



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Desenvolvimento de um aplicativo multiplataforma para compras mensais de supermercado

Ícaro Bicalho Quintão

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Euler Horta Marinho

**Dezembro, 2019
João Monlevade–MG**

Ícaro Bicalho Quintão

**Desenvolvimento de um aplicativo
multiplataforma para compras mensais de
supermercado**

Orientador: Euler Horta Marinho

Monografia apresentada ao curso de Engenharia da Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Dezembro de 2019

Q79d

Quintão, Icaro Bicalho.

Desenvolvimento de um aplicativo multiplataforma para compras mensais de supermercado [manuscrito] / Icaro Bicalho Quintão. - 2019.

72f.:

Orientador: Prof. MSc. Euler Horta Marinho.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Engenharia de Software. 2. Aplicativos móveis. 3. Comércio eletrônico. 4. Supermercados. I. Marinho, Euler Horta. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.777

Catalogação: ficha.sisbin@ufop.edu.br



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



FOLHA DE APROVAÇÃO

Ícaro Bicalho Quintão

Desenvolvimento de um aplicativo multiplataforma para compras mensais de supermercado

Membros da banca

Euler Horta Marinho - Mestre - UFOP

Diego Zuquim Guimarães Garcia - Doutor - UFOP

Daniela Rodrigues Dias - Mestre - Doutoranda em Educação UFOP

Versão final

Aprovado em 09 de dezembro de 2019

De acordo

Professor (a) Orientador (a)

Euler Horta Marinho



Documento assinado eletronicamente por **Euler Horta Marinho, PROFESSOR DE MAGISTERIO SUPERIOR**, em 09/01/2020, às 11:28, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0031322** e o código CRC **94EEA1F5**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.000135/2020-14

SEI nº 0031322

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: - www.ufop.br

Este trabalho é dedicado à minha família, por acreditar em mim.

Agradecimentos

Agradeço a Deus por ter me dado saúde e força para superar as dificuldades.

Aos meus pais, Darilton e Soraia, pelo amor, confiança e apoio incondicional.

Ao meu irmão, José, pela amizade e carinho.

Aos meus padrinhos: Adilson Ferreira (em memória) e Arlene Félix, por sempre torcerem por mim.

A esta universidade, seu corpo docente e administração que oportunizaram minha formação profissional.

Ao meu orientador Euler, pelo suporte, correções e ensinamentos.

E a todos que fizeram parte da minha formação, o meu muito obrigado.

“Science is more than a body of knowledge; it is a way of thinking.”

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

Este trabalho tem como proposta o desenvolvimento de um aplicativo móvel que auxilia nas compras feitas em supermercados, principalmente compras maiores, às que possuem um volume maior e geralmente são feitas uma vez ao mês. Na aplicação, o usuário escolhe os produtos, faz o pagamento e recebe os produtos em casa. Esse sistema foi desenvolvido utilizando ferramentas que permitem a sua execução nos sistemas Android e iOS. Para isso, utilizou-se das tecnologias ofertadas pelo FireBase e Ionic Creator. Durante o desenvolvimento do projeto, foram usados recursos específicos de Engenharia de Software, Sistemas Móveis e Banco de Dados.

Palavras-chaves: supermercado. consumo. comércio eletrônico.

Abstract

This work aims to develop a mobile application that assists purchases made in supermarkets, especially larger purchases, those that have a larger volume and are usually made once a month. In the application, the user chooses the products, makes the payment and receives the products at home. This system was developed using tools that allow it to run on Android and iOS systems. For this, we used the technologies offered by FireBase and Ionic Creator. During the development of the project, specific features of Software Engineering, Mobile Systems and Database were used.

Key-words: supermarket. consumption. e-commerce.

Lista de ilustrações

Figura 1 – Tecnologias relacionadas ao Xamarin.	21
Figura 2 – Tecnologias relacionadas ao Ionic.	21
Figura 3 – Total de acessos à pesquisa e localidade dos cliques.	22
Figura 4 – Quantidade de usuários de Android vs iOS.	22
Figura 5 – Localidade dos clientes em potencial.	23
Figura 6 – Idas ao Supermercado ao mês.	23
Figura 7 – Formas de pagamento.	24
Figura 8 – Valores de Frete.	24
Figura 9 – Atratividade de um aplicativo.	25
Figura 10 – Interessados em participar dos futuros testes.	25
Figura 11 – Diagrama Casos de Uso	27
Figura 12 – Protótipo de telas.	30
Figura 13 – Diagrama Entidade-Relacionamento.	31
Figura 14 – FireBase.	31
Figura 15 – Painel do trabalho no FireBase	32
Figura 16 – Usuários no FireBase <i>Realtime Database</i>	32
Figura 17 – Usuários no FireBase <i>Authentication</i>	33
Figura 18 – Testes executados no FireBase <i>Test Lab</i>	33
Figura 19 – Ionic Creator	34
Figura 20 – Visualização da página Cadastro pelo Ionic Creator no iPhone	35
Figura 21 – Visualização da página Cadastro pelo Ionic Creator no Android	35
Figura 22 – Trecho do código-fonte da página Cadastro	37
Figura 23 – Trecho do código-fonte da página de Login	38
Figura 24 – Cadastro de produto via Ionic Creator	39
Figura 25 – Aplicativo no sistema Android	40
Figura 26 – Página de <i>Login</i> do aplicativo no Android	41
Figura 27 – Página de Termo de Uso do aplicativo no Android	42
Figura 28 – Página de Cadastro de Usuário do aplicativo no Android	43
Figura 29 – Página de escolha de avatar do aplicativo no Android	44
Figura 30 – Página de <i>Home</i> do aplicativo no Android	46
Figura 31 – Página de Categorias do aplicativo no Android	47
Figura 32 – Página da Categoria Limpeza do aplicativo no Android	48
Figura 33 – Página de acesso ao produto via Categoria do aplicativo no Android	49
Figura 34 – Página de Sugestão do aplicativo no Android	50
Figura 35 – Página de Sugestão fluxo 1 do aplicativo no Android	51
Figura 36 – Página de Sugestão fluxo 2 do aplicativo no Android	52

Figura 37 – Página de produto sugerido do aplicativo no Android	53
Figura 38 – Página de Pesquisa do aplicativo no Android	54
Figura 39 – Página de Pesquisa do aplicativo no Android (Continuação)	55
Figura 40 – Página de Produto acessada pela Pesquisa do aplicativo no Android . .	56
Figura 41 – Confirmação de Produto adicionado ao Carrinho do aplicativo no Android	57
Figura 42 – Página do Carrinho do aplicativo no Android	58
Figura 43 – Página de Compra Efetuada do aplicativo no Android	59
Figura 44 – Página de Perfil de Usuário do aplicativo no Android	60
Figura 45 – Página de Perfil de Usuário para editar dados do aplicativo no Android	61
Figura 46 – Página inicial do aplicativo no iPhone	62
Figura 47 – Página do carrinho do aplicativo no iPhone	63
Figura 48 – Página de Perfil de Usuário do aplicativo no iPhone	64
Figura 49 – Gráfico de rastreamento	66
Figura 50 – Detalhes do dispositivo do teste 1	66
Figura 51 – Detalhes do dispositivo do teste 1 parte 2	67
Figura 52 – Desempenho ao longo do tempo teste 1	67
Figura 53 – Detalhes do dispositivo do teste 2	67
Figura 54 – Detalhes do dispositivo do teste 2 parte 2	68
Figura 55 – Desempenho ao longo do tempo teste 2	68

Lista de abreviaturas e siglas

APK	<i>Android Package</i>
CEP	Código de Endereçamento Postal
CPU	<i>Central Processing Unit</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
iOS	<i>iPhone Operating System</i>
SDK	<i>Software Development Kit</i>
SO	Sistema Operacional
TI	Tecnologia da Informação
XCTest	<i>XCode Test</i>

Sumário

1	INTRODUÇÃO	14
1.1	O problema de pesquisa	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	16
1.3	Organização do trabalho	16
2	REVISÃO BIBLIOGRÁFICA	17
3	DESENVOLVIMENTO	19
3.1	Aplicativos Semelhantes	19
3.1.1	Rappi	19
3.1.2	Meu Mercado em Casa	19
3.1.3	Listonic	20
3.2	Frameworks	20
3.3	Levantamento e Análise de Requisitos	21
3.4	Casos de Uso	27
3.4.1	Caso de Uso 01 relacionado ao ator Usuário	27
3.4.2	Caso de Uso 02 relacionado ao ator Funcionário	28
3.5	Protótipos de telas	29
3.6	Banco de dados	29
3.6.1	FireBase	30
3.7	Ionic Creator	34
4	RESULTADOS	39
4.1	O aplicativo	39
4.1.1	Login	39
4.1.2	Cadastro de usuário	42
4.1.3	Página inicial	45
4.1.4	Sugestões	50
4.1.5	Pesquisa	54
4.1.6	Carrinho	58
4.1.7	Minha conta	60
4.1.8	iPhone	62
4.2	Testes	65
4.2.1	FireBase Test Lab	65

4.2.2	Testes por outros desenvolvedores	66
4.2.3	Testes por clientes em potencial	68
5	CONSIDERAÇÕES FINAIS	70
5.1	Trabalhos Futuros	71
	REFERÊNCIAS	72
	ANEXOS	73

1 Introdução

A tecnologia da informação, através de suas aplicações e soluções, é conhecida por contribuir para simplificar a vida das pessoas. O status atual da TI capacita as práticas de trabalho do dia-a-dia nas organizações, mas serve também como ponto de partida para as inovações que se tornam necessárias à sua sobrevivência nos negócios. As deficiências nas práticas de trabalho das empresas podem revelar necessidades, oportunidades e riscos. Essas necessidades e oportunidades podem motivar fortemente a busca por melhoramentos tecnológicos, tais como a captação de dados estratégicos (FILHO, 1994).

Segundo (HAYDEN; WEBSTER, 2014), existe uma tendência cada vez maior de as empresas disponibilizarem seus produtos para venda em sistemas móveis e, além disso, os consumidores estão se sentindo mais à vontade em fazer compras nesses aplicativos. (DINIZ, 1999) divide as vantagens que a TI traz para as empresas em três grupos, como veremos a seguir. Menores custos: a informática, quando bem utilizada, reduz os custos da empresa, porque agiliza e possibilita maior segurança e confiabilidade nos processos, rotinas e controles administrativos; simplifica as tarefas burocráticas; reduz os erros e praticamente elimina a repetição do trabalho. Maior produtividade: possibilita que as pessoas produzam mais, em menos tempo, com menor dispêndio de recursos; permite aproveitar melhor a capacidade produtiva da empresa com o planejamento e o controle da produção; armazena e localiza imediatamente informações fundamentais para os negócios; agiliza os processos de tomada de decisões em relação a preços, estoques, compras e vendas, entre outros. E por fim, maior qualidade: a qualidade dos produtos e serviços é melhorada, pois as tecnologias de informação ajudam a manter o padrão dos produtos dentro das especificações estabelecidas; proporciona melhores condições de trabalho para os empregados; reduz esforços com a burocracia para concentrá-los nas atividades fins da empresa.

Assim, foi proposto um aplicativo que ofereça os produtos que as pessoas conhecem dos supermercados, para que elas possam fazer suas compras, sem que esse aplicativo tenha limitações de plataforma operacional.

1.1 O problema de pesquisa

O trabalho foi idealizado quando foi detectada uma demanda na Região do Vale do Aço através de conversas com familiares e amigos, na qual percebemos que ir ao supermercado demanda tempo e esforço, os quais não são mais necessários com as tecnologias alcançadas hoje em dia. Alguns supermercados da região oferecem a entrega das compras, porém o cliente deve ir até o estabelecimento para efetuar a compra e após a conclusão, solicitar que

a ela seja entregue em sua casa. Caso a compra não ultrapasse um valor pré-estabelecido pelo supermercado, o cliente paga um frete referente proporcional ao valor total. Além disso, não existe nenhum aplicativo com esse modelo de negócio na região.

A Região do Vale do Aço fica localizada no Vale do Rio Doce e é composta pelas cidades de Coronel Fabriciano, Ipatinga, Santana do Paraíso e Timóteo e pelo colar metropolitano, que é constituído por outros 24 municípios. Segundo o (IBGE, 2019), a população das 4 principais cidades do Vale do aço somadas são de 497.770 habitantes aproximadamente. Assim, podemos assumir que existe um mercado a ser explorado e que ele é completamente rentável, pois as pessoas fazem compras todos os dias. Trazer um aplicativo para o dia a dia dos clientes que propicie conforto, simplicidade e economia de tempo será muito bem acolhido, como vamos ver na pesquisa de clientes em potencial disponível no Capítulo 2. Após o lançamento do aplicativo, um trabalho simples de *marketing* focando promoções nas quais os clientes indiquem mais clientes e ganham descontos poderá ser feito para que mais pessoas conheçam o *software*.

O outro lado do aplicativo diz respeito ao supermercado que terá suas filas diminuídas e onde podem ser feitas parcerias com os mesmos visando o lucro das duas partes. Pode-se também terceirizar as entregas tornando uma via de mão dupla entre os comerciantes e o *software*. Porém, a ideia inicial é coletar as compras que os clientes efetuarem durante uma semana e entregá-la em um dia pré-estabelecido e no horário determinado pelo cliente, sendo assim o aplicativo contará com um funcionário que fará as compras e as entregas realizadas. Após ser completamente implementado na região, o aplicativo não acabará com os supermercados, pois muitas pessoas gostam de escolher certos tipos de produtos que comprem, como carne, legumes e frutas, porém as compras maiores, normalmente feitas uma vez ao mês que possuem produtos fixos, como produtos da cesta básica e de limpeza, por exemplo, serão o foco das operações deste trabalho.

Outro problema que alguns aplicativos possuem é a limitação de execução em um único sistema operacional. Como a ideia é atingir o maior número de clientes, é de fundamental importância para o desenvolvimento deste trabalho que o aplicativo desenvolvido seja multiplataforma.

1.2 Objetivos

1.2.1 Objetivo Geral

O presente trabalho consiste em elaborar um aplicativo multiplataforma para simplificar o trabalho que existe em fazer compras no supermercado. Como consequência, poderá haver diminuição das filas dos caixas, economia de tempo e maior comodidade, já que as pessoas são poupadas de carregar os produtos.

1.2.2 Objetivos Específicos

Este trabalho possui os seguintes objetivos específicos:

- Desenvolver todo o sistema em linguagem multiplataforma, visando alcançar o maior número de clientes em potencial.
- Incorporar ao trabalho ideias provenientes de pesquisa com os clientes em potencial e de aplicativos semelhantes.
- Desenvolver o trabalho utilizando conceitos e ferramentas de Engenharia de Software e, principalmente, Teste de Software para gerar um software de qualidade e útil.
- Validar em meio a usuários em potencial o aplicativo desenvolvido, efetuar testes visando fazer as devidas correções em relação a erros e falhas do sistema.

1.3 Organização do trabalho

O restante deste trabalho é organizado como se segue. O [Capítulo 2](#) apresenta uma revisão bibliográfica na qual o trabalho se baseia. O [Capítulo 3](#) aborda um debate sobre os *frameworks* e ferramentas que foram considerados para o desenvolvimento do trabalho e o que fora escolhido, uma pesquisa de aplicativos semelhantes ao proposto, uma pesquisa entre clientes em potencial e os resultados obtidos, bem como o desenvolvimento do *software* em si e a comunicação do mesmo com o servidor. Já o [Capítulo 4](#) discute os resultados obtidos, os testes que foram efetuados e os problemas encontrados, além de apresentar o aplicativo finalizado. Por fim, o [Capítulo 5](#) encerra o trabalho com as considerações finais e propostas para trabalhos futuros.

2 Revisão bibliográfica

O varejo de alimentos que apresenta vários formatos, e em especial o setor supermercadista, vem passando por transformações, que são consequência das mudanças mercadológicas. Uma das mudanças mais importantes é a que ocorreu no comportamento do consumidor, que busca conveniência e dispõe de menos tempo (COSTA; VILARINDO; COSTA, 2007). O futuro das maiores empresas está diretamente ligado a capacidade das mesmas em se conectar com os usuários (SCHWALBE, 2015). De acordo com (LAUDON; LAUDON, 2011), aplicativos e sistemas de negócios são “a principal ferramenta que as empresas dispõem para criar novos produtos e serviços, assim como modelos de negócios inteiramente novos”. A possibilidade de fazer obrigações que antes demandavam tempo e esforço, sem precisar sair de casa, é um dos benefícios que a tecnologia traz.

De acordo com (FILHO, 1994), entende-se por tecnologias de informação o conjunto de hardware e software que desempenha uma ou mais tarefas de processamento das informações do sistema, tal como coletar, transmitir, estocar, recuperar, manipular e exibir dados. A escolha de uma estratégia para o desenvolvimento de aplicativos móveis e o desenvolvimento de software em geral deve ser refletida de um ponto de vista organizacional específico, compreendendo as questões gerenciais e técnicas. A organização deve avaliar, entre outras, as opções de mercado, as habilidades de desenvolvimento e as características do projeto (MARINHO; RESENDE, 2015). De acordo com uma arquitetura móvel proposta com categorias, as classes de aplicativos para dispositivos móveis incluem aplicativos Nativo, Web (nomeado sem cliente), HTML5, Especial, Híbrido e Mensagem (MARINHO; RESENDE, 2015).

As aplicações nativas são as que foram desenvolvidas especialmente para aquela plataforma e somente para ela, utilizando para isso as ferramentas disponibilizadas para a mesma, como a linguagem de desenvolvimento, ambiente e emulador. Aplicativos desenvolvidos na linguagem nativa do sistema tem um melhor desempenho, apesar das outras tecnologias de desenvolvimento terem evoluído bastante quase que se equiparando (PREZOTTO; BONIATI, 2014). Além das aplicações nativas existe também os Web Apps que são basicamente aplicações Web que rodam no *browser* do dispositivo, dessa forma não é necessário se preocupar com as diferentes linguagens, pois ela utiliza padrões Web como HTML, CSS e JavaScript para o lado do cliente e PHP, Java ou outras linguagens para o lado servidor (PREZOTTO; BONIATI, 2014). Por fim, as aplicações Híbridas possuem como finalidade funcionar em qualquer que seja o dispositivo, sendo que para as diferentes plataformas, será utilizado o mesmo código-fonte. Partindo do princípio de utilizar uma mesma aplicação para diferentes plataformas, normalmente as aplicações híbridas são desenvolvidas em linguagens Web que são interpretadas pelo *browser* nativo do sistema

(PREZOTTO; BONIATI, 2014). A grande vantagem de sistemas híbridos diz respeito ao maior número de clientes que conseguem acessar o *software*. De toda forma, um sistema só é validado se for testado corretamente, sendo livre de *bugs* e erros que possam causar prejuízos maiores no futuro.

De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos (NETO, 2007).

3 Desenvolvimento

Este capítulo descreve o desenvolvimento do trabalho, no qual, serão apresentados os levantamentos de requisitos, os casos de uso, a ideia visual inicial que se espera do trabalho através dos protótipos de telas, o banco de dados e as ferramentas tecnológicas utilizadas.

A primeira etapa do trabalho consistiu na busca por aplicativos semelhantes ao que foi proposto. A partir disso, foi elaborado o levantamento e a análise de requisitos, onde foi divulgada uma pesquisa voltada para clientes em potencial e, assim, elaborados os requisitos funcionais e não funcionais do sistema. Tratamos os casos de uso, além de um estudo para a escolha de um *framework* a ser utilizado. Em seguida, o trabalho começou a ser colocado em prática, com a elaboração dos protótipos de telas e do banco de dados do sistema.

3.1 Aplicativos Semelhantes

3.1.1 Rappi

O primeiro aplicativo a ser analisado, Rappi ¹ é um aplicativo bem desenvolvido e com boas propostas. Podemos destacar alguns dos seus recursos como a pontuação de usuário, quanto mais pontos mais benefícios o cliente tem. A forma de adicionar produtos como favoritos, facilitando compras posteriores. Por fim, o modo como a tela inicial foi construída, destacando as promoções.

O principal ponto negativo é que esse aplicativo se propõe a entregar desde pastéis de lanchonete até itens de farmácia, sendo assim, ele não possui foco em um único serviço. Outro ponto negativo diz respeito a localização, o Rappi só funciona em cidades grandes como Belo Horizonte, São Paulo e Rio de Janeiro. Por fim, como ele permite que lojas cadastradas insiram produtos, alguns produtos tem seus dados incompletos, como não possuir imagens, por exemplo.

3.1.2 Meu Mercado em Casa

O Meu Mercado em Casa² é um aplicativo que faz o que propõe, porém possui diversos pontos negativos. O primeiro ponto que chama a atenção é a poluição visual, onde, desde o menu, todas as informações parecem perdidas. É um aplicativo que não

¹ <https://www.rappi.com.br/>

² <https://www.meumercadoemcasa.com.br/>

passa confiabilidade, o que é essencial quando estamos falando de vendas *online*. Por fim, ele é até mais radical do que o Rappi, pois quando inserido um CEP que eles não atendem, o aplicativo não funciona mais.

3.1.3 Listonic

O último aplicativo analisado, Listonic³, possui uma proposta diferente dos demais, ele é um guia para compras, no qual o usuário faz a lista do que deve ser comprado, selecionando produtos dentro das categorias definidas pelo Listonic. A ideia é que o usuário tenha sempre na lista tudo o que ele ainda não comprou no supermercado. É um aplicativo muito bem desenvolvido, percebemos até uma cautela dos desenvolvedores através de um termo de uso, para resguarda-los de qualquer problema judicial. Um outro destaque é a introdução ao aplicativo, como um manual para o usuário em seu primeiro acesso. Além dos produtos serem divididos por categorias, eles são separados por popularidade. O único ponto negativo é que todas as ações dependem do usuário, desde adicionar os produtos na lista até efetuar a compra, ou seja, o serviço não é automatizado.

3.2 Frameworks

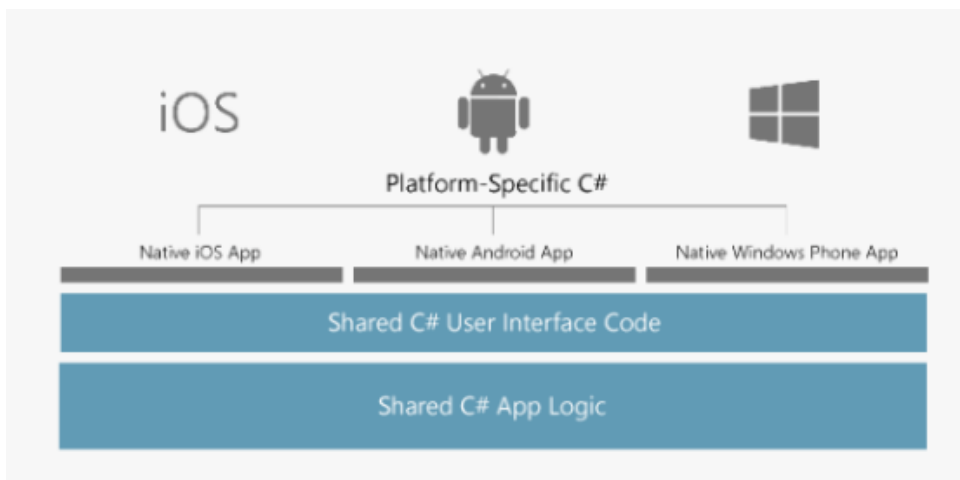
A escolha de um *framework* é de fundamental importância para todo o desenvolvimento do trabalho, pois ele será a base de tudo. Assim, comparamos três dos principais *frameworks* disponíveis para o desenvolvimento de aplicativos móveis multiplataforma: Xamarin, PhoneGap e Ionic.

O Xamarin é baseado em C# e possui diferenças quanto à compilação dependendo da plataforma de destino (SCHMITZ, 2016). Por questão de não afinidade com a linguagem C#, optamos por não utilizá-lo. A Figura 1 nos mostra um esquema elaborado pela Microsoft que explica o funcionamento do Xamarin, no qual C#, uma linguagem de nível mais baixo, é utilizada para executar o aplicativo em qualquer sistema operacional.

As ferramentas Ionic e PhoneGap possuem estruturas semelhantes assim como, o desenvolvimento de aplicativos é parecido. Segundo (SCHMITZ, 2016), “isto se deve ao fato das plataformas disponibilizarem as linguagens JavaScript, HTML5 e CSS para o desenvolvimento de aplicações móveis. Contribui também o fato destas ferramentas gerarem um aplicativo híbrido ou web app e não um aplicativo nativo”. Entretanto, o Ionic foi escolhido para o desenvolvimento do trabalho por ser um dos *frameworks* mais utilizados e já estarmos familiarizado com ele. A Figura 2 nos mostra as tecnologias relacionadas ao Ionic, como o JavaScript, Angular e TypeScript.

³ <https://listonic.com/>

Figura 1 – Tecnologias relacionadas ao Xamarin.



Fonte: (MICROSOFT, 2019)

Figura 2 – Tecnologias relacionadas ao Ionic.



Fonte: (IONIC, 2019)

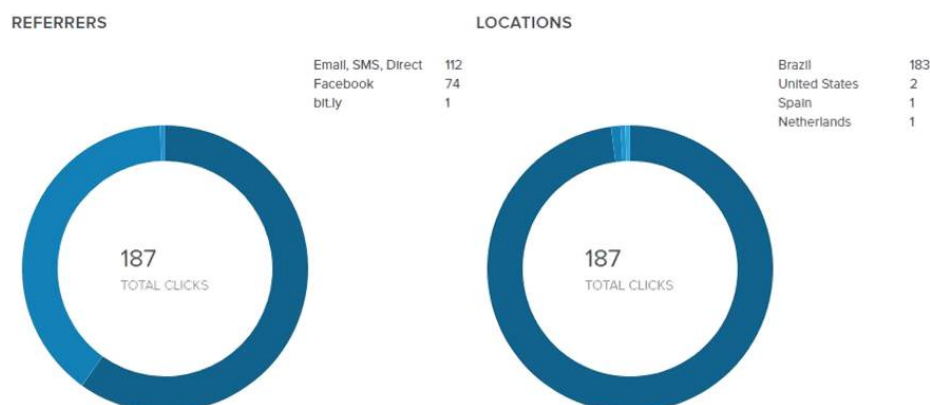
3.3 Levantamento e Análise de Requisitos

O levantamento de requisitos se deu através de uma pesquisa que abordava se a pessoa era residente da Região do Vale do Aço, se ela estaria disposta a usar o aplicativo, e como ela se sentia em ter que ir ao supermercado. A Figura 3 nos mostra os gráficos do alcance da pesquisa. Eles foram obtidos através do site Bitly ⁴, mapeando, assim, os acessos à pesquisa, que teve 187 acessos em 4 países diferentes.

Uma das perguntas, em especial, sustenta a proposta do aplicativo ser desenvolvido como multiplataforma. Perguntamos para os clientes em potencial qual era o sistema operacional de seus celulares e na Figura 4 o gráfico nos mostra que 20,4% responderam ser usuários de iOS, o que é uma porcentagem a ser levada em consideração visto que

⁴ <https://bitly.com>

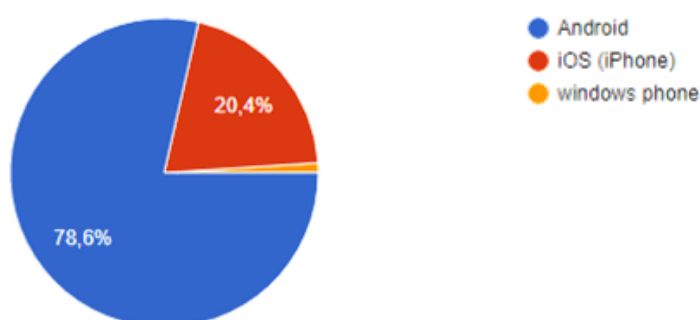
Figura 3 – Total de acessos à pesquisa e localidade dos cliques.



Fonte: Elaborado pelo autor (2019)

queremos abranger o maior número de clientes possível.

Figura 4 – Quantidade de usuários de Android vs iOS.



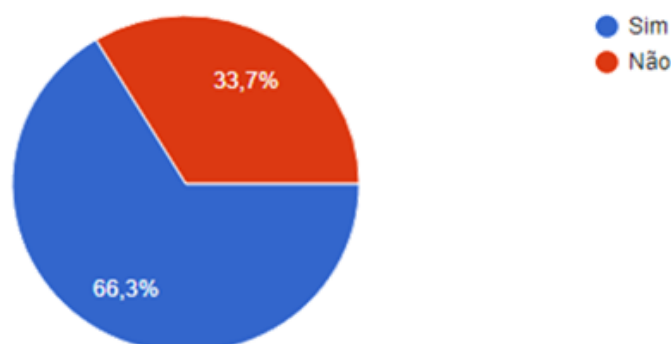
Fonte: Elaborado pelo autor (2019)

A pesquisa foi amplamente divulgada na Região do Vale do Aço, através de redes sociais, mas também, ela conseguiu abranger pessoas interessadas no trabalho que não residem na região de nosso interesse. Dessa forma, a [Figura 5](#) nos mostra que 66,3% das pessoas que responderam o levantamento residem no Vale do Aço, validando a pesquisa.

Um ponto interessante que queríamos abordar é a quantidade de vezes que as pessoas vão ao supermercado durante um mês, isso porque a ideia do nosso trabalho não é livrar as pessoas totalmente do contato com o supermercado, mas sim, das compras padrões mensais que elas possuem. A [Figura 6](#) nos mostra que mais de 93% das pessoas que responderam a pesquisa tem o hábito de ir ao supermercado mais de 3 vezes em um mês. Acreditamos que pelo menos uma dessas compras poderia ser feita em nosso aplicativo.

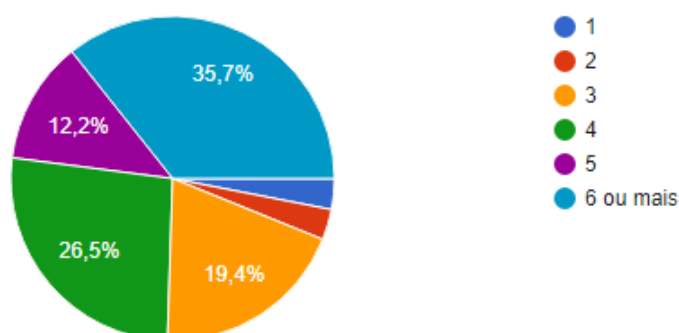
Hoje em dia um aplicativo deve procurar aceitar o maior número de formas de

Figura 5 – Localidade dos clientes em potencial.



Fonte: Elaborado pelo autor (2019)

Figura 6 – Idas ao Supermercado ao mês.



Fonte: Elaborado pelo autor (2019)

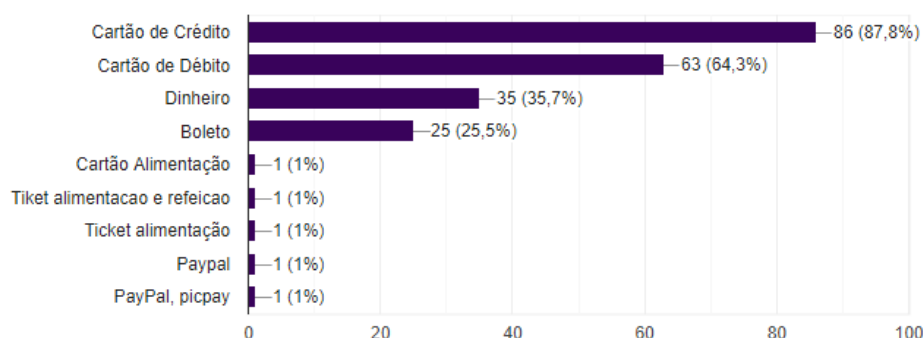
pagamento possível. Sendo assim, perguntamos para os clientes em potencial qual forma de pagamento ele acha indispensável dentro de um aplicativo. A Figura 7 nos mostra que os cartões de crédito e débito são os meios mais populares, entretanto as novas formas de pagamento, como PicPay⁵ e PayPal⁶, também conhecidas como carteiras virtuais apareceram como sugestões, o que poderia ser um diferencial para o nosso aplicativo.

Outra pergunta importante que foi feita e é fundamental para estabelecer o aplicativo no sentido financeiro, é em relação a como deveria ser cobrado o frete de entrega. A Figura 8 nos mostra que a maioria dos usuários preferem pagar um valor proporcional à

⁵ <https://www.picpay.com>

⁶ <https://www.paypal.com>

Figura 7 – Formas de pagamento.



Fonte: Elaborado pelo autor (2019)

compra efetuada no aplicativo, apesar de estarem abertos às sugestões,. Essa pergunta nos abre a possibilidade, através das respostas coletadas, de promover descontos para alguns clientes, como o frete grátis.

Figura 8 – Valores de Frete.

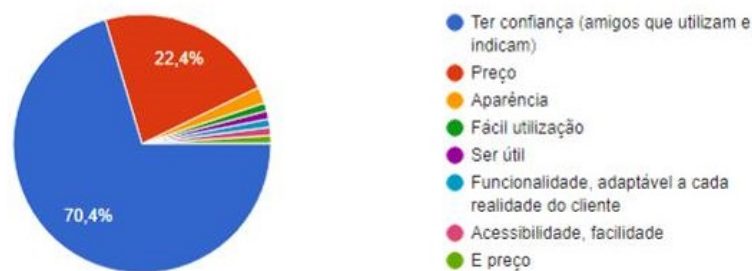


Fonte: Elaborado pelo autor (2019)

Um ponto importante dentro do modelo de comércio eletrônico é saber o que leva uma pessoa a começar a usar um aplicativo novo. A [Figura 9](#) nos mostra que é determinante a pessoa confiar no aplicativo, ou seja, ter amigos que usam e indicam. Muitos aplicativos utilizam um recurso para que um usuário convide outro e tenha, dessa forma, algum tipo de benefício, como descontos. Nossa pesquisa mostra que este recurso é necessário para adquirir novos clientes.

O último gráfico é, na verdade, uma representação das pessoas que responderam a pesquisa e ficaram interessadas em utilizar o aplicativo. Por questão de segurança, os contatos foram preservados e a [Figura 36](#) nos mostra alguns nomes de clientes em potencial que tem interesse no teste do aplicativo. Ao todo 41 pessoas deixaram contato

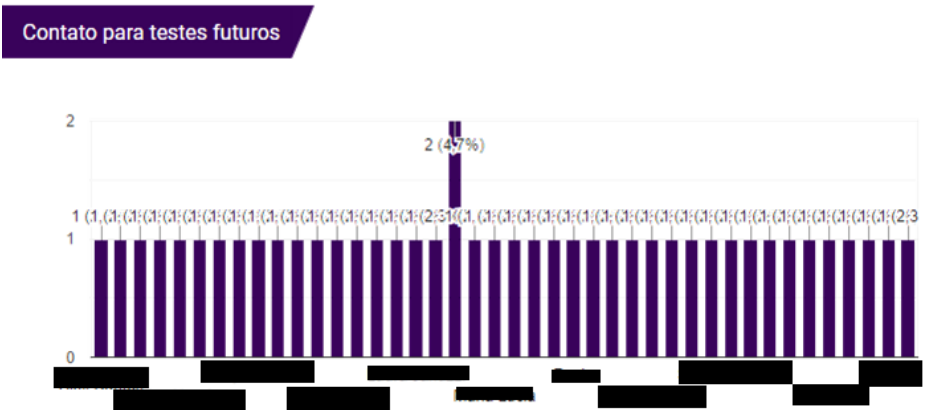
Figura 9 – Atratividade de um aplicativo.



Fonte: Elaborado pelo autor (2019)

para possíveis testes. A ideia foi de inseri-las em testes futuros, quando o sistema estiver pronto.

Figura 10 – Interessados em participar dos futuros testes.



Fonte: Elaborado pelo autor (2019)

As respostas das perguntas e os gráficos coletados e exibidos nessa subseção foram a base para a elaboração dos requisitos do nosso *software*. Assim, eles estão ficaram representados da seguinte forma:

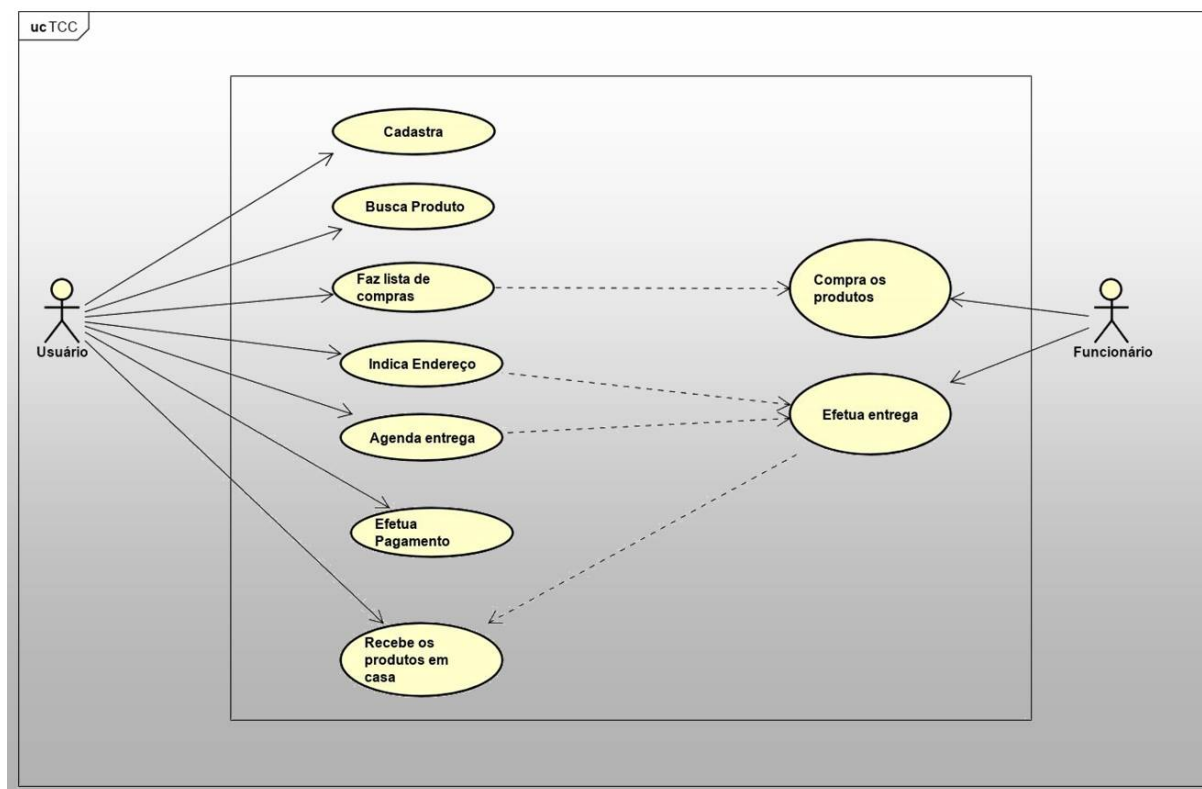
Requisitos funcionais:

- O cliente faz Login ou cadastro, caso seja o primeiro acesso.
- O cliente pode pesquisar por produto, selecionar alguma categoria de produto, escolher a quantidade, finalizar compra, inserir endereço e forma de pagamento.
- O funcionário recebe duas listas de produtos. A primeira referente ao pedido individual de cada cliente e a segunda se refere aos produtos que devem ser comprados no dia.
- O cliente pode acessar uma tela com funcionalidades desenvolvidas através de técnicas de Inteligência Artificial, onde as sugestões de produtos são baseadas no perfil de cada cliente.
- O cliente pode agendar o dia e horário da entrega.
- O cliente pode avaliar cada produto.
- O cliente pagará um valor de frete proporcional à sua compra.
- O cliente pode indicar o aplicativo para alguém e ganhar descontos por isso.
- O funcionário, ao fazer as compras do dia, recebe uma rota com o menor caminho para efetuar a entrega.
- O cliente possui a opção de pagamento por cartão de vale alimentação e carteiras virtuais, como PayPal e PicPay.
- O cliente pode acompanhar em tempo real onde está sua compra.

Requisitos não funcionais:

- O aplicativo deve ser fácil de usar.
- O aplicativo não pode ter limite quanto ao sistema operacional a ser executado.
- O aplicativo deve ser seguro o suficiente para realizar as transações financeiras.

Figura 11 – Diagrama Casos de Uso



Fonte: Elaborado pelo autor (2019)

3.4 Casos de Uso

A [Figura 11](#) apresenta o diagrama de casos de uso da aplicação desenvolvida. Ele é o nosso ponto inicial do trabalho, onde já conseguimos ter uma noção de como o sistema deve funcionar e o que deve ser implementado. A ideia é que tenham dois atores atuando dentro do sistema, o usuário está presente no caso de uso 01 na [subseção 3.4.1](#), e o funcionário que tem seu papel dentro do sistema especificado no caso de uso da [subseção 3.4.2](#). O usuário nada mais é do que o cliente do aplicativo e o funcionário é uma pessoa contratada para suprir as demandas do aplicativo no sentido de efetuar as compras que os clientes solicitaram pelo sistema e entregá-las, ou seja, a parte prática do sistema.

3.4.1 Caso de Uso 01 relacionado ao ator Usuário

1. - Ator Principal: Usuário;
2. - Interessados e Interesses: Usuário: deseja entrada rápida, precisa, sem erros, com confiabilidade e a diversidade de produtos esperado;
3. - Pré-condições: Login efetuado com sucesso;

4. - Garantias de Sucesso (pós-condições): A lista de produtos é armazenada com sucesso. O valor da compra está correto. Pagamento realizado, aprovado e registrado. Gerado recibo;
5. - Cenário de Sucesso Principal (ou Fluxo Básico): O usuário (cliente) seleciona o produto de interesse, consulta a descrição do mesmo, preço, imagem e histórico de preços. Repete até que a lista de compra esteja finalizada. Finaliza a compra. Obtém preço final. Informa endereço de entrega. Efetua pagamento. Pagamento é confirmado. Entrega efetuada dentro do prazo estipulado previamente;
6. - Extensões (ou Fluxos Alternativos): Falha no sistema. Erro em alguma etapa do processo. O usuário não possui cadastro. Informações erradas. Produto não está disponível. Alteração de preço base do produto. Cancelamento de compra. Garantir a recuperação das informações. Fazer cadastro de novos clientes. Ser comunicado através dos meios de contato informados, ao fazer o cadastro, sobre alterações e imprevisto nos produtos do pedido;
7. - Requisitos Especiais: Compras volumosas possuem desconto. Usuários frequentes pontuam e conseguem desconto. Excluir itens da lista de compras. Devolução de produtos;
8. - Lista de Variações Tecnológicas e de Dados: Pesquisar por descrição, palavra semelhante ou histórico de compras. Inteligência Computacional para sugestão de produtos na página inicial do aplicativo;
9. - Frequência de Ocorrência: Contínuo;
10. - Problemas em Aberto: Não consta;

3.4.2 Caso de Uso 02 relacionado ao ator Funcionário

1. - Ator Principal: Funcionário;
2. - Interessados e Interesses: Usuário: deseja receber todos os produtos da lista de compras. Funcionário: escolher local de compra, efetuar a compra e entregar todos os produtos;
3. - Pré-condições: Ter realizado ao menos uma venda;
4. - Garantias de Sucesso (pós-condições): A lista de produtos é gerada com sucesso. A lista de entrega é gerada com sucesso. O funcionário recebe o valor para efetuar a compra. Efetua a compra. Efetua entrega. Gera lucro;
5. - Cenário de Sucesso Principal (ou Fluxo Básico): O funcionário recebe do sistema 2 listas corretas. 1ª referente a todos os produtos que devem ser comprados. 2ª referente

a quais produtos devem ser entregues em cada endereço. Recebe o pagamento. Efetua a compra. Entrega corretamente e no prazo estabelecido;

6. - Extensões (ou Fluxos Alternativos): Falha no sistema. Erro em alguma etapa do processo. Listas com informações erradas. Produto não está disponível. Alteração de preço base do produto. Cancelamento de compra. Garantir a recuperação das informações. Comunicar através dos meios de contato informados, ao fazer o cadastro, sobre alterações e imprevisto nos produtos do pedido. Efetuar devoluções. Reembolsar produtos;
7. - Requisitos Especiais: Em semanas com maior número de compras, serão necessários mais funcionários. Efetuar devolução de produtos. Encontrar a menor rota para efetuar todas as entregas;
8. - Lista de Variações Tecnológicas e de Dados: Terceirizar o processo de compra e entrega de produtos. Gerar a menor rota para efetuar todas as entregas;
9. - Frequência de Ocorrência: Semanal;
10. - Problemas em Aberto: Não consta;

3.5 Protótipos de telas

A elaboração dos protótipos de telas, como percebemos na [Figura 12](#), se deu através da ferramenta Adobe XD, que se mostrou muito útil para se ter uma visão geral do trabalho. Para uma melhor visualização criou-se o nome fictício de *iSuper* para o aplicativo.

A ideia inicial era que a barra de navegação inferior não fosse estática, ou seja, quando o usuário navegasse pelo aplicativo, ela iria ficar oculta. Assim que ele fizesse um movimento pré-definido, ela apareceria. Isso se deu quando tudo estava ainda no campo das ideias e veremos, principalmente no [Capítulo 4](#), que muitos detalhes do trabalho foram adaptados principalmente devido a plataforma de programação escolhida e alguns limites que ela impôs ao *software*. Entretanto, isso não interferiu no resultado final. Todavia a prototipação de telas serviu de guia durante toda a parte de desenvolvimento das telas, ou seja, ela tem fundamental importância dentro desse trabalho.

3.6 Banco de dados

O diagrama da [Figura 13](#) representa a etapa de implementação do trabalho que começou pelo desenvolvimento do Banco de Dados. Nele percebemos toda a estrutura necessária dos dados e chaves primárias em todas as tabelas do sistema. Para a estruturação do Banco de Dados optamos pela utilização do *Realtime Database* do Firebase da Google.

Figura 12 – Protótipo de telas.



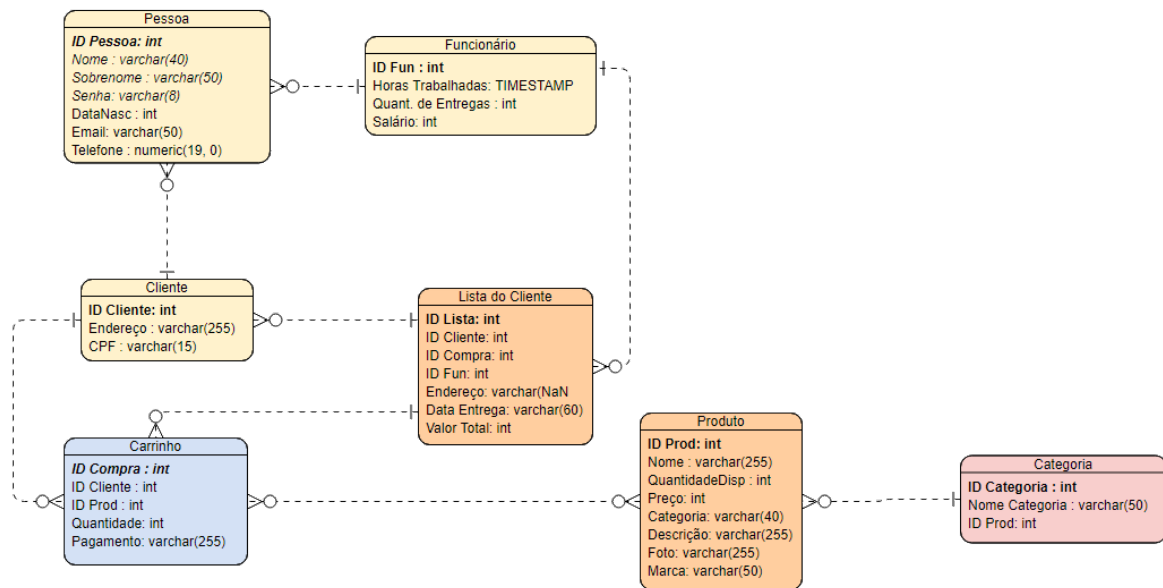
Fonte: Elaborado pelo autor (2019)

3.6.1 Firebase

O Firebase, na verdade, é uma ferramenta para desenvolvimento de aplicativos móveis e Web e não só de banco de dados. Para o nosso trabalho de aplicativo, além do *Realtime Database*, iremos utilizar as ferramentas de autenticação e testes. A Figura 14 nos mostra as diversas possibilidades existentes dentro dessa plataforma, desde recursos para notificar o desenvolvedor de que alguma coisa está errada no sistema até uma ferramenta de propagandas que pode ser anexada ao aplicativo e gera lucro. Para o desenvolvimento do nosso trabalho, a versão utilizada foi a gratuita, o que nos deu uma limitação no que diz respeito ao tamanho das tabelas que poderiam ser armazenadas, porém foi o suficiente para o sucesso do mesmo.

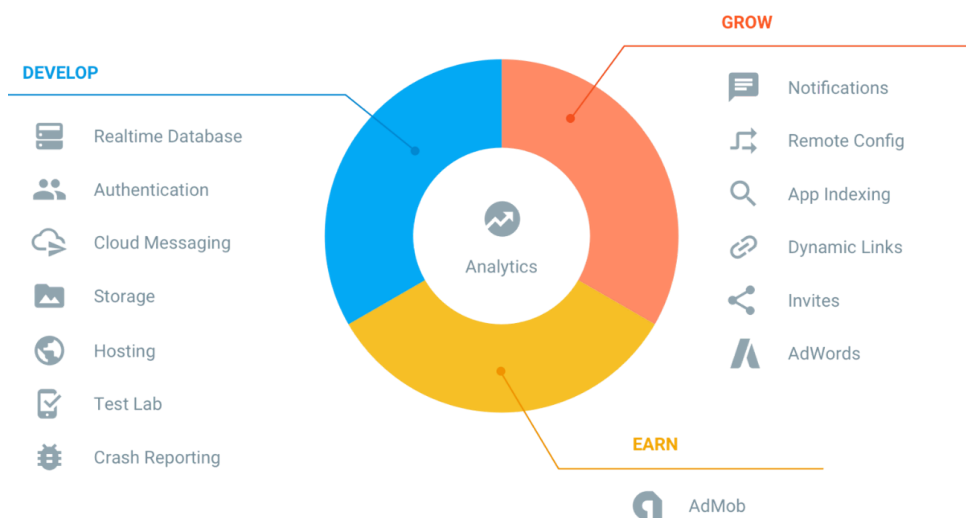
A Figura 15 é a tela inicial do Firebase, onde temos acesso a quantidade de *downloads* que o banco de dados teve nos últimos sete dias, é importante ressaltar que essa captura de tela foi feita em um período em que o aplicativo não estava em testes, portanto a semana está representada como -100% de uso. Além dos *downloads*, temos também um

Figura 13 – Diagrama Entidade-Relacionamento.



Fonte: Elaborado pelo autor (2019)

Figura 14 – FireBase.

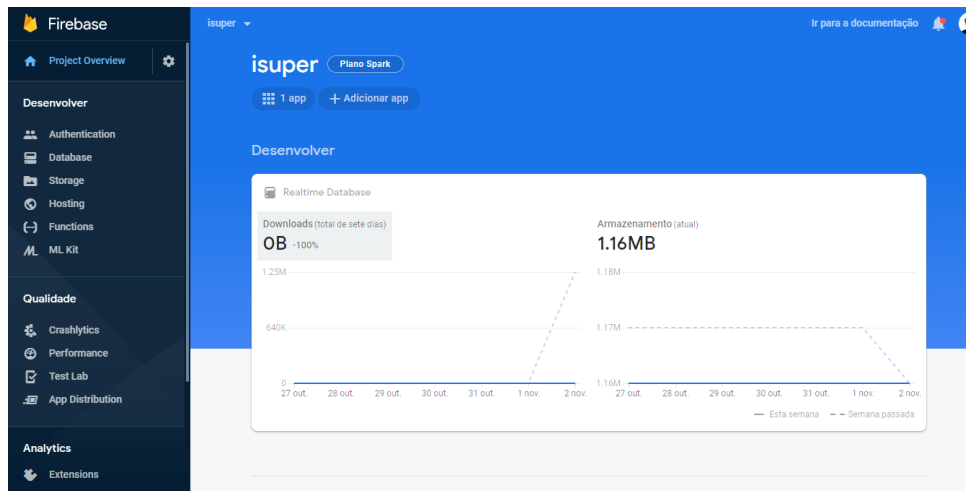


Fonte: (FIREBASE, 2019)

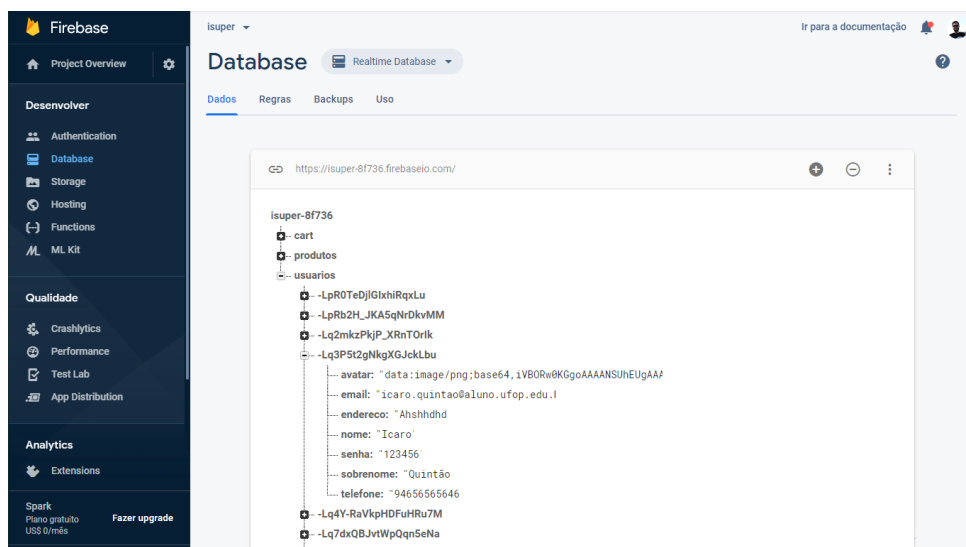
gráfico com a quantidade de armazenamento do banco. Na barra de navegação à esquerda percebemos todos os recursos e ferramentas disponíveis. Como já descrito anteriormente, a [Figura 16](#) nos mostra o recurso do Banco de dados do nosso aplicativo. É interessante ressaltar que o desenvolvedor tem acesso à todas as informações do aplicativo, inclusive senhas e endereços dos usuários e funcionários cadastrados, bem como suas chaves de identificação.

Outro recurso utilizado foi o *Authentication* que está representado na [Figura 17](#).

Figura 15 – Painel do trabalho no FireBase



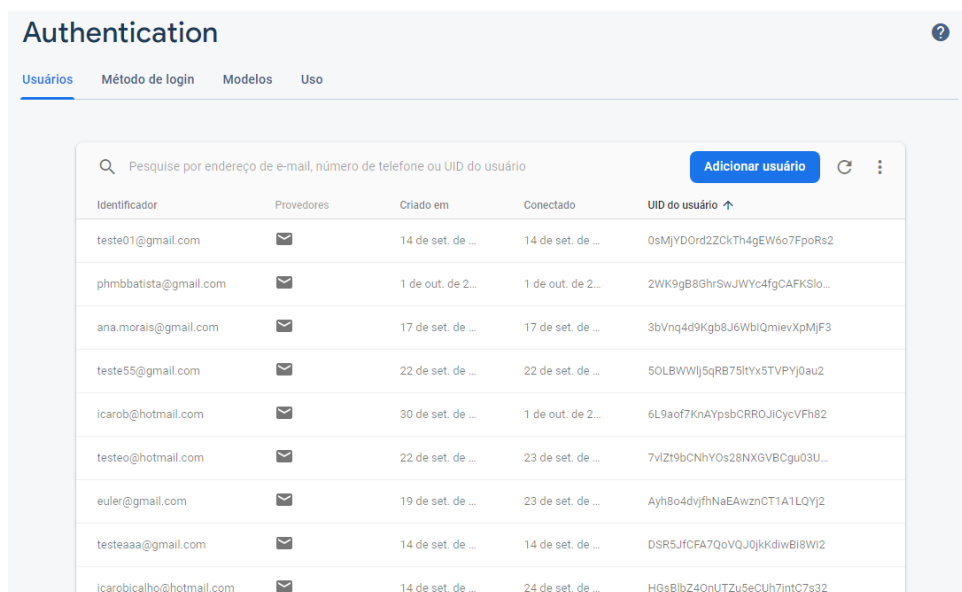
Fonte: (FIREBASE, 2019)

Figura 16 – Usuários no FireBase *Realtime Database*

Fonte: (FIREBASE, 2019) e elaborado pelo autor (2019)

Esse recurso agrega na segurança do aplicativo, pois ele só aceita *e-mails* válidos e não aceita senhas curtas, além disso ele trata o retorno dos dados, se caso a pessoa optar por fazer login com o Facebook ou Conta Google. Por fim, esse recurso possui uma ferramenta que envia mensagem automática para os *e-mails* dos usuários, para boas-vindas, confirmação de *e-mail* ou até mesmo outro tipo de informação que deve ser passada.

A última ferramenta do FireBase que foi utilizada foi o *Test Lab* que possibilita o teste do aplicativo tanto para Android quanto para iOS e está representada pela Figura 18. O que chama atenção nesse recurso é o fato de que podemos escolher em qual aparelho será o teste, se queremos aparelhos físicos ou virtuais e até mesmo colocar *scripts* de execução para testes específicos. Vamos abordar os resultados dos testes no Capítulo 4, porém, no

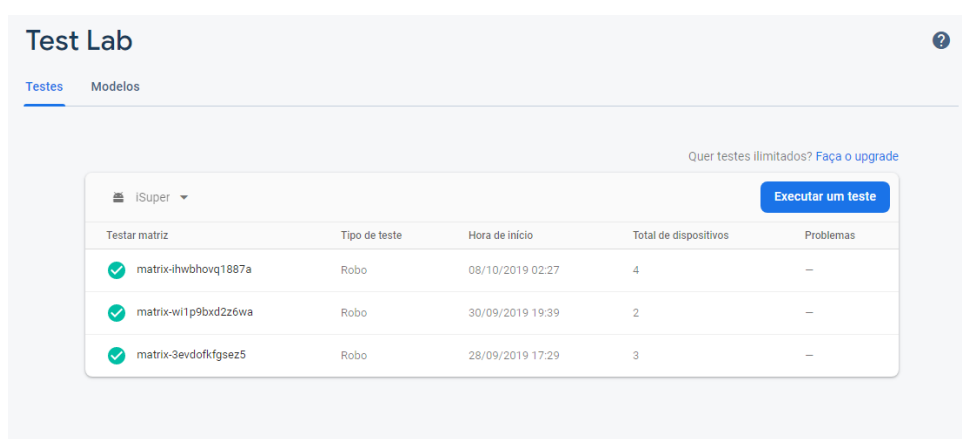
Figura 17 – Usuários no FireBase *Authentication*


The screenshot shows the 'Authentication' page in the Firebase console, specifically the 'Usuários' (Users) tab. It features a search bar at the top with the placeholder text 'Pesquise por endereço de e-mail, número de telefone ou UID do usuário'. Below the search bar is a table listing users. The table has five columns: 'Identificador', 'Provedores', 'Criado em', 'Conectado', and 'UID do usuário'. There are 9 users listed in the table.

Identificador	Provedores	Criado em	Conectado	UID do usuário
teste01@gmail.com	✉	14 de set. de ...	14 de set. de ...	0sMjYDOrd2ZCkTh4gEW6o7FpoRs2
phmbbatista@gmail.com	✉	1 de out. de 2...	1 de out. de 2...	2WK9gB8GhrSwJWYc4fgCAFKSlo...
ana.morais@gmail.com	✉	17 de set. de ...	17 de set. de ...	3bVnq4d9Kgb8J6WbiQmievXpMJF3
teste55@gmail.com	✉	22 de set. de ...	22 de set. de ...	50LBWWJl5qRB75ltyx5TVFYJ0au2
icarob@hotmail.com	✉	30 de set. de ...	1 de out. de 2...	6L9aof7KnAYpsbCRROJiCycVFh82
teste@hotmail.com	✉	22 de set. de ...	23 de set. de ...	7vIZt9bCNhYOs28NXGVBCgu03U...
euler@gmail.com	✉	19 de set. de ...	23 de set. de ...	Ayh8o4dvjfhNaEawznCT1A1LQYj2
testeaaa@gmail.com	✉	14 de set. de ...	14 de set. de ...	DSR5JfcFA7QoVQJ0jkkdiwBi8Wl2
icarobicalho@hotmail.com	✉	14 de set. de ...	24 de set. de ...	HGsBibZ40nUTZu5eCUh7intC7s32

Fonte: (FIREBASE, 2019)

nosso caso, eles foram limitados aos dispositivos Android através do APK criado. Para os testes no *Test Lab* de iOS seria necessário o pacote de XCTest do nosso aplicativo, o que não era possível devido a opção de plataforma de desenvolvimento escolhida. Assim, os testes no iOS foram feitos por outros desenvolvedores.

Figura 18 – Testes executados no FireBase *Test Lab*


The screenshot shows the 'Test Lab' page in the Firebase console, specifically the 'Testes' (Tests) tab. It features a dropdown menu for the test target, currently set to 'iSuper'. Below the dropdown is a table listing test results. The table has five columns: 'Testar matriz', 'Tipo de teste', 'Hora de início', 'Total de dispositivos', and 'Problemas'. There are 3 test results listed in the table.

Testar matriz	Tipo de teste	Hora de início	Total de dispositivos	Problemas
matrix-ihwbhovq1887a	Robo	08/10/2019 02:27	4	—
matrix-wi1p9bxd2z6wa	Robo	30/09/2019 19:39	2	—
matrix-3evdofkfgsez5	Robo	28/09/2019 17:29	3	—

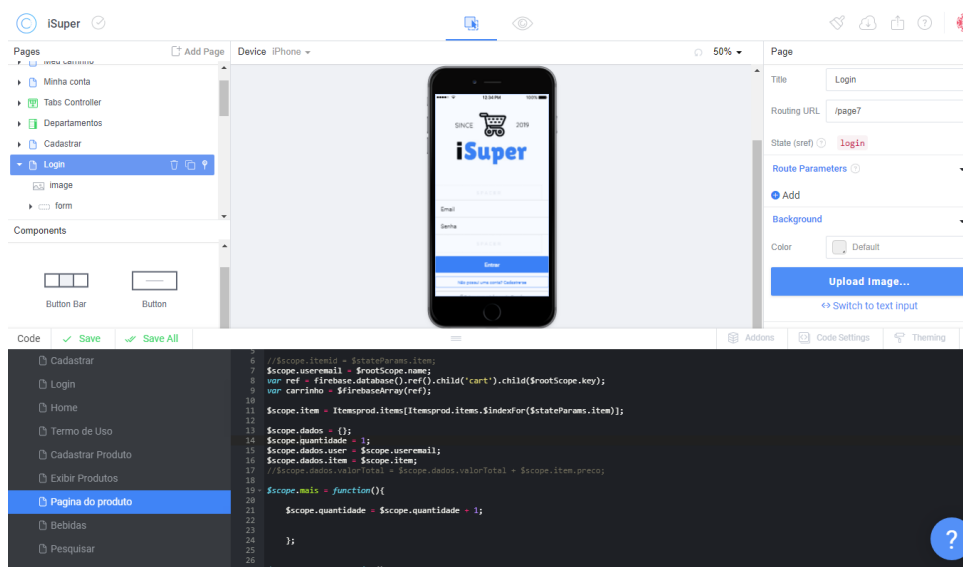
Fonte: (FIREBASE, 2019)

3.7 Ionic Creator

Inicialmente, o aplicativo seria desenvolvido usando o editor de código-fonte Visual Studio Code da Microsoft. Entretanto, optamos por mudar para o Ionic Creator que possui ferramentas que aceleram o processo de desenvolvimento. A programação é feita em nuvem, ou seja, não temos o risco de perder nada do código e, além disso, em sua versão paga, que foi a utilizada em nosso trabalho, ele oferece a hospedagem do nosso aplicativo multiplataforma.

O Ionic Creator, [Figura 19](#), nos mostra em tempo real como cada página fica enquanto estamos desenvolvendo, além de nos dar a opção de escolher visualizar em iPhone ou Android, como percebemos na [Figura 20](#) e na [Figura 21](#).

Figura 19 – Ionic Creator

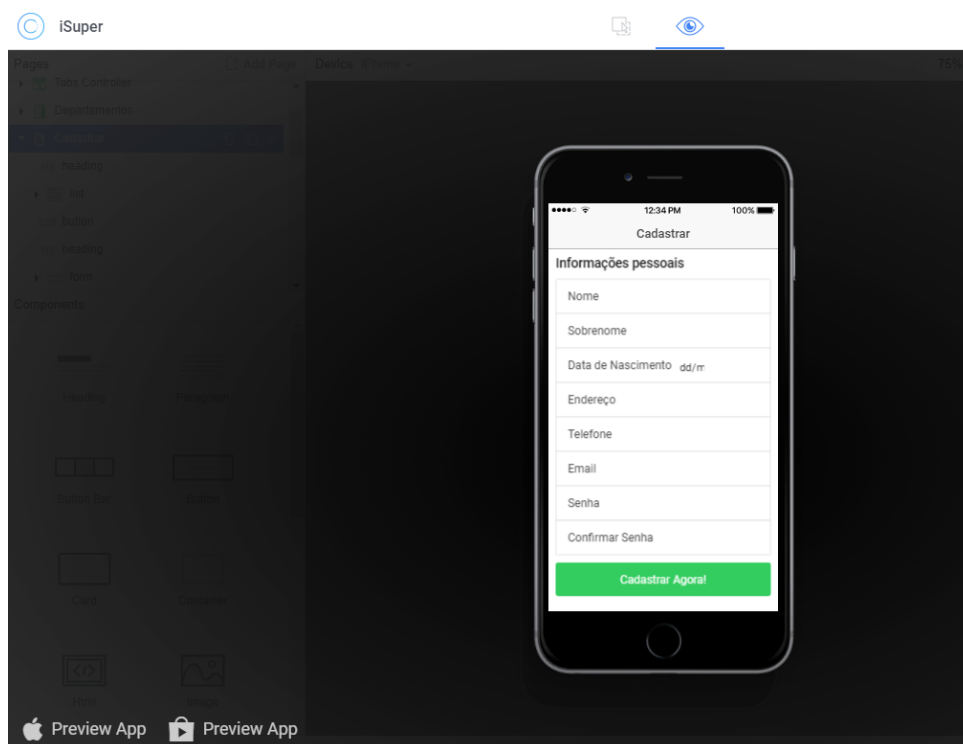


Fonte: Elaborado pelo autor (2019)

Para que seja feita a conexão com o Firebase, temos que passar 4 variáveis de identificação. Este é um modo de proteger o banco de dados e nossos clientes de invasores. Após a conexão, a comunicação com o Firebase é feita no código-fonte de cada tela desenvolvida, assim como podemos observar na [Figura 22](#), onde temos a *controller* da tela de cadastro do *software*. Na linha 4, mostramos para o Firebase qual tabela usaremos. Caso não exista essa tabela, ele a cria com os dados que são passados. Na linha 27, usamos o `firebaseAuth` para criar um usuário. Observe que ele só é criado se a confirmação de senha for igual à senha (linha 21). A função `createUserWithEmailAndPassword` utilizada verifica se o *e-mail* é válido e envia uma mensagem de confirmação. Por fim, nas linhas 36 e 37, criamos o usuário no banco de dados do firebase e redirecionamos para a página de login.

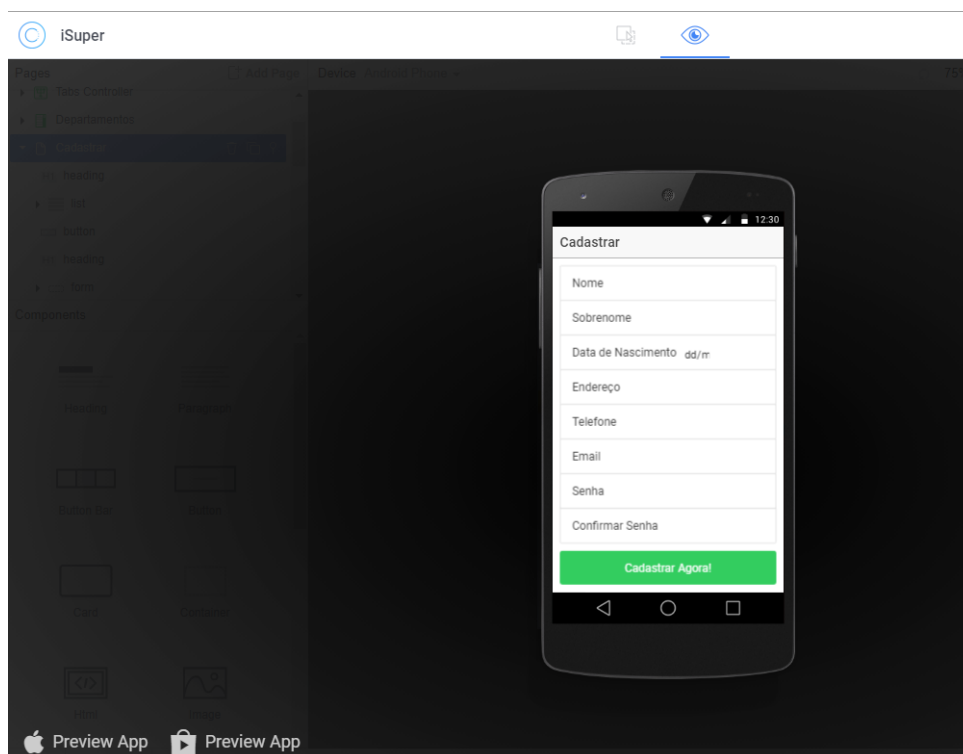
A [Figura 23](#) nos mostra a *controller* da tela de Login do sistema, para fazer um

Figura 20 – Visualização da página Cadastro pelo Ionic Creator no iPhone



Fonte: Elaborado pelo autor (2019)

Figura 21 – Visualização da página Cadastro pelo Ionic Creator no Android



Fonte: Elaborado pelo autor (2019)

paralelo com a página de cadastro. Na linha 7, verificamos se o usuário está cadastrado no sistema e, caso esteja, ele é direcionado para a função da linha 12, onde iniciamos as variáveis globais do sistema, como, por exemplo, o valor que está no carrinho e direcionamos o usuário para a página *home* do sistema. Caso ocorra algum erro, a função da linha 28 é chamada, onde é exibido um erro na tela relatando que o *e-mail* ou a senha estão errados.

O único ponto negativo do Ionic Creator se deu quando tentamos exportar o aplicativo para outra plataforma, como por exemplo o Ionic Studio ou o Visual Studio Code, pois ele executa uma versão antiga do Ionic o que não interfere no resultado final do nosso trabalho, porém causou uma limitação. Por exemplo, houve problemas na personalização do aplicativo em alguns aspectos, como a não utilização das cores que foram escolhidas no protótipo de telas ou a criação do pacote de XCTest, que era necessário para o desenvolvimento de testes melhores dentro do Firebase *Test Lab*, principalmente no que diz respeito ao comportamento do aplicativo executando no iPhone.

Figura 22 – Trecho do código-fonte da página Cadastro

```

1  function ($scope, $stateParams, $firebaseArray, $firebaseAuth,
2      $state, $firebase, $ionicPopup) {
3      var auth = $firebaseAuth();
4      var ref = firebase.database().ref('usuarios');
5      var usuarios = $firebaseArray(ref);
6
7      $scope.dados = {};
8      $scope.dados2 = {};
9      $scope.dados.nome = $scope.dados.nome;
10     $scope.dados.sobrenome = $scope.dados.sobrenome;
11     $scope.dados.datanas = $scope.dados.datanas;
12     $scope.dados.endereco = $scope.dados.endereco;
13     $scope.dados.telefone = $scope.dados.telefone;
14     $scope.dados.avatar = $stateParams.ProfilePic;
15     $scope.dados.email = $scope.dados.email;
16     $scope.dados.senha = $scope.dados.senha;
17     $scope.dados2.senha2 = $scope.dados2.senha2;
18
19     $scope.cadastrar = function() {
20
21         if($scope.dados.senha != $scope.dados2.senha2){
22             var alertPopup = $ionicPopup.alert({
23                 title: 'Erro!',
24                 template: 'A confirmação de senha deve ser igual a senha.',
25             });
26         } else {
27             auth.$createUserWithEmailAndPassword($scope.dados.email,
28                 $scope.dados.senha).then($scope.cadastrarSucesso);
29         };
30         $scope.cadastrarSucesso = function() {
31
32             var alertPopup = $ionicPopup.alert({
33                 title: 'Cadastrado!',
34                 template: 'Cadastro efetuado com sucesso!'
35             });
36             usuarios.$add($scope.dados);
37             $state.go('login');
38         };
39     }

```

Fonte: Elaborado pelo autor (2019)

Figura 23 – Trecho do código-fonte da página de Login

```
1 function ($scope, $stateParams, $state, $firebaseAuth, $window,  
2     $ionicPopup, $rootScope) {  
3     var auth = $firebaseAuth();  
4     $scope.dados = {};  
5  
6     $scope.login = function() {  
7         auth.$signInWithEmailAndPassword($scope.dados.email, $scope.  
8             dados.senha).then($scope.loginSucesso).catch($scope.  
9                 loginErro);  
10  
11         // if (!$scope.dados.email) { } else {}  
12     };  
13  
14     $scope.loginSucesso = function() {  
15         $rootScope.name = $scope.dados.email;  
16         $rootScope.valorTotal = 0;  
17         $rootScope.contalim = 0;  
18         $rootScope.contbeb = 0;  
19         $rootScope.contcarne = 0;  
20         $rootScope.contcozi = 0;  
21         $rootScope.contcong = 0;  
22         $rootScope.contlimp = 0;  
23         $rootScope.flag = -1;  
24  
25         $state.go('tabsController.home_tab1');  
26     };  
27  
28     $scope.loginErro = function() {  
29         var erroMsg = "Email ou senha inv lidos!";  
30         var titulo = "Erro ao tentar logar";  
31         $scope.showAlert(titulo, erroMsg);  
32     };  
33  
34     $scope.showAlert = function(titulo, erroMsg) {  
35         var alertPopup = $ionicPopup.alert({  
36             title: titulo,  
37             template: erroMsg  
38         });  
39         alertPopup.then(function(res) {  
40             console.log('Err');  
41         });  
42     };  
43  
44     // $scope.googleIn = function() {  
45     };  
46 }
```

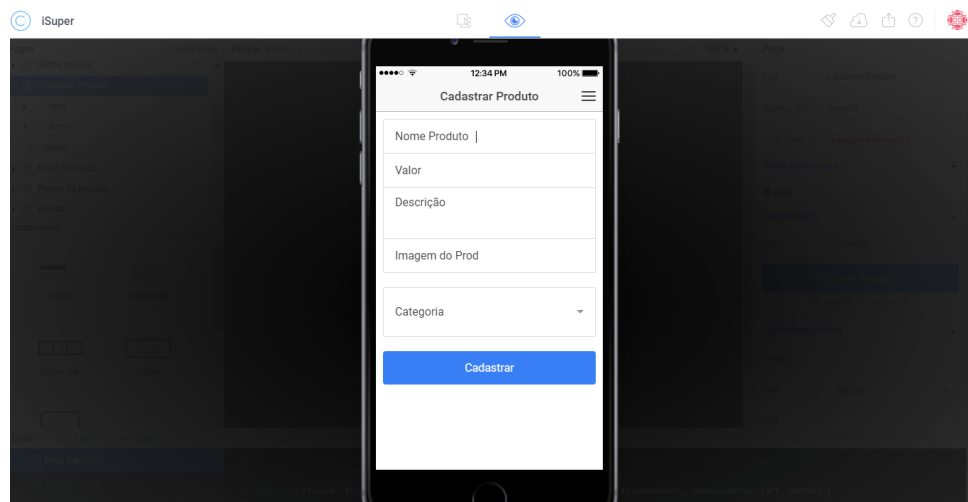
4 Resultados

Este capítulo aborda os resultados obtidos com a aplicação *iSuper*, bem como uma comparação entre os aplicativos Android e iOS obtidos. Será discutido também os testes e os resultados obtidos através dos mesmos.

4.1 O aplicativo

O aplicativo, como descrito nos capítulos anteriores, foi desenvolvido utilizando as ferramentas Ionic Creator Pro e o FireBase. Visando mais praticidade e segurança, os produtos são inseridos na aplicação apenas pelo Ionic Creator, e não tem conexão com o aplicativo. A adição de produtos pode ser feita numa tela separada, como podemos ver na [Figura 24](#). Dessa forma, apenas os desenvolvedores têm acesso à adição, edição ou remoção de algum produto, o que garante segurança para quem usa o *software*. As subseções dessa seção estão divididas da seguinte forma: todo o funcionamento do aplicativo será explicado considerando a versão para o Android e, na subseção seguinte, vamos apontar algumas diferenças das versões para o Android para o iOS.

Figura 24 – Cadastro de produto via Ionic Creator



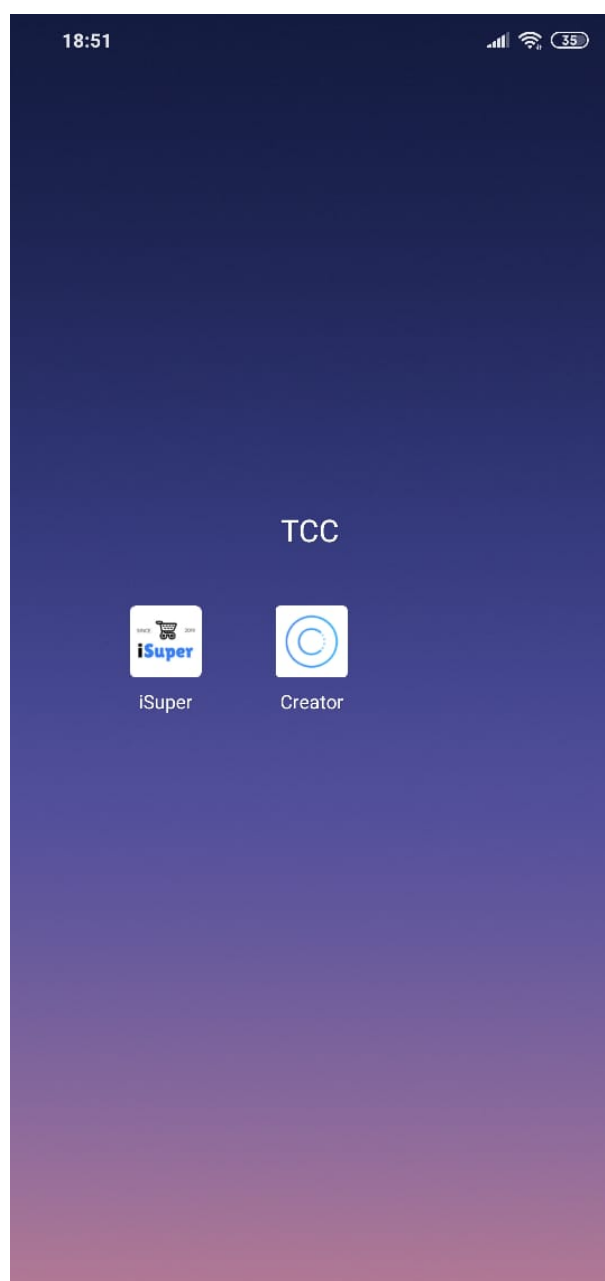
Fonte: Elaborado pelo autor (2019)

4.1.1 Login

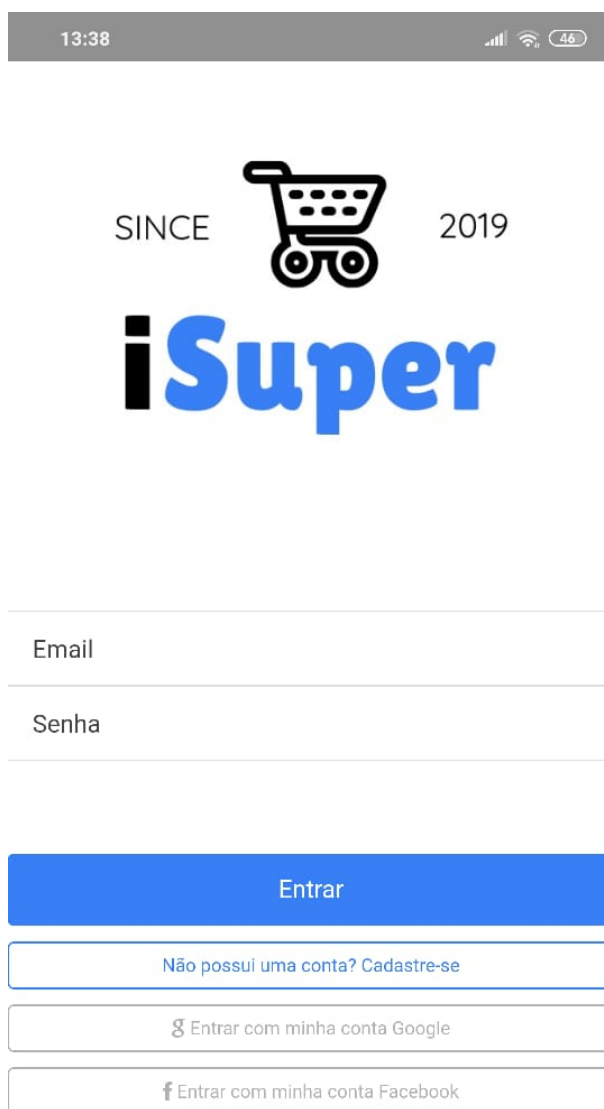
Os dispositivos Android são, sem dúvida, os mais usados atualmente com cerca de 87,8% do mercado mundial ([GUNAWAN, 2019](#)). Vamos navegar pelo aplicativo nessa subseção com as capturas de telas de um celular Android modelo Xiaomi Note 6 Pro e

discutir alguns fluxos dentro do *software*. A Figura 25 mostra o aplicativo dentro de uma pasta no sistema Android, logo após clicarmos para iniciá-lo o usuário é direcionado para a página de *Login*, como podemos observar na Figura 26. Na tela de *Login*, o usuário tem algumas opções, como *logar* pela conta Google, cadastrar e entrar no sistema com seu *e-mail* e senha.

Figura 25 – Aplicativo no sistema Android



Fonte: Elaborado pelo autor (2019)

Figura 26 – Página de *Login* do aplicativo no Android

13:38

SINCE  2019

iSuper

Email

Senha

Entrar

[Não possui uma conta? Cadastre-se](#)

 Entrar com minha conta Google

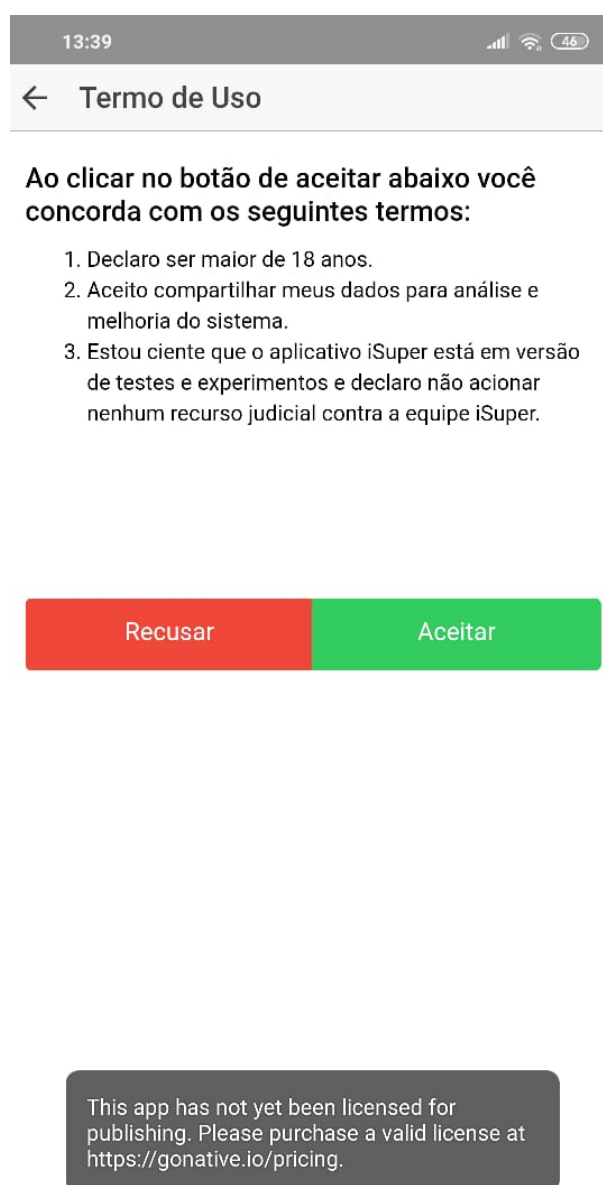
 Entrar com minha conta Facebook

Fonte: Elaborado pelo autor (2019)

4.1.2 Cadastro de usuário

Para dar início, vamos tomar como exemplo um usuário novo: ao clicar para criar uma nova conta, ele irá para a tela da Figura 27, onde ele deve aceitar os termos de uso do nosso *software*, esse processo serve para resguardar o aplicativo de qualquer tipo de problema judicial. Caso o usuário não se sinta confortável com os termos descritos e decida recusar, ele é redirecionado para a página de *login*.

Figura 27 – Página de Termo de Uso do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Caso a decisão do usuário seja aceitar os termos, o cadastro continua e ele deve informar seus dados, como na Figura 28. Além dos dados pessoais, como nome e endereço,

o usuário deve obrigatoriamente inserir um *e-mail* válido e a senha correta duas vezes. Caso alguma dessas duas requisições não seja satisfeita, o cadastro falha e indica o que deu errado. É opcional para o usuário também, escolher um avatar pré-determinado, como podemos ver na [Figura 29](#). Não foi possível deixar o usuário escolher uma foto sua ou personalizada como avatar pelo fato de que o armazenamento de dados dentro do Firebase é limitado e isso iria consumir boa parte do banco de dados gratuito disponível para a construção do nosso sistema.

Figura 28 – Página de Cadastro de Usuário do aplicativo no Android

13:39

← Cadastrar

Avatar de perfil

 Avatar atual

Alterar

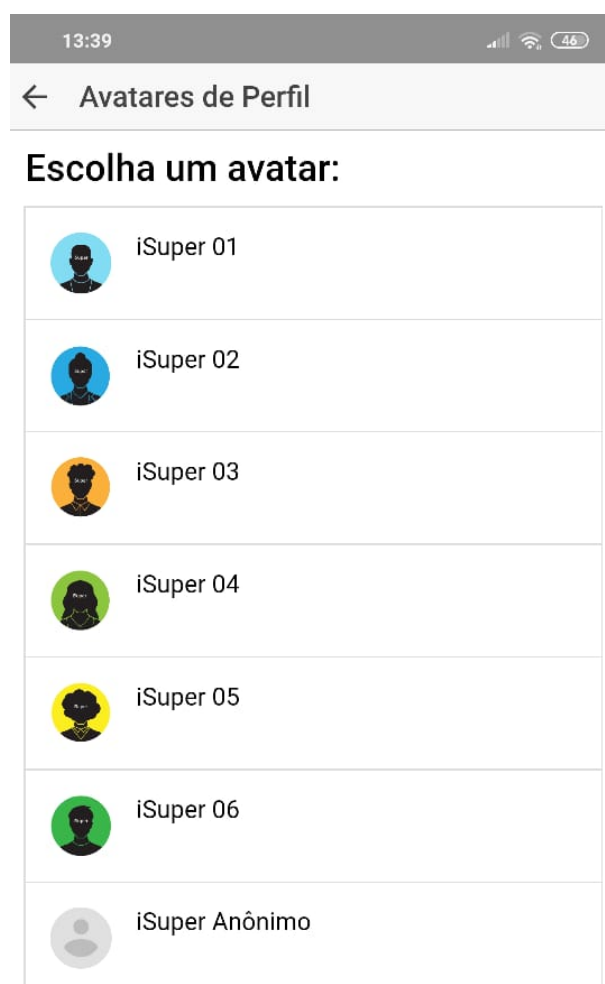
Informações pessoais

Nome
Sobrenome
Data de Nascimento
Endereço
Telefone
Email
Senha
Confirmar Senha

Cadastrar Agora!

Fonte: Elaborado pelo autor (2019)

Figura 29 – Página de escolha de avatar do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

4.1.3 Página inicial

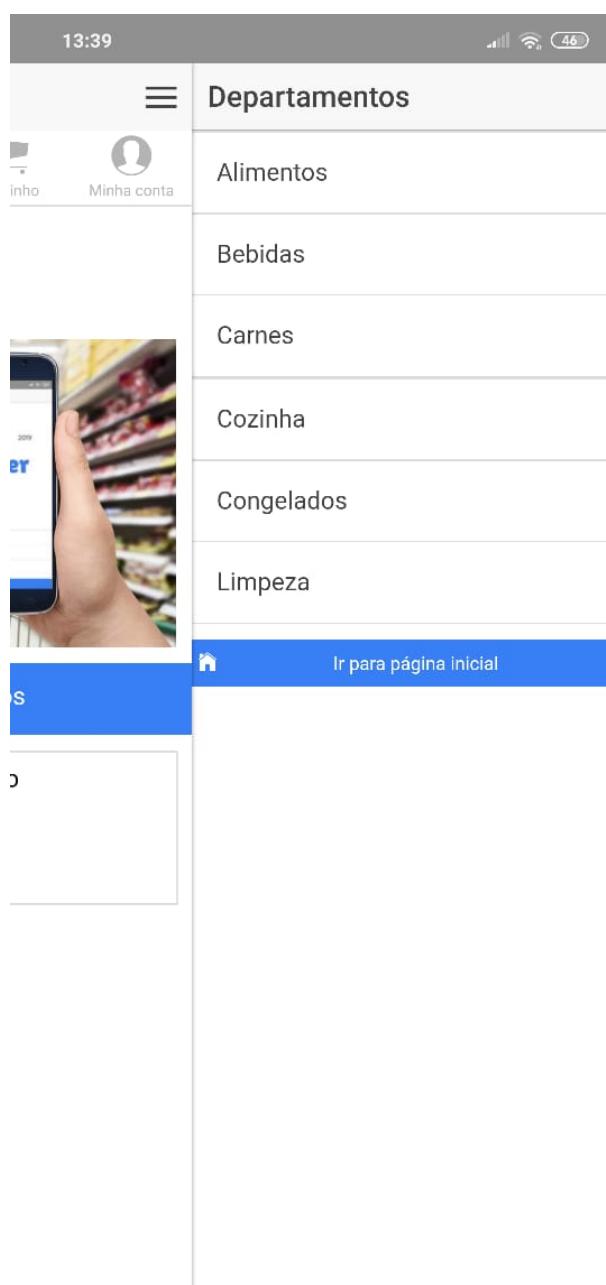
Após todo o processo de cadastro descrito, o usuário é direcionado novamente para a página de *login* e pode acessar o sistema com seu *e-mail* e senha. Caso ele digite alguma das duas informações de maneira errada, uma mensagem aparece na tela indicando tal erro. Em caso de sucesso, ele segue para a página inicial do nosso sistema, como vemos na [Figura 30](#). Nessa tela, temos alguns *banners*, a opção para que o usuário veja todos os produtos e um local onde ele é redirecionado para prestar *feedbacks* sobre o aplicativo, que foi muito usado durante os testes com outros desenvolvedores em que iremos abordar mais a frente nesse capítulo.

A barra de navegação, nesse caso, superior também aparece pela primeira vez no sistema. Nesta tela, além da página inicial, temos a aba de sugestões, o botão de pesquisar centralizado, o carrinho e a página da conta de usuário. Acima dessa barra de navegação, temos o nome da tela e no canto superior direito o botão para acessar os departamentos do nosso mercado virtual, como podemos ver na [Figura 31](#). A partir dessa divisão em categorias, podemos acessar alguma específica, como a [Figura 32](#) exhibe. Por fim, ao clicar em algum dos produtos dentro da categoria o usuário chega à tela do produto, como vemos na [Figura 33](#). Nesta tela, ele escolhe a quantidade e pode adicionar ao carrinho. Essa página de produto foi elaborada da seguinte forma: ela é uma página genérica que recebe, quando o usuário clica em algum produto, a chave do produto e, através dela, ele faz a busca no banco de dados e recupera os dados desse produto, como imagem, preço e descrição. Uma proposta inicial era de que o usuário escrevesse uma avaliação do produto e desse alguma nota para o mesmo, porém não foi possível por gerar uma grande demanda de uso do banco de dados, e como explicado anteriormente, tínhamos que limitá-lo.

Figura 30 – Página de *Home* do aplicativo no Android

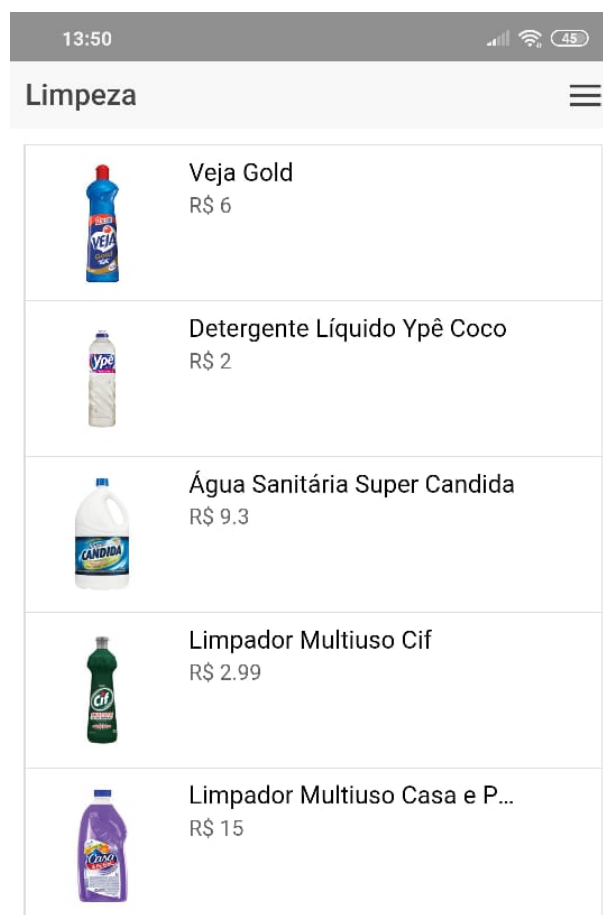
Fonte: Elaborado pelo autor (2019)

Figura 31 – Página de Categorias do aplicativo no Android



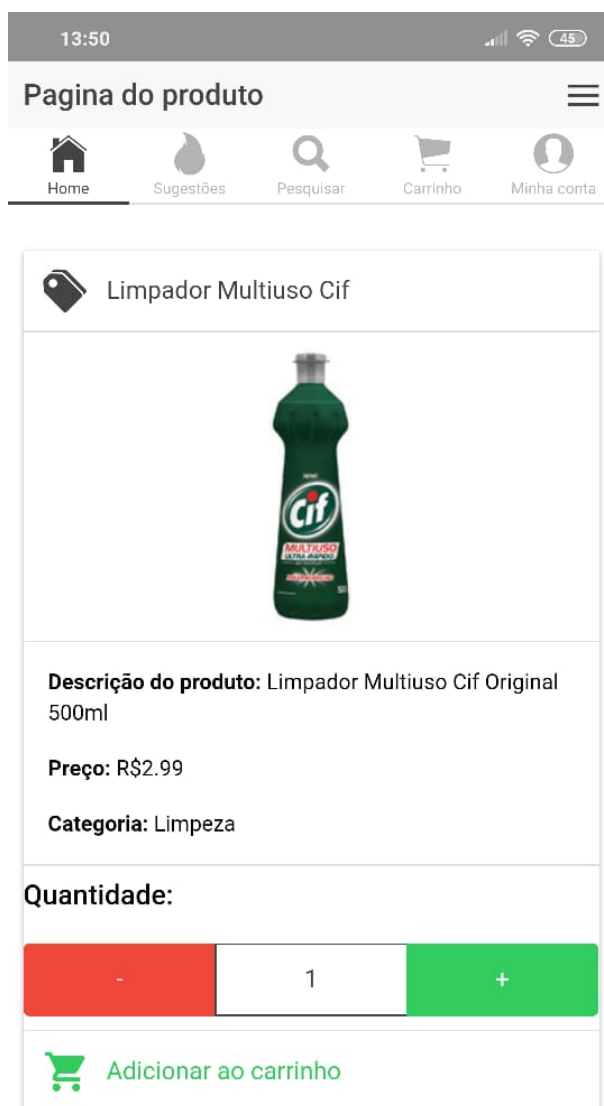
Fonte: Elaborado pelo autor (2019)

Figura 32 – Página da Categoria Limpeza do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 33 – Página de acesso ao produto via Categoria do aplicativo no Android

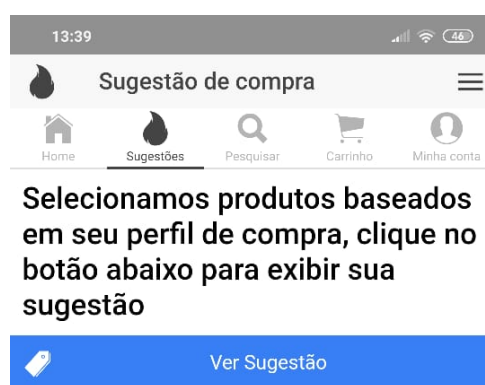


Fonte: Elaborado pelo autor (2019)

4.1.4 Sugestões

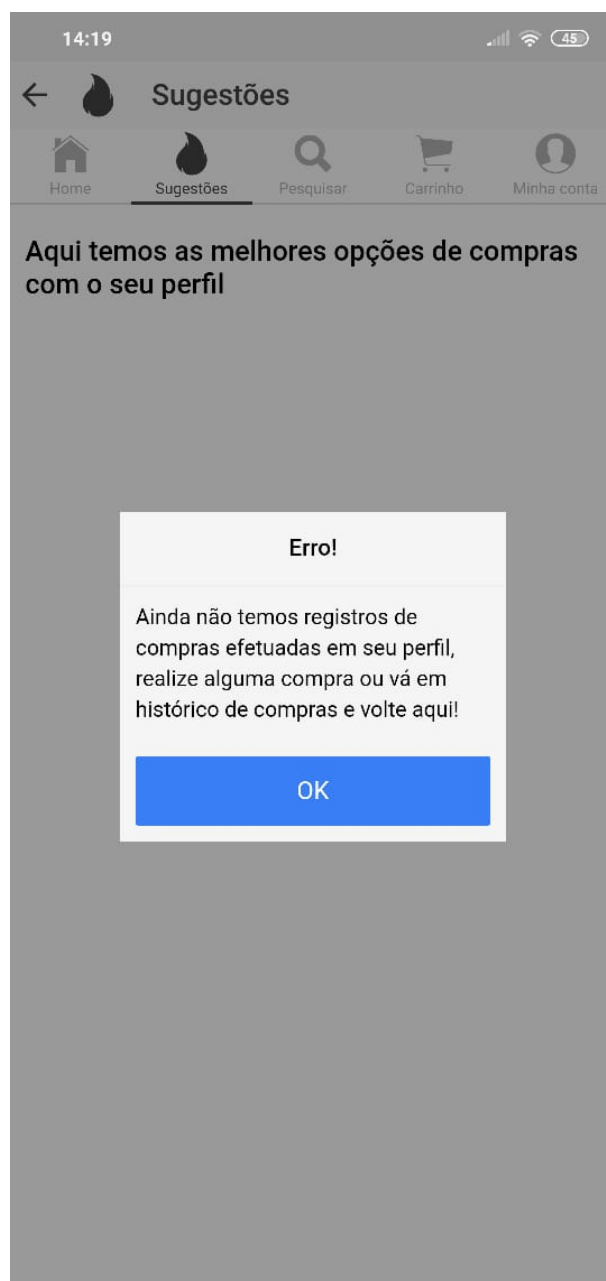
A aba de Sugestões, como observamos na Figura 34, funciona da seguinte maneira: quando o usuário clica para ver a sugestão e ele ainda não efetuou nenhuma compra no sistema, o erro da Figura 35 aparece na tela, por não termos dados suficientes de compras anteriores, não podemos sugerir nada para o usuário. Caso ele já tenha efetuado ao menos uma compra no aplicativo, a tela da Figura 36 aparece. Ela faz uma busca no banco de dados pelos produtos que o usuário comprou em todas as compras efetuadas e calcula em qual categoria ele comprou mais produtos e, a partir daí, passa para a página do produto um chave de produto referente à categoria que o usuário comprou mais produtos, como vemos na Figura 37. A ideia inicial era utilizar técnicas de inteligência artificial para a realização de uma sugestão mais refinada, porém o tempo não era viável para a execução dessa maneira e a forma descrita nesse parágrafo foi a utilizada.

Figura 34 – Página de Sugestão do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 35 – Página de Sugestão fluxo 1 do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 36 – Página de Sugestão fluxo 2 do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 37 – Página de produto sugerido do aplicativo no Android

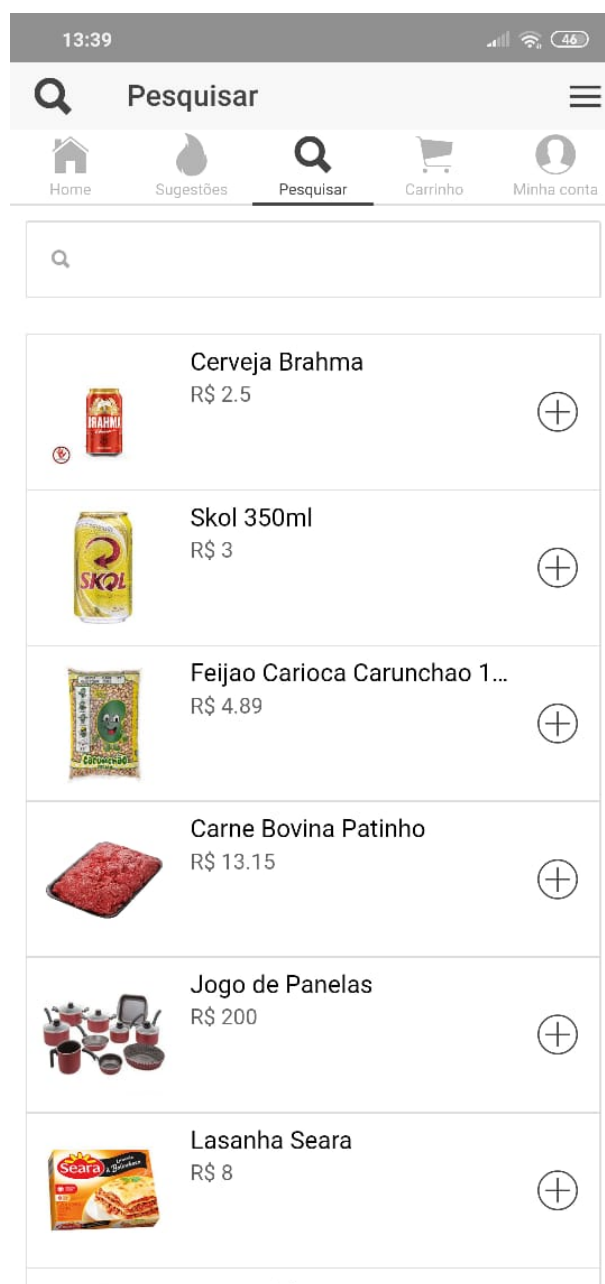


Fonte: Elaborado pelo autor (2019)

4.1.5 Pesquisa

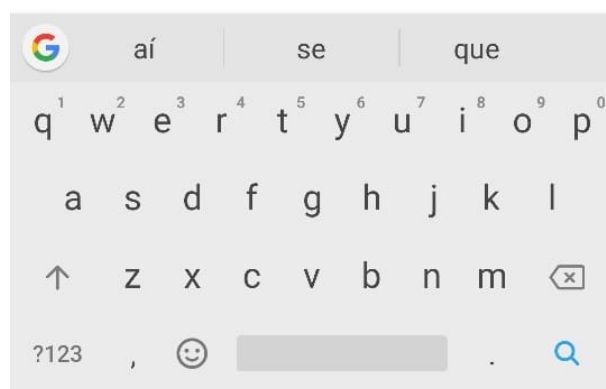
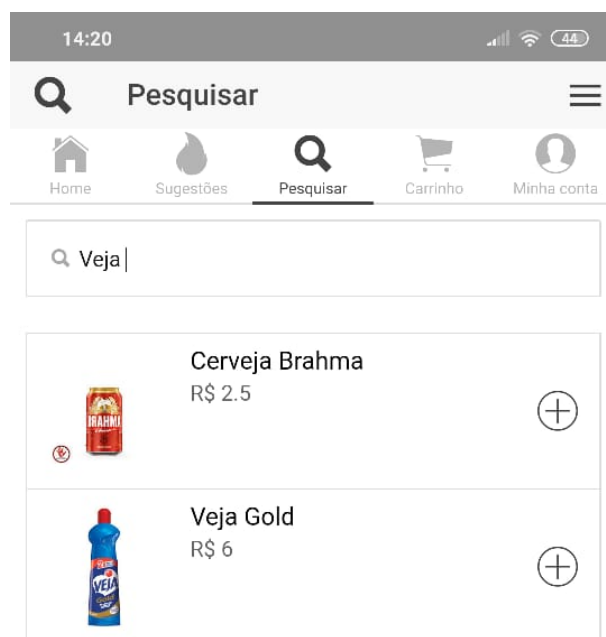
A tela de pesquisa, como visto na [Figura 38](#), exibe uma barra de pesquisa e abaixo todos os produtos cadastrados. Ao digitarmos alguma coisa, como na [Figura 39](#), os produtos são filtrados, sem discriminação de maiúsculo e minúsculo, bem como se o nome do produto contém o que está na busca. Efetuando uma compra ao clicar no produto e selecionar a quantidade desejada, como observamos na [Figura 40](#), ao clicar no botão adicionar ao carrinho, a mensagem da [Figura 41](#) aparece na tela, confirmando a adição do produto.

Figura 38 – Página de Pesquisa do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 39 – Página de Pesquisa do aplicativo no Android (Continuação)



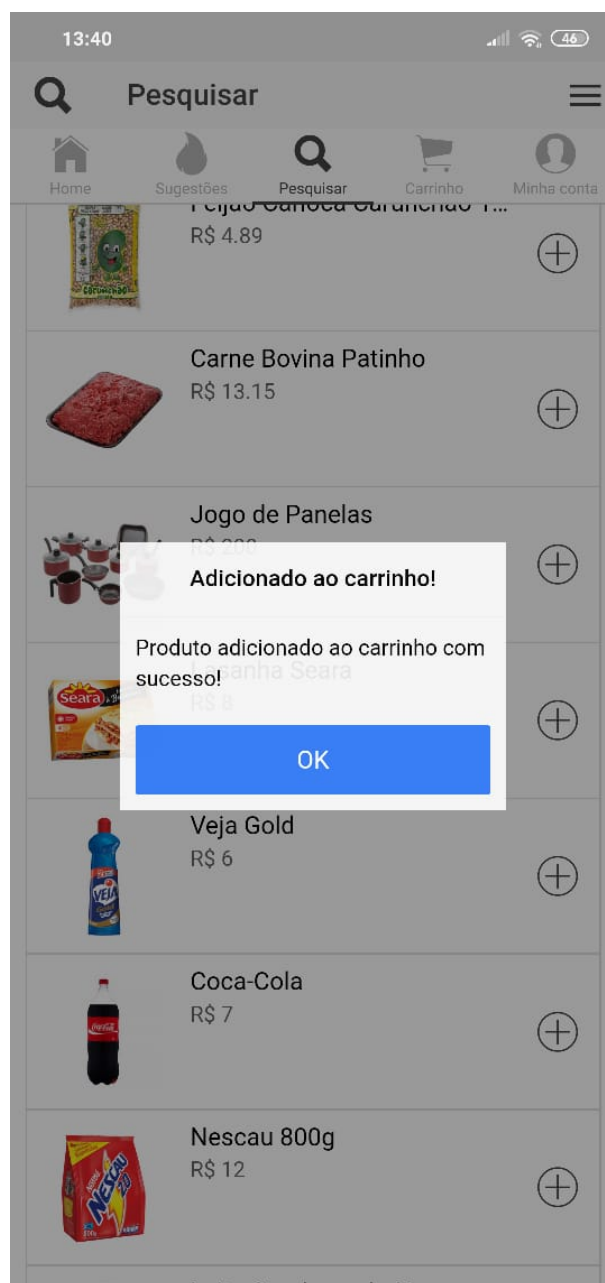
Fonte: Elaborado pelo autor (2019)

Figura 40 – Página de Produto acessada pela Pesquisa do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 41 – Confirmação de Produto adicionado ao Carrinho do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

4.1.6 Carrinho

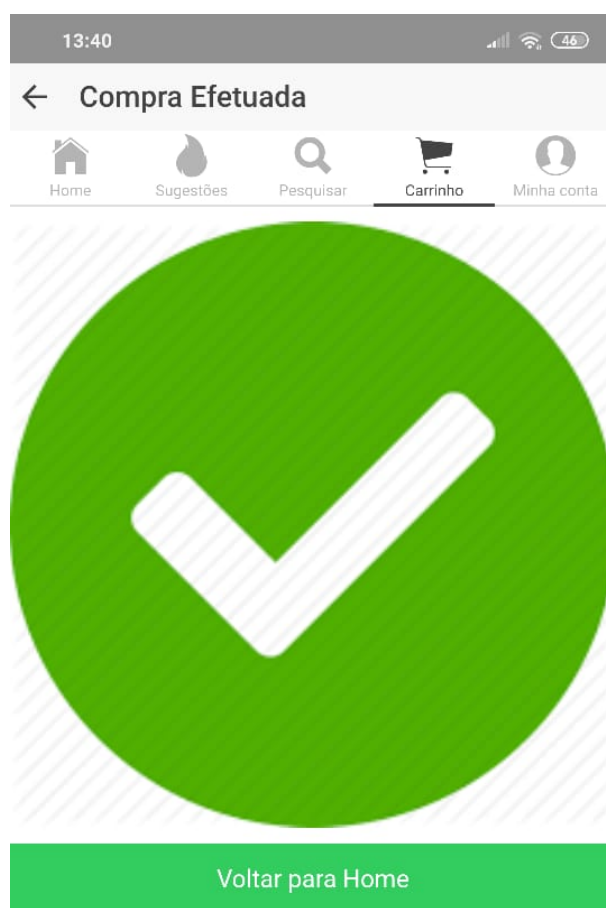
O carrinho de compras da aplicação, como vemos na [Figura 42](#), exibe todos os produtos adicionados ao carrinho e a quantidade de cada um, bem como o valor unitário do produto. Para que o usuário exclua algum produto, ele deve clicar no mesmo, assim a quantidade é diminuída em 1 até zerar, e, quando isso acontece, o produto some da lista. Podemos ver também a barra para confirmar a compra, e nela temos o valor total dos itens listados, que é alterado conforme a adição ou remoção de cada unidade de produto. Ao clicar nesse botão, a ideia é que o usuário já tenha algum cartão cadastrado, o que não foi implementado, e então ele efetue a compra com um clique. Sendo assim, ele é redirecionado para a tela da [Figura 43](#) e finaliza o ciclo de compras do sistema.

Figura 42 – Página do Carrinho do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 43 – Página de Compra Efetuada do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

4.1.7 Minha conta

O usuário pode ver e editar seus dados, isso é possível ser feito na página da conta de usuário, como vemos na Figura 44, ele não consegue ver sua senha, porém todos os outros dados estão disponíveis para ele. Caso ele clique em alterar dados, a tela passa a ser como a Figura 45, e ele pode alterar todos os dados, como o *e-mail* de acesso e senha, o telefone e o endereço. Não disponibilizamos a mudança de nome e sobrenome por não enxergarmos a necessidade.

Figura 44 – Página de Perfil de Usuário do aplicativo no Android



Fonte: Elaborado pelo autor (2019)

Figura 45 – Página de Perfil de Usuário para editar dados do aplicativo no Android

14:04

Minha conta

Home Sugestões Pesquisar Carrinho Minha conta

Olá! Icaro Quintão

 **Icaro**
Quintão 94656565646

Nome: Icaro
Sobrenome: Quintão
Telefone: 94656565646
E-mail: icaro.quintao@aluno.ufop.edu.br
Endereço: Rua Café-Filho número 456 Bairro São José

Alterar meus dados pessoais

Digite os dados novos:

Email Novo
Senha Nova
Telefone Novo
Endereço Novo

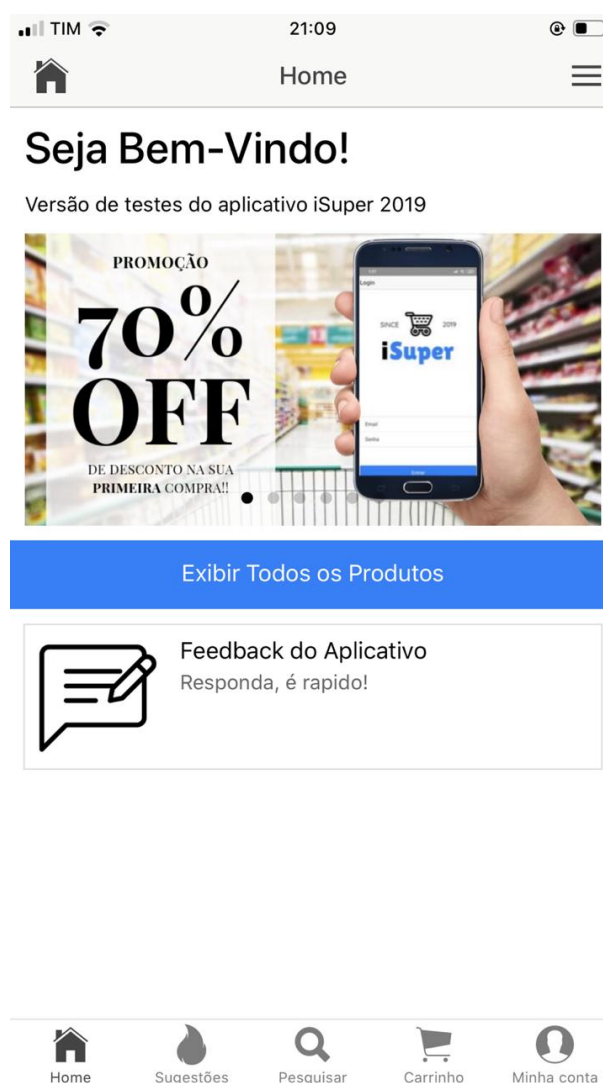
Atualizar **Cancelar**

Fonte: Elaborado pelo autor (2019)

4.1.8 iPhone

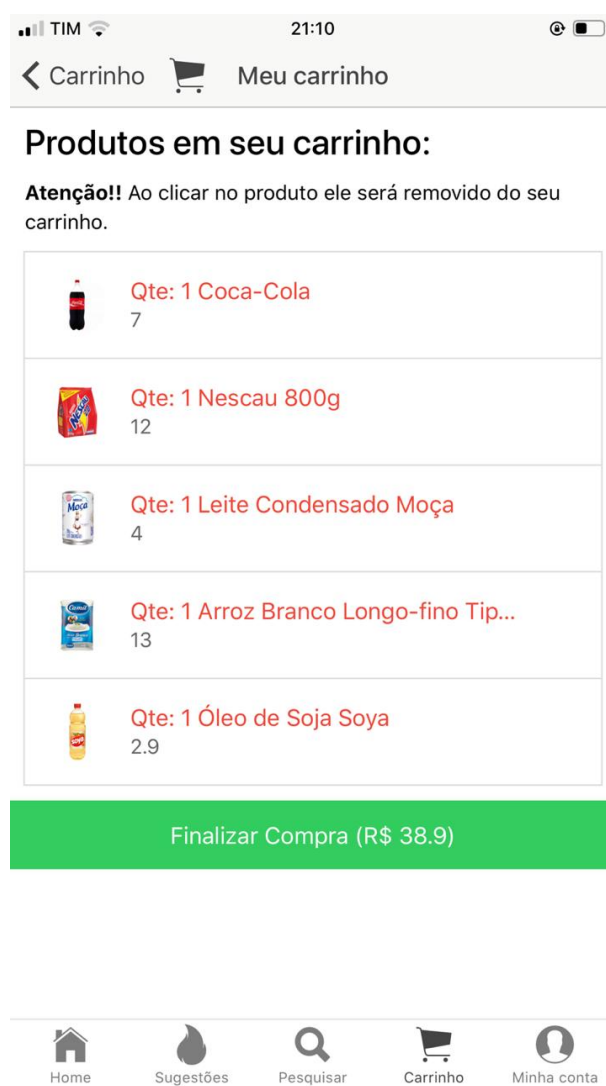
O dispositivo utilizado para efetuar as execuções foi o iPhone 8 Plus com a versão 13.1.3 do iOS. Para mostrar as diferenças existentes, vamos analisar algumas telas para comparação com o que foi visto. A [Figura 46](#) nos mostra a página inicial do aplicativo e a diferença que chama mais atenção é a barra de navegação ser inferior. Outro ponto diferente é o tamanho da fonte, que é um pouco maior do que nos dispositivos Android. A [Figura 47](#) mostra um carrinho de compras com alguns produtos e, além da barra de navegação, no canto superior esquerdo, ele mostra a página anterior em que o usuário se encontrava. Por fim, na [Figura 48](#), percebemos que o título da página está centralizado e no Android ele ficava à esquerda da página (como já vimos na [Figura 44](#)).

Figura 46 – Página inicial do aplicativo no iPhone



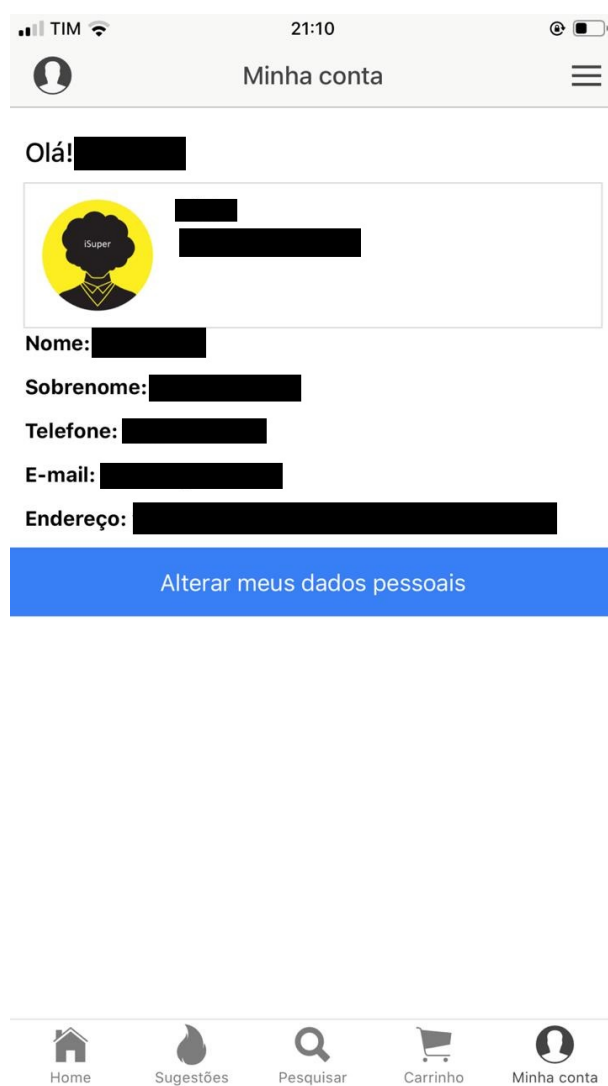
Fonte: Elaborado pelo autor (2019)

Figura 47 – Página do carrinho do aplicativo no iPhone



Fonte: Elaborado pelo autor (2019)

Figura 48 – Página de Perfil de Usuário do aplicativo no iPhone



Fonte: Elaborado pelo autor (2019)

4.2 Testes

Para que o *software* seja validado, foram realizados diversos tipos de testes. O primeiro foi disponibilizado pelo Firebase que é conhecido como *Monkey Testing*. Já no segundo, alguns desenvolvedores foram chamados para testar o aplicativo e avaliá-lo de maneira mais crítica. Por fim, os clientes em potencial experimentaram o aplicativo.

4.2.1 FireBase *Test Lab*

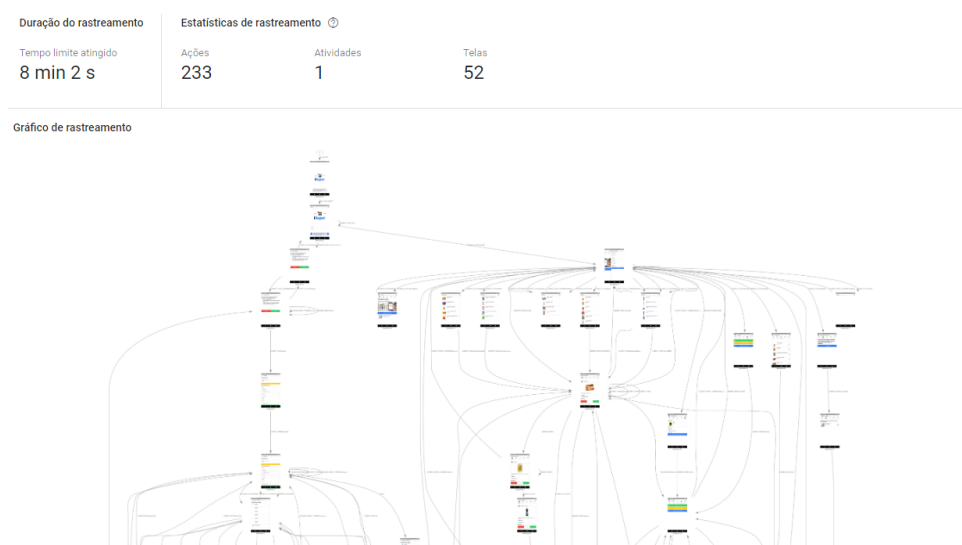
O teste inicial se deu através da ferramenta *Test Lab* ofertada pelo Firebase. Após enviarmos o APK do nosso aplicativo, escolhemos quais dispositivos queremos para que o teste seja feito. Podemos escolher entre dispositivos físicos ou virtual. A diferença maior é que o dispositivo virtual não gera gráficos de desempenho. Por usarmos a versão gratuita do Firebase, nossos testes eram limitados diariamente em quantidade. Dessa forma, no total, foram realizados em torno de 20 testes em modelos diversos de aparelhos. Para cada teste, temos acesso aos problemas encontrados, temos a opção de inserir algum *script* de execução, o fluxo de telas que o teste seguiu, como vemos na [Figura 49](#), bem como quantas ações foram feitas, o tempo de duração do teste, as telas que foram acessadas e um vídeo de todo esse processo. Além disso, temos o mais interessante dessa ferramenta, o desempenho, ou seja, como o nosso *software* se comportou no ambiente daquele dispositivo. Como não utilizamos nenhum *script* de execução, o teste foi feito de forma aleatória, ou seja, os botões eram clicados de forma randômica.

Tomemos como exemplo dois dos testes realizados, um com o *smartphone* Pixel 2 da Google e o segundo com o Galaxy S9+ da Samsung. Sabe-se que o aparelho do segundo teste possui componentes mais rápidos do que o primeiro, isso por que é um aparelho moderno e lançado tempos depois do Pixel. Assim, o tempo para exibição inicial do primeiro é maior do que o do segundo, como podemos observar nas figuras [Figura 50](#) e na [Figura 53](#). A [Figura 51](#) e a [Figura 54](#) nos mostram os detalhes de cada dispositivo. No lado direito, são exibidas todas as ações executadas e, no lado esquerdo, é exibido o vídeo.

Por fim, os gráficos da [Figura 52](#) e da [Figura 55](#) nos mostram, ao decorrer do tempo, o comportamento da CPU, memória e acesso à *Internet*. No que diz respeito à CPU, percebemos que o Galaxy S9+ consegue realizar as tarefas de maneira mais rápida e dessa forma ele fica milésimos de segundos ocioso, o que não acontece no primeiro teste. Já a memória têm acesso constante durante toda a execução do *software*. O acesso à Internet por sua vez, só é utilizado quando o teste abre telas que necessitam do banco de dados, como a tela de pesquisa ou a da conta de usuário, por isso o gráfico têm pontos de pico específicos.

O grande ponto positivo desses testes foi o fato de que em alguns deles o robô conseguia acessar o botão de categorias antes do usuário estar logado, ou seja, um *bug* no

Figura 49 – Gráfico de rastreamento



Fonte: Elaborado pelo autor (2019)

Figura 50 – Detalhes do dispositivo do teste 1



Fonte: Elaborado pelo autor (2019)

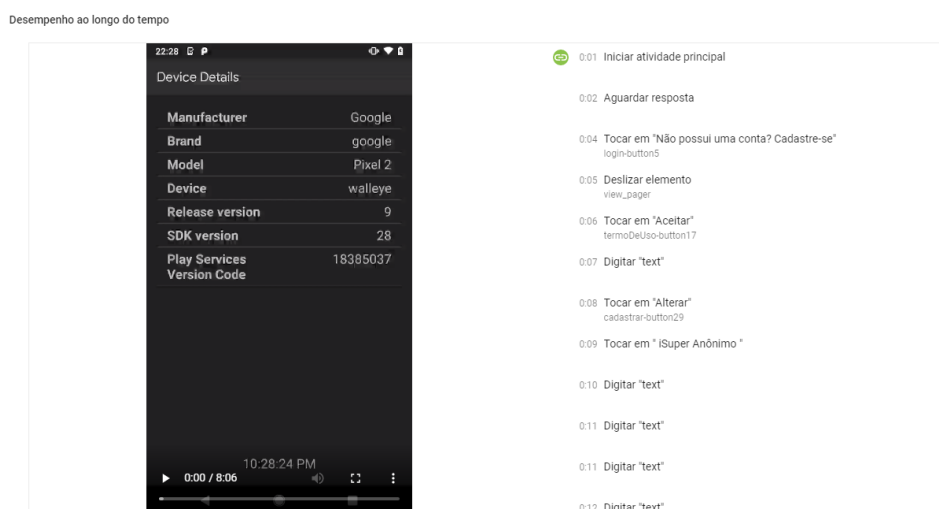
aplicativo que foi corrigido devido à esses testes. Outro ponto interessante é ter informações de desempenho do aplicativo. Dessa maneira, podemos trabalhar em cima de pontos falhos e operações que possam causar lentidões no nosso sistema.

4.2.2 Testes por outros desenvolvedores

Nesta etapa do trabalho, convidamos três desenvolvedores do Instituto de Ciências Exatas Aplicadas, da Universidade Federal de Ouro Preto, para conhecer o sistema e fazer, se necessário as devidas sugestões de mudanças. Por razões de ética os nomes dos mesmos serão preservados.

Sucintamente, as respostas à essa etapa do trabalho foram positivas e agregaram, principalmente em pontos fracos do sistema que não tínhamos visto antes. O primeiro *bug* relatado foi de que, se caso o usuário visualizasse seu carrinho e fosse direto para a pesquisa, por exemplo, ao adicionar novos produtos, eles apareciam na lista de compras.

Figura 51 – Detalhes do dispositivo do teste 1 parte 2



Fonte: Elaborado pelo autor (2019)

Figura 52 – Desempenho ao longo do tempo teste 1



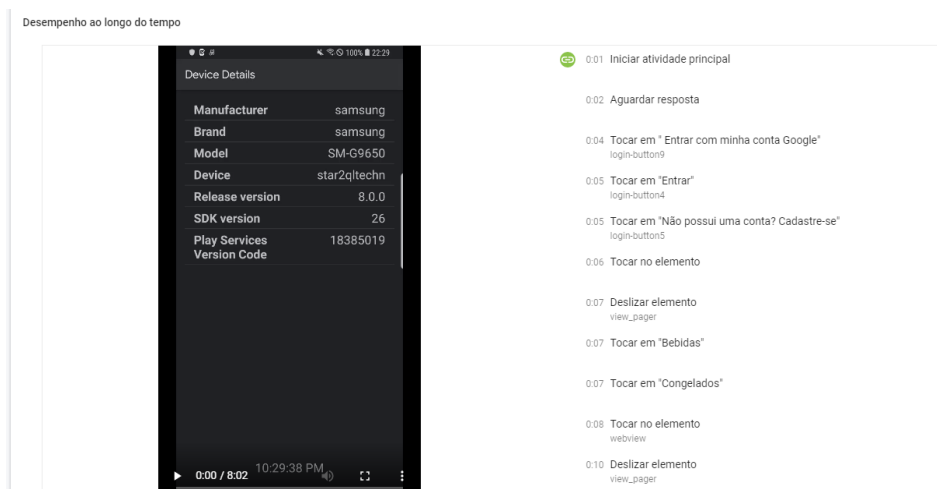
Fonte: Elaborado pelo autor (2019)

Figura 53 – Detalhes do dispositivo do teste 2



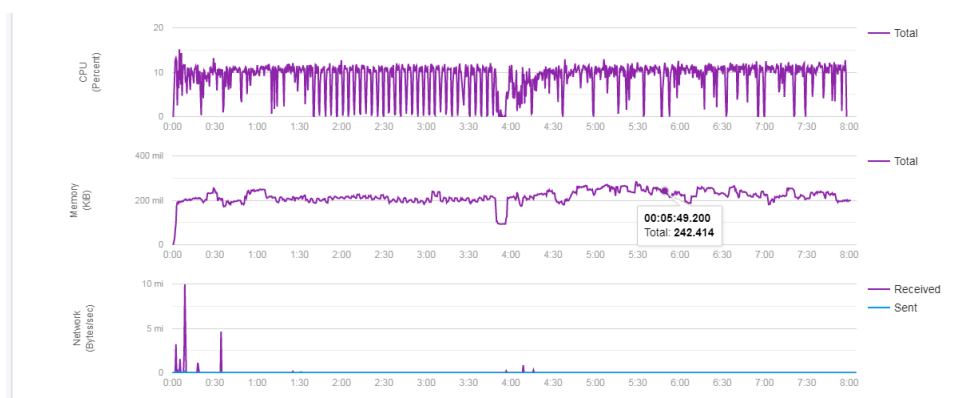
Fonte: Elaborado pelo autor (2019)

Figura 54 – Detalhes do dispositivo do teste 2 parte 2



Fonte: Elaborado pelo autor (2019)

Figura 55 – Desempenho ao longo do tempo teste 2



Fonte: Elaborado pelo autor (2019)

Porém o valor da compra total não era atualizado, isso porque o sistema não gerava uma nova *activity*. Um segundo problema relatado foi quando o usuário tentava mudar o *e-mail* cadastrado em seu perfil, isso porque o *e-mail* era atualizado no banco de dados, porém o Firebase Auth não reconhecia o novo *e-mail*. Este foi um erro que parece ser simples, mas fazia com que o usuário, após encerrar a seção, não conseguisse logar com o *e-mail* novo, o que não deveria ocorrer.

4.2.3 Testes por clientes em potencial

A última etapa de testes se deu pelo contato com os clientes que solicitaram participação na pesquisa de clientes em potencial, que foi um dos primeiros passos do trabalho. A grande maioria não respondeu o *e-mail* que foi enviado com as instruções para participar de fato dos testes. Dos usuários que responderam, o *feedback* foi positivo e eles aprovaram o aplicativo. Acreditamos que, por ainda não realizarem as compras de fato

dentro do sistema, talvez eles não tenham analisado de forma crítica o mesmo. De toda forma, podemos concluir que, após a aprovação por parte desses usuários, o sistema está pronto para ser lançado.

5 Considerações Finais

Este trabalho apresentou o desenvolvimento de um aplicativo de compras de supermercado multiplataforma que visa diminuir o trabalho exigido para a realização de compras maiores, que normalmente demoram mais tempo e fazem com que as pessoas carreguem peso e gastem tempo indo até o supermercado. A ideia de ser multiplataforma é para que possa ser atingido o maior número de usuários e ao adicionar ferramentas como o Firebase, trazer novas tecnologias para o trabalho e o resultado, que é a junção das duas, foi satisfatório.

Na Região do Vale do Aço, que é a área de estudo desse trabalho, não existe nenhum aplicativo que possui o tipo de serviço que o aplicativo elaborado faz. De forma mais ampla, no Brasil, a venda de itens de alimentação pela Internet ainda é modesta, o que se tem é um aumento no número de soluções para a entrega de alimentos de comida pronta, como o iFood e UberEats.

Através das pesquisas efetuadas dentro do nosso trabalho, concluímos que existe a demanda de fato de uma solução tecnológica para otimizar o tempo das pessoas no supermercado. Essa solução não será uma concorrente do supermercado, muito pelo contrário, esperamos que o investimento por parte de um supermercado em trazer esse tipo de tecnologia para o seu negócio acarretaria em aumento de vendas, ganho de mercado e alto índice de visibilidade na região. Além de todos esses pontos, traria para o supermercado a ampliação do canal de vendas, que poderia ser tanto *online* quanto físico.

Uma outra demanda comprovada pela pesquisa é de que esse aplicativo acertou em ser construído de forma multiplataforma, pois em torno de 20% das pessoas na região usam iPhone e as demais Android. Perder mercado nunca é interessante em qualquer ramo de vendas. Um outro ponto diz respeito à expansão do aplicativo para outras regiões, isso poderia acontecer desde que fosse feita uma pesquisa de validação, onde as vantagens do uso do *software* elaborado neste trabalho possam ser aprovadas ou não pelos clientes da nova região alvo.

Durante o trabalho alguns problemas apareceram e a busca por soluções e aprendizado foi constante. Um problema que ocorreu durante os testes iniciais com o aplicativo se deu com o carregamento dos dados na aplicação. Os produtos, por exemplo, demoravam em torno de 9 minutos para serem exibidos na aplicação. Após algumas tentativas de alterar a maneira como a aplicação comunicava com o Firebase, que foram em vão, gerou-se um novo APK que carrega os dados instantaneamente. O problema ocorria por algum tipo de *bug* relacionado a rota dos dados ou erro na compilação do APK anterior.

Foram apresentados todos os aspectos relacionados ao desenvolvimento do aplicativo

utilizando o *Ionic* e outras ferramentas tecnológicas modernas, sua validação na região escolhida, o porquê da escolha de um sistema multiplataforma e os testes para mostrar que o aplicativo está pronto para o mercado. Foi mostrado também, as funcionalidades do sistema, as barreiras encontradas durante o desenvolvimento e as adaptações necessárias para sua conclusão. Na sua versão atual, o aplicativo cumpre com tudo o que foi proposto como requisito funcional do sistema e assim, cumpre com o objetivo do trabalho.

5.1 Trabalhos Futuros

O sistema apresentado nesse trabalho ainda é embrionário, acreditamos que alguns pontos podem ser melhorados, principalmente no que diz respeito à segurança contra *hackers* o que é importante quando se trata de um negócio que visa o lucro. Um ponto de limitação que aconteceu durante o desenvolvimento do aplicativo foi que não conseguimos utilizar a inserção dos pedidos em planilhas, no caso, Google Sheets, por se tratar de inserções múltiplas e as mesmas serem pagas. Além disso, alguns recursos não foram implementados devido ao não acesso à algumas ferramentas pagas, principalmente o banco de dados escolhidos. Alguns requisitos, que ajudariam os usuários ou facilitariam o aplicativo à ganhar mercado, poderiam ser tratados:

- Desenvolver o módulo de pagamentos do aplicativo, de forma confiável e usual.
- Incorporar ao trabalho um sistema de indicações de amigos, como visto dentro da pesquisa, ter amigos que indicam o uso de um aplicativo é fator decisivo para que uma pessoa baixe um programa.
- O cliente pode avaliar a entrega, bem como escolher o dia e a hora da entrega do mesmo.
- Desenvolver um sistema de rotas de entrega para o funcionário utilizando técnicas de Inteligência Artificial, onde a rota será sempre a que tenha um menor custo de financeiro de entrega.
- Desenvolver um sistema de Sugestões mais complexo utilizando também técnicas de Inteligência Artificial, onde a sugestão de produto se dá por produtos que tenham mais chances de que o usuário efetue a compra e não simplesmente um produto da categoria que ele mais fez compras.

Referências

- COSTA, M. F. D.; VILARINDO, J. C. N.; COSTA, T. G. M. D. Determinantes da decisão de compra do consumidor no setor supermercadista. 2007. Citado na página 17.
- DINIZ, E. H. Comércio eletrônico: fazendo negócios por meio da internet. *Revista de Administração Contemporânea*, SciELO Brasil, v. 3, n. 1, p. 71–86, 1999. Citado na página 14.
- FILHO, M. P. de C. Os sistemas de informação e as modernas tendências da tecnologia e dos negócios. *RAE-Revista de Administração de Empresas*, SciELO Brasil, v. 34, n. 6, p. 33–45, 1994. Citado 2 vezes nas páginas 14 e 17.
- FIREBASE. *Firestore helps mobile and web app teams succeed*. 2019. Disponível em: <<https://firebase.google.com/>>. Citado 3 vezes nas páginas 31, 32 e 33.
- GUNAWAN, J. Factors that influence employees' intention to use enterprise social media as knowledge sharing media. *CommIT (Communication and Information Technology) Journal*, v. 13, n. 2, 2019. Citado na página 39.
- HAYDEN, T.; WEBSTER, T. *The mobile commerce revolution: business success in a wireless world*. [S.l.]: Que Publishing, 2014. Citado na página 14.
- IBGE. *CONHEÇA CIDADES E ESTADOS DO BRASIL*. 2019. Disponível em: <<https://cidades.ibge.gov.br/>>. Citado na página 15.
- IONIC. *One codebase. Any platform*. 2019. Disponível em: <<https://ionicframework.com/>>. Citado na página 21.
- LAUDON, K.; LAUDON, J. *Sistemas de Informação gerenciais*. [S.l.]: Person, 2011. Citado na página 17.
- MARINHO, E. H.; RESENDE, R. F. Native and multiple targeted mobile applications. In: SPRINGER. *International Conference on Computational Science and Its Applications*. [S.l.], 2015. p. 544–558. Citado na página 17.
- MICROSOFT. *Ferramentas do Visual Studio para Xamarin*. 2019. Disponível em: <<https://visualstudio.microsoft.com/pt-br/xamarin/>>. Citado na página 21.
- NETO, A. Introdução a teste de software. *Engenharia de Software Magazine*, v. 1, p. 22, 2007. Citado na página 18.
- PREZOTTO, E. D.; BONIATI, B. B. Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. *Anais do EATI*, ano, v. 4, p. 72–79, 2014. Citado 2 vezes nas páginas 17 e 18.
- SCHMITZ, L. Análise de ferramentas de desenvolvimento multiplataforma para criação de aplicativos móveis. 2016. Citado na página 20.
- SCHWALBE, K. *Information Technology Project Management*. 8. ed. [S.l.]: Cengage Learning, 2015. Citado na página 17.