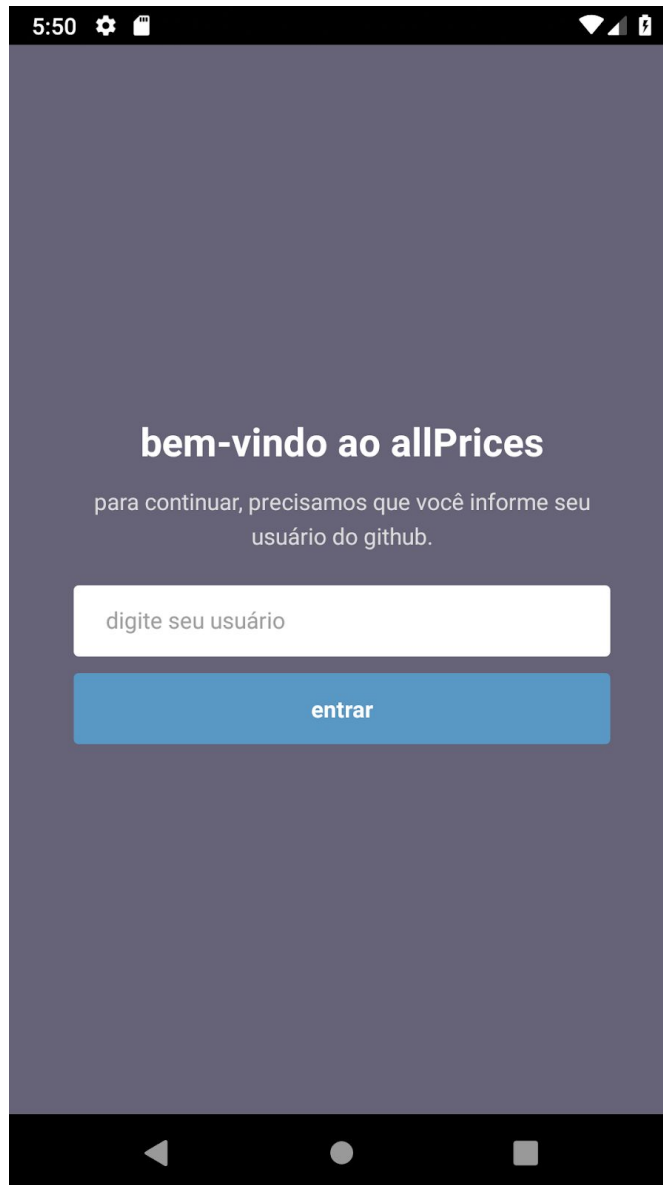
Esta seção apresenta os resultados obtidos neste trabalho. Abaixo serão apresentados os resultados do produto desenvolvido, um aplicativo multiplataforma para os sistemas operacionais Android e iOS.

### **5.1 TELA DE ACESSO**

A primeira tela da aplicação, é a tela de acesso, ela possui o nome do aplicativo e um texto informando que o usuário precisa digitar o nome de usuário do github para acessar o aplicativo.

Ao centro da tela, observamos que existe uma campo de entrada e um botão para acesso ao aplicativo.

A figura 7 demonstra a tela de acesso do aplicativo.



**Figura 7.** Tela de acesso ao aplicativo  
Fonte: Autor

Após informar o usuário e tocar no botão entrar, exibimos um *loader* no botão para melhorar a experiência do usuário e informar que está carregando.

A figura 8, após inserido o usuário e tocado o botão acessar, exibe um *loader*.



**Figura 8.** Página inicial do aplicativo com o *loader*  
Fonte: Autor

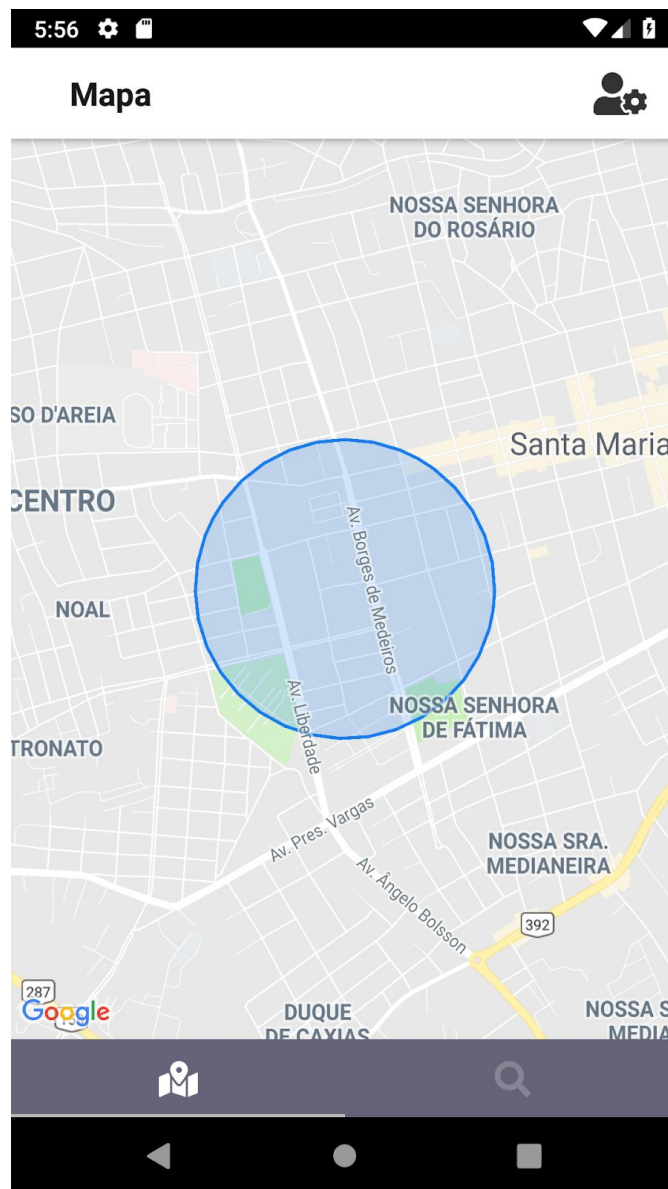
## 5.2 TELA INICIAL

A próxima tela, é a tela inicial do aplicativo após o login. Nessa tela podemos ver no canto superior esquerdo o nome da aba que estamos, neste caso é a aba “Mapa”, no canto superior direito existe um ícone de usuário com engrenagem, ao clicar nesse ícone, o aplicativo nos leva a tela de configurações do usuário.

Na parte inferior do aplicativo, existem dois botões que fazem parte da navegação por abas, na primeira aba, possui um ícone de mapa destacado com a cor branca para identificar a aba que estamos. A segunda aba é um ícone de lupa, ao clicarmos nesse ícone, o aplicativo realiza a mudança de aba para a aba de busca do usuário, nessa aba, o usuário poderá informar seus filtros de busca.

Na parte central do aplicativo, podemos ver o mapa com um círculo central, esse círculo é o *range* de busca do aplicativo, por padrão essa busca é 500 metros mas podem ser alterados na tela de perfil que veremos logo em seguida. Esse círculo destaca a posição atual do usuário, o usuário não tem permissão de mover o mapa para os lados ou dar zoom.

Na figura 9 é apresentado a página inicial do aplicativo.



**Figura 9.** Primeira aba do aplicativo.

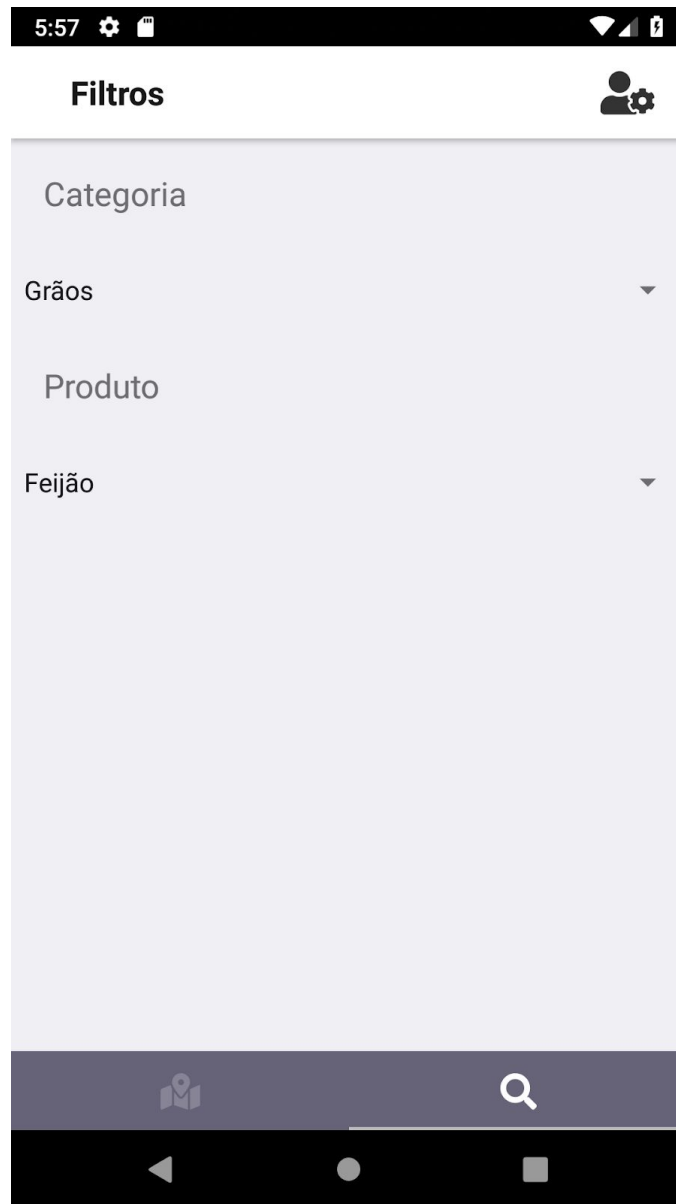
Fonte: Autor

### 5.3 TELA DE FILTROS

Nessa tela, o usuário poderá informar os filtros que deseja para realizar a busca, como esse projeto é uma aplicação apenas para solucionar o problema presente neste trabalho, usamos dados fictícios.

No primeiro filtro o usuário escolhe a categoria do produto, no segundo filtro, o usuário escolhe o produto em si.

Na figura 10, abaixo, mostra a escolha dos filtros pelo usuário.



**Figura 10.** Segunda aba do aplicativo.  
Fonte: Autor

## 5.4 TELA DE PERFIL DO USUÁRIO

Ao clicar no ícone do perfil no canto superior direito, o aplicativo *navega* até a tela do perfil do usuário.

No topo da página, existe um seta para retorno até a tela onde foi feita a primeira ação, do lado da seta, está o nome da tela, logo em baixo, temos a imagem do usuário, do lado da imagem, temos o nome do usuário, seguido de uma breve descrição. Todas essas informações sobre o usuário, foram retiradas do Github a partir do nome de usuário informado na tela de acesso.

A figura 11 demonstra a tela de configurações do usuário, bem como todas as suas configurações padrões.



**Figura 11.** Tela de perfil do usuário com informações padrão.  
Fonte: Autor

Em baixo das informações do usuário, temos a seção de preferências, onde o usuário pode escolher o *range* de busca dos produtos. Por padrão o *range* é de 500 metros.

Ao centro inferior, existe um botão em vermelho que realiza saída do usuário da aplicação.

Na figura 12, podemos ver a alteração do *range* de distância de busca, no qual o simulamos que o usuário tenha feito a mudança para 800 metros.



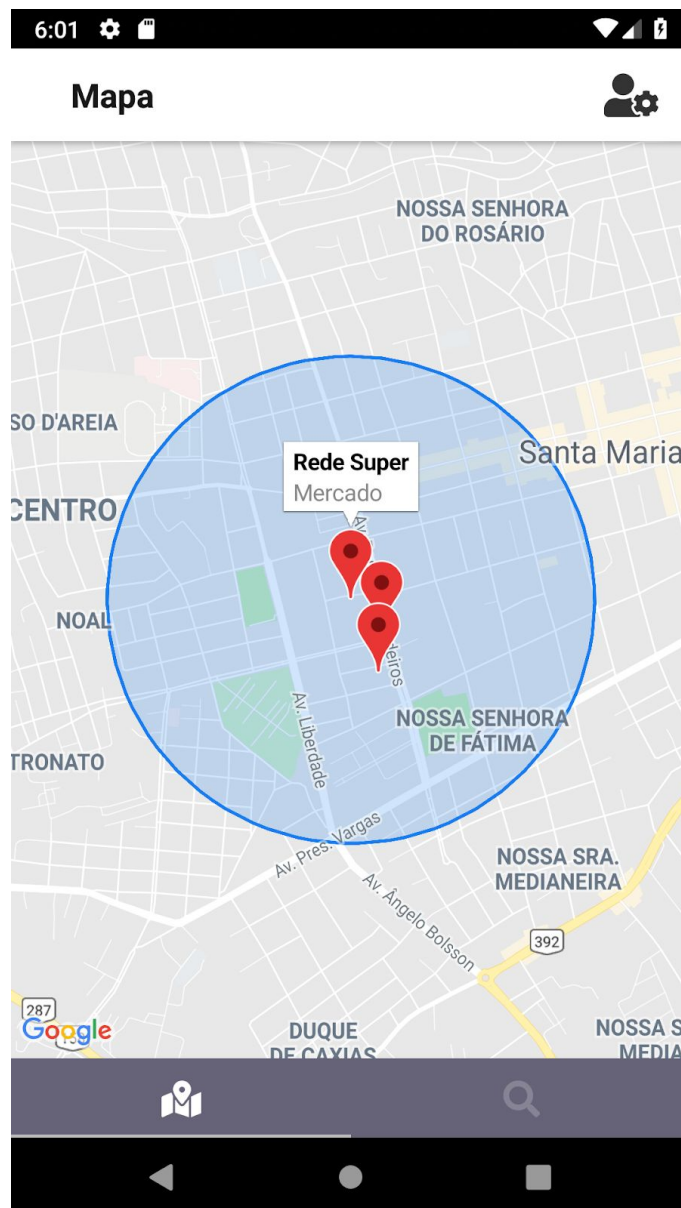
**Figura 12.** Tela de perfil do usuário com informações modificadas.  
Fonte: Autor



## 5.5 RESULTADO DA CONSULTA

Após informar todos os filtros de pesquisa e o *range* da busca, o usuário retorna para a tela inicial onde vão estar demarcados no mapa todos os estabelecimentos mais próximos e com os produtos mais baratos.

Na figura 13 podemos ver os estabelecimentos demarcados dentro do mapa, também podemos notar que a distância de busca aumentou, pois o usuário havia selecionado o range de 800 metros.



**Figura 13.** Resultado da consulta de produtos.  
Fonte: Autor

## 6 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou o desenvolvimento de uma aplicação móvel que auxilia os usuários na busca pelos produtos de supermercados mais baratos e mais próximos, de acordo com suas preferências e localização atual. Em outras palavras, o resultado do trabalho consiste em um sistema de recomendação de itens para usuários, operando de maneira colaborativa na busca e no cadastro de produtos de supermercados, traçando um comparativo entre distância e preço para identificar a melhor opção de compra. Dentre os objetivos especificados para a pesquisa proposta, o autor avalia que todos eles foram alcançados de maneira satisfatória.

No que diz respeito à compreensão das ferramentas selecionadas para a elaboração do projeto (framework React Native e API do Google Maps), entende-se que o desenvolvimento da aplicação proposta possibilitou a construção de um conhecimento sólido, o qual permite que o autor trabalhe de maneira bastante criativa na construção de futuros aplicativos.

Já a respeito da construção do sistema de recomendação em si, avalia-se, também, que a aplicação móvel multiplataforma desenvolvida com o framework React Native e algumas bibliotecas de suporte, utilizando-se de informações como filtros de preferência e localização, opera de maneira satisfatória como um sistema de recomendação simples e, além disso, supre as necessidades do usuário.

### 6.1 TRABALHOS FUTUROS

Seguem abaixo, algumas sugestões de trabalhos futuros a serem desenvolvidos sobre a aplicação desenvolvida:

- Desenvolvimento da tela de cadastro de produtos para os usuários. Atualmente a aplicação possui os dados cadastrados diretamente na aplicação;
- Adição de categorias e filtros para facilitar a busca do usuário: por exemplo, filtros de peso, unidades e setor do supermercado;

- Adicionar um histórico de buscas para o usuário: com isso, o usuário teria mais facilidade quando desejasse pesquisar por um produto já buscado anteriormente;
- Permitir o compartilhamento de buscas de produtos específicos através de redes sociais: com isso, o usuário poderia compartilhar com outras pessoas algum produto de seu interesse ou de interesse da pessoa com a qual o produto está sendo compartilhado;
- Melhorar a experiência do usuário com animações e efeitos no aplicativo. Hoje, ele está visualmente agradável, mas precisa de um identidade, como logo, ícones diferenciados no mapa, etc;
- Adicionar um meio de monetização para a aplicação: por exemplo parceria com os supermercados para que os mesmos possam expor suas ofertas e anúncios dentro do aplicativo;
- Desenvolvimento do cadastro de usuário e login com redes sociais: por exemplo, login com Facebook ou Google.

## REFERÊNCIAS

NODE.JS FOUNDATION. **About Node.js**. Disponível em: <<https://goo.gl/hrNqPN>>. Acesso em: 31 jul. 2018.

GOOGLE DEVELOPERS. **Google's high performance, open source, JavaScript engine**. Disponível em: <<https://goo.gl/2W7gbh>>. Acesso em: 31 jul. 2018.

DEV MEDIA. **Como começar com React Native?** Disponível em: <<https://goo.gl/4qTLNu>>. Acesso em: 13 nov. 2018.

FACEBOOK INC. **A JavaScript library for building user interfaces**. Disponível em: <<https://goo.gl/y6mn1s>>. Acesso em: 13 nov. 2018.

FACEBOOK INC. **Build Native Mobile apps using JavaScript and React**. Disponível em: <<https://goo.gl/JXAFqb>>. Acesso em: 13 nov. 2018.

FERNANDES, Diego. **Ambiente Javascript: Dicas, VSCode e terminal**. Disponível em: <<https://goo.gl/QAMvwe>>. Acesso em: 13 nov. 2018.

DIONISIO, Edson. **Introdução ao Visual Studio Code**. Disponível em: <<https://goo.gl/qqDsL8>>. Acesso em: 13 nov. 2018.

HAYERBEK, Marijn. **Eloquent Javascript - 2 edição**. Disponível em: <<https://goo.gl/b9ymuS>>, Acesso em: 18 nov. 2018.

GOOGLE DEVELOPERS. **API JavaScript do Google Maps**. Disponível em: <<https://goo.gl/XpgT2B>>, Acesso em: 19 nov. 2018.

EISENMAN, Bonnie. **Learning React Native**. Disponível em: <<https://goo.gl/t88aPw>>, Acesso em: 19 nov. 2018.

MINAYO, Maria Cecília de Sousa. **Pesquisa social: Teoria, método e criatividade.** Disponível em <<https://goo.gl/HvFo46>>. Acesso em: 20 nov. 2018.

SALLES, Felipe. **Top 10 linguagens de programação mais usadas no mercado.** Disponível em: <<https://goo.gl/b4rccl>>. Acesso em: 20 nov. 2018.