

UNIVERSIDADE DO VALE DO SAPUCAÍ
BRAYAN VINÍCIUS DOS SANTOS
TÚLIO LIMA SILVA

IWOOF PET SHOPS ONLINE

POUSO ALEGRE - MG.

2018

UNIVERSIDADE DO VALE DO SAPUCAÍ

BRAYAN VINÍCIUS DOS SANTOS

TÚLIO LIMA SILVA

IWOOF PET SHOPS ONLINE

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação da Universidade do Vale do Sapucaí como requisito parcial para obtenção de título de bacharel em Sistemas de Informação.

Orientador: Prof. Rodrigo Luís de Faria.

POUSO ALEGRE – MG.

2018

UNIVERSIDADE DO VALE DO SAPUCAÍ
BRAYAN VINÍCIUS DOS SANTOS
TÚLIO LIMA SILVA

IWOOF PET SHOPS ONLINE

Trabalho de Conclusão de Curso defendido e aprovado em ____/____/2018 pela banca examinadora constituída pelos professores:

Prof. Rodrigo Luís de Faria
Orientador

Prof. Me. Marcio Emilio Cruz Vono Azevedo
Avaliador

Prof. Me. José Luiz da Silva
Avaliador

DEDICATÓRIA

Dedicamos esse trabalho primeiramente a Deus por mais essa conquista em nossas vidas, por nos ter proporcionado a entrada em uma Universidade e a conclusão de um Curso Superior. Dedicamos também aos nossos familiares que nos deram todo apoio possível durante essa jornada; aos professores pela paciência e ensinamentos durante todo o curso; aos nossos colegas de classe que sempre estiveram juntos e ao governo pelo programa de Financiamento Estudantil FIES, sem o auxílio não iríamos conseguir concluir o curso.

AGRADECIMENTOS

Agradecemos, primeiramente, a Deus por mais essa conquista, ao orientador desse trabalho Prof. Rodrigo Luís de Faria pela sua experiência e competência, ao professor Prof. Me José Luiz da Silva aos ensinamentos, experiência, sabedoria e disponibilidade sempre que precisamos, a professora Joelma Pereira de Faria pelo auxílio na parte textual e correções durante toda a documentação.

Importante agradecer também aos ex-colegas de curso Carlos Henrique Motta e Marcelo Borges Pereira, que nos ajudaram muito com seu trabalho de conclusão de curso, nos servindo como base e inspiração para desenvolvermos a parte documental deste trabalho.

De Túlio Lima Silva:

Agradeço, primeiramente, a Deus por mais essa conquista em minha vida, pois significa muito ter uma graduação, agradeço aos meus pais Adalgisa Brandão de Lima e Carlos Humberto da Silva por todo apoio e companheirismo durante todo o curso, agradeço a minha namorada Alessandra de Souza Janerine por sempre estar ao meu lado me dando forças e me apoiando nos momentos mais difíceis. Agradeço também aos meus colegas e amigos durante todo o curso, em especial ao meu amigo Brayan Vinícius dos Santos, meu companheiro de trabalho e amigo, pela contribuição e paciência. Agradeço também aos professores pelo incentivo e conhecimento compartilhado, sempre passando as experiências vividas e problemas vivenciados na prática. Agradeço ao Governo pelo programa de Financiamento Estudantil FIES, sem esse programa não iria conseguir concluir por questões financeiras, enfim agradeço a todos que estiveram comigo nestes quatro anos de curso.

De Brayan Vinícius dos Santos:

Agradeço primeiramente a minha família, principalmente a minha mãe Cristina Aparecida da Rosa, por todo apoio e incentivo ao longo desses quatro anos, agradeço a minha namorada Sabrina de Cássia Gomes por todo apoio, me incentivando ao longo dessa jornada. Agradeço ao Governo pelo programa de Financiamento Estudantil FIES, sem esse programa não iria conseguir concluir por questões financeiras, agradeço aos meus fiadores Cristina Aparecida da Rosa e Aguinaldo Rodrigues da Silva, a minha tia Ana Paula Rodrigues da Silva e ao meu primo Bruno Washington Salles, por depositar sua confiança nessa jornada. Agradeço a todos os colegas e amigos durante a faculdade, mais especialmente ao meu amigo Túlio Lima Silva, pelo companheirismo e amizade durante todo o curso. Agradeço a todos os professores pelos ensinamentos e conhecimentos compartilhados durante esses anos, entre eles: Edy Segura, pelo prazer e paixão no que faz; José Luiz da Silva, pelo conhecimento compartilhado e as críticas construtivas nos momentos corretos e ao Rodrigo Luís de Faria, pela orientação do trabalho de conclusão de curso e aos e-mails com aspecto tecnológico enviados ao grupo da sala diariamente, sempre motivando o pessoal a nunca desistir, sempre mostrando o quanto a tecnologia vem evoluindo durante os anos; enfim, agradeço a todas as pessoas envolvidas durante toda esta jornada acadêmica que fiz durante esses quatro anos de curso.

“As pessoas não sabem o que querem, até mostrarmos a elas”

(Steve Jobs)

SANTOS, Brayan Vinícius; SILVA, Túlio Lima. **IWOOF PET SHOPS ONLINE**: Curso de Sistemas de Informação, Universidade do Vale do Sapucaí, Pouso Alegre, 2018.

RESUMO

Muitas empresas de *pet shops* não possuem um aplicativo para venda de produtos, serviços e localização. O IWOOF foi proposto com o intuito de auxiliar essas empresas a serem encontradas mais facilmente e vender seus produtos e serviços com mais facilidade. Por meio de uma página web, as empresas têm a opção de se cadastrar e gerenciar suas vendas e produtos, como também a criação de perfis para seus funcionários e regras de acesso. Esse portal foi desenvolvido utilizando tecnologias *Angular*, *CSS 3* e um *framework Firebase* que possui tecnologia de autenticação pelo *Gmail*. O aplicativo foi desenvolvido em *Ionic* utilizando *Api's* como *Google Maps*. Por meio dele, os usuários têm a opção de encontrar *pet shops* mais próximos e disponibilizar as várias opções de pagamento pelos produtos e serviços via aplicativo, o qual integra-se com a página web de modo que, ao se cadastrarem, as empresas já possam aparecer diretamente no aplicativo, o que possibilita serem encontradas de maneira mais fácil. Por meio das tecnologias utilizadas podemos contar com uma plataforma web e um aplicativo. Este software tem por finalidade ajudar empresas e clientes a ficarem mais próximos, utilizando um meio de comunicação comum a todos, os resultados demonstram uma aplicação e uma página web no qual facilite a busca por *pet shops* e empresas similares, utilizando tecnologias que estão em alta no mercado.

Palavras chave: Angular. Ionic. FireBase.

SANTOS, Brayan Vinícius; SILVA, Túlio Lima. **IWOOF PET SHOPS ONLINE**: Curso de Sistemas de Informação, Universidade do Vale do Sapucaí, Pouso Alegre, 2018.

ABSTRACT

Many pet store companies do not have an application for sale of products, services and location. The IWOOF was designed to help these companies find themselves more easily and sell their products and services more easily. Through a web page, companies have the option of registering and managing their sales and products, as well as creating profiles for their employees and access rules. This portal was developed using Angular, CSS 3 technologies and a Firebase framework that has Gmail authentication technology. The application was developed in Ionic using APIs as Google Maps. Through it, users have the option of finding pet shops nearby and making available the various payment options for products and services via the application, which integrates with the web page so that, when they register, companies can already appear directly in the application, which makes it easier to find them. Through the technologies used we can count on a web platform and an application. This software aims to help companies and clients to be closer, using a common means of communication, the results demonstrate an application and a web page in which to facilitate the search for pet shops and similar companies, using technologies that are in high in the market.

Keywords: Angular. Ionic. Firebase.

LISTA DE CÓDIGOS

| | |
|---|----|
| Código 1 - Configuração Firebase | 36 |
| Código 2 - Setup @NgModule para o AngularFireModule | 36 |
| Código 3 - Renderização dos componentes | 39 |
| Código 4 - Configuração das rotas | 39 |
| Código 5 - Exemplo de validação dos formulários | 40 |
| Código 6 - Criar usuário | 41 |
| Código 7 - Serviço de criar usuário | 42 |
| Código 8 - Gerar senha e montagem e-mail | 43 |
| Código 9 - Serviço enviar e-mail | 44 |
| Código 10 - Menu lateral | 45 |
| Código 11 - Menu inferior | 45 |
| Código 12 - Tela login | 46 |
| Código 13 - Método login | 46 |
| Código 14 - Login service | 47 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Projeção de faturamento 2016. | 16 |
| Figura 2 - Diagrama de casos de uso | 28 |
| Figura 3 - Arquitetura do trabalho | 30 |
| Figura 4 - Instalação do Angular | 31 |
| Figura 5 - Criação do Iwoof-Web | 31 |
| Figura 6 - Página inicial do projeto em Angular | 32 |
| Figura 7 - Página inicial do Ionic | 33 |
| Figura 8 - Estrutura dos diretórios | 34 |
| Figura 9 - Modelagem do banco de dados | 38 |
| Figura 10 – Cadastro da empresa | 48 |
| Figura 11 - Lista de empresas | 49 |
| Figura 12 - Visualização das empresas | 49 |
| Figura 13 - Aprovação das empresas | 50 |
| Figura 14 - E-mail com login e senha | 50 |
| Figura 15 - Primeiro acesso ao sistema | 51 |
| Figura 16 - Tela login Iwoof-Web | 52 |
| Figura 17 - Informações de login e senha | 53 |
| Figura 18 - Validação login e senha | 54 |
| Figura 19 - Tela de perfil do sistema | 55 |
| Figura 20 - Cadastrar perfil de usuário | 56 |
| Figura 21 - Cadastrar usuário | 57 |
| Figura 22 – Inserindo imagens | 57 |
| Figura 23 – Cadastro de produto ou serviço | 58 |
| Figura 24 - Lista de produtos | 59 |

| | |
|---|----|
| Figura 25 - Login Iwoof-Mobile | 60 |
| Figura 26 - Tela de login Iwoof-Mobile | 61 |
| Figura 27 - Cadastro usuário Iwoof-Mobile | 62 |
| Figura 28 - Confirmação conta Gmail | 63 |
| Figura 29 - Mapa aplicativo | 64 |
| Figura 30 - Listar empresas | 65 |
| Figura 31 - Endereço inválido | 66 |
| Figura 32 - Exemplo de produtos | 67 |
| Figura 33 - Detalhes do produto | 68 |
| Figura 34 – Itens no carrinho | 69 |
| Figura 35 - Opções de pagamento | 70 |
| Figura 36 - Gerenciamento de pedidos | 71 |
| Figura 37 - Recusar pedido | 72 |
| Figura 38 - Detalhes do pedido | 72 |

LISTA DE SIGLAS E ABREVIATURAS

| | |
|-------|--|
| API | Application Program Interface |
| WWW | World Wide Web |
| HTML | Hyper Text Markup |
| CERN | Organização Européia para Pesquisa Nuclear |
| CSS | Cascading Style Sheets |
| TCC | Trabalho de Conclusão de Curso |
| NOSQL | Not only Structured Query Language |
| MIT | Massachusetts Institute of Technology |
| SDK | Software Development Kit |
| GPS | Global Positioning System |

SUMÁRIO

| | |
|--|----|
| 1 INTRODUÇÃO..... | 16 |
| 2 QUADRO TEÓRICO..... | 19 |
| 2.1 HTML 5 | 19 |
| 2.2 CSS 3 | 19 |
| 2.3 Angular | 20 |
| 2.4 Ionic Framework..... | 20 |
| 2.5 Firebase..... | 21 |
| 2.6 API Google Maps | 21 |
| 2.7 Marketplace..... | 22 |
| 2.8 Git..... | 23 |
| 2.9 Node.Js | 23 |
| 3 QUADRO METODOLÓGICO..... | 25 |
| 3.1 Tipo de pesquisa..... | 25 |
| 3.2 Contexto da pesquisa | 25 |
| 3.3 Instrumentos..... | 26 |
| 3.4 Procedimentos e resultados | 26 |
| 3.4.1 Coleta de requisitos do sistema | 27 |
| 3.4.2 Elaboração do diagrama de casos de uso | 27 |
| 3.4.3 Seleção das tecnologias | 29 |
| 3.4.4 Configuração do ambiente..... | 30 |
| 3.4.5 Estrutura dos diretórios..... | 33 |
| 3.4.6 Configuração do Firebase e AngularFire..... | 35 |
| 3.4.7 Modelagem do banco de dados | 37 |
| 3.4.8 Desenvolvimento da página web | 39 |
| 3.4.9 Desenvolvimento do aplicativo..... | 44 |
| 4 DISCUSSÃO DE RESULTADOS..... | 48 |

| | |
|---|-----------|
| 4.1 Possibilitar as empresas se associarem por meio de uma solicitação eletrônica. | 48 |
| 4.2 Possibilitar a localização de pet shops e empresas relacionadas através do aplicativo..... | 63 |
| 4.3 Possibilitar as empresas gerenciar pedidos através do website | 71 |
| 5 CONCLUSÃO | 74 |
| REFERÊNCIAS | 75 |

1 INTRODUÇÃO

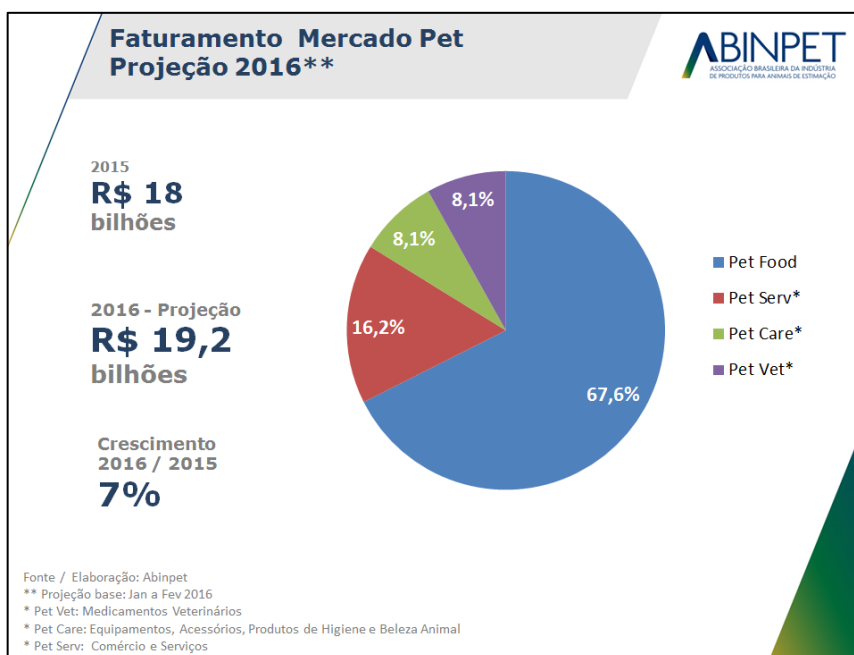
Muitas pessoas tem um animal de estimação e pensam nele como um membro da família. Assim como os seres humanos, os *pets* precisam de cuidados com sua higiene e bem-estar, razão pela qual este trabalho propõe o desenvolvimento de um aplicativo que possa auxiliar pessoas nesse nicho de mercado.

O *marketplace* é um ambiente virtual onde vários vendedores e compradores se unem a um canal único como: eBay ou MercadoLivre. Ele reúne muitas lojas, de diferentes donos, que podem, inclusive, vender produtos iguais a preços diferentes. O aplicativo proporcionará aos usuários meios de encontrar esses produtos e serviços mais próximos em sua região.

As empresas de *pet shops* e similares poderão solicitar a participação através de uma página Web. Após a aprovação feita pelos administradores do aplicativo, a interface Web irá possibilitar as empresas cadastrar produtos e serviços além de gerenciar os pedidos realizados pelos clientes.

Segundo a Associação Brasileira da Indústria de Produtos para Animais de Estimação (Abinpet, 2016), o faturamento anual em 2016 chegou a 18,9 bilhões de reais, teve um crescimento de 4,9% em relação ao ano anterior. Apesar, da projeção que foi de 19,2 bilhões de reais com crescimento esperado de 7% ao ano.

Figura 1 - Projeção de faturamento 2016.



Fonte: abinpet.org.br

Dados de 2016 entre os segmentos da indústria para animais de estimação, o crescimento anual continua em alta, *Pet Food* possui 67,3% do faturamento anual em 2016, em segundo lugar *Pet Serv* (como serviços de banho e tosa) com 16,8% do faturamento, em seguida temos *Pet Care* (acessórios e produtos de beleza) com 8,1%, em quarto lugar *Pet Vet* (produtos veterinários) com 7,8% do faturamento anual (ABINPET, 2016).

Vender seus produtos em um *marketplace* significa poder ser encontrado mais facilmente pelo comprador, podendo optar por seus melhores preços e produtos, tendo a liberdade de escolher entre um vendedor e outro.

O objetivo geral, deste trabalho é desenvolver uma aplicação *marketplace* para comercialização de produtos e serviços relacionados a *pet shops*, sendo os objetivos específicos:

- a. Possibilitar as empresas se associarem por meio de uma solicitação eletrônica;
- b. Possibilitar a localização de *pet shops*, clínicas veterinárias, cuidadores de animais de estimação e hotéis especializados próximos na região através do endereço dos dispositivos móveis;
- c. Possibilitar as empresas gerenciar pedidos através do website.

Este trabalho organiza-se em cinco capítulos, os quais se apresentam a seguir. Na Introdução é demonstrado toda a parte introdutória do trabalho. No Quadro Teórico, são discutidas as tecnologias que foram utilizadas e os dados técnicos, entre as principais: HTML5, ANGULAR, IONIC, CSS3. No Quadro Metodológico é descrito como foi feito todo o procedimento até chegar ao resultado final. Na Discussão de Resultados, são detalhados e discutidos os resultados obtidos durante o desenvolvimento do trabalho. Na Conclusão é descrito todo o resultado obtido, demonstrando a conclusão do trabalho.

Com a realização deste trabalho, espera-se contribuir com os alunos que pretendem desenvolver uma aplicação em *marketplace*. O trabalho proporcionará um auxílio no desenvolvimento de uma aplicação utilizando tecnologias que atualmente estão em alta no mercado.

Este trabalho tem por finalidade contribuir com a sociedade por meio de um aplicativo que possibilitará: fazer seus pedidos e escolher as opções de pagamento por meio do aplicativo, suprir a escassez de tempo dos dias atuais. Espera-se contribuir com empresas de *pet shops* e similares que pretendem divulgar seus produtos e serviços em um ambiente virtual.

Espera-se que esse trabalho seja referência não só para universidades, mas que seja uma ferramenta de auxílio aos estudantes que desejam ingressar na área de tecnologia da informação.

2 QUADRO TEÓRICO

Neste capítulo serão discutidas as tecnologias utilizadas durante o desenvolvimento do trabalho.

2.1 HTML 5

Em setembro de 1991, foi criada a lista de discussão eletrônica denominada *www-talk*, o propósito era trocar ideias e experiências sobre o HTML.

O HTML é a sigla em inglês que significa *Hyper Text Markup Language*¹ e denomina-se na língua portuguesa Linguagem para Marcação de Hipertexto.

O hipertexto pode se resumir em todo conteúdo inserido em um documento web e que pode ser interligado a outros documentos da web, os *links*² são os principais responsáveis para a construção de hipertextos que vemos na web atualmente (SILVA, 2014).

Tim Berners-Lee trabalhava na seção de Computação da Organização Européia de Pesquisa Nuclear (CERN), com sede em Genebra, na Suíça, e visava descobrir um método que fizesse com que cientistas do mundo todo conseguissem compartilhar eletronicamente seus textos e isso tivesse a funcionalidade de interligar um documento ao outro. Ele também acreditava que seria possível a interligação de hipertextos com a utilização de links globais, chamados de hiperlinks (SILVA, 2014).

O HTML5 marca a informação dando valor semântico a estes objetos, quando marcamos um título como h1, h2 ou h3 isso significa um elemento que representa um título em um site web e assim por diante como vários outros elementos (EIS e FERREIRA, 2012).

2.2 CSS 3

CSS é a abreviação do termo em inglês (*Cascading Style Sheet*), traduzindo isso para a língua portuguesa teríamos a expressão Folhas de Estilo em Cascata, ou seja, estilizar páginas web (SILVA, 2011).

O HTML foi criado para ser uma linguagem de estruturação e marcação de conteúdo. Segundo seus idealizadores, não cabe ao HTML fornecer informações como apresentação de elementos como cores de fonte, tamanho dos textos, posicionamento e o aspecto visual, o responsável por toda essa estilização é o CSS (SILVA, 2011).

¹ Linguagem de Marcação de Hipertexto

² Elemento de hipermídia

Definição dos componentes de uma regra CSS:

- **Seletor:** elementos de marcação do HTML onde será aplicada a regra CSS.
- **Declaração:** parâmetros de estilização, propriedades e valores.
- **Propriedade:** características do seletor.
- **Valor:** quantificações ou qualificações do seletor a ser estilizado.

Regra CSS é a unidade básica de uma folha de estilo, ela significa a menor porção de código capaz de produzir um efeito de estilização. Ela é composta por duas partes: seletor e a declaração (SILVA, 2015).

2.3 Angular

De acordo com o portal Angular (s.d.), o próprio consiste em ser uma plataforma e estrutura para a criação de aplicativos clientes desenvolvidos em HTML e Typescript, sendo escrito em Typescript que pode ser importado para os aplicativos atuais.

O Angular possui blocos de construção básicos que podem ser chamados de *NgModules*³, ele consiste em coletar códigos relacionados em seus conjuntos funcionais.

De acordo com o portal Angular (s.d.), o aplicativo possui pelo menos um módulo raiz podendo ser chamado de AppModule, o mesmo pode oferecer mecanismos de inicialização, o aplicativo pode possuir vários outros módulos funcionais.

Os módulos funcionais ajudam a gerenciar e organizar o desenvolvimento de aplicativos mais complexos, e auxiliar a projetar a sua reutilização, eles podem aproveitar o carregamento lento, minimizando assim a quantidade de códigos a ser inicializado (ANGULAR, S.D.).

2.4 Ionic Framework

De acordo com o portal Ionic framework (s.d.), o próprio foi construído pela empresa Drifty, uma empresa de software independente e fabricante de produtos finos como Codiqa e Jetstrap.

³ Blocos de construção de um aplicativo em Angular

O Ionic é um *open source SDK* ⁴ que usa um conceito chamado *native-feeling mobile apps*,⁵ utilizando tecnologias web como, HTML, CSS, JavaScript e TypeScript (GOIS, 2017).

O foco do Ionic é fornecer componentes para o desenvolvimento da interface dos aplicativos, não substituindo o PhoneGap (foca nos recursos do aplicativo), ele fornece uma gama de componentes que podem ser utilizados na *view*, fazendo o controle através do Angular (GOIS, 2017).

O Ionic é lançado sobre uma licença MIT permissiva (licença de programas de computadores), ou seja, você pode utilizar o Ionic em seus projetos pessoais ou comerciais gratuitamente (IONIC, S.D).

2.5 Firebase

Em 2014, o Google concluiu a aquisição de uma empresa sediada em San Francisco chamada Firebase. Fornece uma variedade de soluções para desenvolvedores projetadas para acelerar a integração de recursos baseados em nuvem e aplicativos móveis (SMYTH, 2017).

Depois de comprar a empresa, o Google combinou os serviços fornecidos pelo Firebase com diversos recursos complementares anteriormente incluídos como parte do *Google Cloud Platform*⁶.

Muitos aplicativos e serviços da Web precisam fornecer algum tipo de sistema de autenticação para identificar usuários, controlar o acesso a conteúdo premium e proteger os dados dos usuários, sem alguma forma de identificar um usuário de outro, também seria impossível para o aplicativo saber quais dados e configurações pertencem ao usuário.

As opções de autenticação podem variar desde a exigência de um endereço de e-mail e uma senha para permitir que os usuários façam login usando credenciais de plataformas de terceiros, como Facebook, Google e Twitter (SMYTH, 2017).

2.6 API Google Maps

O Google é uma empresa que, mais do que qualquer outra, molda a forma como usamos computadores hoje. Google Maps é muito mais do que um mapa digital, terceiros adicionam todos os tipos de informações no mapa e novos aplicativos criativos baseados no Google Maps surgem todos os dias (PETROUTSOS, 2014).

⁴ Kit de desenvolvimento Web

⁵ Desenvolvimento de aplicativos móveis

⁶ Suíte de computação em nuvem

Todos os eventos têm duas dimensões básicas: uma dimensão temporal (quando ocorreu) e uma dimensão espacial (onde ocorreu). O Google Maps é a ferramenta que ajuda a visualizar o espaço de um evento.

Com o Google Maps, você pode criar seus próprios mapas personalizados de seu bairro e os lugares que você gosta de visitar, pode ver a Terra de relance, obter uma visão geral de sua vizinhança (PETROUTSOS, 2014).

O Maps oferece mais contexto aos usuários com mapas estáticos ou interativos incorporados no seu site ou aplicativo. Os mapas recebem mais de 25 milhões de atualizações diárias, as informações são atualizadas exatamente quando precisam (CLOUD, S.D).

Com o Google Maps você poderá ser encontrado com mais facilidade pois o mesmo possui um serviço de geolocalização e geocodificação onde esses dados são atualizados rapidamente para que os usuários possam ver informações dos quais precisam (CLOUD, S.D).

A API do Google Maps permite aos usuários ter acesso total ao banco de dados mundial do Google, que tem mais de 100 milhões de listagens de empresas e pontos de interesse. Fornecendo uma lista filtrada dos lugares mais relevantes para eles.

Decidimos utilizar a API do Google Maps, pois utilizamos seu banco de dados com muitos pontos de referência, no qual facilitou a busca por empresas de *pet shops* e similares através de seu sistema de geolocalização e geocodificação.

2.7 Marketplace

Marketplace é um portal de *e-commerce*⁷ colaborativo. Trata-se de um shopping virtual com dois níveis de acesso, um para consumidores e outra para lojistas. O lojista de um segmento de nicho ou não, dependendo da proposta do *Marketplace*, pode vender seus produtos através de um *Marketplace* de terceiros e o cliente, dentro de um só lugar, encontra produtos de diversos fornecedores e marcas (WEBJUMP, S.D).

Decidimos utilizar um ambiente *Marketplace* por que além de englobar várias empresas em um canal único, também proporcionam aos clientes mais comodidade no momento de sua compra, por meio dele o cliente poderá escolher vários produtos com vários preços e compará-los, afim de obter uma compra mais justa.

⁷ Comércio eletrônico

Para o cliente que está comprando é como se estivesse utilizando uma loja virtual convencional, porém o cliente terá acesso a centenas de outras lojas em um único serviço. Depois da compra ser aprovada o sistema divide o pagamento entre administrador, lojista e integrador de pagamento, permitindo transparência entre as partes.

2.8 Git

Git é o sistema de controle de versão de arquivos mais utilizado no mundo. Atráves dele podemos desenvolver projetos no qual várias pessoas possam contribuir com o projeto, editando e criando novos arquivos.

O *Bitbucket* é um serviço para hospedagem de projetos através do Mercurial, um sistema de controle de versões distribuído. É semelhante ao GitHub, onde possui um serviço grátis e outro comercial. É possível criar repositórios privados e ilimitados além de ser gratuito.

Construído para equipes profissionais, é mais do que apenas gerenciamento de códigos do Git. Ele proporciona às equipes um lugar para planejar projetos, colaborar em código, testar e implantar (BITBUCKET, 2018).

Uma das vantagens do que podemos citar é o uso de *commits*⁸, que são pedaços de códigos onde podemos colocar um título ou uma breve descrição, ele permite uma visão mais detalhada de todo o código, além de facilitar a revisão e detecção de falhas.

Decidimos utilizar o *Bitbucket* para controlar as versões do software de modo que possa facilitar o desenvolvimento do trabalho, a fim de obter mais produtividade e controle dos códigos.

2.9 Node.js

No final de 2009, Ryan Dahl criou o Node.js, esta tecnologia possui um modelo inovador, sua arquitetura é totalmente *non-blocking-thread*⁹, além de trabalhar apenas em *single-thread*¹⁰.

O Node.js é uma plataforma altamente escalável e de baixo nível, pois você vai programar diretamente com diversos protocolos de rede e internet, e também utiliza bibliotecas que acessam recursos do sistema operacional (PEREIRA, 2014).

⁸ Ato de enviar

⁹ Não bloqueante

¹⁰ Único processo

Segundo Pereira (2016), o Node.js é orientado a eventos, ele segue a mesma filosofia de orientação de eventos do JavaScript *client-side* ¹¹ a única diferença que podemos encontrar são os tipos de eventos, não existem eventos de click do mouse ou *keyup* do teclado, ele trabalha com eventos de I/O (entrada e saída) como exemplo: o evento de *connect* de um banco de dados, um *open* de um arquivo e muitos outros eventos.

O Node.js foi utilizado para criar a API REST onde essa disponibiliza um serviço de cadastro de usuário e envio de e-mail. Juntamente com o Node.js utilizamos um framework chamado express.js essa biblioteca é projetada para construir aplicativos da Web e APIs.

¹¹ Aplicação executada do lado do cliente

3 QUADRO METODOLÓGICO

No quadro metodológico são descritos os métodos para a realização deste trabalho.

3.1 Tipo de pesquisa

A pesquisa científica tem um papel fundamental na sociedade e no desenvolvimento, principalmente na área industrial e tecnológica, pois por meio de métodos e procedimentos, possibilitam encontrar soluções para problemas existentes na sociedade atual (PARANHOS e RODOLPHO, 2014).

Segundo Gil (2002, p. 17):

Pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. A pesquisa é requerida quando não se dispõe de informação suficiente para responder ao problema, ou então quando a informação disponível se encontra em tal estado de desordem que não possa ser adequadamente relacionada ao problema.

Para o desenvolvimento deste trabalho foi adotada a pesquisa aplicada, utilizamos a pesquisa aplicada por que a mesma consiste em desenvolver uma aplicação prática no qual poderemos utilizar para resolver problemas emergentes no mercado. Segundo Paranhos e Rodolpho (2014), a pesquisa de natureza aplicada ou tecnológica tem como objetivo gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos.

3.2 Contexto da pesquisa

Este trabalho foi desenvolvido com base em um mercado emergente, que visa a criação de um *marketplace* para atender as necessidades dos clientes de *pet shops*, no qual as empresas poderão se beneficiar podendo vender mais e divulgarem seus produtos e serviços com mais facilidade, os clientes poderão ser atendidos por meio do aplicativo.

O objetivo é facilitar a procura de *pet shops* próximos à região, possibilitando a consulta de produtos e serviços. Além disso, suprir as necessidades das empresas, permitindo se cadastrarem, gerenciarem e publicarem seus produtos no mundo virtual.

Espera-se que, com este trabalho, seja possível divulgar novos *pet shops* na região do cliente alvo desta aplicação, para que a distância entre clientes e empresas seja a menor possível. A principal motivação para este trabalho foi o desenvolvimento de uma interface

Web e um aplicativo *mobile*, que possibilite uma solução rápida de tal forma que supra a escassez de tempo dos clientes de *pet shops* nos dias atuais.

3.3 Instrumentos

O instrumento utilizado foram as reuniões, este instrumento foi escolhido por ajudar na organização e desenvolvimento da documentação.

No começo do trabalho foram abordados, em algumas reuniões entre os desenvolvedores, o tema do trabalho, os requisitos do sistema e quais seriam as tecnologias utilizadas para a sua realização. Para a realização deste trabalho, optou-se por reuniões semanais juntamente com o orientador, para acompanhamento e supervisão.

Essas reuniões até o momento estão sendo realizadas na própria universidade, tendo como principal objetivo a verificação da documentação e do desenvolvimento do trabalho, a fim de se obter resultados sempre atentos as entregas e prazos estipulados.

As reuniões atenderam aos objetivos esperados além de ser um excelente instrumento.

3.4 Procedimentos e resultados

A seguir estão relacionados os procedimentos que foram realizados para desenvolvimento do trabalho.

- Coleta de requisitos do sistema;
- Elaboração do diagrama de Casos de Uso;
- Seleção das tecnologias;
- Configuração do ambiente;
- Estrutura dos diretórios;
- Configuração do Firebase e AngularFire;
- Modelagem do banco de dados;
- Desenvolvimento da página web;
- Desenvolvimento do aplicativo.

Estes procedimentos estão descritos de forma detalhada na seção seguinte, a partir de quando pode-se ter uma visão ampla e completa de como se deu a realização, desenvolvimento e construção do produto proposto por esta pesquisa.

3.4.1 Coleta de requisitos do sistema

Para o desenvolvimento desta pesquisa foram levantados alguns requisitos que o sistema deveria ter antes de começar o seu real desenvolvimento. Foram visitados alguns aplicativos para entendermos o que realmente é necessário para desenvolver um sistema completo. Com base no que coletamos podemos montar a lista com os principais requisitos do sistema:

- Cadastro de usuários, empresas, produtos e serviços;
- Definição de níveis de acesso ao usuário;
- Opções de pagamento por meio do aplicativo;
- Gerenciamento de pedidos por meio da página web;
- Localização das empresas por meio do aplicativo.

Visitamos alguns *Pet Shops* e vimos a real necessidade de se ter um aplicativo voltado para a área de *pets*; fizemos o download de um aplicativo já em funcionamento como o DogHero o qual pode-se tirar algumas ideias para melhorar e evoluir o desenvolvimento.

3.4.2 Elaboração do diagrama de casos de uso

Representamos aqui o diagrama de casos de uso afim de demonstrar a funcionalidade básica do sistema. O diagrama de casos de uso faz parte da linguagem de modelagem UML, em inglês *Unified Modeling Language*¹² ela é utilizada em Engenharia de Software afim de representar um sistema de forma padronizada, facilitando a compreensão do sistema.

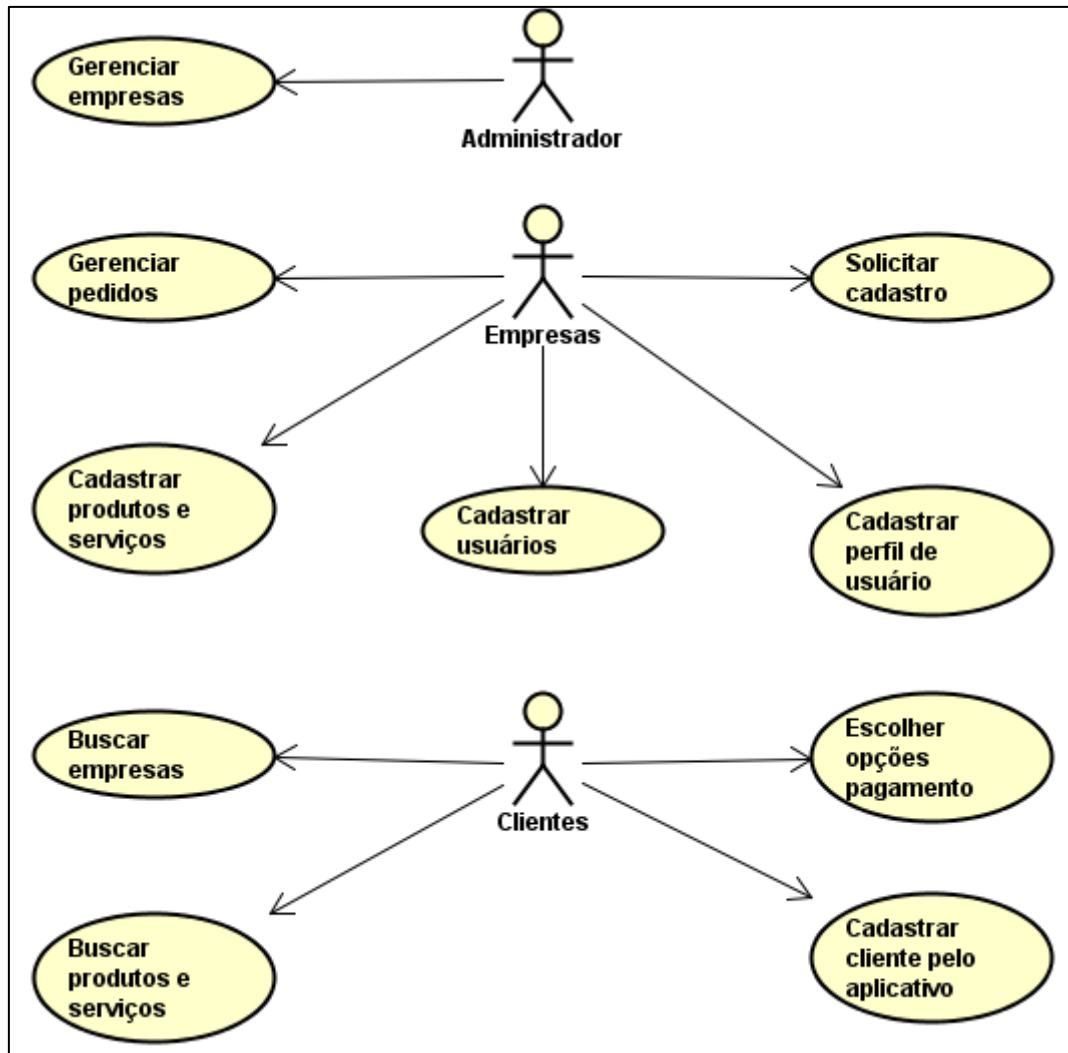
A UML é uma linguagem que serve para especificar, construir, visualizar e documentar os artefatos de um sistema de software (RAMOS, 2006).

Cada caso de uso deve ser documentado e detalhado. Se o detalhamento do caso de uso não for suficiente para a compreensão do seu funcionamento, o risco de uma implementação incorreta será grande. No momento do desenvolvimento, o pessoal terá que inferir os requisitos ou redescobri-los (MARTINS, 2010).

¹² Linguagem de modelagem unificada

Em seguida são representadas as funcionalidades básicas do sistema observada através de um diagrama de casos de uso, conforme mostra a Figura 2:

Figura 2 - Diagrama de casos de uso



Fonte: Elaborado pelos autores

Em relação a Figura 2, demonstraremos cada funcionalidade dos casos de uso em seguida:

Administrador: terá a opção de gerenciar as empresas, fazendo aprovação das empresas, recusando as solicitações das empresas e excluindo as empresas.

Empresas: terão a opção de solicitar o cadastro no web site, cadastrar seus produtos e serviços, cadastrar usuários e perfil dos usuários além de gerenciar seus pedidos.

Clientes: terão a opção de se cadastrar no aplicativo, buscar empresas de *pet shops*, produtos e serviços próximos de seu município, além de escolher as opções de pagamento na entrega do produto.

3.4.3 Seleção das tecnologias

Foi realizada uma pesquisa para selecionar as melhores tecnologias a serem utilizadas no desenvolvimento deste trabalho. Optamos por tecnologias que estão em alta no mercado, utilizamos tecnologias que são tendência atualmente, escolhemos essas tecnologias tanto para auxiliar no desenvolvimento quanto para aquisição de experiências inovadoras.

Decidiu-se que este trabalho seria desenvolvido em IONIC que engloba três tecnologias sendo: HTML 5, CSS 3 e TYPESCRIPT. O Ionic além de ser uma aplicação que funciona em vários sistemas também possui tecnologia híbrida para o desenvolvimento de múltiplas plataformas. O Angular faz toda a estruturação do trabalho, tornando-o modular, além de tornar o sistema modular o que facilita muito o desenvolvimento. Auxilia também pois possui os módulos funcionais que ajudam na inicialização do sistema, no gerenciamento e organização do código.

Utilizamos também um *Baas*¹³ chamado Firebase, o qual fornece um banco de dados *NOSQL*¹⁴, também fornece um esquema dinâmico para dados não estruturados, armazenando dados de várias formas. Fornece uma *API*¹⁵ na qual podemos minimizar o desenvolvimento do *back-end*¹⁶, ele fornece toda estrutura necessária para fazer ações mínimas como um cadastro ou até mesmo ações mais avançadas como controle de permissões, autenticações de login etc.

Para o desenvolvimento do trabalho foi utilizado uma API de mapas chamado Google Maps, com a qual é possível fazermos a localização dos *pets shops* através do GPS dos smartphones e também definirmos qual será o frete a ser pago por meio de seu sistema de rota, em seguida demonstraremos a arquitetura do trabalho, conforme Figura 3.

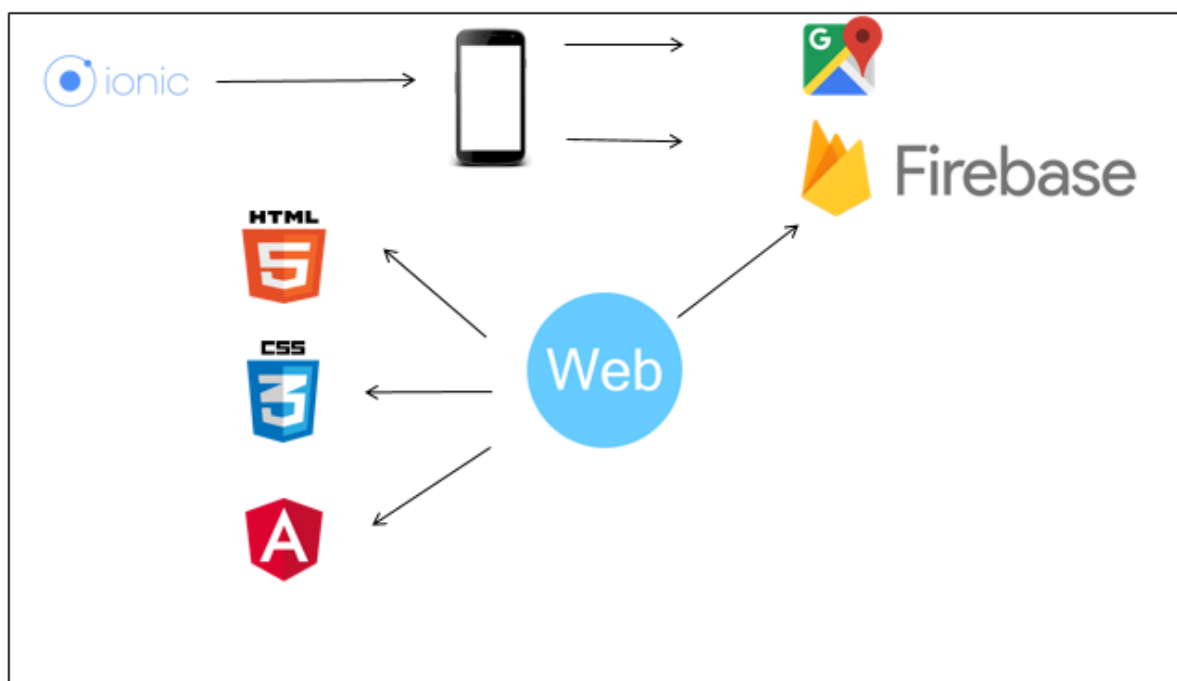
¹³ Back-end como um serviço

¹⁴ Banco de dados não relacional

¹⁵ Interface de programação de aplicações

¹⁶ Implementação na parte de “trás” da aplicação

Figura 3 - Arquitetura do trabalho



Fonte: Elaborado pelos autores

Em relação a Figura 3, foi utilizado a tecnologia IONIC para criação da aplicação mobile, no qual faz-se a comunicação com Api's como Google Maps, foi utilizado a plataforma web e mobile para fazer a comunicação com a plataforma Firebase no qual, auxilia na autenticação e base de dados, foi utilizado a tecnologia Angular para fazer toda a estrutura da página web, além de usarmos a tecnologia HTML 5 para fazermos a base do site e CSS3 para adicionar estilo a página web.

3.4.4 Configuração do ambiente

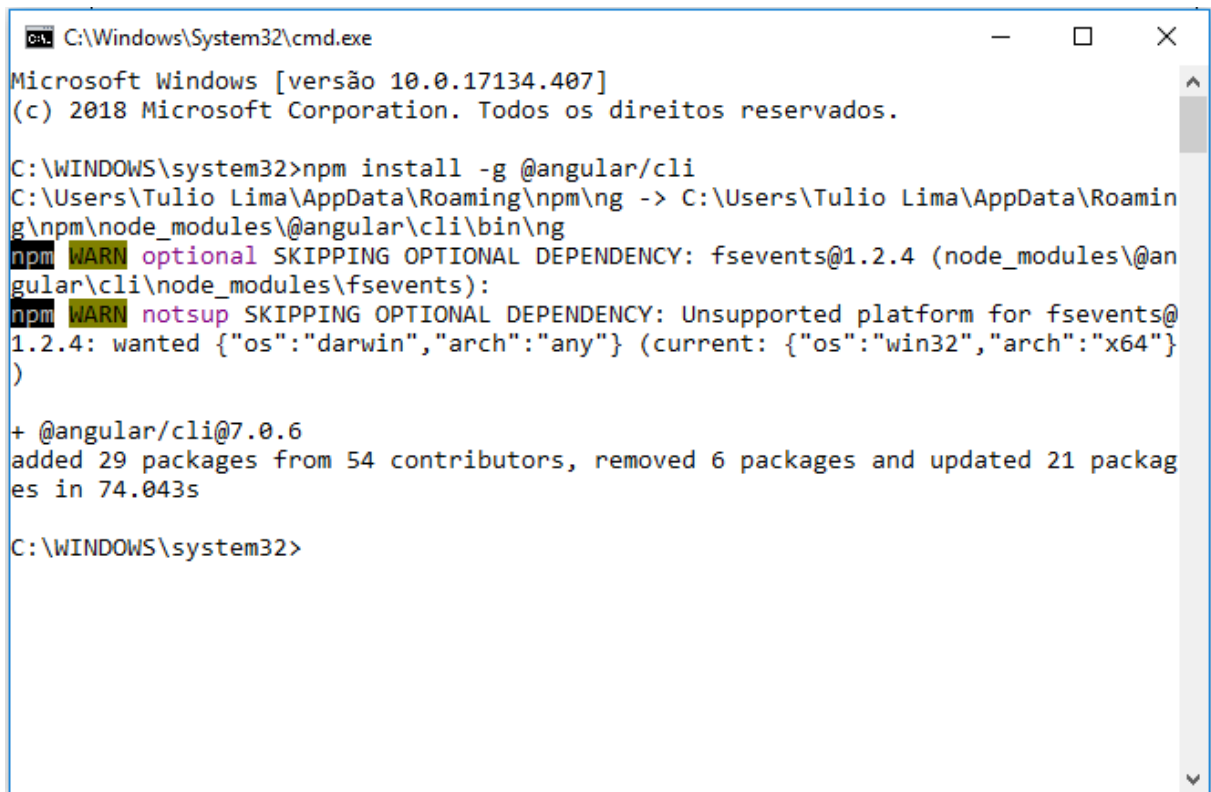
Para o desenvolvimento deste trabalho, foi necessária a instalação de plugins e pacotes responsáveis pelo funcionamento da aplicação. Na sequência serão listadas as ferramentas utilizadas.

3.4.4.1 Criação do projeto em Angular

Nesse ponto é necessário abrir o terminal do sistema operacional e digitar:

npm install -g @angular/cli, conforme apresenta a Figura 4.

Figura 4 - Instalação do Angular



```
C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.17134.407]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>npm install -g @angular/cli
C:\Users\Tulio Lima\AppData\Roaming\npm\ng -> C:\Users\Tulio Lima\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\@angular\cli\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

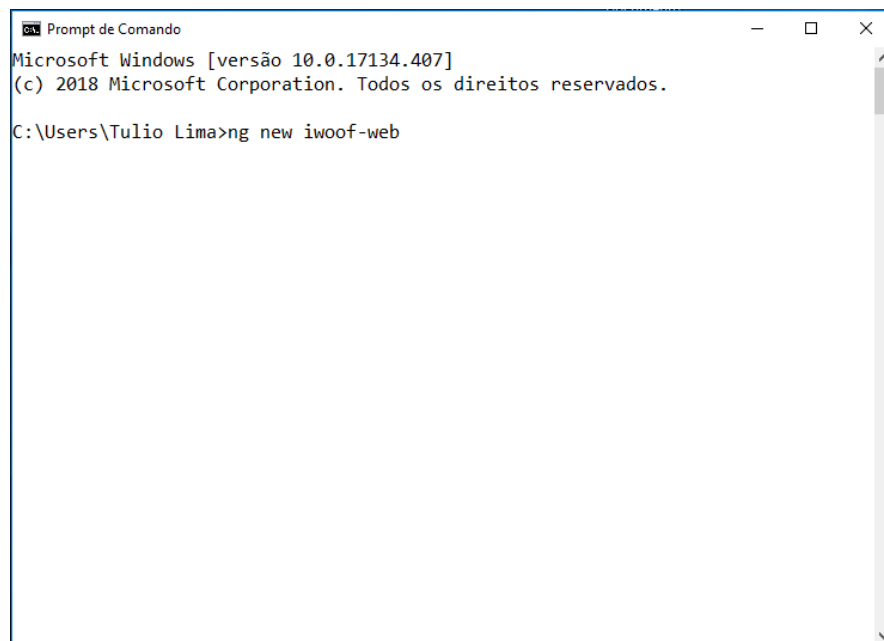
+ @angular/cli@7.0.6
added 29 packages from 54 contributors, removed 6 packages and updated 21 packages in 74.043s

C:\WINDOWS\system32>
```

Fonte: Elaborado pelos autores

Após a instalação do Angular, para a criação do projeto devemos inserir o seguinte comando: **ng new iwoof-web**, conforme mostra a Figura 5.

Figura 5 - Criação do Iwoof-Web



```
Prompt de Comando
Microsoft Windows [versão 10.0.17134.407]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

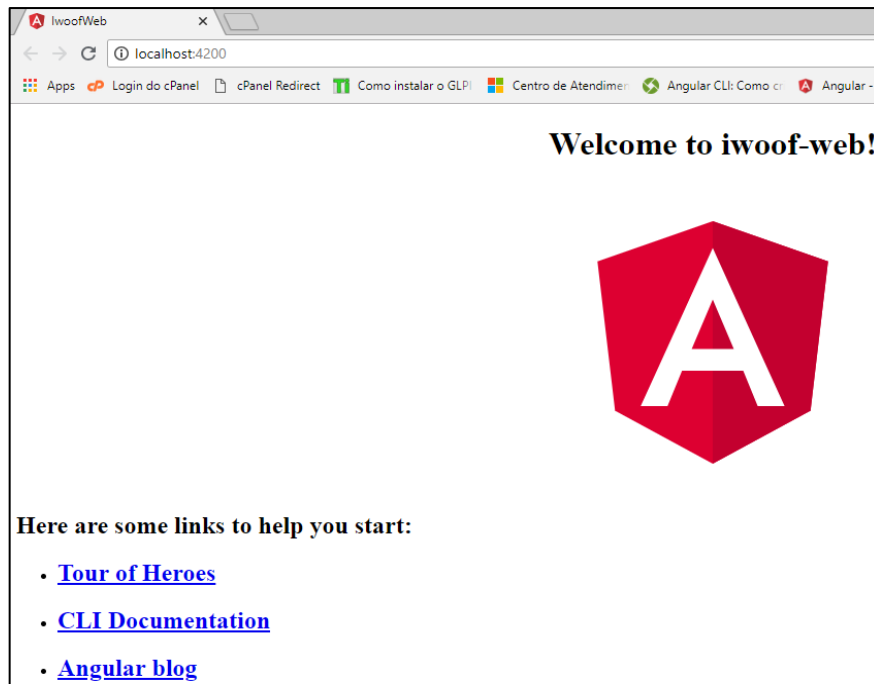
C:\Users\Tulio Lima>ng new iwoof-web
```

Fonte: Elaborado pelos autores

Após a criação do projeto em Angular, para prosseguirmos e mostrar a página web no navegador, devemos inserir os seguintes comandos:

ng serve -o

Figura 6 - Página inicial do projeto em Angular



Fonte: Elaborado pelos autores

No exemplo da Figura 06, podemos observar a página inicial do projeto em Angular, no qual podemos acessar a página web através do endereço <http://localhost:4200/>.

3.4.4.2 Criação do projeto Ionic e Cordova

Primeiramente, para a instalação do cordova, é necessário ter o NodeJs instalado. Feito isso abriremos o terminal do Windows e digitaremos o seguinte comando:

npm install -g cordova

Na sequência, o cordova começará a ser instalado em sua máquina e então, instalaremos o Ionic, para isso devemos abrir o terminal do Windows e digitar o seguinte comando:

npm install -g ionic

Feita a instalação do Ionic e do Cordova, para a criação do projeto em Ionic, devemos inserir o seguinte comando:

ionic start iwoof-mobile sidemenu

Depois de iniciar o projeto, entraremos no diretório e inseriremos o seguinte comando:

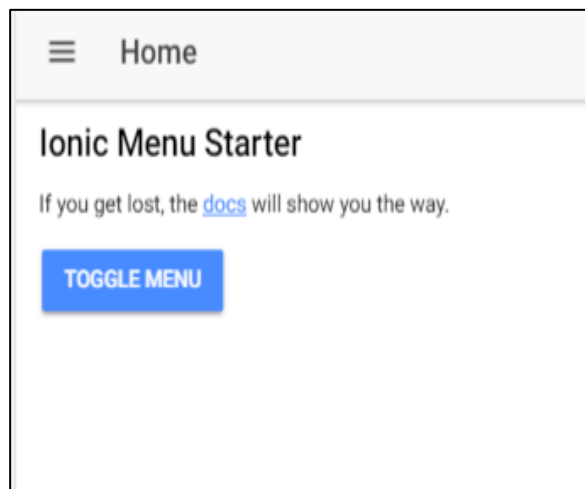
cd iwoof-mobile

Dado esse comando entramos no diretório, então poderemos subir o servidor inserindo o seguinte comando:

ionic serve

Após o comando obtivemos o seguinte resultado, conforme Figura 7.

Figura 7 - Página inicial do Ionic



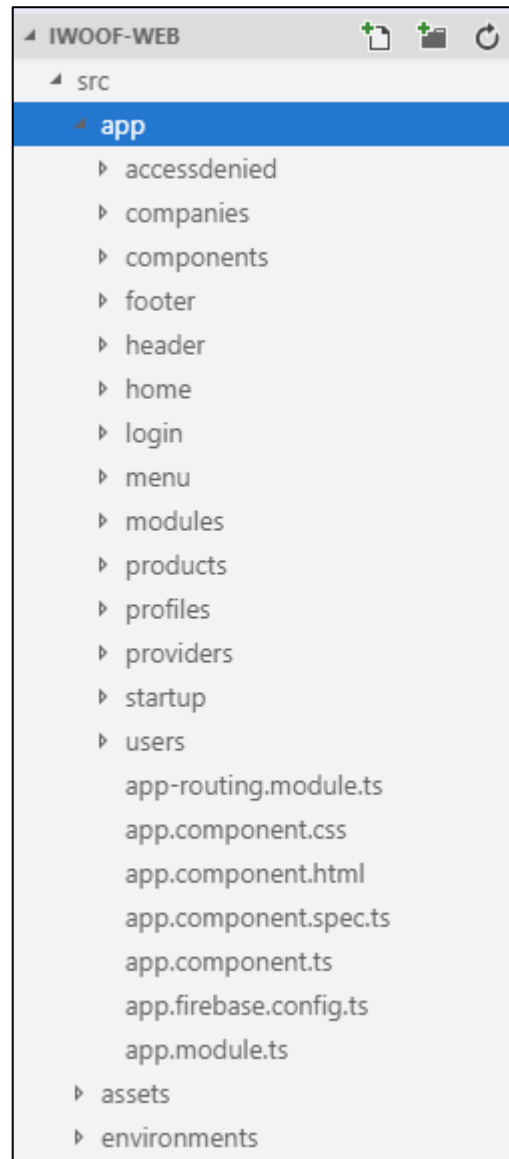
Fonte: Elaborado pelos autores

No exemplo da Figura 7 já é possível acessar a aplicação através do navegador no endereço `http://localhost:8100/`, pois esta é uma aplicação criada automaticamente.

3.4.5 Estrutura dos diretórios

O Angular possui um sistema de diretórios muito completa e modular. As pastas que mais utilizamos durante o desenvolvimento do trabalho foram `src/app`, pois nelas que se encontram todos os arquivos do trabalho, conforme Figura 8.

Figura 8 - Estrutura dos diretórios



Fonte: Elaborado pelos autores

Em relação a Figura 8, foi demonstrado a estrutura que foi adotada para o desenvolvimento deste trabalho. Em seguida mostraremos detalhadamente cada diretório.

3.4.5.1 e2e/

Dentro desse diretório nós temos três arquivos: **app.e2e-spec.ts**, **app.po.ts** e **tsconfig.e2e.json**. Iremos utilizá-los para criar nossos testes unitários.

3.4.5.2 node_modules/

Com esse diretório é possível armazenar as bibliotecas. Quando adicionamos algo no nosso package.json, esse fica responsável por gerenciar os pacotes e suas versões dentro desse diretório.

3.4.5.3 src/app/

Dentro desse diretório se encontram vários arquivos que são responsáveis pelo desenvolvimento do trabalho, conforme os apontados:

- **app.component.css:** Arquivo responsável pelo estilo do nosso módulo. No Angular é possível trabalhar com estilos separados por módulos;
- **app.component.html:** Arquivo responsável pela página HTML da aplicação, onde ficará, por exemplo, o index (página principal) do trabalho;
- **app.component.spec.ts:** Arquivo de teste do nosso componente;
- **app.component.ts:** Arquivo responsável pelo controle da aplicação, pensando em um modelo MVC, seriam as Controllers;
- **app.module.ts:** Arquivo responsável por englobar vários outros módulos dentro dele.

3.4.5.4 src/assets/

Diretório responsável pelos recursos externos do trabalho, como CSS, imagens, bibliotecas JavaScript etc.

3.4.5.5 src/environments/

Diretório responsável pela configuração do ambiente da aplicação tais como: desenvolvimento, produção etc.

3.4.6 Configuração do Firebase e AngularFire

Para a configuração do Firebase precisamos primeiramente instalar o Firebase e o AngularFire por meio do seguinte comando:

npm instalar angularfire2 firebase --save

Feito isso, vamos adicionar a configuração do Firebase à variável de ambientes, para isso devemos abrir o /src/app/app.firebase.config.ts e adicionar a seguinte configuração:

Código 1 - Configuração Firebase

```

1 export const FIREBASE_CONFIG = {
2     apiKey: "your-key",
3     authDomain: "your-project-authdomain>",
4     databaseURL: "your-database-URL",
5     projectId: "your-project-id",
6     storageBucket: "your-storage-bucket",
7     messagingSenderId: "your-messaging-sender-id"
8 }

```

Fonte: Elaborado pelos autores

Continuando com a configuração, injetaremos os provedores do Firebase e especificaremos sua configuração, para isso devemos abrir o `/src/app/app.module.ts` e adicionar a seguinte configuração, conforme mostra o Código 2:

Código 2 - Setup @NgModule para o AngularFireModule

```

1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 @NgModule({
5     declarations: [],
6     imports: [
7         CommonModule,
8         AngularFireModule.enablePersistence(),
9         AngularFireModule.initializeApp(FIREBASE_CONFIG),
10        AngularFireAuthModule,
11        AngularFireStorageModule, ],
12
13     exports: [],
14     providers: [],
15 })
16 export class FeatureModule {}

```

Fonte: Elaborado pelos autores

Feito isso devemos fazer a configuração individual do NgModules, para isso devemos adicionar os módulos que o aplicativo precisa.

- AngularFireModule: serviço de configuração do banco de dados;
- AngularFireModule: configurações da API;

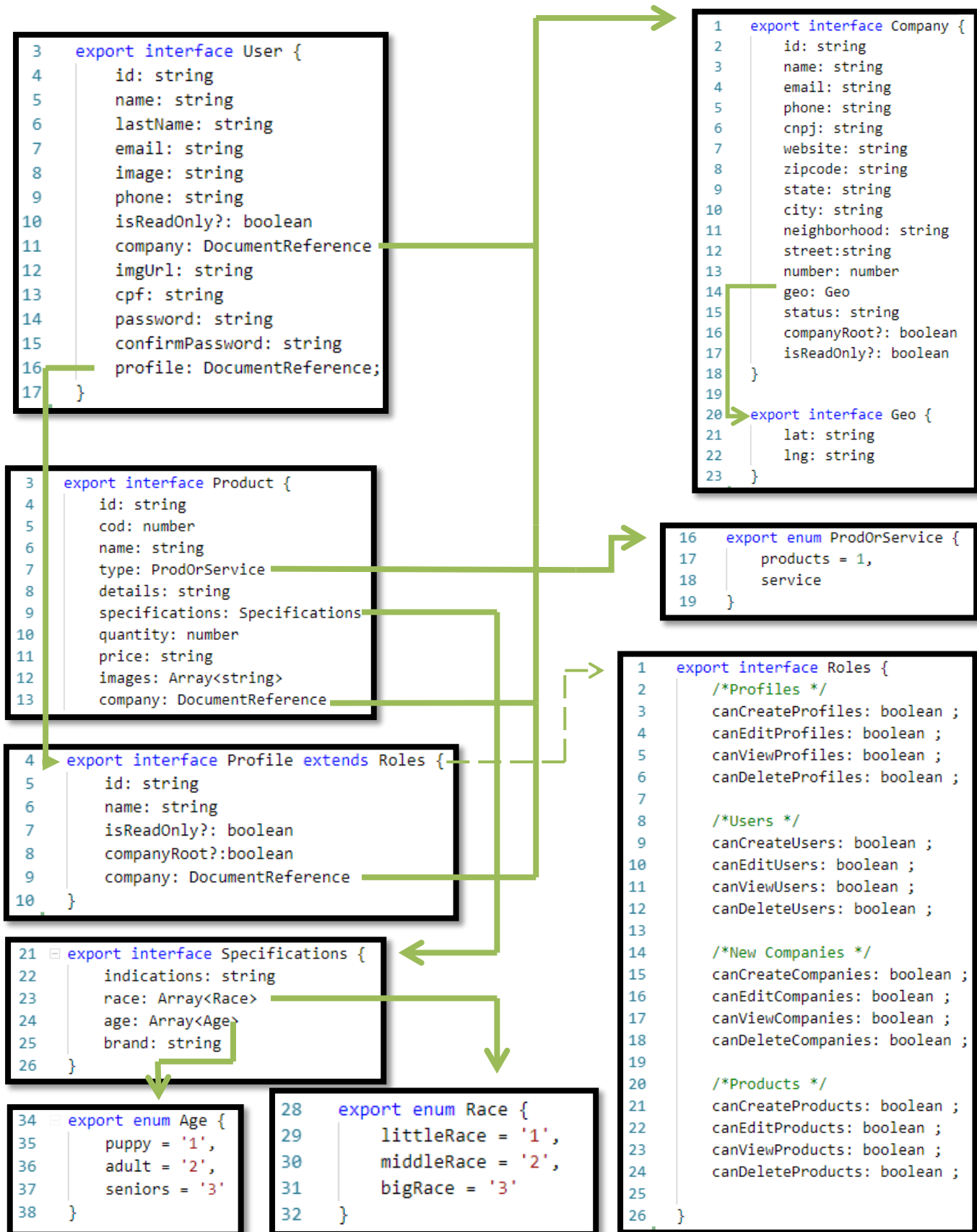
- AngularFireAuthModule: serviço para autenticação do usuário;
- AngularFireStorageModule: serviço para acesso ao Storage do Firebase.

3.4.7 Modelagem do banco de dados

Para o desenvolvimento do trabalho, fez-se necessário a criação de um modelo de banco de dados, que possa explicar as características de funcionamento e comportamento de um sistema, facilitando seu entendimento para evitar assim erros de programação futuramente.

É apresentado a modelagem de dados utilizadas pelos desenvolvedores do trabalho conforme a Figura 9 a seguir.

Figura 9 - Modelagem do banco de dados



Fonte: Elaborado pelos autores

3.4.8 Desenvolvimento da página web

Para o desenvolvimento da página web utilizamos o HTML 5 como base para estruturar a página e utilizamos elementos CSS3 para estilização da página web, juntamente com *bootstrap*, no Código 3 é demonstrado a estrutura da página web.

Código 3 - Renderização dos componentes

```

1 <app-header></app-header>
2 <app-menu></app-menu>
3 <div class="content-wrapper">
4   <router-outlet></router-outlet>
5 </div>

```

Fonte: Elaborado pelos autores

No Código 3 é demonstrado o app-header que compõem todo o cabeçalho do sistema, além de possuir o app-menu, onde faz toda a parte lateral do sistema. O router outlet carrega os componentes da rota conforme Código 4.

Código 4 - Configuração das rotas

```

1 const routes: Routes = [
2   { path: '', redirectTo: 'home', pathMatch: 'full' },
3   {
4     path: 'home', component: HomeComponent, canActivate: [AuthGuard],
5     children: [
6       { path: 'users', component: UsersComponent, canActivate: [AuthGuard], data: {
7         roles: ['canViewUsers'] } },
8       { path: 'profiles', component: ProfilesComponent, canActivate: [AuthGuard],
9         data: { roles: ['canViewProfiles'] } },
10      { path: 'companies', component: CompaniesComponent, canActivate: [AuthGuard],
11        data: { roles: ['canViewCompanies'] } },
12      { path: 'products', component: ProductsComponent, canActivate: [AuthGuard],
13        data: { roles: ['canViewProducts'] } },
14      { path: 'accessdenied', component: AccessdeniedComponent },
15    ]
16  },
17  { path: 'login', component: LoginComponent, canActivate: [AuthGuard] },
18  { path: 'startup', component: StartupComponent, canActivate: [AuthGuard] },
19 ];

```

Fonte: Elaborado pelos autores

Em seguida é descrito os processos e códigos que foram utilizados para o cadastro de um usuário com as devidas permissões.

Para o cadastro dos usuários foi utilizado um formulário, onde o sistema faz toda a parte de validações dos campos, representaremos o campo nome inserido pelo usuário, conforme o Código 5.

Código 5 - Exemplo de validação dos formulários

```

1 <div class="form-group has-feedback has-error col-6">
2   <label for="name"> {{ 'pages.users.Name' | translate }}:</label>
3   <input type="text" id="name" name="name" ngModel #name="ngModel" [(ngModel)]
4     = "user.name" class="form-control padding-imput"
5     placeholder="{{ 'pages.users.type the name' | translate }}" minlength="3"
6     maxlength="30" required>
7   <div *ngIf="name.invalid && (name.dirty || name.touched)">
8     <label *ngIf="name.errors.required" class="control-label">
9       <i class="fa fa-times-circle-o"></i>
10      {{ 'pages.users.Name field is required' | translate }}
11    </label>
12    <label *ngIf="name.errors.minlength" class="control-label">
13      <i class="fa fa-times-circle-o"></i>
14      {{ 'pages.users.Name must be at least 3 characters long' | translate }}
15    </label>
16    <label *ngIf="name.errors.pattern && !name.errors.minlength" class="control-label">
17      <i class="fa fa-times-circle-o"></i>
18      {{ 'pages.users.Name format is invalid' | translate }}
19    </label>
20  </div>
21 </div>

```

Fonte: Elaborado pelos autores

Em relação ao Código 5, a linha 2 faz referência ao nome do campo, como podem observar, o *translate*¹⁷ é um serviço responsável por fazer a tradução de idiomas.

A linha 3 é responsável por persistir o nome do usuário. Entre a linha 5 e 19 é feita a validação no campo nome do usuário, tais validações apresentam a mensagem informando o usuário com uma mensagem de erro como: formato inválido, mínimo de 3 caracteres e campo requerido.

As informações acima representam todos os formulários que compõem o sistema.

¹⁷ Serviço de tradução

Após o usuário preencher o formulário e clicar no botão “cadastrar” o seguinte método é chamado conforme mostra o Código 6 a seguir:

Código 6 - Criar usuário

```
1  createUser() {  
2    this.loading = true;  
3    if (this.user.id) {  
4      this.userService.editUser(this.user).then(data => {  
5        this.loading = false  
6        this.hideModal();  
7      }).catch((err) => {  
8        this.loading = false  
9      })  
10   } else {  
11     this.userService.createUser(this.user).then(data => {  
12       this.loading = false  
13       this.hideModal();  
14     }).catch((err) => {  
15       this.loading = false  
16     })  
17   }  
18 }
```

Fonte: Elaborado pelos autores

Em relação ao Código 6, o método é responsável por verificar se o usuário já existe no banco de dados.

Supondo que o usuário não exista, o método chama o serviço criar usuário, conforme mostra o Código 7.

Código 7 - Serviço de criar usuário

```
1  createUser(user: User) {  
2      let id = this.afs.createId();  
3      user.id = id;  
4      user.company = this.companyRef;  
5      let copUser = { ...user };  
6      return this.prodcollectionUser.doc(id).set(user).then((sucess) => {  
7          this.toasterService.pop('success', this.translateService.instant  
            ('common.Success'), this.translateService.instant('pages.users.User  
            created succeed'));  
8          this.getDocumentReference(this.companyRef.path).first().toPromise().then(  
            (data) => {  
9              this.http.post(this.url, { "name": copUser.name, "lastName":  
                user.lastName, "email": copUser.email, "id": copUser.id, "company":  
                data.name, type: "user" }).pipe(map(response => response.json()))  
                .subscribe();  
10             this.requestEmmitter.emit();  
11         })  
12     }).catch((err) => {  
13         this.requestEmmitter.emit();  
14         this.toasterService.pop('error', this.translateService.instant  
            ('common.Error'), this.translateService.instant('pages.users.Error  
            creating user'));  
15     })  
16 }
```

Fonte: Elaborado pelos autores

Em relação ao Código 7, a linha 6 salva o usuário no banco de dados, e logo em seguida a linha 9 consome a *API* para realizar a autenticação do usuário e enviar o e-mail com usuário e senha.

O serviço a seguir é responsável por gerar a senha e montar o e-mail, conforme mostra o Código 8.

Código 8 - Gerar senha e montagem e-mail

```

1  app.post('/user', (req, res) => {
2
3      var password = generator.generate({ length: 6, numbers: true });
4      let user = { uid: req.body.id, email: req.body.email, password: password,
5                  disabled: false
6      }
7      let htmlEmail = '';
8      if (req.body.type == "user") {
9          htmlEmail = `<div>
10             <h2>${req.body.company}</h2>
11             <h4>Seja bem vindo ${req.body.name} ${req.body.lastName};
12             </h4>
13             <h4>Você é um novo usuário do ${req.body.company}</h4>
14             <h3>Email e Senha para realizar o primeiro login:</h3>
15             <h4>Email: <span>${req.body.email}</span></h4>
16             <h4>Senha: <span>${password}</span></h4>
17             </div>`
18     }
19 }

```

Fonte: Elaborado pelos autores

Em relação ao Código 8, a linha 3 gera um *password*¹⁸ aleatório de no mínimo 6 dígitos, entre a linha 9 e 17 é construído um modelo de e-mail, onde será enviado ao usuário, informando seu login e senha.

O serviço a seguir é responsável por enviar o e-mail ao usuário, conforme mostra o Código 9.

¹⁸ Senha

Código 9 - Serviço enviar e-mail

```
36     const mailOptions = {
37         from: 'iwoofmarketplace@gmail.com',
38         to: req.body.email,
39         subject: 'Iwoof Market Place',
40         html: htmlEmail
41     };
42
43     transporter.sendMail(mailOptions, function (err, info) {
44         if (err)
45             console.log(err)
46         else
47             console.log(info);
48     });
49
50     admin.auth().createUser(user)
51         .then(function (userRecord) {
52             res.status(200).send(userRecord);
53         })
54         .catch(function (error) {
55             res.send(error);
56         });
```

Fonte: Elaborado pelos autores

Em relação ao Código 9, entre as linhas 43 e 48 utilizamos um método para conseguir enviar o e-mail ao usuário, após na linha 50 é feito a autenticação no Firebase, sendo possível o usuário fazer o login através de seu usuário e senha.

3.4.9 Desenvolvimento do aplicativo

Para o desenvolvimento do aplicativo foi utilizado tecnologia IONIC, após a criação do projeto em IONIC, começamos a desenvolver o menu lateral do aplicativo, conforme mostra Código 10.

Código 10 - Menu lateral

```

1 private setMenuPages() {
2     this.pagesList = [
3         { title: 'pages.app.Home', component: 'TabsPage', icon: 'home', iconContent: false },
4         { title: 'pages.app.Message', component: '', icon: 'mail', iconContent: false },
5         { title: 'pages.app.Requests', component: '', icon: 'list', iconContent: false },
6         { title: 'pages.app.My Favorites', component: '', icon: 'star', iconContent: false },
7     ];
8     this.menuList = [
9         { title: 'pages.app.Pet Shops', component: '', icon: 'icon-petshop', iconContent: true },
10        { title: 'pages.app.Caregivers', component: '', icon: 'icon-caregivers', iconContent:
11            true },
12        { title: 'pages.app.Hotel', component: '', icon: 'icon-hotel', iconContent: true },
13        { title: 'pages.app.Clinic', component: '', icon: 'icon-clinics', iconContent: true },
14        { title: 'pages.app.Categories', component: '', icon: 'albums', iconContent: false },
15    ];
16    this.settingsList = [
17        { title: 'pages.app.Help', component: '', icon: 'help-circle', iconContent: false },
18        { title: 'pages.app.Support', component: '', icon: 'call', iconContent: false },
19        { title: 'pages.app.App Settings', component: '', icon: 'settings', iconContent: false }
20    ];
21 }

```

Fonte: Elaborado pelos autores

Em relação ao Código 10, cada *title* indica o nome do título do menu, o componente *icon* significa o ícone que cada componente está utilizando, o componente *component* indica que quando um objeto é tocado ele renderiza a página para qual o componente foi designado.

Foi implementado o menu inferior do aplicativo de modo que fica mais facilitado a utilização do aplicativo, conforme mostra o Código 11.

Código 11 - Menu inferior

```

1 <ion-tabs>
2     <ion-tab [root]="tab1Root" tabTitle="{{ 'pages.tabs.Stores' | translate
3         }}" tabIcon="home"></ion-tab>
4     <ion-tab [root]="tab2Root" tabTitle="{{ 'pages.app.Categories' |
5         translate }}" tabIcon="albums"></ion-tab>
6     <ion-tab [root]="tab3Root" tabTitle="{{ 'pages.app.Requests' |
7         translate }}" tabIcon="list"></ion-tab>
8 </ion-tabs>

```

Fonte: Elaborado pelos autores

Em relação ao Código 11, *root* é o componente que será renderizado quando houver o click do usuário na tela do aplicativo, *tabTitle* é o título do componente que irá aparecer no campo inferior do menu e *tabIcon* é o ícone que o componente irá utilizar.

Para o cadastro e utilização do usuário no aplicativo foi desenvolvido uma tela de login e cadastro de usuários, conforme mostra Código 12.

Código 12 - Tela login

```

1 <ion-item>
2   <ion-label>
3     <ion-icon name="person"></ion-icon>
4   </ion-label>
5   <ion-input placeholder="{{ 'pages.login.type your e-mail' | translate }}" [(ngModel)]
      ="user.email" type="text"></ion-input>
6 </ion-item>
7 <ion-item>
8   <ion-label>
9     <ion-icon name="lock"></ion-icon>
10  </ion-label>
11  <ion-input placeholder="{{ 'pages.login.type your password' | translate }}" [(ngModel)]
      ="user.password" type="password"></ion-input>
12 </ion-item>

```

Fonte: Elaborado pelos autores

Em relação ao Código 12, representamos uma tela de login onde constitui um formulário para preenchimento do e-mail e senha para efetuar o login no sistema, este formato de formulário foi utilizado em todos os formulários do aplicativo.

Após o usuário clicar sobre o botão entrar o seguinte método é acionado conforme mostra Código 13.

Código 13 - Método login

```

1 login() {
2   if(this.validators.login(this.user)){
3     this.loading.showDismissPage('common.loadingAuth');
4     this.loginService.login(this.user).then((res) => {
5       this.navCtrl.setRoot(TabsPage, {}, {animate: true,
6         direction: 'forward'});
7     }).catch(err => {
8       this.loading.hide();
9       this.toast.showResponseError(err);
10    });
11  }
12 }

```

Fonte: Elaborado pelos autores

Em relação ao Código 13, esse método é responsável por chamar o serviço *loginService* no qual é responsável por fazer o login do sistema. A linha 2 torna-se responsável por fazer toda a validação do usuário, a linha 3 responsável por chamar o *loading* do sistema, em seguida demonstraremos como é feito o procedimento de login dentro do sistema, conforme Código 14 abaixo:

Código 14 - Login service

```
1  async login(user: any): Promise<any> {  
2      return this.fireAuth.auth.setPersistence  
      (firebase.auth.Auth.Persistence.LOCAL).then(() => {  
3          return  
          this.fireAuth.auth.signInAndRetrieveDataWithEmailAndPassw  
          ord(user.email, user.password);  
4      })  
5  }
```

Fonte: Elaborado pelos autores

Em relação ao Código 14, a linha 2 é responsável por manter o usuário logado, mesmo se o usuário fechar o aplicativo, já na linha 3 é responsável por fazer a autenticação do usuário e gerar um *token*¹⁹.

¹⁹ Chave eletrônica

4 DISCUSSÃO DE RESULTADOS

Neste capítulo são discutidos os resultados obtidos durante o desenvolvimento do trabalho, cujo objetivo principal consiste em desenvolver uma aplicação marketplace para comercialização de produtos e serviços relacionados a *pet shops*.

Com base nos objetivos específicos são apresentados os resultados obtidos.

4.1 Possibilitar as empresas se associarem por meio de uma solicitação eletrônica.

Para que as empresas tenham acesso ao Iwoof, primeiramente elas precisam estar cadastradas. As empresas deverão se cadastrar através de um formulário conforme Figura 10.

Figura 10 – Cadastro da empresa



O formulário, intitulado "Cadastre seu Pet Shop", possui um fundo verde claro. Ele contém campos de entrada para as seguintes informações:

| Empresa: | E-mail: |
|--|--|
| <input type="text" value="digite o nome"/> | <input type="text" value="digite o email"/> |
| Celular: | CNPJ: |
| <input type="text" value="(00) 0 0000-0000"/> | <input type="text" value="00.000.000/0000-00"/> |
| Site: | CEP: |
| <input type="text" value="digite o campo site"/> | <input type="text" value="digite o CEP"/> |
| UF: | Cidade: |
| <input type="text" value="digite o campo estado"/> | <input type="text" value="digite o campo cidade"/> |
| Rua: | bairro: |
| <input type="text" value="digite o campo rua"/> | <input type="text" value="digite o campo bairro"/> |
| Número: | |
| <input type="text" value="digite o campo número"/> | |

Na base do formulário, há um botão azul arredondado com o texto "Cadastrar-se".

Fonte: Elaborado pelos autores

Após o cadastro, as empresas irão para uma tela central onde os administradores do sistema poderão aprovar sua solicitação ou não, conforme a Figura 11.

Figura 11 - Lista de empresas

Empresas

3 Empresas Aprovadas | 0 Aguardando Aprovação | 0 Empresas Reprovadas

Lista de Empresas

| Estado | Nome | E-mail | CNPJ | Contato | Cidade | |
|--------|--------------------|--------------------------|--------------------|------------------|---------------|--------------------|
| ● | Iwoof Market Place | admin@iwoof.com | 5485548484847 | (35) 9 8428-1354 | Paraisópolis | Visualizar Deletar |
| ● | Fofinhos Pet Shops | v.brayansantos@gmail.com | 55.123.564/3115-89 | (35) 98428-1354 | Pouso Alegre | Visualizar Deletar |
| ● | Clínicas Tulinho | tuuhlimaa@gmail.com | 00.554.548/7895-55 | (35) 9 8458-6555 | Borda da Mata | Visualizar Deletar |

Fonte: Elaborado pelos autores

Os administradores do sistema poderão visualizar as informações das empresas, clicando sobre o ícone “Visualizar”, conforme mostra a Figura 12.

Figura 12 - Visualização das empresas

Empresa

Nome: Iwoof Market Place

CNPJ: 5485548484847

Telefone: (35) 9 8428-1354

E-mail: admin@iwoof.com

Site: iwoof.com.br

Cep: 37660-000

Cidade: Paraisópolis

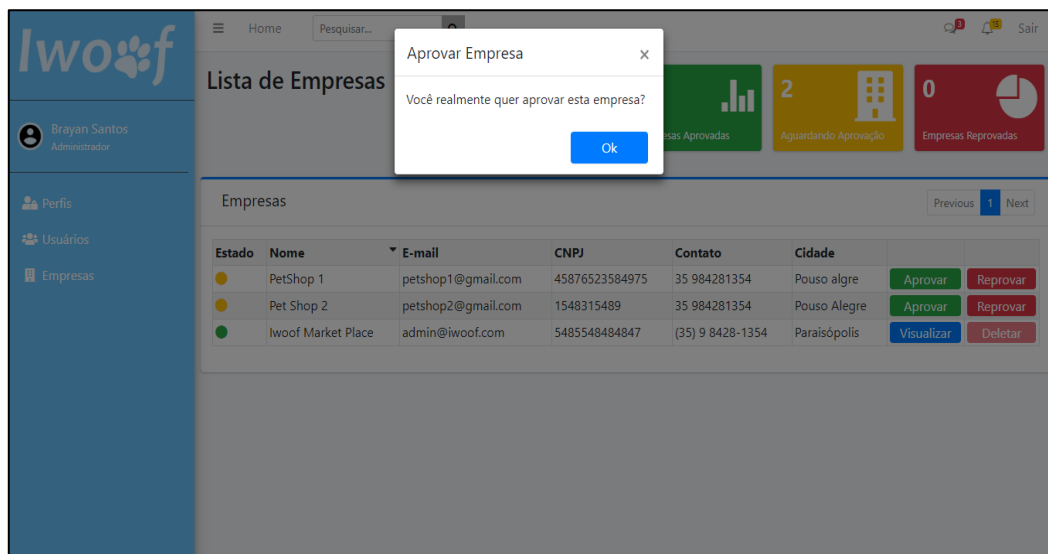
Rua: Rua Diamantina

Número: 26

Fonte: Elaborado pelos autores

Após as empresas estarem listadas, ao clicar sobre o ícone aprovar, haverá uma mensagem de confirmação, conforme mostra a Figura 13.

Figura 13 - Aprovação das empresas



Fonte: Elaborado pelos autores

Após a empresa ser aprovada, o sistema irá enviar uma mensagem para o e-mail cadastrado informando seu login e senha, conforme mostra a Figura 14.

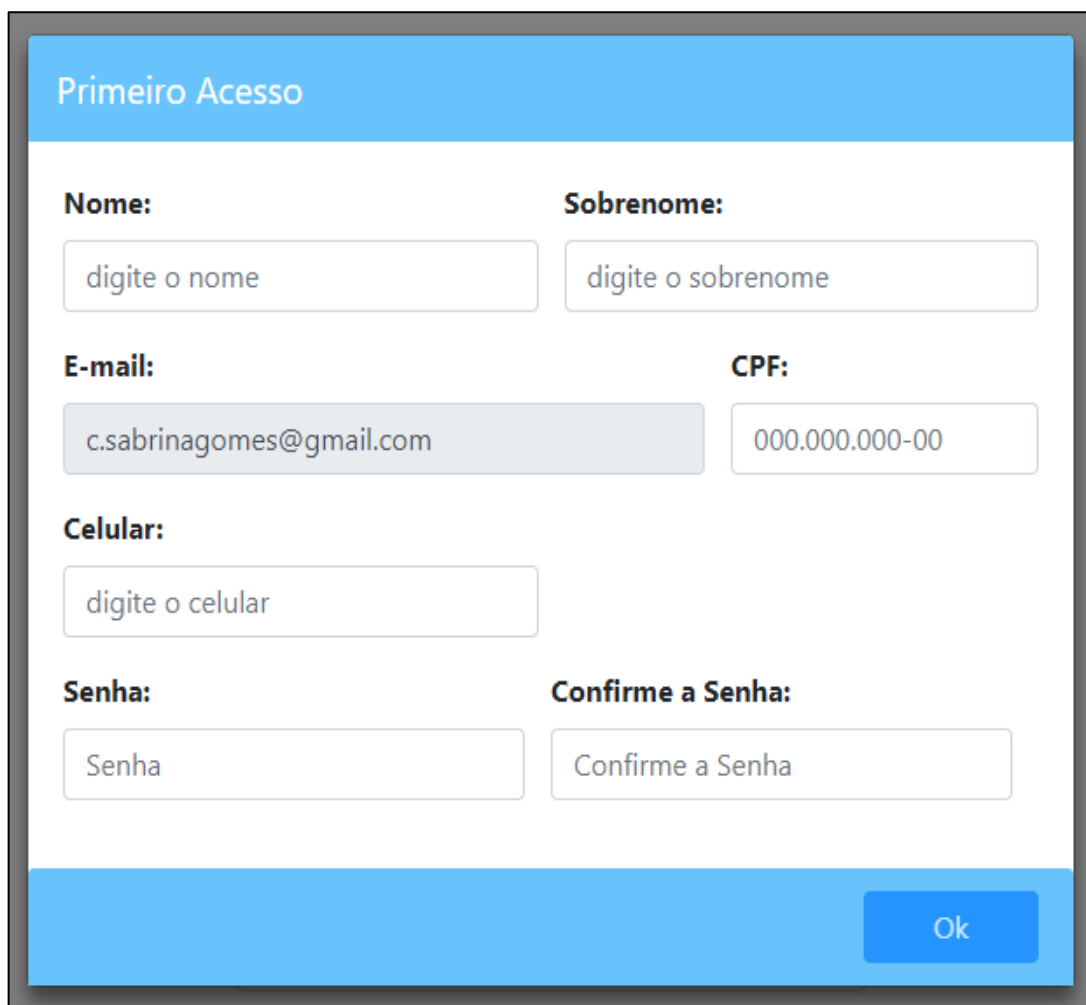
Figura 14 - E-mail com login e senha



Fonte: Elaborado pelos autores

Feito isso, as empresas poderão entrar no sistema informando seu login e senha. No seu primeiro acesso, o sistema irá possibilitar a redefinição de sua senha e suas informações pessoais, conforme mostra a Figura 15.

Figura 15 - Primeiro acesso ao sistema



O formulário, intitulado "Primeiro Acesso", é dividido em seções para coleta de dados pessoais e de contato. As seções são:

- Nome:** Campo de texto com o placeholder "digite o nome".
- Sobrenome:** Campo de texto com o placeholder "digite o sobrenome".
- E-mail:** Campo de texto com o valor "c.sabrinagomes@gmail.com".
- CPF:** Campo de texto com o valor "000.000.000-00".
- Celular:** Campo de texto com o placeholder "digite o celular".
- Senha:** Campo de texto com o placeholder "Senha".
- Confirme a Senha:** Campo de texto com o placeholder "Confirme a Senha".

Um botão azul "Ok" está localizado no canto inferior direito do formulário.

Fonte: Elaborado pelos autores

Em relação a Figura 15, são demonstrados os campos onde os usuários terão a opção de preencherem e informarem seus dados cadastrais.

Para o desenvolvimento do site utilizamos o *admlte* que consiste em um painel administrativo *open source*²⁰ o qual disponibiliza os *layouts*²¹ e componentes baseados em *bootstrap*²² para facilitar o desenvolvimento deste trabalho.

Foi criada uma empresa chamada IWOOF MARKETPLACE manualmente no banco de dados, para comandar as operações iniciais e gerenciar as próximas empresas que irão se cadastrar, com isso foi criado também um perfil administrador que possui todas as permissões necessárias para gerenciar o site. Baseado nisso criamos um usuário *root* (administrador) e

²⁰ Software de código aberto

²¹ O plano sobre o qual será construído um conteúdo na internet

²² Framework web com código-fonte aberto

definimos uma senha padrão para o administrador no qual ele poderá fazer o login, conforme mostra a Figura 16:

Figura 16 - Tela login Iwoof-Web

The image shows a login interface for 'Iwoof-Web'. At the top is the logo 'Iwoof' in a large, black, stylized font, where the 'o' is replaced by a paw print. Below the logo is a light gray rounded rectangle containing the text 'Entre para iniciar sua sessão'. Underneath this text are two input fields: the first is labeled 'Email' and has a dark gray envelope icon on its right; the second is labeled 'Senha' and has a dark gray padlock icon on its right. At the bottom of the rounded rectangle, there is a blue link 'Esqueci a minha senha' and a solid blue button labeled 'Login'.

Fonte: Elaborado pelos autores

Para efetuar o login devemos colocar o usuário e senha, conforme mostra a Figura 17.

Figura 17 - Informações de login e senha



The image shows a login interface for a system named 'Iwoof'. At the top, the logo 'Iwoof' is displayed in a large, black, stylized font, where the 'o' is replaced by a paw print. Below the logo, the text 'Entre para iniciar sua sessão' (Enter to start your session) is centered. Underneath this text are two input fields. The first field contains the email address 'admin@iwoof.com' and has an envelope icon on its right side. The second field contains a series of dots, representing a password, and has a lock icon on its right side. Below the password field, there is a blue link that says 'Esqueci a minha senha' (I forgot my password) and a blue button labeled 'Login'.

Fonte: Elaborado pelos autores

Em relação a Figura 17, como podem observar foi inserido um usuário chamado “admin@iwoof.com” como login e uma senha para fazer o logon no sistema.

Foram implementadas validações em todos os formulários do sistema, conforme mostra a Figura 18.

Figura 18 - Validação login e senha



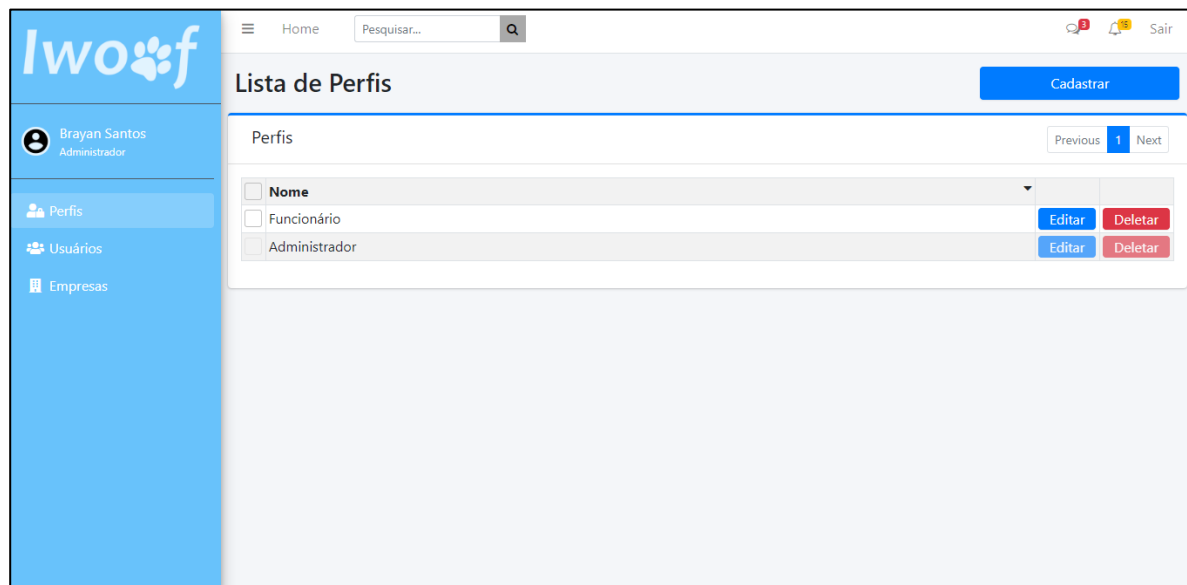
The image shows a login form for a system called "Iwoof". The logo "Iwoof" is at the top, with the "o" replaced by a paw print. Below the logo is a light gray box containing the text "Entre para iniciar sua sessão". There are two input fields: the first is for an email address, containing "admin@iwoo", with a red error message "© Formato do email é inválido" below it; the second is for a password, containing two dots, with a red error message "© A senha deve ter pelo menos 6 caracteres" below it. At the bottom of the box are a blue link "Esqueci a minha senha" and a blue "Login" button.

Fonte: Elaborado pelos autores

Em relação a Figura 18, foi utilizado validações nos campos, como: formato de e-mail inválido, validação de limite de caracteres.

Depois de entrar no sistema aparecerá a tela, com os usuários existentes, perfis e empresas. Para cadastrar os perfis e usuários basta clicar sobre o botão “cadastrar” encontrado no canto superior direito, conforme mostra a Figura 19.

Figura 19 - Tela de perfil do sistema



Fonte: Elaborado pelos autores

Na Figura 19 é demonstrada a tela de perfil do sistema. É importante ressaltar que só irá aparecer a aba “Empresas” para os administradores do sistema no caso os desenvolvedores, pois através dessa tela será possível aprovar ou reprovar as empresas que irão solicitar a participação no aplicativo.

Através dessa tela é possível “Editar e Deletar” usuários, perfis de usuários e empresas.

As empresas poderão criar perfis de acordo com suas necessidades, para isso haverá um formulário de cadastro de perfil, conforme mostra a Figura 20.

Figura 20 - Cadastrar perfil de usuário

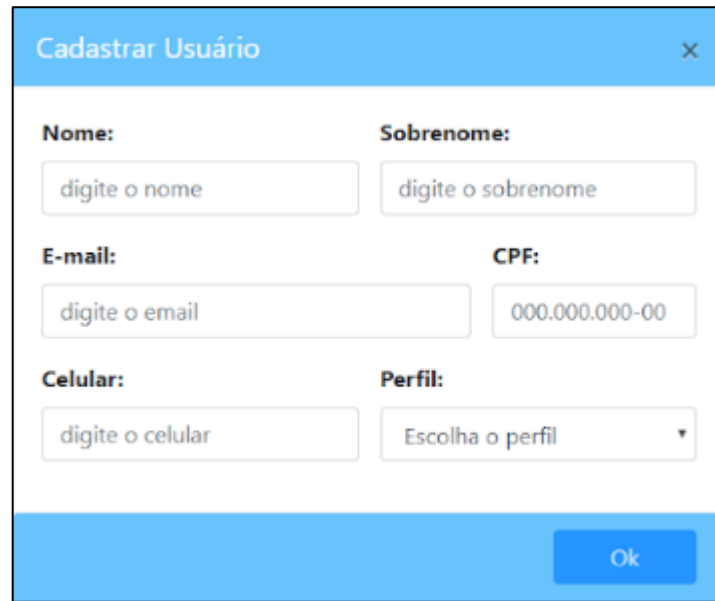
| | Criar | Editar | Visualizar | Deletar |
|-------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Perfis | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Usuários | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Produtos | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Gerenciar Pedidos | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Fonte: Elaborado pelos autores

Na Figura 20 é demonstrado o formulário para o cadastro do perfil dos usuários, com ele é possível estabelecer algumas regras como: criar, editar, visualizar e deletar. Isso torna possível a empresa cadastrar vários níveis de usuários no sistema, de forma a ter um gerenciamento mais amplo e objetivo.

As empresas terão a opção de cadastrar usuários e atribuir perfis a esses usuários. Para o cadastro dos usuários que irão utilizar o sistema web será preciso cadastrá-los, através de um formulário, conforme mostra a Figura 21.

Figura 21 - Cadastrar usuário



Formulário de Cadastro de Usuário com campos para:

- Nome:** digite o nome
- Sobrenome:** digite o sobrenome
- E-mail:** digite o email
- CPF:** 000.000.000-00
- Celular:** digite o celular
- Perfil:** Escolha o perfil

Botão de confirmação: Ok

Fonte: Elaborado pelos autores

Para efetuar este cadastro, a empresa terá que informar várias informações do usuário como: Nome, Sobrenome, E-mail, CPF, Celular e o Perfil que deseja atribuir a este usuário.

Foi desenvolvido uma tela onde empresas terão a opção de cadastrar seus produtos, serviços e colocar as imagens que desejam através de um formulário, conforme a Figura 22:

Figura 22 – Inserindo imagens



Tela de Edição de Produto (Editar Produto) com o seguinte conteúdo:

- Título:** Arraste e solte as fotos do produto
- Subtítulo:** A 1ª Deve ser a principal imagem do produto
- Botão:** Selecione a Imagem
- Imagens:** Três imagens de produtos (um pote de metal, um pote de plástico e três potes de metal).
- Formulário:**
 - Nome:** Comedouro Jambo Rosa Metálico
 - Código:** 8751
- Botão de Confirmação:** Ok

Contexto da tela: Interface de administração do Iwoof, com menu lateral (Perfis, Usuários, Produtos) e painel de controle de produtos no fundo.

Fonte: Elaborado pelos autores

As empresas terão a opção de arrastar e soltar as imagens dos produtos ou clicar sobre o botão “Selecione a Imagem”, feito isso terão a opção de colocar algumas informações como: nome do produto e código do produto, conforme mostra Figura 22.

Após feito a inserção da imagem, as empresas também terão a opção de cadastrar suas informações, conforme mostra a Figura 23.




Figura 23 – Cadastro de produto ou serviço

Fonte: Elaborado pelos autores

Na Figura 23 é possível as empresas cadastrarem seus produtos e serviços, para efetuarem o cadastro as empresas devem preencher alguns campos como: Detalhes, Raça, Idade, Indicação, Marca, Quantidade e o Preço.

Após as empresas cadastrarem seus produtos na página web, terá a opção de visualizar esses produtos, clicando sobre a aba “Produtos” encontrado na tela inicial do sistema, conforme Figura 24.

Figura 24 - Lista de produtos

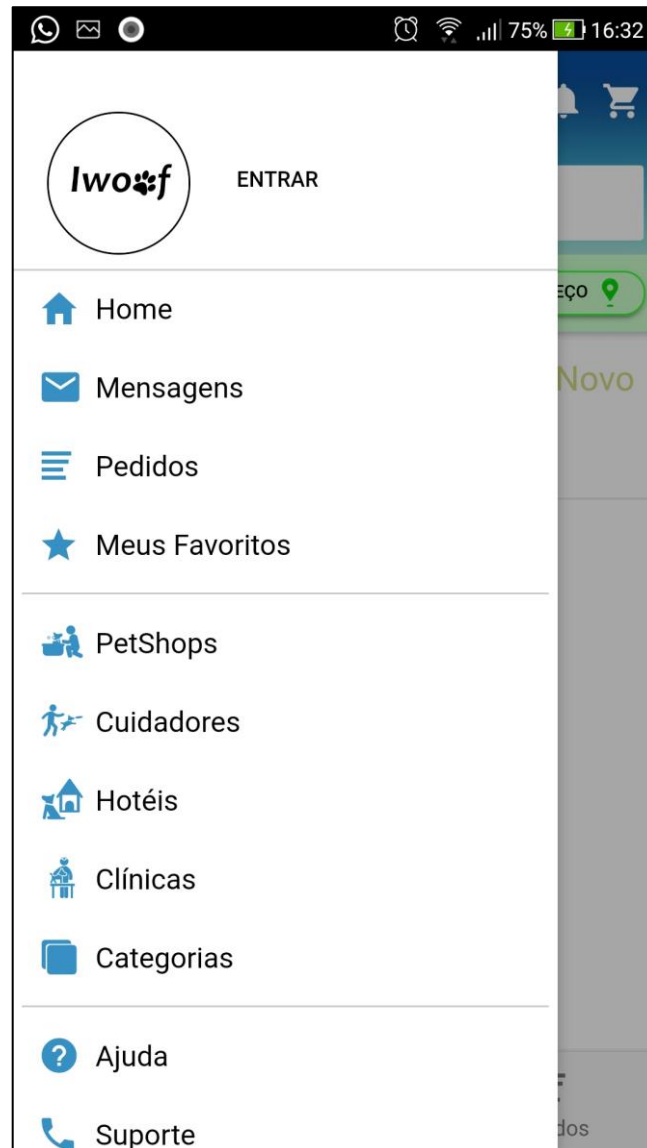
| | Nome ▾ | Tipo | Indicação | Quantidade | Preço | |
|--------------------------|--|---------|------------------|------------|------------|--|
| <input type="checkbox"/> |  Manta Soft | Produto | Cachorros, Gatos | 356 | R\$ 35,90 | Editar Deletar |
| <input type="checkbox"/> |  Casa Carlu Pet House White House | Produto | Cachorros | 180 | R\$ 998,90 | Editar Deletar |
| <input type="checkbox"/> |  Comedouro Jambo Rosa Metálico | Produto | Cachorros | 458 | R\$ 30,00 | Editar Deletar |

Fonte: Elaborado pelos autores

Em relação a Figura 24, são demonstrados os produtos cadastrados pela empresa “Fofinhos Pet Shops”, demonstrando informações como: Nome, Tipo, Indicação, Quantidade e Preço.

O usuário (cliente do Iwoof) poderá se cadastrar pelo aplicativo, a partir de quando poderá entrar no aplicativo e clicar em “Entrar”, conforme Figura 25.

Figura 25 - Login Iwoof-Mobile



Fonte: Elaborado pelos autores

Logo em seguida irá aparecer uma tela de login onde o usuário poderá se cadastrar, utilizando uma conta existente do “Gmail” ou se já tiver uma conta, informar seu e-mail e senha, conforme mostra a Figura 26.

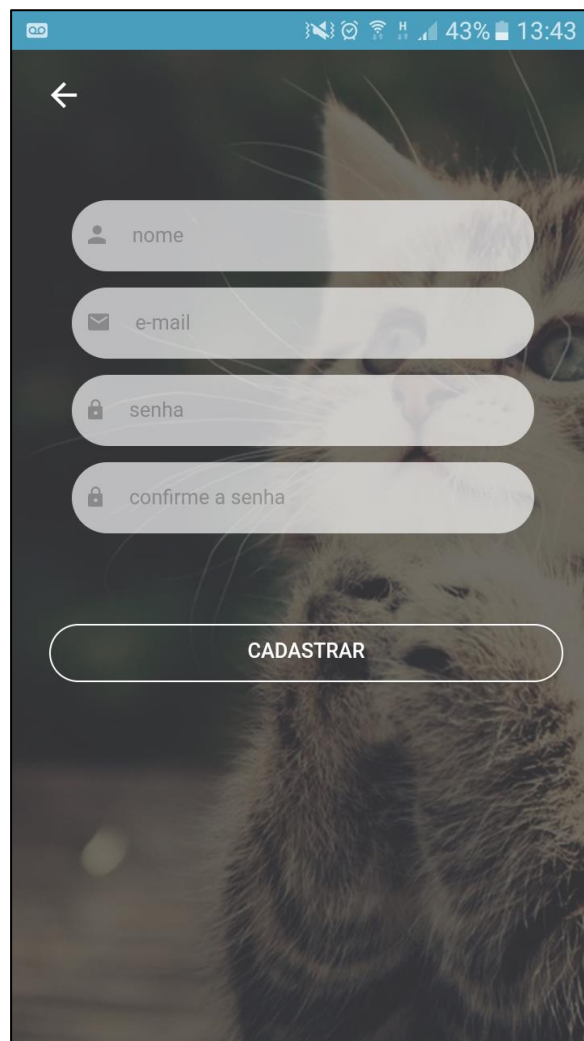
Figura 26 - Tela de login Iwoof-Mobile



Fonte: Elaborado pelos autores

Ao clicar sobre o link “Crie a sua conta aqui”, o aplicativo irá abrir uma tela de formulário onde o cliente poderá digitar suas informações como: Nome, E-mail, Senha e Confrimar Senha, conforme Figura 27.

Figura 27 - Cadastro usuário Iwoof-Mobile

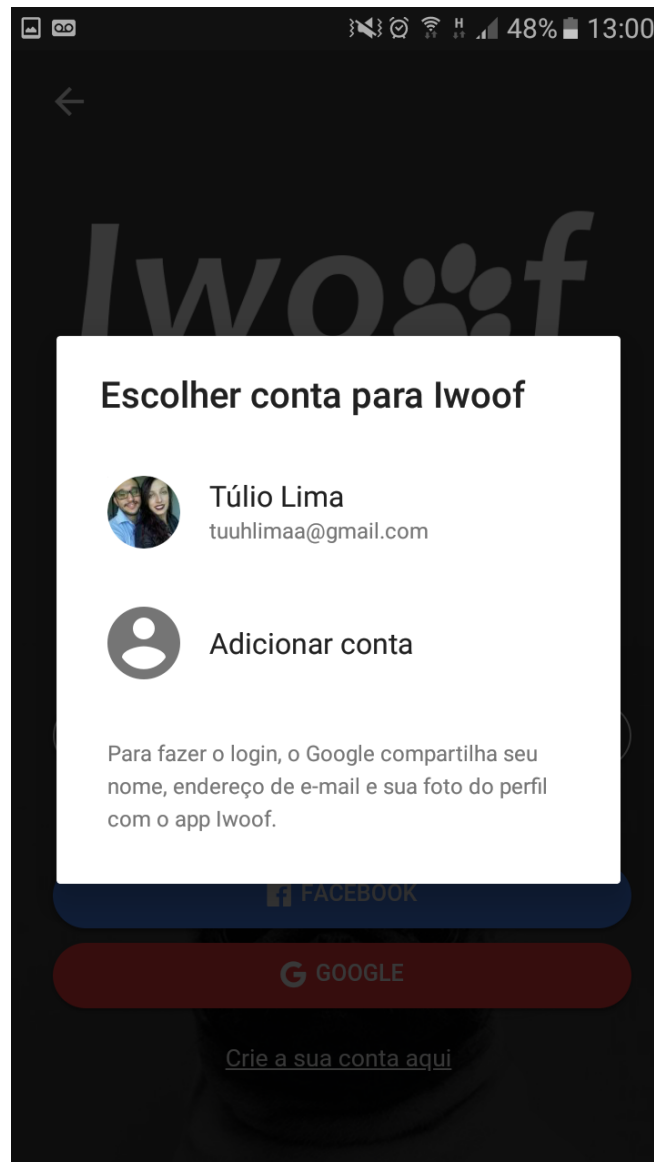


The screenshot displays the user registration interface of the Iwoof-Mobile application. At the top, there is a status bar with various icons and a battery level of 43%. Below the status bar, a back arrow is visible in the top left corner. The main form consists of four rounded rectangular input fields stacked vertically, each with a small icon on the left and placeholder text: 'nome' (with a person icon), 'e-mail' (with an envelope icon), 'senha' (with a lock icon), and 'confirme a senha' (with a lock icon). Below these fields is a prominent 'CADASTRAR' button. The entire form is overlaid on a background image of a dog's face.

Fonte: Elaborado pelos autores

Em relação a Figura 26, caso o usuário clicar em cadastrar pelo Gmail irá abrir uma tela para confirmação de conta, conforme mostra a Figura 28 a seguir.

Figura 28 - Confirmação conta Gmail



Fonte: Elaborado pelos autores

Em relação a Figura 28, o cliente poderá escolher qual conta deseja efetuar o login no sistema.

4.2 Possibilitar a localização de pet shops e empresas relacionadas através do aplicativo

Para o desenvolvimento desta parte do aplicativo utilizamos uma API da Google chamada de Google Maps, essa API fornece um mapa com todas as informações que precisamos. Após o cliente fazer o login no aplicativo dentro da página inicial terá um botão

escrito “meu endereço” no qual ao cliente clicar sobre o ícone, o sistema irá encaminhá-lo para uma página de mapa conforme Figura 29.

Figura 29 - Mapa aplicativo



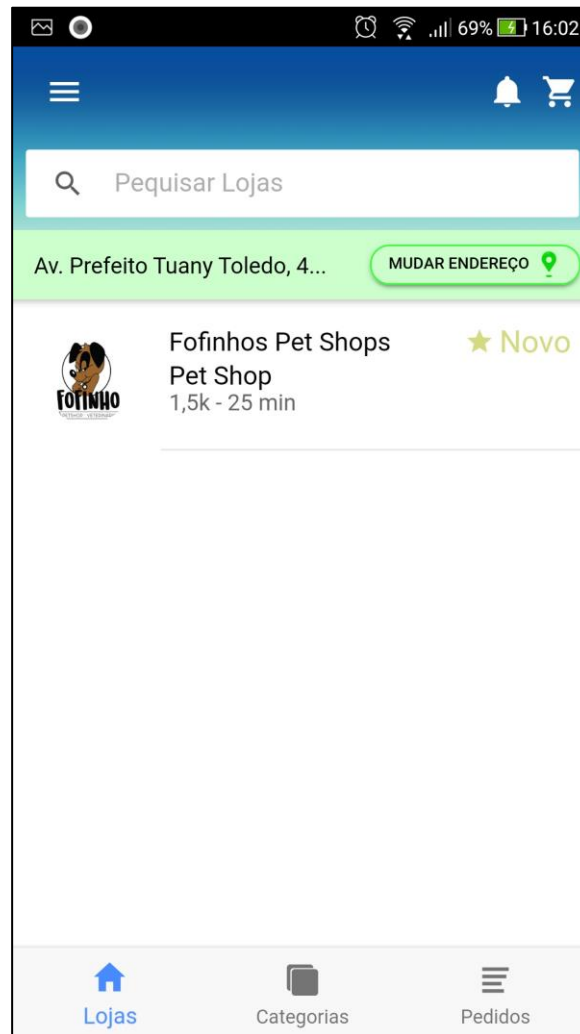
Fonte: Elaborado pelos autores

Em relação a Figura 29, o cliente terá a opção de clicar sobre o botão “Minha Localização”, no qual o aplicativo irá buscar sua localização pelo *GPS*²³ do seu *smartphone*.

Após isso, o cliente terá a opção de selecionar seu endereço clicando sobre o botão “Selecionar Endereço”, no qual o cliente será redirecionado para uma página onde será listado todas as empresas de *pet shops* e similares de seu município, conforme mostra Figura 30.

²³ Sistema de posicionamento global

Figura 30 - Listar empresas



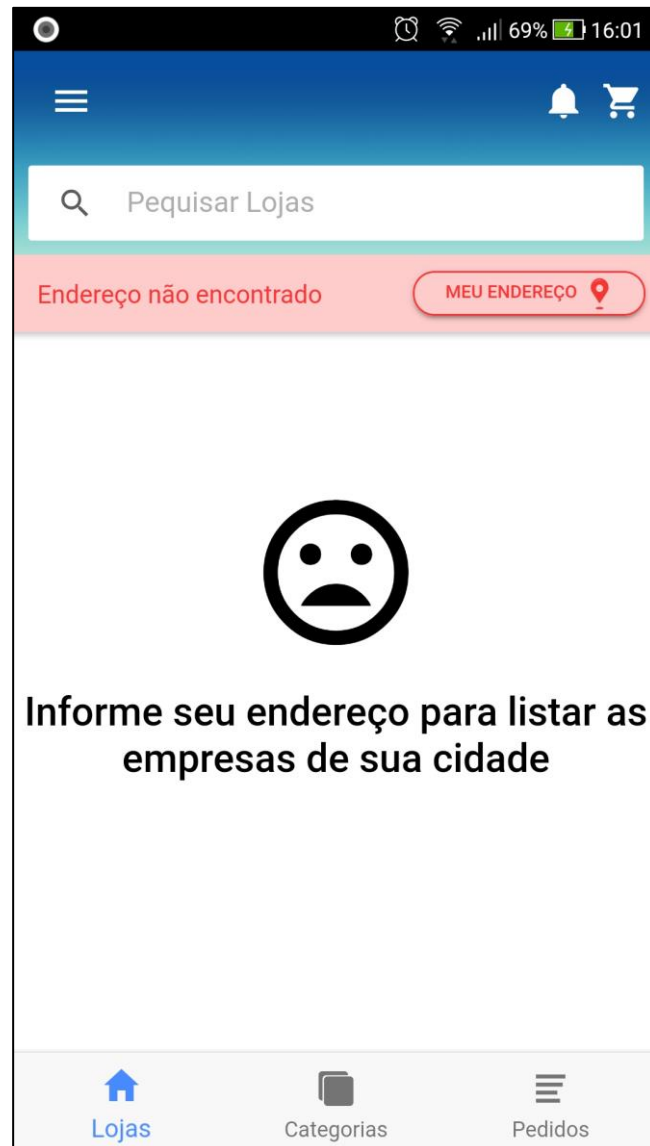
Fonte: Elaborado pelos autores

Em relação a Figura 30, foi listado as empresas que foram cadastradas no banco de dados, foi utilizado uma empresa teste como exemplo: “Fofinhos Pet Shops”.

Se caso o cliente desejar mudar o endereço de sua pesquisa, basta clicar sobre o ícone “Mudar endereço”, no qual irá ser encaminhado a página de mapas conforme a Figura 29.

Se caso o cliente informar um endereço inválido será demonstrado uma tela de erro, conforme a Figura 31.

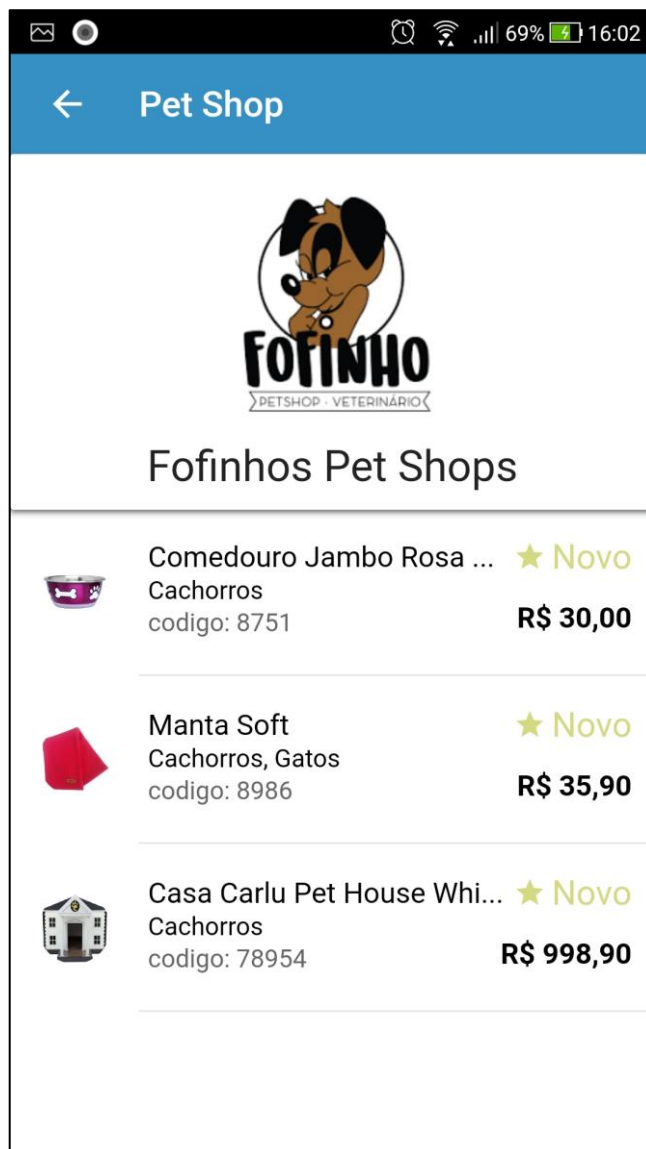
Figura 31 - Endereço inválido



Fonte: Elaborado pelos autores

Após o cliente colocar seu endereço e o aplicativo listar as empresas, o cliente poderá clicar sobre a empresa que deseja, no qual será listado os produtos para visualização de seus preços e tipos de produtos disponíveis, conforme mostra a Figura 32.

Figura 32 - Exemplo de produtos



Fonte: Elaborado pelos autores

Em relação a Figura 32, foi demonstrado os produtos que foram cadastrados pela empresa “Fofinhos Pet Shops”, demonstrando os produtos com suas respectivas descrições, preços e indicações.

Após o cliente clicar sobre o ícone do produto, será redirecionado para uma página no qual poderá escolher a quantidade que deseja comprar daquele determinado produto, como também terá a opção de adicioná-lo no carrinho, conforme mostra a Figura 33.

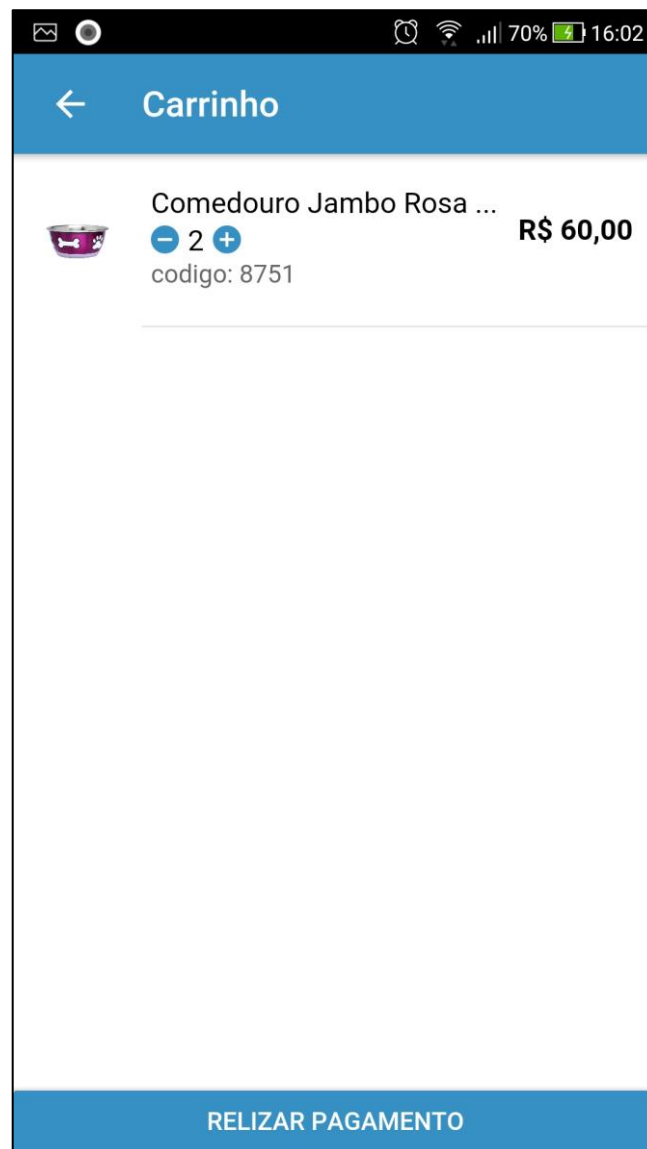
Figura 33 - Detalhes do produto



Fonte: Elaborado pelos autores

Em relação a Figura 33, o cliente terá a opção de colocar a quantidade a ser comprada do produto, foi demonstrado a descrição do produto como qualidades e indicações do produto, o cliente terá a opção de adicionar o produto no carrinho, para isso basta clicar sobre o botão “Adicionar no carrinho”, clicando sobre o botão o cliente será redirecionado para uma página onde poderá realizar o pagamento, conforme mostra Figura 34.

Figura 34 – Itens no carrinho



Fonte: Elaborado pelos autores

Em relação a Figura 34, foi demonstrado os itens que estão no carrinho e seus respectivos valores, o cliente terá a opção de realizar o pagamento clicando sobre o ícone “Realizar Pagamento”, ao clicar o cliente será redirecionado a uma página onde terá a opção de escolher as formas de pagamento, conforme mostra Figura 35.

Figura 35 - Opções de pagamento



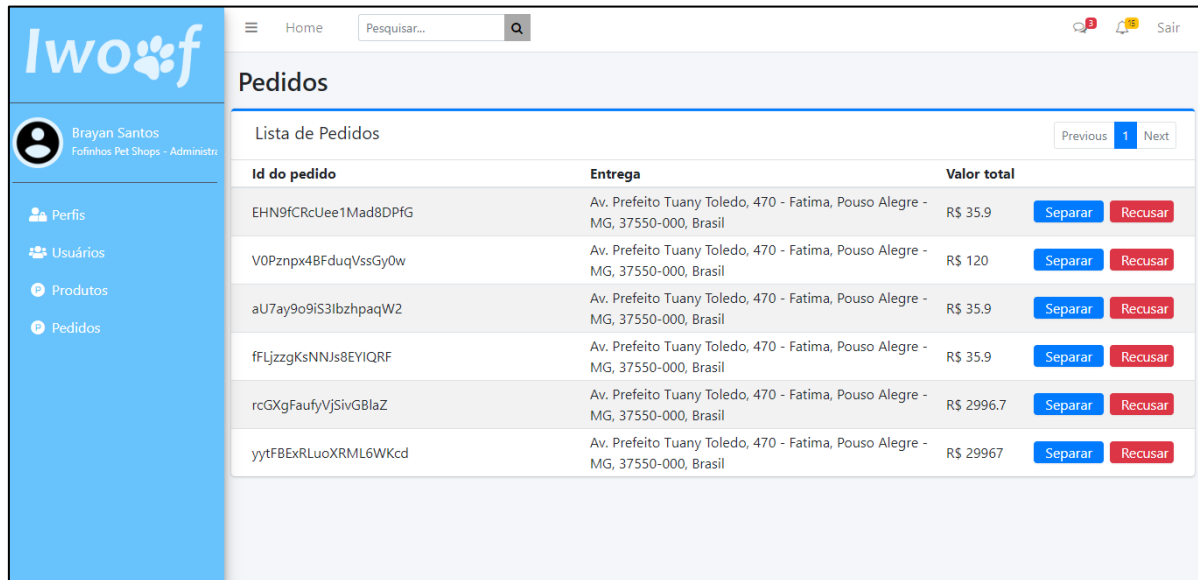
Fonte: Elaborado pelos autores

Em relação a Figura 35, foi demonstrado as formas de pagamento, no qual pode variar de acordo com cada empresa de *pet shop* e similares, o pagamento será realizado na entrega do produto. Em seguida é demonstrado como as empresas irão gerenciar seus pedidos através do web site.

4.3 Possibilitar as empresas gerenciar pedidos através do website

Após o cliente escolher a forma de pagamento, os pedidos serão listados na lista de pedidos conforme Figura 36.

Figura 36 - Gerenciamento de pedidos



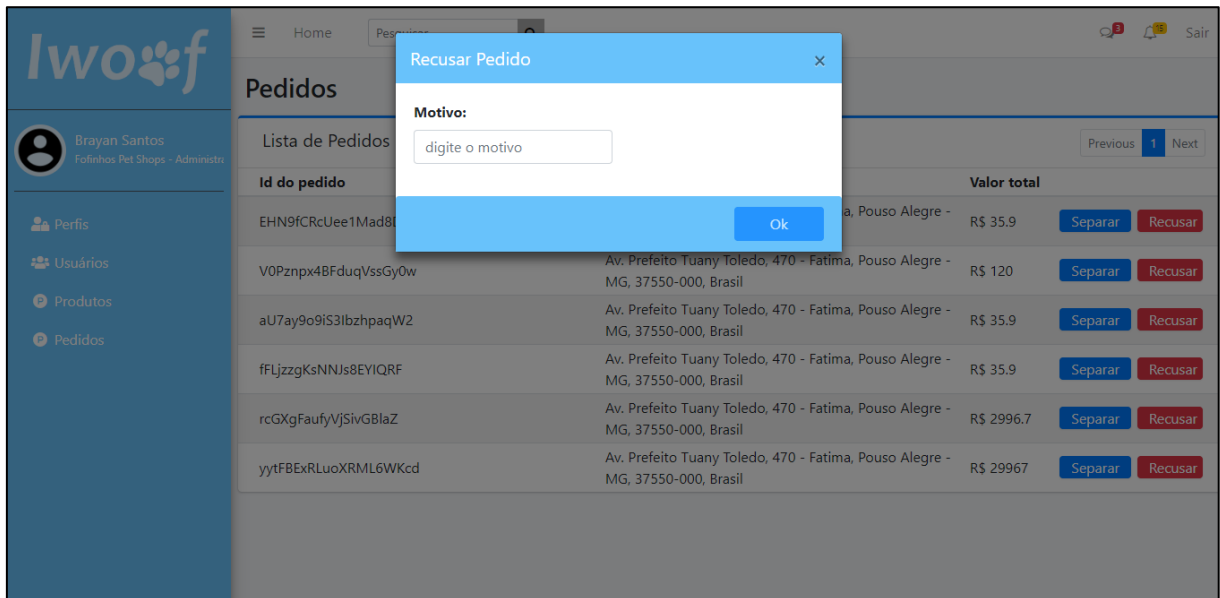
| Id do pedido | Entrega | Valor total | | |
|----------------------|---|-------------|---------|---------|
| EHN9fCRcUee1Mad8DPfG | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 35.9 | Separar | Recusar |
| V0Pznpx4BFduqVssGy0w | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 120 | Separar | Recusar |
| aU7ay9o9iS3lbzhpqW2 | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 35.9 | Separar | Recusar |
| ffLjzzgKsNNJs8EYIQRf | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 35.9 | Separar | Recusar |
| rcGXgFauFyVjSivGBlaZ | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 2996.7 | Separar | Recusar |
| yytFBExRLuoXRML6WKcd | Av. Prefeito Tuany Toledo, 470 - Fatima, Pouso Alegre - MG, 37550-000, Brasil | R\$ 2996.7 | Separar | Recusar |

Fonte: Elaborado pelos autores

Em relação a Figura 36, os pedidos serão listados na tela com as seguintes informações: número do pedido, endereço de entrega e valor da compra, no qual terá um botão de separar ou recusar o pedido.

Se caso a empresa optar por recusar o pedido, a empresa deverá clicar sobre o botão “recusar” no qual abrirá uma tela para a empresa digitar o motivo da recusa, conforme Figura 37.

Figura 37 - Recusar pedido

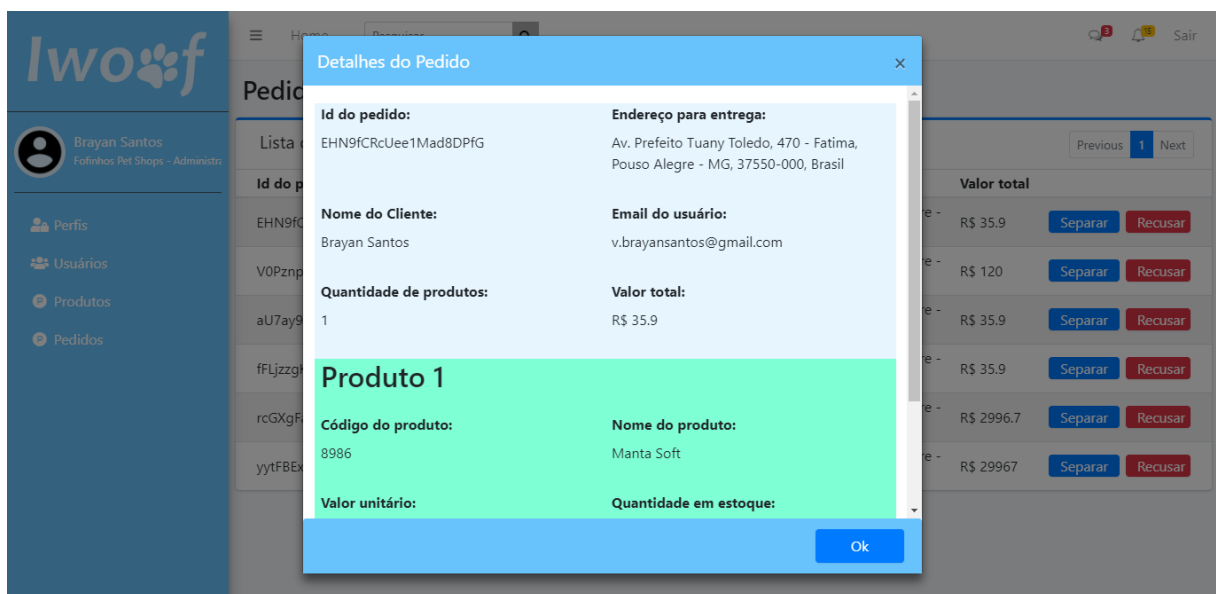


Fonte: Elaborado pelos autores

Em relação a Figura 37, a empresa terá a opção de colocar o motivo pelo qual o pedido foi recusado, clicando sobre o botão “ok” o pedido será recusado, removendo-o da lista de pedidos.

Em relação a Figura 36, se caso a empresa optar por separar o pedido, ela deverá clicar sobre o botão “separar” no qual abrirá uma tela com os detalhes do pedido, conforme Figura 38.

Figura 38 - Detalhes do pedido



Fonte: Elaborado pelos autores

Em relação a Figura 38, foi demonstrado a tela de detalhes do pedido, exibindo as informações como: Id do pedido, endereço para entrega, nome do cliente, e-mail do cliente, quantidade do produto e valor total da compra. Se a empresa optar por separar o pedido, ela deverá clicar sobre o botão “ok”, no qual irá separar o pedido para entrega, removendo-o da lista de pedidos.

Este trabalho nos auxiliou muito na parte técnica de desenvolvimento, pois utilizamos muitas tecnologias, as quais tivemos que pesquisar afim de obter informações como: vídeos tutoriais, sites e livros. A escolha das tecnologias foi bem satisfatória, as tecnologias utilizadas nos auxiliaram por serem tecnologias novas e de fácil utilização, o Firebase foi bastante satisfatório pois nos auxiliou na parte de autenticações e base de dados.

As principais dificuldades enfrentadas dentre elas foram; a falta de tempo para desenvolver o que realmente queríamos, pois tivemos que criar uma API para a criação de um usuário conseguir criar outro usuário e tivemos que utilizar um sistema de e-mail no qual não conhecíamos muito bem. A ideia inicial era possibilitar o cliente fazer pagamentos através do aplicativo, por falta de tempo e imprevistos que foram ocorrendo durante seu desenvolvimento, optamos por possibilitar opções de pagamento para o cliente e como trabalho futuro iremos implementar essa funcionalidade no sistema.

5 CONCLUSÃO

Este trabalho constitui na criação de uma página web e um aplicativo, no qual possibilite as empresas cadastrarem-se, possibilitando inserir seus produtos e serviços no qual integra-se com o aplicativo. Os clientes irão conseguir visualizar essas empresas e escolher vários produtos e serviços iguais a preços variados.

Conclui-se que este trabalho irá auxiliar pessoas a procurar empresas de *pet shops* e similares afim de suprir a escassez de tempo nos dias atuais. Este trabalho irá contribuir para área de desenvolvimento web e mobile, servindo de referência para universidades e alunos que desejam trabalhar com tecnologias que estão em alta no mercado atualmente. Concluímos que o caminho para o sucesso se faz com muita luta e esforço, temos que nos esforçar ao máximo para obter os melhores resultados.

REFERÊNCIAS

ABINPET. **Faturamento 2016 do setor pet aumenta 4,9% e fecha em r\$ 18,9 bilhões, revela abinpet.** 2016. Disponível em: < <http://abinpet.org.br/site/faturamento-2016-do-setor-pet-aumenta-49-e-fecha-em-r-189-bilhoes-revela-abinpet/> > Acesso em: 10 jun. 2018.

ANGULAR (s.d.). **Visão Geral da Arquitetura.** Disponível em: <<https://angular.io/guide/architecture>> Acesso em: 23 mar. 2018.

BITBUCKET (2018). **Construído para equipes profissionais.** Disponível em: <<https://bitbucket.org/product>> Acesso em: 03 out. 2018.

CLOUD GOOGLE (s.d.). **Mapas.** Disponível em: <<https://cloud.google.com/maps-platform/maps/?hl=pt-BR>> Acesso em: 10 agost. 2018.

EIS, D. ; FERREIRA, E. **HTML5 e CSS3 com farinha e pimenta.** 2012. Acesso em: 20 mar. 2018.

GOIS, A. **Construa aplicativos para todas as plataformas mobile.** 2017. Acesso em: 11 abr. 2018.

GIL, A.C. **Como elaborar projetos de pesquisa.** 2002. Disponível em: <https://professores.faccat.br/moodle/pluginfile.php/13410/mod_resource/content/1/como_elaborar_projeto_de_pesquisa_-_antonio_carlos_gil.pdf> Acesso em: 10 jun. 2018.

IONIC FRAMEWORK (s.d.). **Sobre o Ionic.** Disponível em: < <https://ionicframework.com/docs/v1/overview/#about> > Acesso em: 13 abr. 2018.

MARTINS, J.C.C. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML – 5º edição.** 2010. Disponível em: <https://books.google.com.br/books?id=8ect3L-yozkC&pg=PA163&dq=casos+de+uso&hl=pt-BR&sa=X&ved=0ahUKEwjih-zMnYHeAhWDfZAKHYdJABoQ6AEILjAB#v=onepage&q=casos%20de%20uso&f=false> > Acesso em: 12 out. 2018.

PARANHOS, L.R.L.; RODOLPHO, P.L. **Metodologia da pesquisa aplicada à tecnologia.** 2014. Acesso em: 17 abr. 2018.

PEREIRA, C.R. **Aplicações web real-time com Node.js.** 2014. Disponível em: https://books.google.com.br/books?id=Wm-CCwAAQBAJ&printsec=frontcover&dq=node+js&hl=pt-BR&sa=X&ved=0ahUKEwiBs4ys3-_dAhWBDZAKHZv8D2QQ6AEITDAE#v=onepage&q=node%20js&f=false > Acesso em 05 out. 2018.

PEREIRA, C.R. **Construindo APIs REST com Node.js.** 2016. Disponível em: < https://books.google.com.br/books?id=byCjCwAAQBAJ&printsec=frontcover&dq=node+js&hl=pt-BR&sa=X&ved=0ahUKEwit8KX-5u_dAhWCIZAKHYQOCnAQ6AEIUjAF#v=onepage&q=node%20js&f=false > Acesso em 05 out. 2018.

PETROUTSOS, E. **Google Maps: Power Tools for Maximizing the API**. Acesso em: 15 abr. 2018.

RAMOS, R.A. **Treinamento prático em UML**. 2006. Disponível em: <https://books.google.com.br/books?id=cE4qBWwJM1sC&printsec=frontcover&dq=UML+diagrama+de+casos+de+uso&hl=pt-BR&sa=X&ved=0ahUKEwjOsu_H5urdAhUMg5AKHRZbB3oQ6AEIKDAA#v=onepage&q=UML%20diagrama%20de%20casos%20de%20uso&f=false> Acesso em: 03 out. 2018.

SILVA, M.S. **A linguagem de marcação que revolucionou a web**. 2º edição 2014. Acesso em: 20 mar. 2018.

SILVA, M.S. **CSS3 desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. 2011. Acesso em: 20 mar. 2018.

SILVA, M.S. **Fundamentos de HTML5 e CSS3**. 2015. Acesso em: 21 mar. 2018.

SMYTH, N. **Firebase Essentials**. 2017. Acesso em: 15 abr. 2018.

WEBJUMP (s.d.). **O que é marketplace?**. Disponível em: <<http://www.webjump.com.br/marketplace/o-que-e-marketplace/>> Acesso em 03 out. 2018.