



GOVERNO DE
SÃO PAULO **Fatec**
São Caetano
do Sul
Antonio Russo

FATEC – FACULDADE DE TECNOLOGIA DE SÃO CAETANO DO SUL

TÉCNOLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ANDREZA CADÊTE DA SILVA

BRUNO MONTEIRO DE MORAIS

RAFAEL ROCHA DE MELO

VINICIUS HERRERA DA SILVA

**TRABALHO DE CONCLUSÃO DE CURSO:
LOCALIZAPET**

SÃO CAETANO DO SUL - SP

Junho, 2020

ANDREZA CADÊTE DA SILVA
BRUNO MONTEIRO DE MORAIS
RAFAEL ROCHA DE MELO
VINICIUS HERRERA DA SILVA

**TRABALHO DE CONCLUSÃO DE CURSO:
LOCALIZAPET**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do(a) Professor(a) MSc Flávio Viotti, como requisito parcial para a obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas.

SÃO CAETANO DO SUL - SP

Junho, 2020

AGRADECIMENTOS

Considerando todo o processo de idealização e desenvolvimento deste trabalho, gostaríamos de agradecer à todas as pessoas que dividiram esses momentos conosco, valendo ressaltar algumas pessoas que tornaram este trabalho uma realidade.

Dentre estas pessoas, gostaríamos de agradecer ao nosso orientador, professor Flávio Viotti, que durante todo o processo se colocou sempre à disposição para sanar nossas dúvidas e nos guiar para que pudéssemos finalizar o desenvolvimento deste trabalho. Também vale mencionar que foi o responsável por nos ensinar a base do desenvolvimento orientado à objetos, paradigma de desenvolvimento adotado em boa parte do desenvolvimento do nosso projeto.

Nossos amigos e familiares, que estiveram do nosso lado durante todo esse período de desenvolvimento deste trabalho, apoiando-nos de forma compreensiva, entendendo a importância deste trabalho para os integrantes deste grupo, tanto pessoal quanto no âmbito educacional.

À professor Rosana Maria Traversa, que dentre algumas matérias que nos lecionou está a matéria de banco de dados, que nos proporcionou uma boa base de banco de dados relacional, o que facilitou bastante o desenvolvimento da persistência do nosso projeto.

Ao professor Adilson Ferreira da Silva, que nos ensinou sobre gestão de projetos compartilhando situações as quais viveu durante sua vida profissional, o que facilitou bastante o nosso entendimento do assunto, e mais do que isso, nos proporcionou uma visão maior sobre as possibilidades da área de tecnologia da informação.

RESUMO

Este é um trabalho que tem por sua finalidade apresentar todas as etapas do processo de desenvolvimento de um sistema para a solução de um determinado problema. Será documentado o desenvolvimento de um aplicativo, para aparelhos com o sistema Android, chamado “LocalizaPet”, desenvolvido utilizando a ferramenta Android Studio, com uma interface bastante simples e limpa, porém com ferramentas visuais para facilitar o seu uso, como por exemplo, a API de mapas da Google, o GoogleMaps. Este aplicativo foi desenvolvido com a finalidade de proporcionar uma ferramenta que facilite a busca por animais perdidos, o contato de outros usuários com os donos de animais que foram perdidos ou contato dos donos com usuários que encontraram seu animal perdido, a doação de animais por aqueles que não podem dar os cuidados necessários ao animal, e dispor para seus usuários informações de instituições sem fins lucrativos voltadas a animais domésticos com intuito de que eles saibam quem procurar para sanar suas dúvidas em relação aos animais encontrados, abandonados ou mal tratados.

Palavras -chave: Animais Domésticos; Android; Mapas; API; GoogleMaps

ABSTRACT

This final's paper objective is to present every step of the process of developing a system to solve a specific problem. It will be the development documentation of an application called "LocalizaPet" for mobile devices that uses Android as its operational system. It was developed using Android Studio IDE, and was developed with a very simple and clean interface, alongside with some visual tools to make its use easier like the Google Maps API. LocalizaPet was developed in due to provide a tool that makes easier the search for lost animals, the contact of other users with owners of lost pets or contact of pet owners and other users whom may have found a lost pet, pet donations for those who owns pets but may not have condition to take care of them, and provide users information about non profitable organizations which have as principal subject domestic animals so them know who to find that may answer questions about found animals, abandoned or mistreated.

Palavras -chave: Domestic animals; Android; Maps; API; GoogleMaps

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de Casos de Uso	18
Figura 2 - Diagrama de Classes: Pacote Viewer	25
Figura 3 - Diagrama de Classes: Pacote Controller	26
Figura 4 - Diagrama de Classes: Pacote Model	26
Figura 5 - Diagrama de Classes: Pacote DAO	27
Figura 6 - Diagrama de Sequência: Buscar Animal (Fluxo Principal)	27
Figura 7 - Diagrama de Sequência: Buscar Animal (Fluxo Alternativo 1)	28
Figura 8 - Diagrama de Sequência: Buscar Animal (Fluxo Alternativo 2)	28
Figura 9 - Diagrama de Sequência: Analisar Proximidade (Fluxo Principal)	29
Figura 10 - Diagrama de Sequência: Analisar Proximidade (Fluxo Alternativo)	30
Figura 11 - Diagrama de Transição de Estado: Análise de Proximidade	31
Figura 12 - Diagrama de Transição de Estado: Busca Animal	31
Figura 13 - Diagrama de Transição de Estado: Enviar Mensagem	32
Figura 14 - Diagrama de Componentes	33
Figura 15 - Diagrama de Implantação	33
Figura 16 - Diagrama de Atividades: Busca Animal	34
Figura 17 - Diagrama de Atividades: Análise de Proximidade	35
Figura 18 - Diagrama de Atividades: Enviar Mensagem	36
Figura 19 - Modelo Entidade-Relacionamento	37
Figura 20 - Modelo Lógico do Banco de Dados	38
Figura 21 - Tela de Acesso (Login)	46
Figura 22 - Tela de Registrar Usuário	47
Figura 23 - Tela de Recuperação de Senha	48
Figura 24 - Tela de Recuperação de Senha (E-mail)	48
Figura 25 - Tela de Redefinição de Senha	49
Figura 26 - Tela de Anúncios	50
Figura 27 - Tela de Registrar Anúncio	51
Figura 28 - Menu Principal	52
Figura 29 - Tela de Perfil	53
Figura 30 - Tela de ONGs	54
Figura 31 - Tela de Registro de ONG	55
Figura 32 - Tela de Conversas (Chat)	55

Figura 33 - Tela de Conversa (Chat)	56
Figura 34 - Tela de “Meus Anúncios”	57
Figura 35 - Tela de Mapa (Anúncios)	58
Figura 36 - Tela de Detalhes (Mapa)	59

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais	16
Quadro 2 - Requisitos Não Funcionais	17
Quadro 3 - CSU: Manter usuário	19
Quadro 4 - CSU: Manter Anúncio	20
Quadro 5 - CSU: Buscar Animal	21
Quadro 6 - CSU: Analisar Proximidade	22
Quadro 7 - CSU: Manter ONG	23
Quadro 8 - CSU: Enviar Mensagem	24
Quadro 9 - Dicionário de Dados - Tabela usuário	39
Quadro 10 - Dicionário de Dados - Tabela Log	39
Quadro 11 - Dicionário de Dados - Tabela Chat Msg	39
Quadro 12 - Dicionário de Dados - Tabela Chat	40
Quadro 13 - Dicionário de Dados - Tabela Foto	40
Quadro 14 - Dicionário de Dados - Tabela Animal	40
Quadro 15 - Dicionário de Dados - Tabela Anúncio	40
Quadro 16 - Dicionário de Dados - Tabela Tipo Anúncio	41
Quadro 17 - Dicionário de Dados - Tabela Anúncio End	41
Quadro 18 - Dicionário de Dados - Tabela ONG	41
Quadro 19 - Dicionário de Dados - Tabela ONG End	41
Quadro 20 - Dicionário de Dados - Tabela Endereço	41
Quadro 21 - Cenário: Manter Usuário	61
Quadro 22 - Cenário: Enviar Mensagem	62
Quadro 23 - Cenário: Buscar Animal	62
Quadro 24 - Cenário: Analisar Proximidade	63
Quadro 25 - Cenário: Manter Anúncio	63
Quadro 26 - Cenário: Manter ONGs	65

LISTA DE ABREVIACÕES E SIGLAS

SGBD – Sistema de Gerenciamento de Banco de Dados

ONG – Organização Não Governamental

API – *Application Programming Interface*

MVC – Modelo-Visão-Controlador

REST - *Representation State Transfer*

SOAP - *Simple Object Access Protocol*

HTTP - *HyperText Transfer Protocol*

XML - *eXtensible Markup Language*

WSDL - *Web Services Description Language*

URL - Uniform Resource Locator

JSON - *JavaScript Object Notation*

UML - *Unified Modeling Language*

SGDB - Servidor e Gerenciador de Banco de Dados

SUMÁRIO

1. INTRODUÇÃO	12
1.1. Problema	12
1.2. Hipóteses	12
1.3. Objetivos	13
1.3.1. Objetivo Geral	13
1.3.2. Objetivos Específicos	13
1.4. Justificativa	13
1.5. Metodologia	13
1.6. Resultados Esperados	15
2. ENGENHARIA DE REQUISITOS	16
2.1. Requisitos Funcionais	16
2.2. Requisitos Não Funcionais	17
2.3. Regras de Negócios	17
2.4. Diagrama de Casos de Uso	18
2.4.1. Descrição dos Casos de Uso	19
3. ANÁLISE E PROJETO	25
3.1. Diagrama de Classes	25
3.2. Diagrama de Sequência e de Comunicação	27
3.3. Diagrama de Transições de Estados	31
3.4. Diagrama de Componentes e de Implantação	33
3.5. Diagrama de Atividades	34
3.6. Modelo Entidade – Relacionamento	37
3.7. Modelo Lógico do Banco de Dados	38
3.8. Pequeno Dicionário de Dados	39
4. SOFTWARE E TECNOLOGIAS EMPREGADAS NA PROGRAMAÇÃO DO	

SISTEMA	42
4.1 <i>Cloud Computing</i> (computação em nuvem)	42
4.2 <i>Web Service</i>	42
4.3 <i>JSON</i>	43
4.4 API Google Maps	44
4.5 Android Studio	44
4.6 MySQL	44
4.7 SQLite	45
5. DETALHAMENTO DO PROTÓTIPO	46
5.1 Telas e Navegação	46
6. TESTES	60
6.1 Teste caixa-branca (ou Teste Estrutural)	60
6.2 Teste caixa-preta (ou Teste Funcional)	60
7. RESULTADOS	67
8. REFERÊNCIAS	69

1. INTRODUÇÃO

Este trabalho é sobre o desenvolvimento de um aplicativo mobile para facilitar o encontro de animais perdidos e seus donos.

O tema foi escolhido com base na dificuldade que os donos de *pets* possuem em encontrar seus animais de estimação, na dificuldade que as pessoas que encontram animais na rua possuem em encontrar seus respectivos donos para devolvê-los e também na dificuldade que as pessoas encontram em disponibilizar ou procurar por animais para adoção.

O aplicativo funciona como uma comunidade colaborativa para compartilhar informações de animais perdidos, encontrados e disponíveis para adoção.

Foi realizada uma busca de aplicativos que tenham a mesma ideia deste projeto como núcleo, através dessa busca, constatamos que existem aplicativos semelhantes, porém alguns não estão mais disponíveis na Google Store como o *TagMyPet* (aplicativo gratuito) e o *PinMyPet* (aplicativo pago). Os demais aplicativos encontrados possuem uma restrição de plataforma, por exemplo, o aplicativo chamado “Procura-se Cachorro” está disponível apenas para aparelhos da Apple (IOS), diferente do “Pet.me” que encontrasse disponível apenas na Play Store para aparelhos que utilizam o sistema operacional Android, ambos são gratuitos.

O aplicativo permite o cadastro de animais, sendo este composto por informações dos animais, tais como: raça, características físicas que facilitam a sua identificação, se ele é um animal desaparecido, encontrado ou disponível para adoção, sua última localização conhecida e uma foto do animal. Todo o registro de animal permite que outras pessoas entrem em contato com a pessoa responsável por aquele registro através de um chat dentro do próprio aplicativo. Nossa ferramenta também disponibiliza informações sobre ONGs onde os usuários podem entrar em contato e sanar dúvidas sobre como fazer denúncias de maus tratos a animais que foram encontrados.

A interface será simples e amigável, evitando complicações na hora de cadastro e atualizações de informações dos animais.

1.1. Problema

O problema que encontramos quando pensamos neste domínio foi: Como ajudar pessoas a encontrarem seus *pets* desaparecidos com o auxílio tecnológico?

1.2. Hipóteses

Para ajudar de forma tecnológica a criação e divulgação de uma comunidade onde as pessoas possam compartilhar informações sobre os *pets* foi o meio escolhido.

1.3. Objetivos

Os objetivos estão divididos entre objetivos gerais, que é o onde apresentaremos a ideia principal e objetivos específicos, através dos quais serão apresentados maiores detalhes da execução do trabalho.

1.3.1. Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma ferramenta capaz de auxiliar na busca de animais perdidos, donos de animais encontrados e possíveis donos para animais disponíveis para doação.

1.3.2. Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Ter interfaces amigáveis para o usuário;
- Apresentar informações detalhadas do desaparecimento, encontro ou adoção de algum animal;
- Disponibilizar um feed de animais perdidos, encontrados e disponíveis para adoção;
- Permitir a postagem de fotos dos animais;
- Disponibilizar como forma de contato entre os usuários um chat dentro da aplicação;
- Cruzar as informações inseridas pelo usuário com os dados dos animais cadastrados e exibir o resultado da busca.

1.4. Justificativa

O que nos impulsionou na realização deste trabalho foi a dificuldade que notamos ao buscar por animais desaparecidos e achar os donos de animais que foram encontrados, visto que as alternativas de tecnologia não são tão efetivas ou até mesmo foram descontinuadas por seus idealizadores.

A melhor maneira que enxergamos de proporcionar uma solução para esse problema foi fornecer um aplicativo mobile para o compartilhamento de informações sobre desaparecimentos, encontros e adoções, visto que hoje em dia boa parte da população possui um aparelho *smartphone* e isto aumentará o alcance dessas informações, conseqüentemente melhorando as chances de um resultado positivo.

1.5. Metodologia

1.5.1. Tipos de Metodologia

Para o desenvolvimento de um trabalho científico, é necessário que seja feita uma série de escolhas relacionadas à onde, como e com quais ferramentas serão feitas as pesquisas para que o trabalho seja realizado. Ou seja, é necessário definir o tipo de pesquisa que será

realizada e o seu universo, bem como os instrumentos que serão utilizados para coleta e análise de dados.

1.5.2. Tipos de Pesquisa

Segundo Antonio Carlos Gil, autor da obra “Métodos e Técnicas de Pesquisa Social”, é possível dividir a pesquisa em três tipos segundo o seu objetivo.

1.5.2.1. Pesquisa Exploratória

A pesquisa exploratória é o levantamento de informações sobre determinado objeto de pesquisa, de forma a aumentar a familiaridade com ele e formular problemas e hipóteses mais precisos. Geralmente, trata-se de uma pesquisa bibliográfica ou um estudo de caso.

1.5.2.2. Pesquisa Descritiva

A pesquisa descritiva é o registro e análise de características de um objeto de pesquisa, evitando a interferência do pesquisador, a partir de técnicas padronizadas de coleta e análise de dados.

1.5.2.3. Pesquisa Explicativa

A pesquisa explicativa é o registro, análise e interpretação de um objeto de pesquisa, com o principal objetivo de identificar e explicar suas causas. É uma pesquisa mais complexa, que busca verificar hipóteses causais. Nas ciências naturais, costuma-se usar o método experimental; nas ciências humanas, é adotado especialmente o método observacional.

1.5.3. Métodos de Pesquisa

Os métodos de pesquisa são as descrições de como serão as abordagens para a obtenção de informações e de dados relacionados ao objeto de pesquisa. Sendo algumas delas descritas a seguir.

1.5.3.1. Pesquisa Experimental

A pesquisa experimental tem como base a observação a partir de experimentos controlados, com alterações de variáveis e instrumentos de coleta de dados submetidos a testes que assegurem a sua eficácia, além de análise estatística de resultados.

1.5.3.2. Pesquisa Bibliográfica

A pesquisa bibliográfica baseia-se no levantamento de informações e conhecimentos a cerca de um tema a partir de diferentes materiais bibliográficos já publicados, colocando em diálogo diferentes autores e dados.

1.5.3.3. Pesquisa Documental

A pesquisa documental tem como característica o levantamento e análise de materiais que ainda não receberam um tratamento analítico, como documentos oficiais, cartas, filmes, *softwares*, entre outros.

1.5.4. Metodologia Adotada

A partir do que foi pesquisado, foi decidido adotar a pesquisa exploratória como base deste trabalho, por ser um tema que foi considerado estar muito próximo das pessoas e por ser considerado importante a aproximação dos autores desta pesquisa ao cenário utilizado como objeto de pesquisa.

Dito isto, o método de pesquisa adotado foi o método de pesquisa documental, considerando que trata-se do desenvolvimento de uma ferramenta que possivelmente estaria concorrendo com outras ferramentas que possam propor soluções similares para o problema identificado, então tornando bastante importante a análise de *softwares*, de algumas aparições deste cenário em ferramentas do dia-a-dia das pessoas, como por exemplo as redes sociais e de documentações técnicas e teóricas relacionadas a desenvolvimento de ferramentas móveis com valor.

1.6. Resultados Esperados

Com o desenvolvimento e conclusão deste trabalho, é esperado que o conteúdo instruído pelos docentes e instituição seja aplicado de forma correta, fazendo assim com que o trabalho venha a ter sucesso e a experiência adquirida pelo grupo adquirida com o desenvolvimento dele seja um bom preparatório para o mercado de trabalho.

Além disso, é esperado que as pessoas desfrutem desse aplicativo como uma ferramenta para auxiliá-las a encontrar seus animais de estimação.

2. ENGENHARIA DE REQUISITOS

Segundo MACIASZEK (2000), citado por Eduardo Bezerra, um requisito é uma condição ou capacidade que deve ser alcançada ou possuída por um sistema ou um de seus componentes para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos.

Os requisitos são separados em dois grupos, os funcionais e não funcionais. Como o próprio nome já diz, os requisitos funcionais estão relacionados as funcionalidades do sistema, já os requisitos não funcionais estão relacionados às características de qualidade que este sistema deve possuir e que estão relacionadas às suas funcionalidades.

Tendo em mente estas definições, a seguir serão apresentados os requisitos que foram levantados para a solução do problema proposto neste trabalho.

2.1. Requisitos Funcionais

Quadro 1 - Requisitos Funcionais

ID do Requisito	Descrição	Caso de Uso
RF01	O aplicativo deverá permitir o cadastro e a alteração de usuários.	CSU001
RF02	O aplicativo deverá permitir o acesso a suas funcionalidades somente à usuários cadastrados.	CSU001
RF03	O aplicativo deverá apresentar se o anúncio é de um animal perdido, encontrado ou para adoção.	CSU002
RF04	O aplicativo deverá cruzar os anúncios registrados de animais perdidos e encontrados para auxiliar a busca do usuário.	CSU003
RF05	Os anúncios dos animais deverão acompanhar foto do animal.	CSU002
RF06	O aplicativo deverá disponibilizar um chat para os usuários se comunicarem.	CSU006
RF07	O aplicativo deverá permitir o cadastro, a alteração, remoção e consulta de anúncios de animais encontrados.	CSU002
RF08	O aplicativo deverá permitir o cadastro, a alteração, remoção e consulta de anúncios de animais perdidos.	CSU002
RF09	O aplicativo deverá permitir a busca de anúncios de	CSU004

	animais desaparecidos ou encontrados em um raio de 300 metros em torno do usuário.	
RF10	O aplicativo deverá permitir o cadastro, a alteração, remoção e consulta de anúncios de animais disponíveis para adoção.	CSU002
RF11	O aplicativo deverá permitir o cadastro, a alteração, remoção e consulta de ONGs.	CSU005

FONTE: Autoria Própria, 2019

2.2. Requisitos Não Funcionais

Quadro 2 - Requisitos Não Funcionais

ID do Requisito	Descrição
RNF01	As telas deverão ser intuitivas, contendo textos autoexplicativos, imagens e botões
RNF02	O sistema deverá ser desenvolvido para dispositivos com sistema operacional Android
RNF03	O sistema deverá possuir uma base de dados utilizando SGBD MySQL.
RNF04	O sistema deverá ser desenvolvido na linguagem Java
RNF05	A aplicação deverá ser desenvolvida no Android Studio
RNF06	A aplicação deverá realizar a atualização do raio de busca com base na localização atual do usuário
RNF07	O servidor <i>web</i> da aplicação deverá ser construída com PHP7.4
RNF08	A aplicação deverá funcionar utilizando o servidor <i>web Restful</i>

FONTE: Autoria Própria, 2019.

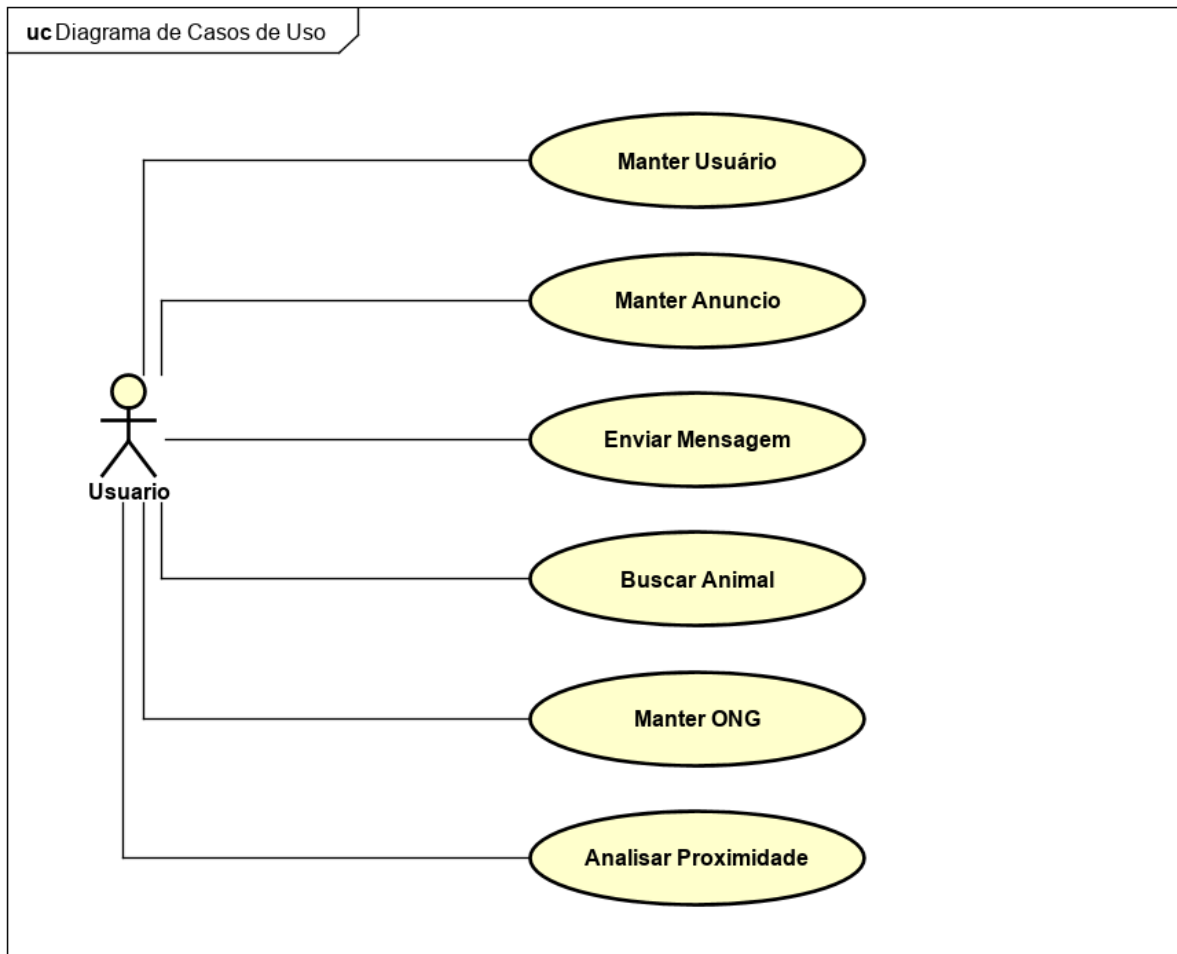
2.3. Regras de Negócios

- Em caso de busca por proximidade, o usuário poderá escolher entre fazer uma busca a partir da sua localização atual.
- Uma busca por proximidade apresentará no mapa somente anúncios registrados dentro de um raio de 300 metros.
- Os usuários poderão se comunicar utilizando um sistema de chat dentro do próprio aplicativo.

- O sistema apresentará para o usuário uma forma de busca de anúncios de encontros, desaparecimentos ou adoção que possam estar relacionados a algum anúncio feito por ele.

2.4. Diagrama de Casos de Uso

Figura 1 - Diagrama de Casos de Uso



FONTE: Autoria Própria, 2019

2.4.1. Descrição dos Casos de Uso

Quadro 3 - CSU: Manter usuário

Caso de uso: Manter usuário
ID: CSU001
Resumo: Este caso de uso descreve as etapas do processo de cadastro, alteração de usuários da aplicação.
Atores Primários: usuário.
Atores Secundários: N/A
Pré-Condições: <ol style="list-style-type: none">1. O usuário ter acesso à internet
Fluxo de eventos principal: <ol style="list-style-type: none">1. O Sistema apresenta a tela de carregamento para o usuário2. O Sistema verifica a conexão com o servidor3. O Sistema apresenta a tela de login para o usuário (A2)4. O usuário clica em novo cadastro (A1)5. O usuário informa seus dados6. O Sistema salva os dados no banco de dados7. O Sistema informa ao usuário que o cadastro foi efetuado com sucesso8. O Sistema apresenta ao usuário a tela para realizar o login9. Finalizar caso de uso
Pós-Condições: <ol style="list-style-type: none">1. O usuário adquire acesso ao Sistema
Fluxo de eventos alternativo: Alterar dados do usuário (A1) <ol style="list-style-type: none">4. O usuário efetua o login5. O Sistema apresenta a tela de Anúncios6. O usuário clica no botão “Perfil”7. O Sistema apresenta a tela de Perfil com as informações do usuário8. O usuário clica no botão “Editar”9. O Sistema apresenta a tela de Alterar Informações do usuário10. O usuário informa seus dados atualizados11. O Sistema atualiza os dados no banco de dados12. O Sistema informa ao usuário que a alteração foi efetuada com sucesso13. Finalizar caso de uso
Fluxo de eventos alternativo: Falha ao conectar com o servidor (A2) <ol style="list-style-type: none">3. O Sistema informa ao usuário que ocorreu uma falha ao conectar com o servidor.4. Finalizar caso de uso

FONTE: Autoria Própria, 2019

Quadro 4 - CSU: Manter Anúncio

Caso de uso: Manter Anúncio
ID: CSU002
<p>Resumo:</p> <p>Este caso de uso descreve as etapas do processo de cadastro, atualização e exclusão de anúncios de animais desaparecidos, encontrados e para adoção.</p>
Atores Primários: usuário.
Atores Secundários: N/A
<p>Pré-Condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar logado 2. O usuário deve estar na tela de Anúncios
<p>Fluxo de eventos principal:</p> <ol style="list-style-type: none"> 1. O usuário clica no botão “Novo Anúncio” (A1) 2. O Sistema apresenta a tela de registrar anúncio 3. O usuário insere os dados do animal 4. O usuário insere o endereço onde o animal foi perdido, encontrado ou do dono do animal (adoção) 5. O usuário insere fotos do animal 6. O sistema registra o anúncio no banco de dados 7. O sistema informa que o anúncio foi cadastrado com sucesso 8. O Sistema apresenta a tela de Anúncios com a lista atualizada 9. Finalizar caso de uso
<p>Pós-Condições:</p> <ol style="list-style-type: none"> 1. O animal está cadastrado
<p>Fluxo de eventos alternativo: Atualizar Anúncio (A1)</p> <ol style="list-style-type: none"> 1. O usuário clica no botão “Meus Anúncios” 2. O Sistema apresenta a tela de Anúncios do usuário 3. O usuário clica nas opções do <i>postcard</i> de um anúncio cadastrado 4. O sistema apresenta as opções 5. O usuário clica em “Editar” (A2) 6. O sistema apresenta a tela com os campos editáveis 7. O usuário altera as informações do anúncio 8. O usuário confirma qual o real status do anúncio (encontrado, perdido ou para adoção) 9. O sistema altera no banco de dados 10. O sistema informa ao usuário que as informações foram alteradas com sucesso 11. Finalizar caso de uso
<p>Fluxo de eventos alternativo: Remover Anúncio (A2)</p> <ol style="list-style-type: none"> 5. O usuário clica no botão "Remover" 6. O Sistema apresenta um aviso pedindo uma confirmação do usuário 7. O usuário confirma a remoção do Anúncio 8. O Sistema remove o anúncio do banco de dados 9. O Sistema informa ao usuário que o anúncio foi removido com sucesso 10. O Sistema apresenta a tela de Anúncios do usuário atualizada 11. Finalizar caso de uso

Quadro 5 - CSU: Buscar Animal

Caso de uso: Buscar Animal
ID: CSU003
Resumo: Este caso de uso descreve as etapas do processo em que os dados referentes a animais cadastrados são cruzados com os dados inseridos pelo usuário.
Atores Primários: usuário
Atores Secundários:
Pré-Condições: <ol style="list-style-type: none">1. O usuário estar logado2. O usuário estar na tela de Busca
Fluxo de eventos principal: <ol style="list-style-type: none">1. O usuário inicia a busca inserindo as informações do animal (A1)2. O sistema compara com as características dos animais cadastrados no banco de dados3. O sistema cria uma lista com anúncios de animais que possuem dados semelhantes (A2)4. O sistema disponibiliza a lista ao usuário5. Finaliza caso de uso.
Pós-Condições: <ol style="list-style-type: none">1. O usuário tem acesso a uma lista de animais compatíveis com as informações fornecidas.
Fluxo de eventos alternativo 1: Erro ao conectar com o servidor (A1) <ol style="list-style-type: none">2. O sistema não consegue se conectar com o servidor3. O sistema informa ao usuário que a conexão precisa ser reestabelecida4. Finaliza caso de uso.
Fluxo de eventos alternativo 2: Nenhum animal encontrado (A2) <ol style="list-style-type: none">3. O sistema não encontra nenhum animal com as características informadas para formar uma lista4. O sistema informa ao usuário5. Finaliza caso de uso.

FONTE: Autoria Própria, 2019

Quadro 6 - CSU: Analisar Proximidade

Caso de uso: Analisar Proximidade
ID: CSU004
Resumo: Este caso de uso descreve as etapas do processo em que o usuário terá acesso a animais desaparecidos ou encontrados próximos a sua localização atual.
Atores Primários: usuário
Atores Secundários:
Pré-Condições: <ol style="list-style-type: none">1. O usuário estar logado2. O dispositivo possuir conexão com a rede
Fluxo de eventos principal: <ol style="list-style-type: none">1. O usuário clica no botão “Mapa”2. O sistema apresenta a tela de Mapa para o usuário com as marcações dos anúncios registrados3. O usuário clica no botão “Proximidade”4. O sistema verifica a localização do usuário (A1)5. O sistema compara a localização do usuário e as localizações dos anúncios6. O sistema cria uma lista com os anúncios com uma distância de até 300 metros7. O sistema coloca no mapa os pins dos anúncios listados8. O sistema verifica se houve alteração na localização do usuário para atualização dos pins9. O sistema retorna ao passo 4 se detectar alteração de localização10. Finaliza caso de uso.
Pós-Condições: <ol style="list-style-type: none">1. A relação dos anúncios próximos ao usuário é apresentada com pins no mapa
Fluxo de eventos alternativo: Sem sinal de GPS (A1) <ol style="list-style-type: none">5. O sistema não encontra o sinal do GPS6. O sistema informa ao usuário7. Finalizar caso de uso.

FONTE: Autoria Própria, 2019

Quadro 7 - CSU: Manter ONG

Caso de uso: Manter ONG
ID: CSU005
<p>Resumo:</p> <p>Este caso de uso descreve as etapas do processo de cadastro, atualização e exclusão de ONGs relacionadas a animais.</p>
Atores Primários: usuário.
Atores Secundários:
<p>Pré-Condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar logado
<p>Fluxo de eventos principal:</p> <ol style="list-style-type: none"> 1. O usuário clica na opção “ONG” no menu principal 2. O sistema apresenta a tela de ONGs para o usuário 3. O usuário clica no botão flutuante “+” (A1) 4. O sistema apresenta a tela de registro de ONGs para o usuário 5. O usuário preenche o formulário com as informações da ONG 6. O usuário clica no botão “Registrar” 7. O sistema verifica a existência do registro (A3) 8. O sistema registra no banco de dados e informa ao usuário que o registro foi efetuado com sucesso 9. Finalizar caso de uso
<p>Pós-Condições:</p> <ol style="list-style-type: none"> 1. A ONG é registrada e suas informações são disponibilizadas
<p>Fluxo de eventos alternativo: Alterar informações de ONG (A1)</p> <ol style="list-style-type: none"> 3. O usuário clica no menu de opções de uma ONG que ele cadastrou 4. O sistema apresenta as opções disponíveis 5. O usuário clica em “Editar” (A2) 6. O sistema apresenta a tela de alteração com os campos habilitados para edição 7. O usuário clica em “Atualizar” 8. Caso de uso volta para o passo 7 do fluxo principal 9. Finalizar caso de uso
<p>Fluxo de eventos alternativo: Remover ONG (A2)</p> <ol style="list-style-type: none"> 5. O usuário clica em "Remover" 6. O Sistema apresenta um aviso pedindo uma confirmação do usuário 7. O usuário confirma a remoção da ONG 8. O Sistema remove a ONG do banco de dados 9. O Sistema informa ao usuário que a ONG foi removida com sucesso 10. O Sistema apresenta a tela de ONGs do usuário atualizada 11. Finalizar caso de uso
<p>Fluxo de eventos alternativo: ONG já cadastrada (A3)</p> <ol style="list-style-type: none"> 7. O sistema informa o usuário que já existe uma ONG registrada com as informações fornecidas 8. Finalizar caso de uso

Quadro 8 - CSU: Enviar Mensagem

Caso de uso: Enviar Mensagem
ID: CSU006
Resumo: Este caso de uso descreve as etapas do processo onde um usuário pode enviar mensagens para outros usuários via chat.
Atores Primários: usuário.
Atores Secundários:
Pré-Condições: <ol style="list-style-type: none">1. O usuário deve estar logado2. Um dos usuários deve possuir um anúncio cadastrado
Fluxo de eventos principal: <ol style="list-style-type: none">1. O usuário clica no ícone de chat (A1)2. O sistema abre um chat com os dois usuários3. O usuário envia uma mensagem4. O sistema registra a mensagem no banco de dados5. O sistema apresenta a mensagem na tela de chat6. Finalizar caso de uso
Pós-Condições: <ol style="list-style-type: none">1. A mensagem do usuário é enviada.
Fluxo de eventos alternativo: Falha ao conectar com o servidor (A1) <ol style="list-style-type: none">2. O sistema informa que ocorreu um erro de conexão3. Finalizar caso de uso

FONTE: Autoria Própria, 2019

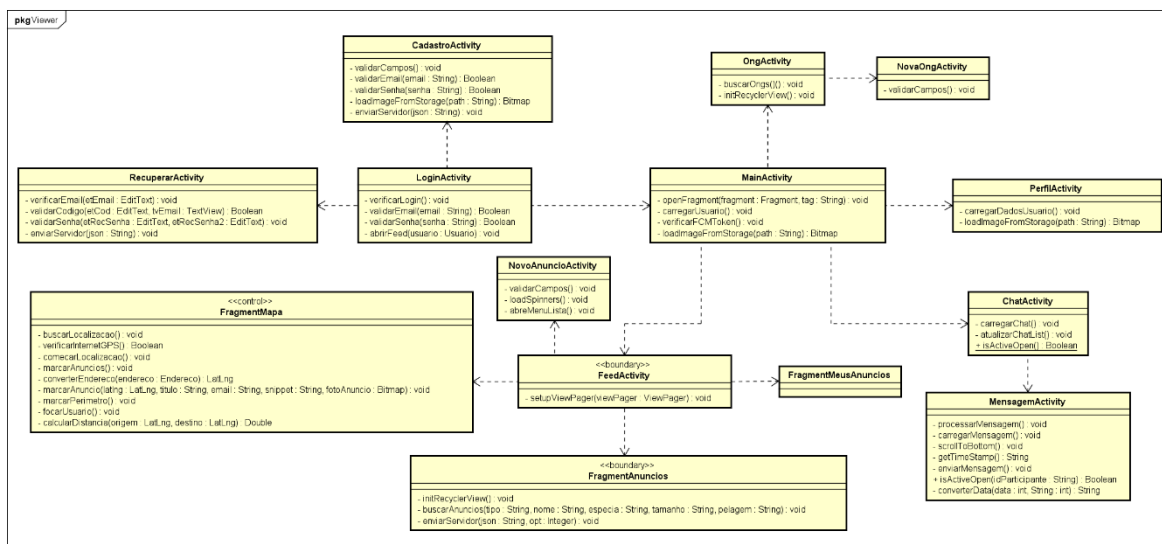
3. ANÁLISE E PROJETO

Ao iniciar um projeto é necessário realizar uma análise para que seja discutido e definido que caminho ele vai seguir, ao fazer a elaboração da análise as surpresas na hora da execução são evitadas.

A UML, Linguagem de Modelagem Unificada, em português, é uma linguagem visual onde podemos enxergar como o projeto deve ser elaborado. Ao realizarmos a confecção de diagramas UML neste projeto vemos os diagramas de classes, de sequencias, de atividade, de transições de estados, de componentes e de implantação, e as modelagens realizadas para o banco de dados onde vemos o modelo de entidade e relacionamento e o modelo lógico.

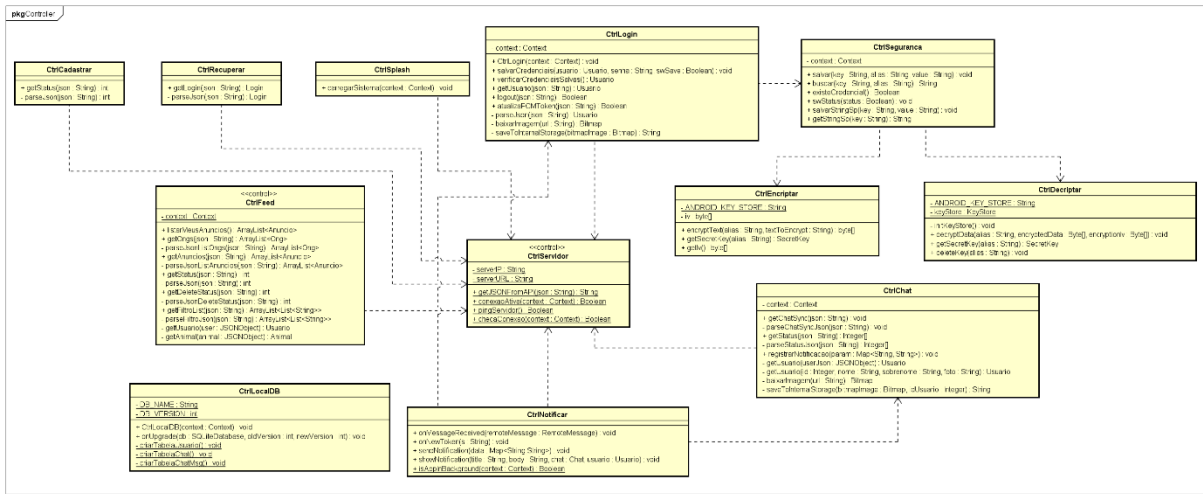
3.1. Diagrama de Classes

Figura 2 - Diagrama de Classes: Pacote *Viewer*



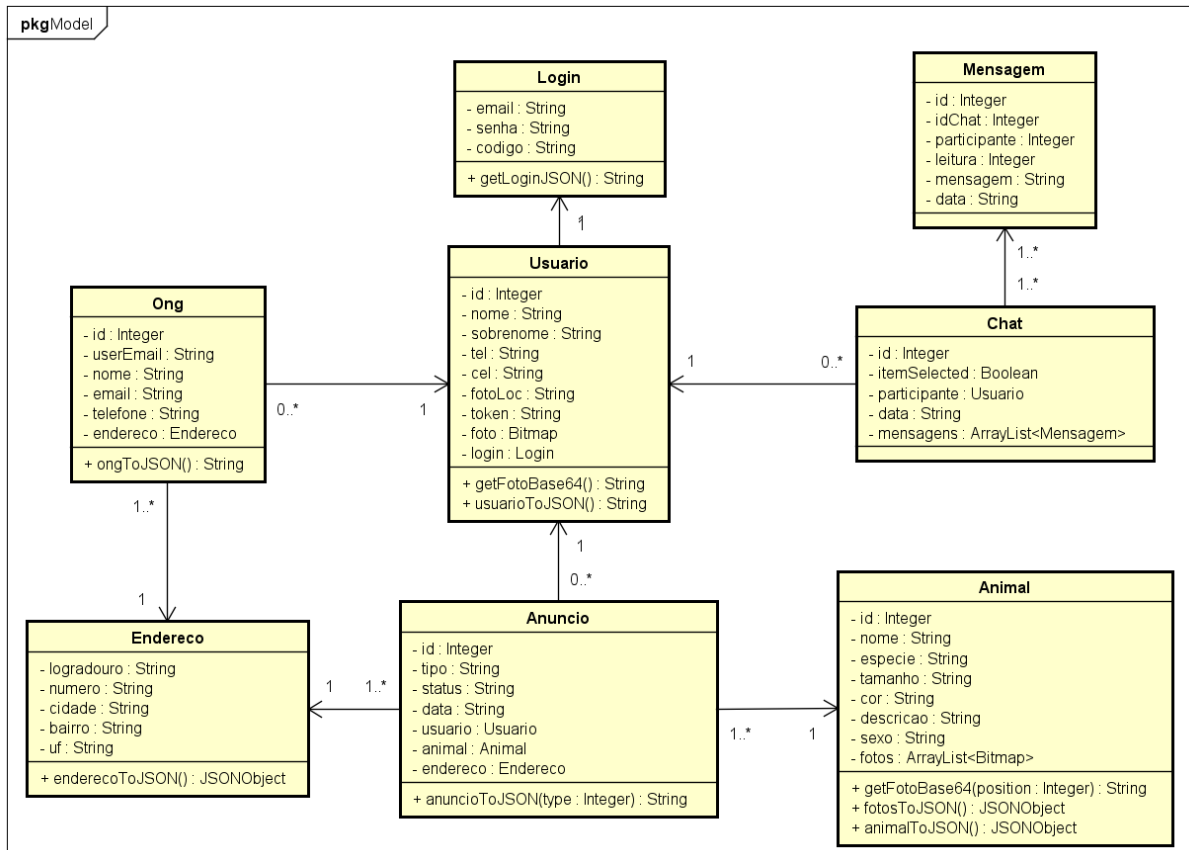
FONTE: Autoria Própria, 2020

Figura 3 - Diagrama de Classes: Pacote Controller



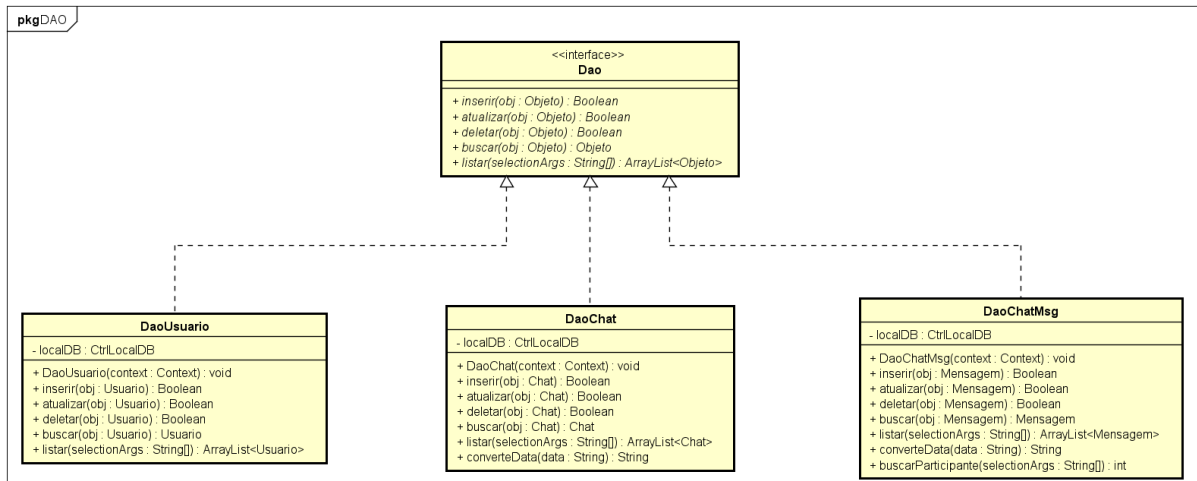
FONTE: Autoria Própria, 2020

Figura 4 - Diagrama de Classes: Pacote Model



FONTE: Autoria Própria, 2020

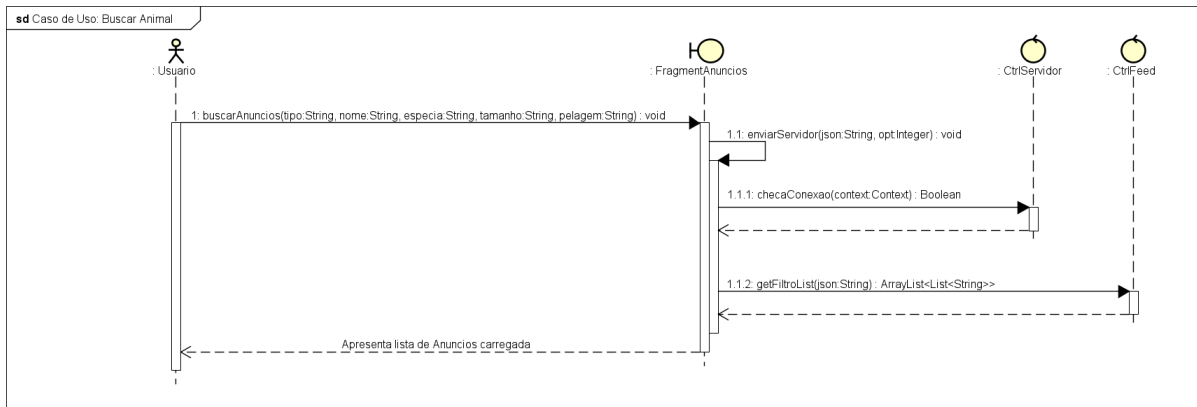
Figura 5 - Diagrama de Classes: Pacote DAO



FONTE: Autoria Própria, 2020

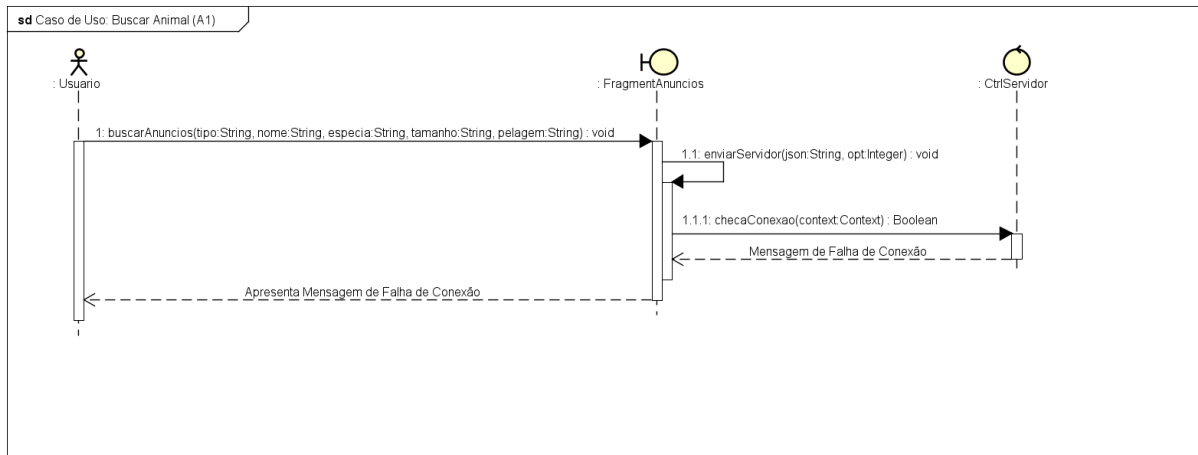
3.2. Diagrama de Sequência e de Comunicação

Figura 6 - Diagrama de Sequência: Buscar Animal (Fluxo Principal)



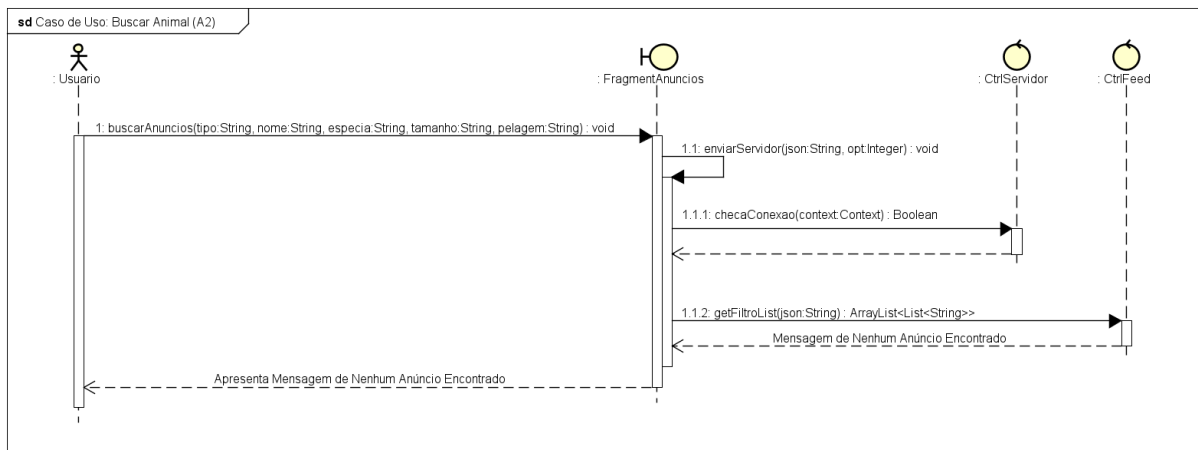
FONTE: Autoria Própria, 2020

Figura 7 - Diagrama de Sequência: Buscar Animal (Fluxo Alternativo 1)



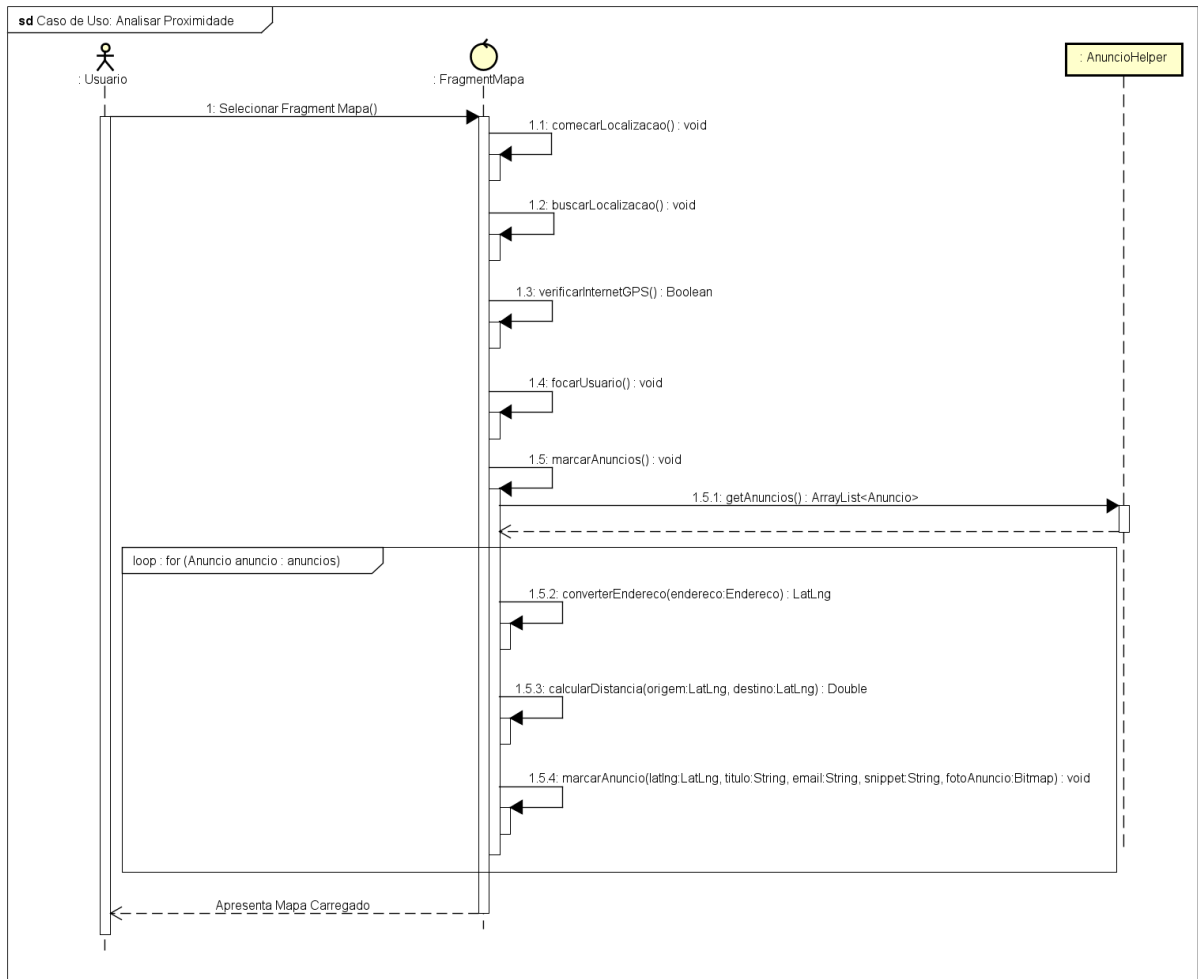
FONTE: Autoria Própria, 2020

Figura 8 - Diagrama de Sequência: Buscar Animal (Fluxo Alternativo 2)



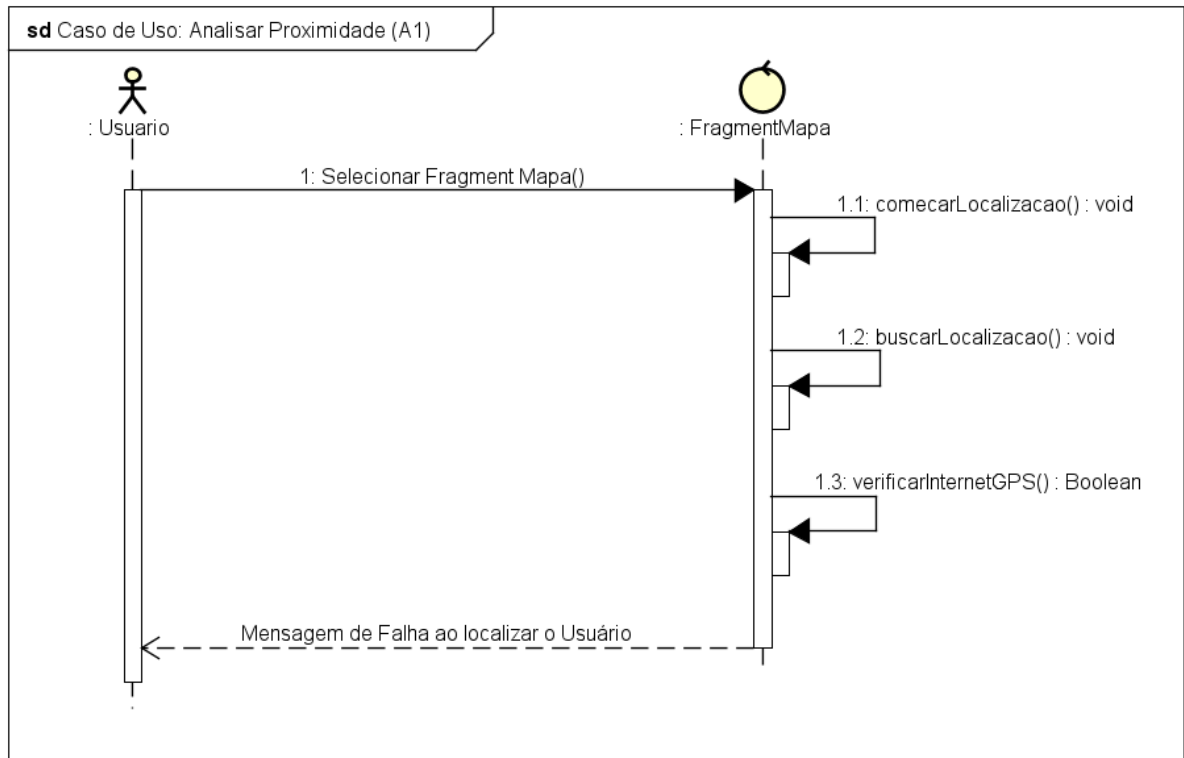
FONTE: Autoria Própria, 2020

Figura 9 - Diagrama de Sequência: Analisar Proximidade (Fluxo Principal)



FONTE: Autoria Própria, 2020

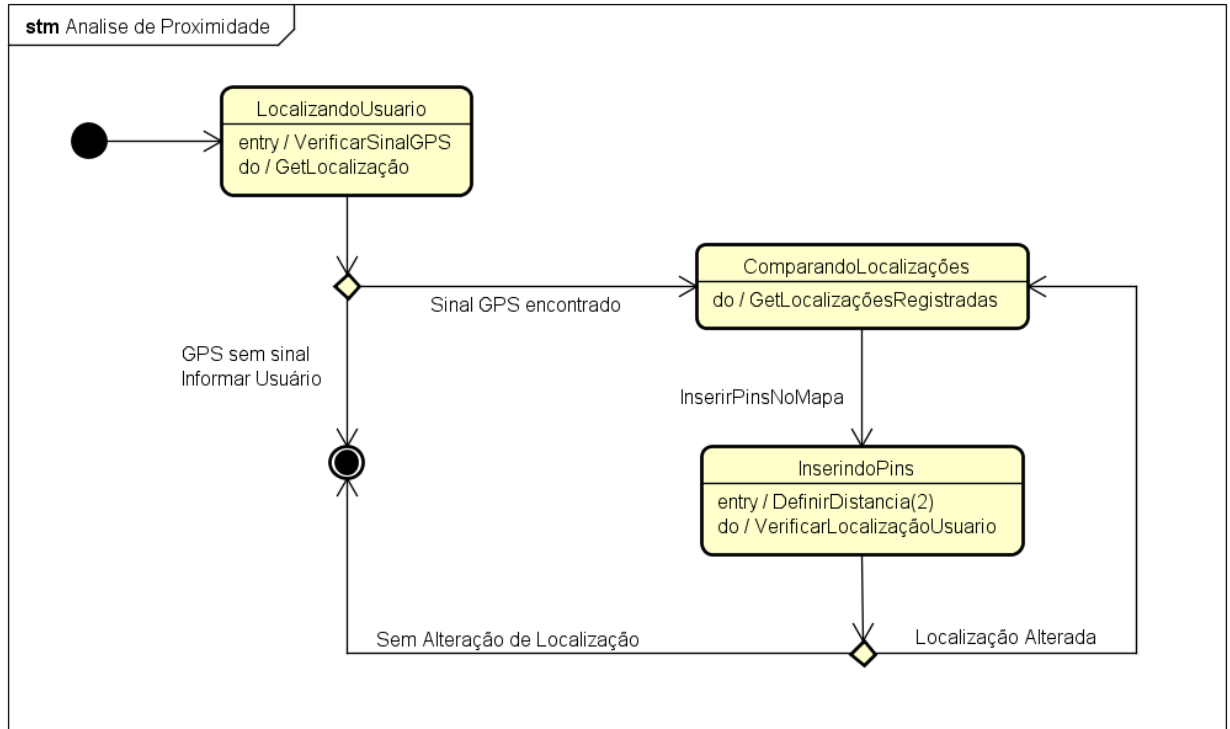
Figura 10 - Diagrama de Sequência: Analisar Proximidade (Fluxo Alternativo)



FONTE: Autoria Própria, 2020

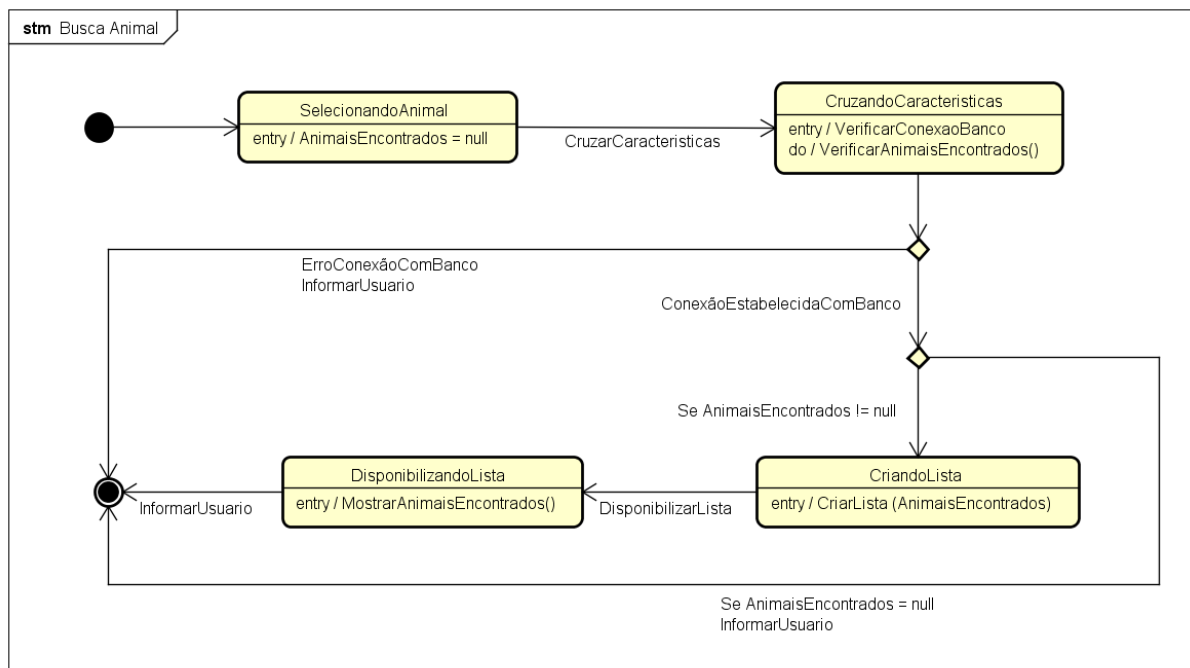
3.3. Diagrama de Transições de Estados

Figura 11 - Diagrama de Transição de Estado: Análise de Proximidade



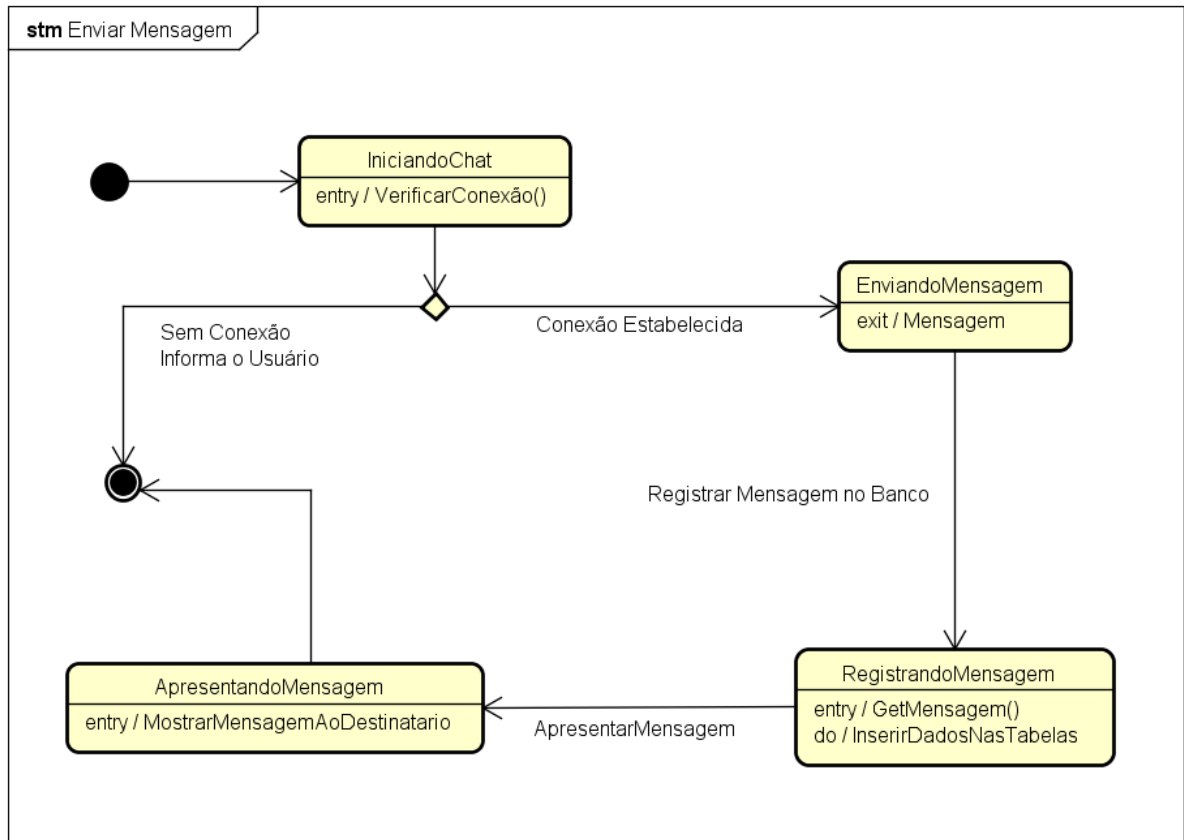
FONTE: Autoria Própria, 2019

Figura 12 - Diagrama de Transição de Estado: Busca Animal



FONTE: Autoria Própria, 2019

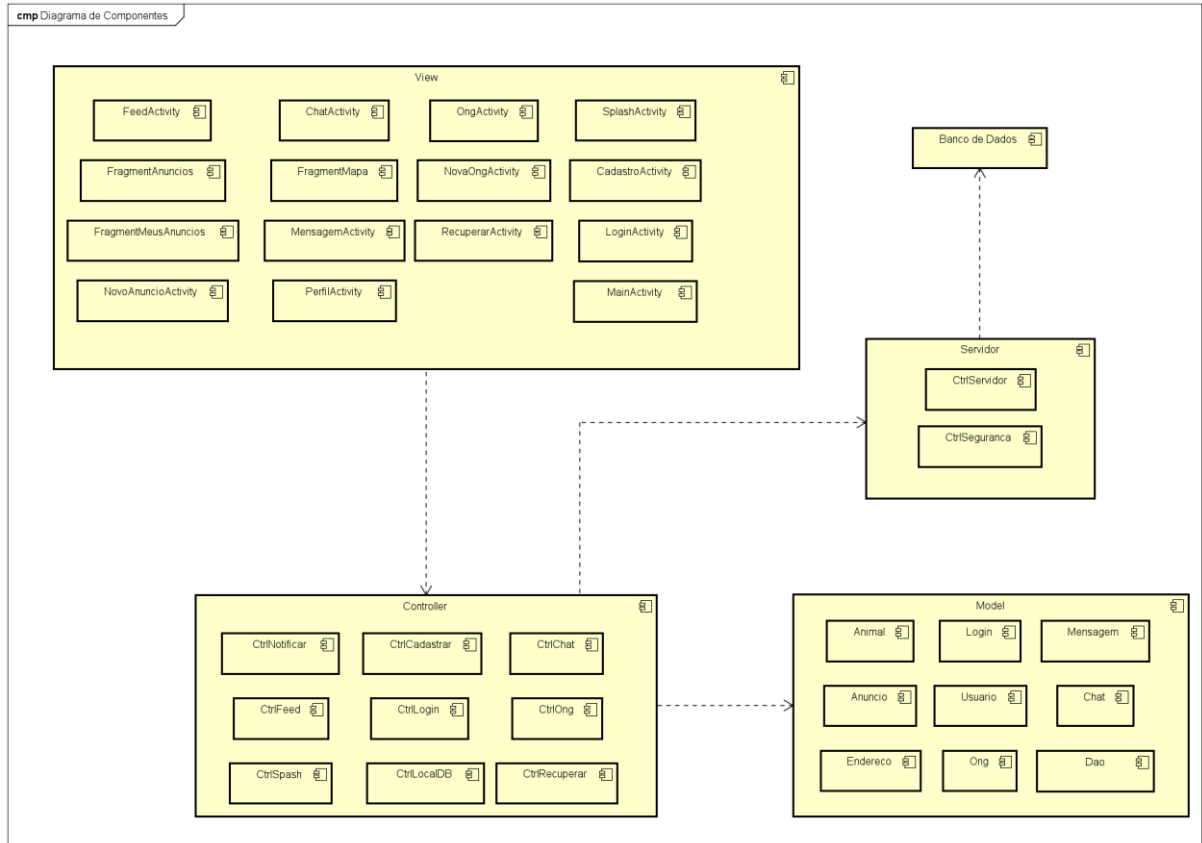
Figura 13 - Diagrama de Transição de Estado: Enviar Mensagem



FONTE: Autoria Própria, 2019

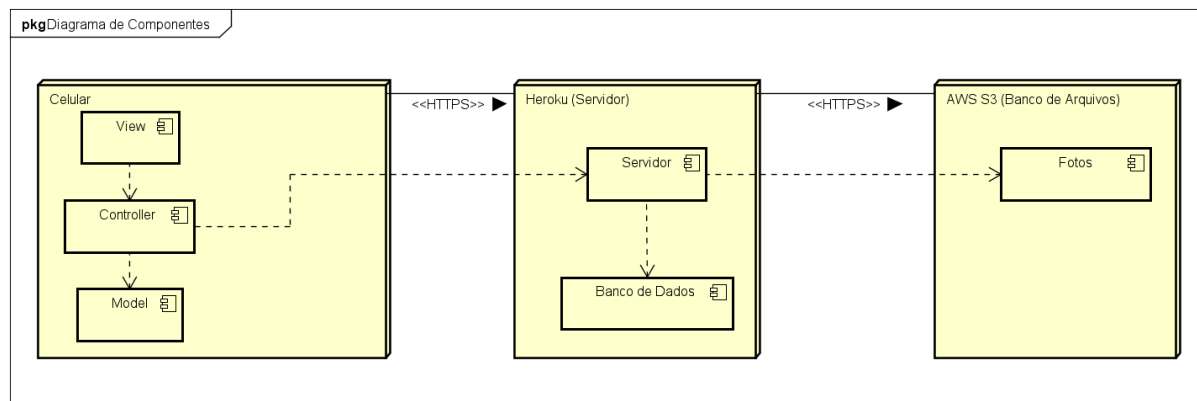
3.4. Diagrama de Componentes e de Implantação

Figura 14 - Diagrama de Componentes



FONTE: Autoria Própria, 2020

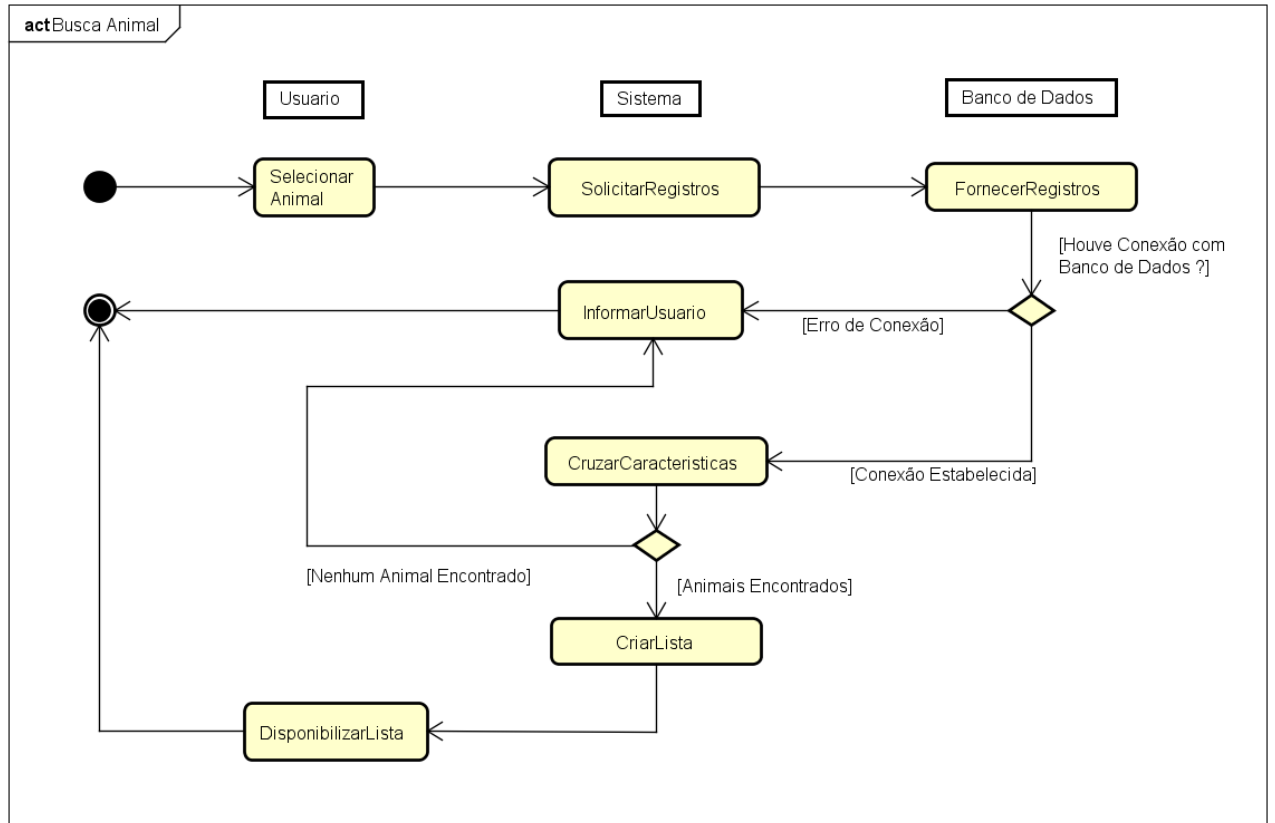
Figura 15 - Diagrama de Implantação



FONTE: Autoria Própria, 2020

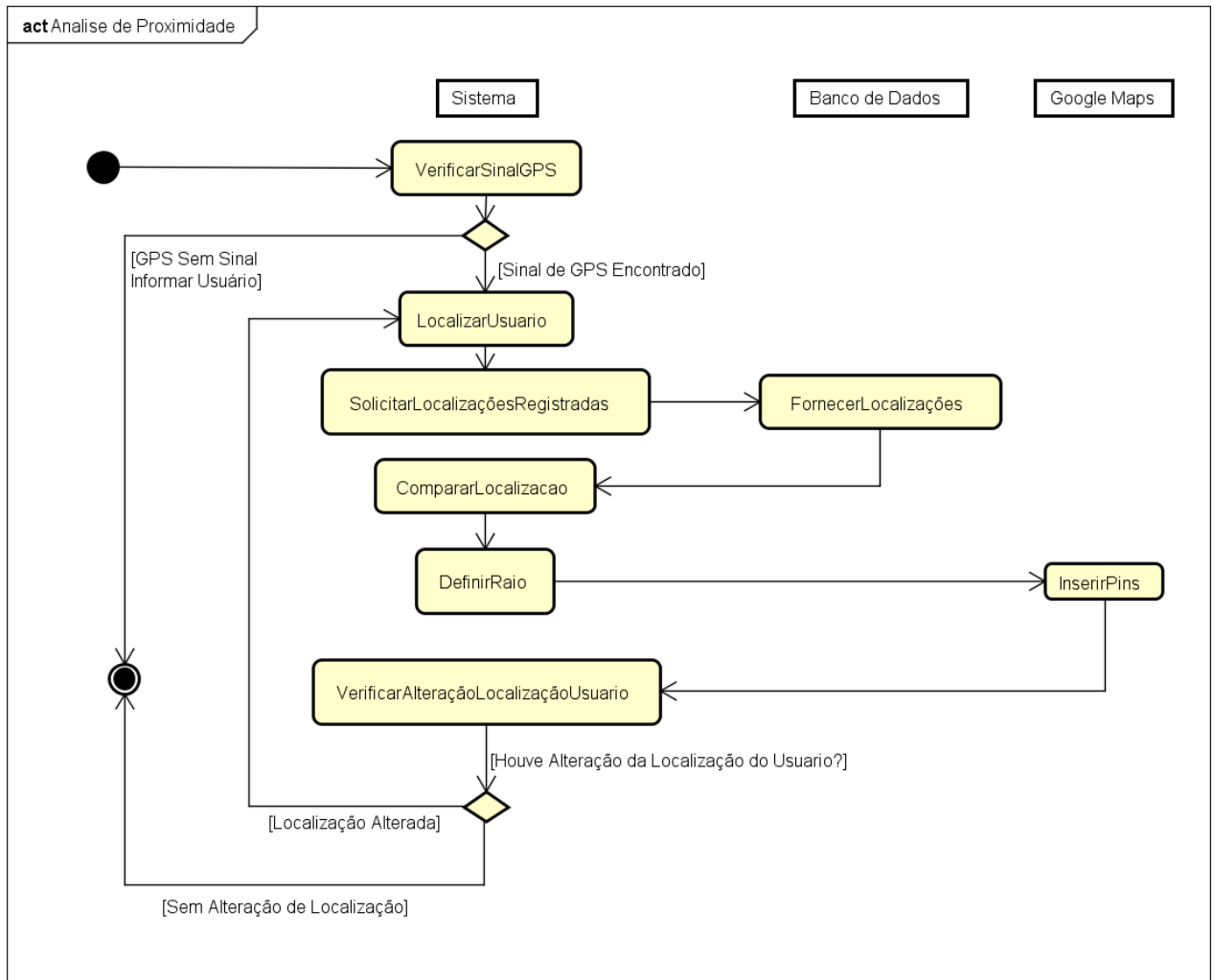
3.5. Diagrama de Atividades

Figura 16 - Diagrama de Atividades: Busca Animal



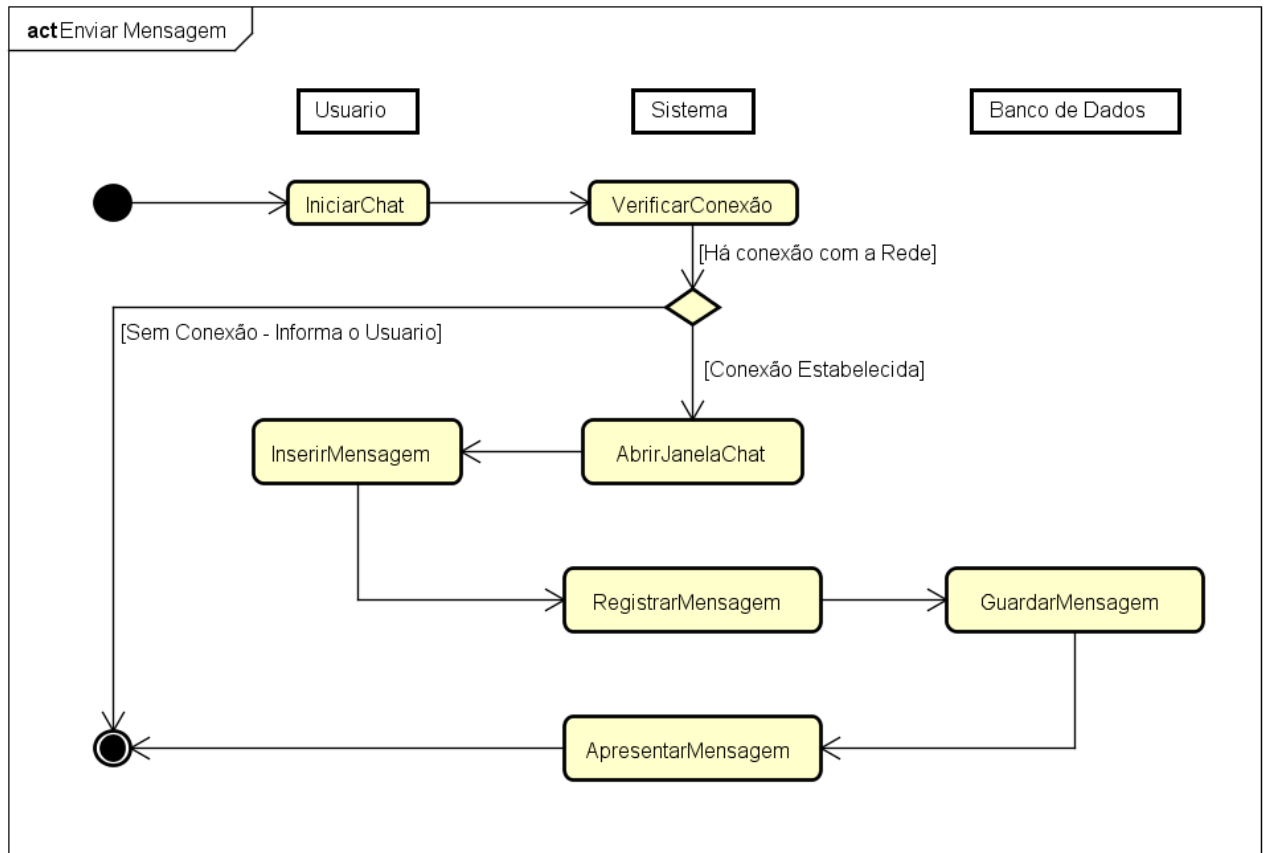
FONTE: Autoria Própria, 2019

Figura 17 - Diagrama de Atividades: Análise de Proximidade



FONTE: Autoria Própria, 2019

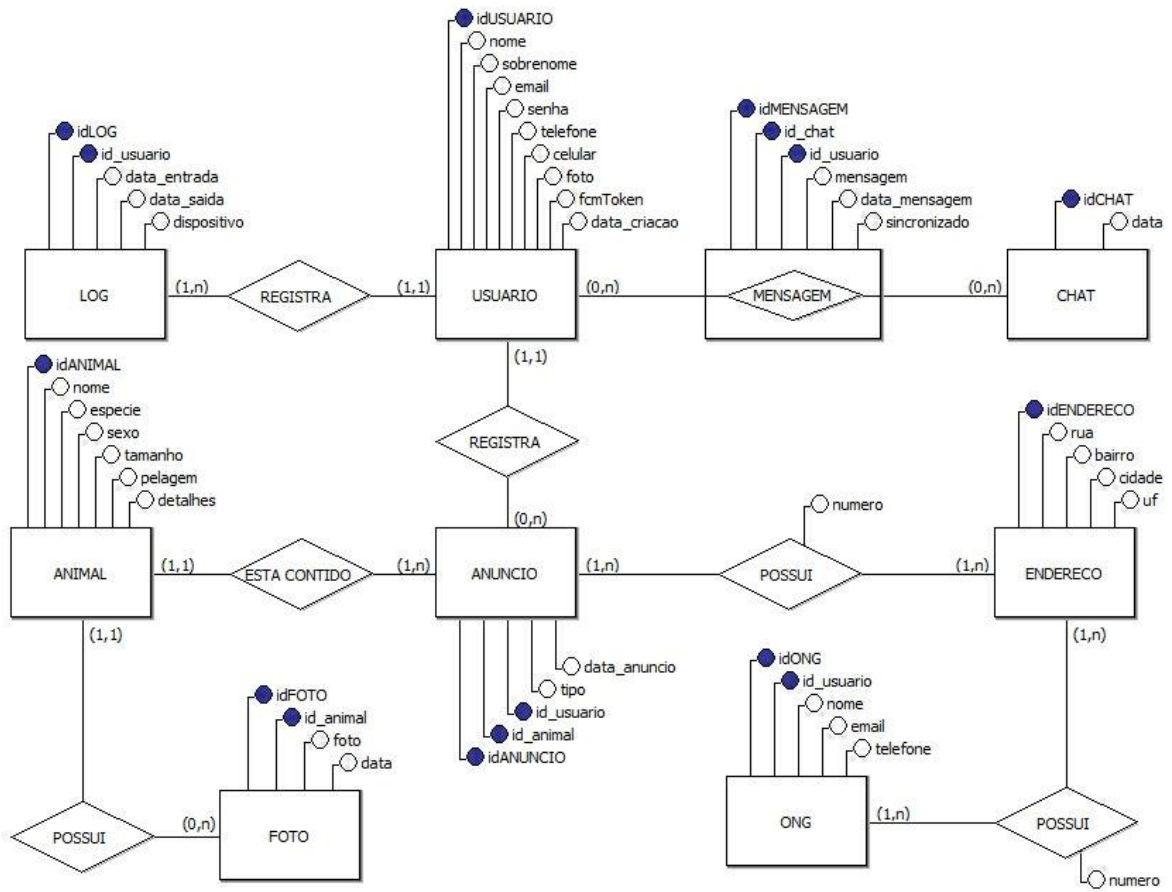
Figura 18 - Diagrama de Atividades: Enviar Mensagem



FONTE: Aatoria Própria, 2019

3.6. Modelo Entidade – Relacionamento

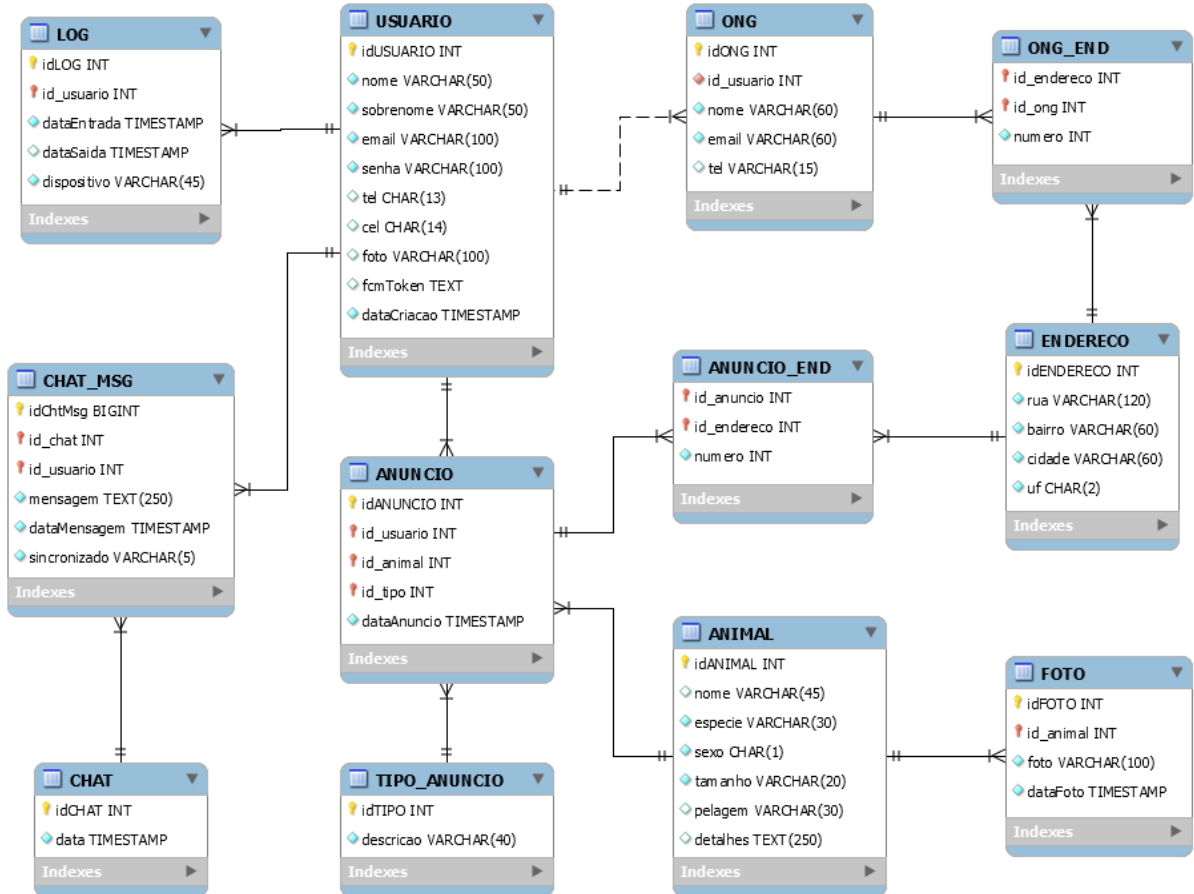
Figura 19 - Modelo Entidade-Relacionamento



FONTE: Autoria Própria, 2020

3.7. Modelo Lógico do Banco de Dados

Figura 20 - Modelo Lógico do Banco de Dados



FONTE: Autoria Própria, 2020

3.8. Pequeno Dicionário de Dados

Quadro 9 - Dicionário de Dados - Tabela usuário

USUÁRIO				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idUSUARIO	INT	-	SIM	-
nome	VARCHAR	50	-	-
sobrenome	VARCHAR	50	-	-
email	VARCHAR	100	-	-
senha	VARCHAR	45	-	-
Tel	CHAR	13	-	-
cel	CHAR	14	-	-
foto	VARCHAR	100	-	-
fcmToken	TEXT	-	-	-
data_criacao	TIMESTAMP	-	-	-

FONTE: Autoria Própria, 2019

Quadro 10 - Dicionário de Dados - Tabela Log

LOG				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idLOG	INT	-	SIM	-
id_usuario	INT	-	-	-
dataEntrada	TIMESTAMP	-	-	-
dataSaida	TIMESTAMP	-	-	-
dispositivo	VARCHAR	45	-	-

FONTE: Autoria Própria, 2019

Quadro 11 - Dicionário de Dados - Tabela Chat Msg

CHAT_MSG				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idChtMsg	BIGINT	-	SIM	-
id_usuario	INT	-	-	Tabela USUARIO
id_chat	INT	-	-	Tabela CHAT
mensagem	TEXT	-	-	-
data_mensagem	TIMESTAMP	-	-	-
sincronizado	VARCHAR	5	-	-

FONTE: Autoria Própria, 2019

Quadro 12 - Dicionário de Dados - Tabela Chat

CHAT				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idCHAT	INT	-	SIM	-
data	TIMESTAMP	-	-	-

FONTE: Aatoria Própria, 2019

Quadro 13 - Dicionário de Dados - Tabela Foto

FOTO				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idFOTO	INT	-	SIM	-
id_animal	INT	-	-	-
dataFoto	DATETIME	-	-	-
foto	VARCHAR	100	-	-

FONTE: Aatoria Própria, 2019

Quadro 14 - Dicionário de Dados - Tabela Animal

ANIMAL				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
IdANIMAL	INT	-	SIM	-
nome	VARCHAR	45	-	-
especie	VARCHAR	20	-	-
sexo	CHAR	1	-	-
tamanho	VARCHAR	20	-	-
pelagem	CHAR	30	-	-
detalhes	TEXT	-	-	-

FONTE: Aatoria Própria, 2019

Quadro 15 - Dicionário de Dados - Tabela Anúncio

ANUNCIO				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idANUNCIO	INT	-	SIM	-
id_usuario	INT	-	-	Tabela USUARIO
id_animal	INT	-	-	Tabela ANIMAL
id_tipo	INT	-	-	Tabela TIPO_ANUNCIO
dataAnuncio	DATETIME	-	-	-

FONTE: Aatoria Própria, 2019

Quadro 16 - Dicionário de Dados - Tabela Tipo Anúncio

TIPO_ANUNCIO				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idTIPO	INT	-	SIM	-
descrição	VARCHAR	40	-	-

FONTE: Autoria Própria, 2019

Quadro 17 - Dicionário de Dados - Tabela Anúncio End

ANUNCIO_END				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
id_anuncio	INT	-	-	Tabela ONG
id_endereco	INT	-	-	-Tabela ENDERECO
numero	INT	-	-	-

FONTE: Autoria Própria, 2019

Quadro 18 - Dicionário de Dados - Tabela ONG

ONG				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idOng	INT	-	SIM	-
nome	VARCHAR	60	-	-
email	VARCHAR	60	-	-
tel	VARCHAR	15	-	-
id_usuario	INT	-	-	Tabela USUARIO

FONTE: Autoria Própria, 2019

Quadro 19 - Dicionário de Dados - Tabela ONG End

ONG_END				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
id_ong	INT	-	-	Tabela ONG
id_endereco	INT	-	-	-Tabela ENDERECO
numero	INT	-	-	-

FONTE: Autoria Própria, 2019

Quadro 20 - Dicionário de Dados - Tabela Endereço

ENDERECO				
ATRIBUTO	TIPO	TAMANHO	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
idENDERECO	INT	-	SIM	-
rua	VARCHAR	120	-	-
bairro	VARCHAR	60	-	-
cidade	VARCHAR	60	-	-
uf	CHAR	2	-	-

FONTE: Autoria Própria, 2019

4. SOFTWARE E TECNOLOGIAS EMPREGADAS NA PROGRAMAÇÃO DO SISTEMA

As tecnologias agregadas durante e após a programação deste sistema foram separadas nos tópicos abaixo:

4.1 *Cloud Computing* (computação em nuvem)

Segundo TAURION (2009, p.2), para alguns, *Cloud Computing* é apenas um nome novo para iniciativas já feitas no passado, como o *outsourcing* (obter recursos computacionais de terceiros) e *Grid Computing*, que é uma rede de computadores ligados por baixo acoplamento. Para outros, a Computação em nuvem é uma evolução natural da convergência de várias tecnologias e conceitos, como o próprio *Grid*, mais o conceito de *Utility Computing* (que são serviços computacionais comercializados como os serviços utilitários, como energia elétrica), virtualização e *autonomic computing*, que são sistemas capazes de auto gerenciar e corrigir problemas e falhas, acrescidos de tecnologias e tendências como *Web 2.0*, *SOA (Service Oriented Architecture)* e o modelo de software como Serviço (*Software-as-a-Service*).

Podemos dizer que a Computação em nuvem é um termo para descrever um ambiente de computação baseado em uma imensa de servidores, seja estes virtuais ou físicos. O *Cloud Computing* pode ser definido como o fornecimento sob demanda de recursos como poder computacional, armazenamento de dados, conectividade, plataformas, aplicações e outros serviços através da *internet*, seguindo o princípio da computação em grade à uma definição de preço conforme uso. Estes recursos computacionais ficam hospedados em *datacenters* distribuídos ao redor do mundo.

4.2. *Web Service*

Ricardo Lecheta em sua obra sobre *webservices* (2015, p.19), diz que *webservices* são utilizados como forma de integração e comunicação de sistemas, de modo que um sistema possa realizar uma chamada para um serviço de outro sistema a fim de obter informações. Estas chamadas podem enviar e receber informações em diversos formatos, sendo que atualmente os mais populares são *XML* e *JSON*. Uma das grandes vantagens na construção de *webservices* é que eles permitem acessar os serviços de uma forma padronizada e independente de linguagem de programação. Este conceito trouxe um grande avanço na forma que os sistemas são construídos, pois permite separar e quebrar o sistema em partes.

Existem várias formas de criar webservices, mas podemos dizer que duas das mais conhecidas são webservices em *SOAP* ou *REST*.

O webservice pode ser compreendido como um “serviço” correspondente a um componente de *software* acessível através de um endereço na *Web*. Desta forma, tanto o consumidor quanto o fornecedor do serviço utilizam mensagens com formato próprio para enviar requisições e receber as respostas sem a necessidade de conhecer a tecnologia empregada em qualquer uma das pontas de comunicação.

Devido a estas características dos *webservices* é possível desenvolver sistemas que funcionem em ambientes heterogêneos, sem fazer distinção de tecnologias empregadas.

Segundo Lecheta (2015, p.105), o *webservice REST* é um estilo de arquitetura de *software* cada vez mais utilizado no mundo inteiro, principalmente para criar serviços web e auxiliar na integração de sistemas. O *REST* utiliza o protocolo *HTTP* para criar serviços que retornam dados geralmente nos formatos *XML* ou *JSON* e tem o objetivo de fornecer uma alternativa para o *SOAP* e o *WSDL*. o Termo *RESTful* foi criado para indicar que determinado sistema ou serviço segue os princípios do *REST*.

O *Web Service Restful* é um estilo de arquitetura mais adequado para utilização em cenários mais básicos, também são melhor adaptados ao uso protocolo *HTTP*. Outra característica dos serviços *Restful* é que são mais leves, permitindo que sejam desenvolvidos com menos esforço.

A aplicação envia solicitações para o *webservice* através de URL's que são interpretadas pelo *webservice* e respondidas.

4.3. *JSON*

Segundo Douglas Crockford (2006), *JSON* é uma formatação compacta e leve de troca de dados, de padrão aberto independente e muito fácil de analisar, que pode ser facilmente lido por seres humanos e fácil interpretação e criação para máquinas. *JSON* é um subconjunto de JavaScript.

O *JSON* está presente nas respostas enviadas pelo *Web Service* e é interpretada pela aplicação.

4.4. API Google Maps

Evangelos Petroustos (2014), afirma que uma API é um conjunto de funções, classes, métodos e padrões para serem utilizados em um software sem precisar entender detalhes da implementação desta API, mas apenas saber como utilizar seus serviços.

Uma API permite utilizar um ou mais conjuntos de rotinas e padrões de programação para acesso a recursos da aplicação Google Maps. Estes recursos foram utilizados na sessão de busca por proximidade na aplicação.

4.5. Android Studio

Segundo o Guia do Usuário do próprio Android Studio (2019), ele é o Ambiente de desenvolvimento integrado oficial para o desenvolvimento de *apps* Android e é baseado no *IntelliJ IDEA*. Além do editor de código e das ferramentas de desenvolvedor avançadas do *IntelliJ*, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de *apps* Android.

O Android Studio foi escolhido para o desenvolvimento deste projeto pois, além de ser um sistema operacional popular é o sistema que todos os integrantes possuem mais familiaridade para trabalhar. Os programas desenvolvidos na plataforma podem ser escritos com as linguagens Kotlin (incluída a partir da versão 3.0), Java e C++ usando o Android SDK, dentre as possíveis linguagens o grupo escolheu desenvolver com a linguagem nativa Java.

4.6. MySQL

Milani (2007, p.22) diz que o MySQL é um SGDB relacional, de licença dupla, que foi projetado para trabalhar com aplicações de pequeno e médio portes inicialmente, mas atualmente atende a aplicações de grande porte e com mais vantagens do que seus concorrentes. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados *open source* com maior capacidade para concorrer com os programas similares de código fechado, tais como SQL server e Oracle"

A escolha do MySQL como o banco de dados deste projeto aconteceu por todos os membros do grupo conhecerem essa ferramenta devido os ensinamentos do curso. O MySQL é um banco *free* para *webservice*.

4.7.SQLite

Segundo a documentação do próprio SQLite (2020), ele é uma biblioteca em linguagem C que implementa um pequeno, rápido, independente, de alta confiabilidade, com todos os recursos, mecanismo de banco de dados SQL. O projeto do SQLite foi iniciado em 09 de maio de 2000, desde então os desenvolvedores continuam expandindo seus recursos e aprimorando sua confiabilidade e desempenho. Seu código é de domínio público, portanto, é gratuito para qualquer finalidade, seja ela, comercial ou privada.

O SQLite não é diretamente comparável mecanismos de banco de dados cliente / servidor como MySQL, Oracle, SQL Server ou PostgreSQL, pois está tentando resolver um problema diferente. Seu propósito é atender algumas situações como, por exemplo, dispositivos incorporados e IoT, arquivamento de arquivos e / ou contêiner de dados, substituição de arquivos de disco ad hoc, banco de dados internos ou temporários, entre outros.

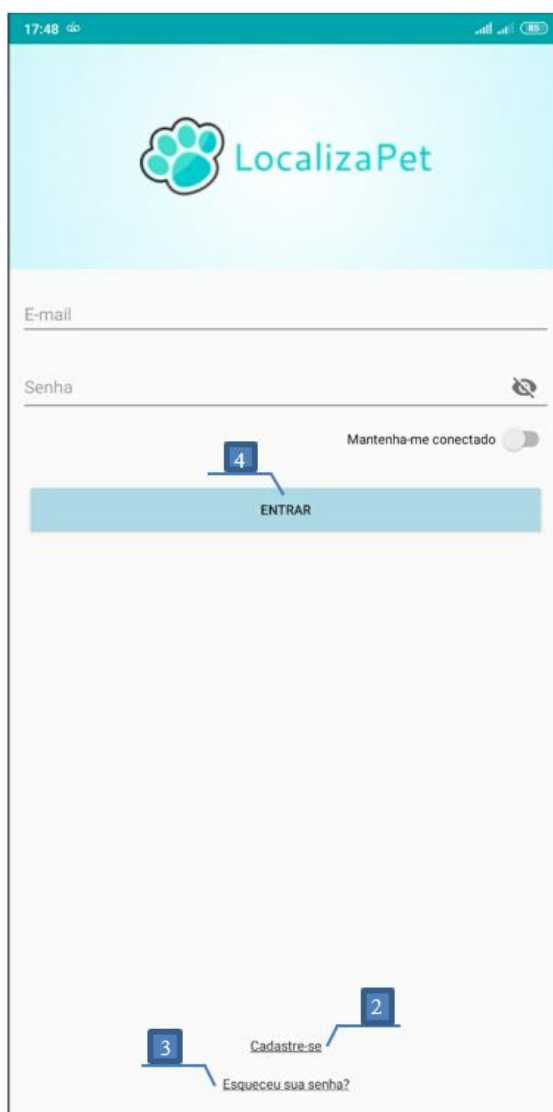
Optamos por utilizar o SQLite para o armazenamento de alguns dados que tornem a usabilidade da aplicação mais performática, como por exemplo, a busca de mensagens enviadas e recebidas no chat.

5. DETALHAMENTO DO PROTÓTIPO

5.1. Telas e Navegação

5.1.1. Tela Inicial (*Login*)

Figura 21 - Tela de Acesso (*Login*)



Legenda de Navegação

- 4 – Anúncios (*Feed*)
- 2 – Cadastro Usuário
- 3 – Recuperar Senha

FONTE: Autoria Própria, 2019

5.1.2. Tela de Cadastro de Usuário

Figura 22 - Tela de Registrar Usuário

The image displays two side-by-side screenshots of a mobile application's registration screen, titled "Cadastre-se".

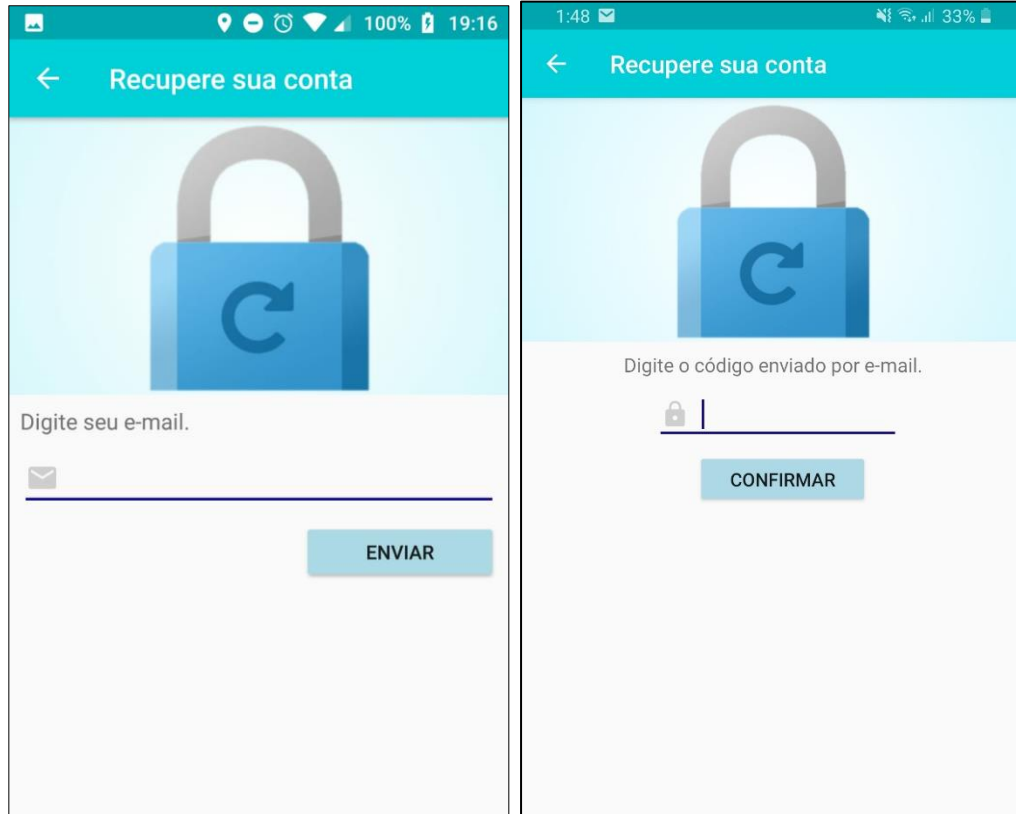
The left screenshot shows the top portion of the form. It features a large blue silhouette of a person's head and shoulders. To the right of the silhouette is a circular camera icon. Below the silhouette are two input fields: "Nome" and "Sobrenome".

The right screenshot shows the bottom portion of the form. It contains five more input fields: "Telefone", "Celular", "E-mail", "Senha", and "Confirmar senha". The "Senha" and "Confirmar senha" fields include an eye icon to toggle password visibility. At the bottom of this section is a large blue button labeled "CONFIRMAR".

FONTE: Autoria Própria, 2019

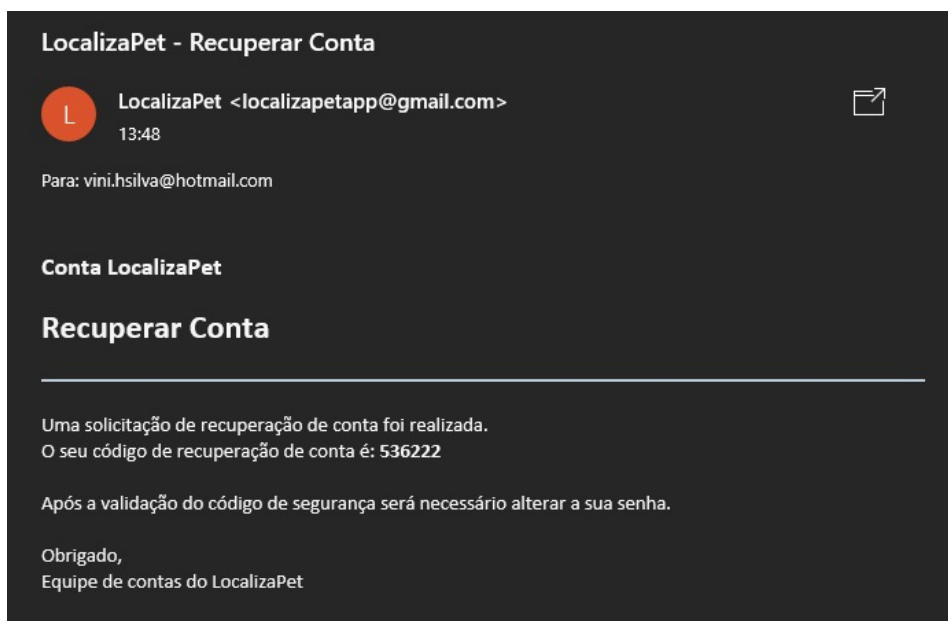
5.1.3. Tela de Recuperação de Senha

Figura 23 - Tela de Recuperação de Senha



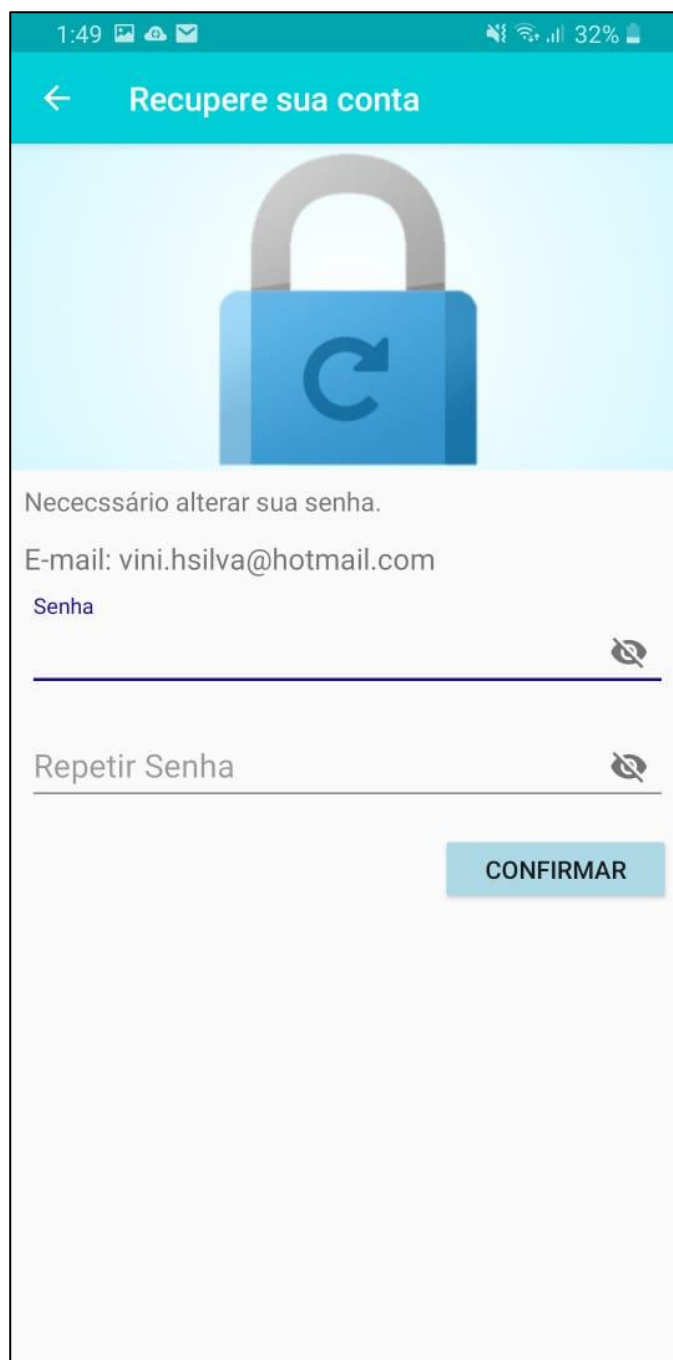
FONTE: Aatoria Própria, 2019

Figura 24 - Tela de Recuperação de Senha (E-mail)




FONTE: Aatoria Própria, 2019

Figura 25 - Tela de Redefinição de Senha




1:49 [status icons] 32%


← Recupere sua conta



Neccessário alterar sua senha.

E-mail: vini.hsilva@hotmail.com

Senha 

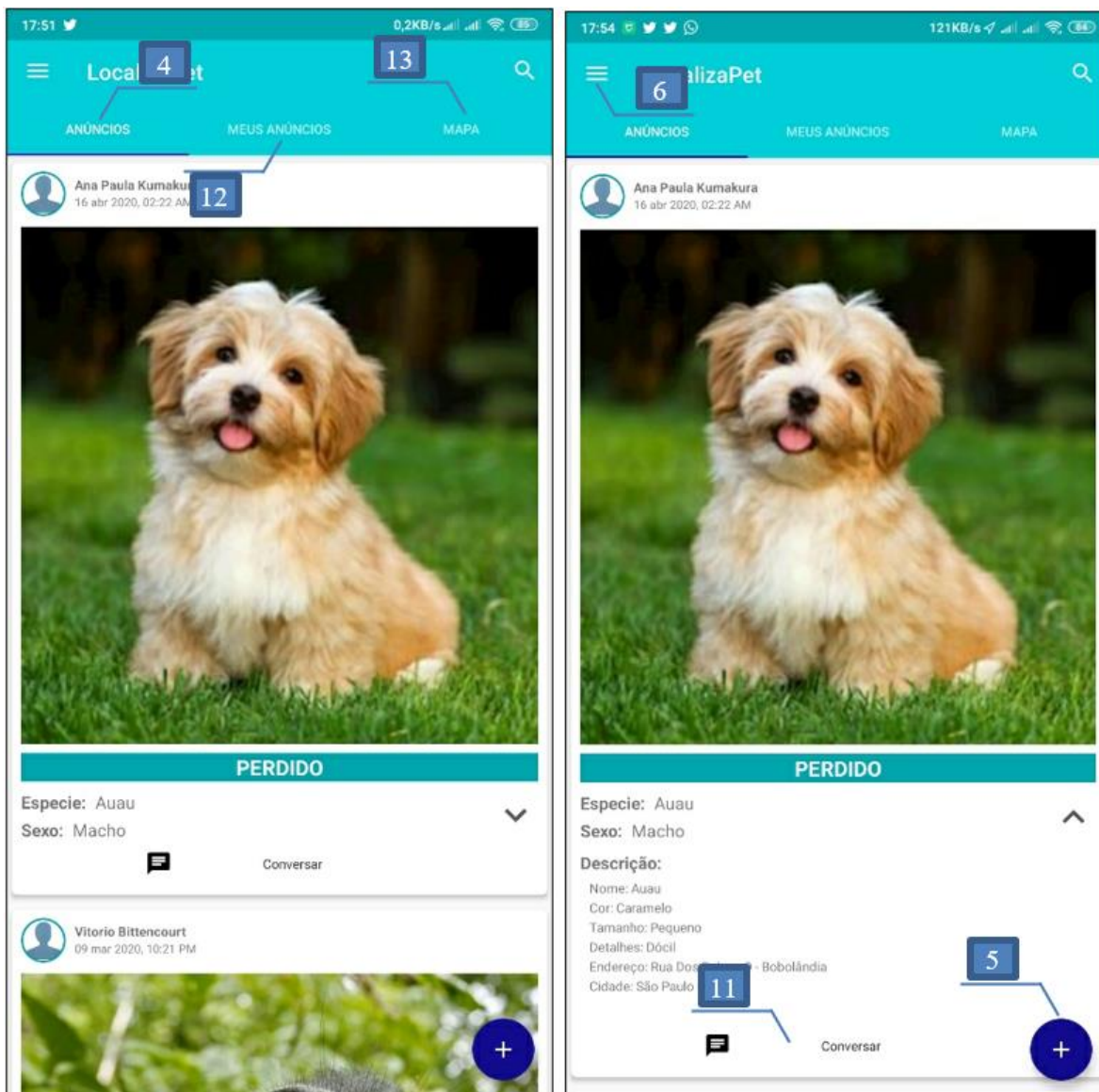
Repetir Senha 

CONFIRMAR

FONTE: Autoria Própria, 2019

5.1.4. Tela “Anúncios” (Feed)

Figura 26 - Tela de Anúncios



Legenda de Navegação

- 4 – Anúncios (Feed)
- 12 – Meus Anúncios
- 13 – Mapa com Anúncios
- 6 – Menu Principal
- 5 – Registrar Anúncio
- 11 – Conversa (Chat)

FONTE: Autoria Própria, 2019

5.1.5. Tela “Novo Anúncio”

Figura 27 - Tela de Registrar Anúncio

The image displays two side-by-side screenshots of a mobile application interface for creating a new advertisement. Both screens have a teal header with a back arrow and the text "Novo Anúncio".

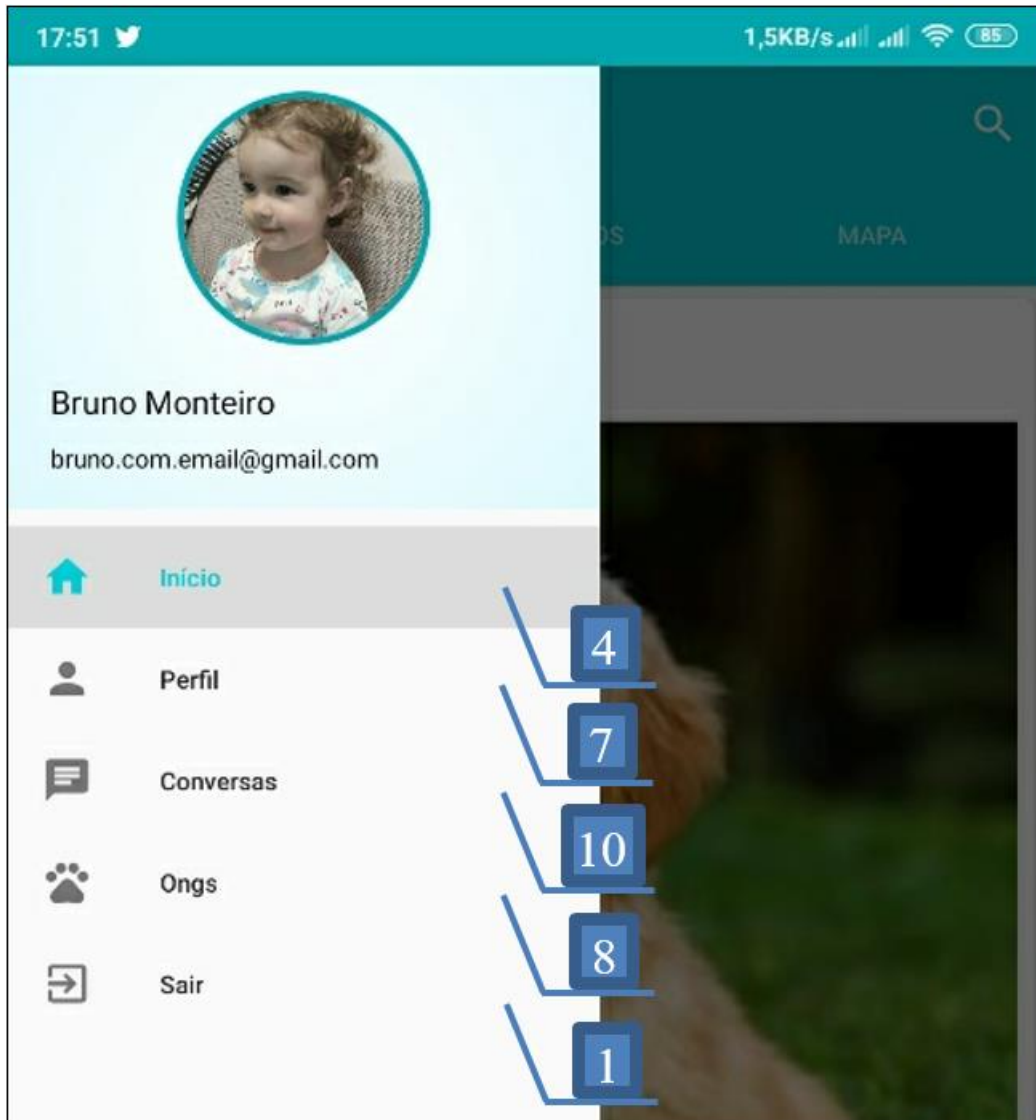
The left screenshot (timestamp 17:52) shows the top portion of the form. It features the LocalizaPet logo (a paw print) and a camera icon in the top right corner. Below the header, the form fields are partially visible: "Tipo do anúncio" (set to "Perdido"), "Nome", "Cor", "Espécie", "Características", "Sexo" (with "Macho" selected), and "Tamanho" (set to "Pequeno").

The right screenshot (timestamp 17:53) shows the full form. The fields are: "Tipo do anúncio" (Perdido), "Nome", "Cor", "Espécie", "Características", "Sexo" (Macho selected, Fêmea unselected), "Tamanho" (Pequeno), "Endereço", "Número", "Bairro", "Cidade", and "Estado". A teal button labeled "ANUNCIAR" is positioned at the bottom of the form.

FONTE: Autoria Própria, 2019

5.1.6. Menu Principal

Figura 28 - Menu Principal



Legenda de Navegação

4 – Anúncios (*Feed*)

7 – Perfil do Usuário

10 – Conversas do Usuário

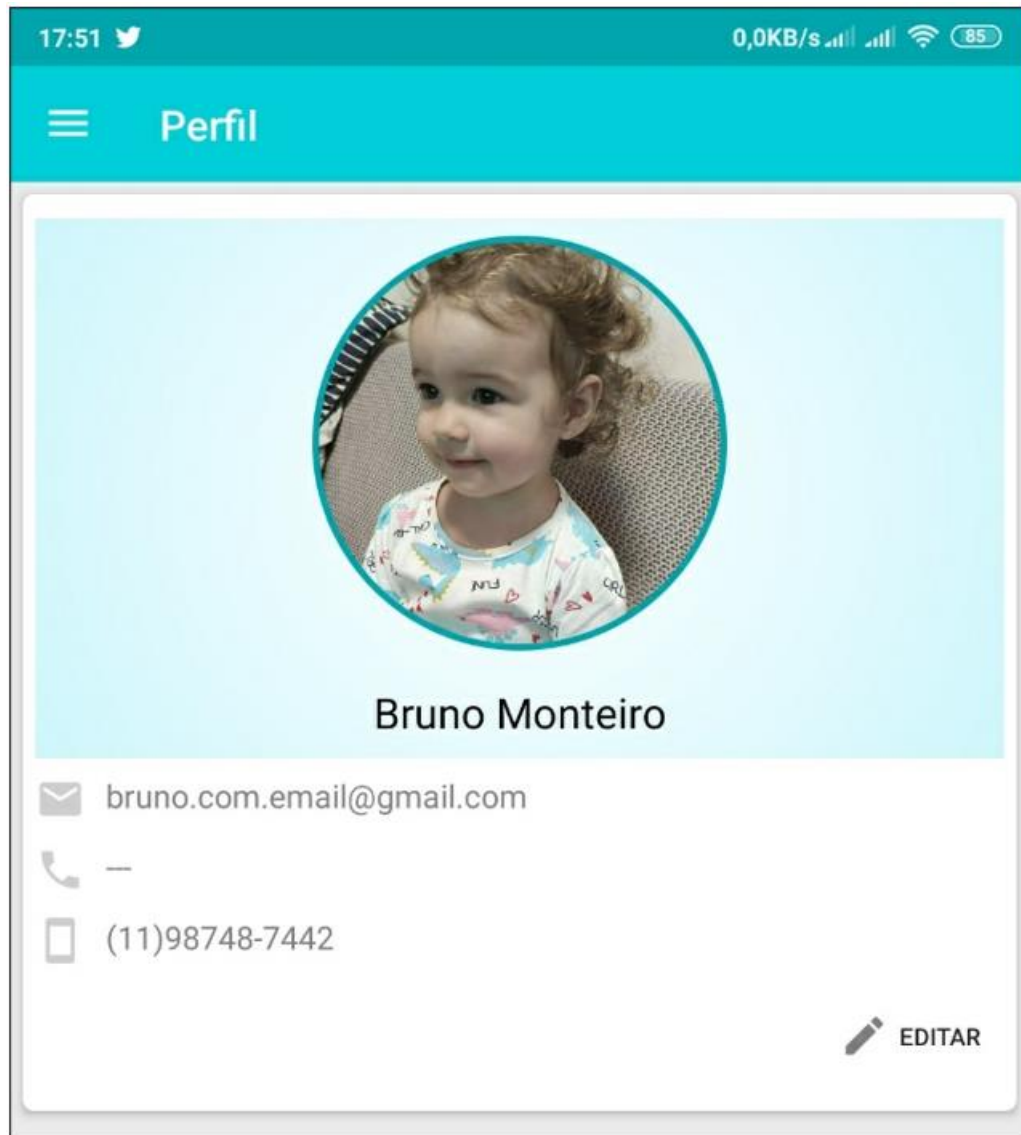
8 – ONGs (*Feed*)

1 – Retorno à tela de Login

FONTE: Autoria Própria, 2019

5.1.7. Tela “Perfil” (Perfil do Usuário)

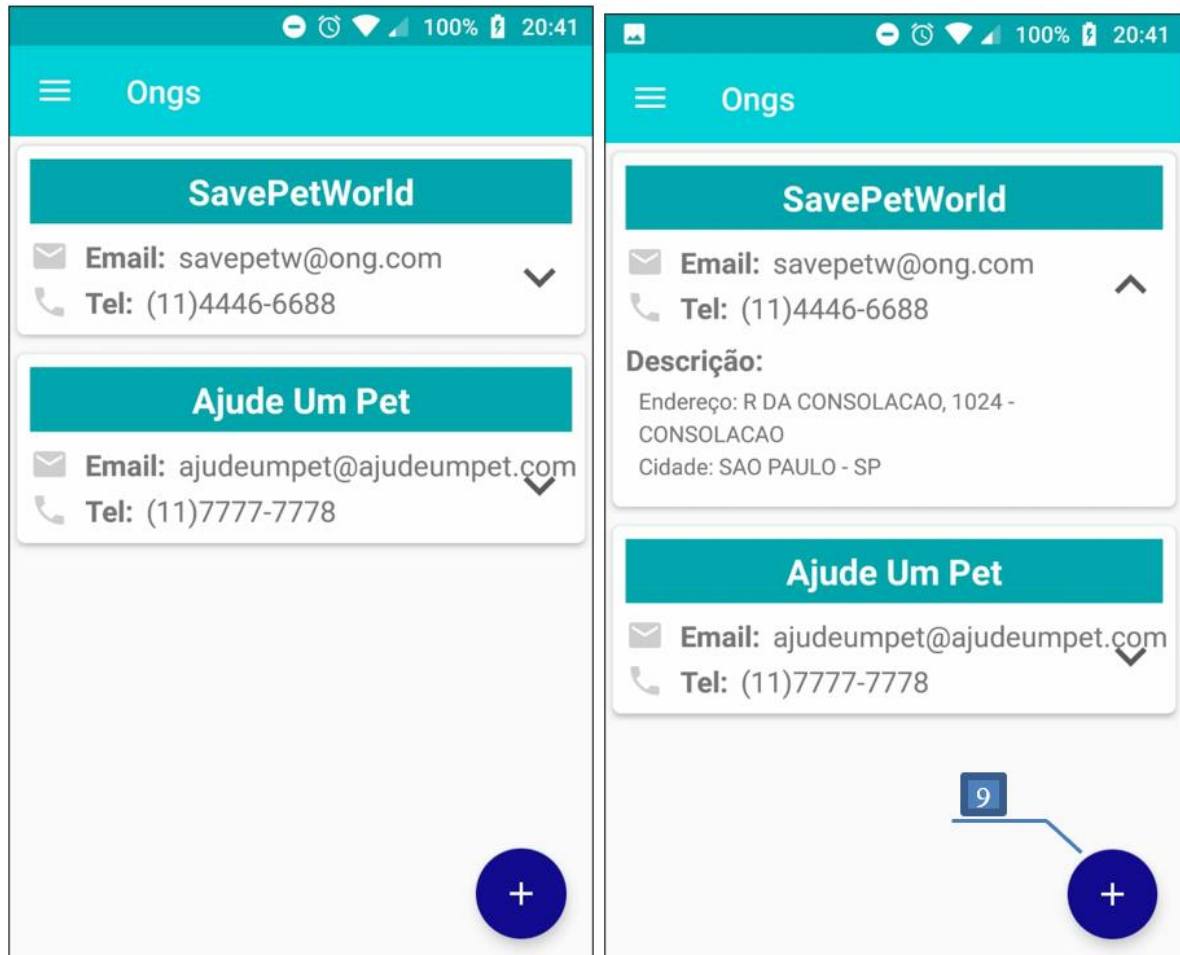
Figura 29 - Tela de Perfil



FONTE: Autoria Própria, 2019

5.1.8. Tela “Ongs”

Figura 30 - Tela de ONGs



Legenda de Navegação

9 – Registrar ONG

FONTE: Autoria Própria, 2019

5.1.9. Tela “Nova Ong”

Figura 31 - Tela de Registro de ONG

Novo Ong

Ong

Email:

Telefone

Rua

Número

Bairro

Cidade

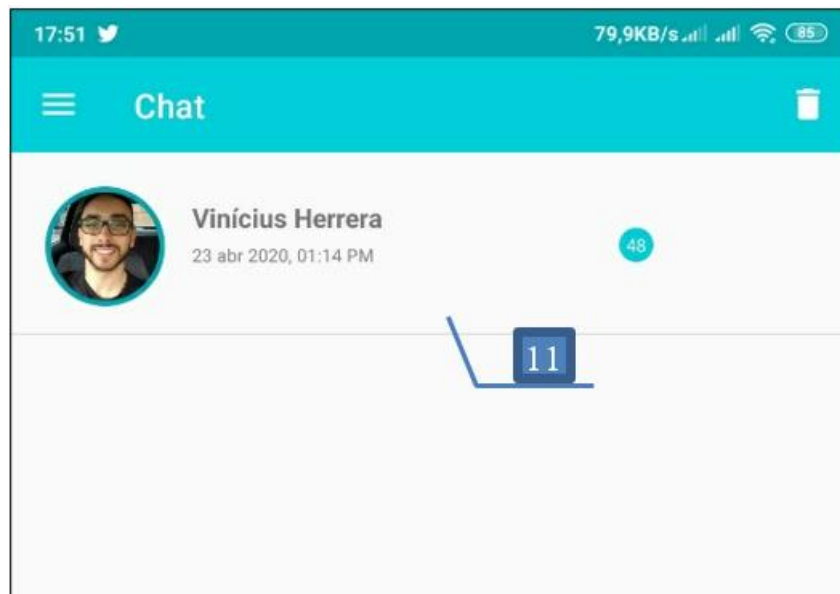
Estado

REGISTRAR

FONTE: Autoria Própria, 2019

5.1.10. Tela “Chat”

Figura 32 - Tela de Conversas (Chat)



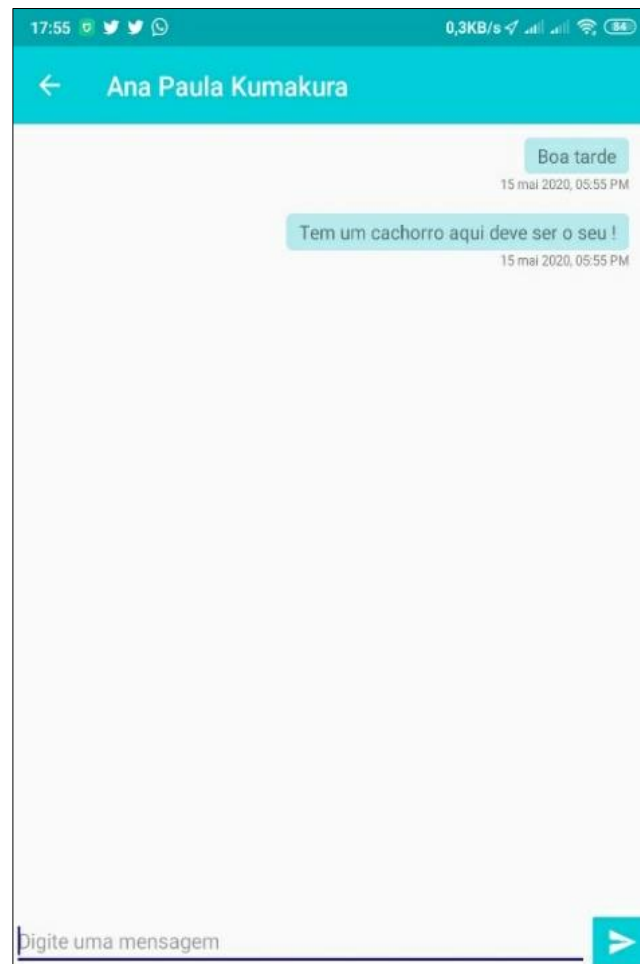
Legenda de Navegação

11 – Conversa (Chat)

FONTE: Autoria Própria, 2019

5.1.11. Tela de Conversa (Chat)

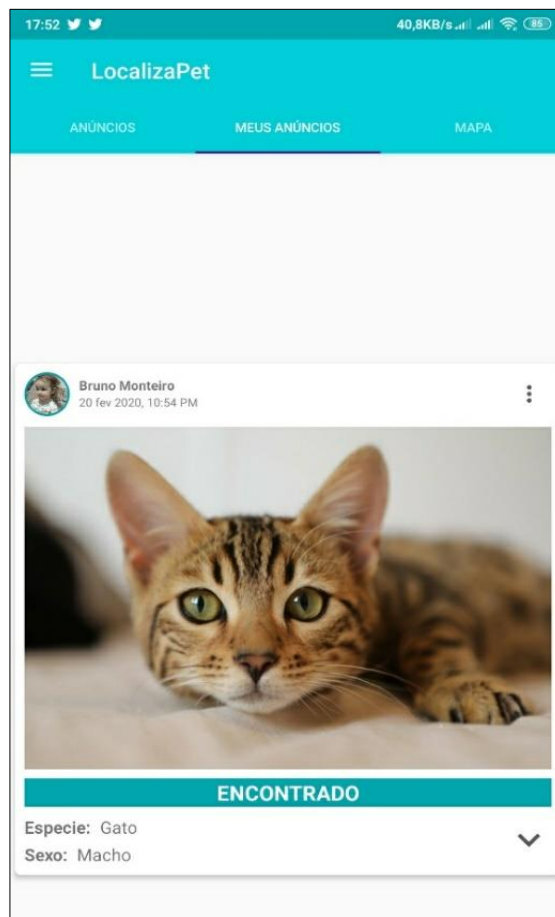
Figura 33 - Tela de Conversa (Chat)



FONTE: Aatoria Própria, 2019

5.1.12. Tela “Meus Anúncios”

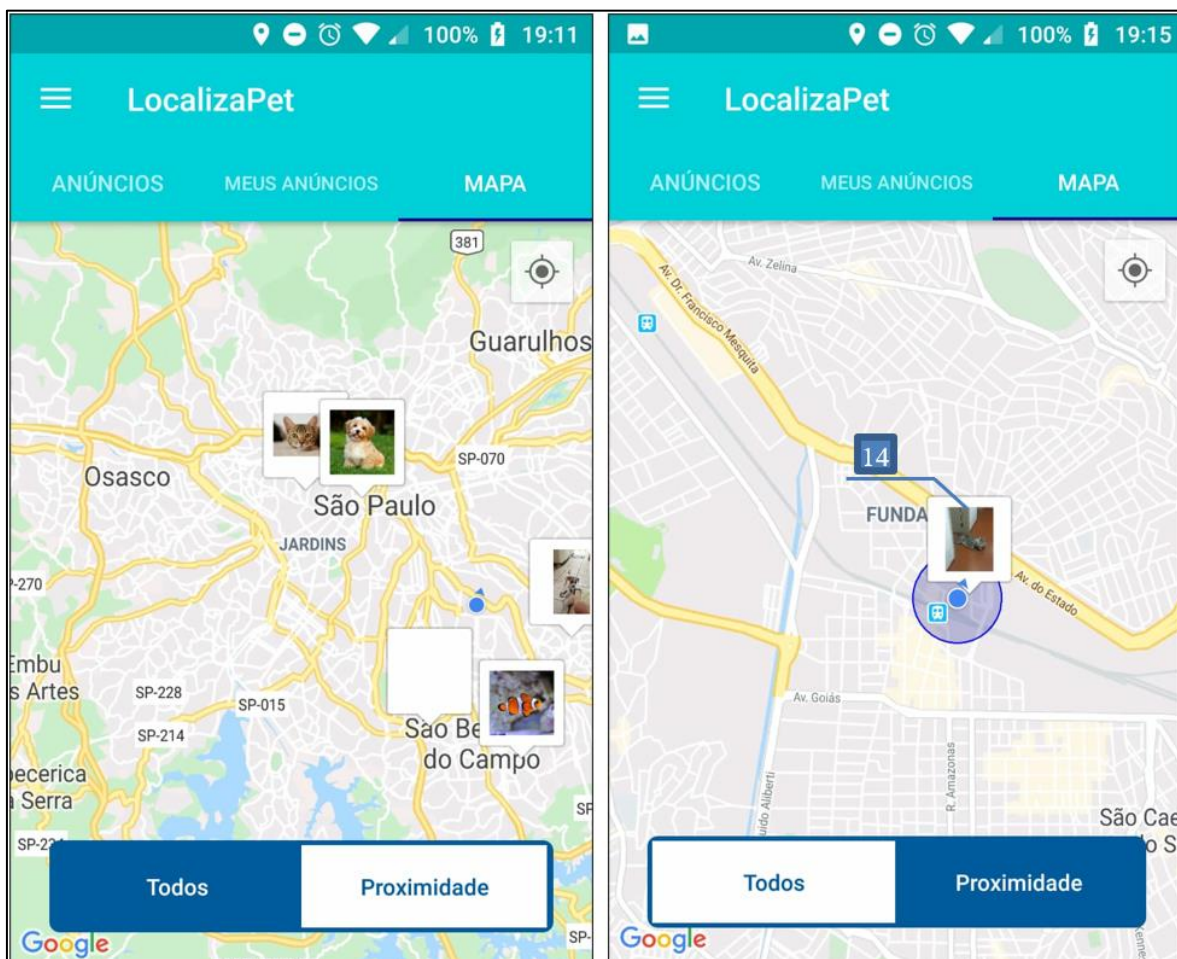
Figura 34 - Tela de “Meus Anúncios”



FONTE: Autoria Própria, 2019

5.1.13. Tela “Mapa”

Figura 35 - Tela de Mapa (Anúncios)



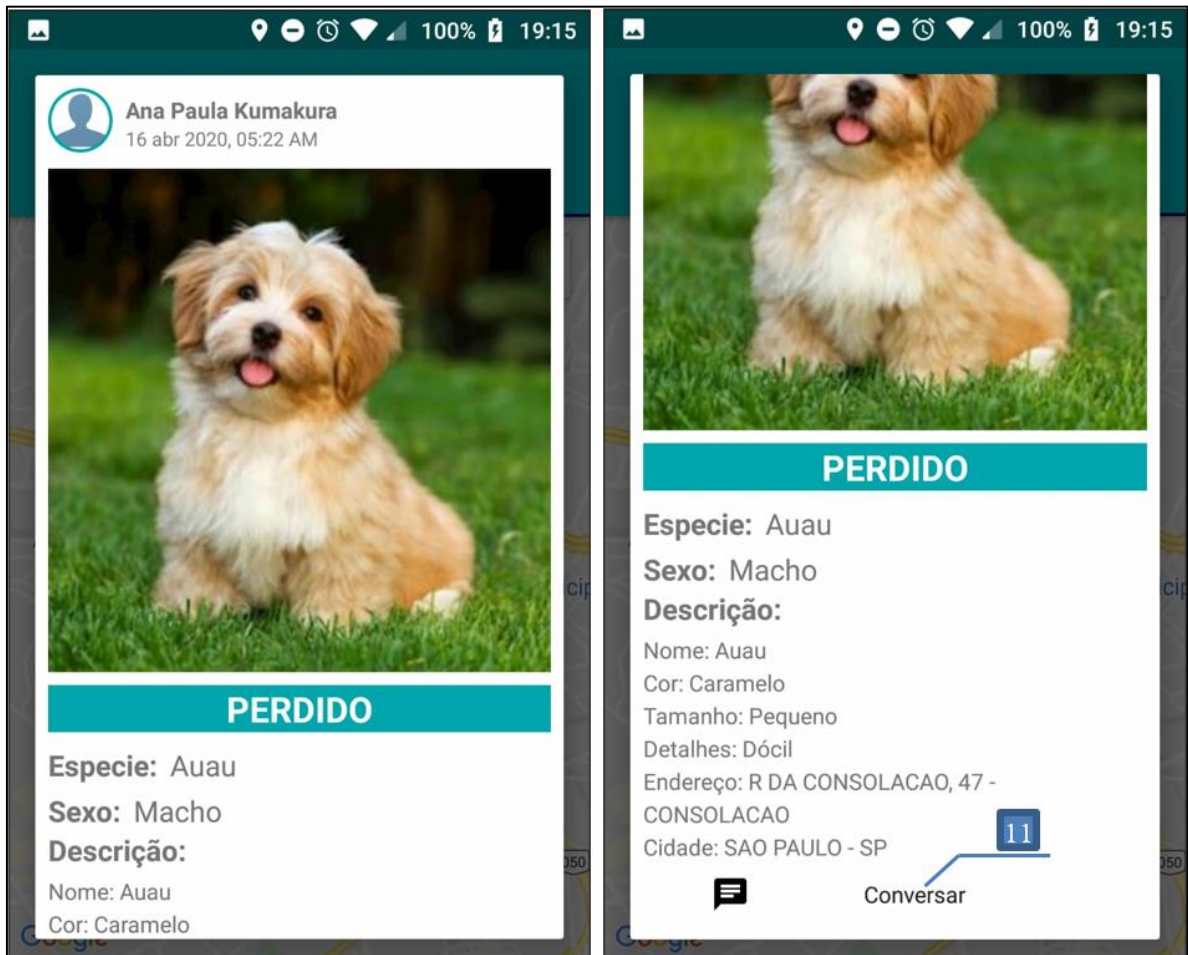
Legenda de Navegação

14 – Informações do Anúncio

FONTE: Autoria Própria, 2019

5.1.14. Tela de Mapa (Detalhes do Anúncio)

Figura 36 - Tela de Detalhes (Mapa)



Legenda de Navegação

11 – Conversa (Chat)

FONTE: Autoria Própria, 2019

6. TESTES

O teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionalidades como o esperado dentro do ambiente para o qual foi projetado. Há diversos tipos de teste de software que podem ser feitos durante o desenvolvimento de um projeto, dentre os principais temos os testes de performance, de usabilidade, os funcionais, os de caixa branca e de caixa preta. Os tipos de testes escolhidos para este projeto foram os testes de caixa branca e os testes funcionais que pertencem ao tipo de teste de caixa preta.

6.1. Teste caixa-branca (ou Teste Estrutural)

O teste de caixa branca é uma técnica de teste que avalia o comportamento interno do componente de software, e por isso permite uma verificação mais precisa do seu funcionamento. Trabalha diretamente sobre o código-fonte do componente de software para avaliar aspectos tais como: teste de condição, teste de fluxo de dados, teste de ciclos e teste de caminhos lógicos. Essa técnica é vista como complementar à técnica funcional e informações obtidas pela aplicação desses critérios têm sido consideradas relevantes para as atividades de manutenção, depuração e para a confiabilidade de software. O testador tem acesso ao código fonte da aplicação e pode construir códigos para efetuar a ligação de bibliotecas, componentes e reparação para possíveis erros.

6.2. Teste caixa-preta (ou Teste Funcional)

Diferentemente da técnica estrutural, o teste de caixa preta é uma técnica de teste em que o componente de software a ser testado não considera o comportamento interno do mesmo. Dados de entrada são fornecidos, o teste é executado e o resultado obtido é comparado a um resultado esperado previamente conhecido. Haverá sucesso no teste se o resultado obtido for igual ao resultado esperado. O componente de software a ser testado pode ser um método, uma função interna, um programa, um componente, um conjunto de programas e/ou componentes ou mesmo uma funcionalidade.

Os quadros apresentam todas as informações sobre os testes de caixa preta realizados com base nos casos de uso deste projeto:

Quadro 21 - Cenário: Manter Usuário

Cenário: Manter Usuário			
ID e nome do CT	001 – Novo cadastro		
Objetivo	Verificar se a funcionalidade de cadastro está executando corretamente	Tipo do Teste	Funcional
Cadastro Correto			
Passo	Ação	Resultado Esperado	
1	Clique na opção “Cadastre-se”	A tela de cadastro é apresentada pelo sistema	
2	Insira os dados para cadastramento e em seguida “Cadastrar”	O sistema informa ao usuário que o cadastro foi realizado e retorna para a tela de login.	
Cadastro Incorreto			
Passo	Ação	Resultado Esperado	
1	Clique na opção “Cadastre-se”	A tela de cadastro é apresentada pelo sistema	
2	Insira os dados para cadastramento e em seguida “Cadastrar”	O sistema sinaliza ao usuário que foram inseridos dados inválidos.	
ID e nome do CT	002 – Alterar dados do usuário		
Objetivo	Verificar se a funcionalidade de alterar os dados de cadastro do usuário está executando corretamente	Tipo do Teste	Funcional
Alteração Correta			
Passo	Ação	Resultado Esperado	
1	Efetue o login	O sistema apresenta a tela de anúncios.	
2	Clique no menu	O sistema apresenta as opções do menu.	
3	Clique no botão “Perfil”	A tela de perfil é apresentada ao usuário.	
4	Clique no botão “Editar”	O sistema libera os campos para alteração.	
5	Insira os dados atualizados, senha, confirmação de senha e clique em “Atualizar”	O sistema realiza a atualização e volta para a tela de Perfil.	
Alteração Incorreta 1			
Passo	Ação	Resultado Esperado	
1	Efetue o login	O sistema informa que ocorreu uma falha ao conectar com o servidor	
Alteração Incorreta 2			
Passo	Ação	Resultado Esperado	
1	Efetue o login	O sistema apresenta a tela de anúncios.	
2	Clique no menu	O sistema apresenta as opções do menu.	
3	Clique no botão “Perfil”	A tela de perfil é apresentada ao usuário.	
4	Clique no botão “Editar”	O sistema libera os campos para alteração.	
5	Insira os dados atualizados, senha, confirmação de senha e clique em “Atualizar”	O sistema informa uma mensagem de que não há conexão e volta para a tela de Perfil.	

FONTE: Autoria Própria, 2019

Quadro 22 - Cenário: Enviar Mensagem

Cenário: Enviar Mensagem			
ID e nome do CT	001 – Iniciar o Chat		
Objetivo	Verificar se a funcionalidade que inicia o chat está executando corretamente	Tipo do Teste	Funcional
Iniciação Correta			
Passo	Ação	Resultado Esperado	
1	Clique no ícone de Chat no anúncio.	A tela de Chat é apresentada pelo sistema	
2	Insira a mensagem e clique no ícone de enviar.	O sistema registra a mensagem no banco de dados e apresenta a mensagem na tela de Chat.	
Iniciação Incorreta			
Passo	Ação	Resultado Esperado	
1	Clique no ícone de Chat no anúncio.	O sistema informa que ocorreu uma falha ao conectar com o servidor.	

FONTE: Autoria Própria, 2019

Quadro 23 - Cenário: Buscar Animal

Cenário: Buscar Animal			
ID e nome do CT	001 – Busca de animais desaparecidos/encontrados		
Objetivo	Verificar se a funcionalidade que cruza os dados dos animais cadastrados com os dados inseridos está executando corretamente	Tipo do Teste	Funcional
Busca Correta			
Passo	Ação	Resultado Esperado	
1	Insira as informações do animal	O sistema compara as características no banco de dados, cria uma lista e a informa ao usuário.	
Busca Incorreta 1			
Passo	Ação	Resultado Esperado	
1	Insira as informações do animal	O sistema compara as características no banco de dados, não encontra animais com as características informadas e a apresenta uma mensagem ao usuário.	
Busca Incorreta 2			
Passo	Ação	Resultado Esperado	
1	Insira as informações do animal	O sistema não consegue conectar com o servidor informa ao usuário.	

FONTE: Autoria Própria, 2019

Quadro 24 - Cenário: Analisar Proximidade

Cenário: Analisar Proximidade			
ID e nome do CT		001 – Análise de Animais cadastrados próximos	
Objetivo	Verificar se a funcionalidade de análise de proximidade está executando corretamente	Tipo do Teste	Funcional
Análise Correta			
Passo	Ação	Resultado Esperado	
1	Clique na guia “Mapa”	A tela de mapa é apresentada mostrando as marcações de anúncios registrados	
2	Clique no botão “Verificar Proximidade”	O sistema apresente as marcações de anúncios com uma distância inferior a 300 metros na tela.	
Análise Incorreta			
Passo	Ação	Resultado Esperado	
1	Clique na guia “Mapa”	A tela de mapa é apresentada pelo sistema mostrando as marcações de anúncios registrados	
2	Clique no botão “Verificar Proximidade”	O sistema apresente uma mensagem de que não obteve resposta do GPS com a localização do usuário na tela.	

FONTE: Autoria Própria, 2019

Quadro 25 - Cenário: Manter Anúncio

Cenário: Manter Anúncio			
ID e nome do CT		001 – Novo anúncio	
Objetivo	Verificar se a funcionalidade de registrar um novo anúncio está executando corretamente	Tipo do Teste	Funcional
Cadastro Correto			
Passo	Ação	Resultado Esperado	
1	Clique no botão flutuante “+”	A tela para registro é apresentada pelo sistema	
2	Insira os dados para cadastramento e fotos do animal, em seguida “Anunciar”	O sistema informa ao usuário que o registro foi realizado e abre o feed de anúncios atualizado.	
Cadastro Incorreto			
Passo	Ação	Resultado Esperado	
1	Clique no botão flutuante “+”	A tela de cadastro é apresentada pelo sistema	
2	Insira os dados para cadastramento e fotos do animal, em seguida “Anunciar”	O sistema informa ao usuário que o cadastro não pode ser realizado.	
ID e nome do CT		002 – Atualizar Anúncio	

Objetivo	Verificar se a funcionalidade de alterar os anúncios registrados está executando corretamente	Tipo do Teste	Funcional
Alteração Correta			
Passo	Ação	Resultado Esperado	
1	Clique na guia “Meus Anúncios”	O sistema apresenta a tela de anúncios que aquele usuário registrou.	
2	Clique nas opções do anúncio	O sistema apresenta as opções disponíveis.	
3	Clique na opção de “Editar”	O sistema apresenta a tela de registro com os campos editáveis	
4	Altere os dados necessários e clique em “Atualizar”	O sistema salva os dados, informa ao usuário e retorna a tela de “Meus Anúncios”.	
Alteração Incorreta 1			
Passo	Ação	Resultado Esperado	
1	Clique no botão “Meus Anúncios”	O sistema não consegue contatar o servidor e informa que não há conexão.	
Alteração Incorreta 2			
Passo	Ação	Resultado Esperado	
1	Clique no botão “Meus Anúncios”	O sistema apresenta a tela de anúncios que aquele usuário registrou.	
2	Clique nas opções do anúncio	O sistema apresenta as opções disponíveis.	
3	Clique na opção de “Editar”	O sistema apresenta a tela de registro com os campos editáveis	
4	Altere os dados necessários e clique em “Atualizar”	O sistema informa ao usuário que há dados preenchidos incorretamente e retorna a tela de registros com os campos editáveis.	
ID e nome do CT	003 – Remover Anúncio		
Objetivo	Verificar se a funcionalidade de remover anúncio está executando corretamente	Tipo do Teste	Funcional
Remoção Correta			
Passo	Ação	Resultado Esperado	
1	Clique no botão “Meus Anúncios”	O sistema apresenta a tela de anúncios que aquele usuário registrou.	
2	Clique nas opções do anúncio	O sistema apresenta as opções disponíveis.	
3	Clique na opção de “Remover”	O sistema pede uma confirmação antes de excluir o anúncio.	
4	Confirme a solicitação de exclusão	O sistema exclui os dados do banco e informa que a exclusão foi realizada com sucesso e retorna a tela de registros.	
Remoção Incorreta 1			

Passo	Ação	Resultado Esperado
1	Clique no botão “Meus Anúncios”	O sistema não consegue contatar o servidor e informa que não há conexão.

FONTE: Autoria Própria, 2019

Quadro 26 - Cenário: Manter ONGs

Cenário: Manter ONGs			
ID e nome do CT		001 – Nova ONG	
Objetivo	Verificar se a funcionalidade de registrar uma nova ONG está executando corretamente	Tipo do Teste	Funcional
Cadastro Correto			
Passo	Ação	Resultado Esperado	
1	Clique na opção “ONG” no menu principal	A tela de ONGs é apresentada pelo sistema	
2	Clique no botão flutuante “+”	A tela de registro de ONGs é apresentada pelo sistema	
3	Insira os dados para cadastramento, em seguida “Registrar”	O sistema informa ao usuário que o registro foi realizado e abre a tela de ONGs atualizada.	
Cadastro Incorreto 1			
Passo	Ação	Resultado Esperado	
1	Clique na opção “ONG” no menu principal	A tela de ONGs é apresentada pelo sistema	
2	Clique no botão flutuante “+”	A tela de registro de ONGs é apresentada pelo sistema	
3	Insira os dados para cadastramento, em seguida “Registrar”	O sistema informa ao usuário que o cadastro não foi realizado por possuir dados inválidos.	
Cadastro Incorreto 2			
Passo	Ação	Resultado Esperado	
1	Clique na opção “ONG”	A tela de ONGs é apresentada pelo sistema	
2	Clique na opção “Registrar ONG”	A tela de registro de ONGs é apresentada pelo sistema	
3	Insira os dados para cadastramento, em seguida “Registrar”	O sistema informa ao usuário que a ONG já possui registro no sistema.	
ID e nome do CT		002 – Alteração de Informações	
Objetivo	Verificar se a funcionalidade de alterar os dados das registradas está executando corretamente	Tipo do Teste	Funcional
Alteração Correta			
Passo	Ação	Resultado Esperado	
1	Clique na opção “ONG” no menu principal	Uma tela com uma lista de ONGs é apresentada pelo sistema	

2	Clique no menu de uma ONG da lista que o seu usuário cadastrou	O sistema apresenta as opções disponíveis.
3	Clique na opção de “Editar”	A tela de alterar informações da ONGs é apresentada com os campos habilitados para edição.
4	Insira os dados para atualização, em seguida clique em “Atualizar”.	O sistema informa ao usuário que o registro foi realizado e abre a tela de ONGs atualizada.
Alteração Incorreta 1		
Passo	Ação	Resultado Esperado
1	Clique na opção “ONG” no menu principal	O sistema não consegue contatar o servidor e informa que não há conexão.
Alteração Incorreta 2		
Passo	Ação	Resultado Esperado
1	Clique na opção “ONG” no menu principal	Uma tela com uma lista de ONGs é apresentada pelo sistema
2	Clique no menu de uma ONG da lista que o seu usuário cadastrou	O sistema apresenta as opções disponíveis.
3	Clique na opção de “Editar”	A tela de alterar informações da ONGs é apresentada com os campos habilitados para edição.
4	Insira os dados para atualização, em seguida clique em “Atualizar”.	O sistema informa ao usuário que o cadastro não foi realizado por possuir dados inválidos.
ID e nome do CT 003 – Remoção de ONG		
Objetivo	Verificar se a funcionalidade de remover ONGs está executando corretamente	Tipo do Teste Funcional
Remoção Correta		
Passo	Ação	Resultado Esperado
1	Clique na opção “ONG” no menu principal	Uma tela com uma lista de ONGs é apresentada pelo sistema
2	Clique no menu de uma ONG da lista que o seu usuário cadastrou	O sistema apresenta as opções disponíveis.
3	Clique em “Remover”	O sistema pede a confirmação de remoção.
4	Confirme a remoção	O sistema exclui os dados do banco e informa que a exclusão foi realizada com sucesso e retorna a tela com a lista de ONGs.
Remoção Incorreta 1		
Passo	Ação	Resultado Esperado
1	Clique na opção “ONG”	O sistema não consegue contatar o servidor e informa que não há conexão.

FONTE: Autoria Própria, 2019

7. RESULTADOS

A partir dos resultados obtidos através da execução dos cenários de testes, é possível aferir que o projeto em questão atende aos requisitos funcionais, não funcionais e as regras de negócio levantadas na fase inicial de seu desenvolvimento.

O projeto apresenta uma interface amigável para o usuário, de fácil utilização e sem muitos processos para se executar uma única ação. Ele apresenta também uma interface gráfica para facilitar o entendimento do usuário, permitindo-o localizar anúncios em um mapa, anúncios próximos a ele ou não, sendo essa uma escolha do usuário.

Esta funcionalidade foi possível através da aplicação de uma fórmula bastante conhecida no universo da navegação conhecida como “Fórmula de Haversine”, utilizada para encontrar a distância entre dois pontos em uma esfera, utilizando seus valores de longitude e latitude.

Além disso, por tratar-se de um projeto para *smartphones*, boa parte relacionada à persistência do projeto foi implementada utilizando uma arquitetura de serviços conhecida como *REST*, para funcionamento utilizando servidores na nuvem, resultando assim em uma diminuição na carga sob os aparelhos dos usuários.

Disto isso, as configurações destes servidores foram realizadas para fins unicamente acadêmicos, portanto suas capacidades são bastante limitadas com relação à armazenamento e acessos simultâneos, porém, tratando-se de servidores na nuvem, essa limitação pode ser resolvida com bastante facilidade de acordo com as necessidades levantadas para o projeto.

CONSIDERAÇÕES FINAIS

Este trabalho de conclusão de curso tem como seu objeto de estudo o desenvolvimento de um aplicativo para dispositivos móveis, mais especificamente *smartphones* que utilizam o sistema operacional Android, que utiliza de recursos como serviços distribuídos em servidores na nuvem para realizar a persistência do aplicativo e serviços de localização do Google, o *Google Maps API*, para funcionalidades relacionadas à localização do usuário e manipulação de mapa.

Este projeto foi escolhido principalmente com o intuito de aplicar conhecimentos adquiridos durante todo o percurso acadêmico do grupo, conhecimentos esses que tinham grande potencial de aprimoramento por ter tópicos específicos e muito interessantes, como por exemplo a utilização de mapas e localização, ou o desenvolvimento de um sistema para consumo através de servidores na nuvem.

Dito isto, pode-se considerar que este projeto atingiu os objetivos propostos pelo grupo por terem sido aplicados vários dos conceitos ensinados pela instituição, como por exemplo conceitos de engenharia de software, onde a estrutura do sistema desenvolvido é apresentada através de diagramas UML; ou conceitos de desenvolvimento orientado à objetos, que foi a base de todo o sistema, bem como os conceitos e as ferramentas para o desenvolvimento móvel voltado à aparelhos com o sistema operacional Android.

Além disso, também foram realizados estudos sobre tecnologias as quais os membros do grupo possuíam um conhecimento básico, como por exemplo a linguagem de programação PHP, que foi a linguagem adotada para todo o desenvolvimento relacionado à servidores na nuvem. Inclusive, considerando a plataforma e os conceitos mais atuais de sistemas distribuídos, toda essa parte de desenvolvimento relacionado à servidores foi elaborada aplicando conceitos da arquitetura REST para serviços *web*.

Considerando todos esses fatores, o aplicativo apresentado ao final deste processo é considerado um produto que apresenta valor como solução para o problema estudado durante todo o decorrer deste trabalho, porém é um produto que está aberto para futuras melhorias, como por exemplo melhorias no seu desempenho ao utilizar o mapa, aplicando conceitos de processamento paralelo.

8. REFERÊNCIAS

BELIX, J. E. **Um estudo sobre MDA: Suporte fornecido pela UML e Reuso de soluções pré-definidas**. 2006. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Área de Concentração: Sistemas Digitais. São Paulo. [s.d.].

BERNARDO, P. C. **Padrões de testes automatizados**. Junho/2011. Dissertação (Mestrado) – Instituto de Matemática e Estatística da Universidade de São Paulo. Programa: Ciência da Computação. São Paulo. [s.d.].

BEZERRA, Eduardo **Princípios de Análise e Projeto de Sistemas com UML**. 2ª ed. Rio de Janeiro: Elsevier, 2007.

BIANCHI, M. J. **Ferramenta para configuração de modelos híbridos de gerenciamento de projetos**. 2017. Dissertação (Mestrado) – Escola de Engenharia de São Carlos da Universidade de São Paulo. Área de Concentração: Processos e Gestão de Operações. São Carlo. [s.d.].

CASTIGLIONI. MATHEUS, **Consumindo Web Service No Android** - Não encontrada. Disponível em: <https://blog.matheuscastiglioni.com.br/consumindo-web-service-no-android/> - Acesso em: 18 de Maio de 2019.

CODEXWORLD, **How to POST and Receive JSON Data using PHP cURL** – 10 de agosto de 2017. Disponível em: <https://www.codexworld.com/post-receive-json-data-using-php-curl/> - Acesso em: 25 de maio de 2019.

CROCKFORD. DOUGLAS, **JSON: The Fat-Free Alternative to XML** - 06 de dezembro de 2006. Disponível em; <http://www.json.org/fatfree.html> - Acesso em: 08 de agosto de 2019.

ESAUCE. **Qual a diferença entre site responsivo, site mobile (web app) e aplicativo para celular (app)?**. 2017. Disponível em <<https://www.esauce.com.br/blog/qual-diferenca-entre-site-responsivo-site-mobile-web-app-e-aplicativo-para-celular-app/>> Acesso em 10 de abril de.2018.

FIDAO. CHRIS, **Laravel 4: A Start at a RESTful API (Updated)** - 22 de maio de 2013. Disponível em: <https://code.tutsplus.com/tutorials/laravel-4-a-start-at-a-restful-api-updated-net-29785> – Acesso em 20 de Maio de 2019.

HOMEHOST, **PHPMailer: Como enviar email pelo PHP** – Não encontrada. Disponível em: <https://www.homehost.com.br/blog/tutoriais/php/enviar-email-php-com-phpmailer-smtp/> - Acesso em: 28 de maio de 2019.

MILANI, ANDRÉ. **MySQL - Guia do Programador**. - São Paulo: Editora Novatec, 2007.

NUDELMAN, G. **Padrões de projeto para o android: Soluções de projetos de interação para desenvolvedores**. Primeira edição. Editora WILEY Novatec, agosto/2013.

OLIVEIRA, I.S.C.S., DE SOUZA, A.P.C., COUTINHO, S.G. & MIRANDA, E.R. **Explorando conceitos – pesquisa bibliográfica e elaboração de infográfico sobre**

definições do campo de Design da Informação. Revista Brasileira de Design da Informação. São Paulo, v. 14, n. 3, p. 285-308, 2017.

PADOVANI, Stephania. **Design centrado no usuário: uma estratégia de diferenciação no desenvolvimento de aplicativos móveis.** Revista Brasileira de Design da Informação. São Paulo, v. 9, n. 2, p.139-142. [s.d.].

PETROUTSOS, Evangelos. **Google Maps: Power Tools for Maximizing the API.** Editora McGraw Hill Education, 2014.

RUMK, B. de P., RAMOS, D., FABRICIO, A., OBREGON, S. L. & LOPES, L. F. D. **Levantamento das necessidades e elaboração de plano de treinamento e desenvolvimento para empresa de TI.** ReCaPe – Revista de Carreiras e Pessoas. São Paulo., v. 6, n. 2, p.188-205. [s.d.].

ROMANATO. ALLAN, **Consumindo dados de um Web Service com Android** – 2015. Disponível em: <https://www.devmedia.com.br/consumindo-dados-de-um-web-service-com-android/33717> - Acesso em 18 de Maio de 2019

ROMANATO. ALLAN, **Trabalhando com AsyncTask no Android** – 2015. Disponível em: <https://www.devmedia.com.br/trabalhando-com-async-task-no-android/33481> – Acesso em: 24 de maio de 2019.

SENA. JOSIAS, **Using the Android Keystore system to store and retrieve sensitive information** – 12/fevereiro/2017. Disponível em: <https://medium.com/@josiassena/using-the-android-keystore-system-to-store-sensitive-information-3a56175a454b> - Acesso em: 20 de Maio de 2019.

SHAREEF. MOHAMMED BILAL, **How to Build REST API Using PHP** – 27 de novembro de 2015. Disponível em: <https://www.apptha.com/blog/how-to-build-a-rest-api-using-php/> - Acesso em: 20 de Maio de 2019.

SILVA, N. P. **Projeto e desenvolvimento de sistemas.** Oitava edição, Capítulo 2: **O ciclo de vida de um sistema.** Editora Érica, 2002.

TAURION, Cezar. **Cloud Computing - Computação em Nuvem: Transformando o mundo da Tecnologia da Informação.** Rio de Janeiro. Editora Brasport, 2009.

THAKUR. ANEH, **Post JSON data to server in android** - May 13, 2015. Disponível em: <https://trinitytuts.com/post-json-data-to-server-in-android/> - Acesso em: 18 de Maio de 2019.