

PETAMIGO – SUA APLICAÇÃO DE ADOÇÃO DE PETS

Lucio Flavio Lima de Oliveira¹
Poliane Daniele Evangelista Rego²
Silvio Cesar Viegas³
Paulo Tadeu Campos Lopes⁴

RESUMO

Esse trabalho tem por objetivo demonstrar os conhecimentos adquiridos ao longo do curso de análise e desenvolvimento de sistemas, trata-se de um projeto de análise de um protótipo MVP (mínimo produto viável) de aplicação *Android* chamada *PetAmigo*, que propõe uma solução tecnológica, para publicação e divulgação de doações de animais de estimação a pessoas interessadas, dentre os objetivos, vale destacar o cadastro de publicações de possíveis doações de animais de estimação e o cadastro de usuários para utilizar a aplicação. A ideia surgiu para facilitar a divulgação e a comunicação de doações de *pets*, visto que as pessoas tem poucos canais para divulgar, geralmente publicando em suas próprias redes sociais, entrando em contato diretamente com amigos, conhecidos e parentes, publicando em grupos e comunidades espalhadas pelas redes, promovendo feiras e eventos de doações, entre outros, porém com o problema de ter que fazer isso repetidamente e em diversos canais distintos sem a unificação em uma única plataforma, gerando complicações para responder às questões de possíveis adotantes, editar suas publicações, demonstrar o status das doações, entrar em contato com as pessoas, excluir as publicações após atingir seus objetivos, esquecer exatamente em quais canais a pessoa publicou, ter problemas ao entrar em contato com interessados, não ter ferramentas adequadas para divulgar corretamente todos os detalhes relacionados ao *pet*, etc. A aplicação é desenvolvida utilizando a linguagem de programação *JAVA*, o que possibilita a utilização dos paradigmas da programação orientada a objeto na criação de softwares complexos e com uma manutenibilidade fácil e ampla, contando com a utilização de *frameworks* que auxiliam na sua elaboração, tais como, o *Android Studio* que é um ambiente de desenvolvimento integrado para desenvolver aplicativos para a plataforma *Android* e é acoplado ao *Java* tanto no desenvolvimento do código como na publicação de *GUI's* Interface Gráfica com o Usuário, assim, produzindo interfaces de um alto nível de usabilidade

¹ Aluno do curso de Análise e Desenvolvimento de Sistemas da Faculdade QI Brasil (FAQI). Gravataí, 2021. E-mail: luciooliveira.stu@gmail.com

² Tutora da disciplina de Projeto Aplicado do curso de Análise e Desenvolvimento de Sistemas da Faculdade QI Brasil (FAQI). Gravataí, 2021. E-mail: poliane.daniele@qi.edu.br

³ Professor Coordenador do curso de Análise e Desenvolvimento de Sistemas da Faculdade QI Brasil (FAQI). Gravataí, 2021. E-mail: silvio.viegas@qi.edu.br

⁴ Professor do Programa de Pós-graduação em Ensino de Ciências e Matemática da Universidade Luterana do Brasil - ULBRA. Canoas, 2021. E-mail: pclopes@qi.edu.br

e uma performance agradável no seu desenvolvimento. É também utilizado o SGBD *Oracle* que utiliza a linguagem *SQL* para armazenar e consultar as informações no banco de dados.

Palavras-chave: Animais de estimação. Oracle. SQL. Java. Android Studio. Android.

1 INTRODUÇÃO

O mundo dos animais de estimação apenas cresce com o passar dos anos onde mais produtos são lançados, mais inovações humanas também chegam para os amigos de quatro patas e cada dia mais os pets tem uma vida com mais conforto e mais tecnologia.

Esse projeto tem o intuito de entender justamente a raiz dessa vida de companheiros animais, logo na adoção, onde muitas pessoas e famílias recebem seus pets e que acabam virando datas extremamente importantes para as mesmas, pois esses animais acabam se tornando parte da família e da sua rotina.

É possível identificar que existem alguns problemas nessa etapa de adotar um pet, muitas vezes quando alguém quer adotar um animal de estimação não sabe direito onde procurar e como entrar em contato com os responsáveis pelas adoções.

O mesmo acontece com as pessoas que pretendem colocar um pet para adoção, seja por não ter a possibilidade de cuidar (o que muitas vezes acontece, pois, o “animal da casa” acaba tendo filhotes e a ninhada pode ser de quatro até doze filhotes dependendo do porte), seja por estar resgatando um animal de rua e colocando para doação, enfim, os motivos são diversos, porém é difícil encontrar uma ferramenta capaz de auxiliar neste “problema” tendo que recorrer às suas próprias redes sociais, a feiras de adoção e/ou ao contato direto com amigos e familiares, e que destas, muitas vezes, não se obtém o êxito pretendido.

1.1 Tema

A análise de um protótipo MVP de aplicativo de adoção de *pets*.

1.2 Delimitação do tema

A primeira versão deste aplicativo irá abranger o cadastro dos usuários do aplicativo (responsáveis pela adoção e adotantes) e o cadastro de adoções (*pets*, informações dos mesmos, características, contato, endereço, etc.) demonstrando quais serão as funcionalidades e as telas que o aplicativo pretende ter.

1.3 Problema

O problema abordado gira em torno da pergunta: Como criar uma aplicação com boa usabilidade para gerenciar adoções de animais de estimação?

1.4 Objetivo

1.4.1 Objetivo Geral

Desenvolver um aplicativo para gerenciar e otimizar o contato de pessoas que estão interessadas em adotar um *pet* e pessoas que estão interessadas em publicar uma adoção.

1.4.2 Objetivos Específicos

- a) Cadastrar os usuários do sistema (responsáveis e adotantes);
- b) Cadastrar endereços dos usuários da aplicação;
- c) Cadastrar e publicar adoções;
- d) Cadastrar informações dos adotandos (raça, cor, características, etc.);
- e) Desenvolver telas de fácil entendimento e navegação aos usuários;
- f) Demonstrar em qual linguagem de programação a aplicação pretende ser desenvolvida;
- g) Demonstrar qual o SGBD e a linguagem do banco de dados que a aplicação pretende ter;

- h) Demonstrar quais os frameworks que pretendem ser usados na prototipação e no desenvolvimento do software.

1.5 Justificativa

Existe hoje um problema na divulgação e na comunicação de pessoas interessadas em adotar um *pet* e pessoas interessadas em doar um *pet*, geralmente entrando em contato com amigos próximos ou divulgando em suas próprias redes sociais, o que nem sempre é eficaz e ágil como poderia ser se tivesse um aplicativo ou ferramenta destinados apenas a isso.

Além do problema na divulgação, também temos problemas relacionados a animais de rua que estão abandonados e a trazer animais de estimação a pessoas que necessitam ou que poderiam utilizar dos benefícios de ter a companhia dos mesmos, no qual esses assuntos serão mais profundamente abordados na fundamentação teórica.

2 FUNDAMENTAÇÃO TEÓRICA

O mercado trabalhado é na área de domesticação de animais, esses que estão presentes em muitas casas e até em alguns estabelecimentos e instituições privadas pelo Brasil e pelo mundo, segundo o site da revista EXAME:

Com base em dados oficiais do Instituto Brasileiro de Geografia e Estatística (IBGE), Heinze aponta que o Brasil já é o segundo país na quantidade de animais de estimação. Os números de 2018 indicam a presença de 139,3 milhões desses animais. São 54,2 milhões de cães, 39,8 milhões de aves, 23,9 milhões de gatos, 19,1 milhões de peixes e 2,3 milhões de outras espécies (répteis, anfíbios e pequenos mamíferos). O Brasil já tem mais cães e gatos do que crianças em seus lares, segundo o IBGE. (BRASIL...; EXAME, 2020).

A ideia do projeto é diminuir um problema que existe hoje, é o fato de existirem inúmeros animais de rua e que, muitas vezes, tem pouca ou nenhuma visibilidade, passando fome, passando sede, revirando o lixo, passando frio,

desenvolvendo doenças, procriando e fazendo esse problema aumentar ainda mais, passando por perigos como o fato de pessoas envenenarem esses animais simplesmente por acharem que isso é correto, dentre outros motivos.

Além de diminuir o problema de animais de rua, esse projeto pode auxiliar a pessoas que querem ter animais de estimação em casa, ou querem dar para filhos ou parentes, e é de conhecimento popular que animais de estimação podem trazer uma certa felicidade e alguns benefícios: eles podem ser boas companhias para pessoas que se sentem sozinhas, diminuem riscos de malefícios como depressão, estresse, alergias e algumas doenças, basicamente, animais fazem bem à saúde, e prova disso é que o hospital Albert Einstein da zona sul liberou visitas de animais de pequeno porte a pacientes acamados, segundo a revista. (SOARES; FOLHA, 2020).

2.1 Levantamento de Requisitos

O levantamento de requisitos é uma etapa da análise de sistemas que se refere a fazer um mapeamento do processo a fim de fazer o desenvolvimento do sistema com todo o entendimento de todas as questões relacionadas a ele, respondendo perguntas como: qual o objetivo desse sistema? Como ele deve ser? Qual o usuário médio que irá utilizá-lo? O que deve ter no escopo do projeto? Etc., lembrando que tudo isso deve ser feito junto ao usuário que vai utilizá-lo.

Além das perguntas que devem ser respondidas é seguro afirmar que o levantamento de requisitos serve para levantar todos os requisitos que o sistema deve ter, não necessariamente fazendo isso de forma extremamente detalhada, pode ser feito de forma mais superficial, pois mais à frente no processo e em outros levantamentos entraremos no detalhe sobre cada requisito do sistema.

2.2 Aplicação Móvel

Uma aplicação móvel é um *software* que foi desenvolvido com o objetivo de ser instalado em smartphones e outros dispositivos móveis, como *iPads* e *tablets*.

Foi escolhido fazer uma aplicação móvel para esse sistema pois estima-se

que é a mais acessível e popular do mundo, segundo o blog REACH: até 2020, 80% da população deve ter um telefone móvel! Em outubro de 2016, o acesso à internet por meio de dispositivos mobile somaram 51% do total mundial — ou seja, os desktops foram superados pela tecnologia mobile. (REACHBLOG, 2019).

2.2.1 Android

Quando se fala sobre programação mobile ou aplicativos de celulares, um dos assuntos mais relacionados é o *Android*, que nada mais é que um sistema operacional desenvolvido pelo *Google* com o objetivo de gerenciar as ferramentas e as tarefas da maior parte dos dispositivos móveis que estão disponíveis no mercado hoje em dia, ele foi criado em 2003, na Califórnia pelos fundadores Andy Rubin, Rich Miner, Nick Sears e Chris White.

Hoje o *Android* é o sistema operacional mais utilizado do mundo, segundo o que se fala no site Tectudo: São mais de 2,5 bilhões de usuários, segundo dados revelados no evento Google I/O deste ano. (SANTOS; TECHTUDO, 2019).

2.3 Tecnologias

2.3.1 Java

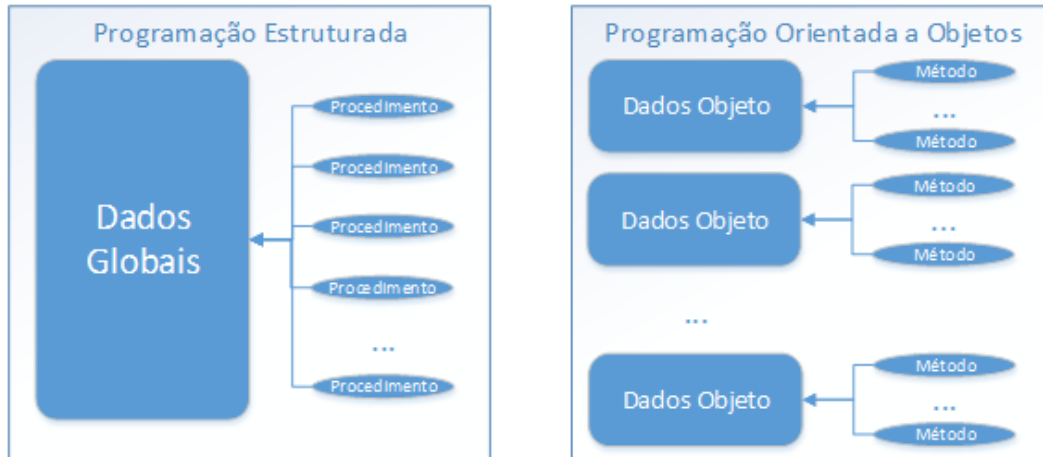
Apesar do projeto ser um protótipo de sistemas, ele é pretendido a ser desenvolvido em Java, que nada mais é que uma linguagem de programação orientada a objetos que foi desenvolvida em 1995 baseado em C++ por uma equipe de desenvolvedores gerenciada por James Gosling, na empresa Sun Microsystems.

A linguagem de programação escolhida para essa aplicação foi o Java por conta de algumas características específicas:

Orientado a objetos: O conceito de Programação Orientada a Objetos (POO) é um conceito que se popularizou por ser um padrão que tem evoluído muito por conta de assuntos voltados ao reaproveitamento de códigos e a segurança do

sistema, ele vem substituindo o conceito de programação estruturada e se tornando cada vez mais popular:

Figura 1 - Programação Estruturada x Programação Orientada a Objetos.



Fonte: DEVMEDIA, 2014.

Multiplataforma: Também chamado de “independência de plataforma”, esta é uma característica importante pois ele atende a quaisquer plataformas, segundo DevMedia: Hoje a maioria das linguagens sofrem na transferência de plataforma quando o sistema desenvolvido tem que migrar para outra plataforma (DEVMEDIA, 2012).

Popularidade: Quando se fala na popularidade do Java em relação a aplicação que pretende ser desenvolvida, pode-se pensar em duas vertentes: a popularidade quanto a linguagem de programação no geral e a popularidade em dispositivos Android. Quanto a popularidade em linguagem de programação, esse gráfico da Tiobe mostra as 20 principais linguagens de programação e a escala de popularidade em dezembro de 2017:

Figura 2 - Gráfico Tiobe de linguagens mais populares de dez de 2017.



Fonte: Danki Code, 2018.

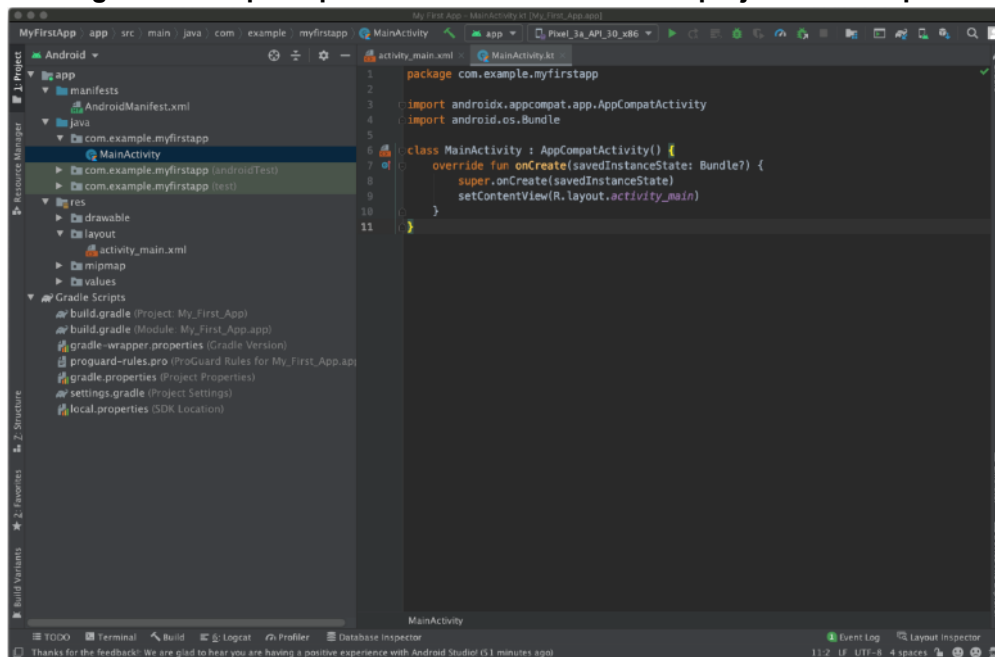
Sobre popularidade em sistemas Android, o Java é tido popularmente como a linguagem oficial para o desenvolvimento em sistemas Android.

Mais um motivo da escolha da linguagem é o fato de ser também suportado pelo Android Studio, a plataforma de desenvolvimento que pretende ser usada para essa aplicação.

2.3.2 Android Studio

Como dito anteriormente o Android Studio é uma IDE (Integrated Development Environment), ou seja, um ambiente de desenvolvimento integrado, essa IDE foi desenvolvida pela Google e disponibilizada gratuitamente para desenvolvedores Android, ela suporta linguagens como Java, Kotlin, C++, entre outras.

Figura 3- Tela principal do Android Studio em um projeto de exemplo.



Fonte: Developers, 2021.

2.3.3 Oracle

O Oracle é um SGBD (Sistema de Gerenciamento de Banco de Dados) que foi lançado em 1978 e foi escrito utilizando a linguagem C e com o passar das versões foi recebendo atualizações escritas em C, C++, Assembly e Java, ele é um SGBD com diversos recursos, segundo o CCM: garante a definição e a manipulação dos dados, a coerência e a confidencialidade deles, a integridade, o backup, a restauração e o acesso a eles. (SPADARI; CCM, 2020).

É pretendido que o SGDB seja o Oracle pois é necessário um banco de dados para armazenar todas as informações da aplicação, seja no cadastro dos usuários do sistema, no cadastro de adoções de animais de estimação e em outras ferramentas disponibilizadas na plataforma e que necessitam de gravar, alterar ou excluir informações.

2.3.4 SQL

O SQL será a linguagem de banco de dados utilizada no SGBD, o SQL é uma

linguagem padrão para bancos de dados relacionais, é relativamente simples e serve para fazer consultas e modificações no banco de dados, ela surgiu inicialmente para padronizar o jeito que os profissionais na área de TI e de computação lidam com o SGBD e com os bancos de dados.

2.4 Prototipação

A prototipação do *software* é uma etapa importante no desenvolvimento de qualquer software, e serve especificamente para auxiliar o desenvolvedor a entender como o software deve funcionar e quais aparências ele deve ter, além desse auxílio ele também serve como visualização para ajustes que existirão em torno do projeto e para o desenvolvedor ter uma ideia de como o software deve ficar em sua versão final.

Prototipação de software é algo que será extremamente abordado e praticado neste projeto, visto que se trata de um protótipo de aplicação e não, ainda, no desenvolvimento do mesmo, o que será trabalhado no quesito prototipação para esse software serão algumas telas que servirão de protótipo para um futuro desenvolvimento.

2.4.1 MarvelApp

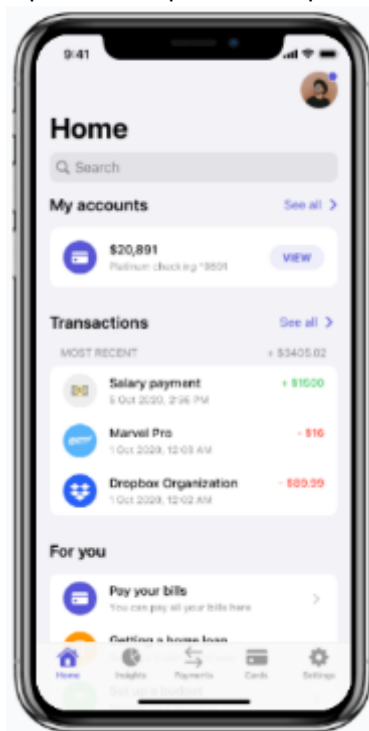
O MarvelApp é o *framework* que será utilizado na prototipação das telas do sistema, ele é ótimo para quem deseja prototipar uma aplicação Android pois ele conta com algumas ferramentas que auxiliam neste processo, com ele é possível usar como base uma tela de celular e ele traz algumas telas padrões como *login*, *home*, aba de busca e que mesmo não sendo do idioma português podem ser facilmente personalizáveis.

MarvelApp também conta com bibliotecas de imagens e ícones, e com a funcionalidade de utilizar imagens da internet ou do próprio dispositivo para inserir na sua prototipação, eles disponibilizam de um design de fácil entendimento e que entregam muitas ferramentas para auxiliar.

Segundo o próprio site:

Wireframe, design e protótipo rapidamente com nossas ferramentas intuitivas de design e prototipagem. Gere instantaneamente especificações de design e conecte integrações que potencializam seu fluxo de trabalho. De baixa a alta fidelidade, a Marvel apoia você em cada etapa do caminho. (MARVELAPP, 2018).

Figura 4 - Protótipo de exemplo de um aplicativo de finanças.



Fonte: Marvel, [2021].

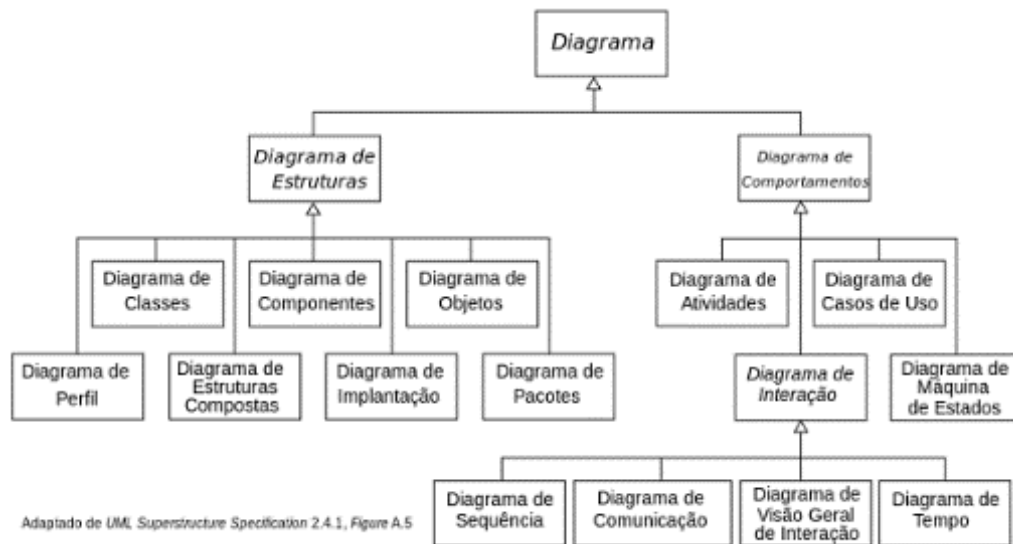
2.5 Estruturas de diagramas UML

O UML (*Unified Modeling Language*) vem da área de engenharia de software e ele é muito importante para projetos da área de tecnologia, sobretudo quando se trata de um novo sistema, ele é um padrão estrutural que consiste em criar alguns diagramas que servem para diferentes funções, seja para demonstrar funções específicas de um recurso do software que está sendo desenvolvido, seja para mostrar com mais detalhes por exemplo, a sequência de atividade que o usuário terá que fazer e como o sistema responderá a isso.

É pretendido demonstrar exemplos de alguns dos principais diagramas e qual

a função do mesmo, e, posteriormente, serão criados diagramas específicos para o software que está sendo trabalhado neste projeto.

Figura 5 - Estrutura de diagramas UML

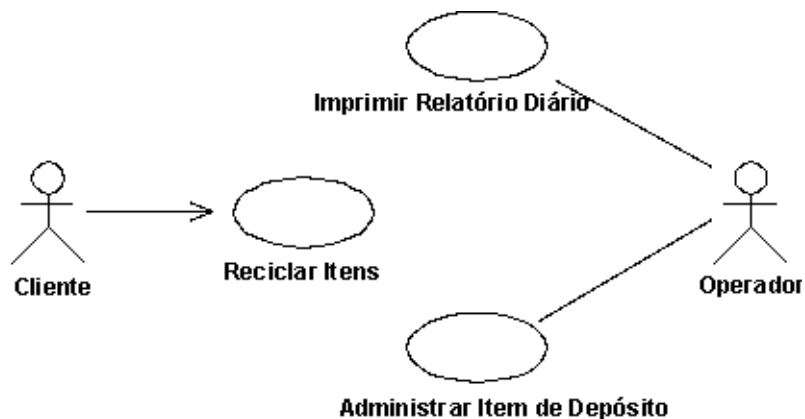


Fonte: VENTURA, 2021.

2.5.1 Exemplo de Diagrama de Casos de Uso

O diagrama de casos de uso tem por objetivo demonstrar os usuários do sistema e as funções que os mesmos devem ter acesso, ele pode servir tanto para abranger o sistema como um todo, quanto para demonstrar as funções de apenas uma das partes do sistema:

Figura 6 - Exemplo de diagrama de casos de uso de um sistema de máquina de reciclagem.



Fonte: DIRETRIZ, 2006.

2.5.2 Exemplo de Descrição expandida de Caso de Uso

Também chamado de caso de uso expandido, ele serve como um detalhamento do diagrama de caso de uso, nele são demonstrados cada uma das funções do software com o detalhamento e com as características de cada uma delas sejam elas: Descrição, atores, Pré-condições, Pós-condições, Requisitos especiais, fluxo principal, fluxos alternativos e todos os detalhes que forem necessários:

Figura 7 - Exemplo de descrição expandida de caso de uso “Atualizar produtos para Venda” de um sistema de estabelecimento comercial.

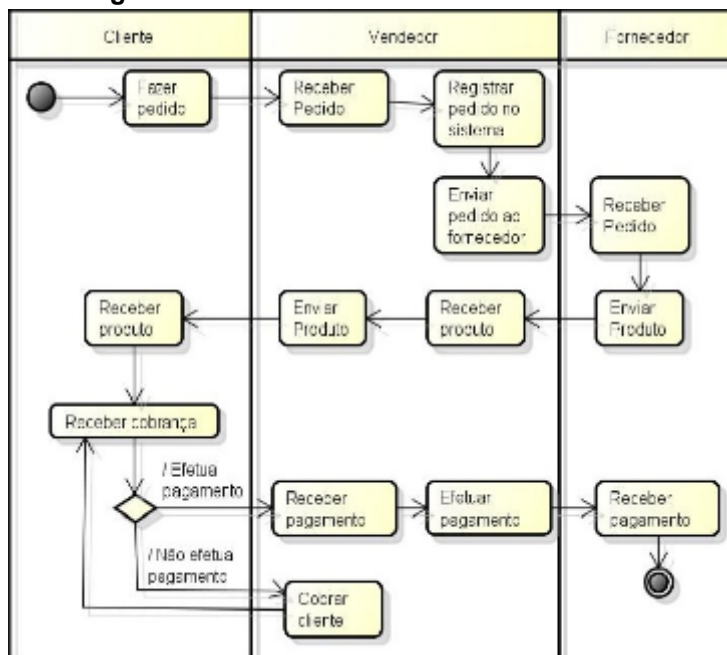
Caso de Uso	Atualizar Produtos para Venda.
Ator	Gerente e Funcionário.
Stakeholder	Nenhum.
Fluxo Principal de Sucesso	<ol style="list-style-type: none"> 1. O Sistema apresenta o formulário correspondente ao produto selecionado. 2. O gerente/funcionário altera os dados do campo selecionado. 3. O Sistema valida os campos alterados. 4. O gerente/funcionário confirma a atualização. 5. O Sistema atualiza os dados.
Fluxo Alternativo	<p>2a Campo em Branco. O sistema alerta ao gerente/funcionário que existem campos em branco em um campo obrigatório.</p> <p>2b Campos com caracteres inválidos. O sistema alerta o gerente/funcionário que existem campos com caracteres inválidos.</p>

Fonte: legacyofrapha, (2009).

2.5.3 Exemplo Diagrama de Atividades

O diagrama de atividades consiste em demonstrar o fluxo de uma ou mais funções do sistema, mostrando qual é o ponto inicial e quais serão os atores e as funcionalidades do sistema que o usuário irá passar até finalizar o objetivo.

Figura 8 - Exemplo de Diagrama de Atividades de um sistema de venda e entrega de produto.

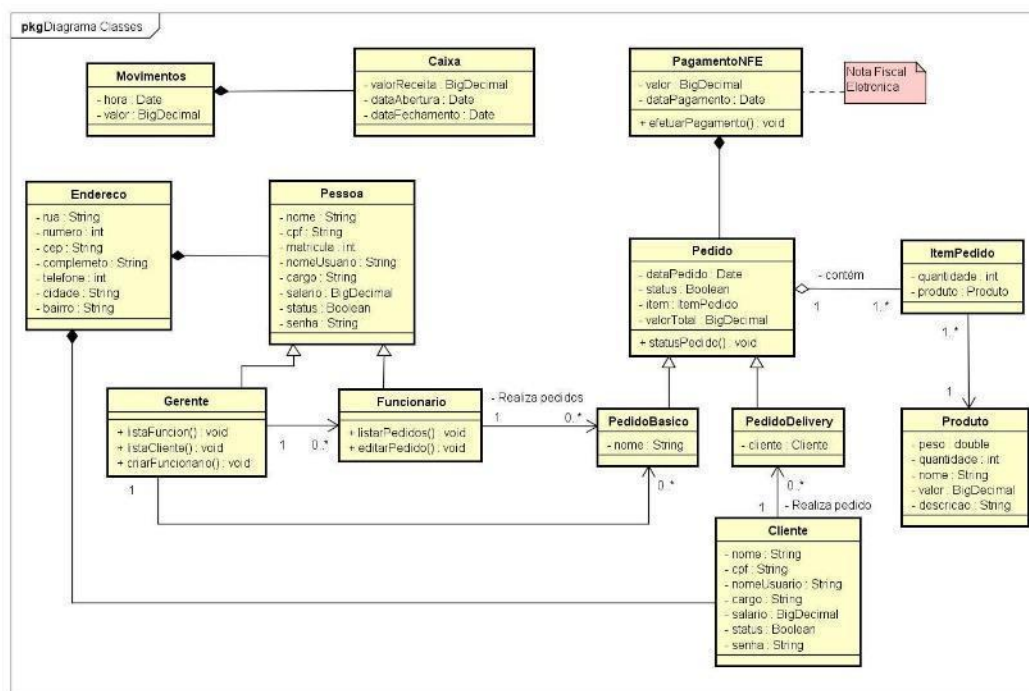


Fonte: OLIVEIRA; RODELLO, 2021.

2.5.4 Exemplo Diagrama de Classes

O diagrama de classes serve para agrupar dados e processos em respectivos grupos, e também demonstrar como esses grupos estão interligados e quais são incluídos dentro do outro.

Figura 9:- Exemplo de diagrama de classe entre gerente e funcionário dentro de um sistema.



Fonte: COMO representar..., [2017].

3 METODOLOGIA

A metodologia usada na criação do projeto consiste em três passos que serão quebrados em outras sub-etapas que vão desde a definição, passa pelo conhecimento e finaliza com a modelagem do mesmo.

Se trata de uma metodologia orientada a objetos e a seguir cada um dos passos serão explicados:

1 – Definição	1.1 Área de atuação; 1.2 Linguagem de programação; 1.3 Banco de dados;
2 – Conhecimento	2.1 Ferramentas utilizadas; 2.2 Requisitos;
3 – Modelagem	3.1 Diagramas; 3.2 Prototipação.

1. **Definição:** Nessa etapa é onde entra a análise do sistema, que identifica a área de atuação, o problema envolvido, a linguagem de programação escolhida e as definições da área de banco de dados.
 - 1.1 Área de Atuação: A área de atuação escolhida será o de animais domésticos, é identificado o problema de dificuldades na publicação de animais de estimação e identificado quais serão os pontos fortes e fracos do projeto.
 - 1.2 Linguagem de Programação: A linguagem de programação escolhida será o Java, por se tratar de uma das linguagens mais populares do mercado, principalmente em dispositivos móveis, e também pois é a linguagem de programação que trabalhamos na grade da FAQI, também foi identificado o dispositivo que será o Android.
 - 1.3 Banco de Dados: O SGBD escolhido foi o Oracle e a linguagem de banco de dados será o SQL.
2. **Conhecimento:** Nessa etapa serão definidas as ferramentas utilizadas na prototipação e na futura programação do projeto, e será definidos os requisitos do software.
 - 2.1 Ferramentas Utilizadas: As ferramentas utilizadas se dividem em dois tipos, o que será utilizado na prototipação das telas que será o Marvel App, uma aplicação Web open source, e o outro tipo é na futura programação do sistema, que será utilizada a API do google, o Android Studio, focado em desenvolvimento android.
 - 2.2 Requisitos: Aqui são definidos os requisitos do sistema, tanto os requisitos funcionais e os requisitos não funcionais, onde é detalhado o objetivo, a descrição do objetivo e os outros detalhes em relação aos requisitos do sistema.
3. **Modelagem:** Na modelagem são definidos os diagramas que servirão como base para o desenvolvimento do sistema e a prototipação que trará uma visão de interface para o sistema.
 - 3.1 Diagramas: Aqui são montados e apresentados os diagramas referentes ao sistema, diagrama de caso de uso, caso de uso

expandidos, diagrama entidade relacionamento conceitual e lógico e que servirão de base para um futuro desenvolvimento.

3.2 Prototipação: Aqui são apresentadas as telas prototipadas para uma visão do front-end e da interface do sistema, demonstrará um fluxo com as funcionalidades e que servirá como base para o desenvolvimento do mesmo.

4 DESENVOLVIMENTO

4.1 Especificação de Requisitos

4.1.1 Requisitos Funcionais

RF01 – Cadastro de usuário	
Pré-Requisitos	-
Pós-Requisitos	- RF02, RF03, RF04, RF05
Descrição	<p>O sistema deve possuir um cadastro de usuários de modo que os seguintes dados devem ser armazenados:</p> <ul style="list-style-type: none"> ● Nome completo ● Data de nascimento ● Celular ● E-mail ● Senha (criptografada) <p>*Após o cadastro, o usuário deve ser redirecionado para a página principal da aplicação.</p>

RF02 – Autenticação no sistema	
Pré-Requisitos	- RF01
Pós-Requisitos	- RF03, RF04, RF05, RF06
Descrição	<p>Deve ser previsto uma tela de <i>login</i> para autenticar o usuário, que deve informar os seguintes campos:</p> <ul style="list-style-type: none"> ● E-mail ou usuário cadastrado ● Senha cadastrada <p>*Após o <i>login</i>, o usuário deve ser redirecionado para a página principal da aplicação.</p>

RF03 – Recuperação de senha	
Pré-Requisitos	- RF01
Pós-Requisitos	- RF04, RF05, RF06
Descrição	<p>O sistema deve contemplar uma tela de recuperação de senha, o usuário deve informar:</p> <ul style="list-style-type: none"> • E-mail cadastrado • Repetir e-mail cadastrado <p>*Após o usuário informar o e-mail cadastrado, o sistema deve enviar uma nova senha para o e-mail informado.</p>

RF04 – Cadastro de adoção	
Pré-Requisitos	- RF01, RF02
Pós-Requisitos	- RF05, RF06
Descrição	<p>O sistema deve possuir um cadastro de adoções de modo que os seguintes dados de animais de estimação devem ser armazenados:</p> <ul style="list-style-type: none"> • Nome • Idade • Peso • Porte • Raça • Cor • Sexo • Restrições • Breve descrição sobre o animal (sobre mim) • Fotos do <i>pet</i> <p>E as seguintes informações do responsável pela adoção devem ser armazenadas:</p> <ul style="list-style-type: none"> • Endereço da adoção • Nome do responsável • E-mail do responsável • Celular do responsável <p>*Ao final do cadastro da adoção será gerada uma nova adoção no sistema.</p>

RF05 – Edição de adoção	
Pré-Requisitos	- RF01, RF02, RF04
Pós-Requisitos	- RF06
Descrição	<p>O sistema deve possuir a funcionalidade de editar adoções que foram cadastradas pelo usuário.</p>

RF06 – Envio de mensagem	
Pré-Requisitos	- RF01, RF02
Pós-Requisitos	-
Descrição	<p>O sistema deve prever a funcionalidade de envio de mensagens, de modo que:</p> <ul style="list-style-type: none"> • O usuário pode enviar mensagens direto de uma adoção publicada que o interessou. • O usuário responsável pela adoção pode responder mensagens que o foram enviadas. • Ambos usuários podem enviar arquivos no formato JPEG, PNG, PDF, MP4, AVI, entre outros. • Ambos usuários podem enviar emojis de uma biblioteca de emojis conectada ao sistema.

4.1.2 Requisitos Não Funcionais

RNF01 – Facilidade de uso	
Pré-Requisitos	-
Pós-Requisitos	-
Descrição	O sistema deve ser de fácil uso e de fácil aprendizado, trazendo padrões para facilitar o uso para qualquer usuário.

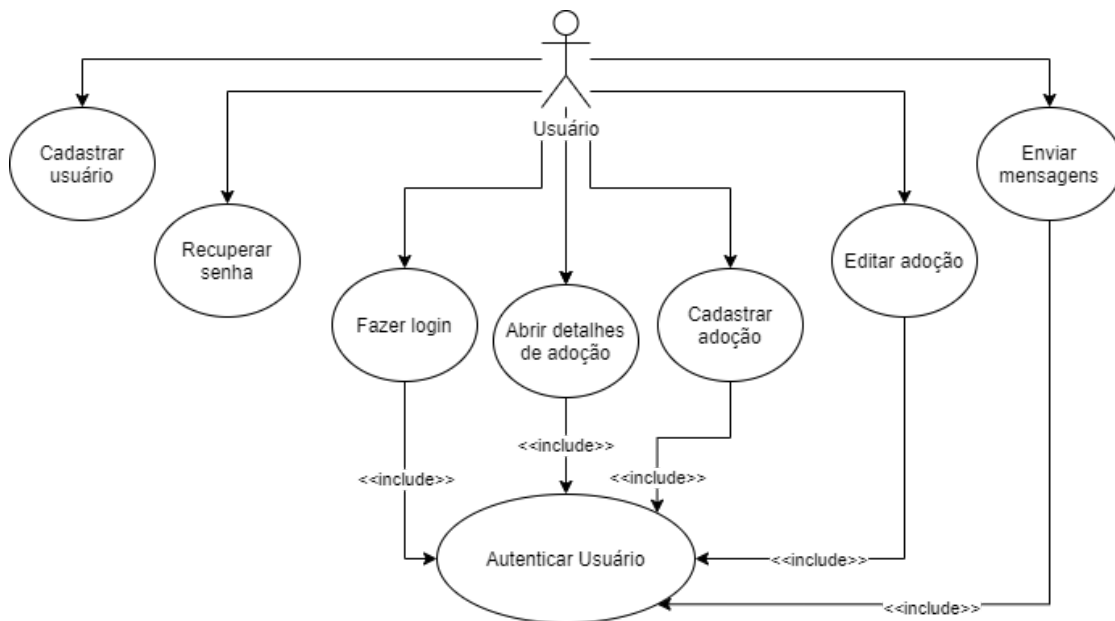
RNF02 – Desempenho e eficiência	
Pré-Requisitos	-
Pós-Requisitos	-
Descrição	O sistema deve ter um tempo de resposta para qualquer ação que envolva o banco de dados não maior do que 5 segundos e o tempo de cadastro de um novo usuário não deve ser superior a 5 minutos.

RNF03 – Segurança do sistema	
Pré-Requisitos	-
Pós-Requisitos	-
Descrição	O sistema deve ser seguro o suficiente de forma a não permitir acesso sem o usuário de uma senha pessoal e intransferível e a base de dados do sistema não deve estar disponível a não ser pelo administrador do sistema.

4.2 Diagramas

4.2.1 Diagrama de Caso de Uso

Figura 10 - Diagrama de caso de uso PetAmigo



Fonte: Elaborado pelo autor (2021)

Na figura 10 representa graficamente os casos de uso, ou seja, as funcionalidades que o usuário fará uso na aplicação, lembrando que o usuário na aplicação é apenas de um tipo, não diferenciando as funcionalidades entre eles. Os balões que tem uma seta com “<<include>>” para o maior e último balão do diagrama representa que essas funcionalidades farão uso da funcionalidade “Autenticar Usuário”.

4.2.2 Diagrama de Casos de Uso Expandidos

UC01 – Cadastrar usuário	
Objetivo	Cadastrar o próprio usuário na aplicação.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	
Fluxo Principal	1.1 Usuário abre a aplicação 1.2 Usuário clica no botão “faça seu cadastro” 1.3 Informa todos os dados necessários (nome, data de nascimento, celular e e-mail) 1.4 Informa a senha de cadastro e repete a senha 1.5 O sistema cadastra o usuário com sucesso
Fluxo Alternativo	1.6 Caso um dos dados de cadastro seja inválido, o sistema retorna o erro: “um dos dados de cadastro é inválido” 1.7 Caso a senha repetida não seja igual a primeira senha informada, o sistema retorna o erro: “as senhas digitadas não combinam”

UC02 – Recuperar senha	
Objetivo	Recuperar a senha de cadastro para <i>logar</i> no sistema.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar cadastrado no sistema
Fluxo Principal	2.1 Usuário abre a aplicação 2.2 Usuário clica no botão “esqueceu sua senha? “Clique aqui” ” 2.3 Informa e-mail cadastrado e repete o e-mail informado 2.4 O sistema valida o e-mail informado 2.5 O sistema envia uma nova senha para o e-mail cadastrado
Fluxo Alternativo	2.6 Caso o e-mail informado seja em formato inválido, o sistema retorna o erro: “e-mail inválido” 2.7 Caso o e-mail repetido não seja igual ao primeiro e-mail informado, o sistema retorna o erro: “os e-mails digitados não combinam” 2.8 Caso o e-mail informado não esteja cadastrado, o sistema retorna o erro “e-mail não cadastrado”

UC03 – Fazer login	
Objetivo	Fazer a autenticação para entrar no sistema.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar cadastrado no sistema
Fluxo Principal	3.1 Usuário abre a aplicação 3.2 Usuário informa e-mail e senha 3.3 O sistema valida os dados informados 3.4 O usuário é logado com sucesso
Fluxo Alternativo	3.5 Caso o e-mail ou a senha informada sejam incorretos, o sistema retorna o erro: “dados incorretos” 3.6 Caso o e-mail informado esteja em um formato inválido, o sistema retorna o erro: “e-mail inválido” 3.7 Caso o e-mail informado não esteja cadastrado, o sistema retorna o erro “e-mail não cadastrado”

UC04 – Abrir detalhes de adoção	
Objetivo	Abrir os detalhes para ver mais informações das adoções publicadas.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar autenticado no sistema
Fluxo Principal	4.1 Usuário inicia no menu principal 4.2 Escolhe uma adoção publicada 4.3 Clica no botão “ver detalhes” 4.4 O sistema redireciona o usuário para os detalhes daquela publicação de adoção
Fluxo Alternativo	

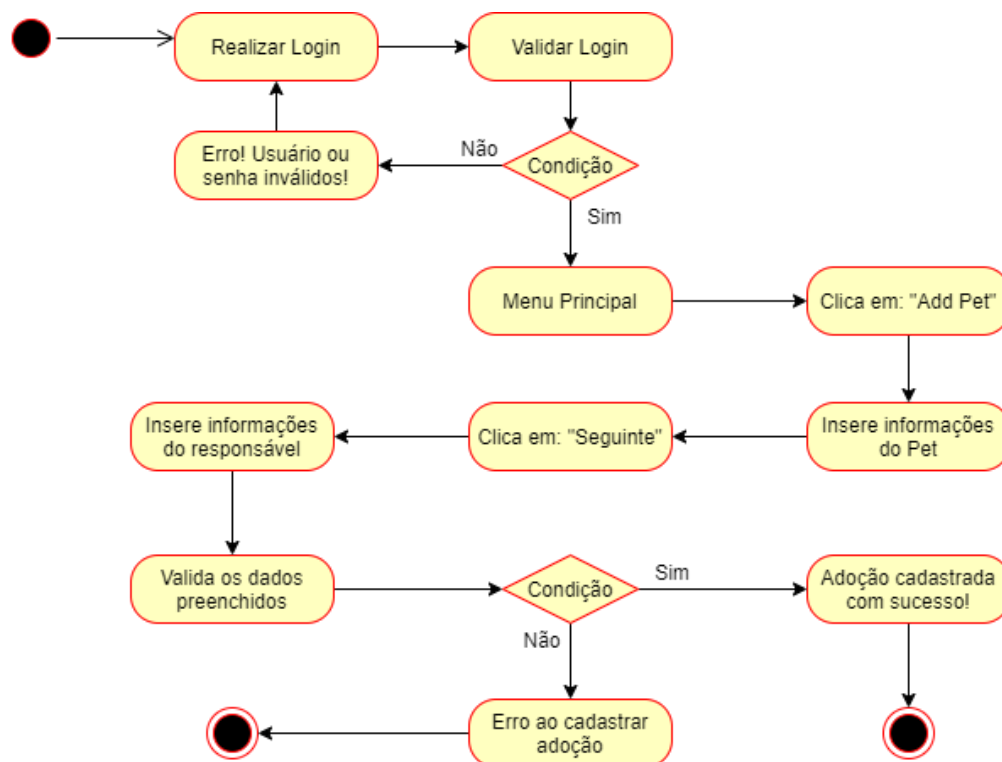
UC05 – Cadastrar adoção	
Objetivo	Cadastrar e publicar uma nova adoção na aplicação.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar autenticado no sistema
Fluxo Principal	5.1 Usuário inicia no menu principal 5.2 Clica no botão “add pet” 5.3 Informa os dados do pet e adiciona fotos do pet 5.4 Clica no botão “seguite” 5.5 Informa os dados do responsável 5.6 Clica em “cadastrar adoção” 5.7 O sistema cadastra uma nova adoção
Fluxo Alternativo	5.8 Caso os dados digitados sejam inválidos, o sistema retorna o erro: “dados inválidos” 5.9 Caso o usuário não adicione fotos do pet e clique em “seguite”, o sistema retorna o erro: “favor adicionar pelo menos uma foto do pet”

UC06 – Editar adoção	
Objetivo	Editar informações das próprias adoções publicadas.
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar autenticado no sistema, Usuário deve ter pelo menos uma adoção cadastrada
Fluxo Principal	6.1 Usuário inicia no menu principal 6.2 Usuário clica na sua adoção publicada 6.3 Sistema redireciona para os detalhes da adoção 6.4 Usuário clica no botão de edição de adoção 6.5 Usuário edita os dados da adoção 6.6 Usuário clica em “salvar” 6.7 O sistema atualiza a adoção conforme edição feita pelo usuário
Fluxo Alternativo	6.8 Caso os dados digitados sejam inválidos, o sistema retorna o erro: “dados inválidos” 6.9 Caso o usuário desista de atualizar os dados basta clicar em “cancelar” no menu de edição de adoção

UC07 – Enviar mensagens	
Objetivo	Enviar mensagens para publicações de adoções interessadas ou responder mensagens de usuários interessados (adotantes).
Atores	Usuário
Prioridade	Alta Prioridade
Pré-condições	Usuário deve estar autenticado no sistema
Fluxo Principal	7.1 Usuário inicia no menu principal 7.2 Usuário clica no botão “ver detalhes” na adoção interessada 7.3 Sistema redireciona para os detalhes da adoção 7.4 Usuário clica no botão de envio de mensagens 7.5 Usuário escreve e envia sua mensagem 7.6 O sistema envia a mensagem ao usuário responsável pela adoção
Fluxo Alternativo	7.7 Caso o usuário queira ir nas suas conversas, basta clicar no ícone de mensagem na tela principal 7.8 Caso o usuário receba uma nova mensagem, basta clicar no ícone de mensagem na tela principal para responde-la

4.2.3 Diagrama de Atividades

Figura 11- Diagrama de Atividades – Cadastrar adoção



Fonte: Elaborado pelo autor (2021)

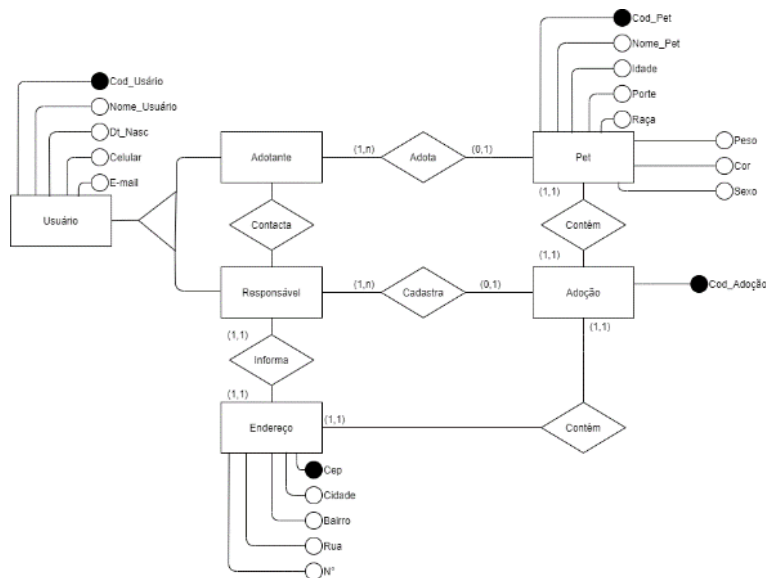
Na figura 11 mostra a representação gráfica de uma das funcionalidades do sistema, o cadastro de adoção, que tem o objetivo de cadastrar uma nova adoção e publicá-la na aplicação.

O círculo preenchido demonstra o estado inicial que o usuário começa e após ele temos os retângulos que representam diferentes passos que o usuário terá que percorrer no fluxo de sistema para realizar a funcionalidade proposta, os losangos representam as condições que fazem o sistema realizar diferentes ações conforme a condição (ex. se sim o usuário é autenticado, se não apresenta um erro), e por fim os círculos finais com borda vermelha não preenchida representa o estado final do usuário, que levam a diferentes caminhos dependendo se a condição proposta foi satisfeita ou insatisfeita.

4.2.4 Diagrama ER

Diagrama Entidade Relacionamento Conceitual

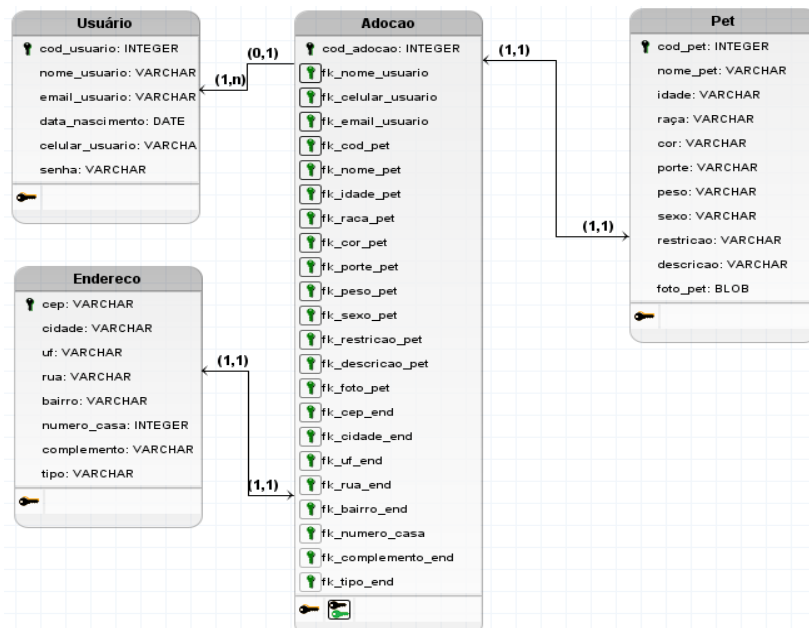
Figura 12 - Diagrama Entidade Relacionamento Conceitual PetAmigo



Fonte: Elaborado pelo autor (2021)

Diagrama Entidade Relacionamento Lógico:

Figura 13- Diagrama Entidade Relacionamento Lógico PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3 Prototipação de Software

4.3.1 Tela de Carregamento

Figura 14- Tela de carregamento da aplicação PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3.2 Tela de Login

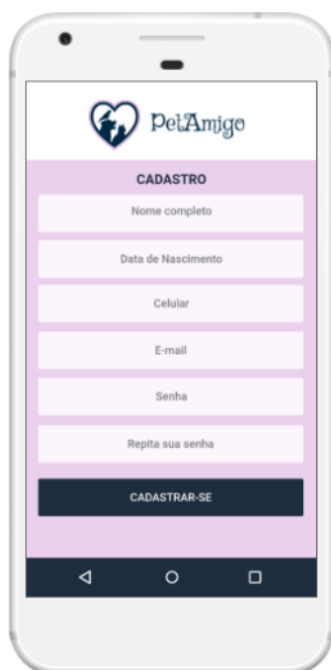
Figura 15- Tela de Login da aplicação PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3.3 Tela de Cadastro

Figura 16 - Tela de Cadastro da aplicação PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3.4 Tela de Recuperação de Senha

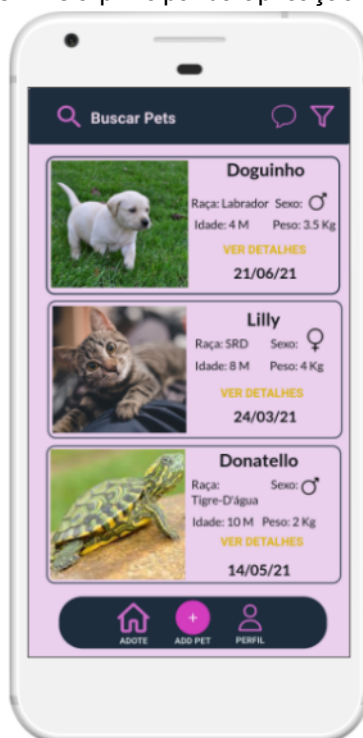
Figura 17 - Tela de Recuperação de Senha da aplicação PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3.5 Tela Principal

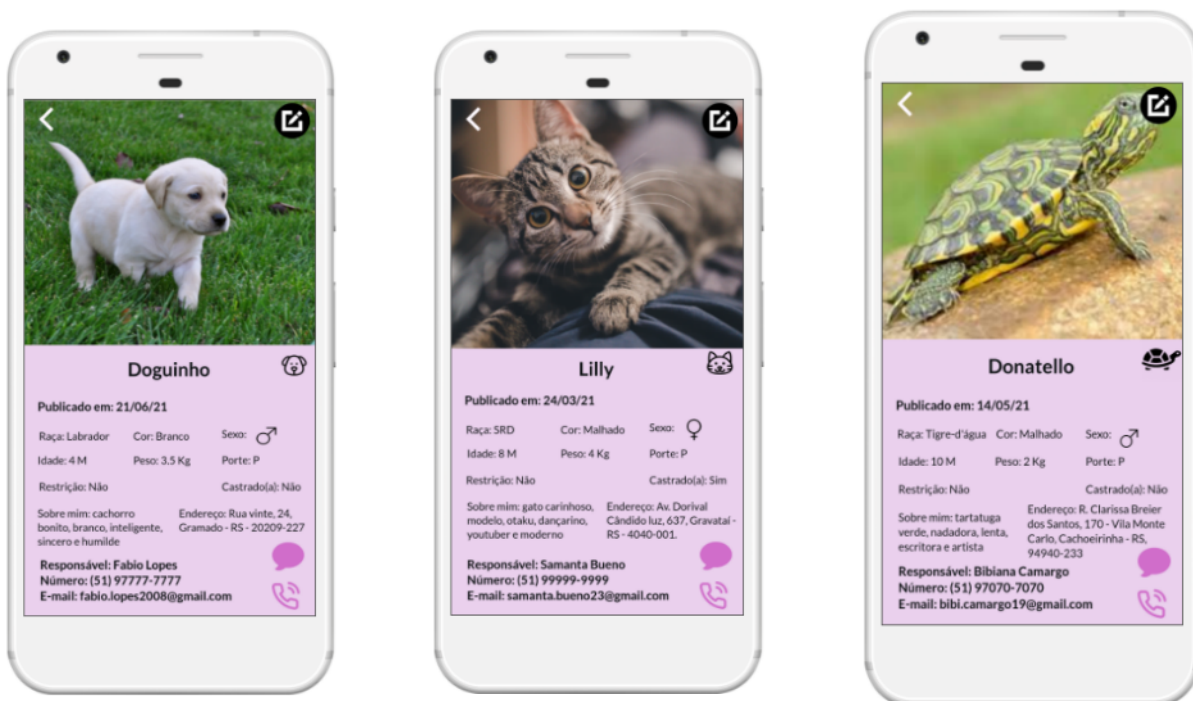
Figura 18 - Tela principal da aplicação PetAmigo



Fonte: Elaborado pelo autor (2021)

4.3.6 Tela de Detalhes de Pet

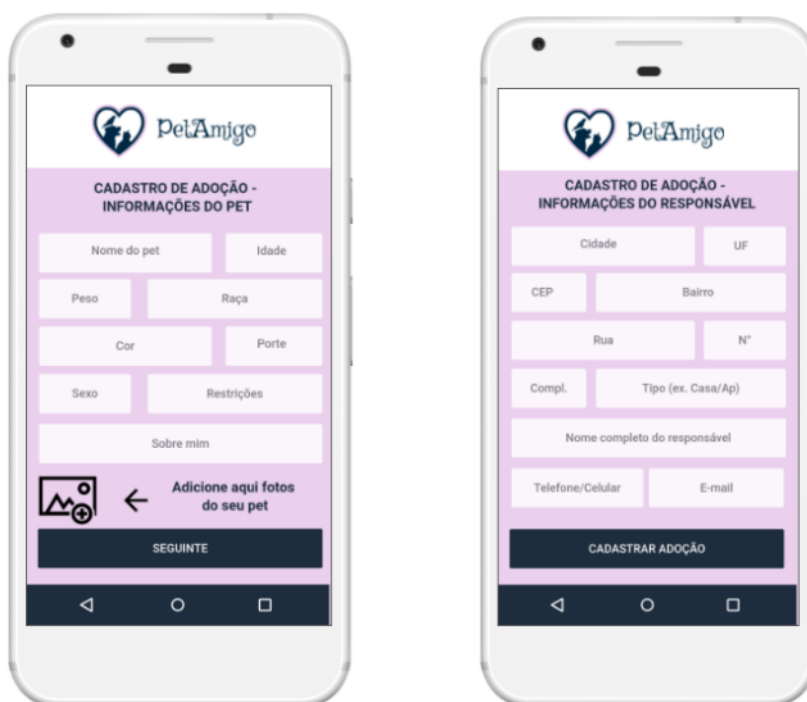
Figura 19- Tela de Detalhes de pet com os três exemplos desenvolvidos



Fonte: Elaborado pelo autor (2021)

4.3.7 Telas de Cadastro de Adoção

Figura 20 - Telas de Cadastro de adoção mostrando os dois passos dessa funcionalidade



Fonte: Elaborado pelo autor (2021)

5 CONSIDERAÇÕES FINAIS

O projeto tem como objetivo fazer a análise de um protótipo de um MVP para uma aplicação Android que se destina a cadastrar e publicar adoções de animais para pessoas interessadas, com isso os objetivos foram alcançados com sucesso e o problema informado anteriormente foi resolvido utilizando a aplicação demonstrada.

No decorrer do projeto foram demonstradas as tecnologias que o software pretende ter, foi informada a fundamentação teórica que justifica os objetivos deste projeto, foram demonstrados os diagramas que serão de auxílio para entender as funcionalidades do sistema e também as telas de prototipação para serem usadas como referência para demonstrar tudo o que sistema deve ter.

Além dos resultados, também foram levantadas algumas futuras atualizações que o sistema pretende ter com mais funcionalidades, focadas em uma melhor usabilidade e melhoria de vida do usuário:

- Desenvolvimento e programação da aplicação (no momento é apenas a análise de um protótipo);
- Outras possibilidades de cadastro e login (*Facebook, Google, Twitter, Instagram, etc.*);
- Filtros para uma melhor busca de adoções publicadas;
- Se conectar a localização para mostrar o quão perto estão as doações
- Um mapa onde mostra o *Google Maps* e onde estão as localizações (se conectando a uma API com base no GPS e nos endereços informados);
- Um ícone para o *Whatsapp* do responsável pela adoção;
- Ícones para as redes sociais dos responsáveis;
- Possibilidade de colocar uma foto no seu perfil;
- Lista de adoções já realizadas na aplicação;
- Área de "meus pets" que irão mostrar os pets que o usuário já adotou pela ferramenta e os que ele colocou para adoção;

- Um menu de informações que irão conter todas as raças de cachorros conhecidas, um resumo e as características das mesmas;
- Tela de edição do perfil;
- Adicionar vacinas dos pets no cadastro de doações;
- Tela de orientações e dicas para os animais de estimação (vacinas, remédios, alimentos, alergias, dicas, o que fazer, dúvidas gerais, etc.);
- Ter as raças, cores, endereços e todas as opções possíveis em um banco de dados para que o usuário precise apenas "escolher" as opções ao invés de informar manualmente;
- Tela de avaliação e sugestões dos usuários (para avaliarem e fazerem sugestões sobre a ferramenta);
- Disponibilizar para IOS e para outros sistemas operacionais;
- Disponibilizar no *Play store*, *Samsung apps*, *Apple store*, *Windows store*, etc.;
- Funcionalidade de compartilhar a ferramenta no *Whatsapp* e outras redes sociais;
- Desenvolver e disponibilizar em outros dispositivos (*desktop*, página *web*, *web app*, etc.);
- Criação das redes sociais da aplicação para divulgações, transparência, dicas, orientações e para uma melhor comunicação com o usuário;
- Mais funcionalidades que surgirão com base nos dados e sugestões coletados dos usuários da aplicação.

REFERÊNCIAS

BRASIL poderá ter marco regulatório dos animais de estimação.2020. **Exame**.

Disponível em:

<https://exame.com/brasil/brasil-podera-ter-marco-regulatorio-dos-animais-de-estimacao/>. Acesso em: 25 jun. 2021.

CARVALHO, Luciana. 9 benefícios que bichos de estimação trazem à saúde.

Exame. 2013. Disponível em:

<https://exame.com/casual/9-beneficios-que-bichos-de-estimacao-trazem-a-saude/>. Acesso em: 25 jun. 2021.

COMO representar um diagrama de classes entre Gerente e funcionário. StackOverflow, [2017]. (Figura). Disponível em: <https://pt.stackoverflow.com/questions/207655/como-representar-um-diagrama-de-classes-entre-gerente-e-funcion%C3%A1rio>. Acesso em: 20 ago. 2021.

DANKI CODE. Qual a Linguagem de Programação mais usada no mercado? 2018. Disponível em: <https://blog.dankicode.com/qual-a-linguagem-de-programacao-mais-usada-no-mercado/>. Acesso em: 26 jun. 2021.

DEVELOPERS ANDROID. **Criar um projeto Android. 2021.** (Figura). Disponível em: <https://developer.android.com/training/basics/firstapp/creating-project?hl=pt-br>. Acesso em: 26 jun.. 2021.

DEVMÉDIA. **Os 4 pilares da programação orientada a Objetos**, 2014. (Figura). Disponível em: <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/92> Acesso em: 23 ago. 2021.

DIRETRIZ. Modelos de Casos de uso, 2006. (Figura). Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/use-case_model_CC121CF4.html Acesso em: 20 ago 2021.

ENTENDA o que é a Tecnologia Mobile e por que sua empresa deve adotá-la. **ReachBlog**, 2019. Disponível em: <https://blog.reach.com.br/entenda-o-que-e-a-tecnologia-mobile-e-por-que-sua-empresa-deve-adota-la>. Acesso em: 25 jun. 2021.

LEGACYOFRAPHA. Caso de uso expandido. **V8 Desenvolvedora**, 2009. (Figura). Disponível em: <https://v8desenvolvedora.wordpress.com/2009/12/11/page/2/>. Acesso em: 20 ago.2021,

OLIVEIRA, Marcio Pires; RODELLO, Ildeberto. Uma Experiência com a Utilização de Sistemas ERP para Auxiliar a Aquisição de Conhecimento sobre Processos de Negócio em um Curso de Bacharelado em Administração. **Research Gate**, 2015. (Figura). Disponível em: https://www.researchgate.net/figure/Figura-3-Diagrama-de-Atividades-modelado-pelo-Grupo-2-depois-da-utilizacao-do-sistema_fig3_280319043 . Acesso em: 20 ago.2021.

MARVEL, 2018. (Tela inicial do site). Disponível em: <https://marvelapp.com/>. Acesso em: 27 jun. 2021.

MARVEL, **Example Prototype Finance**. [2021]. (Figura). Disponível em: <https://marvelapp.com/examples/banking-app> . Acesso em: 29 jun. 2021.

SANTOS, Gabriel. O que significa Android? Conheça dez fatos sobre o sistema do

Google. **Techtudo**, 2019. Disponível em:
<https://www.techtudo.com.br/listas/2019/07/o-que-significa-android-conheca-dez-fatos-sobre-o-sistema-do-google.ghtml>. Acesso em 25 Jun 2021.

SOARES, Regiane. Hospitais recebem a visita de animais para animar pacientes. **Folha de São Paulo**, 2020. Disponível em:
<https://agora.folha.uol.com.br/sao-paulo/2020/02/hospitais-recebem-a-visita-de-animais-para-animar-pacientes.shtml>. Acesso em: 25 jun. 2021.

SPADARI, Ana. Introdução ao SGBD Oracle. **CCM**. Disponível em:
<https://br.ccm.net/contents/872-introducao-ao-sgbd-oracle>. Acesso em: 26 jun. 2021.

VARALLO, Thiago Java: história e principais conceitos. **DEVMEDIA**. Disponível em:
<https://www.devmedia.com.br/java-historia-e-principais-conceitos/25178>. Acesso em: 26 jun. 2021.

VENTURA, Plínio. Entendendo o Diagrama de Atividades da UML. **Até o momento**, 2016. (Figura). Disponível em:
<https://www.ateomomento.com.br/uml-diagrama-de-atividades/>. Acesso em: 27 jun.2021.