

**CENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ  
PRÓ-REITORIA ACADÊMICA - PROAC  
COORDENAÇÃO DOS CURSOS DE CIÊNCIA DA COMPUTAÇÃO  
E TECNOLOGIA**

**LUIZ EUZÉBIO DOS SANTOS NETO**

**SISTEMA DE CONTROLE E RELACIONAMENTO DE ANIMAIS DE ESTIMAÇÃO**

**JOÃO PESSOA - PB  
2016**

**LUIZ EUZÉBIO DOS SANTOS NETO**

**SISTEMA DE CONTROLE E RELACIONAMENTO DE ANIMAIS DE ESTIMAÇÃO**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário de João Pessoa - UNIPÊ, como pré-requisito para a obtenção do grau de Bacharel em Ciência da Computação, sob orientação do Prof. Ms. Walter Travassos Sarinho.

**JOÃO PESSOA - PB**

**2016**

S237s Neto, Luiz Euzébio Dos Santos.  
Sistema de Controle e Relacionamento de Animais de Estimação /  
Luiz Euzébio Dos Santos Neto.- João Pessoa, 2016. 71f.

Monografia (Curso de Ciências da Computação) –  
Centro Universitário de João Pessoa - UNIPÊ

1. Animal de estimação. 2. Aplicação Web. 3. Tecnologia da  
informação. 4. Relacionamento. I. Título.

UNIPÊ / BC

CDU – 636.99

## **LUIZ EUZÉBIO DOS SANTOS NETO**

### **SISTEMA DE CONTROLE E RELACIONAMENTO DE ANIMAIS DE ESTIMAÇÃO**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Centro Universitário de João Pessoa - UNIPÊ, como pré-requisito para a obtenção do grau de Bacharel em Ciência da Computação, apreciada pela Banca Examinadora composta pelos seguintes membros:

Aprovada em \_\_\_\_/\_\_\_\_/2016.

#### **BANCA EXAMINADORA**

---

Prof. (Ms.) Walter Travassos Sarinho (UNIPÊ)

---

Prof. (Ms.) Carlos Eduardo (UNIPÊ)

---

Prof. (Ms.) Wallace Bonfim (UNIPÊ)

Aos meus pais, irmã, minha esposa e a toda  
minha família que, com muito carinho e apoio,  
não mediram esforços para que eu chegasse até  
esta etapa da minha vida.

## **AGRADECIMENTOS**

Aos meus pais e familiares que sempre me apoiaram na minha formação, que sempre acreditaram na minha capacidade, em especial a minha tia Adriana Barbosa que me orientou neste trabalho.

Aos meus amigos, que me ajudaram na hora em que precisei de ajuda em relação neste trabalho.

E aos meus professores, que sem eles não teria o conhecimento para fazer este trabalho, e que me ajudaram na escolha do meu futuro profissional.

## RESUMO

Os animais de estimação estão em crescente muito grande no meio familiar, juntamente com esse crescimento os criadores precisam ter um controle sobre as atividades rotineiras de seus animais de estimação, e outra crescente muito grande é a tecnologia da informação. A criação de um sistema que faça esse controle dos animais de estimação com suas atividades diárias e que facilite o relacionamento entre os animais de estimação e criadores, será de grande ajuda para os criadores. O sistema também possui uma funcionalidade de busca para auxiliar em escolhas de futuros animais, apartir de um questionario respondido pelo usuario. Uma aplicação web, desenvolvido com a linguagem J2EE, banco de dados MySQL e servidor Apache TomCat.

**Palavras Chave:** Animal de estimação. Aplicação Web. Tecnologia da informação. Relacionamento.

## **ABSTRACT**

The pets are in growing too large in the Middle familiar, along with that growth designers need to have a control over the routine activities of their pets, and other growing too large is information technology. The creation of a system to do this control of pets with their daily activities and to facilitate the relationship between pets and breeders, will be of great help to farmers. The system also has a search tool to assist in future animal choices, from a questionnaire answered by the user. The system will be a web application, developed with J2EE, MySQL database and apache TomCat server.

**Keywords:** Pet. Web Application. Information Technology. Relationship.



## LISTA DE ILUSTRAÇÕES

Figura 01 – Estruturada X Orientado a Objetos .....	16
Figura 02 – Diagramas estruturais e comportamentais daUML .....	17
Figura 03 – Exemplo de diagrama Casos de Uso .....	18
Figura 04 – Exemplo de diagrama Classes .....	18
Figura 05 – Exemplo de diagrama de Sequência .....	19
Figura 06 – Exemplo de diagrama de Atividade .....	19
Figura 07 – Arquitetura Lógica do J2EE .....	21
Figura 08 – Principais componentes da arquitetura do tomcat .....	23
Figura 09 – Exemplo tabela de banco de dados relacional .....	24
Figura 10 – Exemplo de implementação em Java utilizando o Hibernate .....	25
Figura 11 –Número de animais de estimação por pessoa .....	28
Figura 12 – Diagrama de Casos de Uso .....	31
Figura 13 – Diagrama de sequência Realizando Cadastro .....	32
Figura 14 – Diagrama de sequência do Acompanhamento Animal .....	33
Figura 15 – Diagrama de sequência Realizar Teste .....	33
Figura 16 – Divisão do sistema na plataforma Eclipse .....	35
Figura 17 – Classe AbstractyEntities .....	35
Figura 18 – Classe entidade de Acompanhamento Animal .....	36
Figura 19 – Classe entidade Animal .....	37
Figura 20 – Classe entidade Pessoa .....	38
Figura 21 – Classe entidade Proprietario Animal .....	38
Figura 22 – Classe entidade Usuário .....	39

## **LISTA DE QUADROS**

Quadro 01 - Modificadores de encapsulamento .....	15
---	----

## **LISTA DE ABREVEATURAS E SIGLAS**

**BDOO** – Banco de Dados Orientado a Objetos

**IBGE** – Instituto Brasileiro de Geografia e Estatística

**JEE** – Java Platform Enterprise Edition

**JSP** – Java Server Pages

**LOO** – Linguagem Orientado a Objetos

**OO** – Orientado a Objeto

**OOA** – Análise Orientada a Objetos

**OOD** – Projeto Orientado a Objetos

**OOP** – Programação Orientada a Objetos

**UML** – Unified Modeling Language

**XML** – eXtensible Markup Language

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	11
1.1 RELEVÂNCIA DO ESTUDO .....	11
1.2 OBJETIVOS .....	12
1.2.1 Objetivo geral .....	12
1.2.2 Objetivos específicos .....	12
1.3 INDICAÇÃO DA METODOLOGIA .....	12
1.3.1 O método .....	12
1.3.2 O tipo de pesquisa .....	13
1.3.3 Publico alvo e amostra .....	13
1.4 ORGANIZAÇÃO DO TRABALHO .....	13
<b>2 REFERENCIAL TEÓRICO</b> .....	14
2.1 ORIENTAÇÃO A OBJETOS .....	14
2.2 LINGUAGEM DE MODELAGEM UNIFICADA .....	15
2.2.1 ASTAH COMMUNITY .....	20
2.3 JAVA PLATFORM ENTERPRISE EDITION .....	20
2.3.1 PLATAFORMA DE DESENVOLVIMENTO ECLIPSE .....	22
2.3.2 APACHE TOMCAT .....	22
2.4 BANCO DE DADOS RELACIONAL .....	23
2.4.1 BANCO DE DADOS MYSQL .....	24
2.4.2 HIBERNATE .....	25
<b>3 DESENVOLVIMENTO</b> .....	27
3.1 PESQUISA DE CAMPO.....	27
3.2 APRESENTAÇÃO DO SISTEMA .....	28
3.3 DIAGRAMAS UML .....	29
3.3.1 DIAGRAMA DE COMPONENTES .....	29
3.3.2 DIAGRAMA DE CLASSES .....	30
3.3.3 DIAGRAMA DE CASOS DE USO .....	30
3.3.4 DIAGRAMA DE SEQUÊNCIA .....	32
3.4 PROTOTIPAÇÃO DE PAGINAS .....	34
3.5 IMPLEMENTAÇÃO DO SISTEMA .....	34
3.6 PAGINAS DO SISTEMA .....	39
<b>4 CONSIDERAÇÕES FINAIS</b> .....	40
4.1 CONTRIBUIÇÕES ALCANÇADAS .....	40
4.2 DIFICULDADES ENCONTRADAS .....	40
4.3 TRABALHOS FUTUROS .....	40
<b>REFERÊNCIAS</b> .....	42
<b>APÊNDICE A</b> .....	46
<b>APÊNDICE B</b> .....	51
<b>APÊNDICE C</b> .....	52
<b>APÊNDICE D</b> .....	53
<b>APÊNDICE E</b> .....	54
<b>APÊNDICE F</b> .....	57
<b>APÊNDICE G</b> .....	66

## 1 INTRODUÇÃO

Hoje vivenciamos a era da busca rápida pela informação onde as pessoas, utilizando a rede mundial de computadores, conseguem informações relevantes sobre diversos temas em poucos segundos. Neste contexto, muitos proprietários de *pets*<sup>1</sup> buscam conciliar a criação de seus animais a sua rotina diária, isso implica numa oportunidade de serviço para o público que seja preciso e eficiente.

A utilização de aplicações que oferecem serviços para o pet não acontece somente na busca por locais para banho, tosa, compra de brinquedos, roupas, etc., ocorre também no quesito “relacionamento entre animais de estimação”. Devido á peculiaridades de raça e *pedigree* dos *pets*, a tarefa de encontrar um parceiro (a) para reprodução torna-se difícil. Disponibilizar uma aplicação que promova esta interação entre os proprietários dos *pets* é algo que facilitaria esta situação.

A cada dia mais pessoas estão procurando pets para ser parte de sua família. De acordo com o Instituto Brasileiro de Geografia e Estatística - IBGE em 2013 no Brasil o número de cães de estimação ultrapassou o número de crianças, sendo 52 milhões de cães para 45 milhões de crianças (Globo, 2015).

### 1.1 RELEVÂNCIA DO ESTUDO

No mundo de hoje, muitas pessoas consideram seus animais de estimação como membros da família, substituindo até namorado (a), filhos e amigos (UAI, 2014). Dessa forma, os cuidados com alimentação, controle de vacinas, cuidados médicos, banho e tosa são primordiais.

Existem estatísticas que evidenciam que país como o Brasil possui cerca de 132 milhões animais, média de 1,5 animais por habitante (ABINPET, 2013). Estes dados evidenciam um número expressivo do interesse nos *pets*. Tal realidade trás para a sociedade uma demanda significativa e que requer um olhar amplo para suas necessidades diante o avanço tecnológico.

Tendo em vista este cenário, o trabalho tem como foco entender como funciona o processo de controle de animais e prover uma solução de tecnologia da informação para auxiliar e maximizar o sucesso em unir pessoas e *pets*.

---

<sup>1</sup> Pet é uma palavra da língua inglesa cuja tradução é “animal de estimação”.

## 1.2 OBJETIVOS

A seguir será apresentado o objetivo geral deste trabalho, ou seja, resultado principal que se espera alcançar com a realização deste trabalho. Em seguida são apresentados os objetivos específicos que compreendem as etapas seguidas para alcançar o objetivo geral.

### 1.2.1 Objetivo geral

Desenvolver um prototipo de sistema web para ter controle sobre as atividades rotineiras dos animais de estimação, proporcionando também ao usuário a possibilidade de encontrar o *pet* ideal para o seu tipo de vida por meio de perguntas e respostas, obtendo conhecimento de cuidados com os *pets*, as principais doenças e como evitá-las.

### 1.2.2 Objetivo específico

São elencados os seguintes objetivos específicos:

- Representar as funções básicas do sistema proposto por meio de diagramas UML, modelando o problema e a solução proposta;
- Entender como funciona o processo de escolha de animais de estimação para específico tipo de vida da família, associando perfis de pessoas e características dos animais;
- Evidenciar as melhores práticas para controle do bem-estar dos animais;
- Desenvolver um protótipo do sistema proposto utilizando a linguagem Java, e executar teste de uso.

## 1.3 INDICAÇÃO DA METODOLOGIA

A seguir será apresentado a metodologia deste estudo por meio do método de pesquisa, o tipo da pesquisa e o público alvo.

### 1.3.1 O Método

O método utilizado para essa pesquisa foi o qualitativo. Esse método trata da repetição e dos conceitos já existentes, preocupando com a descrição direta da experiência, assim se

focando na profundidade da pesquisa para poder ter um destaque entre as já existentes, tendo um foco em como será aceita pelas pessoas. (Portal da Educação, 2013).

### **1.3.2 O Tipo de Pesquisa**

O tipo utilizado para essa pesquisa foi o do ponto de vista dos objetivos, sendo utilizado a pesquisa descritiva. Essa pesquisa descritiva tratasse do levantamento de informações através de questionário e observações, para poder descrever as características para as pessoas em foco (Portal da Educação, 2013).

### **1.3.3 Público Alvo e Amostra**

O público alvo dessa pesquisa são todas as pessoas que possuem ou desejem possuir um animal de estimação, pessoas essas que desejem obter conhecimentos sobre os diversos tipos de animais que possam ser domesticados e criados em casa, pessoas que desejem encontrar qual o seu pet ideal levando em consideração o ambiente em que mora (casa ou apartamento), idade, tempo a ser dedicado ao tal, e também pessoas que já possuam um ou mais pets e deseje fazer um controle de suas atividades rotineiras e dados pessoais.

## **1.4 ORGANIZAÇÃO DO TRABALHO**

Após este capítulo introdutório, o conteúdo deste trabalho organiza-se da seguinte forma:

- Capítulo 2 – <REFERENCIAL TEÓRICO> apresentará quais teorias e respectivos autores mais contribuíram para a realização do estudo e as bases teóricas para a realização deste trabalho, linguagem de modelagem UML;
- Capítulo 3 – <DESENVOLVIMENTO DO PROJETO> apresentará o desenvolvimento da pesquisa juntamente com a solução proposta e sua aplicabilidade;
- Capítulo 4 – <CONSIDERAÇÕES FINAIS> apresentará de forma conclusiva, respostas aos objetivos específicos propostos pelo trabalho, apresentando também limitações desta pesquisa e trabalhos futuros;

## 2 REFERENCIAL TEÓRICO

O referencial teórico deste trabalho tem como foco apresenta a programação orienta a objetos para extrair as características do mundo real para o virtual, UML para o desenvolvimento dos diagramas do projeto, o JEE como plataforma de desenvolvimento e banco de dados MySql para armazena as informações. Estes foram os conceitos utilizados na elicitação, documentação, análise e implementação do projeto.

### 2.1 PROGRAMAÇÃO ORIENTA A OBJETOS

Segundo Fernanda Farinelli (2007, p. 4) a programação orientada a objetos é uma tecnologia que enxerga os sistemas como sendo coleção de objetos integrantes. Ela permite melhorar a reusabilidade e extensibilidade dos softwares.

Essa tecnologia permitiu aproximar o mundo real do mundo virtual, mapeando os objetos tratados por linguagens de modelos ou de programação.

Os conceitos fundamentais relacionados a programação orientada a objetos são:

- **Classe:** Para Fernanda Farinelli (2007, p. 8) as classes são conjuntos de objetos com características e comportamentos comuns. As classes definem características e comportamentos de entidades do mundo real;
- **Encapsulamento:** Para Fernanda Farinelli (2007, p. 17) é utilizado para restringir o acesso a alguns objetos, escondendo seus dados. O acesso só é permitido utilizando métodos para acessar as variáveis de instância.
- **Herança:** Acontece quando uma classe (que é subclasse ou classe filha de outra classe) herda as características da classe superior na hierarquia (superclasse);
- **Objeto:** Para Martin (1995, p. 18) a definição de um objeto é “qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados”;
- **Package:** Traduzindo para o português “Pacote”, é o mecanismo de agrupar as classes, relacionadas entre si;
- **Polimorfismo:** Para Fernanda Farinelli (2007, p. 23) é definido como sendo o nome dado à capacidade que objetos diferentes têm de responder a uma mesma mensagem. Objetos podem ser substituídos sem que os programas que usam a interface sejam afetados.



Além da programação orientada a objetos, existe a programação estruturada. Luiz Fernando (2015, p.1) conceitua este tipo de programa como uma formalização da ideia de divisão em blocos funcionais. Isto força o programador a saber exatamente o estado do programa antes e depois de cada bloco. Evita também, o chamado “código espaguete”, onde não se tem noção em que estágio da execução o programa se encontra durante uma determinada instrução.

A Figura 01 ilustra o comparativo entre a programação estruturada e a programação orientada a objetos.

Figura 01: Estruturada X Orientado a Objetos



Fonte: Os 4 pilares da programação orientada a objetos, 2014.

Na programação estruturada temos a linguagem C como principal exemplo. É considerada uma linguagem de baixo nível. Sua programação estruturada é toda sequencial, tendo um arquivo de dados global e vários procedimentos que são chamados dentro da global e compilados de forma sequencial. Já a programação orientada a objetos podemos ter várias classes distintas com vários métodos cada uma. Outra vantagem é com relação ao encapsulamento (adiciona mais segurança). Outro recurso que dá vantagem a programação orientada a objetos é a herança, podendo assim uma classe herdar de outra reaproveitando código.

Devido a todas as vantagens apresentadas, o sistema proposto neste trabalho será implementado em programação orientada a objetos. Para fazer a representação visual do sistema proposto, será utilizada a UML (*Unified Modeling Language*).

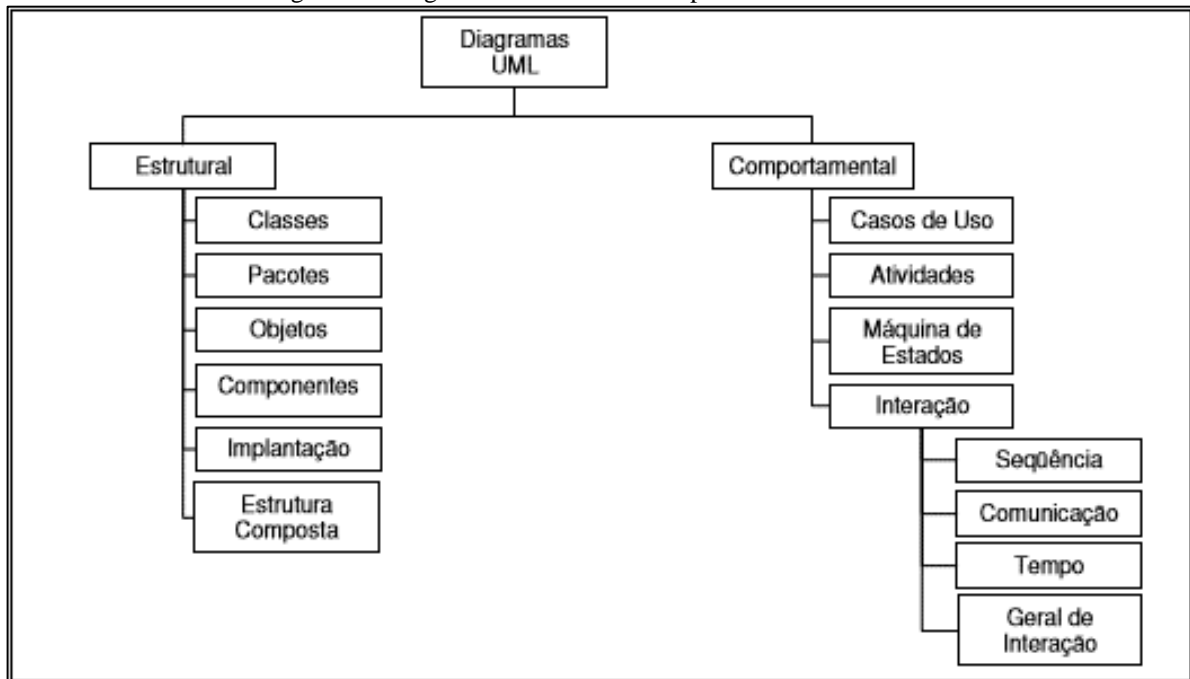
## 2.2 LINGUAGEM DE MODELAGEM UNIFICADA

A UML é uma linguagem visual para especificação de sistemas OO (Orientado a Objeto). Este tipo de modelagem fornece uma representação de forma gráfica os elementos essenciais do paradigma orientado a objetos (Cesar Augusto, 2013).

A UML surgiu no ano de 1995 a partir de Jim Rumbaugh, Grady Booch e Ivar Jacobson, com sua primeira versão sendo a 0.8, e evoluindo com o tempo até a versão 2.1. E esta é a versão utilizada atualmente.

Segundo Cesar Augusto (2013), “A modelagem UML é dividida em diagramas, sendo de modelo estrutural e comportamental”. A Figura 02 representa a divisão e todos os diagramas da UML.

Figura 02: Diagramas estruturais e comportamentais da UML.



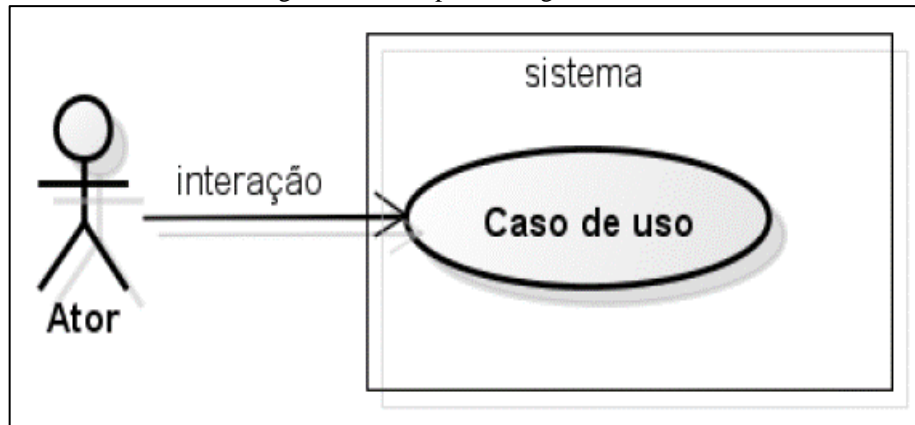
Fonte: Cesar Augusto, 2013.

Os diagramas UML estruturais são os que priorizam a descrição estática das estruturas de um sistema. Os diagramas UML comportamentais são os que detalham o funcionamento de partes ou processos de negócio do sistema.

O diagrama de caso de uso (comportamental) ajuda a descrever como o (s) usuário (s) interagem com o sistema, facilitando a organização dos requisitos, dando uma visão externa do sistema. (FERREIRA, 2009) “Os diagramas de caso de uso apresentam uma visão externa sobre como esses elementos podem ser utilizados no contexto do sistema sendo representado”.

A Figura 03 representa um caso de uso genérico para representação. Onde um ator interage com uma função do sistema.

Figura 03: Exemplo de diagrama de caso de uso.

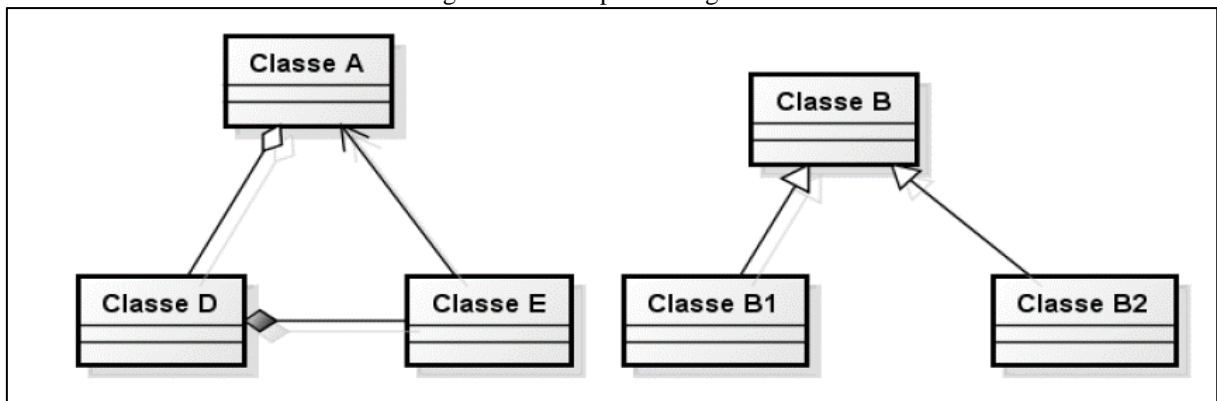


Fonte: Dados da pesquisa, 2016.

O diagrama de classes representa as classes do sistema seguido de seus atributos e métodos, e com os relacionamentos entre as classes.

A Figura 04 representa um diagrama de classes de uso genérico para representação. Onde nas classes podemos ver do lado esquerdo os relacionamentos de associação, composição e agregação. Do lado direito o relacionamento de herança.

Figura 04: Exemplo de diagrama de classes.

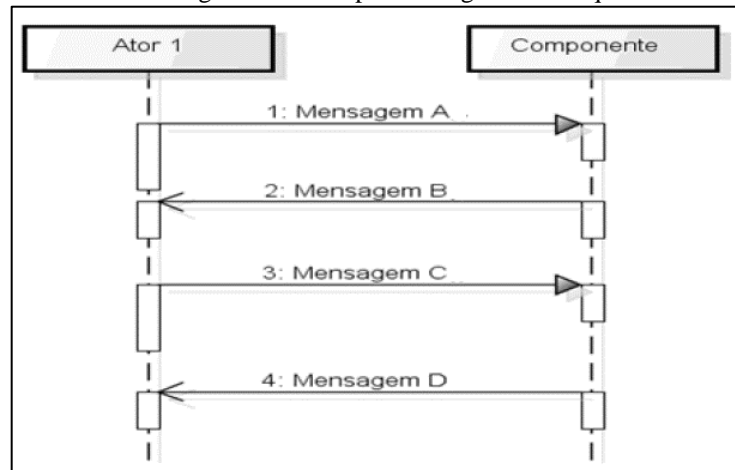


Fonte: Dados da pesquisa, 2016.

A Figura 05 representa um diagrama de sequência genérico para representação. Onde o ator faz a solicitação a outro ator que retorna uma resposta para continuação do processo, repetindo até o termino do processo.

O diagrama de sequência enfatiza a ordenação temporal das mensagens. (Furlan, 1998) diz que o diagrama de sequência expõe o aspecto do modelo que destaca o comportamento dos objetos em um sistema, incluindo suas operações, interações, colaborações, histórias de estado e sequência temporal de mensagem e representação explícita de atividades de operações”.

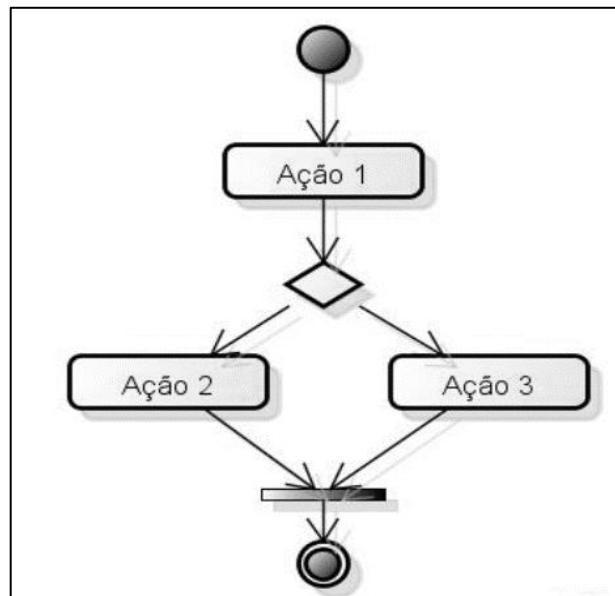
Figura 05: Exemplo de diagrama de sequencia



Fonte: Dados da Pesquisa, 2016.

A Figura 06 representa um diagrama de atividade genérico para representação. O diagrama de atividade é basicamente uma sequência de fluxo, onde é visto o fluxo de controle de uma atividade. Esse diagrama é utilizado para mostrar passo a passo de como cada atividade deve ser realizada desde seu início até o termino.

Figura 06: Exemplo de um diagrama de atividade.



Fonte: Dados da Pesquisa, 2016.

Este diagrama de atividades contém o estado de início, as ações realizadas e o estado de termino.

Para o desenvolvimento dos diagramas existem ferramentas específicas de modelagem virtual. A ferramenta escolhida foi o Astah community<sup>2</sup>.

<sup>2</sup> Link de acesso ao Astah Community <http://astah.net/editions/community>

### 2.2.1 Astah community

O Astah community é um framework para a criação de diagramas UML, sendo um do framework mais conceituado neste quesito. O Astah está disponibilizado em 2 versões uma gratuita e outra paga.

Este framework do suporte a tais recursos:

- UML 2.1;
- Diagramas de Classe, Casos de Uso, Sequencia, Atividade, Comunicação, Componentes, Máquina de Estado, Implantação, Estrutura, Objetos e Pacotes;
- Ajustes de tamanho e do alinhamento dos diagramas;
- Exportação de imagens;
- Impressão dos diagramas.

Como o UML é uma linguagem de modelagem para representação virtual, o JEE (*Java Platform Enterprise Edition*) ou J2EE foi a linguagem de programação escolhida para desenvolver a aplicação a partir dos diagramas da UML.

### 2.3 JAVA PLATFORM ENTERPRISE EDITION

A plataforma Java possui uma grande infinidade de possibilidades para o desenvolvimento de aplicações, A escolha dessa plataforma foi determinada pelo maior conhecimento com tal linguagem.

O J2EE é uma dentre as três divisões que a plataforma Java oferece: (i) o JSE (ou J2SE), (ii) o JEE (ou J2EE) e (iii) o JME (J2ME). A plataforma J2EE é para desenvolvimento de aplicações distribuídas. A arquitetura J2EE apresenta várias camadas composta por componentes e serviços que são providos por um container.

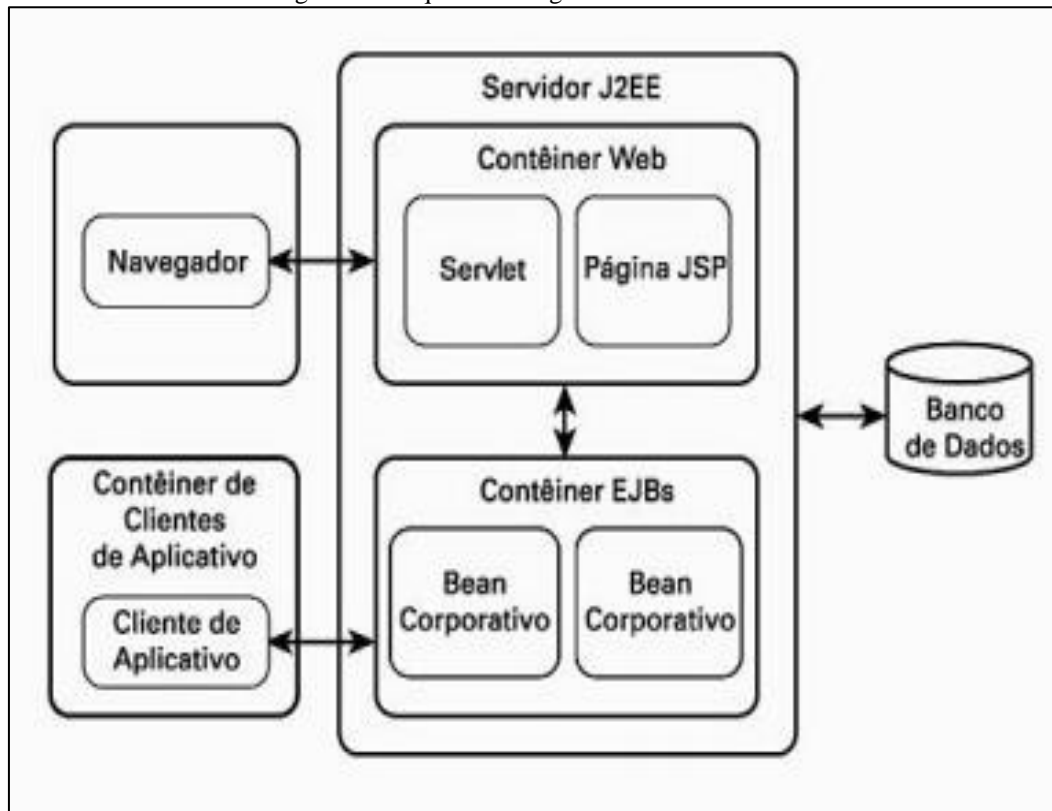
A plataforma JEE contém uma série de especificações, cada uma com funcionalidades distintas, composta por componentes e serviços tudo em referência a um container, que o conjunto de vários containers para forma o JEE.

A Figura 07 representa a arquitetura lógica da plataforma de desenvolvimento J2EE. Os principais blocos são:

- Navegador: Software que permite a visualização da aplicação. Faz a visualização dos Contêiner Web;

- Contêiner de clientes de aplicativo: Esse Contêiner pode ser uma aplicação Java, um applet, uma página JSP, um servlet, um EJB ou outros. Sendo sua conexão feita pelo EJB;
- Contêiner web: A camada web fornece toda a lógica a camada do cliente que são implementadas por JSP (*Java Server Pages*) e *servlets*. O JSP são as páginas de apresentação ao cliente. Já o *servlet* é o controlador das ações do usuário;
- Contêiner EJBs: Oferece a facilidade de desenvolvimento de componentes, tendo comunicação com o Contêiner de Cliente e Contêiner web;
- Banco de dados: Local onde são armazenadas as informações, faz comunicação com o Servidor J2EE para que possa armazenar as informações persistidas.

Figura 07: Arquitetura Lógica do J2EE.



Fonte: JEE entendendo a plataforma, 2015.

O JEE é apenas a linguagem de programação e para que essa linguagem seja desenvolvida é necessária uma plataforma de desenvolvimento. A plataforma escolhida para o desenvolvimento do sistema foi o Eclipse<sup>3</sup>.

<sup>3</sup> Link de acesso a plataforma Eclipse <https://eclipse.org/>.

### 2.3.1 Plataforma de desenvolvimento Eclipse

O Eclipse é uma plataforma de desenvolvimento de software livre extensível baseado na linguagem Java. Extensível pois pode ser adicionado novos conjuntos de *plug-ins*. Esta plataforma também é um software livre (Eclipse, 2016).

Mesmo sendo baseado em Java, esta plataforma não se limita só a desenvolvimento em Java, mas também em: C/C++, COBOL, PHP. Para o desenvolvimento com o JEE é necessário utilizar o Eclipse com o conjunto de *plug-ins* para web, o utilizado neste projeto será o Eclipse Luna SR2. Para a execução do código compilado, se faz necessário a figura de um servidor Java. O servidor escolhido foi o apache TomCat<sup>4</sup>.

### 2.3.2 Apache TomCat

O Apache TomCat é um servidor Java, implementado suas principais tecnologias o Java Servlet e JSP (Java Server Pages). O Apache é um software livre, desenvolvido pela Apache Software Foundation desenvolvida na década de 90 por Rob McCool.

Possui ferramenta de gerenciamento com configuração fácil e gerenciamento simplificado. Também pode ser configurado manualmente por meio de XML (*eXtensible Markup Language*) (Apache, 2016).

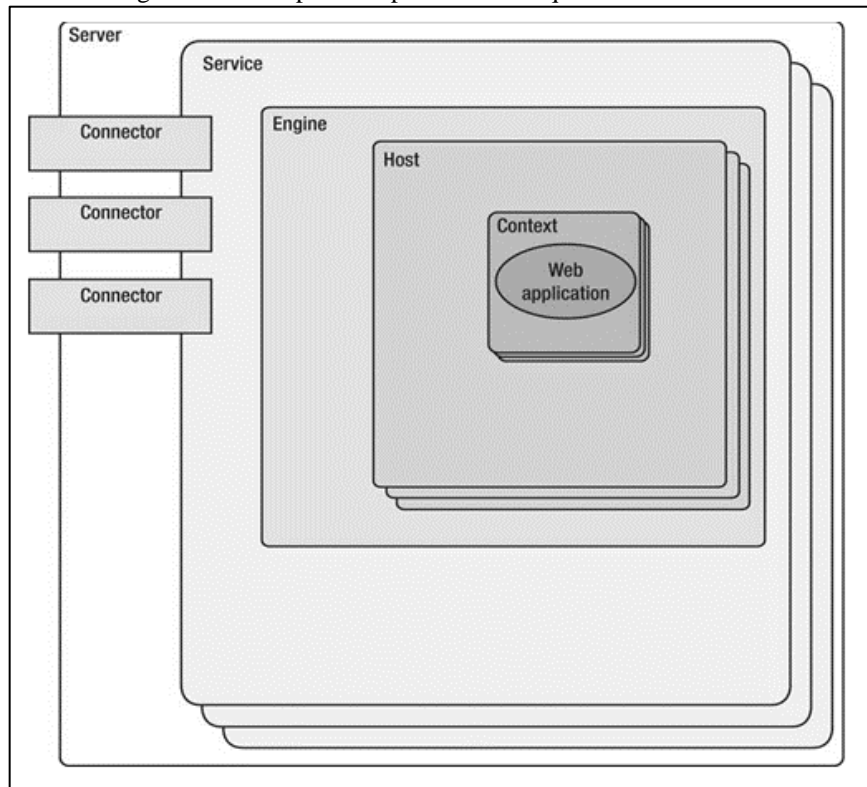
A Figura 08 apresenta os principais componentes da arquitetura do TomCat. É mostrado a divisão desde os conectores até a aplicação em si. Os principais componentes da figura são:

- *Server*: Esta camada é a mais alta em instancia do Tomcat, representando todo o *Catalina Servlet Engine* e podendo possuir um ou mais *Service*;
- *Service*: Esta camada pode possuir vários *Connector* e somente um *Engine*;
- *Connector*: Define as classes de gerenciamento de requisições e respostas entre todas as chamadas da aplicação cliente;
- *Engine*: Gerencia as requisições recebidas pelos *Connectores*;
- *Host*: A camada Host pode possuir um ou mais aplicações web, sendo ele representado por um *Context* individual;
- *Context*: Cada *Context* representa uma única aplicação web.

---

<sup>4</sup> Link de acesso ao servidor Java Apache TomCat <http://tomcat.apache.org/>.

Figura 08: Principais componentes da arquitetura do TomCat.



Fonte: Introduzindo o servidor de aplicação apache tomcat, 2012.

Para manipular e armazenar os dados que serão utilizados na aplicação, é necessário um banco de dados. Foi selecionado o banco de dados relacional.

## 2.4 BANCO DE DADOS RELACIONAL

Um banco de dados (sua abreviatura é BD, em inglês DB, *database*) é uma entidade na qual é possível armazenar dados de maneira estruturada e com a menor redundância possível. Estes dados devem poder ser utilizados por programas, por usuários diferentes. Assim, a noção básica de dados é acoplada geralmente a uma rede, a fim de poder pôr, conjuntamente, estas informações, daí o nome banco. (KIOSKEA, 2014).

O banco de dados relacional é uma ferramenta para armazenamento de dados, o modelo relacional faz a modelagem dos objetos por tabelas, onde cada tabela possui linhas e colunas. (DCA/FEE/Unicamp, 2002).

Eric Lemes (2009) divide os conceitos de bancos de dados relacionais em:

- Chave: A chave é a forma de identificar a tabela, o atributo que a torna única.  
Exemplo: ID, CPF, CNPJ;
- Colunas (ou Atributos): As colunas de uma tabela representam os atributos.  
Exemplo: Uma coluna de telefone, só serão armazenados números telefônicos;



- Registros (ou Tuplas): Os registros são formados pelas linhas da tabela, e nelas estão contidos os dados armazenados para cada atributo da tabela;
- Tabelas (ou Entidades): São estruturas simples de linhas e colunas, podendo em cada banco de dados possuir uma ou várias tabelas.

A Figura 09 ilustra uma tabela de banco de dados com atributos persistidos.

Figura 09: Exemplo tabela de banco de dados relacional.

<b>empregado</b>				
<b>id_empregado</b>	<b>primeiro-nome</b>	<b>último-nome</b>	<b>telefone</b>	<b>endereço</b>
1	Big	BOSS	936854270	big.boss@company.com
2	John	DOE	936854271	john.doe@company.com
3	Linus	TORVALDS	936854272	linus.torvalds@company.com
4	Jimmy	WALES	936854273	jimmy.wales@company.com
5	Larry	PAGE	936854274	larry.page@company.com

Fonte: Banco de dados Relacional, 2015.

Na Figura 09 o empregado é o nome da tabela. A linha imediatamente inferior (em negrito) representam as colunas (ou atributos): o id\_empregado (chave primaria), primeiro-nome, último-nome, telefone, endereço. Já as outras linhas abaixo representam registros.

Para desenvolver o banco de dados relacional é necessário um SGBD (Sistema de Gerenciamento de Banco de Dados), o escolhido foi o MySQL<sup>5</sup>.

### 2.4.1 Banco de dados MySQL

O MySQL é um SGBD que utiliza a linguagem SQL como interface. Sendo muito utilizado nas aplicações web e software de cadastro de lojas. Este SGBD foi criado na década de 80, por David Axmark, Allan Larsson e Michael Widenius (MySQL, 2016).

A grande vantagem do MySQL é por ser protegido por uma licença de software livre, desenvolvida pela GNU. Outras das grandes vantagens são: maior facilidade de programação, funções simples, pode ser totalmente modificado, entre outras.

Algumas das principais empresas utilizadoras do MySQL estão: Nasa, Friendster, Banco bradesco, Dataprev, Sony, Cisco, Google, entre muitas outras.

A programação orientada a objetos e o banco de dados relacional trabalham em paradigmas diferente. É preciso fazer um mapeamento entre os dados que serão persistidos nas tabelas do modelo relacional com as classes e objetos da programação orientada a objetos.

<sup>5</sup> Link de acesso ao SGBD MySQL <https://www.mysql.com/>.

Para facilitar a persistência de dados no banco de dados relacional, foi escolhido o framework de persistência Hibernate<sup>6</sup>. Ele será utilizado no Eclipse.

## 2.4.2 Framework de Persistência Hibernate

O Hibernate é um framework ORM (*Object Relational Mapping*) de persistência em banco de dados, sendo um software livre do grupo JBoss<sup>7</sup>, com muitos anos na liderança do mercado. Este framework faz a persistência de objetos Java em um banco de dados relacional, para isso, os objetos devem ser mapeados nas tabelas do banco de dados. A Figura 10 ilustra um exemplo de implementação na plataforma Java com o framework Hibernate (Caelum, 2016).

Figura 10: Exemplo de implementação em Java utilizando o Hibernate.

```
@Entity
@SequenceGenerator(name = "pessoa_id", sequenceName = "pessoa_seq", allocationSize = 1)
@Inheritance(strategy = InheritanceType.JOINED)
public class Pessoa extends AbstractEntity{

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "pessoa_id")
    private Long id;

    private String nome;
    private Long cpf;
    private String email;
    private String telefone;

    @Embedded
    private Endereco endereco;
```

Fonte: Dados da pesquisa, 2016.

A Figura 10 mostra uma classe escrita em Java com Hibernate. É realizado o mapeamento da classe Pessoa utilizando uma *annotation* **@Entity** para que possa ser criado no banco de dados relacional uma tabela com o nome da classe, neste caso Pessoa.

Com a apresentação dos conceitos e ferramentas que foram escolhidos e utilizados no desenvolvimento desse sistema, sendo eles a: programação orientada a objetos; linguagem de modelagem unificada e sua ferramenta de desenvolvimento o Astah Community; o J2EE com a plataforma de desenvolvimento Eclipse e o Apache TomCat como servidor Java; o banco de dados relacional com o SGBD MySQL e o framework de persistência Hibernate.

<sup>6</sup> Link de acesso ao framework de persistência Hibernate (<http://hibernate.org/>).

<sup>7</sup> Link de acesso ao grupo criador do Hibernate <http://tools.jboss.org/>.

O Capítulo 3 apresentará a aplicação dos conceitos e ferramentas descritas neste capítulo para o desenvolvimento do sistema proposto.

### 3 DESENVOLVIMENTO DO SISTEMA

As etapas para a modelagem e a implementação do sistema para acompanhamento de animais de estimação, seguiram o modelo de desenvolvimento de software sequencial linear proposto por Pressman (2011).

As etapas definidas foram:

- **Levantamento dos requisitos:** Os requisitos foram definidos tendo como base a necessidade de um controle de acompanhamento dos Pets utilizado por criadores. Foi conversado com 5 criadores e 1 empresário de um PetShop, tendo em vista a dificuldade no controle individual para cada espécie de animal de estimação.
- **Análise e projeto:** Os requisitos levantados foram modelados por meio dos diagramas da UML, sendo esses: o diagrama de casos de usos, de classes, componentes, sequencia e atividade. As tabelas e respectivos campos do banco de dados foram definidos a partir das classes e seus respectivos mapeamentos.
- **Implementação:** A implementação do sistema será realizada por meio da linguagem JEE, utilizando a plataforma Eclipse.
- **Testes:** Os testes serão realizados pelo autor deste trabalho visando identificar erros de codificação e testes para verificar se as funcionalidades do sistema foram devidamente implementadas.

Com as etapas para o desenvolvimento do projeto definidas, iniciou-se a etapa de levantamento de requisitos por meio da pesquisa de campo.

#### 3.1 PESQUISAS DE CAMPO

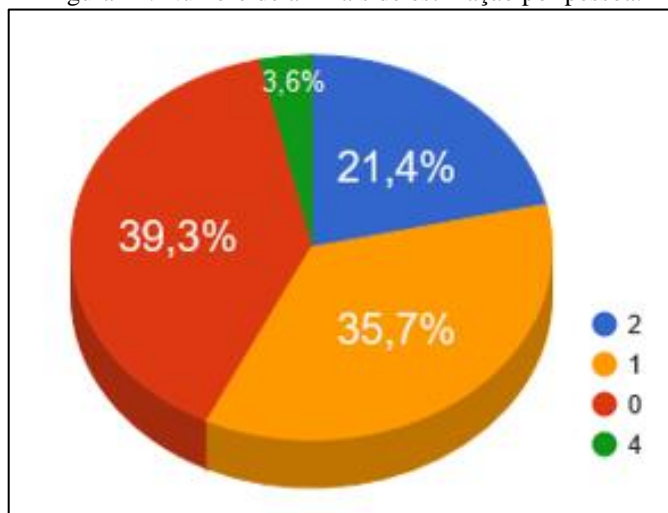
Na pesquisa de campo foi utilizado um questionário (localizado no Apêndice A) criado para levantar informações sobre os animais de estimação. Questionário respondido por pessoas que possui e não possuem animais de estimacões, com objetivo de investigar algumas questões como: quantos animais de estimação cada pessoa possui, as espécies diferentes de animais de estimação e identificar as principais atividades diárias realizadas pelos pets. Abaixo estão os dados obtidos pelo questionário.

- Enviado para **29 pessoas**;
- Total de **29 respostas (100%)**;
- Media de **0.86** animais de estimação por pessoa.

Das 100% das pessoas que responderam o questionário, 60,7% dos entrevistados disseram que possuem animais de estimação e 39,7% disseram que não possuem animais.

A Figura 11 mostra um gráfico de quantos animais de estimação cada pessoa que respondeu o questionário possui.

Figura 11: Número de animais de estimação por pessoa.



Fonte: Dados da pesquisa, 2016.

Das pessoas que responderam o questionário, 39,3% dos entrevistados disseram não possuir nenhum animal de estimação (na cor vermelha), 35,7% disseram que possui 1 (um) animal de estimação (na cor laranja), 21,4% disseram que possui 2 (dois) animais de estimação (na cor azul) e 3,6% disseram que possui 4 (quatro) animais de estimação.

Com o resultado do questionário aplicado, será descrita uma breve apresentação do sistema para melhor entendimento das funcionalidades sistema.

### 3.2 APRESENTAÇÃO DO SISTEMA

O sistema proposto tem como finalidade uma maior interação entre pessoas e animais de estimação. O sistema será dividido em duas partes, sendo uma pública e outra privada. Esta última, com acesso disponibilizado após efetuar cadastro. No Apêndice F 1 está localizado o protótipo da primeira tela do sistema.

O módulo pública contará com acesso direto sem a necessidade de se cadastrar. Nesta parte será apresentado inicialmente a ‘Página Inicial’. É onde será efetuado primeiro contato do usuário com o sistema. A tela ‘Quem Somos’ apresentará a finalidade do sistema e o perfil do criador. A tela de ‘Conhecimento’ será dividido em uma tela de ‘Dicas de como cuidar’ (apresentará dicas para os usuários de cuidados básicos com seus Pets) e outra tela de ‘Principais Doenças’ (apresentará as doenças com seus sintomas mais comuns e o que fazer

caso seu Pet estiver com algum sintoma). A tela de ‘Realizar Teste’ será um questionário com perguntas objetivas e de acordo com as respostas obtidas o sistema indicará qual animal de estimação ideal para o usuário e seu estilo de vida. E a tela de ‘Contato’ será a tela para o usuário apresentar sugestões ou reclamações do sistema.

A parte privada mostra o controle dos Pets. Poderá ser feito o acompanhamento dos animais de estimação de cada criador. Podendo controlar a alimentação através da quantidade e horário de cada refeição, como também, acompanhamento de atividades rotineiras do Pet, por exemplo: caminhada, natação, adestramento e controle do cartão de vacina do (s) animais de estimação.

Todas as funcionalidades disponibilizadas pelo sistema são descritas em diagramas UML.

### 3.3 DIAGRAMAS UML

Na modelagem para o desenvolvimento desse sistema foram utilizados os seguintes diagramas: componentes, classes, casos de uso, sequência e atividade respectivamente.

#### 3.3.1 Diagrama de componentes

O diagrama de classes mostrado no Apêndice C representa a divisão dos componentes do sistema. A divisão ocorre da seguinte maneira:

- **View:** A parte View representa o que o usuário visualiza, no caso as telas. A View contém HTML5 (Hyper Text Markup Language), onde é a linguagem de marcação de texto, o JSP utiliza HTML como linguagem base no desenvolvimento de páginas com J2EE e o CSS (Cascading Style Sheets) que define os estilos das páginas web. Possui somente comunicação com os Controllers;
- **Controllers:** Controla as solicitações da aplicação web e realiza o comando solicitado;
- **Entities:** São as entidades da aplicação. Possui comunicação com os DAOs.
- **Services:** Parte responsável por controlar os serviços middleware da aplicação.
- **DAOs:** Parte responsável por encapsular e abstrair todo acesso aos dados da aplicação, um exemplo é o acesso ao JDBC.
- **Conexão:** Realiza a conexão dos DAOs através do Hibernate para a persistência dos dados no repositório;

- **Repositório:** O repositório é o local onde armazenam os dados persistidos, o banco de dados. Com única comunicação através do Hibernate.

Após uma visão ampla do sistema, de todas as partes contidas, será apresentado o diagrama de classes que contém o pacote Entities.

### 3.3.2 Diagrama de classes

O diagrama de classes mostrado no Apêndice D representam as classes do pacote “entities” persistidas em banco de dados. A classe *AbstractyEntities* é uma classe abstrata que possui dois atributos *abstracty* *getId()* e *setId()*, essa classe faz herança com todas as classes do pacote entities, exceto: *PorteAnimal* e *Endereco*.

A classe *Animal* onde são armazenados os dados dos animais e a classe de *AcompanhamentoAnimal* armazena e controla os dados de acompanhamento das atividades rotineiras dos animais, que esta ligada à classe de *Alimentacao*, que controla a alimentação diária dos animais de estimação.

Na classe *AcompanhamentoAnimal* também está ligada a classe *CartaoDeVacina* que armazena e controla as vacinas já tomadas e as futuras vacinas. Já a classe *ProprietarioAnimal* herda os atributos e métodos da classe *Pessoa* onde são armazenados os dados pessoas do proprietário do animal, e também possui uma lista de todos os seus animais de estimação cadastrados.

### 3.3.3 Diagrama de casos de uso

O diagrama de caso de uso, têm por finalidade descrever e definir os requisitos funcionais do sistema. No Apêndice E encontra-se detalhada cada fluxo dos casos de usos.

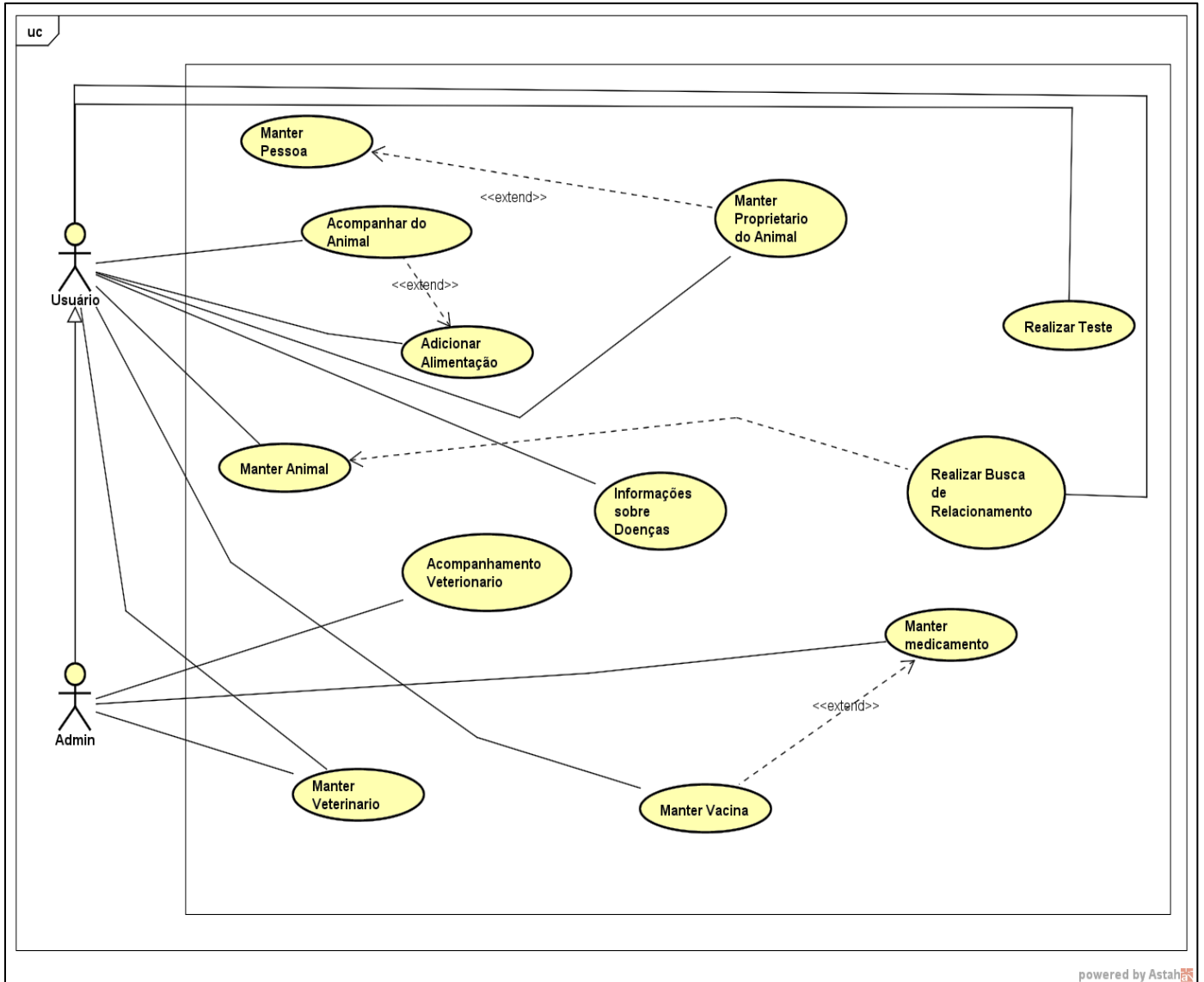
A Figura 12 representa os Casos de Uso do sistema.

O ator *Usuário* tem a permissão para cadastrar seus dados pessoas, adicionar a alimentação do animal de estimação, onde essa alimentação está incluída no acompanhamento animal, tem a permissão para fazer o acompanhamento das atividades, também possui permissão para realizar o teste para o melhor animal de estimação, cadastrar os animais de estimação e cadastrar as vacinas no cartão de vacinas do Pet.

Já o ator *Admin* herda todos as permissões do ator *Usuário* e possui permissão para cadastrar medicamentos, cadastrar veterinario e realizar o acompanhamento do veterinario.

Após apresentar os principais fluxos dos diagramas de casos de usos, será apresentado o diagrama de classes do sistema proposto.

Figura 12: Diagrama de Casos de Uso.



Fonte: Dados da pesquisa, 2016.

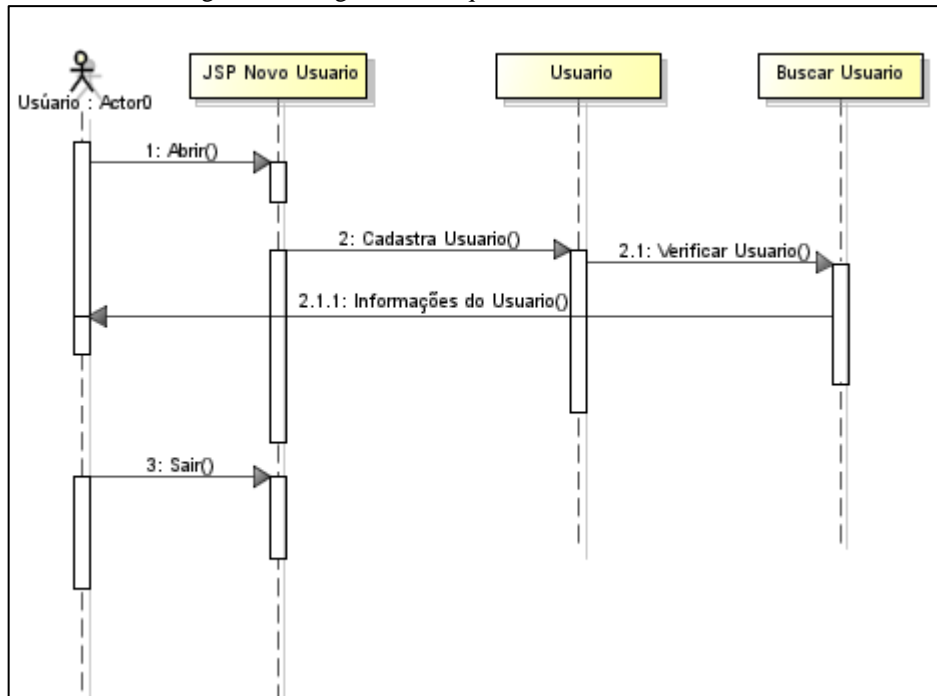
### 3.3.4 Diagrama de sequência

As Figuras 13 até 15 apresentam os diagramas de sequência definidos para a modelagem do sistema. Por meio desses diagramas é possível visualizar a ordem temporal em que os processos são executados.



A Figura 13 apresenta o diagrama de sequência que realiza o cadastro do usuário no sistema.

Figura 13: Diagrama de sequência Realizando Cadastro.



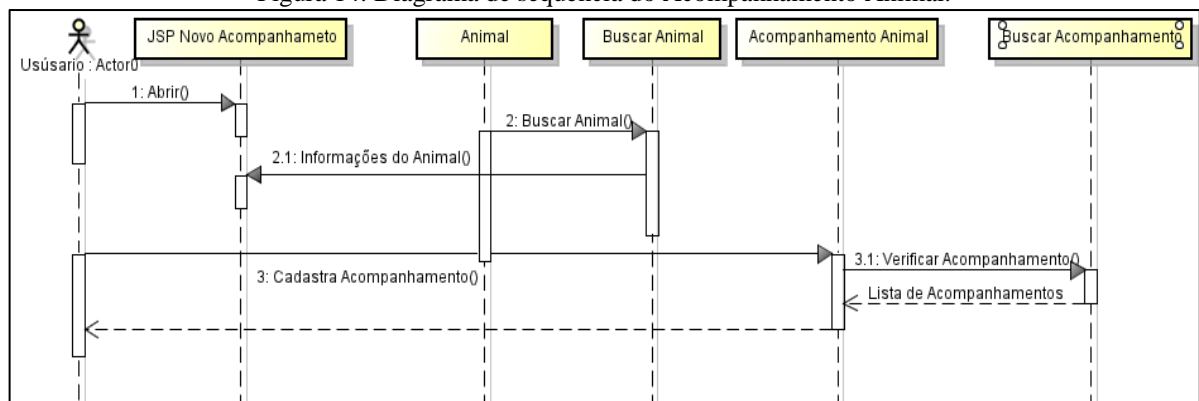
Fonte: Dados da pesquisa, 2016.

O diagrama da Figura 13 pode ser lido da seguinte maneira:

- 1: O Usuário solicita ao sistema a abertura da interface de cadastro de usuários;
- 2: O Usuário preenche os dados solicitados para o cadastro do usuário;
- 2.1: Após a verificação, não havendo o respectivo cadastro, o sistema efetiva o cadastro;
- 2.1.1: O sistema retorna uma mensagem de conclusão do cadastro;
- 3: O Usuário solicita para sair do sistema.

A Figura 14 representa o diagrama de sequência em que o usuário solicita o cadastro de um acompanhamento de Animal.

Figura 14: Diagrama de sequência do Acompanhamento Animal.



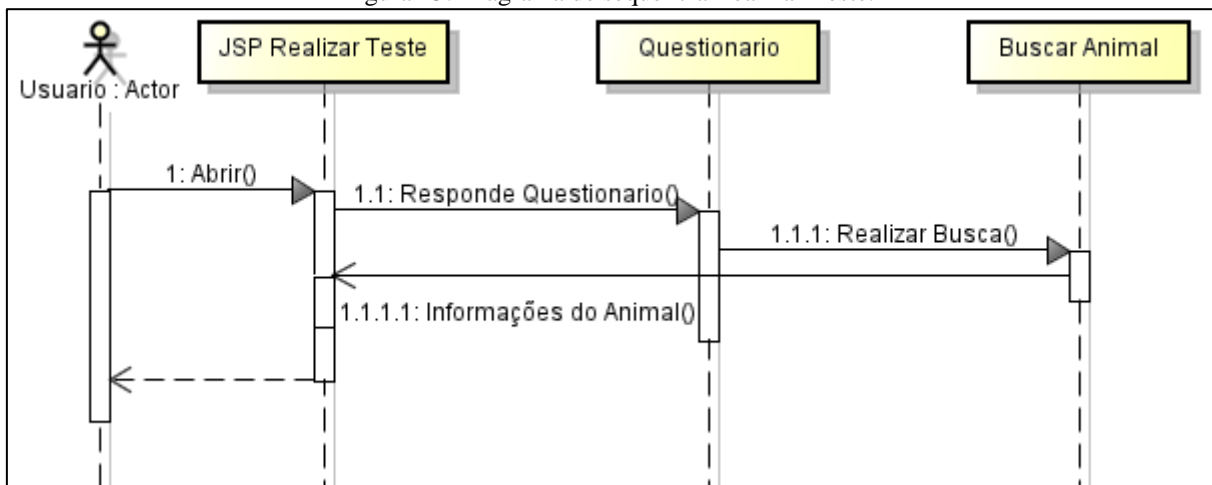
Fonte: Dados da pesquisa, 2016.

O diagrama da Figura 14 pode ser lido da seguinte maneira:

- 1: O Usuário solicita ao sistema a abertura da interface de Acompanhamento do Animal;
- 2: O Usuário seleciona o Animal a realizar o cadastro do Acompanhamento;
- 2.1: Retorna as informações do animal para poder fazer o cadastro do novo acompanhamento;
- 3: O Usuário solicita ao sistema a realização do novo cadastro de um Acompanhamento;
- 3.1: Após a verificação, não havendo o respectivo cadastro, o sistema efetiva o cadastro.

A Figura 15 representa o diagrama de sequência em que o usuário solicita ao sistema para realizar o teste para encontrar o melhor animal de estimação para seu estilo de vida.

Figura 15: Diagrama de sequência Realizar Teste.



Fonte: Dados da pesquisa, 2016.

O diagrama da Figura 15 pode ser lido da seguinte maneira:

- 1: O Usuário solicita ao sistema a abertura da interface para Realizar o Teste;
- 1.1: O sistema abriu o questionário para o Usuário responder;
- 1.1.1: O sistema a partir das respostas do Usuário fará uma busca online;
- 1.1.1.1: Retorna as informações da busca para mostrar ao Usuário o animal de estimação do resultado.

### 3.4 PROTIPACÃO DE PAGINAS

A prototipação ou criação de modelos é um processo de facilitação dos modelos visuais de telas do sistema. Na prototipação foi utilizado a ferramenta Balsamiq Mockup<sup>8</sup>.

O Apêndice F 2 representa o protótipo da tela de quem somos do sistema. Contendo o objetivo e descrição do sistema e perfil do criador do projeto.

O Apêndice F 3 representa o protótipo da tela de conhecimento possui uma seleção de animal de estimação para obter informações, e mais 2 (dois) sub menus, um de ‘Dicas de como cuidar’ e ‘Principais Doenças’.

O Apêndice F 4 representa o protótipo da tela de Realizar Teste, essa tela é responsável por conter as perguntas para identificar o melhor animal de estimação para o usuário que realizar o teste, e o Apêndice F 5 representa o protótipo da tela com o resultado do teste realizado.

O Apêndice F 6 representa o protótipo da tela de Contato, contendo um formulário para os usuários darem sugestões e reclamações do sistema, com campos de nome, email, assunto e descrição.

O Apêndice F 7 representa o protótipo da tela de Login, contendo um formulário com campos de ‘Login’ e ‘Senha’ para entrar no sistema e um botão de ‘Cadastre-se’ para o usuário que ainda não for cadastrado realizar o seu.

Após a prototipação das telas e sabendo o que cada uma ira realizar, precisa implementar o sistema.

### 3.5 IMPLEMENTACÃO DO SISTEMA

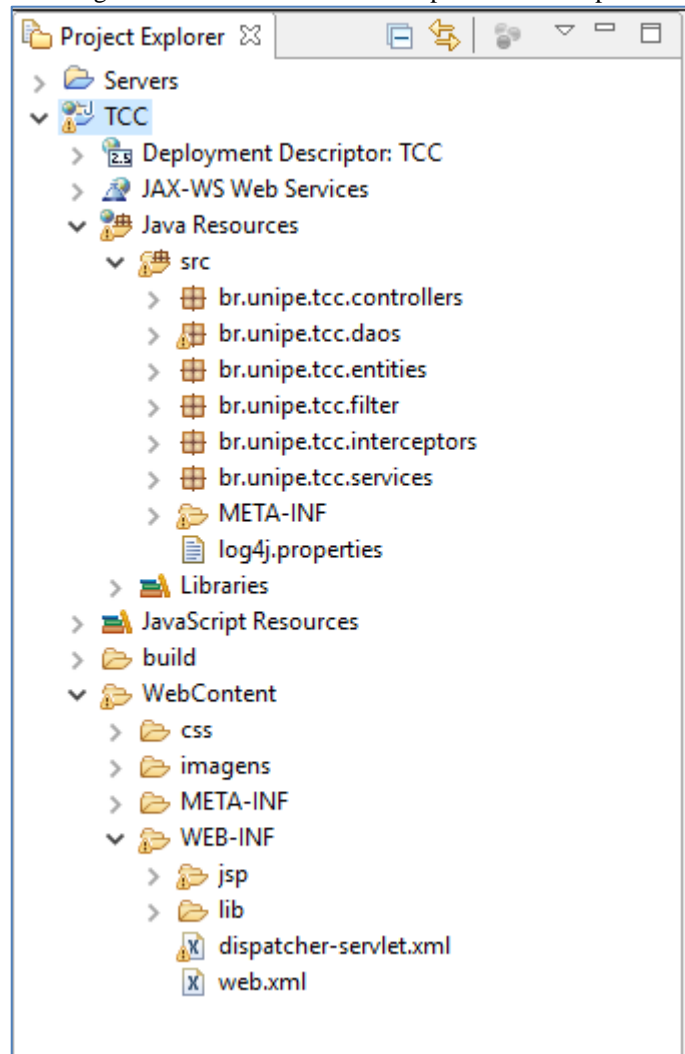
Utilizando a plataforma de desenvolvimento Eclipse, o sistema foi dividido da seguinte forma: Controllers, DAOs, Entities, Filter, Interceptors, Services, Web Content.

A Figura 16 mostra essa divisão na plataforma do Eclipse.

---

<sup>8</sup> Link de acesso ao Balsamiq Mockup .

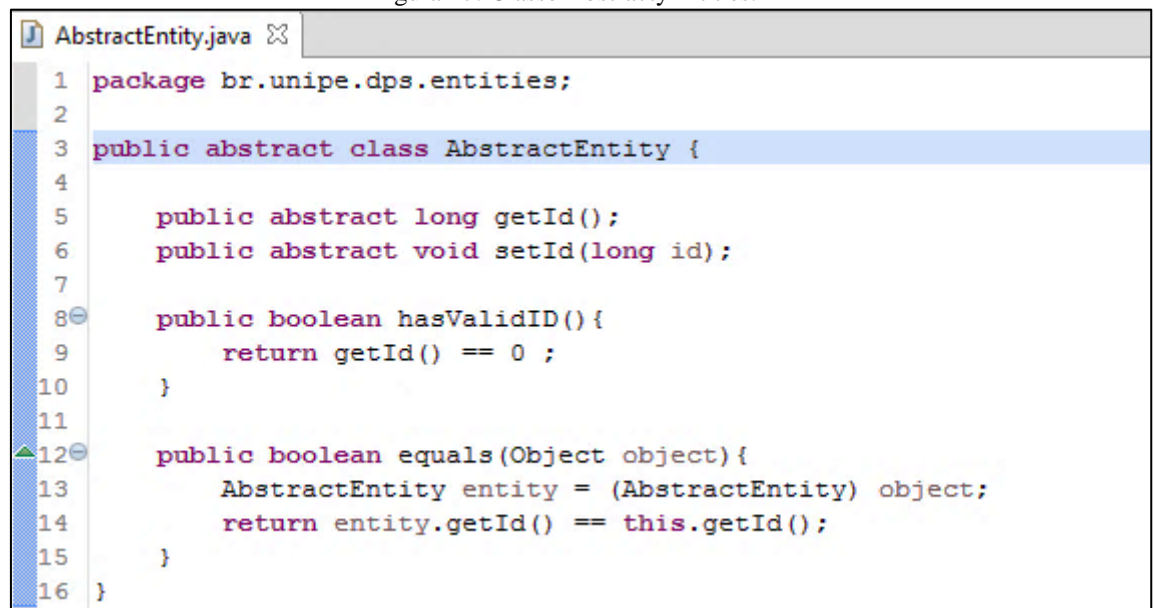
Figura 16: Divisão do sistema na plataforma Eclipse.



Fonte: Dados da pesquisa, 2016.

A Figura 17 mostra a classe AbstractyEntities do sistema proposto.

Figura 17: Classe AbstractyEntities.



Fonte: Dados da pesquisa, 2016.

Abstracty Entities: Nesta classe *abstracty* ela serve de modelo para as demais classes que a herdam, elas herdam os métodos de `getId()` e `setId()`;

A Figura 18 mostra a classe `AcompanhamentoAnimal` do sistema proposto.

Figura 18: Classe entidade `AcompanhamentoAnimal`.

```

19  @Entity
20  public class AcompanhamentoAnimal extends AbstractEntity{
21
22      @Id
23      @GeneratedValue(strategy=GenerationType.IDENTITY)
24      private long id;
25
26      private String nome;
27
28      @ManyToOne
29      private ProprietarioAnimal proprietarioAnimal;
30
31      @ManyToOne
32      private Alimentacao alimentacao;
33
34      @OneToOne(cascade = CascadeType.ALL)
35      private Animal animal;
36
37      @DateTimeFormat(pattern="dd-mm-yyyy")
38      @Temporal(TemporalType.DATE)
39      private Date data;
40
41      @OneToMany(mappedBy = "acompanhamentoAnimal")
42      private List<Medicacao> medicacao;
43
44      private String descricao;

```

Fonte: Dados da pesquisa, 2016.

**Acompanhamento Animal:** Nesta classe entidade possui um atributo `id*`. O atributo `proprietarioAnimal` está mapeando com o `@ManyToOne` (muitos para um), isso implica que um animal só tem um `proprietarioAnimal` (criador) e o `proprietarioAnimal` pode ter um ou mais animais, igual para o atributo `alimentação`. O atributo `data` esta mapeado como um `@Temporal` do tipo `data` e definido o formato da data com o `@DateTimeFormat`. Essa classe é responsável por mostra todo o acompanhamento do animal de estimação de alimentação, atividades diárias e cartão de vacina.

**Acompanhamento Veterinario:** Nesta classe entidade possui um atributo `id*`. Essa classe é responsável por mostrar e controlar os acompanhamentos com o veterinário, com a descrição da consulta.

Alimentação: Nesta classe entidade possui um atributo id\*, um atributo com o nome do alimento, porção (em quilogramas) e o horário da refeição do Pet.

A Figura 19 mostra a classe Animal do sistema proposto.

Figura 19: Classe entidade Animal.

```

19 @Entity
20 public class Animal extends AbstractEntity{
21
22     @Id
23     @GeneratedValue(strategy=GenerationType.IDENTITY)
24     private long id;
25
26     private String especie;
27     private String raca;
28     private boolean pedigree;
29     private String nome;
30     private float Peso;
31
32     @DateTimeFormat(pattern="dd-mm-yyyy")
33     @Temporal(TemporalType.DATE)
34     private Date dataNascimento;
35
36     @Enumerated(EnumType.STRING)
37     private String porteAnimal;
38
39     @ManyToOne
40     private ProprietarioAnimal proprietarioAnimal;
41
42     @ManyToOne
43     private Veterinario veterinario;
44
45     @OneToOne
46     private AcompanhamentoAnimal acompanhamentoAnimal;
47
48     @OneToOne(cascade = CascadeType.ALL)
49     private CartaoDeVacina cartaoDeVacina;
50

```

Fonte: Dados da pesquisa, 2016.

Animal: Nesta classe entidade possui um atributo id\*. O atributo porteAnimal está mapeado com @Enumerated.

A Figura 20 mostra os atributos da classe entidade Pessoa do sistema proposto.

Figura 20: Classe entidade Pessoa.

```

Pessoa.java ✕
14 @Entity
15 @Inheritance(strategy = InheritanceType.JOINED)
16 public class Pessoa extends AbstractEntity {
17
18     @Id
19     @GeneratedValue(strategy=GenerationType.IDENTITY)
20     protected long id;
21
22     // private Image foto;
23
24     private String nome;
25
26     @Column(unique = true)
27     private String CPF;
28
29     private String email;
30
31     private String telefone;
32
33     @Embedded
34     private Endereco endereco;
35
36     @OneToOne(cascade = CascadeType.ALL)
37     private Usuario usuario;

```

Fonte: Dados da pesquisa, 2016.

Pessoa: Nesta classe entidade possui um atributo id\*, atributo nome do tipo String, o CPF definido como unique (único),

A Figura 21 mostra a classe ProprietarioAnimal do sistema proposto.

Figura 21: Classe entidade ProprietarioAnimal.

```

1 package br.unipe.tcc.entities;
2
3 import java.util.List;
10
11 @Entity
12 @PrimaryKeyJoinColumn(name="id")
13 @Transactional(readonly = false)
14 public class ProprietarioAnimal extends Pessoa{
15
16     @OneToMany(mappedBy= "proprietarioAnimal")
17     private List<Animal> animais;
18
19     @OneToMany(mappedBy = "proprietarioAnimal")
20     private List<AcompanhamentoVeterinario> acompanhamentoVeterinario;
21
22     @OneToMany(mappedBy = "proprietarioAnimal")
23     private List<AcompanhamentoAnimal> acompanhamentoAnimal;
24
25     public ProprietarioAnimal() {
26         super();
27     }
28
29     public ProprietarioAnimal(long id) {
30         this.id = id;
31     }

```

Fonte: Dados da pesquisa, 2016.

Proprietário Animal: Classe entidade possui relacionamento de herança com a classe Pessoa, o atributo animais que é uma lista de Animal possui mapeamento de um para muitos com a classe ProprietarioAnimal, o atributo acompanhamentoVeterinario que é uma lista de AcompanhamentoVeterinario possui mapeamento de um para muitos com a classe ProprietarioAnimal, o atributo acompanhamentoAnimal que é uma lista de AcompanhamentoAnimal possui mapeamento de um para muitos com a classe ProprietarioAnimal.

A Figura 22 mostra a classe Usuario do sistema proposto.

Figura 22: Classe entidade Usuario.

```

10 @Entity
11 public class Usuario extends AbstractEntity{
12
13     @Id
14     @GeneratedValue(strategy=GenerationType.IDENTITY)
15     private long id;
16
17     @Column(nullable = false, unique = true)
18     private String login;
19
20     @NotNull(message="A senha não pode ser vazia")
21     private String senha;
22     public Usuario(Long id) {
23         super();
24         this.id = id;
25     }
26

```

Fonte: Dados da pesquisa, 2016.

Usuário: Nesta classe entidade possui um atributo id\*, atributos login do tipo unico e senha do tipo não nulo para o acesso a parte privada da sistema proposto;

\*id atributo utilizado para identificar cada tupla na classe utilizada, na tabela do banco de dados relacional.

### 3.6 PAGINAS DO SISTEMA

Com a implementação do prototipo realizado, será apresentado as paginas criadas. O modulo implementado foi o privado para controle dos animais de estimação.

No Apêndice G 1 Pagina inicial do modulo privado, contendo o menu na lateral esquerda e um botão para acesso ao acompanhamento do animal.

No Apêndice G 2 Pagina Listar Alimentos do modulo privado, contendo o menu na lateral esquerda e centralizado ao meio da pagina o lista com os alimentos já cadastrados.



No Apêndice G 3 Pagina Cadastrar Novo Alimento do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para o cadastro do alimento.

No Apêndice G 4 Pagina Listar Veterinario do modulo privado, contendo o menu na lateral esquerda e centralizado ao meio da pagina o lista com os veterinarios já cadastrados.

No Apêndice G 5 Pagina Cadastrar Novo Veterionario do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para o cadastro do do novo veterionario.

No Apêndice G 6 Pagina Listar Proprietario do Animal do modulo privado, contendo o menu na lateral esquerda e centralizado ao meio da pagina o lista com um único proprietario cadastrado.

No Apêndice G 7 Pagina Detalhar Proprietario Animal do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para a visualização dos dados do proprietario.

No Apêndice G 8 Pagina Listar Animais do modulo privado, contendo o menu na lateral esquerda e centralizado ao meio da pagina o lista com os animais de estimação já cadastrado.

No Apêndice G 9 Pagina Cadastrar Novo Animal do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para o cadastro do novo animais de estimação.

No Apêndice G 10 Pagina Detalhar Animal do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para a visualização dos dados do animal de estimação.

No Apêndice G 11 Pagina Listar Acompanhamento do Animal do modulo privado, contendo o menu na lateral esquerda e centralizado ao meio da pagina o lista com os acompanhamentos já cadastrados.

No Apêndice G 12 Pagina Cadastrar Novo Acompanhamento do Animal do modulo privado, contendo o menu na lateral esquerda e ao meio da pagina o formulario com os campos para o cadastro do novo acompanhamento do animal de estimação.

## 4 CONSIDERAÇÕES FINAIS

As considerações finais fazem um apanhado geral do que foi descrito e em seguida indicamos uma subdivisão em algumas seções para melhor organizar o texto: contribuições alcançadas, dificuldades encontradas e trabalhos futuros.

### 4.1 CONTRIBUIÇÕES ALCANÇADAS

As contribuições alcançadas desse trabalho, foram obtidas a partir dos objetivos específicos que foram propostos no Capítulo 1.

As principais contribuições deste trabalho, foram:

- Proporcionar aos criadores de animais de estimação um controle diário sobre os seus animais;
- Implementação de uma primeira versão do sistema proposto que realizar os cadastro dos criadores, animais e realizar o acompanhamento dos animais.

### 4.2 DIFICULDADES ENCONTRADAS

Falta de tempo para integra api de busca do Google ao sistema. Para que possa ser implementado a funcionalidade da busca pelo *pet ideal*.

### 4.2 TRABALHOS FUTUROS

Realizar a implementação das funcionalidades de realizar teste para encontro do melhor animal de estimação para o usuário, e a funcionalidade de relacionamento entre os animais de estimação para formação de casais.

Melhorar a interface das páginas web, tanto no visual quanto na acessibilidade. Melhorias para uma melhor utilização do sistema. Criar e aplicar um plano de teste.

Adquirir com uma empresa especializada um servidor web, para colocar o sistema online para o público.

## REFERÊNCIAS

ABINPET – Associação brasileira da indústria de produtos para animais de estimação. **IBGE- População de animais de estimação no Brasil – 2013 – em milhões.** Disponível em: <[http://www.agricultura.gov.br/arq\\_editor/file/camaras\\_tematicas/Insumos\\_agropecuarios/79RO/IBGE\\_PAEB.pdf](http://www.agricultura.gov.br/arq_editor/file/camaras_tematicas/Insumos_agropecuarios/79RO/IBGE_PAEB.pdf)>. Acessado em: 12 setembro de 2015.

Apache. Disponível em: < <http://tomcat.apache.org/>>. Acessado em: 10 de abril de 2016.

Augusto, C. **Análise e Projeto OO & UML 2.0.** Disponível em: <<http://www.dainf.ct.utfpr.edu.br/~tacla/UML/Apostila.pdf>>. Acessado em: 20 de março de 2016.

Caelum. **Persistindo os dados com o hibernate.** Disponível em: <<https://www.caelum.com.br/apostila-vraptor-hibernate/persistindo-os-dados-com-o-hibernate/#4-1-a-camada-de-persistencia-do-projeto>>. Acessado em: 08 de abril de 2016.

Devmedia. **Conceitos da linguagem Java.** Disponível em: <<http://www.devmedia.com.br/conceitos-da-linguagem-java/5341>>. Acessado em: 06 de abril de 2016.

Devmedia. **Introduzindo o servidor de aplicação Apache TomCat.** Disponível em: <<http://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939>>. Acessado em: 31 de março de 2016.

Devmedia. **JEE entendendo a plataforma.** Disponível em: <<http://www.devmedia.com.br/jee-entendendo-a-plataforma/30195>>. Acessado em: 30 de março de 2016.

Domina tudo. **Conceito de banco de dados relacional.** Disponível em: <<http://dominatudo.blogspot.com.br/2009/01/conceito-de-banco-de-dados-relacional.html>>. Acessado em: 08 de abril de 2016.

Eclipse. Disponível em: < <https://eclipse.org/>>. Acessado em: 08 de abril de 2016.

Eric, L. **Conceitos de bancos de dados relacionais.** Disponível em: <<http://ericlemes.com/2009/01/27/bd-conceitos/>>. Acessado em: 10 de abril de 2016.

Farinelli, F. **Conceitos básicos de programação orientada a objetos**. Disponível em: <[http://www.marcelohsantos.com.br/aulas/downloads/apostilas/Apostila\\_Java\\_B.pdf](http://www.marcelohsantos.com.br/aulas/downloads/apostilas/Apostila_Java_B.pdf)>.

Acessado em: 12 de abril de 2016.

Fernando, L. M. **Fundamentos de programação estruturada em c**. Disponível em: <[http://webserver2.tecgraf.puc-rio.br/ftp\\_pub/lfm/CIV2801ProgramC.pdf](http://webserver2.tecgraf.puc-rio.br/ftp_pub/lfm/CIV2801ProgramC.pdf)>. Acessado em: 11 de abril de 2016.

Fonseca R. G. **Os principais diagramas da uml: resumo rápido**. Disponível em: <<http://www.profissionaisti.com.br/2011/07/os-principais-diagramas-da-uml-resumo-rapido>>.

Acesso em: 08 de novembro de 2015.

FURLAN, J. D. **Modelagem de objetos através da UML**. São Paulo: Makron Books, 1998.

Caelum. **O que é Java EE?**. Disponível em: <<https://www.caelum.com.br/apostila-java-web/o-que-e-java-ee>>. Acesso em: 08 de novembro de 2015.

Gasparotto, M. H. **Os 4 pilares da programação orientada a objetos**. Disponível em: <<http://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>>.

Acesso em: 07 novembro de 2015.

Gustavo Luiz. **Astah Community**. Disponível em: <<https://luizgustavoss.wordpress.com/tag/astah-community/>>. Acessado em: 28 de março 2016.

Apache. Disponível em: <<http://tomcat.apache.org/index.html>>. Acessado em: 31 de março de 2016.

Ibm. **Introdução à plataforma Eclipse**. Disponível em: <<http://www.ibm.com/developerworks/br/library/os-eclipse-platform/>>. Acessado em: 28 de março de 2016.

Jacques. **História da uml**. Disponível em: <[http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/historia\\_uml/historia\\_uml.htm](http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/historia_uml/historia_uml.htm)>. Acesso em: 20 outubro de 2015.

Lopes, C. **Guia Prático - Construindo Aplicações Jee Com Frameworks**. Ed. Ciência Moderna, 2011.

Kioskea.net. **Banco de dados introdução.** Disponível em: <<http://static.ccm2.net/br.ccm.net/contents/pdf/bancos-de-dados-introducao-65-m2za19.pdf>>. Acessado em: 08 de abril de 2016.

MARTIN, J. **Análise e projeto orientados a objeto.** São Paulo: Makron Books, 1995.

Tacla, C. A. **Análise e projeto OO & UML.** Disponível em: <http://www.dainf.ct.utfpr.edu.br/~tacla/UML/Apostila.pdf>>. Acessado em: 15 de março de 2016.

MySQL. Disponível em: <<https://www.mysql.com/>>. Acessado em: 12 de abril de 2016.

PRESSMAN, R. **Engenharia de software.** Rio de Janeiro: Mc Graw-Hill, 2005.

Portal educação. **Tipos de metodologia de pesquisa.** Disponível em: <<https://www.portaleducacao.com.br/pedagogia/artigos/50264/metodologia-cientifica-tipos-de-pesquisa>>. Acesso em: 10 setembro de 2015.

RICARTE, I. L. M. **Introdução a orientação a objetos.** Disponível em: <[http://www.dca.fee.unicamp.br/cursos/POO\\_CPP/node3.html](http://www.dca.fee.unicamp.br/cursos/POO_CPP/node3.html)>. Acesso em: 07 novembro de 2015.

UAI. **Cada vez mais animais de estimação são tratados como gente e recebem cuidados especiais.** Disponível em: <[http://sites.uai.com.br/app/noticia/saudeplena/noticias/2014/10/07/noticia\\_saudeplena,150682/cada-vez-mais-animais-de-estimacao-sao-tratados-como-gente-e-recebem-c.shtml](http://sites.uai.com.br/app/noticia/saudeplena/noticias/2014/10/07/noticia_saudeplena,150682/cada-vez-mais-animais-de-estimacao-sao-tratados-como-gente-e-recebem-c.shtml)>. Acessado em: 31 de março de 2106.

Unicamp. **Banco de dados relacionais.** Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/javadb/bdrel.html>>. Acessado em: 07 de abril de 2016.

Wikibooks. **Banco de dados relacional.** Disponível em: <[https://pt.wikibooks.org/wiki/SQL/Banco\\_de\\_dados\\_Relacional](https://pt.wikibooks.org/wiki/SQL/Banco_de_dados_Relacional)>. Acessado em: 07 de abril de 2016.



## **APÊNDICE I**

## Apêndice A – Questionário aplicado para coleta de dados

**CENTRO UNIVERSITÁRIO DE JOÃO PESSOA**

Nº \_\_\_\_\_

Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

Sistema de busca baseado em personas para relacionar pessoas e animais de estimação

Caro respondente,

Este questionário é parte de um trabalho de conclusão de curso (TCC) e suas respostas tem grande importância. Tem como objetivo desenvolver um sistema para cadastrar e ter controle sobre animais de estimação(Pet), encontrar o Pet ideal para tipos de vida específicos e obter conhecimento sobre os mesmos. Só ira precisar de 15 minutos. Por favor, responda as questões abaixo:

**1ª parte:** Caracterização do respondente:

Nº:	Perguntas:	Respostas:
1	Qual a sua idade?	_____ Anos



2	Qual o seu sexo?	Masculino ( ) Feminino ( )
3	Qual é o seu nível de Escolaridade?	Ensino Fundamental I ( ) Ensino Fundamental II ( ) Ensino Médio ( ) Ensino Superior ( )
4	Qual a sua profissão?	_____

**2ª parte:** Caracterização do(s) animal de estimação:

Nº:	Perguntas:	Respostas:
1	Possui animal de estimação?	Sim ( ) Não ( )
2	Possui quantos animais de estimação?	1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( ) +5 ( )
3	Qual a espécie e raça de cada animal de estimação?	<p style="text-align: center;">(Espécie)                                             (Raça)</p> <p>1 _____   _____</p> <p>2 _____   _____</p> <p>3 _____   _____</p>

		4 _____   _____ 5 _____   _____ 6 _____   _____ 7 _____   _____ 8 _____   _____ 9 _____   _____ 10 _____   _____
<b>4</b>	De acordo com a resposta anterior, coloque na mesma sequência a data de desde quando possui o animal de estimação: Formato: dd/mm/aaaa.	1 ____ / ____ / ____ 2 ____ / ____ / ____ 3 ____ / ____ / ____ 4 ____ / ____ / ____ 5 ____ / ____ / ____ 6 ____ / ____ / ____ 7 ____ / ____ / ____ 8 ____ / ____ / ____ 9 ____ / ____ / ____ 10 ____ / ____ / ____
<b>5</b>	Possui algum tipo de acompanhamento para seu(s) animais de estimação?	Sim ( ) Não ( )
<b>3ª parte: Caracterização do sistema:</b>		
<b>Nº:</b>	<b>Perguntas:</b>	<b>Respostas:</b>

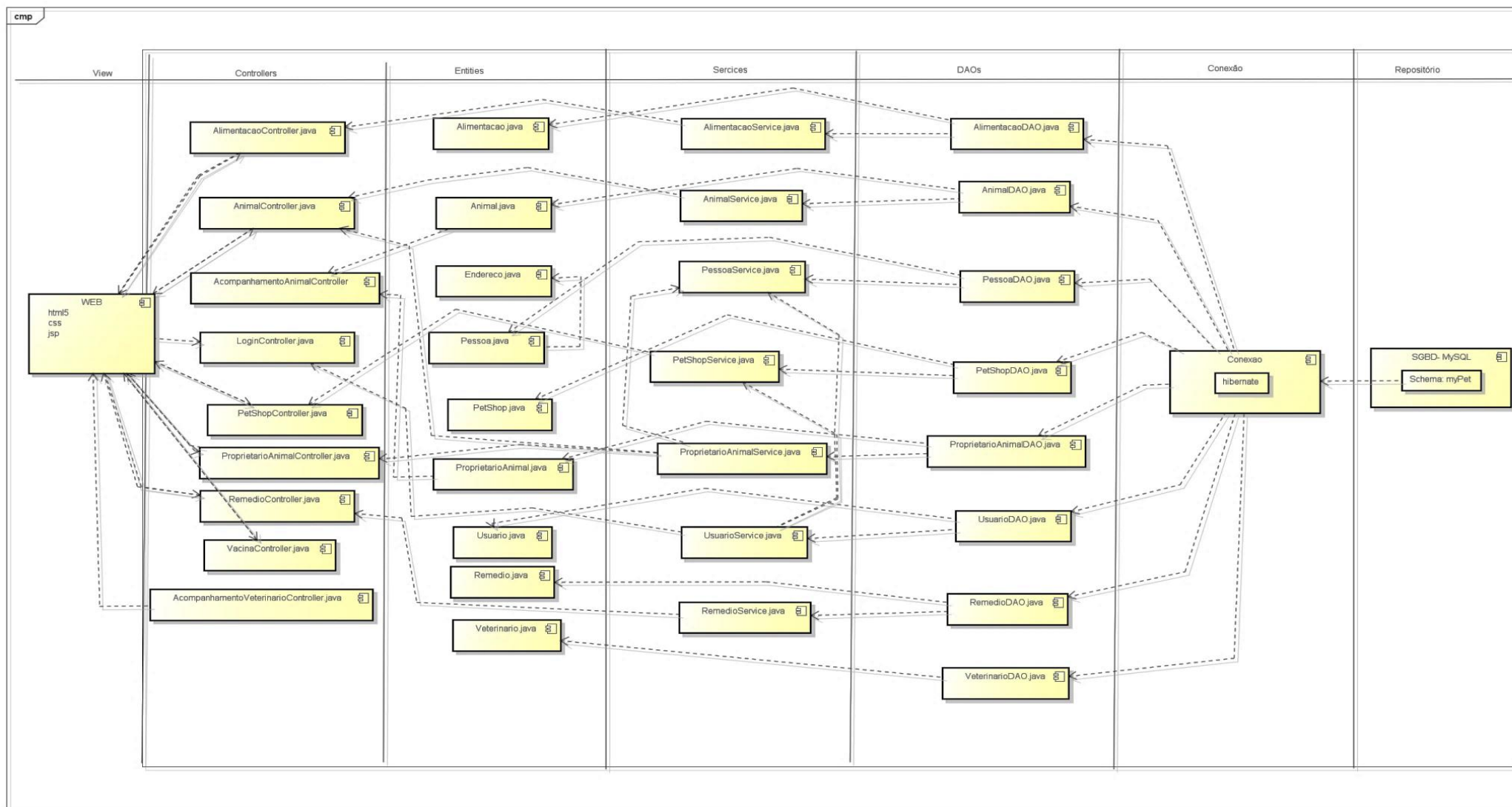
1	Tem dificuldade de encontrar informações e dicas para os cuidados com os animais:	Sim ( ) Não ( )
2	De acordo com a resposta anterior, como você avalia se encontrasse todas essas informações em um único site?	Ruim ( ) Regular ( ) Boa ( ) Ótima ( ) Excelente ( )
3	Qual seria sua opinião em encontrar um site que lhe ajude a escolher um animal de estimação ideal para você e seu tipo de vida ou necessidades?	Ruim ( ) Regular ( ) Boa ( ) Ótima ( ) Excelente ( )
4	Qual seria sua opinião em encontrar um site que possa se cadastrar e cadastrar seus animais de estimação?	Ruim ( ) Regular ( ) Boa ( ) Ótima ( ) Excelente ( )
5	Qual seria sua opinião em encontrar um site que você mesmo possa fazer o acompanhamento das atividades rotineiras de seus animais de estimação?	Ruim ( ) Regular ( ) Boa ( ) Ótima ( ) Excelente ( )

**4ª parte:** Caracterização da opinião do respondente do necessário para melhorar seus cuidados com seu(s) animal(is) de estimação:

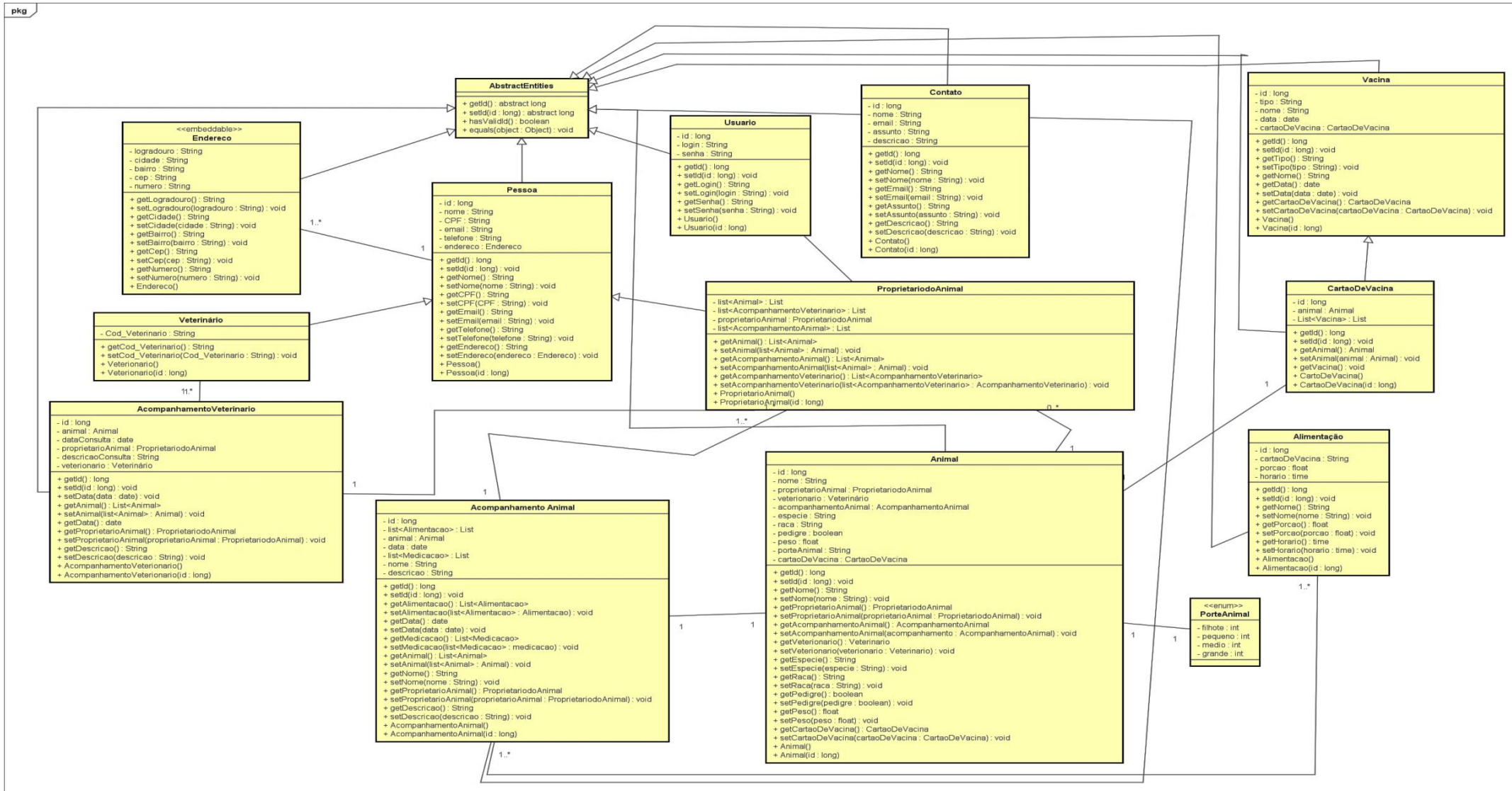
Nº:	Perguntas:	Respostas:
1	Quais características / atributos / atividades são necessário para poder ter um acompanhamento de seus animais de estimação?	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
2	Quais informações são necessário e interessante de possuir no site para seu auxílio com seu(s) animal(is) de estimação?	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
3	Alguma dica de como teria de ser o site para que ele possa virar o site ideal?	<hr/> <hr/>



Apêndice C – A figura representa o Diagrama de Componentes do sistema.



## Apêndice D – A figura representa o Diagrama de Classes do sistema.



### Apêndice Eº - Prototipações

**E 1** - O Quadro 01 representa os fluxos principais do caso de uso Manter Proprietário Animal.

Quadro 01: UC001 - Manter Proprietário Animal.

<b>Admin</b>	<b>Usuário</b>
Quando estiver autenticado no sistema selecionará no menu a opção Proprietário Animal. Na tela do Proprietário Animal, o Admin poderá fazer a inclusão, alteração, exclusão e pesquisa dos Proprietários dos Animais.	O Usuário na tela de login selecionará a opção Cadastre-se. Na tela de Cadastre-se, o Usuário poderá fazer a inclusão, alteração e exclusão somente de seus dados.

Fonte: Dados da pesquisa, 2016.

**E 2** - O Quadro 02 representa os fluxos principais do caso de uso Manter Animal.

Quadro 02: UC002 - Manter Animal.

<b>Admin</b>	<b>Usuário</b>
O Admin quando estiver autenticado no sistema selecionará no menu a opção Animal. Na tela de Animal, o Admin poderá fazer a inclusão, alteração, exclusão e pesquisa dos animais filtrando por seus Proprietários.	O Usuário quando estiver autenticado no sistema selecionará no menu a opção Animal. Na tela de Animal, o Usuário poderá fazer a inclusão, alteração e exclusão somente de seus animais.

Fonte: Dados da pesquisa, 2016.

**E 3**- O Quadro 03 representa os fluxos principais do caso de uso Manter Alimento.

Quadro 03: UC003 - Manter Medicamento.

<b>Admin</b>	<b>Usuário</b>
O Admin quando estiver autenticado no sistema selecionará no menu a opção Medicamentos. Na tela de Medicamentos, o Admin poderá fazer a inclusão, alteração, exclusão e pesquisa dos medicamentos.	O Usuário quando estiver autenticado no sistema selecionará no menu a opção Medicamentos. Na tela o Usuário terá a opção de selecionar um medicamento já cadastrado, quantidade e horários para medicação



	do animal de estimação.
--	-------------------------

Fonte: Dados da pesquisa, 2016.

**E 4** - O Quadro 04 representa os fluxos principais do caso de uso Manter Vacina.

Quadro 04: UC004 - Manter Vacina.

<b>Admin</b>	<b>Usuário</b>
<p>O Admin quando estiver autenticado no sistema selecionará no menu a opção Vacina.</p> <p>Na tela de Vacina, o operador poderá fazer a inclusão, alteração, exclusão e pesquisa de vacinas.</p>	<p>O Admin quando estiver autenticado no sistema selecionará no menu a opção Vacina. Na tela o Usuário terá a opção de selecionar uma vacina já cadastrada, e cadastrara com a data da vacina.</p>

Fonte: Dados da pesquisa, 2016.

**E 5** - O Quadro 05 representa os fluxos principais do caso de uso Acompanhamento Animal.

Quadro 05: UC005 – Acompanhamento Animal.

<b>Admin</b>	<b>Usuário</b>
	<p>O Usuário quando estiver autenticado no sistema selecionará no menu a opção Animal. Na tela de Animal, o Usuário poderá fazer a inclusão, alteração, exclusão e pesquisa do seu acompanhamento para cada animal seu cadastrado.</p>

Fonte: Dados da pesquisa, 2016.

**E 6** - O Quadro 06 representa os fluxos principais do caso de uso Adicionar Alimentação.

Quadro 06: UC006 – Adicionar Alimentação.

<b>Admin</b>	<b>Usuário</b>
<p>O Admin quando estiver autenticado no sistema selecionará no menu a opção Alimentação. Na tela de alimentação, o Admin poderá fazer a</p>	<p>O Usuário quando estiver autenticado no sistema selecionará no menu a opção de Alimentação. Na tela o Usuário terá a opção de selecionar um</p>

inclusão, alteração, exclusão e pesquisa dos alimentos.	alimento já cadastrado, quantidade e horários para à alimentação do animal de estimação.
---	--

Fonte: Dados da pesquisa, 2016.

**E 7 - O Quadro 07** representa os fluxos principais do caso de uso Acompanhamento Veterinário.

Quadro 07: UC007 – Acompanhamento Veterinário.

<b>Admin</b>	<b>Usuário</b>
	O Usuário quando estiver autenticado no sistema selecionará no menu a opção Animal. Na tela de Animal, o Usuário poderá fazer a inclusão, alteração, exclusão e pesquisa do acompanhamento do veterinário em cada consulta para cada animal seu cadastrado.

Fonte: Dados da pesquisa, 2016.

**E 8 - O Quadro 08** representa os fluxos principais do caso de uso Informações Sobre Doenças.

Quadro 08: UC008 – Informações Sobre Doenças.

<b>Admin</b>	<b>Usuário</b>
	O Usuário ira selecionar no menu a opção de Informações Sobre Doenças, terá um campo para selecionar o animal e terá informações sobre as doenças que o mesmo pode ter.

Fonte: Dados da pesquisa, 2016.

**E 9 - O Quadro 09** representa os fluxos principais do caso de uso Cartão de Vacina.

Quadro 09: UC009 – Cartão de Vacina.

<b>Admin</b>	<b>Usuário</b>
	O Usuário autenticado ao sistema selecionara a opção Animal no menu e

	depois Cartão de Vacina para poder visualizar as vacinas já tomadas e as próximas previstas.
--	--

Fonte: Dados da pesquisa, 2016.

**E 10** - O Quadro 10 representa os fluxos principais do caso de uso Realizar Teste.

Quadro 10: UC010 – Realizar Teste.

<b>Admin</b>	<b>Usuário</b>
	O Usuário selecionara a opção de Realizar Teste no menu, respondera um questionário e depois obter o resultado.

Fonte: Dados da pesquisa, 2016.

**E 11** - O Quadro 11 representa os fluxos principais do caso de uso Busca de Relacionamento.

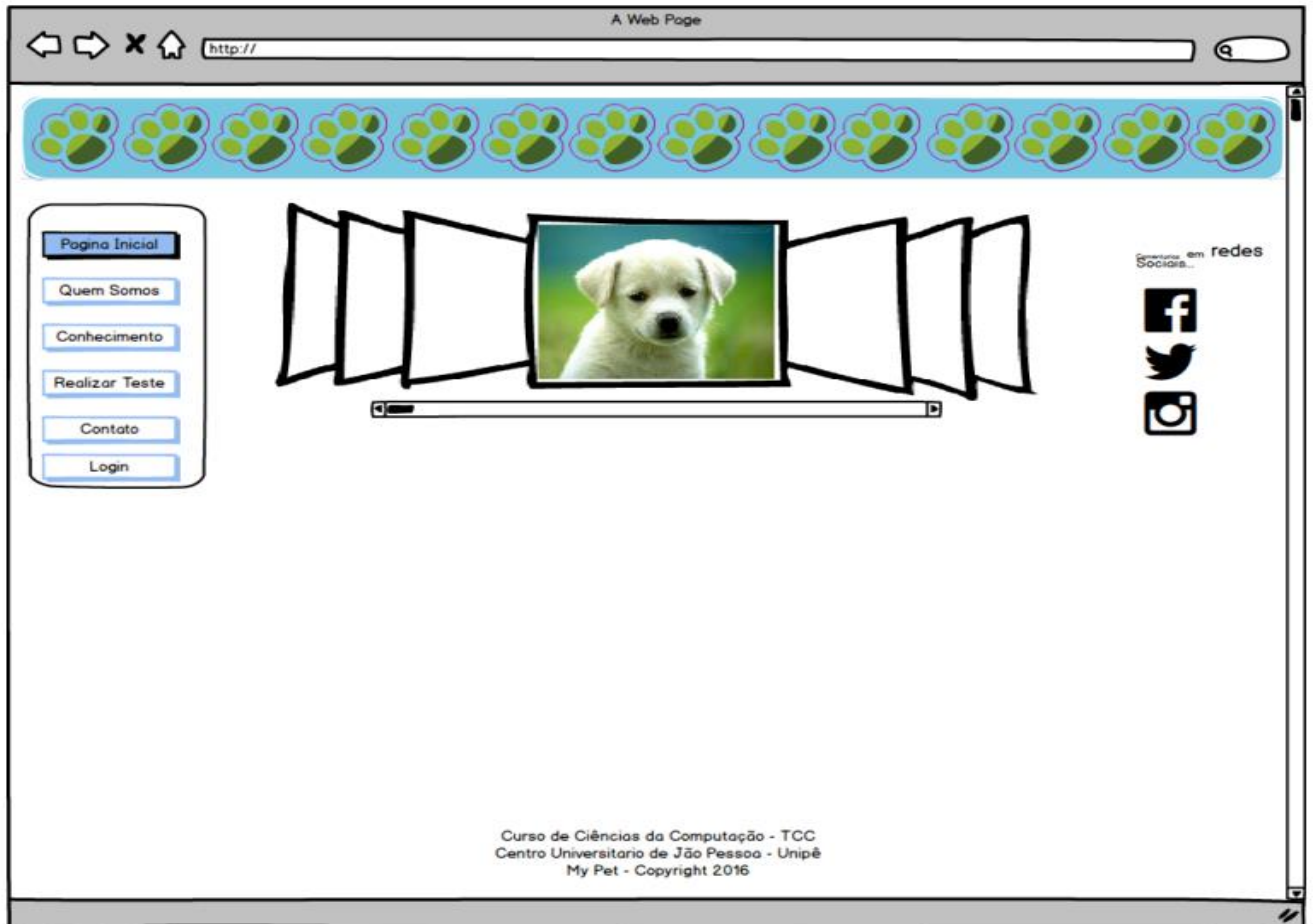
Quadro 11: UC011 – Buscar de Relacionamento.

<b>Admin</b>	<b>Usuário</b>
	O Usuário autenticado no sistema selecionara a opção de Buscar Relacionamento no menu e aparecera uma lista com os animais de mesma espécie que mais combine.

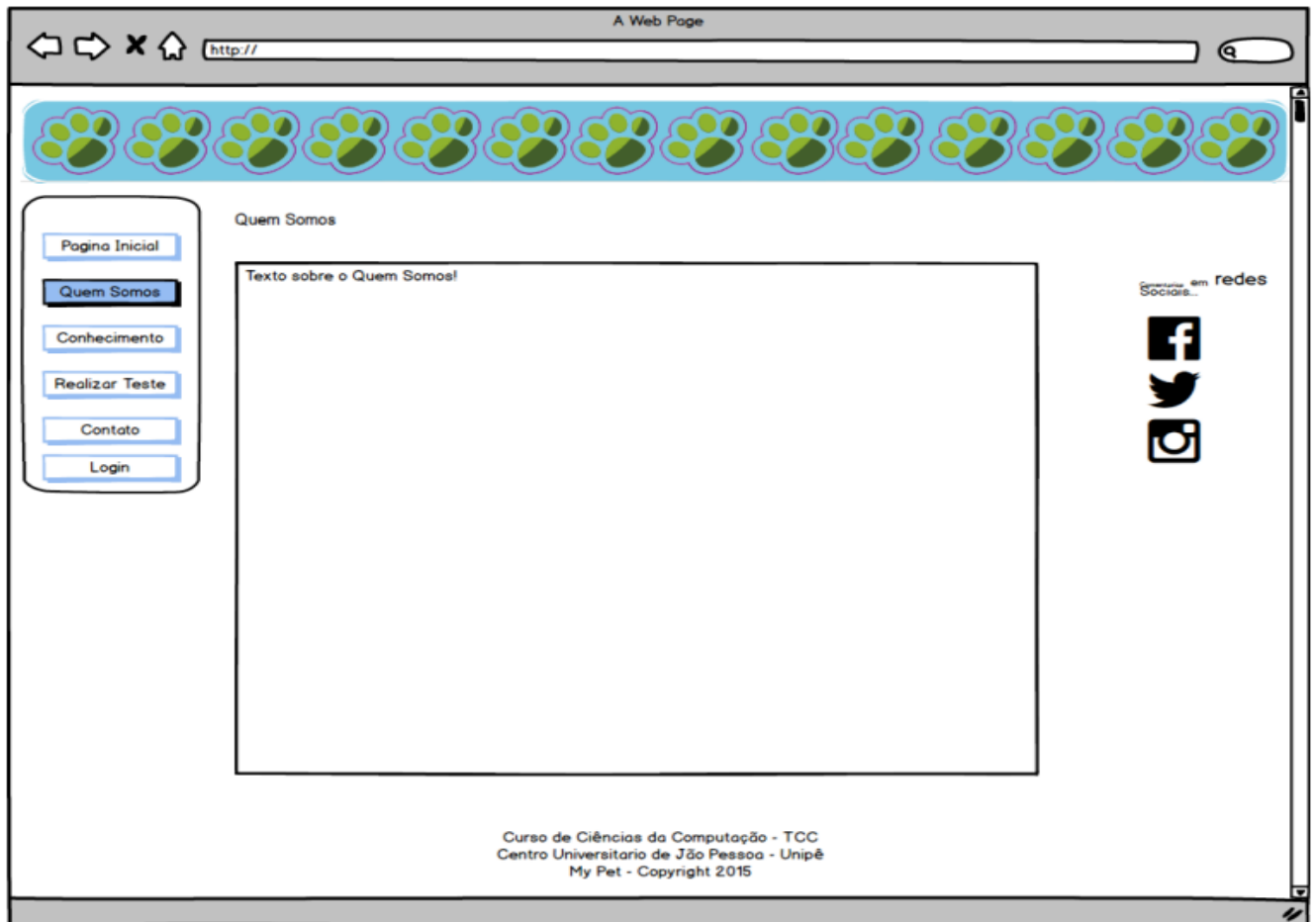
Fonte: Dados da pesquisa, 2016.

## Apêndice F – Prototipações de Tela

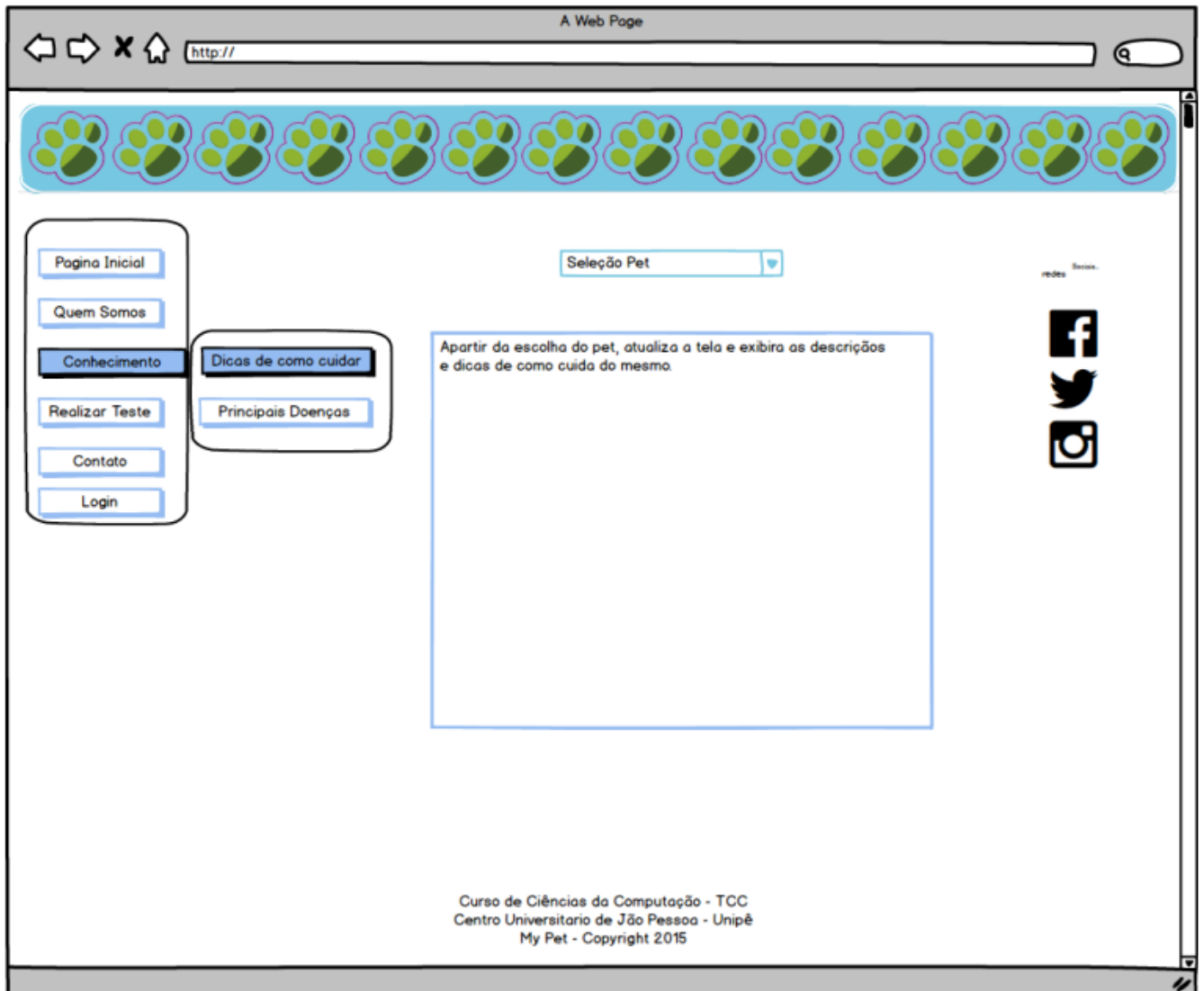
F 1 - Protótipo da tela inicial do sistema.



## F 2– Protótipo da tela Quem Somos.



## F 3 – Protótipo da tela Conhecimento.



## F 4 – Protótipo da tela Realizar Teste.

A Web Page

http://

Página Inicial

Quem Somos

Conhecimento

**Realizar Teste**

Contato

Login

Pergunta..

opção 1 ( ) opção 2 ( ) opção 3 ( ) opção 4 ( ) opção 5

Pergunta..

opção 1 ( ) opção 2 ( ) opção 3 ( ) opção 4 ( ) opção 5

Pergunta..

opção 1 ( ) opção 2 ( ) opção 3 ( ) opção 4 ( ) opção 5

Pergunta..

opção 1 ( ) opção 2 ( ) opção 3 ( ) opção 4 ( ) opção 5

Próxima Página >>

Login

Social<sub>em</sub> redes

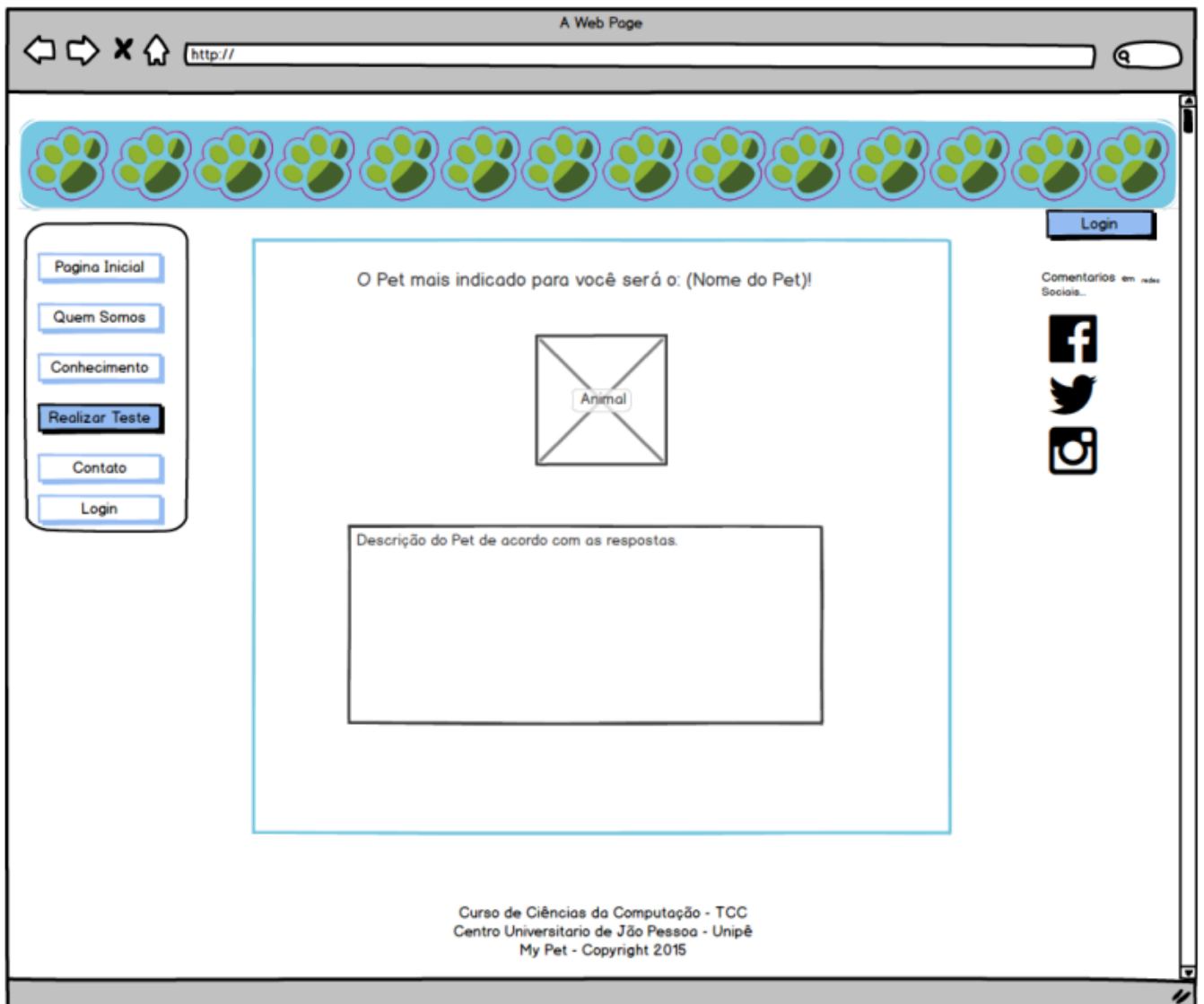
f

t

Q

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
My Pet - Copyright 2015

## F 5 – Protótipo da tela do resultado do teste.





## F 6– Protótipo da tela de Contato.

The image shows a wireframe of a web page titled "A Web Page" in the browser's title bar. The browser's address bar contains "http://". The page features a decorative header with a row of green paw prints on a light blue background. On the left side, there is a vertical menu with buttons for "Página Inicial", "Quem Somos", "Conhecimento", "Realizar Teste", "Contato" (highlighted), and "Login". The main content area contains a contact form with the following fields: "Nome:" (text input), "Email:" (text input), "Assunto:" (text input), and "Descrição:" (text area). Below the "Descrição:" field is an "Enviar" button. On the right side, there is a "Login" button, a "Comentários" section with "Social..." text, and icons for Facebook, Twitter, and Instagram. At the bottom center, the footer text reads: "Curso de Ciências da Computação - TCC", "Centro Universitário de João Pessoa - Unipê", and "My Pet - Copyright 2015".

Nome:

Email:

Assunto:

Descrição:

Enviar

Curso de Ciências da Computação - TCC  
Centro Universitário de João Pessoa - Unipê  
My Pet - Copyright 2015

## F 7 – Protótipo da tela de Login.

A Web Page

http://

Página Inicial

Quem Somos

Conhecimento

Realizar Teste

Contato

Login

Login

Senha

Se Cadastrar

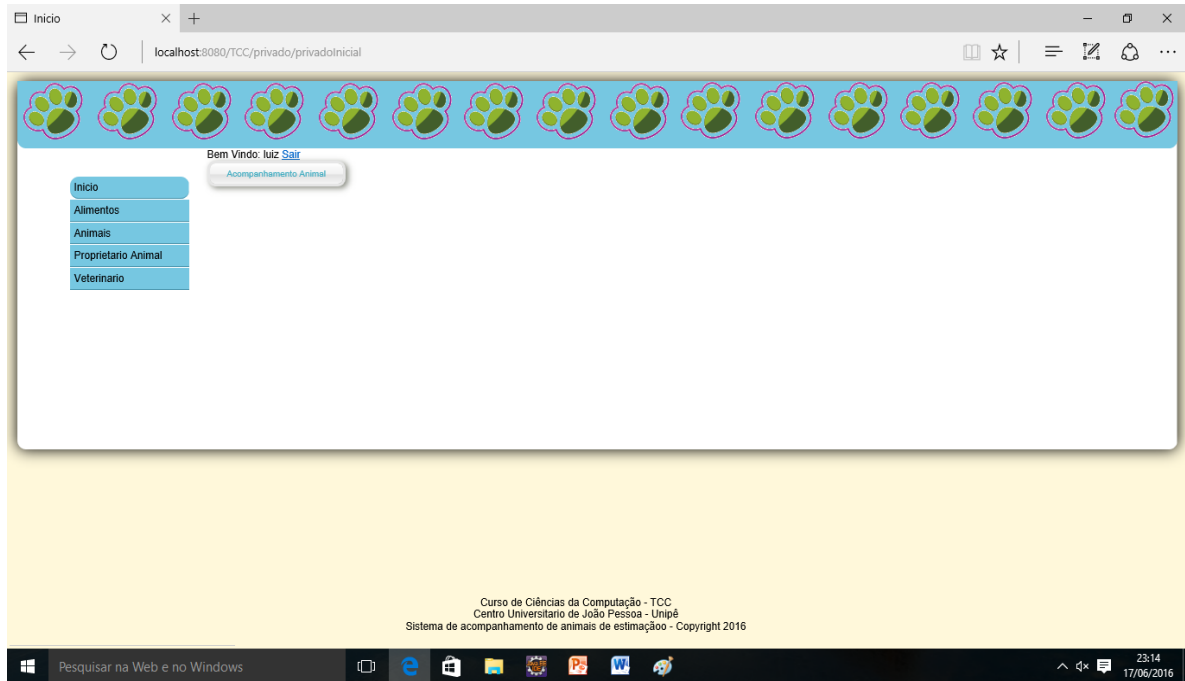
Fazer Login

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
My Pet - Copyright 2015

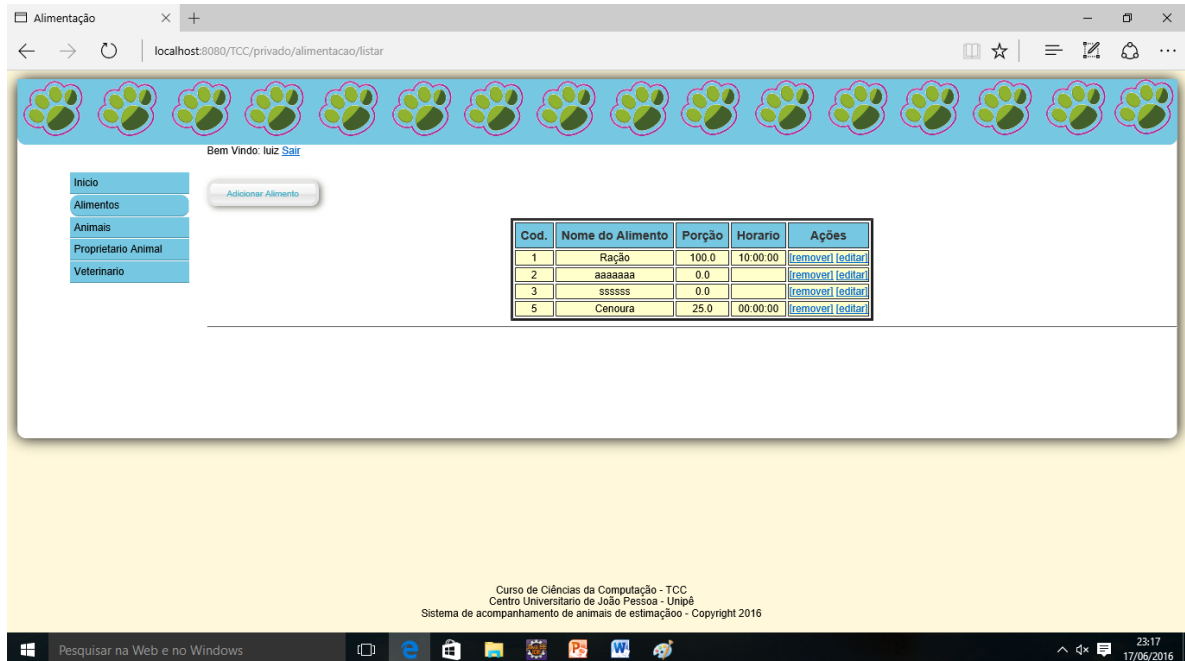
Detailed description: The image shows a wireframe of a web browser window. The browser's address bar contains 'http://'. Below the address bar is a decorative header with a light blue background and a repeating pattern of green paw prints. On the left side, there is a vertical menu with buttons for 'Página Inicial', 'Quem Somos', 'Conhecimento', 'Realizar Teste', 'Contato', and 'Login'. The main content area features a login form with two input fields labeled 'Login' and 'Senha'. Below these fields are two buttons: 'Se Cadastrar' and 'Fazer Login'. At the bottom center of the page, there is a footer containing the text: 'Curso de Ciências da Computação - TCC', 'Centro Universitario de João Pessoa - Unipê', and 'My Pet - Copyright 2015'.

## Apêndice G – Páginas do Prototipo

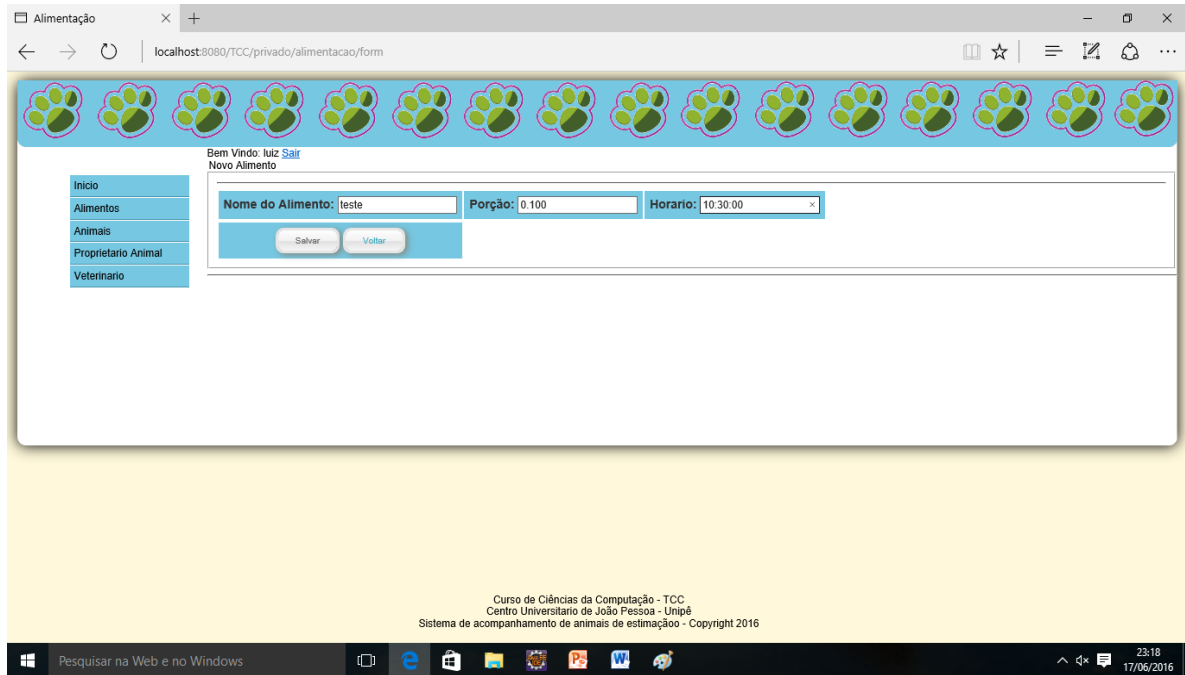
### Apêndice G 1 - Pagina inicial



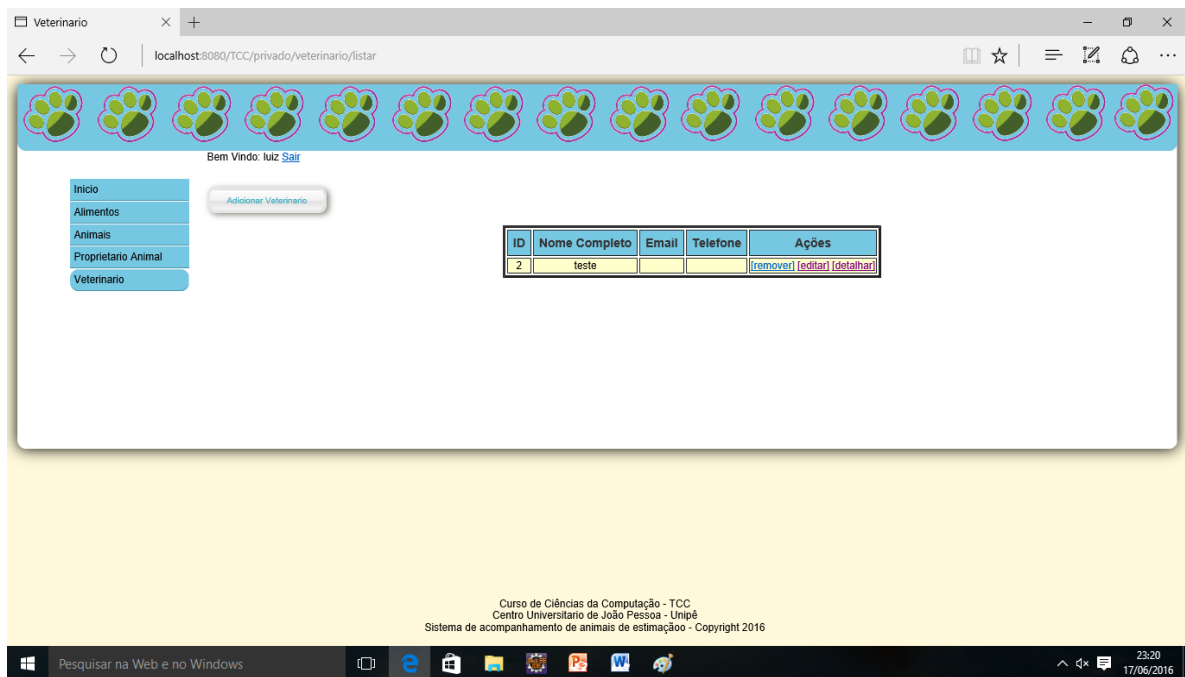
### Apêndice G 2 - Pagina Listar Alimentos



### Apêndice G 3 - Pagina Cadastrar Novo Alimento



### Apêndice G 4 - Pagina Listar Veterinario



## Apêndice G 5 - Pagina Cadastrar Novo Veterionairo

Bem Vindo: [luiz](#) [Sair](#)  
Novo Veterinario

Nome Completo:   
 Cpf:   
 CTPS:   
 Email:   
 Telefone:   
 Cep:   
 Longradouro:   
 Bairro:   
 Cidade:   
 Numero:

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 6 - Pagina Listar Proprietario do Animal

Bem Vindo: [luiz](#) [Sair](#)

Cod.	Nome Completo	Ações
1	Teste	<a href="#">[editar]</a> <a href="#">[detalhar]</a>

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 7 - Pagina Detalhar Proprietario Animal

Bem Vindo: [lutz](#) [Sair](#)

**Inicio**  
Alimentos  
Animais  
Proprietario Animal  
Veterinario

Nome Completo: **Teste**  
CPF: **12345678911**  
Email: **teste@teste.com**  
Telefone: **1111-1111**  
Cep: **111111**  
Longradouro:  
Bairro:  
Cidade:  
Numero:

Curso de Ciências da Computação - TCC  
Centro Universitário de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 8 - Pagina Listar Animais

Bem Vindo: [lutz](#) [Sair](#)

Nome do Proprietario	Cod. do Animal	Nome do Animal	Especie	Raça	Pedigree	Peso	Veterinario	Ações
Teste	8				false	0.0	teste	<a href="#">[remover]</a> <a href="#">[editar]</a> <a href="#">[detalhar]</a>
Teste	9	pingo	cachorro	shitsu	true	4.0	teste	<a href="#">[remover]</a> <a href="#">[editar]</a> <a href="#">[detalhar]</a>

Curso de Ciências da Computação - TCC  
Centro Universitário de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 9 - Pagina Cadastrar Novo Animal

Bem Vindo: [Iulz](#) [Sair](#)  
Nova Categoria

**Proprietario do Animal:**  **Nome do Animal:**  **Especie:**   
**Raça:**  **Pedigre:**  **Peso:**   
**Veterinario:**

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 10 - Pagina Detalhar Animal

Bem Vindo: [Iulz](#) [Sair](#)

**Nome Proprietario:** **Teste**  
**Nome Animal:** **pingo**  
**Especie:** **cachorro**  
**Raça:** **shitsu**  
**Pedigre:** **true**  
**Peso:** **4.0 KG**  
**Nome Veterinario:** **teste**

Curso de Ciências da Computação - TCC  
Centro Universitario de João Pessoa - Unipê  
Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 11 - Pagina Listar Acompanhamento do Animal

Bem Vindo: lutz [Sair](#)

[Início](#)  
[Alimentos](#)  
[Animais](#)  
[Proprietario Animal](#)  
[Veterinario](#)

Novo Acompanhamento Animal

id	Data do Acompanhamento	Ações
1	2012-12-12	<a href="#">remover</a> <a href="#">editar</a> <a href="#">detalhar</a>
2	2003-12-10	<a href="#">remover</a> <a href="#">editar</a> <a href="#">detalhar</a>
3	2012-12-02	<a href="#">remover</a> <a href="#">editar</a> <a href="#">detalhar</a>

Curso de Ciências da Computação - TCC  
 Centro Universitário de João Pessoa - Unipê  
 Sistema de acompanhamento de animais de estimação - Copyright 2016

## Apêndice G 12 - Pagina Cadastrar Novo Acompanhamento do Animal

Bem Vindo: lutz [Sair](#)

[Início](#)  
[Alimentos](#)  
[Animais](#)  
[Proprietario Animal](#)  
[Veterinario](#)

Novo Acompanhamento do Animal

Proprietario do Animal:

Nome do Animal:

Alimento 1:

Alimento 2:

Alimento 3:

Alimento 4:

Alimento 5:

Data:

Descrição:

Curso de Ciências da Computação - TCC  
 Centro Universitário de João Pessoa - Unipê  
 Sistema de acompanhamento de animais de estimação - Copyright 2016