

CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ
ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO – ARGO
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

FLÁVIA SORIANO DE MELLO BARROSO

**APLICATIVO MÓVEL EM PLATAFORMA ANDROID PARA ADOÇÃO DE
ANIMAIS DOMÉSTICOS**

Belém

2019

FLÁVIA SORIANO DE MELLO BARROSO

**APLICATIVO MÓVEL EM PLATAFORMA ANDROID PARA ADOÇÃO DE
ANIMAIS DOMÉSTICOS**

Trabalho de Curso na modalidade Produto, apresentado como requisito parcial para obtenção do grau de Bacharelado em Engenharia de Computação do Centro Universitário do Estado do Pará – CESUPA, sob orientação do professor M.Sc. Odlaniger Lourenço Damaceno Monteiro.

Belém

2019

FLÁVIA SORIANO DE MELLO BARROSO

**APLICATIVO MÓVEL EM PLATAFORMA ANDROID PARA ADOÇÃO DE
ANIMAIS DOMÉSTICOS**

Trabalho de Curso na modalidade Produto apresentado como requisito parcial
para obtenção do grau de Bacharelado em Engenharia de Computação do Centro
Universitário do Estado do Pará – CESUPA.

Data da Defesa: 13/06/2019

Banca Examinadora:

Prof. Orientador M.Sc. Odlaniger Lourenço Damaceno Monteiro – CESUPA

Prof. M.Sc. Ricardo Melo Casseb do Carmo – CESUPA

Prof. M.Sc. Marcos Douglas Da Silva Gomes – CESUPA

Belém

2019

DEDICATÓRIA

Dedico esse trabalho à minha querida amiga Marina Rachid (in memoriam), cuja lembrança me inspira e me faz persistir.

AGRADECIMENTOS

Agradeço aos meus pais, por terem me dado todo o suporte necessário durante minha jornada acadêmica e terem deixado como herança a minha educação.

Aos meus irmãos, por terem servido como exemplo de profissionais bem-sucedidos, me inspirando a buscar meu futuro.

Aos meus amigos e namorado, por sempre terem me dado apoio e incentivo a conclusão deste trabalho.

A todos os professores e funcionários desta universidade, que garantiram o meu aprendizado. E em especial, gratidão ao meu orientador, ao qual o trabalho não estaria completo sem suas correções e direcionamentos.

RESUMO

A adoção responsável é uma questão cuja visibilidade vem aumentando recentemente e tende a continuar em crescimento nos próximos anos devido ao ativismo, principalmente através da difusão da causa online. Em contrapartida, ainda há grande número de animais nas ruas, fazendo-se necessário a criação de modos inovadores que facilitem e incentivem o processo de adoção. Com isso, propôs-se o desenvolvimento de um aplicativo móvel em plataforma Android para adoção de animais domésticos por meio de tecnologias populares no mercado de trabalho atual, como Android Studio e Laravel. Os resultados deste trabalho culminam na publicação do aplicativo na loja virtual da Google e nos benefícios trazidos em cunho social, como a serventia enquanto novo meio de adoção responsável.

Palavras-chave: Adoção de animais; Dispositivos móveis; Android; Aplicativos.

ABSTRACT

Responsible adoption is a matter whose visibility has been increasing recently and is likely to continue to grow in the coming years due to activism, especially through the dissemination of the cause online. In contrast, there are still a large number of abandoned animals, making it necessary to create innovative ways to facilitate and encourage the adoption process. The development of a mobile application in Android platform for adoption of domestic pets was proposed, using popular technologies in the current labour market to do so, such as Android Studio and Laravel. The results of this work culminate in the publication of the application in Google virtual store and also the benefits brought socially, as the use as a new means of responsible adoption.

Palavras-chave: Pet adoption; Mobile devices; Android; Applications.

LISTA DE FIGURAS

Figura 1 – Cães em situação de maus tratos	19
Figura 2 – Gráfico de moradores em domicílios particulares permanentes com pelo menos um cão e/ou gato, por grupos de idade	21
Figura 3 – Domicílios particulares permanentes com pelo menos um cão e/ou gato, por classes de rendimento nominal mensal domiciliar	21
Figura 4 – Informações de um animal para adoção no site “Amigos de São Francisco”	27
Figura 5 – Página de busca de animal para adoção no site “Procure 1 Amigo”	28
Figura 6 – Página de lista de animais disponíveis no site “Amigo Não Se Compra”	29
Figura 7 – Aplicativo móvel “Adota Aqui”	30
Figura 8 – Sistema de filtragem do aplicativo “SaveBob”	31
Figura 9 – Resultado vazio para proximidades da cidade de Belém, aplicativo “SaveBob”	31
Figura 10 – Estrutura das camadas do padrão MVC	34
Figura 11 – Espectro de cores visíveis ao olho humano	35
Figura 12 – Círculo de cores complementares	36
Figura 13 – Exemplo de diagrama Entidade-Relacionamento	39
Figura 14 – Exemplo de modelagem lógica de BD relacional	39
Figura 15 – Exemplo de modelagem física de uma tabela	40
Figura 16 – Exemplo de comandos de DML	42
Figura 17 – Exemplo de comandos de DQL	42
Figura 18 – Design da tela de <i>login</i>	49
Figura 19 – Excerto do diagrama de banco de dados	50
Figura 20 – Excerto do serviço web	51
Figura 21 – Excerto de chamadas de serviço	52

Figura 22 – Excerto de implementação gráfica em XML e respectivo produto53

Figura 23 – Excerto de telas presentes na aplicação54

LISTA DE QUADROS

Quadro 1 – Principais tipos de dados na linguagem SQL	41
Quadro 2 – Comparação da sintaxe no padrão PHP e <i>Blade</i>	45

LISTA DE ABREVIATURAS E SIGLAS

ANDA	Agência de Notícias de Direitos Animais
ANSI	<i>American National Standards Institute</i>
BD	Banco de Dados
BEPA	Boletim Epidemiológico Paulista
CCZ	Centro de Controle de Zoonoses
DBA	<i>Database Administrator</i>
DDL	<i>Data Definition Language</i>
DML	<i>Data Manipulation Language</i>
DQL	<i>Data Query Language</i>
FGV	Fundação Getúlio Vargas
GUI	Interface Gráfica do Utilizador
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Ambiente de Desenvolvimento Integrado
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
ISO	<i>International Organization for Standardization</i>
MVC	<i>Model-View-Controller</i>
OMS	Organização Mundial da Saúde
ONG	Organização Não Governamental
PEA	Projeto Esperança Animal
SGBD	Sistema de Gerenciamento de Banco de Dados
SPAC	Sociedade Protetora de Animais de Curitiba
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
1.2	JUSTIFICATIVA	14
1.3	ESTRUTURA DO DOCUMENTO	15
1.4	METODOLOGIA	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	ABANDONO DE ANIMAIS	17
2.2	CRUELDADE NA VENDA DE ANIMAIS	18
2.3	IMPACTO E IMPORTÂNCIA DOS ANIMAIS EM AMBIENTE FAMILIAR	20
3	SOLUÇÕES EXISTENTES	23
3.1	ASSOCIAÇÃO DE PROTEÇÃO AOS ANIMAIS	23
3.1.1	Sociedades Protetoras dos Animais	25
3.2	LARES TEMPORÁRIOS	26
3.3	APLICAÇÕES VIRTUAIS	27
3.3.1	Aplicações Web	27
3.3.2	Aplicações Mobile	29
4	CONHECIMENTOS TÉCNICOS	32
4.1	ENGENHARIA DE SOFTWARE	32
4.1.1	Levantamento de Requisitos	33
4.1.2	Arquitetura de Software	33
4.1.2.1	Padrão de Arquitetura MVC	34
4.2	DESIGN DO APLICATIVO	34
4.2.1	Teoria das cores	35
4.2.2	Design em Adobe XD	37
4.3	DESENVOLVIMENTO BACK-END	37
4.3.1	Banco de Dados	38
4.3.1.1	Linguagem de Programação SQL	40
4.3.1.2	PhpMyAdmin	42
4.3.2	Regra de Negócio	43

4.3.2.1	Linguagem de Programação PHP	44
4.3.2.2	Framework Laravel.....	44
4.4	DESENVOLVIMENTO FRONT-END	45
4.4.1	Desenvolvimento Android	46
5	PROPOSTA DA APLICAÇÃO.....	47
5.1	PLANEJAMENTO	47
5.1.1	Requisitos	47
5.1.2	Arquitetura	48
5.1.3	Design	48
5.2	DESENVOLVIMENTO	49
5.2.1	Banco de Dados	49
5.2.2	Serviço Web.....	50
5.2.3	Chamadas de Serviço	51
5.2.4	Telas	52
6	CONSIDERAÇÕES FINAIS.....	55
6.1	RESULTADOS.....	55
6.2	DIFICULDADES ENCONTRADAS	55
6.3	TRABALHOS FUTUROS	56
7	REFERÊNCIAS	57
	APÊNDICE A – Mapa Visual de Modelo de Negócios.....	63
	APÊNDICE B – Documentação de Requisitos do Projeto.....	64
	APÊNDICE C – Design das Telas do Aplicativo.....	68
	APÊNDICE D – Codificação de uma Activity.....	70
	APÊNDICE E – Codificação de um Fragment	73

1 INTRODUÇÃO

Os meios de adoção na região norte limitam-se à eventos promovidos por Organizações Não Governamentais que recolhem animais de rua e visitas físicas à canis, sociedades protetoras ou abrigos onde os animais se encontram alojados. Segundo pesquisa do Centro de Controle de Zoonoses de Belém para o jornal O Liberal em 2016, estima-se que cerca de trinta mil animais se encontrem em situação de rua somente na capital do estado do Pará, número este que poderia ser bem maior sem a atuação das entidades anteriormente citadas.

Com o advento da Tecnologia da Informação, o perfil apresentado pela região na área de adoções se demonstra ultrapassado, deixando em aberto espaço para inovações que facilitem e agilizem esse processo, tornando-o mais cômodo e impulsionando aumento no número de animais adotados.

Visando atender a essa necessidade, propõe-se o desenvolvimento de uma aplicação para dispositivos móveis com capacidade de conectar usuários dispostos a adotar animais e usuários que possuam animais à adoção. Possibilitando procura e escolha de animal em qualquer tempo ou espaço, além de garantir maior exposição dos animais para adoção, sem envolver esforço ou locomoção física. Com isso, busca-se facilitar o acesso ao processo de adoção de animais através da tecnologia móvel.

Acerca do aspecto tecnológico como alicerce, segundo estudo realizado pela Fundação Getúlio Vargas na 29ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas (2018), existem 220 milhões de aparelhos smartphones ativos no país, apontando grande propensão de crescimento desse número, que já ultrapassa a população nacional. Por conseguinte, a evolução do mercado indica tendência de maior utilização de smartphones à de computadores desktop. Em vista disso, afirma-se que a aplicação proposta não se tornaria obsoleta em seu nascimento ou anos seguintes.

Com a presença de informações e maior exposição dos animais a todo tempo e todo local com conexão à internet, retira-se as limitações antes encontradas no meio de adoção que se caracterizavam como dificuldades às famílias que desejassem adotar, além de propiciar uma visibilidade que poderá estimular maior vontade de adoção aos usuários.

1.1 OBJETIVOS

Os objetivos encontram-se divididos em geral e específicos, de acordo com os tópicos apresentados abaixo.

1.1.1 Objetivo geral

Desenvolver um aplicativo que facilite o acesso ao processo de adoção de animais domésticos.

1.1.2 Objetivos específicos

São objetivos específicos deste trabalho:

- Identificar informações necessárias para o processo de adoção;
- Planejar meio de uso dessas informações na aplicação;
- Definir funcionalidades requisito aos tipos de usuário do aplicativo;
- Aplicar conceitos de engenharia de software e de modelagem de banco de dados;
- Utilizar a plataforma Android Studio e o *framework* Laravel para desenvolvimento;
- Conceder maior visibilidade à causa através da virtualização do processo;
- Conectar usuários dos dois lados do processo, garantindo comunicabilidade;

1.2 JUSTIFICATIVA

O número de animais abandonados no Brasil é de aproximadamente 30 milhões, segundo pesquisa realizada pela Organização Mundial da Saúde (2014) e divulgada pelo portal Agência de Notícias de Direitos Animais e entidade Projeto Esperança Animal. Esse alto índice se deve à variados fatores, como a desistência de uma família a cuidar desse animal ou a reprodução sem controle dos animais que vivem nas ruas, e por consequência, apresenta uma situação de risco tanto para os animais, quanto para a população, devido à falta de vacinas e outros cuidados com a saúde do animal que podem desencadear doenças em humanos.

Dentre os animais abandonados, estima-se, ainda pela OMS (2014), que 20 milhões sejam cachorros e 10 milhões sejam gatos. Em contrapartida, estatísticas do Instituto Brasileiro de Geografia e Estatística (2013) apontam que em cerca de 44% dos lares brasileiros habitam pelo menos um cachorro de estimação, sendo que

apenas 41% desses provém de adoções, enquanto que somente 17% dos lares possuem pelo menos um gato, porém 85% desses são adotados. Faz-se possível a observação que ainda há grande preferência pela compra de animais de raças bem definidas, sobretudo em cães, aumentando o abandono por “vira-latas”, ainda que comprovada e evidenciada a crueldade presente na comercialização de animais.

Portanto, fundamenta-se a necessidade de promover a adoção responsável de animais negligenciados, tendo a aplicação em questão na pesquisa como medida para facilitar esse fim. Apresentando como pretensão a capacidade de proporcionar ampliação dos números de adoção através da comodidade de utilizar um aplicativo em sua própria residência ou qualquer localidade com conexão à internet e do aumento de visibilidade como consequência da plataforma a qual se situa, que é meio de comunicação presente no cotidiano da população.

1.3 ESTRUTURA DO DOCUMENTO

A estrutura do trabalho se apresenta organizada da seguinte forma:

O capítulo 1 contém a porção introdutória, apresentando contextualização geral para inserção do trabalho, além de objetivos e justificativa para o mesmo.

O capítulo 2 apresenta a fundamentação social, explanando as problemáticas e conceitos acerca do assunto de adoção, abandono de animais e outras situações relevantes.

O capítulo 3 cita as soluções existentes para a problemática, independente de conter tecnologia ou não, diferenciando-as da solução oferecida pelo trabalho à fim de evidenciar seus benefícios.

O capítulo 4 expõe a proposta do produto a ser desenvolvido por meio deste trabalho, contendo desde seu planejamento até o funcionamento da aplicação final.

O capítulo 5 provê fundamentação técnica dos conhecimentos utilizados para execução do trabalho.

O capítulo 6 aborda os resultados obtidos, dificuldades encontradas ao longo do processo e perspectivas futuras do trabalho como considerações finais.

1.4 METODOLOGIA

Visando prover base para a problemática que buscará ser solucionada pelo produto a ser desenvolvido, será necessária investigação da situação atual do assunto

através de notícias recentes e estatísticas geradas por instituições confiáveis, além de verificar estudos voltados à área, à fim de compor a pesquisa exploratória a ser realizada como base ao capítulo de fundamentação teórica e de soluções existentes.

Será necessário também pesquisar o nível de viabilidade do produto como solução disposta, analisando o contexto atual em que se inserem a presença de *smartphones* e aplicações móveis no cotidiano da população, estudando a capacidade de vida útil do produto e como aumentá-la segundo as tendências. Para isso, serão apresentados, no capítulo que diz respeito à proposta do trabalho, os aspectos do modelo de negócios, visando iniciar a produção com estes já delimitados.

Após verificação dessas informações, se dará início ao desenvolvimento do aplicativo, onde será indispensável a busca de estudos complementares ao conhecimento já possuído sobre as tecnologias a serem utilizadas em caso de dúvidas ou necessidade de sanar erros de programação.

Dentre as tecnologias escolhidas destacam-se o Adobe XD, utilizado para construção do design a ser seguido durante composição da interface do aplicativo; o Android Studio como plataforma de desenvolvimento de aplicação móvel para *smartphones* com sistema operacional baseado em Android e o Laravel como *framework* para elaboração do Banco de Dados em MySQL e disposição do serviço web a ser manipulado pela aplicação. Essas tecnologias e adjacentes serão detalhadas no capítulo de fundamentação técnica.

As dúvidas referentes a esses conhecimentos técnicos serão pesquisadas na documentação e guias para desenvolvedores nos próprios sites do Android e do Laravel ou em fóruns de programação de escala internacional online.

Após término do desenvolvimento do aplicativo, haverá a publicação deste na loja virtual da Google, em conjunto a estratégias de propaganda online, buscando inserção de público e consequentes entradas de dados para tornar a aplicação funcional.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica abrange questões sociais que influenciam diretamente no processo de adoção de animais. Em destaque, trata do abandono e suas consequências, da comercialização e crueldade e, por último, da convivência familiar com um animal de estimação e suas vantagens.

2.1 ABANDONO DE ANIMAIS

O abandono de animais, de acordo com Alves *et al.* (2013), é um problema frequente no Brasil e em toda América Latina que desencadeia uma série de consequências, sobretudo nas áreas de saúde pública, ecologia, economia e social, devido a presença dos animais em local público sem qualquer tipo de supervisão ou cuidados veterinários.

Na área de saúde, destaca-se a raiva canina, uma doença letal que pode ser transmitida por cães não vacinados. Segundo dados fornecidos pela OMS e publicados por Kotait *et al.* (2007) no Boletim Epidemiológico Paulista, cerca de 55.000 óbitos humanos são registrados anualmente ao redor do mundo, com estatística de que a cada 15 minutos falece uma pessoa infectada pelo vírus da raiva, 40% sendo crianças menores de 15 anos.

No domínio ecológico, analisa-se o impacto causado ao ecossistema devido à vivência do animal doméstico sem providências fornecidas por humanos. Diversos estudos retificam o potencial de predação desses animais. Aves, insetos, répteis e pequenos mamíferos são principais presas para gatos ferais (DICKMAN, 1996). Já para cães, é comum ocorrer perseguição, perturbação e deslocamento de animais nativos, infligindo estresse muitas vezes fatal para a fauna, segundo Rangel e Neiva (2013).

Na esfera econômica, são relevantes os gastos governamentais para colocar em prática estratégias de controle populacional dos animais à fim de impedir maiores impactos. Essas estratégias englobam fundação de unidades de saúde pública para controle de zoonoses e programas de vacinação e esterilização para animais abandonados.

O âmbito social abrange desde a geração de poluição sonora e excreções em áreas públicas, até a depredação de propriedade privada ou pública acometida pelos animais sem supervisão. Para Alves *et al.* (2013), outro aspecto social importante é o impacto do abandono ao bem-estar dos animais. Esses animais frequentemente

apresentam deficiências na saúde física e/ou mental, que são agravadas pela maior probabilidade de exposição a condições prejudiciais, segundo Stafford (2007).

O abandono de animais ocorre por variadas causas. É possível apontar como principais os problemas comportamentais do animal indisciplinado, a falta de disponibilidade de espaço, custo em tempo e dinheiro com cuidados requeridos, falta de responsabilidade depois da aquisição e expectativas não atingidas sobre a convivência com um bicho de estimação.

2.2 CRUELDADE NA VENDA DE ANIMAIS

De acordo com o art. 32 da Lei nº 9.605 de 12 de fevereiro de 1998, “praticar ato de abuso, maus tratos, ferir ou mutilar animais silvestres, domésticos ou domesticados, nativos ou exóticos confere ao réu, pena de detenção de três meses a um ano” (BRASIL, 1998). Infelizmente, embora haja legislação, ocorrem diversas práticas abusivas contra animais que não costumam ser fiscalizadas.

É comum, em criadouros, utilização de uma animal fêmea como produto para reprodução à cada cio, inclusive tendo seus movimentos constringidos por amarras para facilitar o processo de cruza. Complicações no parto podem levar à fatalidade do animal e a incidência constante da gravidez pode trazer diversas complicações incômodas pós-parto, como infecções.

Atingindo idade em que não é mais possível realizar reprodução, esses animais são descartados para abandono ou deixados para falecer. O ciclo do tratamento do animal como produto continua no nascimento da próxima geração, onde os filhotes são obrigados a parar de mamar prematuramente e são mantidos isolados em espaços insuficientes até a comercialização.

Evidenciado na figura 1, em um canil de Piedade, interior de São Paulo, que foi fechado por maus tratos em 2019, está um conjunto de animais domésticos a serem comercializados, sendo mantidos confinados em pequenos espaços e sem receber devidos tratamentos (G1, 2019).

Figura 1 – Cães em situação de maus tratos



Fonte: Divulgação/Polícia Militar Ambiental (2019)

Além dos impactos diretos causados aos animais dentro desse sistema de mercado, há também impactos indiretos causados aos animais abandonados. Afinal, estima-se que haja cerca de 30 milhões de animais abandonados no Brasil (ANDA, 2014; PEA, 2014).

Havendo uma exagerada quantidade de animais abandonados disponíveis, precisando de um lar e família, a realização da compra de um animal toma espaço que poderia ser ocupado por um que necessite. Além de impulsionar o número de abandonos, visto que a compra incentiva continuação do comércio e geração cada vez maior de animais para venda.

Outro fator que impulsiona aumento do número de animais abandonados nesse cenário é o consequente arrependimento após compra de um animal devido à facilidade de realizá-la sem fazer qualquer tipo de estudo anterior. Após verificar que a realidade de se ter um animal de estimação é distinta das expectativas formadas sem base, esse animal comercializado sofre risco de ser descartado na porta de clínicas ou até mesmo na rua.

“A nossa sociedade atribuiu pouca importância ao abuso de animais, considerando-os como criaturas de natureza inferior, destinadas a ser utilizadas pelo homem que sobre elas tinha direitos e poder absolutos.” (FONSECA; DIAS, 2011, p. 72). Essa visão é fundamento dos diversos abusos realizados em prol de beneficiamento humano, enxergando o animal como utilidade ao invés de ser vivo sentiente.

2.3 IMPACTO E IMPORTÂNCIA DOS ANIMAIS EM AMBIENTE FAMILIAR

De acordo com Hart (1985), a interação humana com outras espécies sempre foi fundamental para sobrevivência, passando de predação para domesticação. Essa necessidade é facilmente evidenciada atualmente ao observar dados do IBGE (2013) que apontam a existência de 1,5 bilhões de animais de estimação ao redor do mundo, e ainda, cerca de 132 milhões somente no Brasil, sendo a quarta maior população de animais de estimação do mundo.

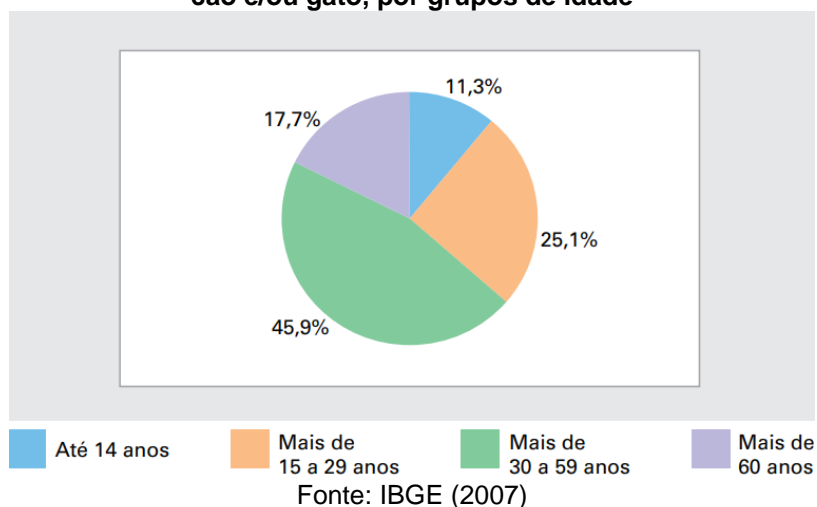
Ter um animal de estimação pode prover vários benefícios à diversas faixas etárias, se tornando uma adição ideal a qualquer família. Segundo Tatibana e Costa-Val (2009), esses benefícios podem se manifestar em forma de aprendizado de afetividade, sensibilidade e senso de responsabilidade em crianças, estímulo à realização de exercícios em adultos sedentários e companhia, proteção e até auxílio em terapias, com a melhora psicológica e emocional na junção entre pessoas e animais de estimação sendo cada vez mais relatada.

Outro benefício relevante seria o conforto provido aos idosos, visto que, de acordo com Suthers-McCabe (2001), animais de estimação são muitas vezes considerados membros da família, especialmente por pessoas de maior idade, sendo a relação homem-animal mais forte e mais profunda na velhice do que em qualquer outra idade, como auxílio na adaptação à terceira idade.

Com isso, a presença de animais domésticos em residências familiares proporciona interações físicas, psicológicas, emocionais e sociais mútuas, ocasionando melhoria na qualidade de vida com aumento de estados de felicidade e redução de estresse.

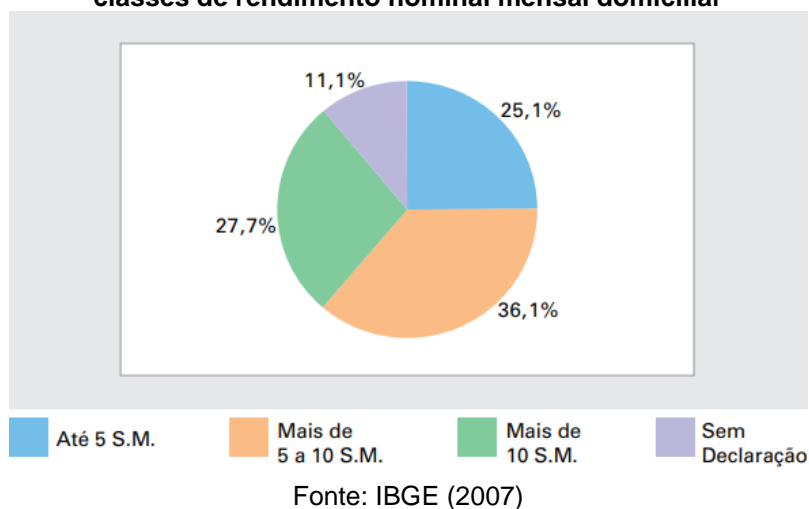
Motivada pelos benefícios apresentados, destaca-se como evidente a procura por aquisição de animais domésticos por parte das famílias, sendo possível observar através da figura 2, em relatório de pesquisa realizado pelo IBGE (2007) com 4.848 domicílios particulares permanentes com pelo menos um cão e/ou gato, que a obtenção de animais domésticos é adequada a todas as faixas etárias.

Figura 2 – Gráfico de moradores em domicílios particulares permanentes com pelo menos um cão e/ou gato, por grupos de idade



Ainda pelo mesmo relatório de pesquisa do IBGE (2007), na figura 3 observa-se que a aquisição de animais domésticos não possui relação direta com a renda nominal mensal domiciliar, apresentando dados acerca da presença de animais que evidenciam indiferença de acordo com renda pelo balanceamento das porcentagens ao longo das faixas de intervalos de salários mínimos recebidos no domicílio.

Figura 3 – Domicílios particulares permanentes com pelo menos um cão e/ou gato, por classes de rendimento nominal mensal domiciliar



Embora haja essa independência entre renda e presença de animal doméstico, é importante ressaltar que existem custos financeiros a serem arcados com a manutenção e cuidados requeridos pelo animal, além de custos supérfluos realizados por pura satisfação do dono em busca de agradar o bicho de estimação. Devido a esses custos, o mercado envolvendo a área de animais domésticos cresce

exponencialmente impulsionado pela relação entre homem e animal, segundo Oliveira (2006).

Considerando o tratamento de animais com perspectiva utilitária, destacam-se custos como vacinas periódicas, consultas veterinárias, banho, tosa, alimentação e eventual castração, entre outros itens necessários para criação saudável do mesmo, já na perspectiva que os considera como membro da família, há ademais gastos como roupas, coleiras personalizadas, guloseimas, bolsas para carregá-los ou qualquer tipo de apetrecho que vise satisfação extra do animal ou dono e até mesmo espaços da casa dedicados ao animal.

A divisão de perspectivas é baseada em Konecki (2007), em que a consideração como membro da família explicita o “antropomorfismo” do animal, atribuindo-o com qualidades e necessidades humanas, o que justificaria a propriedade de caracterizar o consumo de certos itens superficiais como indispensáveis, enquanto que a consideração utilitária ou econômica visualiza o animal apenas como recurso capaz de trazer benefícios através do convívio pessoal, evitando gastos frívolos.

Segundo o IBGE em pesquisa publicada no ano de 2015,

Em 2013, 44,3% dos domicílios do país possuíam pelo menos um cachorro, o equivalente a 28,9 milhões de unidades domiciliares. [...] A população de cachorros em domicílios brasileiros foi estimada em 52,2 milhões, o que indicou uma média de 1,8 cachorro por domicílio, considerando-se o conjunto de domicílios com este animal. Em relação à presença de gatos, 17,7% dos domicílios possuíam pelo menos um, o equivalente a 11,5 milhões de unidades domiciliares. [...] A população de gatos em domicílios brasileiros foi estimada em 22,1 milhões, o que representa aproximadamente 1,9 gato por domicílio com este animal.

Os dados obtidos demonstram a importância da presença de animais de estimação nos domicílios brasileiros. Essa presença tem como uma das consequências diretas na sociedade o crescimento do ativismo a favor dos direitos dos animais domésticos.

3 SOLUÇÕES EXISTENTES

Este capítulo cita e explica soluções existentes no campo de adoção de animais de estimação, sejam elas soluções físicas ou virtuais, independentemente de haver aplicação de tecnologia.

3.1 ASSOCIAÇÃO DE PROTEÇÃO AOS ANIMAIS

De acordo com estudo realizado pelo IBGE (2016) acerca de fundações privadas e associações sem fins lucrativos no Brasil, conforme salientado pela tabela 1, há cerca de 1.689 unidades voltadas ao meio ambiente e proteção animal, representando um percentual de apenas 0,7 da totalidade de fundações existentes.

Essas unidades citadas compreendem, entre outros, hospitais veterinários para tratamento cirúrgico e odontológico, serviços de vacinação e/ou esterilização em animais, assistência veterinária em domicílios e consultórios, além de associações protetoras dos animais e natureza.

O baixo percentual apresentado destoa a reduzida participação dessas entidades no cenário brasileiro. Isso se dá devido ao fato de que muitas dessas organizações sobrevivem de doações ou ajuda solidária da população.

Logo, as associações de proteção aos animais podem ser públicas ou privadas – ONGs, sem fins lucrativos e com participação voluntária, dedicadas a gerenciar suas atividades à fim de proteger, reabilitar, realizar controle de natalidade, promover serviços de saúde veterinária, proibir maus tratos e encaminhar à adoção responsável animais abandonados.

A adoção desses animais se dá principalmente por feiras de adoção locais, onde os animais são transportados para localização pública específica e ficam expostos para visualização de pessoas com desejo de se tornarem possíveis donos. Porém, esse método apresenta certas dificuldades quanto à reserva do local, o horário que pode não ser o ideal para todas as partes envolvidas, dependência de pessoal voluntariado, o transporte dos animais e a estrutura física necessária para exposição.

Tabela 1 – Fundações Privadas e Associações sem Fins Lucrativos

Classificação das entidades sem fins lucrativos	Fundações Privadas e Associações sem Fins Lucrativos - FASFIL		
	Total	Percentual (%)	
		Em relação ao total	Em relação ao grupo
Total	236 950	100,0	-
Habitação	163	0,1	100,0
01.0 Habitação	163	0,1	100,0
Saúde	4 721	2,0	100,0
02.1 Hospitais	2 464	1,0	52,2
02.2 Outros serviços de saúde	2 257	1,0	47,8
Cultura e recreação	32 268	13,6	100,0
03.1 Cultura e arte	12 552	5,3	38,9
03.2 Esporte e recreação	19 716	8,3	61,1
Educação e pesquisa	15 828	6,7	100,0
04.1 Educação infantil	4 872	2,1	30,8
04.2 Ensino fundamental	3 349	1,4	21,2
04.3 Ensino médio	1 627	0,7	10,3
04.4 Educação superior	1 999	0,8	12,6
04.6 Estudos e pesquisas	1 315	0,6	8,3
04.7 Educação profissional	464	0,2	2,9
04.8 Outras formas de educação/ensino	2 202	0,9	13,9
Assistência social	24 067	10,2	100,0
05.0 Assistência social	24 067	10,2	100,0
Religião	83 053	35,1	100,0
06.0 Religião	83 053	35,1	100,0
Associações patronais profissionais	28 962	12,2	100,0
07.3 Associações empresariais e patronais	7 457	3,1	25,7
07.4 Associações profissionais	11 132	4,7	38,4
07.5 Associações de produtores rurais	10 373	4,4	35,8
Meio ambiente e proteção animal	1 689	0,7	100,0
08.0 Meio ambiente e proteção animal	1 689	0,7	100,0
Desenvolvimento e defesa de direitos	30 266	12,8	100,0
09.1 Associações de moradores	9 162	3,9	30,3
09.2 Centros e associações comunitárias	10 918	4,6	36,1
09.3 Desenvolvimento rural	4 291	1,8	14,2
09.4 Emprego e treinamento	871	0,4	2,9
09.5 Defesa de direitos de grupos e minorias	3 430	1,4	11,3
09.6 Outras formas de desenvolvimento e defesa de direitos	1 594	0,7	5,3
Outras instituições privadas sem fins lucrativos não especificadas anteriormente	15 933	6,7	100,0
10.8 Outras instituições privadas sem fins lucrativos não especificadas anteriormente	15 933	6,7	100,0

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Cadastro e Classificações, Cadastro Central de Empresas 2016.

Com o avanço da tecnologia e popularização das redes sociais, um meio encontrado para dar visibilidade à causa e iniciar processos de adoção de forma facilitada é a divulgação online dos animais. Essa divulgação ocorre normalmente através de publicações nas páginas próprias das associações, do *Facebook*, *Instagram* e até mesmo por aplicativos de mensagens instantâneas como o *Whatsapp*.

3.1.1 Sociedades Protetoras dos Animais

As sociedades protetoras de animais são uma modalidade encontrada dentro da definição de Associação de Proteção aos animais, sendo ONGs, sem fins lucrativos, que dependem da colaboração e doações de pessoas sensibilizadas com a causa de recolher animais abandonados à fim de reabilitá-los para adoção.

À exemplo da Sociedade Protetora de Animais de Curitiba, que recebe animais feridos ou doentes, vítimas de maus tratos e abandono e serve de lar provisório enquanto presta atendimento a esses animais para que possam ser adotados e retornar a ter um lar permanente (SPAC, 2018).

As sociedades protetoras surgiram como medida resposta à solução imposta pelo governo no que diz respeito ao recolhimento de animais da rua. Solução esta que consistia em realizar coleta dos animais das ruas e mantê-los guardados em canis ou gatis no aguardo de que os donos viessem à busca-los.

Porém, o problema real dessa solução surgia em casos de abandono, quando ninguém aparecia para resgatar um animal e era dado permissão para que fosse efetuado extermínio do mesmo, sem ser apontado como prática criminosa. Atualmente vigora projeto de lei que proíbe sacrifício de cães e gatos abandonados em órgãos de controle de zoonoses e canis públicos (CARAM, 2017).

Segundo a SPAC (2018), para realizar adoção de forma responsável, é necessário verificar disposição acerca das seguintes medidas:

- Espaço suficiente para a espécie escolhida, considerando suas características e necessidades;
- Disponibilidade para cuidar do animal como novo integrante da família durante toda a vida dele, considerando vida média de 12 anos;
- Condição de arcar com despesas de cuidados veterinários e remédios, além de alimentação e abrigo;

- Tempo livre a ser reservado para dar atenção e carinho ao animal, além de dar banho e escová-lo regularmente;
- Disposição para levar o animal em passeios onde poderá realizar exercícios físicos com regularidade;
- Capacidade de não deixar o animal sozinho por longos períodos de tempo a fim de não infligir estresse e sofrimento ao mesmo.

Sendo assim, além de reabilitar e prover lar temporário para bichos de estimação, as sociedades protetoras também se encarregam de conscientizar a população sobre as responsabilidades da posse de um animal. Essa conscientização é realizada por meio de campanhas e eventos especiais em prol da causa, divulgação de informações sobre os cuidados necessários e leis existentes.

3.2 LARES TEMPORÁRIOS

Os lares temporários servem de abrigo à animais retirados das ruas dentro do período em que se busca por um lar permanente sob a guarda de um adotante responsável (OSÓRIO, 2013).

O lar temporário pertence a um voluntário que se disponibilize a atender diretamente animais e suas necessidades durante um período de tempo, se tornando tutor responsável por eles nessa fase.

A maior diferença entre lares temporários e outros modos de recolhimento de animais abandonados é o atendimento direto à menores grupos de animais ou um indivíduo apenas durante sua estadia. Enquanto que canis/gatis se encontram lotados em quantidade de animais, lares temporários utilizam de comprometimento direto de voluntários para cuidar de menor número de bichos.

Dessa forma, garante-se melhor tratamento do animal de forma personalizada e com maior atenção, embora seja de difícil emprego remanejar grande quantidade de animais pela dependência da boa vontade de grande número de pessoas voluntárias.

A busca por um lar permanente a esses animais pode ser realizada pelo próprio voluntário ou com auxílio de entidades de proteção aos animais. A busca, quando realizada, comumente compreende meios de anúncio como jornais, rádios, cartazes e sites ou redes sociais na internet.

Além de ajudar a abrigar animais que anteriormente não teriam espaços disponíveis em outros locais de abrigo, lares temporários também são capazes de ajudar na adaptação do animal à futura família definitiva, acostumando-o com um ambiente distinto do vivenciado nas ruas.

3.3 APLICAÇÕES VIRTUAIS

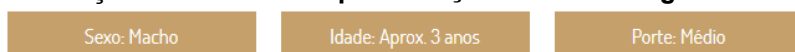
Devido à quantidade de animais abandonados, à procura de um lar, iniciou-se uso de tecnologias como estratégia para impulsionar o processo de adoções. Entre os meios mais comuns e criados especificamente para essa função, estão os sites e aplicativos móveis voltados a promover informações acerca dos animais necessitados e do procedimento de adoção em geral.

3.3.1 Aplicações Web

As aplicações Web são compostas por sites fundados por ONGs, abrigos, associações e sociedades protetoras de animais à fim de divulgar seu trabalho com intuito de gerar doações e anunciar animais disponíveis à adoção.

Dentre essas aplicações, destaca-se o site de uma ONG de São Paulo chamada “Amigos de São Francisco”, que consta com páginas informando como ajudar com voluntariado, auxílio financeiro e lar temporário.

Figura 4 – Informações de um animal para adoção no site “Amigos de São Francisco”



O Cascão foi resgatado quando corria entre os carros no meio da marginal, correndo muito risco de vida. Ele deu a sorte de cruzar o caminho do seu padrinho Ricardo, que ofereceu a chance do recomeço e para ficar ainda melhor, só falta ele ter um lar cheio de amor!



Quero Adotar



Quero Apadrinhar



Posso dar um lar temporário

Fonte: *Print* de tela de página da aplicação Web “Amigos de São Francisco” (2019).

Além de contar com uma página dedicada à adoção dos animais recolhidos, é capaz de filtrar pelo porte dos animais e tornar visível os dados do animal caso selecionado, fazendo possível escolher entre as opções de adotar, prover lar temporário ou doar ajuda financeira específica – apadrinhar – àquele animal, como exibido na figura 4.

Outro site notável é o “Procure 1 Amigo”, que é composto por ONGs e voluntários, atuando na defesa e bem-estar do animal doméstico. Contém páginas voltadas à adoção de cães, gatos, à exposição de animais perdidos e encontrados e de casos de adoção realizados com sucesso.

Figura 5 – Página de busca de animal para adoção no site “Procure 1 Amigo”

The screenshot shows the 'Procure 1 Amigo' website interface for searching dogs in PA - Belém. On the left, there is a sidebar with a search filter section titled 'Encontre um Animal'. This section includes a dropdown for 'Finalidade' (ADOÇÃO), a dropdown for 'Tipo' (CAO), and dropdowns for 'Porte' (TODOS), 'Sexo' (TODOS), and 'Idade' (TODAS). The main content area is titled 'CÃES para DOAÇÃO em PA - Belém' and displays a grid of dog listings. Each listing includes a photo of the dog, its name, sex, size, and age. The listings are as follows:

Nome	Sexo	Tamanho	Idade
Lobinha	Fêmea	Medio	1 ano
Pérola	Fêmea	Medio	2 a 6 meses
Sem nome	Ambos	Medio	1 ano
Herus	Macho	Grande	5 anos
Belinha	Fêmea	Pequeno	1 ano
Berê	Fêmea	Pequeno	Abaixo de 2 meses
Camel	Fêmea	Medio	2 a 6 meses
Perola	Fêmea	Medio	2 a 6 meses
NERO			
Sem nome			

Fonte: *Print* de tela de página da aplicação Web “Procure 1 Amigo” (2019).

A figura 5 é um *print* de tela da área de adoção de cães do site, onde é possível filtrar por diversas categorias e ainda alterar localidade de busca, sendo funcional para todo território brasileiro.

Como última exemplificação, apresenta-se o site “Amigo Não Se Compra”, existente desde 2012 e oficializado como uma Associação em 2017, que permite divulgação de animais para adoção por ONGs ou protetores que realizem cadastro em sua plataforma.

Conta também com uma área de dicas e cuidados, de como ajudar a causa e de depoimentos. Com mais de 12.000 animais cadastrados e mais de 5.000 já adotados, o site conta com um sistema de busca por animal de acordo com espécie,

porte e localidade, além de mostrar lista de animais disponíveis em ordem cronológica caso não haja pesquisa feita, evidenciado na figura 6.

Figura 6 – Página de lista de animais disponíveis no site “Amigo Não Se Compra”

The screenshot shows the 'Amigo Não Se Compra' website interface. At the top, there's a large orange banner with the text 'Encontre seu novo amigo'. Below this, there are four search filters: 'Espécie', 'Estado', 'Porte', and 'Cidade', each with a dropdown arrow. A purple arrow points from the filters to a purple button labeled 'PROCURAR'. Below the button is a purple paw print icon. Underneath, there's a section titled 'Novos no Site' with a subtitle: 'Nosso site está cheio de peludos ansiosos pra ter uma família. Tente diferentes buscas até encontrar um peludo pra chamar de seu. :)'. Below this, there are five animal profiles, each with a photo, name, location, and gender icon.

Nome	Localização	Gênero
Luke	Compo Limpo Paulista / São Paulo	♂
Leia	Compo Limpo Paulista / São Paulo	♀
Teco	São Paulo / São Paulo	♂
Euclides Jorge	Osasco / São Paulo	♂
Princesa	Juliz de Fora / Minas Gerais	♀

Fonte: *Print* de tela de página da aplicação Web “Amigo Não Se Compra” (2019).

3.3.2 Aplicações Mobile

As aplicações móveis são compostas por aplicativos para celulares com acesso à internet, geralmente desenvolvidas por voluntários ou encomendadas por ONGs, visando um meio dinâmico e remoto de realizar divulgação de animais sem lar permanente.

O “Adota Aqui” é um aplicativo com as funcionalidades de visualizar cães e gatos disponíveis para adoção – exposto na figura 7 – e registrar animais como disponíveis. Apesar de não dispor de filtros por categorias, como gênero ou idade do animal, contém sistema de pesquisa por estado e cidade, sendo que uma vez escolhido o estado não é possível alterar.

Figura 7 – Aplicativo móvel “Adota Aqui”



Fonte: *Print de tela da aplicação Mobile “Adota Aqui” (2019).*

Possui sistema de mensagens instantâneas embutida no aplicativo, porém o perfil de usuário é limitado somente ao nome e e-mail do utilizador, o que proporciona facilidade de acesso, mas dificuldade de identificação. Também não consta funcionalidade de editar ou excluir anúncio do animal e aceita apenas um anúncio por usuário.

O “SaveBob” é um aplicativo que busca expor a importância dos animais de estimação e proporcionar a adoção de maneira ágil e remota. Possui sistema de filtros por diversos requisitos, demonstrado na figura 8, seção de achados e perdidos, de pets cadastrados e pets favoritos.

Figura 8 – Sistema de filtragem do aplicativo “SaveBob”

Tipo de Pet

Cachorro Gato Outro

Distância

40.00 km

Gênero

♂ ♀

Idade

0 a 11 meses 1 a 2 anos 3 a 5 anos

6 a 9 anos > 10 anos

Temperamento

Dócil Carente Ativo

Protetor Assustado Bravo

Castrado

Sim Não

Fonte: *Print* de tela da aplicação Mobile “SaveBob” (2019).

Classificado com objetivo de aproximar bichos de estimação de seus possíveis donos e concretizar um relacionamento eterno. Entretanto, pela falta de popularização e por sua exibição de animais ocorrer de acordo com a localidade do usuário, não possui nenhum resultado de animal disponível na região Norte do Brasil – figura 9.

Figura 9 – Resultado vazio para proximidades da cidade de Belém, aplicativo “SaveBob”

Fonte: *Print* de tela da aplicação Mobile “SaveBob” (2019).

4 CONHECIMENTOS TÉCNICOS

Este capítulo visa expor tecnologias e conteúdos técnicos utilizados como base durante o planejamento e desenvolvimento do aplicativo como solução proposta. Para a fase de planejamento, relevam-se estratégias de engenharia de software e recursos de design, enquanto que para a fase de desenvolvimento, ressaltam-se as ferramentas utilizadas nas etapas *back-end* e *front-end*.

4.1 ENGENHARIA DE SOFTWARE

Segundo Pressman e Maxim (2015), um software consiste em instruções executáveis que fornecem características, funções e desempenho desejados. De forma simplificada, software é um produto virtual com capacidade de fornecer resultados a partir de processamento. Tendo os resultados fornecidos como pertencentes à diversas áreas possíveis, como aplicações voltadas à cálculos, composição textual ou processamento de transações comerciais.

O software produz, gerencia, adquire, modifica, exibe e transforma o produto mais importante da nossa era, que é a informação (PRESSMAN; MAXIM, 2015). O fator de destaque para utilização é a entrega de informações de forma simples e ágil, sendo o software apenas um programa com instruções a circular em uma máquina e potencializado pelo hardware da mesma.

Com a necessidade de preparar profissionais para lidar com o processo de desenvolvimento de software de forma organizada, surge a engenharia de software, que, para o Instituto de Engenheiros Eletricistas e Eletrônicos (1990), compreende a aplicação sistemática, disciplinada e quantitativa no desenvolvimento, operação e manutenção do software.

A engenharia de software trata, então, de todas as tarefas envolvidas para concretizar a programação de um software, mantendo em vista custo, prazo e qualidade do produto final. De acordo com Paula Filho (2001, p. 16),

Estimar prazos e custos faz parte da rotina de qualquer ramo da engenharia. Para um produto ser viável, não basta que atenda aos requisitos desejados; tem de ser produzido dentro de certos parâmetros de prazo e custo. Se isto não for possível, o produto pode não ser viável do ponto de vista de mercado, ou pode ser preferível adquirir outro produto, ainda que sacrificando alguns dos requisitos. Ter estimativas de prazos e custos, portanto, é uma expectativa mais que razoável de clientes e gerentes.

Portanto, temos como fatores críticos na engenharia de software, o levantamento de requisitos, a estimativa de prazos e a análise dos custos. Sendo relevante apontar que são valores dependentes entre si, visto que, por exemplo, quanto maior a lista de requisitos e menor o prazo, maior seria o custo.

Para Engholm Júnior (2010), o desenvolvimento de software inclui os processos de levantamento e análise de requisitos, arquitetura, design, implementação, testes e implantação. Essas etapas são concluídas com auxílio da utilização de ferramentas presentes na engenharia de software.

4.1.1 Levantamento de Requisitos

O levantamento e análise de requisitos são formados por um conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto de software (PAULA FILHO, 2001). Os requisitos são tudo aquilo que se pretende tornar possível na aplicação final, necessitando de documentação para ser seguida ao longo do desenvolvimento e validação para confirmar a execução satisfatória daquilo que foi proposto anteriormente.

Em outras palavras, ao início do desenvolvimento de software, faz-se necessário definir aquilo que será a composição de funcionalidades e características da implementação final. Essa definição deve ser feita em conjunto, reunida a equipe de desenvolvimento, ou em sondagem do cliente, caso a aplicação seja encomendada.

4.1.2 Arquitetura de Software

Segundo Bass, Clements e Kazman (2013), a arquitetura de um software abrange a representação da estrutura do sistema, os componentes do software, as propriedades visíveis desses componentes e suas relações. Essa representação permite análise do projeto, o que por sua vez, permite realização de mudanças nesse estágio caso haja necessidade.

A arquitetura de software facilita a comunicação entre todas as partes envolvidas, destaca as decisões de projeto e constitui um modelo compreensível da estruturação do sistema e de seus componentes em conjunto (BASS; CLEMENTS; KAZMAN, 2013). O desenvolvimento desse modelo se dá de acordo com o padrão de arquitetura empregado. Para Pressman e Maxim (2015), os padrões tratam de um

problema inserido em contexto específico com determinadas limitações e restrições, propondo uma solução de arquitetura como base para o projeto.

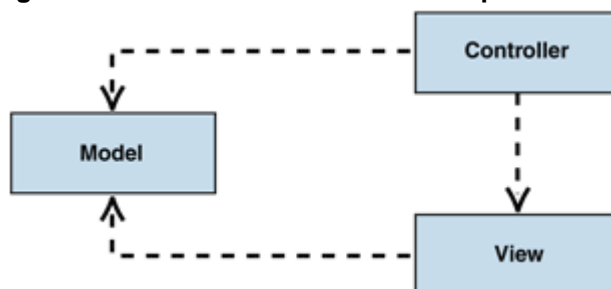
4.1.2.1 Padrão de Arquitetura MVC

O padrão de arquitetura Model-View-Controller é o padrão escolhido para execução deste trabalho devido às características do projeto e facilidades proporcionadas pelo mesmo em questão de implementação.

De acordo com a Microsoft (2014), este padrão, concebido na década de 1970, separa o modelo de domínio, a vista de apresentação e as decisões baseadas nas ações de entrada da aplicação em três principais camadas:

- O modelo, que lida com regras e dados do domínio da aplicação;
- A vista, que administra a disposição de informações na interface;
- O controlador, que modifica estados no modelo e/ou na vista a partir de ações de entrada.

Figura 10 – Estrutura das camadas do padrão MVC



Fonte: Microsoft (2014).

A figura 10 ilustra a relação das três camadas em um projeto, explicitando a dependência das camadas de vista e controlador à camada de modelo, que por sua vez não depende das demais, podendo ser implementada e testada separadamente.

“A grande vantagem de se utilizar o padrão MVC é a separação de lógica e apresentação, sendo que isso favorece o trabalho em equipe.” (MINETTO, 2007, p. 19). Com isso, enquanto um componente da equipe trabalharia na interface, outro seria capaz de desenvolver o controlador simultaneamente, e assim por diante.

4.2 DESIGN DO APLICATIVO

O design de uma aplicação móvel é composto por padrões de navegação, formulários, listas, ordenação e visualização de componentes ferramentas (NEIL,

2012). Compreendendo tudo aquilo que entrará em interação direta com o usuário através dos sentidos corporais.

É importante para acompanhar a marca do aplicativo, relacionando aspectos do design com a aplicação propositalmente no subconsciente do usuário frequente. De modo que os símbolos, as cores, o tema geral do aplicativo se torne reconhecível sem precisar de elementos textuais para identificação do mesmo.

Uma preocupação crítica do design é desenvolver produtos interativos que sejam de fácil utilização, eficazes e que proporcionem uma experiência positiva ao usuário (ROGERS; SHARP; PREECE, 2011). Por conta dessa preocupação, um dos objetivos principais do design é a composição de uma interface amigável a qualquer tipo de usuário.

Para compor uma interface amigável é necessário levar em consideração os elementos visuais mais significativos e capazes de dispor ergonomia, à exemplo de ícones e fontes. Como elementos marcantes a serem discutidos destacam-se as cores.

4.2.1 Teoria das cores

Newton (1671) realizava distinção teórica entre cores primárias e cores compostas a partir das propriedades da luz ao invés da sensação visual, sendo a composta produzida por duas ou mais componentes de luz, enquanto a primária por apenas uma.

Figura 11 – Espectro de cores visíveis ao olho humano



Fonte: Pixabay (2012).

Entretanto, fora ressaltado, ainda por Newton (1671), que os olhos não são capazes de diferenciar esses dois tipos, mas é importante notar que a mesma sensação pode ser produzida por diferentes tipos de luz, formando o espectro de cores visíveis exibido na figura 11.

Segundo Goethe (1810), cada cor tem um efeito específico na psique humana, provocando sensações e reações similares em indivíduos diferentes. Por isso, de acordo com o objetivo da aplicação faz-se essencial escolher a combinação de cores correta para evocar determinados sentimentos no usuário.

Embora seja um dos assuntos mais impactantes acerca de marketing e persuasão, os efeitos das cores como psicologia ainda são baseados em poucos dados fatuais, devido às diferenças de preferências pessoais, experiências, contexto e cultura de cada indivíduo, sendo necessário prever a reação do consumidor à adequação da cor ao produto (CIOTTI, 2014). Sendo assim, faz-se de maior importância associar as cores escolhidas ao produto criado do que seguir a associação estereotípica relacionada a cada uma dessas cores.

Figura 12 – Círculo de cores complementares



Fonte: Projeto Literacia (2013).

Outro fator a se considerar na escolha do tema de cores da aplicação é o conforto provocado na sensação visual. Da Vinci (1651) descreve que as cores sofrem influência em percepção de outras cores que se encontrem relacionadas. A fim de manipular essa percepção de forma positiva, o design de aplicativos faz uso de estratégias para escolha das cores.

Uma dessas estratégias compõe-se da utilização de cores complementares, observável na figura 12, onde, por exemplo, as tonalidades de roxo são

complementares às tonalidades de dourado, promovendo um contraste chamativo, porém confortável visualmente.

4.2.2 Design em Adobe XD

O design de aplicações pode ser feito a partir de desenhos à mão, como um esboço das ideias que se pretende concretizar na versão de implementação, ou com o auxílio de ferramentas que disponham de componentes que sejam fiéis representações dos presentes na aplicação real. Uma dessas ferramentas é o software Adobe XD, que proporciona facilidade na criação de design e protótipos de aplicativos Web, iOS, Android, entre outros.

O Adobe XD permite uso de ferramentas intuitivas para modelar *layouts*, adicionar objetos, símbolos e navegação e conecta várias áreas de trabalho em função de prover interação na fase de design, como protótipo, explica a Adobe (2019). Além disso, conta com a possibilidade de utilizar *plug-ins* e conjuntos prontos de interface das plataformas à que se pretendem aplicar, visando agilizar o processo de prototipação.

Apresenta funções de compartilhamento em nuvem para que o trabalho possa ser realizado a partir de diversas máquinas e visualizado inclusive no dispositivo alvo final. Por exemplo, um protótipo realizado para dispositivos móveis com sistema operacional Android pode ser compartilhado em nuvem e acessado a partir desse dispositivo à fim de verificar se a representação confeccionada se encaixa aos requisitos pretendidos.

O grande destaque desse programa se dá na confiabilidade de armazenamento do trabalho, dado que o sistema de compartilhamento permite fácil formação de duplicatas do projeto como cópia de segurança, e na interface de utilização simples e clara, permitindo a prática de selecionar com o clique um objeto e “arrastá-lo” até a posição desejada, comumente conhecida como “*click and drag*”.

4.3 DESENVOLVIMENTO BACK-END

Back-end é aquilo que o usuário não vê, não compõe aspectos visuais da interface, porém, serve como base para qualquer transação de informações na aplicação. “O *back-end* executa as regras de negócio da aplicação, ele é responsável por validar e consistir os dados vindos do *front-end*. “ (AMARAL; NERIS, 2015, p. 89).

Em outras palavras, *back-end* é o termo utilizado para referir-se ao processamento realizado do lado do servidor.

Então, a criação e gestão do Banco de Dados, a lógica de manipulação dos dados e a disposição das informações filtradas através de serviços web para coleção por parte do *front-end* posteriormente fazem parte do *back-end* de uma aplicação.

4.3.1 Banco de Dados

Segundo Machado e Abreu (2012), os dados são registros de informações originados a partir do tratamento dessas informações, de forma que apenas seus aspectos relevantes sejam registrados. Então, retém-se que dado é a parte importante de uma informação para determinada aplicação, preparado para ser processado por um sistema computadorizado.

O conjunto de dados integrados que tem por objetivo atender a uma comunidade de usuários é chamado de Banco de Dados (HEUSER, 2009). A forma de agrupamento e disposição desses dados depende da aplicação à que se refere, fazendo-se necessário haver um Sistema de Gerenciamento de Banco de Dados.

Um sistema de gerenciamento de banco de dados é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. (...) O principal objetivo de um SGBD é proporcionar uma forma de armazenar e recuperar informações de um banco de dados de maneira conveniente e eficiente. (SILBERSCHATZ; KORTH; SUDARSHAN, 2012, n.p.)

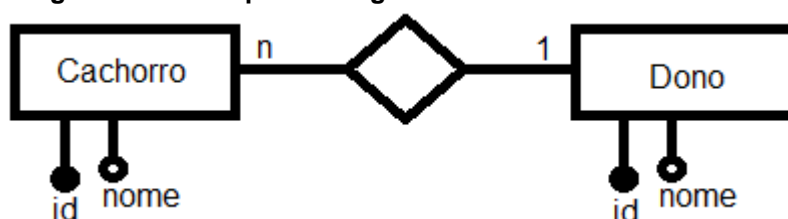
Então, as ferramentas que compõem um SGBD são capazes de providenciar a criação das estruturas de dados de maneira organizada, visando evitar falhas e garantir segurança, além de possibilitar adição, pesquisa, exclusão e alteração de dados. Essas operações também podem ser realizadas na estrutura à fim de adequá-las a novos cenários ou corrigir decisões anteriores.

Por sua vez, a criação das estruturas se dá através da modelagem do BD, que a partir de elementos e suas características no ambiente de estudo, reúne em conjuntos que podem ou não apresentar relacionamentos entre si e que possuem atributos e regras definidos.

Para Cougo (1997), a modelagem dos dados é realizada de acordo com a observação dos objetos, entendimento dos conceitos, representação dos objetos, verificação de coerência e por fim, validação do modelo. A abordagem para representação desse modelo pode ser conceitual, lógica ou física.

Referindo-se ao modelo conceitual, Cougo (1997) afirma que os objetos, suas características e seus relacionamentos devem ser representação fiel ao ambiente observado. Logo, o modelo conceitual é o de maior nível de abstração, mais próximo do usuário comum, provendo fácil entendimento do sistema.

Figura 13 – Exemplo de diagrama Entidade-Relacionamento



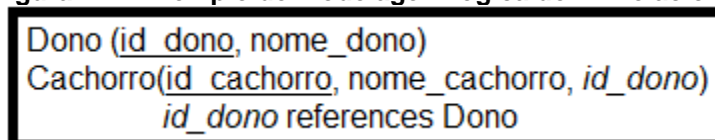
Fonte: Autora (2019).

A figura 13 exibe um modelo conceitual representado em diagrama com a técnica Entidade-Relacionamento, onde o cenário representado permite que cada objeto da entidade “cachorro” pertença a um dono e cada objeto da entidade “dono” tenha vários cachorros, com ambas entidades possuindo como atributos um nome e um código de identificação – o *id*.

Observa-se que cada quadrado significa uma entidade ou classe e cada losango um relacionamento entre as entidades, com a numeração indicando a ordem de relação, de 1 a *n*. Observa-se também a presença de atributos ligados às entidades por círculos alinhados, onde o círculo grafado significa o atributo chave – aquele que será utilizado na identificação de cada registro como único.

A modelagem lógica, para Heuser (2009), representa a estrutura de dados em um BD conforme vista pelo usuário do SGBD, definindo as tabelas e respectivas colunas contidas nesse BD.

Figura 14 – Exemplo de modelagem lógica de BD relacional



Fonte: Autora (2019).

A figura 14 demonstra a modelagem lógica de um BD projetada a partir da modelagem conceitual da figura 13, que evidencia existência de uma tabela “Dono”, com atributos “id_dono” e “nome_dono”, e uma tabela “Cachorro” com atributos “id_cachorro”, “nome_cachorro” e “id_dono”.

O atributo “id_cachorro” e “id_dono” em suas respectivas tabelas indicam a chave primária para identificação única de determinado registro e o atributo “id_dono” na tabela “Cachorro” indica a chave estrangeira, que referencia o relacionamento com a tabela “Dono”, apontando para identificação de um registro específico contido nela.

Só foi possível realizar referência dentro de uma tabela já existente devido à presença de um relacionamento que aceitasse apenas um registro para cada. Caso o relacionamento fosse de ordem n para n – vários se relacionam com vários – seria necessário criação de uma nova tabela como intermediária entre as duas primeiras, contendo as identificações de ambas para realizar ligação entre os registros.

Acerca do modelo físico, Heuser (2009) afirma que é um processo contínuo para detalhamento, influenciando no desempenho e ocorrendo mesmo após implementação do BD, além de ser totalmente dependente do SGBD utilizado.

Figura 15 – Exemplo de modelagem física de uma tabela

```
CREATE TABLE Cachorro (  
  id_cachorro INT PRIMARY_KEY NOT NULL AUTO_INCREMENT,  
  nome_cachorro VARCHAR(80)  
  id_dono INT,  
  FOREIGN KEY(id_dono) REFERENCES Dono (id_dono)  
);
```

Fonte: Autora (2019).

A figura 15 ilustra a criação da tabela “Cachorro” com seus respectivos atributos na linguagem de programação SQL. É feita a declaração dos tipos de cada atributo – e tamanho caso necessário, além da caracterização do funcionamento de determinados atributos, como não nulo, auto incrementável, chave primária ou estrangeira.

Na categoria de servidores de BDs de código aberto destaca-se o *MySQL*. Sendo completo, robusto, extremamente rápido e apresentando todas funções apresentadas por servidores de BDs pagos existentes no mercado, o *MySQL* torna possível a migração de negócios para internet (MILANI, 2006). O *MySQL* é um SGBD relacional que faz uso de *Structured Query Language* para interação do BD.

4.3.1.1 Linguagem de Programação SQL

Com a ascensão do desenvolvimento de SGBDs, tornou-se necessário especificar um padrão de linguagem de acesso, então, em 1986, foi lançada a primeira

versão da linguagem SQL, por colaboração do *International Organization for Standardization* e do *American National Standards Institute* (COSTA; 2006). Assim como qualquer padronização, essa foi fundamental para que ocorresse disseminação da prática de SGBDs.

Como o próprio nome sugere, SQL é uma linguagem de comandos estruturados. Segundo Costa (2006), essa linguagem pode ser dividida em *Data Manipulation Language* e *Data Definition Language*, sendo a primeira responsável pelos comandos relacionados à manipulação dos dados e a última pela definição e criação de estruturas e objetos no BD, como tabelas.

Cada dado é associado a uma coluna em determinado registro da tabela do BD e todos os dados possuem tipo definido na criação da estrutura. O padrão da linguagem SQL apresenta como principais tipos de dados os exibidos no quadro 1 (COSTA; 2006).

Quadro 1 – Principais tipos de dados na linguagem SQL

Tipo de Dados	Descrição
CHARACTER	Caractere de tamanho fixo – usualmente conhecido como CHAR
CHARACTER VARYING	Caractere de tamanho variante – usualmente conhecido como VARCHAR
CHARACTER LARGE OBJECT	Caractere longo – usualmente conhecido como CLOB
BINARY LARGE OBJECT	String binária para objetos longos – usualmente conhecido como BLOB
NUMERIC	Númérico exato
DECIMAL	Númérico exato
SMALLINT	Númérico exato
INTEGER	Númérico exato
BIGINT	Númérico exato
FLOAT	Númérico aproximado
REAL	Númérico aproximado
DOUBLE PRECISION	Númérico aproximado
BOOLEAN	Booleano
DATE	Data com informações de dia, mês e ano
TIME	Hora com informações de hora, minuto e segundo
TIMESTAMP	Determina um momento, com informações de ano, mês, dia, hora, minuto e segundo

Fonte: Costa (2006).

Em suma, a sub linguagem DDL proporciona capacidade de criação de bancos, esquemas, campos e tabelas – como na figura 15 – para conter registros com seus respectivos atributos e características, além da exclusão ou alteração dos mesmos,

enquanto a sub linguagem DML dispõe da capacidade de realizar comandos de manipulação nas estruturas já criadas, interagindo com os dados.

Figura 16 – Exemplo de comandos de DML

```
INSERT INTO Dono(nome_dono) VALUES("Joana")  
UPDATE Dono SET nome_dono = "Maria" WHERE id_dono = 1  
DELETE FROM Dono WHERE id_dono = 2
```

Fonte: Autora (2019).

Na figura 16 temos exemplificações de comandos de DML. Na primeira linha é realizada a inserção de um novo registro na tabela “Dono”, onde o atributo “nome_dono” será “Joana”. Na segunda linha é realizada atualização do registro cuja identificação única – “id_dono” – é “1” de forma que o atributo de nome passe a ser “Maria”. Por fim, na terceira linha é realizada exclusão do registro onde a identificação é “2”.

Outro comando relevante é o *SELECT*. Há autores que o categorizam como parte da DML (COSTA, 2006), enquanto outros o separam em sua própria sub linguagem denominada *Data Query Language* (SINGH, 2006; ASNANI, 2010; KRIEGEL, 2011). A função exercida por esse comando é a de consultar dados específicos de acordo com condições estipuladas.

Figura 17 – Exemplo de comandos de DQL

```
SELECT * FROM Cachorro  
SELECT nome_dono FROM Dono WHERE id_dono = 5
```

Fonte: Autora (2019).

A figura 17 demonstra dois comandos de DQL. O primeiro comando seleciona todos os atributos da tabela “Cachorro” e os expõe para visualização, enquanto que o segundo comando seleciona apenas o atributo “nome_dono” da tabela “Dono” em um registro onde “id_dono” é igual a “5” e o prontifica para exibição.

4.3.1.2 PhpMyAdmin

O *PhpMyAdmin* é uma efetiva ferramenta de interface gráfica para gerenciamento de *MySQL* (DELISLE, 2009). Ou seja, disponibiliza uma saída visual para facilitar processos de definição, alteração, seleção e exclusão dentro de um BD relacional em servidor *MySQL*.

Para Delisle (2009), se tornou indispensável para desenvolvedores *MySQL* e web devido às facilidades trazidas com a interface gráfica e a necessidade de gerenciamento pela presença constante de informações geradas dinamicamente por aplicações. Sua interface conta com acesso a operações no BD, nos esquemas, nas tabelas, nas relações, entre outras funções essenciais para gerência.

“Deve-se ficar muito claro para o usuário que o *PhpMyAdmin* apenas automatiza algumas das linhas de comando mais frequentes do *MySQL*. ” (MILANI, 2006, p. 85). Sendo assim, todas as funcionalidades exercidas nele poderiam ser exercidas na sua ausência, pelo terminal do servidor, porém não apresentaria a mesma facilidade e praticidade.

4.3.2 Regra de Negócio

Antigamente consideradas apenas restrições no BD (APPLETON, 1984), atualmente as regras de negócio são definidas mais amplamente como condições que regem um evento (HALLE, 2001) ou uma declaração relacionada ao negócio de uma organização que pode ser expressa de forma acessível a todos as partes interessadas (MORGAN, 2002).

De acordo com Wagner (2005), existem diversos tipos de regras de negócio:

- Regras de integridade, que restringem unicidade na base de dados;
- Regras de derivação, que inferem fatos acerca de um dado a partir de outros já conhecidos;
- Regras de reação, que executam uma passagem para avaliação de um dado a partir de condições, culminando em uma ação caso a condição seja cumprida.
- Regras de ação, que executam uma passagem para avaliação de um dado sem especificar condições, culminando em uma ação.
- Regras de transformação, que restringem as alterações na base de forma lógica, impedindo mudanças impossíveis no mundo real.

No ambiente de Tecnologia da Informação, essas regras podem ser traduzidas com implementação no próprio BD – prática atualmente em desuso – ou a partir de linguagens de programação compondo o controlador da aplicação.

4.3.2.1 Linguagem de Programação PHP

PHP é uma linguagem de *script* para desenvolvimento web. Iniciou-se como linguagem estruturada e mais tarde foi atualizada com suporte ao paradigma de orientação a objetos, oferecendo recursos que permitiam desenvolvimento de aplicações mais robustas (MINETTO, 2007).

Com a possibilidade de utilizar classes e objetos, foram surgindo *frameworks* de desenvolvimento baseados nessa linguagem. Segundo Minetto (2007), *frameworks* são ferramentas base para desenvolvimento de um software maior ou mais específico, contando com uma coleção de códigos-fonte, classes, funções, técnicas e metodologias para facilitar esse desenvolvimento.

4.3.2.2 Framework Laravel

Um dos mais populares e comumente utilizados *frameworks* em linguagem PHP, o Laravel reutiliza e atrela componentes existentes para prover uma camada na qual uma aplicação web pode ser construída de forma estruturada. Oferecendo um conjunto de ferramentas que incorporam funcionalidades provindas de outros *frameworks* (BEAN, 2015).

Lançado em 2011, o Laravel é um dos *frameworks* mais compactos para linguagem PHP que faz uso do padrão de arquitetura MVC (MINETTO, 2007). Conclui-se então, que é uma ferramenta de base para facilitar codificação em PHP que realiza separação entre as camadas de interface, controlador e modelo.

A documentação de uso é disposta didaticamente em seu próprio site, além de conter uma seção de tutoriais em vídeo para fácil aprendizagem. Em destaque, utiliza implementação da linguagem de *template Blade* para criação de interfaces gráficas e de modelos *Eloquent* para conectar-se com BD, onde cada tabela é representada por um modelo.

Foi projetado para incrementar a qualidade do seu software, reduzindo o custo inicial de desenvolvimento e o custo progressivo da manutenção, e para melhorar a experiência de trabalhar com suas aplicações por prover sintaxe limpa e expressiva e um núcleo de funcionalidades que irá salvar horas de tempo de implementação (MCCOOL, 2012, p. 3, tradução minha)

Logo, a capacidade de reaproveitar funcionalidades já implementadas e a utilização de *templates* e modelos faz com que o Laravel possa ser utilizado no processo de construção de aplicações web seguras, como sites e serviços, com maior agilidade.

Bean (2015) define que a linguagem de *templates* do Laravel, denominada *Blade*, busca reduzir o número de dígitos presentes em comandos frequentemente usados e tornar mais fácil a leitura da sintaxe do código, conforme evidenciado no quadro 2.

Quadro 2 – Comparação da sintaxe no padrão PHP e *Blade*

Standard PHP Syntax	Blade Syntax
<?php echo \$var; ?>	{!! \$var !!}
<?php echo htmlentities (\$var); ?>	{{ \$var }}
<?php if (\$cond) : ?>...<?php endif; ?>	@if (\$cond) ... @endif

Fonte: Bean (2015).

O *Eloquent* é a técnica do Laravel para mapear dados de programação orientada a objetos a BDs relacionais. Agindo como parte da camada de modelo dentro das aplicações, o *Eloquent* possibilita criação de relacionamentos, leitura, escrita e realização de buscas no BD (BEAN, 2015). Com o mapeamento sendo realizado automaticamente pelo *framework* em acordo com os nomes de modelos e tabelas, o programador pode desfrutar de menos preocupações no *Back-end* da aplicação.

Além de possibilitar desenvolvimento de interface gráfica, de modelagem de dados e exportação de tabelas, o Laravel também pode ser utilizado para criação e disposição de serviços web. Serviços web são a disponibilização de funcionalidades através de interfaces via internet (NUNES; DAVID, 2005).

Em outros termos, o Laravel é capaz de disponibilizar informações em páginas web, tendo hospedagem em um servidor, para que essas informações possam ser captadas por outros dispositivos e utilizadas em outras aplicações adjacentes.

4.4 DESENVOLVIMENTO FRONT-END

Front-end é a porção da aplicação que o usuário final interage e visualiza (CHAN, 2016). Sendo assim, pode ser definido como a interface gráfica da aplicação, cujos componentes estão disponíveis para interação direta do usuário através do toque ou visão.

Em bons costumes, a interface é modelada previamente na etapa de design do aplicativo. Porém, essa etapa não apresenta funcionalidade real por trás dos componentes visuais, sendo apenas uma representação do que se pretende obter. A interface contida de funcionalidades e ações concretas é confeccionada a partir de linguagens de codificação.

4.4.1 Desenvolvimento Android

O Android é um ambiente de software escrito para dispositivos móveis, incluindo um sistema operacional, uma rica interface de usuário, aplicativos nativos, bibliotecas de código, suporte a multimídia, entre outros (ABLESON *et al.*, 2012). Com isso, abre-se a possibilidade de desenvolver aplicativos para esse ambiente que faça uso dos recursos disponíveis nele.

O Android Studio é o Ambiente de Desenvolvimento Integrado (IDE) especificamente criado pela Google para criação de aplicativos para Android, fazendo uso da linguagem de programação Java para codificação. Devido à estabilidade e outras vantagens dessa IDE, outros modos de desenvolvimento Android caíram em desuso, como a programação de aplicações na IDE Eclipse (RAJPUT, 2015).

Segundo Cruz Zapata (2013), o sistema operacional Android sempre esteve em competição direta com o sistema operacional da Apple, o iOS, que se apresentava com ambiente de desenvolvimento mais centralizado. Contudo, com o advento da IDE Android Studio, essa centralização finalmente se torna disponível para desenvolvedores Android. Sendo uma plataforma de código aberto, encoraja desenvolvedores à criarem *plug-ins* e extensões que podem ser inseridos e utilizados em outros projetos, contando com uma alta participação da comunidade.

Dentre as vantagens, além da maior estabilidade apresentada, é relevante citar o sistema de autopreenchimento de funções e variáveis durante digitação de código, organização do projeto com divisões em espaço de trabalho, presença de Interface Gráfica do Utilizador específica e amigável e a integração com o *Gradle*, ferramenta de compilação que auxilia na configuração e inserção de itens no projeto.

5 PROPOSTA DA APLICAÇÃO

Este capítulo visa explicitar o processo de criação da solução proposta pelo trabalho, desde a fase inicial de planejamento até o funcionamento final do aplicativo. Serão explicados aspectos do modelo de negócios concebidos especificamente para essa aplicação, os requisitos levantados e analisados, além de dispor capturas de tela do produto final.

5.1 PLANEJAMENTO

Na etapa de planejamento do software, foi necessário criar a concepção de como se pretendia alcançar a solução proposta, visando obter o molde para um produto viável antes de iniciar o desenvolvimento do mesmo. Para isso, foi empregado o uso de um mapa visual de modelo de negócios, inserido no Apêndice A, buscando definir delimitação do escopo do aplicativo.

5.1.1 Requisitos

Pós definição do escopo, foi realizado o levantamento dos requisitos pertinentes à aplicação e respectiva análise de cada um. Os requisitos foram criados levando em consideração dois cenários: O primeiro cenário se refere às necessidades de usuários que desejam colocar animais para adoção – identificado na documentação por “padrinho” – e o segundo cenário se refere às necessidades de usuários que queiram adotar animais na plataforma – referenciado na documentação por “amiguinho”.

Encontram-se abaixo dois requisitos destaque para a aplicação em formato de *User Story*, que é a descrição concisa da necessidade de acordo com o usuário inserido no cenário:

- Como padrinho, gostaria de cadastrar animal;
- Como amiguinho, gostaria de visualizar todos os animais cadastrados por ordem de mais recente.

O primeiro requisito apresentado evidencia a necessidade de registrar um animal na plataforma para que ele seja exibido como animal disponível a outros usuários, enquanto o segundo requisito refere-se à necessidade de visualizar todos animais registrados disponíveis para adoção, em ordem cronológica. A documentação que consta a totalidade dos requisitos levantados para o aplicativo encontra-se inserida no Apêndice B.

5.1.2 Arquitetura

O padrão de arquitetura empregado é o MVC, implementado na seguinte forma durante o desenvolvimento:

- *Model* – Uso de classes *Model* construídas no *framework* Laravel para realizar mapeamento dos objetos às tabelas de dados presentes no BD, permitindo acesso e manipulação.
- *View* – Interface gráfica implementada por codificação XML e programaticamente em Java para Android na IDE Android Studio;
- *Controller* – Controlador de ações programado através do uso de classes *Controller* no Laravel com funções capazes de realizar alterações de informações nos modelos e na interface; Chamadas de serviço realizadas por programação para Android à fim de receber, enviar e atualizar informações providas do servidor.

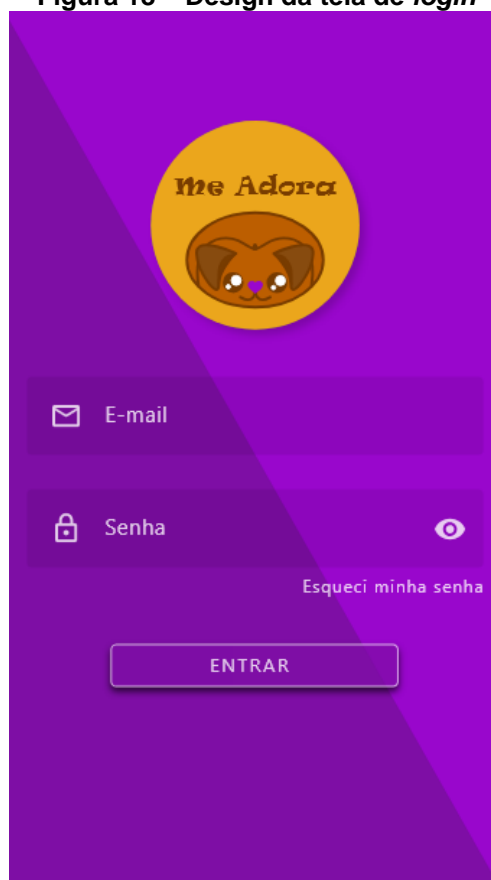
Dessa maneira, o controlador interage com o modelo e a interface para realizar modificações e outras ações requeridas, enquanto a interface tem capacidade de interagir com o modelo de forma a exibir determinados dados contidos nele.

5.1.3 Design

O design do aplicativo foi efetuado por meio da ferramenta Adobe XD, aproveitando-se dos pacotes de componentes disponíveis para criar as telas com visual análogo ao normalmente encontrado nas aplicações Android populares.

Na figura 18 encontra-se o design da tela de *login* da aplicação, constando de componentes como caixas de texto, ícones e botões. É possível observar prevalência de tonalidades das cores roxo e dourado.

Figura 18 – Design da tela de login



Fonte: Autora (2019).

Esse tema de cores foi escolhido com intuito de chamar atenção pelo contraste das cores, que são complementares, e definir esse conjunto de tons como lembrança direta na mente do usuário ao aplicativo. O restante do design das telas encontra-se no Apêndice C.

5.2 DESENVOLVIMENTO

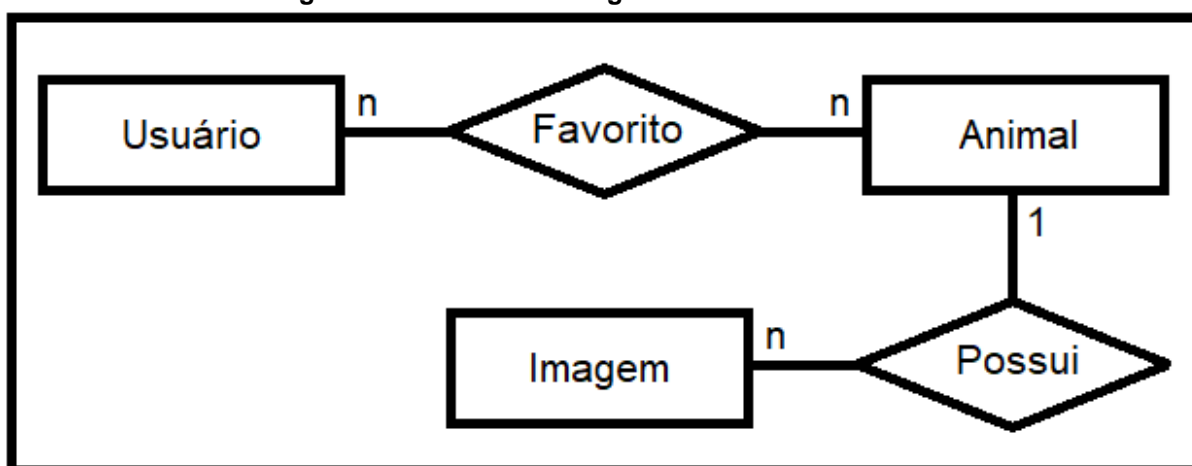
Na etapa de desenvolvimento do software, iniciou-se a transformação dos aspectos de negócios, requisitos e design propostos em produto concreto através de programação e de ferramentas auxiliares.

5.2.1 Banco de Dados

A modelagem do BD foi realizada levando em consideração elementos importantes que se encontrarão presentes na aplicação – como os animais e os usuários – e funcionalidades existentes que necessitem guardar dados – como a ação de tornar animal favorito ou denunciar usuário.

Uma parte da modelagem encontra-se exibida na figura 19, evidenciando as entidades “Usuário”, “Animal”, “Imagem” e as relações entre si. A leitura dessa modelagem permite inferir que um usuário pode marcar como favorito vários animais e um animal pode ser marcado como favorito por vários usuários. Também permite inferir que um animal pode possuir diversas imagens, porém uma imagem pode pertencer à apenas um animal.

Figura 19 – Excerto do diagrama de banco de dados



Fonte: Autora (2019).

Após conclusão da modelagem, a construção do BD se efetivou com auxílio de ferramentas de implementação presentes no *framework* Laravel. Ajustes posteriores necessários nessa construção primária foram realizados com comandos SQL no próprio sistema de gerenciamento utilizado – o PhpMyAdmin.

Ao final, há um total de 9 tabelas no esquema, cada uma com diversas colunas referentes às informações que devem ser mantidas acerca de cada registro, incluindo uma chave única de identificação. Além disso, é utilizado também o *Realm*, que é um BD orientado a objetos para aplicações móveis, garantindo persistência dos dados no dispositivo e sendo mais enxuto que o BD padrão do Android – o SQLite, que exige muito código *boilerplate*.

5.2.2 Serviço Web

O serviço web é o responsável por tornar visível as informações mantidas no BD para que o aplicativo obtenha acesso às mesmas. O processo de tornar essas informações visíveis envolve exibi-las em uma página web que possa ser acessada pelo dispositivo móvel.

A composição da página para exibição foi feita no Laravel a partir da conexão da mesma com o BD e consequente permissão de acesso aos dados. Após acessados, uma cópia dos dados é listada na página de forma que outros canais possam realizar leitura.

Figura 20 – Excerto do serviço web

```
"status": 200,
"mensagem": "Animais listados com sucesso!",
"animais": [
  {
    "id": 1,
    "nome": "Rex",
    "descricao": "asd",
    "genero": "Masculino",
    "idade": "Idoso",
    "vacinado": 1,
    "castrado": 1,
    "latitude": 0.1000000000000000055511151231257827021181583404541015625,
    "longitude": 0.299999999999999988897769753748434595763683319091796875,
    "usuario_id": 1,
    "created_at": "2018-11-28 17:01:45",
    "updated_at": "2018-12-04 06:55:27"
  },
]
```

Fonte: Autora (2019).

A figura 20 dispõe um trecho do serviço referente às informações da tabela “Animais”. Neste trecho em particular é possível observar o retorno de um código de estado, em conjunto com uma mensagem que explana o significado do código, além de uma ocorrência de registro de um animal com suas respectivas características.

5.2.3 Chamadas de Serviço

As chamadas de serviço são os modos de acesso às cópias proporcionadas pelo serviço web dos dados presentes no BD. Quatro tipos de chamadas de serviço foram utilizados, são estes:

- *GET* – Chamadas que permitem leitura do conteúdo no serviço.
- *POST* – Chamadas que permitem enviar conteúdo para o serviço, que consequentemente é inserido no BD.
- *PUT* – Chamadas que permitem enviar conteúdo para o serviço, visando atualizar um conteúdo previamente existente no BD.
- *PATCH* – Chamadas que permitem enviar um fragmento de conteúdo para o serviço, visando atualizar apenas parte de um conteúdo previamente existente no BD.

- **DELETE** – Chamadas que permitem enviar requisição de exclusão de conteúdo para o serviço, consequentemente excluindo no BD

Figura 21 – Excerto de chamadas de serviço

```
@GET("public/denuncia")
Call<DenunciaJSON> getDenuncia();

@POST("public/denuncia")
@FormUrlEncoded
Call<DenunciaJSON> postDenuncia(@Field("usuario_delator_id") long usuario_delator_id,
                                @Field("usuario_denunciado_id") long usuario_denunciado_id,
                                @Field("descricao") String descricao,
                                @Field("ativo") int ativo);

@PATCH("public/denuncia/{id}")
@FormUrlEncoded
Call<Void> updateDenuncia(@Path("id") long id,
                          @Field("usuario_delator_id") long usuario_delator_id,
                          @Field("usuario_denunciado_id") long usuario_denunciado_id,
                          @Field("descricao") String descricao,
                          @Field("ativo") int ativo);

@DELETE("public/denuncia/{id}")
Call<Void> deleteDenuncia(@Path("id") long id);
```

Fonte: Autora (2019).

Na figura 21 temos as chamadas de serviço utilizadas para o sistema de denúncias. Nessa instância, a chamada *GET* realiza leitura de todos os registros de denúncias aos usuários, a chamada *POST* realiza escrita de um registro, a chamada *PATCH* serve para atualizar registro já existente e a chamada *DELETE* serve para excluir registro.

5.2.4 Telas

A implementação gráfica do aplicativo é composta por diversas telas que seguem modelos propostos na etapa de design. Essas telas foram construídas com base em componentes padrão pré-existentes no Android Studio, que podem ser alocados por meio da linguagem de marcação XML. Na figura 22 é ilustrada a codificação em XML para compor o botão mostrado na parte inferior da imagem como elemento gráfico.

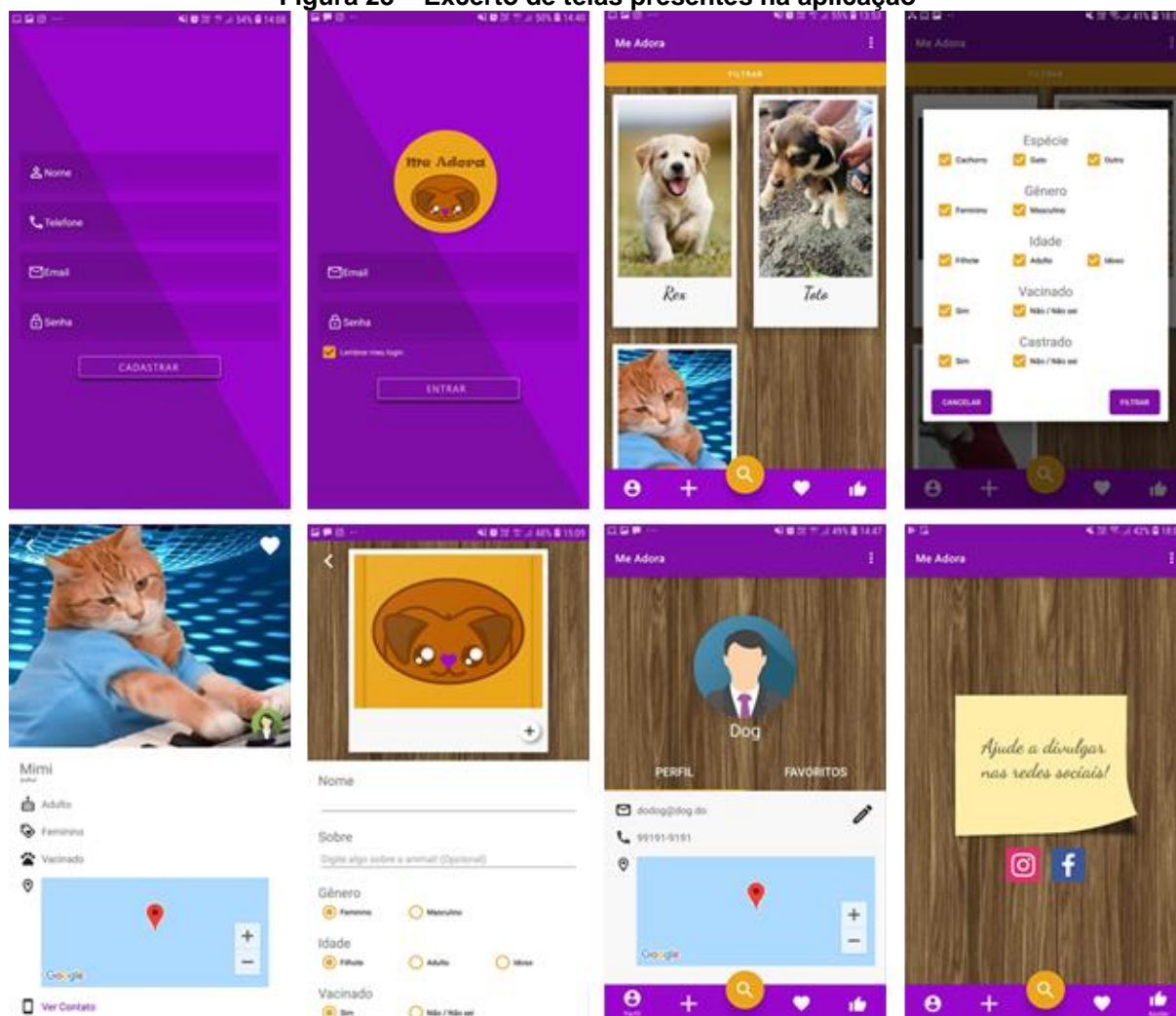
Figura 22 – Excerto de implementação gráfica em XML e respectivo produto



Fonte: Autora (2019).

Após construção da interface gráfica a partir dos componentes, esta é associada à uma *Activity* ou *Fragment*. Uma *Activity* é uma atividade ou janela que será apresentada no dispositivo, enquanto que um *Fragment* é apenas uma porção ou fragmento que pode ser encaixado no interior de uma atividade. Os Apêndices D e E exibem, respectivamente, uma *Activity* e um *Fragment* presentes na codificação da aplicação.

Figura 23 – Excerto de telas presentes na aplicação



Fonte: Autora (2019).

Com a associação da implementação em XML às atividades ou fragmentos, as telas da aplicação se encontram prontas para exibição, se tornando completamente funcionais a partir da comunicação com o *back-end*. Algumas telas presentes no aplicativo estão dispostas na figura 23. São estas, respectivamente, a tela de cadastro, de login, de procura por animais, de filtragem de animais, de informações de animal selecionado, de registro de animais, de perfil de usuário e de divulgação em redes sociais.

6 CONSIDERAÇÕES FINAIS

A adoção de animais como contexto escolhido para inserção do produto entregue neste trabalho é um assunto de relevância há tempos, mas que vem ganhando cada vez mais visibilidade nos últimos anos devido ao grande alcance proporcionado pela internet ao ativismo e campanhas voltadas a esse propósito.

Apesar de haver o enaltecimento da opção de adotar, observa-se que não é suficiente para acompanhar o crescimento do número de animais nas ruas. Com isso, o desenvolvimento da aplicação proposta no trabalho é justificado, visto que prover novos acessos ao processo de adoção ainda é necessário.

6.1 RESULTADOS

A aplicação encontra-se disponível gratuitamente sob o nome de “MeAdora” no serviço de distribuição digital de aplicativos da Google, denominado Play Store.

Analizando a aplicação como resultado do ponto de vista de usuários adotantes, oferece-se um serviço simplificado que possibilita, principalmente, a procura de animais com base em diversos filtros de características, enquanto que do posto de vista de usuários doadores, oferece-se uma plataforma de cadastro de animais sem quantidade limitada por conta, beneficiando organizações e indivíduos igualmente.

Avalia-se que os resultados obtidos ao fim do projeto são de grande valia, tanto na categoria acadêmica, quanto social. Na categoria acadêmica, foi capaz de agregar valor por meio do emprego de tecnologias bastante difundidas no mercado de trabalho à uma aplicação desenvolvida em universidade.

No quesito social, a aplicação contribui com auxílio direto à abrigos de animais, sobretudo aqueles que se encontram em superlotação, e também a indivíduos que precisam de um meio para divulgar animais que procuram lar permanente, além de servir como ferramenta para aqueles que desejam adotar. Acrescenta-se, ainda, que indiretamente a aplicação é capaz de auxiliar no aumento de visibilidade da causa.

6.2 DIFICULDADES ENCONTRADAS

Uma dificuldade encontrada foi a implementação do sistema de doações como descrito no item de fontes de receitas do mapa visual de negócios presente no Apêndice A. Essa dificuldade se deu devido às restrições de leis aplicáveis na distribuição de aplicativos na loja da Google. As restrições encontradas demandam

que apenas em vendas de produtos físicos seja elegível o uso de pagamentos pelo Google Pay, PicPay ou outros meios de monetização terceirizados à Play Store.

Em função da incapacidade de gerar renda financeira de forma direta pela aplicação por meio das doações, o *back-end* encontra-se hospedado em um servidor gratuito, ao contrário do planejado no item de estrutura de custos do mapa visual de negócios no Apêndice A. Este servidor apresenta limitação de quantidade de dados, o que pode afetar no desempenho geral do aplicativo.

6.3 TRABALHOS FUTUROS

Apesar de concluído, o aplicativo pode ser aperfeiçoado de várias formas futuramente. Algumas ideias de aperfeiçoamento foram detectadas durante o desenvolvimento original do projeto, porém apenas após fechamento do escopo, não sendo incluídos nessa versão.

Um aperfeiçoamento de destaque em questão de funcionalidade e praticidade de acordo com a proposta da aplicação é a implementação de filtro por localização, que tornaria mais rápida a pesquisa por animais próximos à área do usuário.

Outra questão importante seria o desenvolvimento de versão para dispositivos com sistema operacional iOS, buscando atingir um número maior de usuários. Mas essa opção exige licença de desenvolvimento e equipamento especializado.

Visando impactar diretamente no dinamismo da aplicação, a possível troca da hospedagem do *back-end* da aplicação para um *host* que ofereça melhores condições como servidor seria outro aperfeiçoamento a ser concretizado, embora dependente de custos financeiros.

Possíveis melhorias que venham a ser identificadas após publicação do aplicativo ainda poderão ser efetuadas, sem alterar o funcionamento principal já desenvolvido, focando na otimização de desempenho ou correção de erros.

7 REFERÊNCIAS

ABLESON, W. Frank *et al.* **Android Em Ação**. Tradução de Eduardo Kraszczuk. 3. ed. Rio de Janeiro: Campus-Elsevier, 2012.

ADOBE. **Vá do wireframe à experiência com mais rapidez**. 2019. Disponível em: <https://www.adobe.com/br/products/xd/wireframing-tool.html>. Acesso em: 09 abr. 2019.

ALVES, Ana Júlia Silva e *et al.* Abandono de cães na América Latina: revisão de literatura. **Revista de Educação Continuada em Medicina Veterinária e Zootecnia do CRMV-SP**. São Paulo: Conselho Regional de Medicina Veterinária, v.11, n.2, p.34-41, 2013.

AMARAL, Rodrigo Augusto do; NERIS, Vânia Paula de Almeida. Análise comparativa entre frameworks de front-end para aplicações web ricas visando reaproveitamento de back-end. **Tecnologias, Infraestrutura e Software**. São Carlos, v.4, n.1, p.88-96, 2015.

AMIGO Não Se Compra. **Encontre Seu Novo Amigo**. Disponível em: <https://www.amigonaosecompra.com.br/?scroll=search>. Acesso em: 06 abr. 2019.

AMIGOS de São Francisco. **Quero Adotar**. 2015. Disponível em: <http://amigosdesaofrancisco.com.br/adotar>. Acesso em: 06 abr. 2019.

ANDA – Agência de Notícias de Direitos Animais. **Brasil tem 30 milhões de animais abandonados**. 2014. Disponível em: <https://anda.jusbrasil.com.br/noticias/100681698/brasil-tem-30-milhoes-de-animais-abandonados>. Acesso em: 22 out. 2018.

APPLETON, Daniel S. Business Rules: The Missing Link. **Datamation**. EUA, v.30, n.16, p.145-150, 1984.

ASNANI, Satish. **Oracle Database 11g: Hands-On SQL and PL/SQL**. New Delhi: PHI Learning Private Limited, 2010.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software Architecture in Practice**. 3. ed. Boston: Addison-Wesley Professional PTG, 2013.

BEAN, Martin. **Laravel 5 Essentials**: Explore the fundamentals of Laravel, one of the most expressive and robust PHP frameworks available. Birmingham, UK: Packt Publishing, 2015.

BRASIL. Decreto-Lei nº 9.605, de 12 de fevereiro de 1998. **Dispõe sobre as sanções penais e administrativas derivadas de condutas e atividades lesivas**

ao meio ambiente e dá outras providências, Poder Executivo, Brasília, DF, 13 fev. 1998. Seção 1, p.1.

CARAM, Bernardo. **Câmara aprova projeto que proíbe órgãos públicos de sacrificar cães e gatos**. 2017. Disponível em:

<https://g1.globo.com/politica/noticia/camara-aprova-projeto-que-proibe-orgaos-publicos-de-sacrificar-caes-e-gatos.ghtml>. Acesso em: 04 abr. 2019.

CHAN, Iana. **O que é front-end e back-end**. 2016. Disponível em:

<https://www.programaria.org/o-que-e-front-end-e-back-end/>. Acesso em: 10 abr. 2019.

CIOTTI, Gregory. **The Psychology of Color in Marketing and Branding**. 2014.

Disponível em:

http://www.usailighting.com/stuff/contentmgr/files/1/2f6fafec517a66b6c3857c06b2bfff/misc/psychologycolormarketingbranding_entrepreneur.pdf. Acesso em: 07 abr. 2019.

COSTA, Rogério Luís de Carvalho. **SQL: Guia Prático**. 2. ed. Rio de Janeiro: Brasport, 2006.

COUGO, Paulo. **Modelagem Conceitual e Projeto de Banco de Dados**. Rio de Janeiro: Elsevier, 1997.

CRUZ ZAPATA, Belén. **Android Studio Application Development: Create visually appealing applications using the new IntelliJ IDE Android Studio**. Birmingham, UK: Packt Publishing, 2013.

DA VINCI, Leonardo. **El Tratado de la Pintura**. 1651. Traducido por Diego Antonio Rejón de Silva. Madrid: Imprenta Real, 1784.

DELISLE, Marc. **Mastering phpMyAdmin 3.1 for Effective MySQL Management**. Birmingham, UK: Packt Publishing, 2009.

DICKMAN, Chris R. **Overview of the impacts of feral cats on Australian native fauna**. Prepared for the Australian Nature Conservation Agency – Institute of Wildlife Research and School of Biological Sciences, University of Sydney. Sydney, 97p. 1996.

ENGHOLM JÚNIOR, Hélio. **Engenharia de Software na Prática**. São Paulo: Novatec, 2010.

FONSECA, António Castro; Dias, Sofia Salgado. O Problema da Crueldade Contra Animais na Infância: Suas Dimensões e Consequências. **Revista Portuguesa de Pedagogia**. Coimbra, vol.45, n.2, p.71-92, 2011.

G1. **Polícia fecha canil com centenas de cães em Piedade**. 2019. Disponível em: <https://g1.globo.com/sp/sorocaba-jundiai/noticia/2019/02/13/policia-fecha-canil-clandestino-com-centenas-de-caes-em-piedade.ghtml>. Acesso em: 04 abr. 2019.

GOETHE, Johann Wolfgang von. **Theory of Colours**. 1810. Translated by Charles Lock Eastlake. London, UK: John Murray, 1840.

HALLE, Barbara von. **Business Rules Applied: Building Better Systems Using the Business Rules Approach**. New Jersey: Wiley Computer Publishing, 2001.

HART, Benjamin L. **The behaviour of Domestic Animals**. New York: W.H Freeman and Company, 1985.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009.

IBGE; Coordenação de Cadastro e Classificações. **As Fundações Privadas e Associações Sem Fins Lucrativos No Brasil 2016**. Rio de Janeiro: IBGE, 2019.

IBGE; Coordenação de Trabalho e Rendimento. **Pesquisa Nacional de Saúde 2013: Acesso e utilização dos serviços de saúde, acidentes e violências**. Rio de Janeiro: IBGE, 2015.

IBGE; Coordenação de Treinamento e Aperfeiçoamento. **Pesquisa Domiciliar sobre Cães e Gatos: Humanização e Padrões de Consumo**. Rio de Janeiro: IBGE, 2007.

IEEE; **IEEE 610.12-1990** – IEEE Standard Glossary of Software Engineering Terminology. Institute of Electrical and Electronics Engineers Inc, 1990.

KONECKI, Krzysztof. Pets of Konrad Lorenz: Theorizing in the social world of pet owners. **Qualitative Sociology Review**, vol.3, n.1, p.110-127, 2007.

KRIEGLER, Alex. **Discovering SQL: A Hands-On Guide for Beginners**. Indianapolis, Indiana: Wiley Publishing, 2011.

KOTAIT, Ivanete *et al.* Reservatórios silvestres do vírus da raiva: um desafio para a saúde pública. **Boletim Epidemiológico Paulista**, vol.4, n.40, p.2-8, 2007.

LARAVEL. **Love beautiful code? We do too.** Disponível em: <https://laravel.com>. Acesso em: 10 abr. 2019.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: Uma Visão Prática.** 17. ed. São Paulo: Érica, 2012.

MCCOOL, Shawn. **Laravel Starter:** the definitive introduction to the Laravel PHP web development framework. Birmingham, UK: Packt Publishing, 2012.

MEIRELLES, Fernando S. **Pesquisa Anual do Uso de TI nas Empresas.** 29ª ed. São Paulo: FGV-EAESP, 2018.

MICROSOFT. **Model-View-Controller.** 2014. Disponível em: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649643\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649643(v=pandp.10)). Acesso em: 07 abr. 2019.

MILANI, André. **MySQL:** Guia do Programador. São Paulo: Novatec, 2006.

MINETTO, Elton Luís. **Frameworks para Desenvolvimento em PHP.** São Paulo: Novatec, 2007.

MORGAN, Tony. **Business Rules and Information Systems:** Aligning IT with Business Goals. Boston: Addison-Wesley, 2002.

NEIL, Theresa. **Padrões de Design para Aplicativos Móveis.** São Paulo: Novatec, 2012.

NEWTON, Isaac. A letter of Mr. Isaac Newton, professor of the Mathematicks in the University of Cambridge; containing his new theory about light and colours; sent by the author to the publisher from Cambridge, Febr. 6. 1671/72; in order to be communicated to the R. Society. **Philosophical Transactions of the Royal Society.** London, v.6, n.80, p.3075-3087, 1671.

NUNES, Sérgio; DAVID, Gabriel. **Uma Arquitetura Web para Serviços Web.** 2005. Disponível em: <https://hdl.handle.net/10216/281>. Acesso em: 10 abr. 2019.

O LIBERAL. **Abandono ameaça 30 mil cachorros e gatos em Belém.** 2016. Disponível em: <http://www.ormnews.com.br/noticia/abandono-ameaca-30-mil-cachorros-e-gatos-em-belem>. Acesso em: 18 fev. 2019.

OSÓRIO, Andréa. A Cidade e os animais: da modernização à posse responsável. **Teoria & Sociedade.** Minas Gerais, v.21, n.1, p.143-176, 2013.

OLIVEIRA, Samantha Brasil Calmon de. **Sobre Homens e Cães**: Um estudo antropológico sobre afetividade, consumo e distinção. Dissertação (Mestrado em Sociologia e Antropologia) – PPGSA/IFCS, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2006

PAULA FILHO, Wilson de Pádua. **Engenharia de Software**: fundamentos, métodos e padrões. Rio de Janeiro: LTC Livros Técnicos e Científicos Editora S/A, 2001.

PEA. **Cresce para 30 milhões o número de animais abandonados no Brasil, afirma OMS**. 2014. Disponível em: <http://www.pea.org.br/news/noticias01.asp?noticia=6968>. Acesso em: 18 fev. 2019.

PIXABAY. **Color's Spectrum Element Design**. 2012. Disponível em: <https://pixabay.com/vectors/colors-spectrum-element-design-35763/>. Acesso em: 07 abr. 2019.

PRESSMAN, Roger. S. Engenharia de software. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Software Engineering**: A Practitioner's Approach. 8. ed. New York, NY: McGraw Hill Global Education Holdings LLC, 2015.

PROCURE 1 Amigo. **Cães para Doação em PA - Belém**. Disponível em: <https://www.procure1amigo.com.br/animais.aspx?cc=2413&cn=pa-belem&tipo=Cao&finalidade=Adocao>. Acesso em: 06 abr. 2019.

PROJETO Literacia. **Classificação em cores em biblioteca infantil**. 2013. Disponível em: <https://projetoliteracia.wordpress.com/2013/12/06/classificacao-em-cores-em-biblioteca-infantil/>. Acesso em: 07 abr. 2019.

RAJPUT, Mehul. **Why Android Studio Is Better For Android Developers Instead Of Eclipse**. 2015. Disponível em: <https://dzone.com/articles/why-android-studio-better>. Acesso em: 29 out. 2018.

RANGEL, Cristiane Hollanda; NEIVA, Carla Helena Mendes Bunn. Predação de Vertebrados por Cães Canis lúpus familiaris (Mammalia: Carnivora) no Jardim Botânico do Rio de Janeiro, RJ, Brasil. **Biodiversidade Brasileira**. Rio de Janeiro, v.3, n.2, p.261-269, 2013.

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. **Interaction Design**: Beyond Human-Computer Interaction. 3. ed. New Jersey: John Wiley & Sons Ltd, 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN S. **Sistema de Banco de Dados**. Tradução de Daniel Vieira. 6. ed. Rio de Janeiro: Elsevier, 2012.

SINGH, S. K. **Database Systems**: Concepts, Design and Applications. India: Dorling Kindersley, 2006.

SPAC. Adote. **SPACuritiba**, 2018. Disponível em: <http://spacuritiba.org.br/adote/>. Acesso em: 04 abr. 2019.

STAFFORD, Kevin. **The Welfare of Dogs**. 4. ed. Netherlands: Springer Netherlands, 2006.

SUTHERS-McCABE, Marie. Take one pet and call me in the morning. **Generations**. Califórnia, vol.25, n.2, p.93-95, 2001.

TATIBANA, Lilian Sayuri; COSTA-VAL, Adriane Pimenta da. Relação homem-animal de companhia e o papel do médico veterinário. **Revista Veterinária e Zootecnia em Minas**. Minas Gerais, ano XXVIII, n.103, p.12-18, 2009


WAGNER, G. Rule modeling and markup. In: **Reasoning Web**, Lecture Notes in Computer Science, v.3564, p.251-274. Heidelberg: Springer-Verlag Berlin, 2005.

YAMAUCHI Apps. **Adota Aqui** - Adoção de Cães e Gatos. 2018. Disponível em: https://play.google.com/store/apps/details?id=com.yamauchi.adotaaqui&hl=pt_BR. Acesso em: 06 abr. 2019.

APÊNDICE A – Mapa Visual de Modelo de Negócios

<p>8</p> <p>Parcerias Principais</p> <p>Possíveis parcerias com Abrigo Au Family e Projeto Peludinhos da UFPA.</p>	<p>7</p> <p>Atividades Principais</p> <p>Desenvolvimento e manutenção do <i>app</i>. Criação e verificação do BD e serviço que permitirá acesso aos dados contidos. Publicação do <i>app</i> na loja virtual. Divulgação do <i>app</i> e publicação das atividades realizadas pelo mesmo.</p>	<p>2</p> <p>Proposta de Valor</p> <p>Plataforma para prover a experiência de conectar usuários que queiram adotar animais e usuários que tenham animais para adoção de forma facilitada remotamente, com capacidade de verificar informações relativas em qualquer tempo e qualquer espaço com conexão à internet.</p>	<p>4</p> <p>Relacionamento com Clientes</p> <p>Sistema de cadastro de usuário ou vinculação com conta Google para verificação. Esquema de denúncia à usuário. Espaço para relatar história e foto do animal escolhido pós adoção.</p>	<p>1</p> <p>Segmento de Clientes</p> <p>Usuários portadores de <i>smartphones</i> com necessidade de botar animais domésticos à adoção. Usuários que desejem realizar adoção de animais.</p>
<p>9</p> <p>Estrutura de Custos</p> <p>Licença de desenvolvedor Android para publicação. Energia elétrica gasta para alimentar computador e celular utilizado. Custo homem-hora de programador. Custo auxílio jurídico. Hospedagem <i>Web</i>.</p>		<p>5</p> <p>Fontes de Receitas</p> <p>Doações realizadas através de pagamento em Google Pay ou PicPay.</p>		
<p>6</p> <p>Recursos Principais</p> <p>Computador capaz de executar Android Studio e <i>smartphone</i> capaz de emular <i>app</i> Android. Programador de <i>app</i> Android, Database Admin e programador de serviço em <i>Framework</i> Laravel. Profissional para auxílio jurídico. <i>Web Hosting</i>.</p>		<p>3</p> <p>Canais</p> <p>Aplicativo gratuito distribuído por loja virtual do sistema operacional <i>mobile</i> Android.</p>		

APÊNDICE B – Documentação de Requisitos do Projeto

	TÍTULO DO DOCUMENTO	AREA	CÓDIGO
	LISTA DE REQUISITOS DO PROJETO <i>(Product Backlog)</i>	Mobile	2018.2
			VERSÃO 0.1

1 – Histórico de Mudança


Versão	Data	Descrição	Autor
0.1	18/10/2018	Edição primária do documento	Flávia Soriano de Mello Barroso

2 – Objetivo do Projeto


Informatizar o processo de disponibilização e adoção de animais domésticos através de plataforma *mobile*, possibilitando acesso à aplicação e suas atividades a qualquer tempo e em qualquer local com conexão à internet.

3 – Requisitos


User Story	Observação	Responsável	Prioridade (Sprint)	Finalizada
1- Como padrinho, gostaria de poder efetuar cadastro.		Time	0	Não
2- Como padrinho, gostaria de poder efetuar login.		Time	0	Não
3- Como padrinho, gostaria de poder visualizar meu perfil.		Time	2	Não
4- Como padrinho, gostaria de poder editar meu perfil.		Time	2	Não

	TÍTULO DO DOCUMENTO	AREA	CÓDIGO
	LISTA DE REQUISITOS DO PROJETO <i>(Product Backlog)</i>	Mobile	2018.2
			VERSÃO 0.1

5- Como padrinho, gostaria de poder desativar meu perfil.		Time	2	Não
6- Como padrinho, gostaria de cadastrar animal.		Time	0	Não
7- Como padrinho, gostaria de visualizar animais cadastrados por mim.		Time	0	Não
8- Como padrinho, gostaria de editar cadastro de animal.		Time	1	Não
9- Como padrinho, gostaria de excluir cadastro de animal.		Time	1	Não
10- Como padrinho, gostaria de poder realizar doações.		Time	2	Não
11- Como amiguinho, gostaria de poder efetuar cadastro.		Time	0	Não
12- Como amiguinho, gostaria de poder efetuar login.		Time	0	Não
13- Como amiguinho, gostaria de poder visualizar meu perfil.		Time	2	Não

	TÍTULO DO DOCUMENTO	AREA	CÓDIGO
	LISTA DE REQUISITOS DO PROJETO <i>(Product Backlog)</i>	Mobile	2018.2
			VERSÃO 0.1

14- Como amiguinho, gostaria de poder editar meu perfil.		Time	2	Não
15- Como amiguinho, gostaria de poder desativar meu perfil.		Time	2	Não
16- Como amiguinho, gostaria de visualizar todos os animais cadastrados por ordem de mais recente.		Time	0	Não
17- Como amiguinho, gostaria de visualizar animais por cuidados veterinários.		Time	3	Não
18- Como amiguinho, gostaria de visualizar animais por categoria de faixa etária.		Time	3	Não
19- Como amiguinho, gostaria de visualizar animais por categoria de gênero.		Time	3	Não
20- Como amiguinho, gostaria de visualizar informações sobre animal selecionado.		Time	0	Não

	TÍTULO DO DOCUMENTO	AREA	CÓDIGO
	LISTA DE REQUISITOS DO PROJETO <i>(Product Backlog)</i>	Mobile	2018.2
			VERSÃO 0.1

21- Como amiguinho, gostaria de salvar animal como favorito.		Time	1	Não
22- Como amiguinho, gostaria de visualizar meus animais favoritos.		Time	1	Não
23- Como amiguinho, gostaria de excluir animal de favoritos.		Time	1	Não
24- Como amiguinho, gostaria de poder realizar depoimento pós adoção.		Time	2	Não
25- Como amiguinho, gostaria de poder visualizar depoimentos pós adoção.		Time	2	Não
26- Como amiguinho, gostaria de poder denunciar padrinhos.		Time	1	Não
27- Como amiguinho, gostaria de poder realizar doações.		Time	2	Não

Considerações

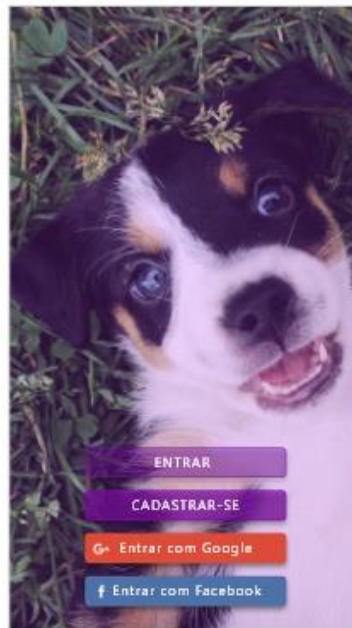
Buscando facilitar a documentação, por “padrinho” identificam-se os usuários dispostos a colocar animais para adoção e por “amiguinho” identificam-se os usuários dispostos a adotar animais.

APÊNDICE C – Design das Telas do Aplicativo

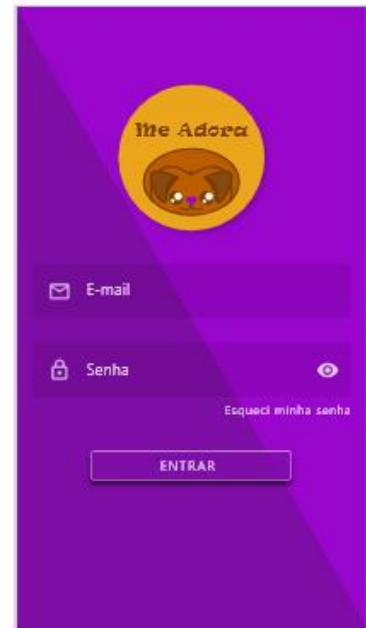
Splash Screen



Initial Screen



Login Screen



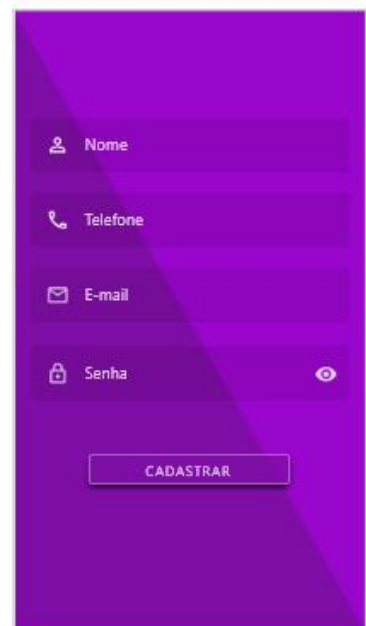
Home Screen



Selected Screen



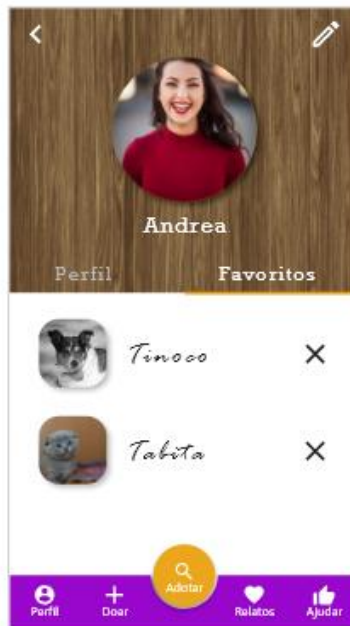
Sign Up Screen – 1



Profile Screen



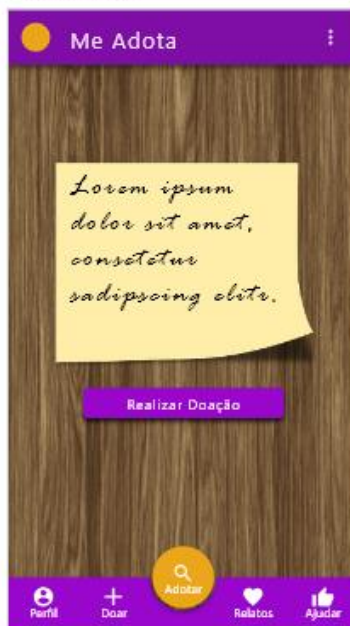
Favorites Screen



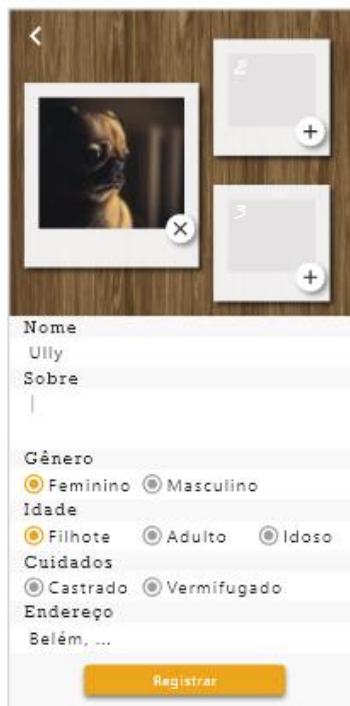
My Registers Screen



Donate Screen



Add Screen



APÊNDICE D – Codificação de uma Activity

```

import android.Manifest;
import android.annotation.TargetApi;
import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageButton;
import android.widget.Toast;
import com.google.android.material.snackbar.Snackbar;

public class InitialActivity extends AppCompatActivity {
    private static final int REQUEST_SEND_SMS = 0;
    private static final int REQUEST_RECEIVE_SMS = 0;
    private int SMSPermission = 0;

    ImageButton btnLogin;
    ImageButton btnSignUp;

    void bindComponents() {
        btnLogin = (ImageButton) findViewById(R.id.btnLogin);
        btnSignUp = (ImageButton) findViewById(R.id.btnSignUp);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_initial);
        //esconder barra superior
        getSupportActionBar().hide();
        //exibir em tela cheia
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        bindComponents();
        makeRegisterButtonAvailable();

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //trocar de activity
                Intent intent = new
Intent(InitialActivity.this, LoginActivity.class);
                startActivity(intent);
                //fechar activity atual
                finish();
            }
        });

        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (SMSPermission == 0) {
                    Toast.makeText(InitialActivity.this,
getString(R.string.info_permission), Toast.LENGTH_LONG).show();
                } else {

```

```

        //trocar de activity
        Intent intent = new
Intent (InitialActivity.this, SignUpActivity.class);
        startActivity(intent);
        //fechar activity atual
        finish();
    }

    });
}

private void makeRegisterButtonAvailable() {
    if (!maySendSMS() || !mayReceiveSMS()) {
        return;
    }
    SMSPermission = 1;
}

private boolean maySendSMS() {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        return true;
    }
    if (checkSelfPermission(Manifest.permission.SEND_SMS) ==
PackageManager.PERMISSION_GRANTED) {
        return true;
    }
    if
(showRequestPermissionRationale(Manifest.permission.SEND_SMS)) {
        Snackbar.make(btnSignUp, R.string.permission_rationale,
Snackbar.LENGTH_INDEFINITE)
            .setAction(android.R.string.ok, new
View.OnClickListener() {
                @Override
                @TargetApi (Build.VERSION_CODES.M)
                public void onClick(View v) {
                    requestPermissions(new
String[] {Manifest.permission.SEND_SMS}, REQUEST_SEND_SMS);
                }
            });
    } else {
        requestPermissions(new String[] {Manifest.permission.SEND_SMS},
REQUEST_SEND_SMS);
    }
    return false;
}

private boolean mayReceiveSMS() {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        return true;
    }
    if (checkSelfPermission(Manifest.permission.RECEIVE_SMS) ==
PackageManager.PERMISSION_GRANTED) {
        return true;
    }
    if
(showRequestPermissionRationale(Manifest.permission.RECEIVE_SMS)) {
        Snackbar.make(btnSignUp, R.string.permission_rationale,
Snackbar.LENGTH_INDEFINITE)
            .setAction(android.R.string.ok, new
View.OnClickListener() {

```



```

        @Override
        @TargetApi(Build.VERSION_CODES.M)
        public void onClick(View v) {
            requestPermissions(new
String[] {Manifest.permission.RECEIVE_SMS}, REQUEST_RECEIVE_SMS);
        }
    });
} else {
    requestPermissions(new
String[] {Manifest.permission.RECEIVE_SMS}, REQUEST_RECEIVE_SMS);
}
return false;
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions,
                                     @NonNull int[] grantResults) {
    if (requestCode == REQUEST_SEND_SMS) {
        if (grantResults.length == 1 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            makeRegisterButtonAvailable();
        }
    }
    if (requestCode == REQUEST_RECEIVE_SMS) {
        if (grantResults.length == 1 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            makeRegisterButtonAvailable();
        }
    }
}
}
}

```

APÊNDICE E – Codificação de um Fragment

```

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import com. ....R;

public class DonateFragment extends Fragment {

    ImageButton btnInstagram;
    ImageButton btnFacebook;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View f = inflater.inflate(R.layout.fragment_donate, container,
false);
        btnInstagram = (ImageButton) f.findViewById(R.id.btnInstagram);
        btnFacebook = (ImageButton) f.findViewById(R.id.btnFacebook);

        btnInstagram.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Uri uri = Uri.parse("http://instagram.com/_u/meadorapp");
                Intent likeIng = new Intent(Intent.ACTION_VIEW, uri);

                likeIng.setPackage("com.instagram.android");

                try {
                    startActivity(likeIng);
                } catch (ActivityNotFoundException e) {
                    startActivity(new Intent(Intent.ACTION_VIEW,
                        Uri.parse("http://instagram.com/meadorapp")));
                }
            }
        });

        btnFacebook.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String id = "677842259333375";
                try {
                    Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("fb://page/" + id));
                    startActivity(intent);
                } catch (ActivityNotFoundException e) {
                    Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.facebook.com/" + id));
                    startActivity(intent);
                }
            }
        });
        return f;
    }
}

```