VI Congress of Industrial Management and Aeronautical Technology

ISSN 2447-5378

DESENVOLVIMENTO DE UMA PLATAFORMA PARA DISPOSITIVOS MÓVEIS PARA APROXIMAR PRODUTORES E CONSUMIDORES DE PRODUTOS ORGÂNICOS.

E. S. M. Filho[1, *]; A. F. Souza[1]

1 Faculdade de Tecnologia de Jacareí - Professor Francisco de Moura
Avenida Faria Lima, 155 – Jardim Santa Maria, Jacareí/SP, CEP.: 12.328 - 150, Brasil.
Telefone: (12) 3953-7926

* eduardosmf_1998@hotmail.com

RESUMO**:** Atualmente, há um aumento na produção de produtos orgânicos, bem como do interesse de brasileiros em consumir produtos orgânicos e frescos. Porém, o consumo é limitado pela baixa disponibilidade e pelo custo superior, essa despesa pode ser atribuída aos custos do intermediário, graças a maior perecibilidade dos produtos. O objetivo deste trabalho é criar um Aplicativo Web Progressivo, ou seja, um Aplicativo Web que funciona da mesma forma de um aplicativo nativo para dispositivos Android e iOS, para os produtores anunciarem os seus produtos e os consumidores localizarem e entrarem em contato com os produtores. A ideia é fomentar pessoas comuns a produzirem e anunciarem produtos naturais. Esta aplicação fornece uma interação agradável ao usuário, para o produtor proporciona gerenciar a disponibilidade de produtos e verificar visualizações de perfil. Para o cliente possibilita utilizar um mapa para visualizar quais produtores estão próximos dele e filtrá-los.
PALAVRAS-CHAVE**:** produto orgânico, marketplace, Aplicativo Web Progressivo.

ABSTRACT**:** Currently, there is an increase in the production of organic products, as well as the interest of Brazilians in consuming organic and fresh products. However, consumption is limited by the low availability and the higher cost, this expense can be attributed to the intermediary costs, thanks to the greater perishability of the products. The goal of this work is to create a Progressive Web Application, that is, a Web Application that works the same way as a native App for Android and iOS devices, for producers to advertise their products and consumers to locate and contact producers. The idea is to encourage ordinary people to produce and advertise natural products. This application provides a user friendly interaction, for the producer provides to manage product availability and check profile views. For the client, it is possible to use a map to visualize which producers are close to him and filter.

KEYWORDS**:** organic product, marketplace, Progressive Web App

1. INTRODUÇÃO

Segundo Liu (2017) diretor da ORGANIS (Conselho Nacional da Produção Orgânica e Sustentável), em 2017 o Brasil tinha 15.519 unidades produtivas, constituídas por produtores e empreendedores individuais, cooperativas, unidades de produção associativas e empresas de processamento e industrialização de produtos orgânicos.

De acordo com o SEBRAE (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas) o Brasil vem se consolidando como grande produtor e exportador de alimentos orgânicos, dados do

ISSN 2447-5378

MDA (Ministério do Desenvolvimento Agrário) apontam que o Brasil cresce 20% ao ano na produção nacional de produtos orgânicos (SEBRAE, 2017).

Dados de uma pesquisa divulgada pela ORGANIS publicada na Agência Brasil, mostra que 15% da população urbana consumiu algum produto orgânico durante os dois meses de pesquisas. Nesta pesquisa 64% das pessoas relatam que consomem esse tipo de produto por ser mais saldável e os demais por indicação médica (BOEHM, 2017).

Segundo Murate (2017), "a agricultura orgânica está ganhando cada vez mais espaço nas mesas dos brasileiros", isso acontece, pois está tendo um "aumento na demanda dos consumidores por produtos mais saudáveis e pela conservação do meio ambiente". Por esse aumento, o preço vem caindo e aumentando o acesso de pessoas com menor poder aquisitivo a esses tipos de produtos.

Mas segundo Boehm (2017) esses produtos são basicamente oferecidos a valores caros em supermercados, feiras e lojas especializadas. O motivo disso acontecer é que os produtores necessitam de certificação adequada para poder disponibilizar os seus produtos no mercado, segundo o Art. 3º da Lei 10.831 (BRASIL, 2003). Por ser difícil de obtê-la, os produtores se unem a esses fornecedores que já possuem uma certificação ou que facilitam a obtenção para conseguir vender seus produtos, fazendo com que os fornecedores ditem os preços no mercado.

Com esses fornecedores agindo como intermediários na venda, faz com que o consumidor perca contato com o produtor, causando problemas, tais como, aumento no preço e perda na qualidade do produto, já que, existe um tempo entre o produto ser colhido e fornecido ao consumidor. Por ser um produto natural, segundo Recine e Radaelli (2018), após ser colhido esses produtos perdem valor nutritivo com maior facilidade, para aproveitar ao máximo os nutrientes deve-se consumir enquanto estão frescos. De acordo com Boehm (2017) na sua reportagem Ming Liu diretor da ORGANIS diz que comprar direto do produtor, ainda é uma promessa.

Da mesma forma, onde os aplicativos de transporte aproximaram as pessoas com necessidade de transporte das pessoas comuns que possuem tempo e veículos ociosos. O objetivo deste trabalho é criar um Aplicativo Web Progressivo, ou seja, um Aplicativo Web que funciona da mesma forma de um aplicativo nativo para dispositivos Android e iOS, para os produtores anunciarem os seus produtos e os consumidores localizarem e entrarem em contato com os produtores. Essa aproximação pode fazer a inserção de pessoas sem ocupação no mercado de trabalho, fomentar um mercado em potencial e mitigar a ação de intermediários que aumentam o custo final e diminuem os valores nutricionais dos produtos.

2. MATERIAIS E MÉTODOS

Com esta aplicação os produtores poderão cadastrar os seus produtos, junto com a localização espacial, para ser visualizada num mapa interativo. Para iniciar o desenvolvimento desta aplicação, foi adotado como nome a palavra "AgroProdutor".

2.1. Área de Estudo

O aplicativo AgroProdutor poderá ser utilizado por qualquer pessoa em diferentes localidades, mas na fase de testes será considerado apenas o município de Jacareí – SP.

2.2. Codificação do Aplicativo

ISSN 2447-5378

A aplicação foi codificada usando as tecnologias Web para a construção de páginas para navegador. Porém, o projeto Web precisa ser estruturado da forma PWA (Progressive Web App - Aplicativo Web Progressivo) para poder ser instalado no dispositivo.

2.3. Progressive Web App

De acordo com a documentação da Google, aplicações em PWA são uma relação do melhor da Web e melhor dos aplicativos (GOOGLE, 2019). Criando a sensação de que o usuário está usando um aplicativo nativo, mas na verdade está acessando uma página Web. Conforme a documentação da Google o PWA proporciona as seguintes funcionalidades:

- Progressivo - qualquer usuário pode usar, em diferentes tipos de navegador;
- Responsivo - adapta-se aos formatos de diferentes tipos de dispositivos;
- Offline – é integrado com *service workers* para funcionar off-line ou em redes de baixa qualidade;
- Semelhante a aplicativos – seu design é semelhante a aplicativos nativos;
- Atualizado – sempre que necessário, pode-se atualizar a aplicação instantaneamente, sem precisar publicar em uma loja de aplicativos, graças ao processo de atualização do *service worker*;
- Seguro – por ser via HTTPS (Hyper Text Transfer Protocol Secure - Protocolo de Transferência de Hipertexto Seguro) evita invasões, garante um conteúdo seguro e sem adulterações;
- Reconhecido como aplicativo – seu dispositivo o reconhece como "aplicativo", pois contém um arquivo Manifesto W3C (World Wide Web Consortium) e um escopo de registro do *service worker*, usado para ser encontrado pelos mecanismos de pesquisa;
- Envolvente – facilita o engajamento com recursos como notificações *push*;
- Instalável - permite a instalação na tela inicial do seu dispositivo sem precisar acessar uma loja de aplicativos;
- Compartilhável – compartilhado facilmente por URL (Uniform Resource Locator), não requer instalação complexa.

2.4. App Shell

De acordo com a documentação da Google (2019) "o App Shell é o mínimo de HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto), CSS – (Cascading Style Sheets - Folha de Estilo em Cascatas) e JavaScript necessário para fornecer energia à interface do usuário de um PWA e é um dos componentes que garante um bom desempenho".

O App Shell possui uma arquitetura que separa a infraestrutura base do aplicativo e a UI (Interface do Usuário) dos dados, pois eles são armazenados localmente em um cache usando *service worker*, para que só os dados necessários sejam carregados (GOOGLE, 2019). Segundo Abreu (2018) "a utilização da arquitetura de shell de aplicativo faz com que você forneça ao seu PWA propriedades que podemos encontrar em aplicativos nativos, tais como: carregamento instantâneo e a disponibilidade de atualizações sem a necessidade da loja de apps".

2.5. Service Worker

É um script executado em segundo plano pelo navegador, separado da página web, que melhora a experiência do usuário deixando semelhantes aos aplicativos nativos (GOOGLE, 2019).

Abreu (2018) diz que o *service worker* impede todas as requisições HTTP realizadas pelo aplicativo e escolhe a dinâmica para responder a elas. Onde, ele pode consultar o cache e dar uma reposta já armazenada, se existir. Não se limitando apenas a requisições feitas por meio de APIs, mas também recursos referenciados em HTML e até mesmo a requisição inicial do arquivo "index.html".

ISSN 2447-5378

2.6. Manifesto

É um arquivo JSON (JavaScript Object Notation) necessário para controlar a aparência da aplicação na área do menu ou tela inicial no dispositivo do usuário (GOOGLE, 2019). Onde proporciona os seguintes itens:

- Insere um ícone na tela de menu no dispositivo do usuário;
- É iniciado no Android sem uma barra de URL;
- Pode mudar a orientação da tela, como em um aplicativo nativo;
- Define uma tela de apresentação ao iniciar a aplicação;
- Os recursos são exibidos de forma padrão no navegador para evitar mudanças bruscas quando os recursos do site ficam disponíveis.

2.7. Arquitetura Cliente–Servidor

Está aplicação tem como serviço um Web Service em Nodes.js que interage com o aplicativo e o banco de dados.

O Nodes.js é uma aplicação criada em JavaScript com eventos assíncronos, projetado de forma que usuários do Node estejam livres de preocupações com o bloqueio do processo (NODE, 2019).

O Node é semelhante em design e instigado por sistemas como o *Ruby Event Machine* ou *Python's Twisted*. O Node eleva o modelo de evento, mostrando loops de eventos como uma construção de tempo de execução, em vez, de uma biblioteca (NODE, 2019).

Para criação da aplicação foram utilizadas algumas APIs (*Application Programming Interface* - Interface de Programação de Aplicações) e frameworks que ajudam o desenvolvimento, sendo um deles o Express. Esse framework é utilizado em aplicativos webs junto ao Node.js e ele é maleável, minimalista e propicia um conjunto robusto de recursos. O Express concede uma camada fina de recursos essenciais para aplicativos da web, sem obscurecer os recursos do Node.js (EXPRESS, 2019).

2.8. Sistema Gerenciador de Banco de Dados (SGBD)

Para o armazenamento dos dados no servidor foi utilizado o SGBD PostgreSQL 10 com a extensão espacial PostGIS, para dar suporte aos tipos de dados e consultas espaciais. O PostgreSQL é um SGBD objeto-relacional livre que usa e estende a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) ajustadas com muitos recursos que armazenam e dimensionam com segurança (POSTGRESQL, 2019).

O PostGIS é um extensor de banco de dados espacial para o SGBD PostgreSQL. Ele inclui suporte a objetos geográficos, no qual proporciona fazer consultas de localização usando SQL (POSTGIS, 2019).

A Figura 1 mostra o modelo de dados com as seguintes tabelas e *views* da aplicação:

- Tabela de produtor – dados cadastrais dos produtores;
- Tabela de cliente – dados pessoais dos clientes;
- Tabela de produto – nomes dos produtos já cadastrados na plataforma;
- Tabela de produto por produtor – produtos associados a cada produtor;
- Tabela de visualização – armazenar os clientes que visualizaram o perfil de algum produtor;
- View de produto por produtor – junção dos dados essenciais das tabelas de produtor, produto e produto por produtor para facilitar o acesso através do Node.js;
- View de produtor – engloba dados importantes da tabela de produtor com acréscimo da coluna “geom” que contém os dados de geometrias do tipo “ponto”, para facilitar o acesso através do Node.js, que enviará para o mapa.

ISSN 2447-5378

As visões foram criadas somente para facilitar as consultas, por isso elas não são materializadas no banco, ou seja, não há registro armazenado no banco, elas simplesmente são modo de visualização rápida de várias tabelas juntas.

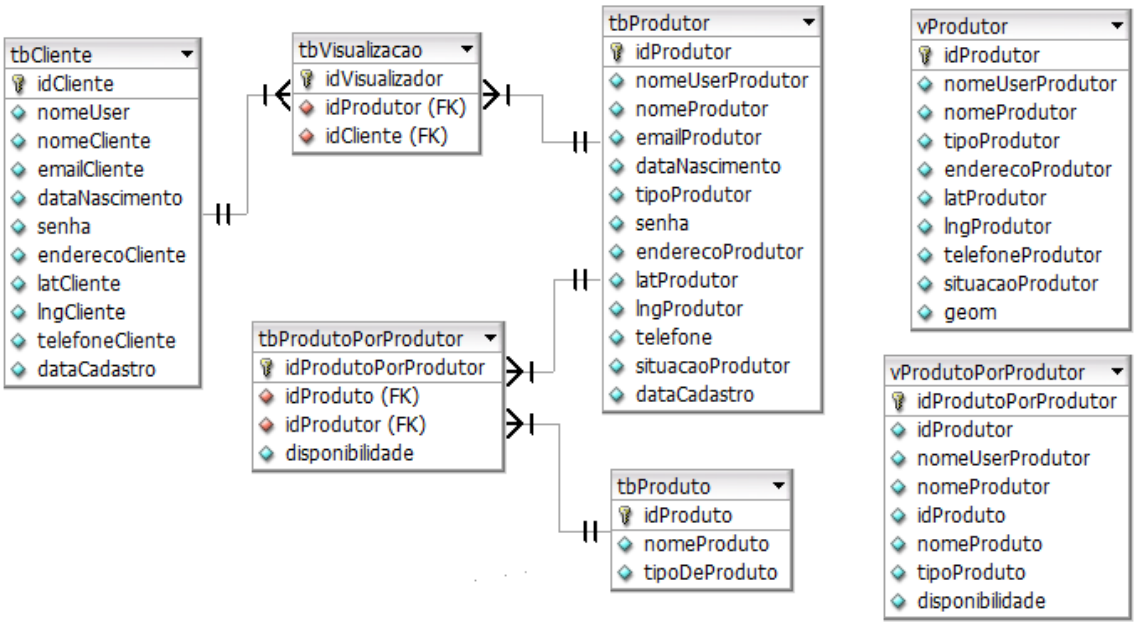

Figura 1. Modelo de dados (AUTOR, 2019).

2.9. Linguagens do Aplicativo

Por ser uma aplicação em PWA, as linguagens para o desenvolvimento deste aplicativo, foi composta basicamente por (HTML, CSS e JavaScript) na criação de layouts, eventos UI, entre outros. O HTML e CSS foram responsáveis pela parte de layout, ou seja, aparência do aplicativo, nesta parte foi utilizada a API Material Design Lite da Google, uma biblioteca que fornece toda a estrutura básica para criação do design do aplicativo. O JavaScript controla as ações do aplicativo, como a interação do usuário com o aplicativo e as requisições com o servidor. Para o desenvolvimento dos componentes diferenciais da aplicação como o mapa, foram usados recursos da API gratuita da Openlayers.

2.10. Material Design Lite

É uma biblioteca que permite criar uma aparência de “Material Design” da Google em páginas Web. Ele não depende de nenhum framework JavaScript e tem como objetivo otimizar o uso em vários dispositivos, oferecendo uma aparência mais elegante aos navegadores antigos e acessibilidade imediata (MDL, 2019).

2.11. Openlayers

É uma API para adicionar mapas dinâmicos em páginas Web. Pode exibir camadas de mapas, dados vetoriais e marcadores carregados de qualquer fonte. A Openlayers foi criada para incentivar o uso de informações geográficas de qualquer tipo. É uma biblioteca grátis, *open source* e JavaScript (OPENLAYERS, 2019).

2.12. Imagens do aplicativo

O aplicativo necessitava de algo que representa uma plantação. Para chegar mais próximo disso foi escolhido a imagem de um broto (Figura 2). Além disso, foram criados elementos para deixar o aplicativo mais exclusivo visualmente, como marcadores para o mapa e cartões de apresentação. As

ISSN 2447-5378

cores destes elementos foram baseadas nas cores primárias e secundárias desta aplicação, ou derivadas das mesmas.

Figura 2. Cartões, marcadores e ícone do Aplicativo (AUTOR, 2019).

2.13. Interface da Aplicação

O aplicativo possui 16 telas com a dinâmica ilustrada na Figura 3. O usuário irá abrir o App, escolher o tipo de acesso, se cadastrar, realizar o login de acesso e usufruir das funcionalidades pelo menu.

A Figura 4 e Figura 5 mostram algumas telas, no total são 16 telas, mas elas foram codificadas em 13 arquivos HTML.

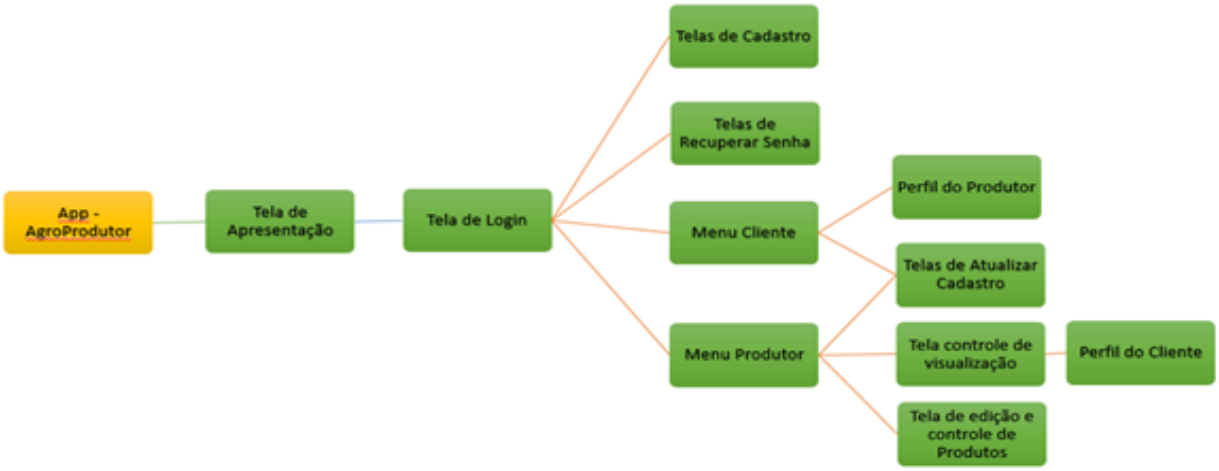

Figura 3. Fluxograma das telas do Aplicativo (AU TOR, 2019).

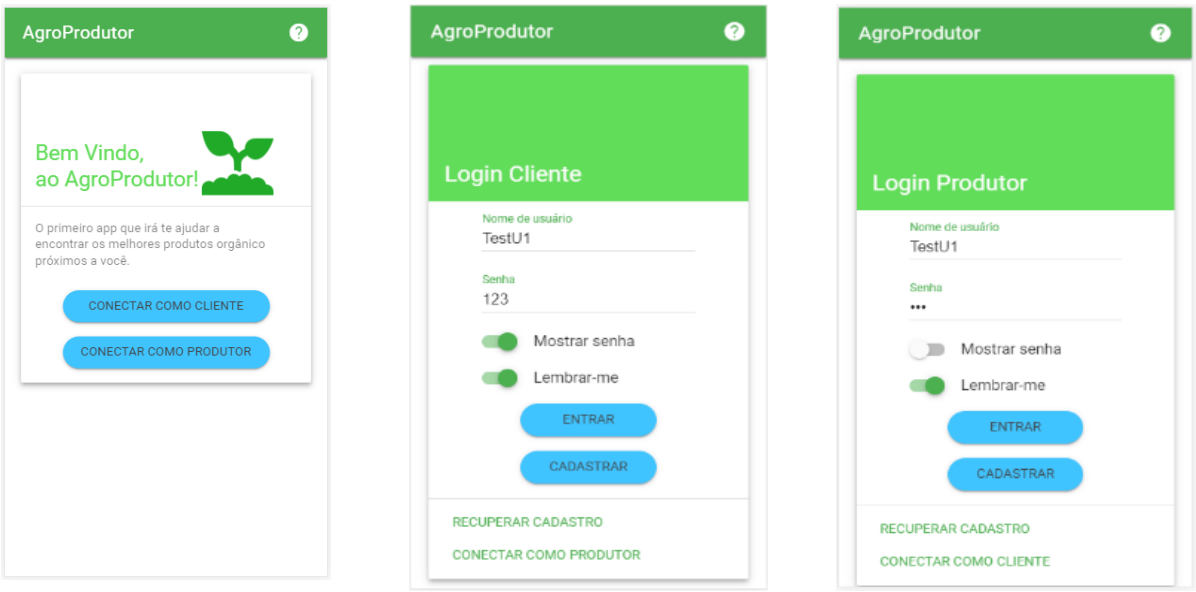

Figura 4. Telas de apresentação e login (AUTOR, 2019).

ISSN 2447-5378

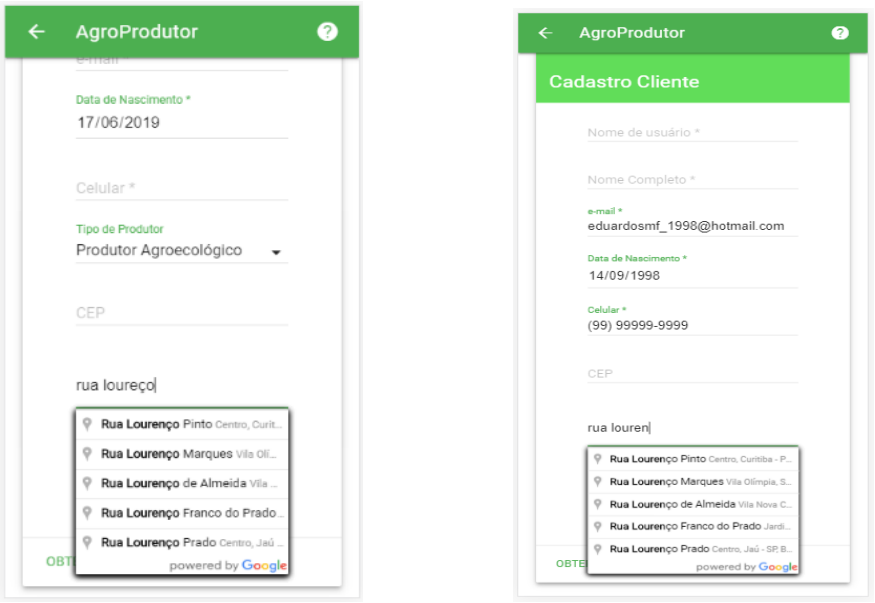

Figura 5. Telas de cadastro (AUTOR, 2019).

As telas de login apresentam funções semelhantes, como autenticação do usuário, direcionar o usuário para tela de recuperar senha e de cadastro, habilitar a visualização da senha e salvar a informações do usuário no *Local Storage* (armazenamento local) do navegador, para que o usuário não precise digitar quando for logar na aplicação.

Assim como a tela de login as telas de cadastro são idênticas em funcionalidade:

- Buscar endereço por CEP: enviado como um objeto JSON em consulta no web service ViaCEP, retornando como resposta outro objeto JSON com o endereço completo;
- Autocompletar o endereço utilizando a API de Autocomplete Address do Google Maps;
- Verificar se o campo está preenchido corretamente.

No arquivo manifest (manifesto) foram inseridas as configurações padrões do aplicativo para serem usadas pelo dispositivo do usuário, as informações contidas neste documento são:

- Nome do Aplicativo;
- Cores do tema base da aplicação;
- Propriedades da imagem do ícone em diferentes tamanhos;
- Orientação '*portrait*' para somente retrato para evitar que o usuário mude a orientação da tela do celular, esta opção foi adotada para evitar a criação de outro documento de estilo.

O *Service worker* foi projetado de forma que deixasse a aplicação mais próxima de um aplicativo nativo.

2.14. Menus

Por ter dois tipos de usuários foram feitos os arquivos HTML 'menuProdutor' e 'menuCliente', para abrir várias telas a partir deles. O arquivo 'menuProdutor' contém as telas que compõem o menu do produtor (Figura 6). Já o arquivo 'menuCliente', só contém a tela do mapa.

O menu do produtor apresenta as seguintes funcionalidades:

- Visualizar a lista de seus produtos cadastrados, onde o produtor pode editar a disponibilidade de seus produtos;
- Direcionada a partir da tela de listagem dos produtos, uma tela para adicionar produtos, onde verifica se eles já estão cadastrados;

ISSN 2447-5378

- Visualizar a lista de clientes que visualizaram o perfil do produtor, onde direciona para a tela do perfil do cliente.

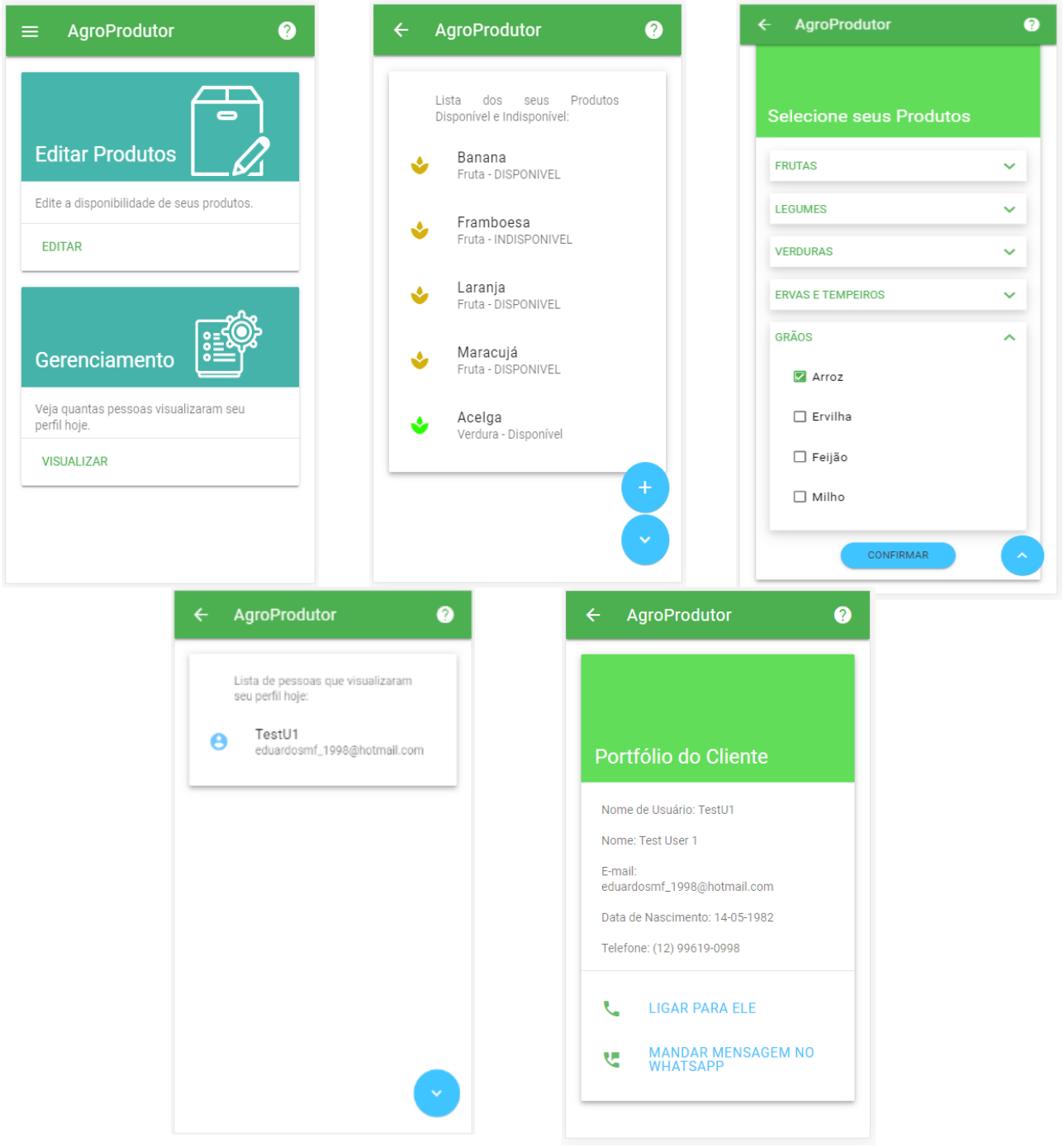

Figura 6. Telas disponíveis a partir do click no botão de Editar e Visualizar na tela do produtor (AUTOR, 2019).

O mapa proporciona as seguintes interações para o cliente (Figura 7):

- Botões para adicionar mais ou menos zoom no mapa, botão de voltar ao centro do mapa, escala, botão para retornar à orientação do mapa de volta ao norte, que também indica a orientação atual do mapa;

ISSN 2447-5378

- Visualizar a informação do produtor ao clicar no ícone dele no mapa, onde pode ser direcionado para o WhatsApp do produtor ou para a tela de perfil completo do mesmo, o usuário pode acessar estas informações também no modo offline, já que é armazenada no banco de dados no *IndexedDB* (Banco de Dados Indexado);
- Realizar filtros com o tipo de produtor ou tipo de produto.

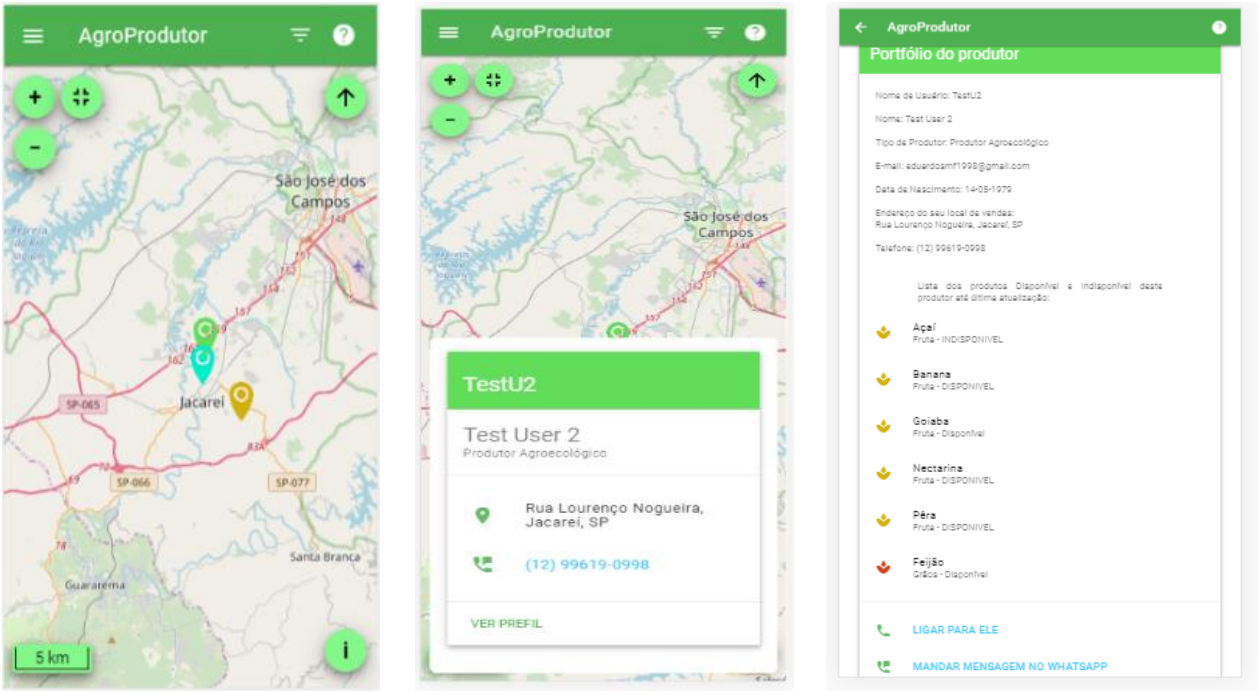

Figura 7. Todas as telas geradas a partir do click no mapa (AUTOR, 2019).

O filtro por distância na Figura 8 é uma consulta SQL de interseção por buffer, feita por uma requisição *get* via Node.js no banco de dados, onde utilizam-se as funções da extensão espacial PostGIS:

- *ST_Buffer* – gera uma geometria circular, que utiliza como parâmetro de criação as coordenadas da localização de usuário, obtidas do GPS, para fixar o centro deste círculo nesta posição geográfica. A distância fornecida pelo usuário como raio do círculo e a projeção (WGS84 – 4326) como sistema de referência geográfica;
- *ST_GeomFromText* – Converte a coordenada do tipo *float* para tipo *string*;
- *ST_Intersects* – verifica se as geometrias se cruzam.

O código SQL a seguir mostra um exemplo de uso das funções:

ST_Intersects(geom, ST_Buffer(ST_GeomFromText('Point(coordinate)', 4326), distance))

O campo geom refere-se as geometrias de pontos dos produtores e a função *ST_Intersects* verifica se a geometria geom possui interseção com o buffer. Essa consulta está na view vProdutor (Figura 1) e ela é usada para compor um objeto JSON para ser enviado para o dispositivo do usuário.

ISSN 2447-5378

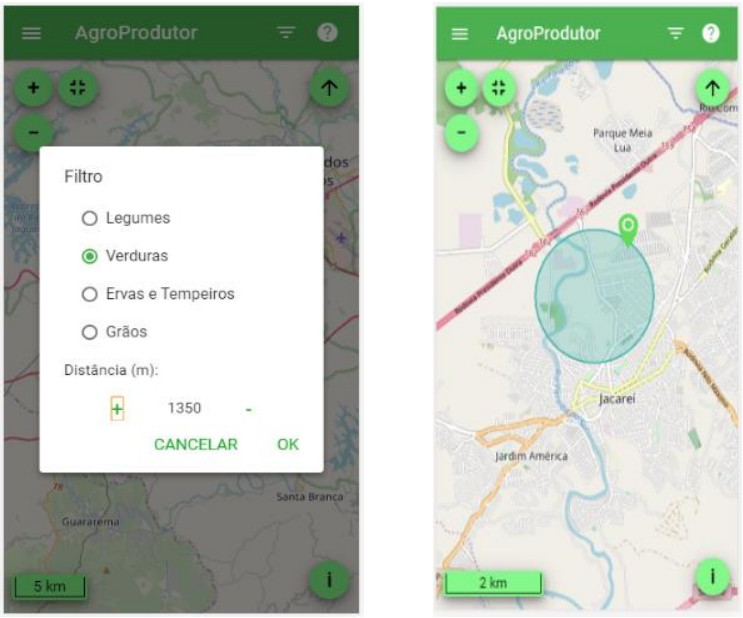

Figura 8. Filtros por tipo de produto e distância (AUTOR, 2019).

2.15. Disponibilização da Aplicação

O Heroku é uma plataforma de hospedagem na nuvem que permite criar, fornecer, monitorar e dimensionar aplicações (HEROKU, 2019).

A construção desta aplicação seguiu de maneira estruturada, agrupando os conteúdos em arquivos e os arquivos em pastas, e para restringir o acesso dos usuários as telas do seu perfil (cliente ou produtor) utilizou-se criação de rotas.

A hospedagem da aplicação no Heroku, possibilitou experimentar a aplicação em um ambiente de uso, podendo ser acessada a partir do link "https://agroprodutor.herokuapp.com/".

4. RESULTADOS E DISCUSSÃO

Como os aplicativos de transporte, o aplicativo AgroProdutor, criado neste trabalho, tem um grande potencial de ajudar os produtores a adquirir clientes e conseguir vender os seus produtos de maneira fácil e prática, sem a necessidade de intermediários como supermercados, feiras e outros.

Esta aplicação fornece uma interação agradável ao usuário, onde para o produtor proporciona gerenciar a disponibilidade de produtos a cada momento e verificar diariamente os clientes que visualizaram o seu perfil. Para o cliente existe a possibilidade de utilizar um mapa para visualizar quais produtores estão mais próximos dele e filtrá-los pelo tipo de produtor e tipos de produtos.

Analisando a aplicação, ela foi construída usando apenas bibliotecas e componentes gratuitos e não foram incluídos recursos além daqueles necessários para aproximar os compradores dos produtores. Para chegar a uma otimização dos recursos computacionais foram atribuídos pesos nas funcionalidades, assim excluindo funções desnecessárias e mantendo as mais importantes para o funcionamento. A exclusão seguiu os seguintes critérios:

- Qualquer função que tivesse algum custo para ser concluída, como exemplo a necessidade de compra de alguma API ou algo semelhante, não foi usado nesta aplicação;
- Funções ou APIs que requeresse estudo aprofundado para sua utilização, não foram desenvolvidas ou utilizadas;

ISSN 2447-5378

- Funções que requeriam um desenvolvimento robusto em estilização no arquivo de estilo CSS, não foram geradas;
- Funções que alterassem várias tabelas no banco de dados;
- Funções no qual demandavam o aprimoramento de outras funções já finalizadas;
- Funções no qual tivesse que gerar muitas telas.

Além da exclusão para facilitar o desenvolvimento, as funções foram codificadas de forma que pudessem ser reutilizadas e assim evitassem que regras de negócio tivessem comportamentos distintos.

Devido a esta simplificação muitas funcionalidades não foram feitas ou perderam um potencial de abrangência, um exemplo, são os filtros no mapa, que foram limitados a apenas duas opções. O aplicativo pode melhorar se forem feitas as seguintes modificações:

- Mapa atualizado – API Openlayers por ser gratuita não mantém os seus mapas atualizados, pois seu principal método de atualização é fornecido manualmente pelo usuário;
- Criar rotas no mapa – Para criar rotas do cliente até o produtor, seria necessário pagar por uma API de navegação, assim como, a Google Maps de Geolocalização;
- Lista interativa com o mapa – Uma lista gerada por um filtro, no qual o cliente poderia escolher qual produtor desejado e ser direcionado para a posição dele no mapa.

Mesmo com as limitações, esta aplicação continua fornecendo recursos excelentes aos usuários.

5. CONCLUSÃO

O trabalho mostrou o desenvolvimento de uma aplicação capaz de conectar consumidores a produtores de produtos naturais. Essa ferramenta constitui uma ótima oportunidade de criar unidades produtivas, pois pessoas que dispõem de uma área no quintal ou sítio podem cultivar para a comercialização sem ter a necessidade de buscar clientes. Por ser uma aplicação web progressiva é fácil de ser instalada e divulgada, pois basta enviar uma URL de acesso por e-mail ou aplicativo de mensagens, assim como, Messenger ou WhatsApp.

Esta aplicação pode ser utilizada em maior escala, porém deve-se melhorar os componentes fornecidos por ela, já que foi feito de maneira simples e sem muito intuito comercial em curto período, além de migrar para servidores dedicados e mais robustos, para que não prejudique o uso.

O aplicativo AgroProdutor ficará melhor se forem utilizadas API de mapas e geolocalização pagas, pois poderão ser utilizados filtros que consideram o tempo de deslocamento até o produtor, além da opção de traçar rotas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, L. H. *Progressive Web App: Aplicando as técnicas de PWA em Angular 6*. 2018. Disponível em https://medium.com/@lucashenriquedeabreu/progressive-web-app-aplicando-as-t%C3%A9cnicas-de-pwa-em-angular-6-87ee22444d19. Acesso em: 19/03/2019.

BOEHM, C. *Pesquisa mostra que produtos orgânicos são consumidos por 15% da população*. 2017. Disponível em http://agenciabrasil.ebc.com.br/geral/noticia/2017-06/pesquisa-mostra-que-produtos-organicos-sao-consumidos-por-15-da-populacao. Acesso em: 28/06/2018.

BRASIL. Lei nº 10.831, de 23 de dezembro de 2003. *Dispõe sobre a agricultura orgânica e dá outras providências*. Brasília, 2003.

EXPRESS. *Fast, unopinionated, minimalist web framework for Node.js*. 2019. Disponível em https://expressjs.com/pt-br/. Acesso em 21/04/2019.

ISSN 2447-5378

GOOGLE. *Seu primeiro aplicativo da Web progressivo*. Disponível em https://codelabs.developers.google.com/codelabs/your-first-pwapp/#0. Acesso em: 18/03/2019.

HEROKU. *What is heroku?*. Disponível em https://www.heroku.com/what. Acesso em 21/04/2019.

LIU, M. *Orgânicos: modismo ou fato?*. 2017. Disponível em https://revistagloborural.globo.com/Noticias/Agricultura/noticia/2017/01/organicos-modismo-ou-fato.html. Acesso em: 28/06/2018.

MDL – Material Design Lite. *Sobre: Material Design Lite*. 2019. Disponível em https://getmdl.io/. Acesso em: 22/03/2019.

MURATE, R. *O crescimento da Agricultura Orgânica e a busca por uma alimentação saudável*. 2017. Disponível em http://www.agroplanning.com.br/2017/09/06/o-crescimento-da-agricultura-organica-e-busca-por-uma-alimentacao-saudavel/. Acesso em: 02 jul. 2018.

NODE. *About Node.js*. 2019. Disponível em https://nodejs.org/en/about/. Acesso em 22/05/2019.

OPENLAYERS. *OVERVIEW***.** 2019. Disponível em https://openlayers.org/. Acesso em: 27/05/2019.

POSTGIS. *About PostGIS*. 2019. Disponível em https://postgis.net/. Acesso em: 27/05/2019.

POSTGRESQL. *What is PostgreSQL?*. 2019. Disponível em https://www.postgresql.org/about/. Acesso em: 27/05/2019.

RECINE, E.; RADAELLI, P. *Alimentação Saudável*. Disponível em http://bvsms.saude.gov.br/bvs/publicacoes/cuidado_alimentos.pdf. Acesso em: 02/07/2018.

SEBRAE – Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. *O mercado para os produtos orgânicos está aquecido*. 2017. Disponível em http://www.sebrae.com.br/sites/PortalSebrae/artigos/o-mercado-para-os-produtos-organicos-esta-aquecido,5f48897d3f94e410VgnVCM1000003b74010aRCRD. Acesso em: 28/06/2018.