

UFSC – UNIVERSIDADE FEDERAL DE SANTA CATARINA  
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

**CARLYLE TORRES B. DE MENEZES FILHO, JOSÉ DE LIMA DE SOUZA**

**REGISTRO GERAL DE ANIMAIS (RGA): UM SISTEMA PARA O  
REGISTRO E IDENTIFICAÇÃO DE ANIMAIS DE COMPANHIA**

**Araranguá, 05 de Junho de 2017**

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Filho, Carlyle Torres B. de Menezes; Souza, José de Lima  
de

Registro geral de animais (rga): um sistema para o  
registro e identificação de animais de companhia / Filho,  
Carlyle Torres B. de Menezes; Souza, José de Lima de ;  
orientador, Vinicius Faria Culmant Ramos, 2017.

61 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Tecnologias da Informação e Comunicação,  
Araranguá, 2017.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2.  
Tecnologias da Informação e Comunicação. 3. Sistema de  
Registro e Identificação. 4. Animais de Companhia. 5.  
Zoonoses. I. Ramos, Vinicius Faria Culmant . II.  
Universidade Federal de Santa Catarina. Graduação em  
Tecnologias da Informação e Comunicação. III. Título.

CARLYLE TORRES B. DE MENEZES FILHO, JOSÉ DE LIMA DE SOUZA

REGISTRO GERAL DE ANIMAIS (RGA): UM SISTEMA PARA O REGISTRO E  
IDENTIFICAÇÃO DE ANIMAIS DE COMPANHIA

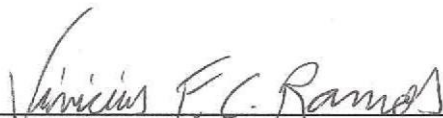
Trabalho de Curso submetido à  
Universidade Federal de Santa Catarina  
como parte dos requisitos necessários  
para a obtenção do Grau de Bacharel em  
Tecnologias da Informação e  
Comunicação. Sob a orientação do  
Professor Vinicius.

**Araranguá, 2017**

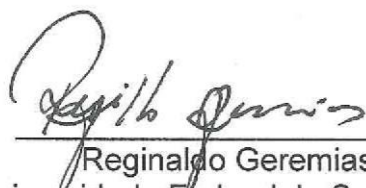
**CARLYLE TORRES B. DE MENEZES FILHO, JOSÉ DE LIMA DE SOUZA**

**Título: REGISTRO GERAL DE ANIMAIS (RGA): UM SISTEMA PARA O  
REGISTRO E IDENTIFICAÇÃO DE ANIMAIS DE COMPANHIA**

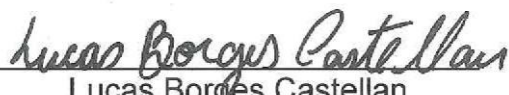
Trabalho de Curso submetido à  
Universidade Federal de Santa Catarina,  
como parte dos requisitos necessários  
para a obtenção do Grau de Bacharel em  
Tecnologias da Informação e  
Comunicação. Sob a orientação do Prof.  
Vinicius Faria Culmant Ramos.



Prof. Vinicius Faria Culmant Ramos, Dr.  
Universidade Federal de Santa Catarina



Reginaldo Geremias, Dr.  
Universidade Federal de Santa Catarina



Lucas Borges Castellán  
Universidade Federal de Santa Catarina

**Araranguá, 05 de Junho de 2017.**

## **AGRADECIMENTOS**

*Agradecemos a todos os que nos ajudaram  
na elaboração deste trabalho e a todos que  
direta ou indiretamente fizeram parte da nossa  
formação.*

*“A menos que modifiquemos a nossa  
maneira de pensar, não seremos capazes  
de resolver os problemas causados pela  
forma como nos acostumamos a ver o  
mundo”.*

***Albert Einstein***

## RESUMO

Diversos municípios brasileiros enfrentam problemas relacionados à falta de políticas de controle das populações de animais de companhia dentre eles o abandono, a superpopulação de animais não domiciliados, os maus tratos e a transmissão de zoonoses. As consequências deste quadro, nos âmbitos ambientais e de saúde pública, levam a necessidade da identificação e registro dos animais de companhia em vista da elaboração e aplicações de ações de controle populacional, educação para guarda responsável e atribuição de responsabilidades no tocante ao direito da posse de animais de companhia. Este trabalho tem como objetivo o desenvolvimento de uma ferramenta para o registro e identificação da população de cães, extensível a população de gatos, utilizando tecnologias voltadas ao desenvolvimento web, tais como os frameworks JHipster, Spring, Angular e Bootstrap, para criar uma base de dados que possibilite aos gestores locais armazenarem informações sobre a população local de animais de companhia e através delas traçarem estratégias de controle populacional e controle de zoonoses, vinculando cada animal a um responsável e armazenando dados relativos a capacidade reprodutiva e histórico de vacinas.

Palavras-chave: Sistema de Registro e Identificação. Animais de Companhia. Zoonoses. Guarda responsável. Bem-estar animal.

## **ABSTRACT**

Several Brazilian municipalities face problems related to the lack of policies to control the populations of companion animals, such as abandonment, overpopulation of non-domiciled animals, maltreatment and the transmission of zoonoses. The consequences of this situation, in the environmental and public health areas, require the identification and registration of domiciled and street animals, bearing in mind the elaboration and application of population control actions, education for responsible custody and attribution of responsibilities regarding the right of pet ownership. This work aims to develop a tool to register and identify the population of dogs, extendable to the cat population, using technologies aimed at web development, such as the JHipster, Spring, Angular and Bootstrap frameworks, to create a database that enables local managers to store information on the local population of companion animals and through them draw strategies of population control and control of zoonoses, linking each animal to a responsible owner and storing data on the reproductive capacity and history of vaccines. Keywords: Registration and Identification System. Animal Companions. Zoonoses. Responsible Guard. Animal welfare.



## LISTA DE ILUSTRAÇÕES

Figura 1 - atendimentos e profilaxia da raiva humana no Brasil, 2000 a 2009.....	24
Figura 2 - Comparação entre os métodos de desenvolvimento baseado em planos e desenvolvimento ágil.....	32
Figura 3 - Área inicial da aplicação em modo usuário, com os links para os formulários de cadastro de proprietários, de animais e de vacinas.....	35
Figura 4 - Relação dos animais cadastrados, com as funcionalidades de alteração, edição, exclusão e inclusão de novo registro.....	36
Figura 5 - Módulo de monitoramento do estado do Sistema em modo administrador com informações relativas ao banco de dados e espaço em disco.....	37
Figura 6 - Ecosistema Spring .....	46
Figura 7 - Arquétipo do projeto.....	47
Figura 8 - Descrição entidades JSON .....	48
Figura 9 - Descrição dependências Bower.....	49
Figura 10 - Arquivo configuração Gradle.....	49
Figura 11 - Diretório src.....	49
Figura 12 - Mapeamento classe de entidades Java .....	50
Figura 13 - Diretório resources.....	51
Figura 14 - Datasource.....	51
Figura 15 - Diretório webapp.....	52

Figura 16 - Diretório i18n.....	53
Figura 17 - Arquivo de tradução .....	53

## **LISTA DE ABREVIATURAS E SIGLAS**

Ajax - Asynchronous Javascript and XML

App - Application

ASL - Apache Software License

ASP - Active Server Pages

CRUD - Create, Read, Update e Delete

CSS - Cascade Style Sheets

FAMA - Fundação Ambiental

HTML - Hipertext markup Language

I/O - Input/Output

ICAM - International Companion Animal Management Coalition

J2EE - Java 2 Enterprise Edition

JDL - JHipster Domain Language

JVM - Java Virtual Machine

MVC - Model View Controler

NEWT - New Exciting Web Technologies

OIE - World Organization for Animal Health

OMS - Organização Mundial da Saúde

ORM - Object Relational Mapper

PHP - Hypertext Preprocessor

RAD - Rapid Application Development

RGA - Registro Geral de Animais

SMS - Secretaria Municipal de Saúde

SQL - Structure Query Language

SRC - Source

UIPA - União Internacional Protetora dos Animais

UML - Unified Modeling Language

UNESCO - United Nations Educational, Scientific and Cultural Organization

W3 - World Wide Web

XML - Extensible Markup Language

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>15</b>
1.1 <i>Objetivos</i>	17
1.1.1 <i>Objetivo Geral</i>	17
1.1.2 <i>Objetivos Específicos</i>	17
1.2 <i>Metodologia</i>	18
<b>2 REFERENCIAL BIBLIOGRÁFICO</b>	<b>19</b>
2.1 <i>Registro e identificação da população de animais de companhia</i>	19
2.2 <i>Controle de zoonoses</i>	22
2.3 <i>Guarda responsável e bem-estar animal</i>	25
2.4 <i>Legislações sobre animais de companhia no Brasil</i>	26
2.5 <i>Controle da população de animais de companhia no exterior</i>	29
<b>3 DESENVOLVIMENTO</b>	<b>31</b>
3.1 <i>Processo de Software</i>	31
3.2 <i>Métodos Ágeis</i>	32
3.2.1 <i>RAD (Rapid Application Development)</i>	32
3.3 <i>Requisitos funcionais e não funcionais</i>	33
3.3.1 <i>Requisitos funcionais</i>	33
3.3.2 <i>Requisitos Não Funcionais</i>	34
3.4 <i>Interface de usuário do Registro Geral de Animais</i>	35
3.5 <i>Sistema de gerenciamento de banco de dados</i>	37
3.6 <i>Tecnologias utilizadas</i>	38
3.6.1 <i>AngularJS</i>	38
3.6.2 <i>Bootstrap</i>	39
3.6.3 <i>Bower</i>	39
3.6.4 <i>Gradle</i>	40
3.6.5 <i>Gulp</i>	40
3.6.6 <i>Hibernate</i>	41
3.6.7 <i>HTML</i>	41
3.6.8 <i>Java</i>	42
3.6.9 <i>JHipster</i>	43
3.6.10 <i>Liquibase</i>	43
3.6.11 <i>NodeJS</i>	44
3.6.12 <i>MySQL</i>	45
3.6.13 <i>Spring Framework</i>	45
3.7 <i>Estrutura do Projeto</i>	46

3.7.1	Arquétipo .....	46
3.7.2	Diretório base .....	47
3.7.3	Server-side .....	49
3.7.4	Front-side .....	52
3.8	<i>Licença de software</i> .....	54
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>55</b>
	<i>ANEXO</i> .....	61
	<i>Anexo 1 – Modelo relacional do banco de dados</i> .....	61

## 1. INTRODUÇÃO

A domesticação de animais remonta há cerca de dez mil anos, durante o período neolítico, quando o homem substituiu o nomadismo pelo estabelecimento de aldeias e a prática da agricultura. Os lobos estão entre os primeiros animais a serem domesticados, tendo estabelecido relações de tolerância e benefício mútuos com os habitantes dos assentamentos. Os descendentes destes animais, os cães, vêm acompanhando os homens deste então, e suas atividades variam de cães de companhia a cães treinados para realizar tarefas específicas, como cães-guia no auxílio de deficientes visuais, e cães empregados nas forças militares.

Uma variedade de aspectos culturais modifica a influência dos cães no ambiente onde estão inseridos e na saúde humana, bem como as responsabilidades relacionadas à posse de animais de companhia apresentam diversos níveis de tolerância de acordo com o contexto social local. Algumas culturas, sobretudo em países industrializados, desenvolveram políticas para minimizar os impactos indesejáveis que os animais domésticos podem causar a uma comunidade, dentre eles danos a propriedades de terceiros, desequilíbrio populacional da fauna local, superpopulação de cães não domiciliados e a transmissão de doenças, dentre elas a raiva.

Diversas ações são empregadas, de acordo com as necessidades locais, para atenuar os aspectos negativos que uma população não monitorada de animais de companhia pode causar. Dentre elas estão campanhas educativas para conscientização sobre a guarda responsável, subsidio de esterilização e vacinação e programas de adoção. Para que estas ações tenham sucesso é indispensável à adoção de um mecanismo para a identificação de cada animal individualmente. Desta forma é possível armazenar e recuperar informações relativas ao histórico de vacinação, capacidade reprodutiva, endereço e demais dados pessoais do

responsável, dentre outros que identifiquem o animal, como nome, idade, sexo e capacidade reprodutiva.

Com o vínculo do animal a um responsável, é possível atribuir ao cidadão um registro oficial de posse ou guarda, o que lhe confere direitos e deveres relativos aos cuidados com o animal. Desta forma, espera-se que ao se adquirir um animal de companhia, o guardião se comprometa em atender as necessidades físicas, psicológicas e ambientais de seu animal, assim como prevenir os riscos que seu animal possa causar à comunidade ou ao ambiente.

Além de causar sérios transtornos ambientais, a superpopulação de cães não supervisionados propicia a proliferação de zoonoses, doenças transmitidas dos animais para os seres humanos. Segundo a OMS, mais de cem doenças zoonóticas são transmitidas dos cães para os humanos. Dentre elas estão doença de Chagas, leishmaniose e raiva. (OMS, 1990) No Brasil, segundo o Ministério da Saúde, entre 1990 e 2016 foram registrados 408 casos de raiva humana tendo como espécie agressora cães e gatos. (BRASIL, 2014)

Um outro fator desencadeado pela falta de ações que reforcem as políticas de manejo de animais de companhia orientada sob os princípios da guarda responsável é a compra impulsiva de animais. Para Santana, a aquisição de animais, estimulada por comerciantes que os expõem como mercadorias, muitas vezes, não desperta o vínculo afetivo que deve nortear a relação entre homem e animal, por terem sido originadas pelo impulso do consumismo, fazendo com que as pessoas acabem descartando seus animais de estimação, por se tornarem desinteressantes após a empolgação inicial (SANTANA et al., 2004).

O abandono destes animais adquiridos de forma impulsiva resulta em sérios problemas ambientais, uma vez que a taxa de reprodução de cães é exponencial, pois eles se tornam sexualmente maduros aos dez meses de idade, e uma fêmea adulta pode gerar a partir de quatro filhotes por ano, proporcionando as condições para a população triplicar a cada ano. (OMS, 1990) Para Moutinho, na maior parte dos municípios brasileiros há superpopulação de cães não domiciliados, os quais trazem problemas associados à ordem urbana, ao meio ambiente e à saúde coletiva, além de sofrerem maus-tratos. Nesse contexto, vislumbra-se a necessidade do desenvolvimento de ações de controle populacional desses animais. (MOUTINHO, 2015)



Com a intenção de servir como suporte as ações de controle populacional de animais de companhia, este trabalho se propõe em apresentar uma ferramenta para o registro e a identificação da população animal local, dos proprietários e do histórico de vacinas de cada animal.

Para esta tarefa, foram utilizadas tecnologias voltadas ao desenvolvimento web, tais como os frameworks JHipster, Spring, Angular e Bootstrap, que propiciam uma rápida curva de implementação, maior controle e organização do projeto, padronização do código, e maior facilidade de manutenção devida a sua estrutura desacoplada.

O intuito da criação de uma aplicação web foi possibilitar uma maior disponibilidade de acesso, através da internet. Graças ao seu design responsivo, é possível o acesso em dispositivos com diferentes resoluções de tela. Também proporciona um menor consumo de recursos de hardware, por ter uma infraestrutura centralizada e escalável. Este modelo de aplicação nos concede também uma maior flexibilidade no gerenciamento de usuários.

## **1.1 Objetivos**

### **1.1.1 Objetivo Geral**

Desenvolver uma aplicação que permita o registro e a recuperação de dados sobre a população de animais de companhia de uma localidade, histórico de vacinas, relacionadas a cada animal, e cadastro dos proprietários.

### **1.1.2 Objetivos Específicos**

- Realizar levantamento de requisitos em conjunto com a Secretaria Municipal de Saúde de Araranguá (SMS) e a Fundação Ambiental de Araranguá (FAMA).

- Realizar um levantamento bibliográfico sobre o registro e identificação de animais de companhia, controle populacional de cães e gatos, controle de zoonoses, guarda responsável e bem-estar animal, legislação brasileira e programas de controle animal no exterior.

- Desenvolvimento de modelo de banco de dados que proporcione as relações necessárias entre as entidades.

- Aplicação de interface web padrão utilizando o framework front-end Bootstrap.

## **1.2 Metodologia**

As ciências da computação adquirem um caráter empírico quando são estudados fenômenos ocorridos no mundo real. Devem-se respaldar as descobertas em observações para fundamentar os assuntos abordados. (WAZLAWICK, 2014). Para o desenvolvimento do sistema de Registro Geral de Animais, foram realizados encontros periódicos com profissionais da Secretaria Municipal de Saúde de Araranguá e da Fundação Ambiental de Araranguá, que se encontram em processo de implementação de ações para a identificação de animais por meio de microchips e realização de mutirões de castração, em vista de controlar a superpopulação de cães e gatos não domiciliados. Através das informações obtidas, foi realizada a elaboração do documento de requisitos funcionais e não funcionais do software, para que a aplicação apresentasse as interações necessárias aos usuários e administradores do sistema, armazenasse de maneira confiável os dados e apresentasse as relações pertinentes entre os formulários de registro de proprietários, animais e vacinas.

Após o levantamento dos requisitos, foi realizada uma revisão bibliográfica pertinente ao registro e identificação de animais de companhia, no Brasil e no exterior, ao impacto das zoonoses na sociedade, aos conceitos de guarda responsável e bem-estar animal, e a legislação brasileira referente à posse e guarda de animais. De posse das funcionalidades necessárias para o correto funcionamento do sistema, a etapa de desenvolvimento foi realizada utilizando frameworks.

## **2 REFERENCIAL BIBLIOGRÁFICO**

### **2.1 Registro e identificação da população de animais de companhia**

A relação entre animais e seres humanos demanda que sejam desenvolvidas práticas e atitudes conscientes para que sejam mantidos os equilíbrios biológico, social e ambiental entre as diversas espécies. Os animais de estimação (cães e gatos) apresentam os maiores números em termos de indivíduos introduzidos no contexto das relações humanas tornando ações relativas ao controle destas populações fatores importantes em termos de saúde pública e gestão ambiental. (VIEIRA et all. 2006)

As ideias sobre a posse e guarda de animais domésticos e as responsabilidades inerentes a esta posse variam de acordo com diferentes culturas. Nas nações industrializadas a lei e consenso público dão o direito das pessoas a possuir animais e mantê-los, mas também preveem a obrigação de cuidar deles. Estes cuidados incluem, mas não são limitados a abrigo e fonte de comida e água. Segundo a Organização Mundial da Saúde, o dono do animal é responsável por garantir que não causem dano a propriedades públicas e de terceiros e que não apresentem nenhuma condição não sanitária, perigosa ou ofensiva. Em algumas culturas, as obrigações dos donos de animais são menos restritivas, mas a posse e as responsabilidades devem ser reguladas por regras com maior ou menor grau de exigências. A influência dos animais no ambiente e na saúde humana depende das práticas com relação aos mesmos, no reconhecimento e controle de zoonoses, tratamento e destinação de animais doentes e controle da taxa de natalidade a partir da esterilização. Em quase todas as circunstâncias informação e educação são tão importantes como regulamentação legal. (OMS, 1990)

Para a ICAM, o comportamento e as atitudes humanas são as forças mais poderosas por trás da dinâmica da população canina. O encorajamento das interações homem-animal de maneira responsável e compensatória leva a melhoria do bem-estar animal e a uma redução de muitas das origens de cães de rua. Diversas questões como crenças e hábitos locais devem determinar como as intervenções referentes a programas de controle devem ser aplicadas. (ICAM, 2007)

Historicamente, a prática mais comum para tratar do controle populacional de animais não domiciliados, em áreas urbanas é o recolhimento, e posterior aplicação de medidas que variam de acordo com a administração local, tendo como exemplos ações de adoção e eutanásia. A Secretaria Estadual de Saúde de São Paulo, através de Programa de Controle de Populações de Cães e Gatos do Estado de São Paulo, preconiza que o recolhimento de cães de uma determinada área não soluciona o problema de animais sem controle no local, se não forem coibidas a disposição de abrigos e, principalmente, de alimentos, como parte de um programa efetivo de controle de populações de cães e gatos que deve incluir ações educativas permanentes, legislação, controle da reprodução e do comércio, registro, identificação animal e concessão de licenças. As atividades isoladas de recolhimento e eliminação de cães e gatos não são efetivas para o controle da dinâmica destas populações, sendo necessário, portanto, atuar na causa do problema: a procriação animal sem controle e a falta de responsabilidade do ser humano quanto à sua posse, propriedade ou guarda. (VIEIRA et all. 2006)

Uma das ações fundamentais no planejamento e elaboração de plano de ações referente às populações de animais de estimação é a identificação da população local de animais. Para a OMS, A identificação e o registro da população animal são essenciais para que a implementação de um programa de controle populacional tenha sucesso. O propósito de um sistema de licenciamento é proporcionar o direito de possuir um animal doméstico, este direito pode ser tirado caso a pessoa seja irresponsável ou cruel com os animais. (OMS, 1990)

A identificação do animal em uma base de dados permite que este seja vinculado a um proprietário. Para a ICAM, os métodos de registro e identificação devem instigar um senso de responsabilidade no proprietário. Registro e identificação são ferramentas importantes para devolver os animais perdidos a seus respectivos proprietários e podem ser importantes recursos para aplicação da

legislação. É importante que as informações de registro e identificação sejam guardadas em um banco central de dados (ou que bancos de dados estejam em conexão de alguma maneira), que sejam acessíveis para todas as pessoas relevantes (ex. veterinários, polícia, guardiões de cães e abrigos municipais). Pode ser necessário o apoio dos governos local e federal para garantir a utilização de um sistema unificado. (ICAM, 2007)

No que tange os abrigos de animais, para onde são encaminhados os animais de rua que não apresentam nenhum tipo de identificação, a ICAM delibera que o registro e identificação compulsórios podem ajudar nos problemas reais enfrentados por abrigos de animais. Quando um cão trazido para um abrigo é identificado, pode ser devolvido sem demora a seu proprietário (evitando o comprometimento do bem-estar do cão e reduzindo o estresse do proprietário). Se não pode ser identificado, é por definição “sem dono”, portanto o abrigo pode aplicar seus procedimentos sem esperar o proprietário se apresentar. Em ambas as circunstâncias, haverá mais espaço nos canis, o que potencialmente aumentará a capacidade dos mesmos. (ICAM, 2007)

Para Vieira (2006), o registro é a anotação oficial dos dados relativos aos proprietários e seus animais. A identificação consiste em atribuir a cada animal um código individual. Ambos compõem um sistema de informação com dados que relacionam os proprietários a seus animais, possibilitando conhecer e dimensionar as populações de cães e gatos, subsidiar o planejamento das políticas de saúde pública, conhecer os proprietários e seus animais, avaliar o controle (supervisão) do proprietário sobre o animal e responsabilizar os proprietários pela manutenção de seus animais. Atividades de informação, educação e comunicação, para incentivar à comunidade ao registro e à identificação de cães e gatos, deverão ser implantadas e permanentemente implementadas.

As informações solicitadas para realizar o registro dos animais e dos proprietários devem garantir a eficácia do sistema em relacionar o proprietário ao seu animal. De acordo com o Programa de Controle de Populações de Cães e Gatos do Estado de São Paulo, o cadastro deverá conter as seguintes informações: Dados do proprietário (Nome, RG, órgão expedidor, CPF, endereço, cidade, estado, CEP, e-mail e telefone), dados do animal (número do microchip; número da plaqueta; nome do animal; espécie; raça; sexo; idade; mês e ano de vacinação

contra a raiva; classificação (restrito, semi restrito, de vizinhança ou comunidade); condição reprodutiva (esterilizado ou não) e endereço de permanência do animal, código do município, data do cadastro e órgão emissor. (VIEIRA et all. 2006)

O sistema de controle e identificação de animais domésticos permite também que critérios sejam adotados para a aquisição de cães, como a emissão de licença para aquisição de animais de raças controladas (cães perigosos). As licenças podem encorajar a guarda responsável, pois solicitam que os novos proprietários completem um “certificado de posse de cão”. (ICAM, 2007)

A quase totalidade dos municípios brasileiros possui as mesmas demandas no tocante ao serviço de controle animal, e enfrentam os mesmos problemas relativos a animais sem controle: crias indesejadas; abandono animal; superpopulação de animais não domiciliados; criação e comercialização desregrada ou irregular; denúncias de maus-tratos; mordeduras e demais agravos; desconhecimento ou não incorporação dos preceitos de bem-estar animal para o desenvolvimento de um programa de controle. (VIEIRA et all. 2006)

Para a OMS, a identificação, e posterior estudo da população de cães permite estabelecer uma série de informações para o planejamento e a implementação de esquemas de gestão e ou controle de zoonoses transmitidas por cães, e monitorar a efetividade de medidas particulares para condução de pesquisas para melhoria no sistema de gestão. (OMS, 1990)

## **2.2 Controle de zoonoses**

A vigilância ambiental em saúde se configura como um conjunto de ações que proporcionam o conhecimento e a detecção de qualquer mudança nos fatores determinantes e condicionantes do meio ambiente que interferem na saúde humana, com a finalidade de recomendar e adotar as medidas de prevenção e controle dos fatores de riscos e das doenças ou agravos relacionados à variável ambiental. (OMS, 1990)

As zoonoses são doenças ou infecções que são transmissíveis naturalmente de animais vertebrados para humanos. Segundo a OMS, os animais têm papel fundamental na manutenção de infecções zoonóticas na natureza. As zoonoses

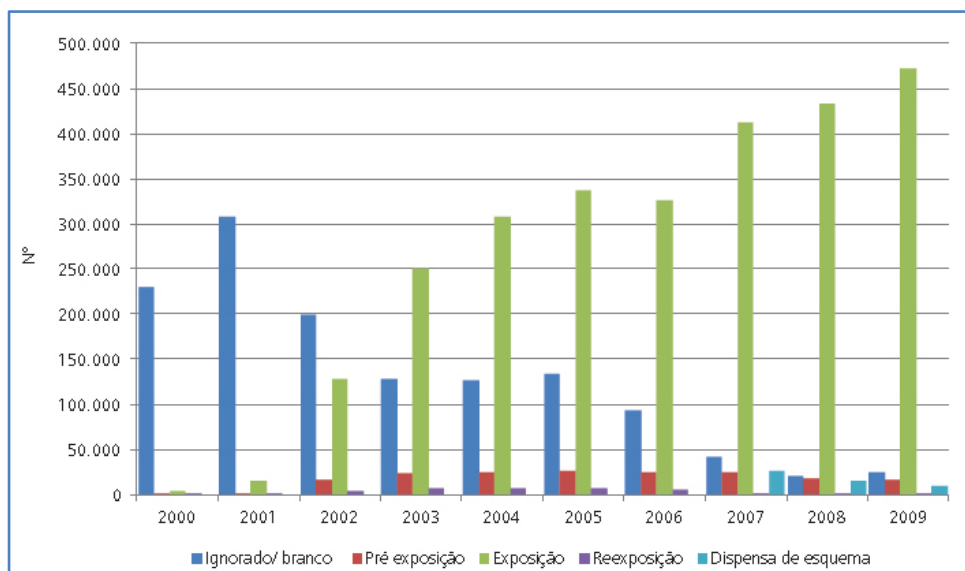
podem ser bacteriais, virais ou parasíticas. Tanto quanto um problema de saúde pública, as zoonoses previnem a produção eficiente de alimentos de origem animal e cria obstáculos para o comércio internacional de produtos de mesma origem. (OMS, 1990)

De acordo com Vieira, historicamente, o controle de zoonoses nasce no cerne da “revolução pasteuriana”, quando a vacina contra a raiva é desenvolvida, abrindo novas perspectivas para a saúde pública. O avanço constante da investigação científica e o desenvolvimento das tecnologias na área da biociência, aliados ao surgimento de novos paradigmas de saúde e bem-estar social, permitem a humanidade implementar mecanismos mais eficientes de controle de agravos e doenças. (VIEIRA et all. 2006)

As zoonoses são muitas vezes a causa primária de preocupação em relação às populações de cães de rua, particularmente para os governos local e federal, que tem responsabilidades quanto às questões de saúde pública. A raiva é uma doença fatal e os cães são os vetores de transmissão mais comuns para humanos (ICAM, 2007), o que torna a identificação e o monitoramento da população de cães e gatos ferramentas centrais nas ações de prevenção das zoonoses, permitindo o armazenamento de informações como histórico de vacinas e de doenças de cada animal individualmente.

Segundo o Ministério da Saúde, no Brasil, de 2000 a 2009, anualmente uma média de 425.400 pessoas procuraram atendimento médico por terem sido expostas ou por se julgarem expostas ao vírus da raiva, conforme apresentado na figura 1. Destas mais de 64% receberam esquema de profilaxia de pós-exposição. De 1998 a 2009, foram notificados 218 casos de raiva humana; 144 pacientes (66%) não receberam nenhum esquema profilático. Destes 23 pacientes (10,5%) foram a óbito por ter sido inadequadamente vacinados e/ou porque abandonaram o esquema profilático. (BRASIL, 2014)

**Figura 1 - atendimentos e profilaxia da raiva humana no Brasil, 2000 a 2009.**



**(BRASIL, 2014)**

Os animais componentes dos estratos populacionais sem controle podem apresentar um crescimento exponencial se não forem impostas limitações da capacidade de suporte do ambiente e desta forma ocasionar na poluição do solo e de cursos hídricos com dejetos, dispersão de resíduos comuns e de alimentos que representantes da comunidade lhes oferecem, sem os cuidados de recolher os restos não aproveitados. Estas práticas propiciam a atração de animais de outras espécies (roedores e aves, entre outros), determinando incômodos à qualidade de vida da comunidade. (VIEIRA et all. 2006)

Os cães de vizinhança ou comunitários, também identificados como cães de rua podem, segundo o ICAM, podem levar a conflitos entre homem e animal e se tornar um problema de bem-estar evidente. Em muitas situações a população de cães de rua deverá ser controlada devido à pressão da população sobre questões de saúde pública e de bem-estar dos próprios animais. (ICAM, 2007)

Pesquisa realizada em Canoas, RS, mostrou que, em levantamento realizado pelos agentes comunitários de saúde junto à comunidade, o principal problema encontrado foi o grande número de cães não domiciliados que agrediam e constrangiam os transeuntes, se envolviam em acidentes de trânsito e eram portadores ou reservatório de agentes biológicos causadores de enfermidades. (MOUTINHO, 2015)



### **2.3 Guarda responsável e bem-estar animal**

Um sistema que permita o registro e a identificação dos animais domésticos não traz benefícios apenas para a sociedade local, mas também para os próprios animais, tendo em vista que as ações desta natureza são preconizadas como uma das práticas que permitem atribuir responsabilidades aos proprietários e ferramentas de fiscalização por parte do poder público. Para Silvano, a guarda responsável de animais de companhia se configura como uma das práticas para promoção do bem-estar animal, fornecendo subsídios para conscientização quanto às necessidades básicas para uma relação saudável tanto para os animais, quanto para seus proprietários. (SILVANO et al., 2010)

De acordo com a União Internacional Protetora dos Animais, na guarda responsável, um conjunto de regras devem ser seguidas de forma a se garantir a saúde, a segurança e o conforto dos animais. Ao decidir-se por acolher um animal, é necessário ter em mente que ele viverá cerca de doze anos e que necessitará de cuidados, independentemente das mudanças que ocorram na rotina do responsável neste período. A escolha do animal deve ser de acordo com as características de comportamento e tamanho condizentes com o espaço disponível e com os próprios hábitos. É necessário garantir a assistência veterinária adequada, bem como indicado a esterilização do animal, macho ou fêmea, para evitar crias indesejadas que resultam no abandono e em superpopulação de animais. O dono do animal deve proporcionar alimentação adequada à espécie, água fresca e espaço adequado, ao abrigo de sol e chuva, sem prendê-lo a correntes, cordas ou aparatos similares. Deve ainda mantê-lo em boas condições de higiene e jamais submetê-lo a maus-tratos. (UIPA, 2014)

Para Vieira, entende-se por propriedade responsável a condição na qual o guardião de um animal de companhia aceita e se compromete em assumir uma série de deveres centrados no atendimento das necessidades físicas, psicológicas e ambientais de seu animal, assim como prevenir os riscos (potencial de agressão, transmissão de doenças ou danos a terceiros) que seu animal possa causar à comunidade ou ao ambiente, como interpretado pela legislação pertinente. (VIEIRA et all. 2006)

A Organização Mundial para Saúde Animal, através do seu Código de Saúde de Animais Terrestres, afirma que existe uma relação crítica entre saúde

animal e bem-estar animal. Um animal é considerado vivendo sob condições de bem-estar quando está saudável, confortável, bem nutrido e seguro e apto a expressar seu comportamento inato. (OIE, 2010)

Em 1993, na Inglaterra, o Comitê de bem-estar de Animais definiu as cinco liberdades para a avaliação do bem-estar animal. Elas são: Liberdade nutricional, liberdade sanitária, liberdade comportamental, liberdade psicológica e liberdade ambiental. A implementação das cinco liberdades vem ao encontro da ideologia de uma grande parte dos médicos veterinários brasileiros. (AMARA, 2012)

#### **2.4 Legislações sobre animais de companhia no Brasil**

As primeiras leis relacionadas ao controle de populações animais remonta ao final do século 19, quando os trabalhos de Louis Pasteur passaram a associar a transmissão da raiva à espécie humana por cães infectados. As leis que regulam as ações de controle animal, como qualquer norma legal, devem acompanhar a evolução técnica, social, histórica, ética e política. E já que tanto elas quanto as sanções que dela advêm tem caráter não somente punitivo, mas também preventivo e educacional, devem conduzir a sociedade e os órgãos competentes a sua aplicação e cumprimento, criando serviços, delimitando atribuições e especificando procedimentos, sob a orientação e atendimento aos princípios norteadores do direito e da ética. (VIEIRA et all. 2006)

A responsabilidade de garantia da manutenção da saúde pública, no que tange as ações de controle da população animal, recai sobre os municípios e órgãos executores do controle de zoonoses, cuja criação e atribuição encontram-se reguladas por lei. Programas de controle animal podem ser desenvolvidos e regulados inicialmente através de portarias, para que sejam viabilizados, experimentados e posteriormente regulamentados em lei. Devem ser previstos um conjunto de ações integradas, tendo em vista que já se comprovou a ineficiência, por exemplo, de campanhas adoção e de ações de controle reprodutivo independente de ações preventivas de saúde animal, como vacinação, desverminação e educação para propriedade responsável. Para que um programa de controle populacional venha a ser implementado com sucesso, as ações devem ser executadas de maneira coordenada para que os resultados desejados sejam atingidos de maneira efetiva. A maioria dos centros urbanos enfrenta o problema de cães e gatos

mantidos sem controle, oferecendo riscos à saúde, à segurança pública, à saúde animal e ao meio ambiente, onerando o poder público com investimentos necessários para o controle populacional. Em situações de desequilíbrio, a intervenção para o controle da reprodução dos cães e gatos é de fundamental importância para a promoção da saúde. (VIEIRA et all. 2006)

Segundo o ordenamento civil brasileiro, os animais são tidos como res (Latim), ou coisa, portanto passíveis de apropriação resguardados todos os efeitos inerentes a este ato. Todos os animais são tutelados pelo Estado e é obrigação do poder público zelar e protegê-los. No entanto, para Vieira, a conceituação dada pela lei civil deve ser apreciada em uma análise sistemática de todo o ordenamento jurídico brasileiro, porque providos de vida biológica, os animais não são simplesmente coisas ou meros números, mas individualidades bio-psicológicas, que vêm recebendo o reconhecimento jurídico em todas as partes do mundo. A adoção de novas posturas e a otimização dos serviços de vigilância em saúde, de controle de zoonoses e de controle de população animal são clamores sociais, morais e jurídicos. A implantação de um programa de controle animal, além da alocação de recursos financeiros, técnicos e equipes de trabalho, exige planejamento que englobe: estudo prévio (diagnóstico), ações preventivas, controle, monitoramento, avaliação e dedicação permanente (que exige o envolvimento e o propósito de todos). (VIEIRA et all. 2006)

**Tabela 1 - Mapeamento das leis federais, relacionadas ao cadastramento a população de animais de companhia.**

Norma	Artigo	Assunto	O que dispõe
Constituição Federal 1988	Art. 225, inciso VII do §1º	Direito ao meio ambiente e dever de defender e preservar.	Para assegurar a efetividade desse direito: ao meio ambiente ecologicamente equilibrado, bem de uso comum do povo e essencial a sadia qualidade de vida, impondo-se ao Poder público e à coletividade o dever de defendê-lo e preservá-lo para as presentes e futuras gerações, incumbe ao poder público: – Proteger a fauna e flora,

			vedadas na forma de lei, as práticas que coloquem em risco sua função ecológica, provoquem extinção de espécies ou submetam os animais a crueldade.
Lei 9605/1998	Art. 32º, §2º	Prática de maus-tratos	-Praticar ato de abuso, maus-tratos, ferir ou mutilar animais silvestres, domésticos ou domesticados, nativos ou exóticos. – A pena é aumentada de um sexto a um terço se ocorre morte do animal.
Decreto 6514/08	Art. 29	Especificação das sanções aplicáveis	Praticar ato de abuso, maus-tratos, ferir ou mutilar animais silvestres, domésticos ou domesticados, nativos ou exóticos: multa de R\$500,00 (quinhentos reais) a R\$3000,00 (três mil reais) por indivíduo.
Resolução CONAMA 394/2007	Art. 1º	Critério na determinação das espécies	Estabelece os critérios na determinação das espécies da fauna silvestre, cuja criação e comercialização poderá ser permitida com o animal de estimação.
Resolução CONAMA 394/2007	Art. 6º	Sistema informatizado	O IBAMA disponibilizará um Sistema informatizado de fauna <i>ex situ</i> para controle, monitoramento e rastreabilidade de animais e criadouros, estabelecimentos comerciais e compradores.

(CARDOSO NETO; RIMES; SOBERÓN, 2016)

Além das leis descritas na Tabela 1, existe um projeto de lei tramitando pelo congresso, no momento em que este trabalho está sendo escrito. O projeto de lei n.º 4.564, de 2016, define quais condutas são consideradas abusivas contra animais e sugere punição mais rigorosa. Pela lei atual, se os maus-tratos forem constatados, o agressor recebe pena de três meses a um ano de prisão, o projeto prevê uma pena de dois a oito anos. As condutas definidas como maus-tratos são: agressões ao ser vivo tais como abandono, espancamento, mutilar órgãos ou membros do animal, machucar ou causar lesões, açoitá-lo ou castigar, envenenar, deixar o animal sem

água ou comida por muito tempo, deixar o animal em local insalubre ou perigoso, privar de assistência veterinária, deixar o animal preso sem condições de proteger do sol ou da chuva ou obrigar os animais a trabalho excessivo. (FLORIANO, 2016)

## **2.5 Controle da população de animais de companhia no exterior**

Diversos países vêm desenvolvendo políticas e implementado ações relacionadas ao controle populacional de animais de companhia nas últimas décadas, sobretudo nações industrializadas da Europa e da América do Norte.

Dos documentos de cunho internacional podemos citar primordialmente a Declaração Universal dos Direitos Animais, que é uma proposta inicialmente sugerida pelo cientista Georges Heuse, em prol da causa pela defesa dos direitos animais à UNESCO em 15 de Outubro de 1978, em Paris. Esta visa criar de forma genérica critérios judiciais com respeito aos direitos animais. Sua motivação foi as práticas de extermínio considerado um crime pelas organizações nacionais e internacionais de proteção. (UNESCO, 1978)

Em 1991, o Parlamento Italiano aprovou a Lei Nacional nº 281, “Animais de Companhia e a Prevenção do Abandono” (Legge 14 de agosto de 1991. Legge quadro in materia de affezione e prevenzione del randagism. Gazzeta Ufficiale 30 de agosto, 1991, n. 203), que trata das regulamentações de abrigos de animais, prove sobre a assistência e prevenção do governo quanto aos cães e gatos de rua e um registro nacional de cães. Esta lei tornou ilegal realizar a eutanásia em cães e gatos a menos que eles estejam submetidos a uma situação de morte incurável, mudando a maneira como se realiza o recolhimento de cães na Itália. (SILVANO et all, 2010)

Na República da Irlanda, a obrigatoriedade da identificação de animais de companhia através de microchips passou a ser exigida em 2015, e em 2016, Inglaterra, Escócia e País de Gales tornaram a prática compulsória. A educação sobre a importância das práticas adotadas, se dá por meios de mídia, fazendo com que toda população tenha acesso e conheça a aplicabilidade das leis. O valor do licenciamento para animais esterilizados deve ser menor do que o valor para animais não esterilizados, assim incentivando o procedimento cirúrgico de esterilização e evitando a reprodução de forma descontrolada. (CARDOSO NETO; RIMES; SOBERÓN, 2016)

A Ville de Verdun, em Mottréal, Canadá, Apresentou no dia 03 de maio de 2011 uma Carta sobre conduta adequada aos proprietários de animais de estimação. É feito o controle do número de cães e gatos não castrados, os limitando, a um por residência, sendo obrigatório registrar filhotes com mais de noventa dias. Além do microchip, cães e gatos devem utilizar a tag de identificação expedido pelo distrito. (ARRONDISSEMENT DE VERDUN, 2017)

Na cidade de Henderson, Estados Unidos, o estatuto municipal requer que cães ou gatos com quatro meses ou mais sejam esterilizados, exceto quando se tratar de circunstâncias específicas. A portaria foi aprovada em três de agosto de 2010. A obtenção da licença é compulsória para todos os animais com idade superior a três meses ou em até no máximo 30 dias após o animal ser trazido a cidade. (CITY OF ANDERSON, 2017).

## **3 DESENVOLVIMENTO**

### **3.1 Processo de Software**

Segundo Sommerville, um processo de software é um conjunto de atividades relacionadas que levam a produção de um produto de software. Ao se discutir os processos, costumamos falar sobre suas atividades, como a especificação de um modelo de dados e o projeto de interface de usuário. As quatro atividades básicas do processo são: especificação, desenvolvimento, validação e evolução. A especificação do software, ou engenharia de requisitos, é o processo da compreensão e definição dos serviços requisitados do sistema e identificação de restrições à operação e desenvolvimento do sistema. Na fase de desenvolvimento ocorre a conversão de uma especificação do sistema em um sistema executável. A maioria dos softwares interage com outros sistemas de software, incluindo o sistema operacional, o banco de dados, o middleware e outros aplicativos. A fase de validação tem a intenção de mostrar que um software se adequa e satisfaz as especificações levantadas na engenharia de requisitos, e por fim a fase de evolução permite utilizar a flexibilidade proporcionada pelos sistemas de software para aplicar novas funcionalidades e aprimoramento nas características atuais. (SOMMERVILLE, 2010)

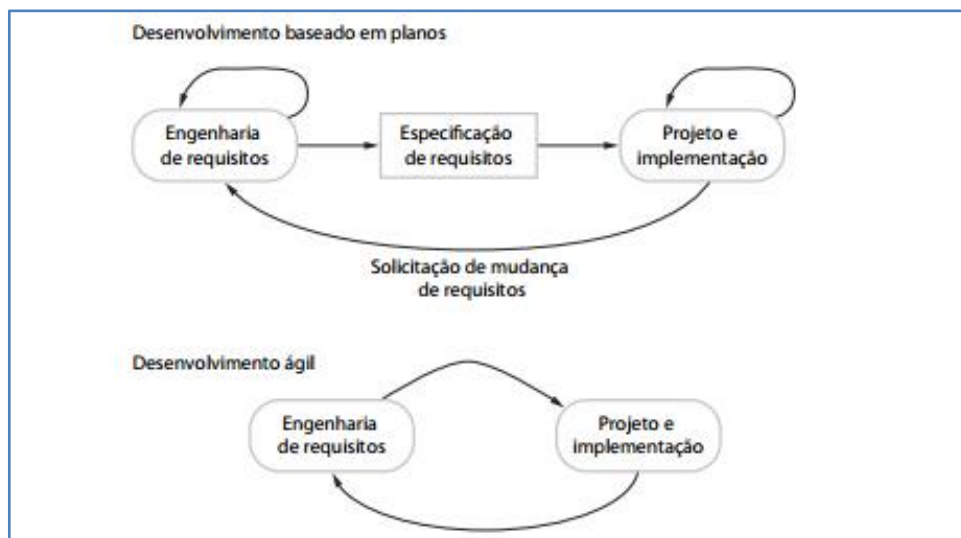
. As especificações e a elaboração do documento de requisitos do sistema apresentado neste trabalho, RGA – Registro Geral de Animais, foram realizadas em conjunto com o quadro de funcionários da Fundação Ambiental de Araranguá e da Secretaria Municipal de Saúde de Araranguá, responsáveis pela aplicação das políticas municipais vigentes de identificação de animais por meio de microchips e controle da capacidade reprodutiva, através de reuniões periódicas e colaborativas durante o processo de desenvolvimento.

### 3.2 Métodos Ágeis

Os processos de desenvolvimento de software foram concebidos para produzir, rapidamente, softwares úteis, que compartilham de características fundamentais, como o projeto e implementação intercalados, o desenvolvimento em uma série de versões e o desenvolvimento da interface de usuário através de um sistema interativo, que permita a criação rápida do projeto de interface. (SOMMERVILLE, 2010)

Por apresentar uma característica dinâmica, onde a equipe de programação e os clientes necessitam estar em comunicação constante durante a implementação da aplicação, e o desenvolvimento ocorre de maneira incremental, os métodos ágeis são usados principalmente para o desenvolvimento de sistemas de pequeno e médio *port* e, diferentemente da maneira linear de desenvolvimento onde as mudanças no projeto original ocorrem apenas ao termino do projeto. As comparações entre os fluxos de desenvolvimento são demonstradas na figura 2. (SOMMERVILLE, 2010)

Figura 2 - Comparação entre os métodos de desenvolvimento baseado em planos e desenvolvimento ágil.



Fonte: Autores

#### 3.2.1 RAD (Rapid Application Development)

Desenvolvimento rápido de aplicações é uma abordagem para desenvolvimento de software com objetivo de entregar software rapidamente.



Comumente envolve o uso de ferramentas de programação de banco de dados e de apoio ao desenvolvimento, como geradores de telas e relatórios. (SOMMERVILLE, 2010)

O desenvolvimento do sistema de registro geral de animais ocorreu de acordo com os preceitos do desenvolvimento ágil, uma vez que as implementações foram ocorrendo de maneira incremental, de acordo com as adequações observadas em encontros periódicos entre os desenvolvedores e as entidades responsáveis pelo controle ambiental e sanitário do município de Araranguá.

### **3.3 Requisitos funcionais e não funcionais.**

Os requisitos funcionais e não funcionais do sistema de Registro Geral de Animais têm a finalidade de descrever as ações que o sistema deverá realizar, as funcionalidades que ele oferece e suas restrições.

#### **3.3.1 Requisitos funcionais**

Os requisitos funcionais são as declarações dos serviços que o sistema deve fornecer como ele deve reagir a entradas específicas e as restrições que deverão ser implementadas.

##### **3.3.1.1 Registrar e armazenar dados sobre o animal**

O sistema deverá armazenar as informações sobre o animal, nome, idade, espécie e histórico de vacinas. Deve ser atribuído um número de identificação individual para cada animal.

##### **3.3.1.2 Registrar e armazenar dados sobre os proprietários**

O sistema deverá armazenar informações sobre o proprietário, nome, CPF, RG, endereço e características sobre o espaço disponível para abrigar o animal.

##### **3.3.1.3 Permitir a alteração e exclusão de registros**

O sistema deverá permitir ao usuário a alteração e exclusão dos registros, de maneira intuitiva, através da interface gráfica.

#### **3.3.1.4 Permitir que os usuários localizem um registro específico**

O sistema deverá permitir que o usuário realize uma busca na base de dados por um registro específico, e consiga realizar as operações de alteração ou exclusão a partir dos resultados.

#### **3.3.1.5 Associar cada animal a apenas um proprietário**

Cada animal deve ser associado a apenas um proprietário, garantindo que as responsabilidades relativas aos cuidados a serem adotados com a posse sejam atribuídos um indivíduo em específico.

#### **3.3.1.6 Permitir a vinculação de diversas vacinas a um mesmo animal**

O sistema deverá permitir que cada animal possa ser associado a várias vacinas, registrando-se o tipo de vacina e a data de aplicação permitindo obter-se um histórico de vacinas por animal.

### **3.3.2 Requisitos Não Funcionais**

#### **3.3.2.1 Operabilidade em modo off-line**

O sistema deve operar a partir de um servidor local, ficando disponível para os usuários mesmo com instabilidades de conexão através da internet.

#### **3.3.2.2 Autenticação de usuários**

O sistema deverá apresentar um sistema de autenticação para vincular cada registro feito a um operador específico do sistema e apresentar diferentes níveis de acesso para funções operacionais e administrativas.

#### **3.3.2.3 Utilização de design responsivo na interface gráfica**

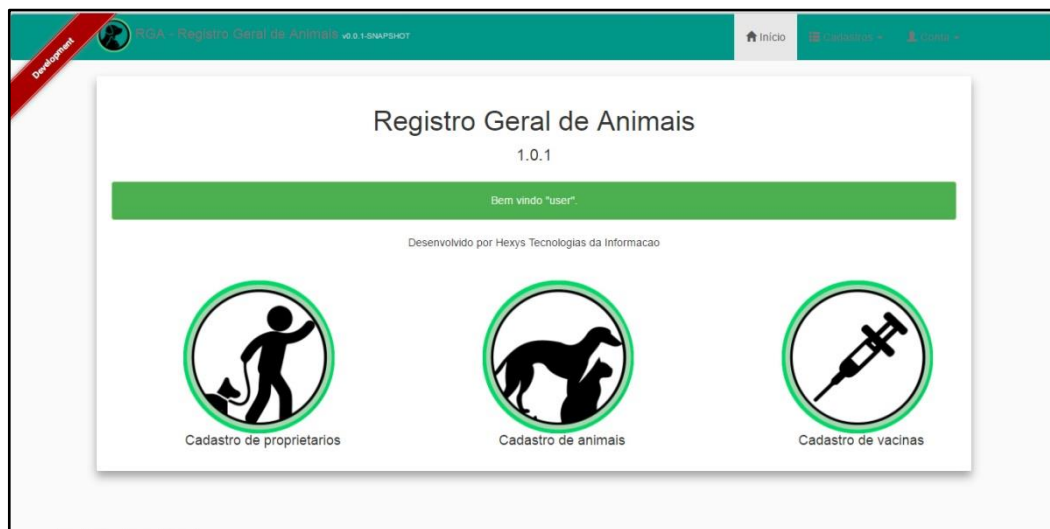
Deverá ser utilizado um layout responsivo na construção da interface gráfica, para que seja possível acessar o sistema de diferentes dispositivos com resoluções de tela distintas.

### 3.4 Interface de usuário do Registro Geral de Animais

O desenvolvimento da interface gráfica da aplicação foi realizado utilizando os frameworks AngularJS e Bootstrap, aplicando as características visuais aos elementos HTML das áreas do sistema.

A área inicial da aplicação, quando acessada com o perfil de operador, apresenta de forma direta as principais ferramentas disponíveis para a operacionalização do Sistema: o cadastro de proprietários, onde é feito o preenchimento das informações relativas ao responsável pela guarda do animal, conforme apresentado na figura 3. O segundo formulário de cadastro, o cadastro de animais, permite o registro das informações relativas ao animal, como capacidade reprodutiva (castrado ou não), nome, idade e raça. Cada cadastro de animal deve ser vinculado a um registro de proprietário realizado previamente, atendendo ao requisito funcional 02. Por último, o Sistema registro de vacinas, apresenta um formulário de cadastro que associa um animal previamente cadastrado a uma vacina específica, selecionada através de um menu dropdown e registra a data da aplicação da vacina.

**Figura 3 - Área inicial da aplicação em modo usuário, com os links para os formulários de cadastro de proprietários, de animais e de vacinas.**



**Fonte: Autores**

Após o preenchimento do formulário de cadastro, todos os registros referentes aquela entidade (responsável, animal e vacina), são apresentados em

uma lista, e podem ser facilmente alterados, editados ou excluídos através de botões presentes ao lado de cada registro, conforme demonstrado na figura 4 na visualização de registros de proprietários.

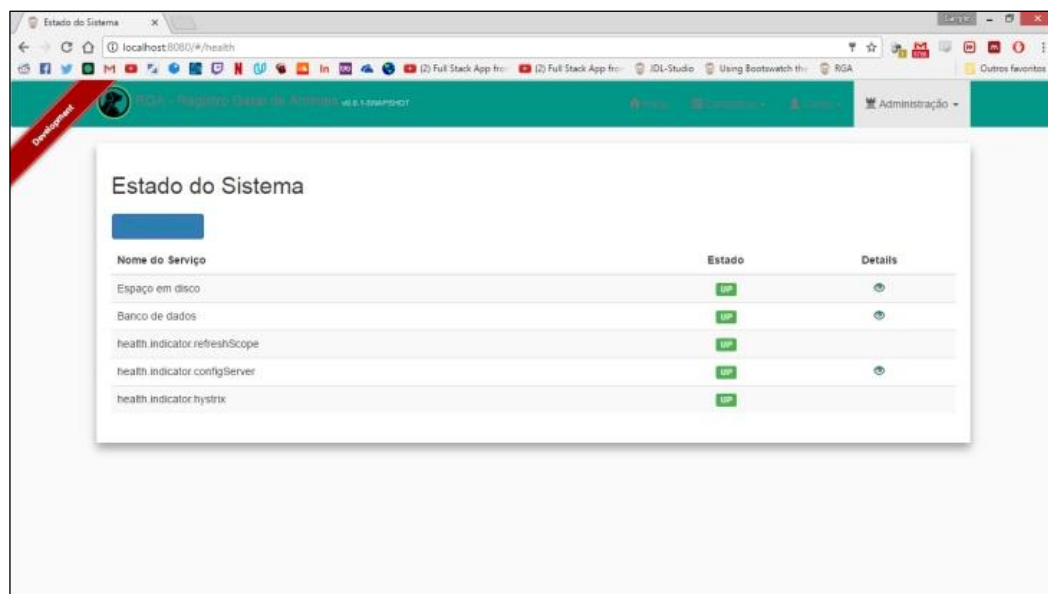
**Figura 4 - Relação dos animais cadastrados, com as funcionalidades de alteração, edição, exclusão e inclusão de novo registro.**

Código	Nome	Nascimento	Chip	Cor	Capacidade Reprodutiva	Sexo	Especie	Proprietario	
65	Jolie	31 de mai de 2017 00:00:00	1	Preta	CASTRADO	FEMEA	CAO	Cariyle Torres Bezerra de Menezes Filho	<a href="#">Visualizar</a> <a href="#">Editar</a> <a href="#">Excluir</a>

**Fonte: Autores**

O framework Jhipster, utilizado no desenvolvimento, é um gerador Yeoman, que se utiliza da linha de comando para gerar partes fundamentais de uma aplicação, como as usabilidades necessárias da base de dados, ferramentas de construção como o Gulp, que proporciona agilidade e eficiência em tarefas como compilar CSS de pré-processadores, verificar erros JavaScript e otimizar imagens, e gerenciadores de pacotes como o Bower, utilizado para resolver as dependências client-side. Estão presentes também uma vasta gama de tecnologias Spring, como Spring Boot, Spring Security e Spring MVC, com uma serie de configurações para monitoramento, apresentada na figura 5, gerenciamento e armazenamento de logs do sistema.

**Figura 5 - Módulo de monitoramento do estado do Sistema em modo administrador com informações relativas ao banco de dados e espaço em disco.**



**Fonte: Autores**

### **3.5 Sistema de gerenciamento de banco de dados**

O modelo de gerenciamento de banco de dados adotado foi o MySQL, pela sua portabilidade, pouca exigência de recursos de hardware e apresentar um excelente desempenho e portabilidade. O framework Hibernate faz o intermédio entre o banco de dados e a aplicação, executando as instruções SQL para recuperar apenas as informações necessárias e evitar consultas complexas para obtenção de determinado dado. O modelo relacional foi concebido de forma a tornar o proprietário à entidade central do sistema, pois através de seu cadastro, é possível vincular um animal ao seu registro, e subsequentemente, com o registro do animal é possível vincular a ele um número 'n' de vacinas. Para os elementos fixos, como capacidade reprodutiva do animal, espécie do animal e sexo do proprietário, foram criadas enumerações com valores fixos, acessíveis através de menus drop down nos formulários de cadastro aos quais estão vinculados.

## **3.6 Tecnologias utilizadas**

### **3.6.1 AngularJS**

AngularJS é um framework de código aberto construído em JavaScript mantido pela Google. Foi originalmente desenvolvido em 2009 por Miško Hevery. Este framework é utilizado para criação de front-end de aplicações web, além disso, fornece também um conjunto de funcionalidades que tornam o desenvolvimento web muito mais fácil e empolgante, tais como o DataBinding, templates, fácil uso do Ajax, controllers, etc. (PANDA, 2014)

AngularJS funciona do lado do cliente e permite estender a sintaxe de HTML através de directivas, tags que servem para alterar o comportamento padrão do HTML através de novos parâmetros. Outra funcionalidade é a injeção de dependências diminuindo a quantidade de código a ser escrita. (SCHMITZ; LIRA, 2016)

Em meados de 2014 foi anunciado o Angular 2.0, este possui significantes mudanças, principalmente na sintaxe, além de modificações em sua estrutura, adição de novos recursos de componentes e melhoria de performance de processamento. Essas alterações tornam oneroso o trabalho de migração da versão 1.x para 2.x. (RAIBLE, 2015)

Alguns Exemplos de outras tecnologias que se assemelham quase que completamente, ou em parte, das funcionalidades do AngularJS são o Thymeleaf e Jsf.

#### **3.6.1.1 Internationalization (i18n)**

AngularJS provê uma diretiva para internacionalização denominada Angular Translate. Com ela é possível a tradução de páginas, através da referência à arquivos .json contendo o correspondente valor para o respectivo atributo HTML.(GOOGLE, 2010)

### **3.6.2 Bootstrap**

Com o crescente uso de dispositivos móveis, o design de interfaces pensadas de forma a se adaptar a diferentes tamanhos de tela se torna extremamente relevante, e existem no mercado alguns frameworks que padronizam e otimizam o processo produtivo de interfaces web responsivas. Dentre estes um que vem se destacado nos últimos anos entre desenvolvedores frontend é o Bootstrap, um produto open source de Marc Otto e Jacob Thornton, o qual foi lançado quando ambos eram empregados da rede social Twitter, como uma maneira de padronizar as ferramentas frontend dos engenheiros da companhia. (SPURLOCK, 2013)

Desde seu lançamento em agosto de 2011, ele vem evoluindo de sua origem como um projeto inteiramente orientado a padronização CSS para incluir uma gama de plugins JavaScript que adicionam interatividade aos layouts, bem como pacotes de ícones para elaboração de botões e formulários. Ele permite a criação de websites responsivos baseado em uma grid de 12 colunas e 940px de largura. Esta característica permite que o conteúdo presente na página seja otimizado para o dispositivo no qual está sendo exibido, através das media queries em CSS que medem a largura da viewport do browser e então mudam as características das folhas de estilo que são carregadas. (TWITTER, 2017)

### **3.6.3 Bower**

Bower é um gerenciador de pacotes front-end de componentes, tais como estruturas, bibliotecas, ativos e utilitários que contenham fontes HTML, CSS, JavaScript, ou até mesmo arquivos de imagem. Bower não concatena nem minifica o código, apenas instala as versões corretas de pacotes necessários para determinada aplicação e suas dependências. Além disso, evita a duplicidade de dependências.

Bower mantém o controle desses ativos em um arquivo de manifesto chamado bower.json. Também fornece a possibilidade de chamar scripts para acionar o uso de outras ferramentas durante o fluxo de trabalho. (BOWER, 2017)

### **3.6.4 Gradle**

Na década de 1970, surgiram os primeiros softwares para automação de compilação com os makefiles. Em 2000, a Sun lançou um equivalente de makefiles para a plataforma Java, que agora é conhecido como Apache Ant. Ant usou arquivos XML para ordenar a compilação. Logo depois surgiu o Apache Maven, que foi lançado em 2005. (LALOU, 2013)

Maven trouxe a inovação da forma declarativa, que possibilita ao programador, determinar somente o que este gostaria de alcançar, especificando parâmetros em um arquivo de configuração. (BHARATHAN, 2015)

Inspirado nestes dois anteriores surgiu o Gradle. Gradle é uma ferramenta para automação de compilação, gerenciamento de dependências, testes, empacotamento e implantação de software, licenciado sob Apache Software License (ASL). O Gradle combina as boas partes de ambas as ferramentas e fornece recursos adicionais como Groovy, como uma linguagem de especificação de domínio (DSL). Apesar de Gradle ser uma ferramenta de compilação de linguagem JVM, também é compatível com C, C ++, Android e outros. (MITRA, 2015)

Uma dependência é algo que o projeto necessita para funcionar, como por exemplo, bibliotecas. As dependências ficam armazenadas geralmente em repositórios na Internet. Os repositórios de Maven e Ivy por serem mais antigos possuem maior número de artefatos. Gradle suporta ambos, sendo a configuração para inclusão de dependência especificada no arquivo "build.gradle", segundo a sintaxe Groovy. (IKKINK, 2012)

### **3.6.5 Gulp**

Gulp é um sistema de compilação automatizado escrito em JavaScript criado com NodeJS. Gulp tem a capacidade de automatizar o fluxo de trabalho, fornecendo funcionalidades como pré-processamento do CSS, a execução de um servidor HTML e a atualização automática da janela do navegador em qualquer alteração em seu código.

O Gulp usa pequenos plugins para modificar e processar os arquivos de projeto, otimizar o tempo de carregamento do aplicativo, compactar o tamanho geral



de arquivos, otimizar imagens e eliminar todos os pedidos HTTP desnecessários. (MAYNARD, 2015)

### **3.6.6 Hibernate**

O Hibernate Object Relational Mapper ou ORM, é uma biblioteca que implementa a especificação Java Persistence API (JPA) e simplifica o uso de bancos de dados relacionais em aplicativos Java, apresentando dados relacionais como objetos Java, e vice-versa. Atualmente o Hibernate está na versão 5. (OTTINGER; LINWOOD; MINTER, 2016)

As vantagens do Hibernate são a manutenção e gerenciamento da serialização de objetos (ou mais propriamente dos seus campos) visando correta e automática persistência de dados, e tradução do paradigma orientado a objetos para o modelo relacional. Com Hibernate é possível armazenar objetos em qualquer forma nativa de um determinado banco de dados através de dialetos específicos. (OTTINGER; LINWOOD; MINTER, 2016)

No Hibernate Core destacam-se, uma linguagem de consultas chamada HQL (parecido com SQL), um conjunto de interfaces para consultas usando critérios (Criteria API). (FARIA; NORMANDES JUNIOR, 2015)

### **3.6.7 HTML**

A linguagem HTML (Hypertext Markup Language), ou HTML, é o código básico que compõe a base para cada site na World Wide Web (W3). O HTML é usado para definir a estrutura do texto e outros conteúdos da página e para definir como uma página da web é estruturada. (FREEMAN, 2011)

Atualmente a versão do HTML é a 5. A especificação oficial para HTML5 descreve um grande número de novos recursos que permitem aos desenvolvedores web criar sites que são mais rápidos e inteligentes do que aqueles que poderiam construir usando versões mais antigas do HTML. Estes novos recursos incluem LocalStorage (que permite que os desenvolvedores armazenem dados no computador do usuário) e HTML5 Video (que permite a reprodução de vídeo em seu

navegador sem precisar de um plug-in como o Flash), bem como novos elementos interativos, como data pickers e controles deslizantes. (GOLDSTEIN, 2011)

HTML5 é apenas uma parte de um grande número de padrões que são chamados coletivamente como New Exciting Web Technologies (NEWT). Junto com o HTML5, o NEWT também engloba coisas como WebGL (gráficos 3D no navegador) e GeoLocation (localizando o local do usuário). (WEST, 2013)

### **3.6.8 Java**

Atualmente a linguagem de programação Java é uma das linguagens de programação mais comuns e populares. Criada por volta de 1991 pela Sun Microsystems, que mais tarde seria adquirida pela Oracle. (LUCKOW; MELO, 2015)

O Java evoluiu de uma linguagem universal para uma plataforma inteira. Seu ecossistema combina uma variedade de tecnologias usadas em uma variedade de tarefas, desde a criação de aplicações desktop até grandes portais e serviços web. Além disso, a linguagem Java é ativamente usada para criar software para uma gama de dispositivos: computadores desktop, tablets, smartphones e telefones celulares, e até mesmo eletrodomésticos, assim como a maioria dos programas para Android. (LUCKOW; MELO, 2015)

É uma linguagem bem diversificada, usada em diferentes tipos de aplicações e tarefas. A linguagem Java já percorreu um longo caminho tendo emitido diversos números de versões. A versão atual é o Java 8. (GOSLING, 2015)

#### **3.6.8.1 Características do Java**

Uma das características fundamentais da linguagem Java é o paradigma de orientação à objetos.

Outra característica fundamental trata de sua arquitetura interna. O código é primeiro traduzido para um código de byte especial, independentemente da plataforma. E então este código (ByteCode) é executado pela JVM (Java Virtual Machine). A este respeito, Java difere de linguagens como PHP ou Perl, onde o código é interpretado, porém, ao mesmo tempo, Java não é uma linguagem puramente compilada como C ou C ++. (LINDHOLM, 2015)

Esta arquitetura fornece aos programas, portabilidade de plataforma cruzada, de modo que estes programas possam ser executados sem recompilação em diferentes plataformas como Windows, Linux, Solaris, etc. (LINDHOLM, 2015)

### **3.6.9 JHipster**

JHipster foi iniciado por Julien Dubois em outubro de 2013 com a versão 0.3.1 escrito em Java e JavaScript.

JHipster é um projeto de código aberto baseado em Yeoman, que combina principalmente três frameworks de desenvolvimento web: Bootstrap, AngularJS ( ou Angular 2 ) e Spring Boot. Porém com o advento de sua evolução, incorporou outros frameworks como Gradle, Gulp, Bower, etc. (RAIBLE, 2015)

#### **3.6.9.1 JDL**

JHipster Domain Language (JDL) é uma linguagem de domínio específico do JHipster, onde é possível descrever entidades e seus relacionamentos em um único arquivo (ou mais de um) com uma sintaxe simples e fácil de usar. Existe um editor disponível on-line e também um IDE JDL-Studio baseado em Eclipse usado para criar uma visualização JDL e UML. O arquivo JDL é importado para o JHipster. (JHIPSTER, 2016)

### **3.6.10 Liquibase**

Basicamente todas as alterações no banco de dados são armazenadas em arquivos de texto (XML, YAML, JSON ou SQL). As mudanças são aplicadas e armazenadas no banco de dados a cada vez que esse é iniciado.

Suas funcionalidades proporcionam aos desenvolvedores uma maneira de ramificar os projetos de banco de dados, versionar as alterações do banco de dados e executar operação de regressão, se necessário. (LIQUIBASE, 2017)

### **3.6.11 NodeJS**

Em 2009, Ryan Dahl introduziu o Node.js para a comunidade JavaScript, inspirado pela percepção da ineficiência do upload no Flickr, o site de compartilhamento de imagem.

NodeJS no fundo, é uma máquina virtual JavaScript autônoma, baseada no motor V8 do Google, desenvolvido em 2008. O motor V8 pode compilar JavaScript em código de nível de máquina de maneira muito rápida e eficiente. (KROL, 2014)

Seu modelo segue uma filosofia diferente de servidores Java, PHP e ASP.NET. Pois ao invés de instanciar um processo para cada solicitação resultante, NodeJS utiliza a mesma thread com recursos compartilhados, fornecendo maior desempenho e escalabilidade. NodeJS não possui bloqueios, para chamadas de I/O. Node também pode suportar dezenas de milhares de conexões simultâneas. (HERRON, 2013)

Segundo Pasquili (2013), NodeJS pretende resolver os seguintes problemas:

- Atender a milhares de clientes simultâneos de forma eficiente;
- Escalonar aplicativos em rede além de um único servidor;
- Impedindo que as operações de E / S se tornem gargalos;
- Eliminar pontos únicos de falha, confiabilidade;
- Conseguir o paralelismo de forma segura e previsível.

#### **3.6.11.1 Npm**

Npm é o gerenciador de pacotes interno do NodeJS. Com ele pode-se encontrar diversos módulos, que podem ser instalados e usados dentro de um aplicativo. O comando “npm install package” é utilizado para esta função. O código-fonte fica armazenado no diretório node\_modules do projeto. (KROL, 2014)

### **3.6.12 MySQL**

O MySQL é um servidor de banco de dados ou sistema de gerenciamento de banco de dados (SGBD) relacional que tem como idioma nativo do SQL (Structured Query Language).

Foi desenvolvido por David Axmark, Allan Larsson e Michael "Monty" Widenius, inicialmente pela empresa MySQL AB, fundada na Suécia. A primeira versão apresentada ao mundo, foi lançada em 1996. Desde o início, era um produto de código aberto. Desde 2000, o produto foi lançado de acordo com as regras especificadas na GPL. Atualmente o MySQL é mantido pela Oracle. (LANS, 2007)

MySql suporta vários motores de armazenamento, tais como, MyISAM, InnoDB, MEMORY, Berkeley DB. MyISAM é um motor otimizado para leitura, porém na suporta transações nem chaves estrangeiras. (PACHEV, 2007)

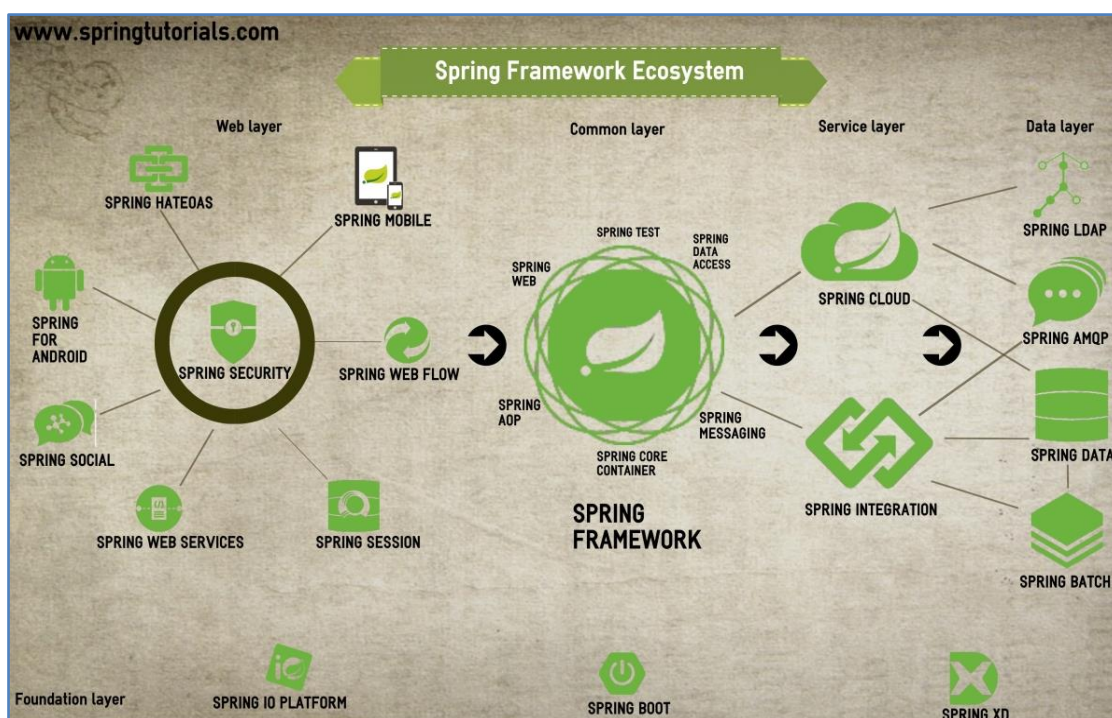
Uma das principais ferramentas que acompanham o MySQL é MySQL Workbench. Trata-se de uma ferramenta para desenvolvedores e arquitetos de dados para modelar, gerar e gerir bancos de dados visualmente. Serve também como RDBMS. (MCLAUGHLIN, 2013)

### **3.6.13 Spring Framework**

O Spring é um Framework de código aberto inicialmente voltado para a Plataforma Java, criado por Rod Johnson e descrito em seu livro Expert One-on-One Desenho e Desenvolvimento J2EE (Wrox, 2002). Tornou-se bastante popular na comunidade Java como alternativa para simplificar o modelo Enterprise JavaBeans (EJB). (KAYAL, 2008)

O Spring possui vários módulos ou projetos como pode ser visto na figura abaixo:

Figura 6 - Ecosistema Spring



Fonte: <http://springtutorials.com/spring-ecosystem>

### 3.6.13.1 Spring Boot

Spring Boot é um projeto que faz parte do ecossistema Spring, e serve para criar aplicativos autônomos que usam um servidor incorporado, tornando-os aplicativos executáveis sem necessidade de deploy e configurações muito complexas. Pode ser combinado com ferramentas de construção automáticas como Maven ou Gradle.

Além disso, fornece módulos atuadores para monitoramento da aplicação, como métricas e auditorias. (GUTIERREZ, 2016)

## 3.7 Estrutura do Projeto

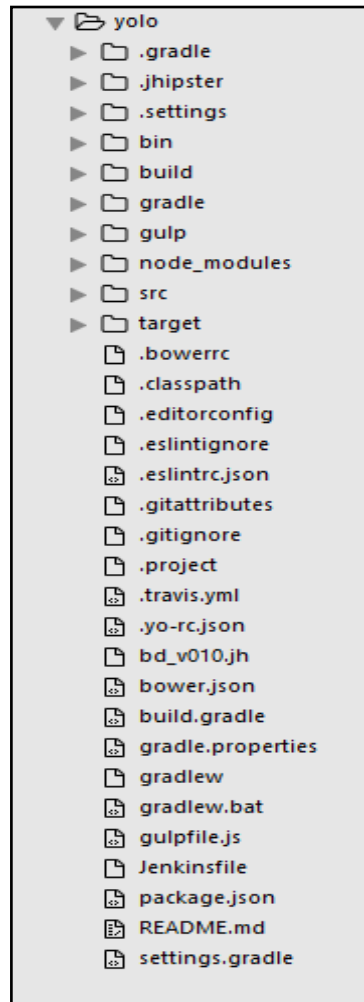
Esta sessão descreve a arquitetura, organização e funcionamento do projeto.

### 3.7.1 Arquétipo

Esta é a estrutura completa de arquivos e diretórios presentes no diretório base. Esta estrutura pode variar com relação ao número de arquivos, dependendo

do editor de código a ser usado. Para este trabalho foi utilizado o editor de código Sublime Text 3.

**Figura 7 - Arquetipo do projeto**



**Fonte: Autores**

### **3.7.2 Diretório base**

Será descrito neste tópico apenas diretórios e arquivos de relevância para este trabalho, os demais são compostos por binários relativos ao funcionamento dos frameworks ou arquivos de configuração relacionados a IDEs ou recursos não constantes nesse projeto, que foram gerados automaticamente pelo JHipster, como é o caso do arquivo *“Jenkinsfile”*, que serve para configurar o framework Jenkins, utilizado na prática de Integração Contínua, através de um servidor de integração.

O diretório *“jhipster”* possui arquivos que descrevem a estrutura de geração de entidades segundo uma sintaxe JSON. Uma interpretação do arquivo *“bd\_v10.jdl”*, que contém uma descrição das entidades do banco de dados e seus

relacionamentos, responsável pela geração de um modelo de entidades, representado pela figura do Anexo I - Modelo relacional de banco de dados.

O arquivo “bd\_v10.jdl”, foi gerado a partir do JDL Studio, e através de sua importação, é uma das formas de se criar entidades no JHipster pelo seu subgerador de entidades.

O código abaixo contém um trecho de exemplo da descrição de entidades segundo uma sintaxe JSON:

**Figura 8 - Descrição entidades JSON**

```
{
  "fluentMethods": true,
  "relationships": [
    {
      "fieldName": "sexo",
      "fieldType": "SexoDoAnimal",
      "fieldValues": "MACHO,FEMEA",
      "fieldValidateRules": [
        "required"
      ]
    },
    {
      "relationshipType": "many-to-one",
      "relationshipName": "especie",
      "otherEntityName": "especie",
      "otherEntityField": "nome"
    }
  ],
}
```

**Fonte: Autores**

O diretório “node\_modules”, contém os módulos ou programas que executam no NodeJS, tais como Bower e Gulp.

O arquivo “.yo-rc.json” contém os parâmetros de configuração, inicialmente escolhidos durante a geração da aplicação, como nome do pacote de código, nome base da aplicação, tipo de banco de dados, etc.

O arquivo “bower.json” possui a descrição do conjunto de dependências front-end a ser utilizada segundo versão de compatibilidade entre elas. O código abaixo da uma mostra deste arquivo.



**Figura 9 - Descrição dependências Bower**

```
{  
  "angular": "1.5.8",  
  "angular-bootstrap": "2.0.0",  
  "jquery": "3.1.0"  
}
```

**Fonte: Autores**

Em contra partida ao arquivo anterior, o arquivo *“build.gradle”*, gerencia dependências back-end, além de listar operações de compilação. No trecho de código abaixo recortado é possível visualizar isso.

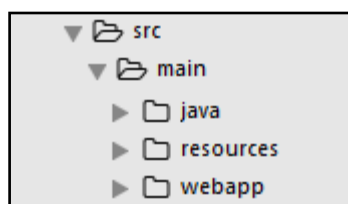
**Figura 10 - Arquivo configuração Gradle**

```
buildscript {  
  repositories {  
    mavenLocal()  
    mavenCentral()  
    maven { url "http://repo.spring.io/plugins-release" }  
  }  
  dependencies {  
    classpath "org.springframework.boot:spring-boot-gradle-plugin:${spring_boot_version}"  
  }  
  compile "org.hibernate:hibernate-core:${hibernate_entitymanager_version}"  
}
```

**Fonte: Autores**

É no diretório *“src”*, que está localizado o código que define as características da aplicação. Este está organizado e dividido como é visto adiante.

**Figura 11 - Diretório src**



**Fonte: Autores**

### **3.7.3 Server-side**

No interior do subdiretório *“Java”*, encontramos a pasta *“domain”*, onde se encontram as classes de entidades do banco de dados. Nelas estão inclusas anotações referentes ao framework Hibernate, para mapeamento objeto-relacional, ou seja, os campos da classe vão se tornar colunas da tabela no banco de dados. Logo abaixo podemos ver parte de uma classe como exemplo.

Figura 12 - Mapeamento classe de entidades Java

```
@Entity
@Table(name = "animal")
@Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
@Document(indexName = "animal")
public class Animal implements Serializable {

    private static final Long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @NotNull
    @ManyToOne
    @Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
    @JoinTable(name = "animal_pessoa",
        joinColumns = @JoinColumn(name="animals_id", referencedColumnName="ID"),
        inverseJoinColumns = @JoinColumn(name="pessoas_id", referencedColumnName="ID"))
    private Set<Pessoa> pessoas = new HashSet<>();

    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return nome;
    }
    public void setName(String nome) {
        this.nome = nome;
    }
    public Set<Pessoa> getPessoas() {
        return pessoas;
    }
    public void setPessoas(Set<Pessoa> pessoas) {
        this.pessoas = pessoas;
    }
}

@Entity
@Table(name = "animal")
@Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
@Document(indexName = "animal")
public class Animal implements Serializable {

    private static final Long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @NotNull
    @Size(max = 50)
    @Column(name = "nome", length = 50, nullable = false)
    private String nome;
    @ManyToOne
    @Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
    @JoinTable(name = "animal_pessoa",
        joinColumns = @JoinColumn(name="animals_id", referencedColumnName="ID"),
        inverseJoinColumns = @JoinColumn(name="pessoas_id", referencedColumnName="ID"))
    private Set<Pessoa> pessoas = new HashSet<>();

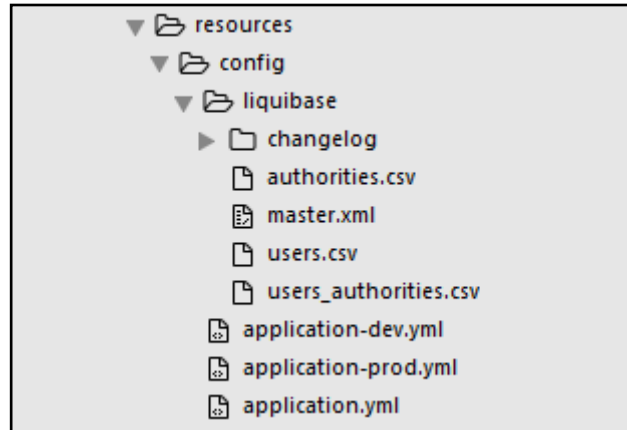
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public Set<Pessoa> getPessoas() {
        return pessoas;
    }
    public void setPessoas(Set<Pessoa> pessoas) {
        this.pessoas = pessoas;
    }
}
```

Fonte: Autores

Dentro de “*domain*”, ainda existe o subdiretório “*security*”, no qual constam arquivos para configuração do Spring Security, módulo do Spring Framework,

responsável pela autenticação de usuários no sistema e permissões de acesso de usuários a determinados locais do sistema.

**Figura 13 - Diretório resources**



**Fonte: Autores**

Adentrando o diretório “resources” segundo a figura 09, encontramos os arquivos “*application-dev.yml*” e “*application-prod.yml*”, que são perfis do Spring de configuração e especificação da fonte de dados. Um é para a fase de desenvolvimento e o outro para a fase de produção (utilização pelo usuário final). Neste projeto foi utilizado o banco de dados MySQL. Abaixo podemos visualizar um pouco do conteúdo desses arquivos.

**Figura 14 - Datasource**

```
datasource:
  type: com.zaxxer.hikari.HikariDataSource
  url: jdbc:mysql://localhost:3306/yolo?useUnicode=true&characterEncoding=utf8&useSSL=false
  name:
  username: root
  password: *****
  hikari:
    data-source-properties:
      cachePrepStmts: true
      prepStmtCacheSize: 250
      prepStmtCacheSqlLimit: 2048
      useServerPrepStmts: true
  jpa:
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    database: MYSQL
```

**Fonte: Autores**

No subdiretório “*liquibase*” segundo a figura 09, o arquivo “*master.xml*” e os arquivos contidos no subdiretório “*changelog*”, são responsáveis pelo gerenciamento da migração do banco de dados, relatando as alterações aplicáveis ao banco.

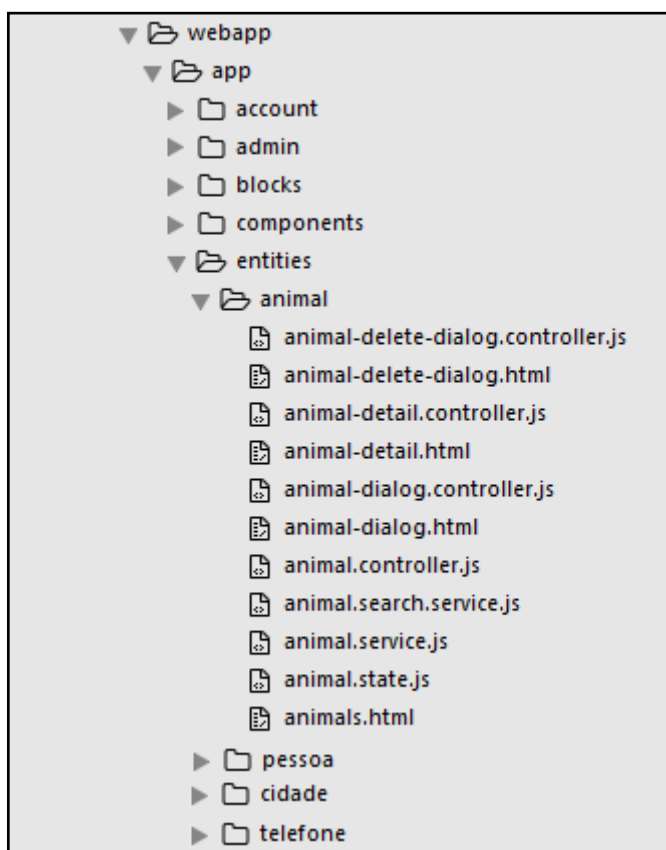
### 3.7.4 Front-side

Esta aplicação possui um padrão de arquitetura de software MVC, porém Angular fica responsável pela view e controller, sendo executado do lado do cliente e obtendo dados através de requisições Ajax.

#### 3.7.4.1 App AngularJS

AngularJS armazena dentro do diretório “*app*”, um subdiretório para cada entidade, como visto na figura abaixo.

Figura 15 - Diretório webapp



Fonte: Autores

A pasta “*animal*” é composta por vários *templates* de extensão HTML. Este conjunto de arquivos compõe um CRUD, acrônimo de Create, Read, Update e Delete. Neste caso vamos ter uma página para cadastrar um animal, uma para atualizar suas informações e uma para removê-lo caso necessário.

Quanto aos arquivos com extensão JavaScript, eles representam a camada de controle. São objetos, que executam ações como, por exemplo, solicitar a inserção de um artefato no banco de dados.

### 3.7.4.2 Internacionalization

O diretório "i18n" da figura abaixo, armazena arquivos JSON responsáveis pela tradução de elementos HTML contidos nos *templates* das páginas, através da *tag data-transalt*, relacionando uma chave (elemento HTML), a um valor que é a referida tradução.

Figura 16 - Diretório i18n



Fonte: Autores

A figura a seguir exemplifica o arquivo de tradução.

Figura 17 - Arquivo de tradução

```
{
  "yoloApp": {
    "animal": {
      "home": {
        "title": "Animais",
        "createLabel": "Criar novo Animal",
        "createOrEditLabel": "Criar ou editar Animal",
        "search": "Pesquisar por Animal"
      },
      "created": "Um novo Animal foi criado com o identificador {{ param }}",
      "updated": "Um Animal foi atualizado com o identificador {{ param }}",
      "deleted": "Um Animal foi deletado com o identificador {{ param }}",
      "delete": {
        "question": "Tem certeza de que deseja excluir o Animal {{ id }}?"
      },
      "detail": {
        "title": "Animal"
      },
      "nome": "Nome",
      "apelido": "Apelido",
      "sexo": "Sexo",
      "nascimento": "Nascimento",
      "chip": "Chip",
      "cor": "Cor",
      "fotoAnimal": "Foto do Animal",
      "atendimento": "Atendimento",
      "animal": "Animal",
      "pai": "Pai",
      "mae": "Mae",
      "pessoa": "Pessoa",
      "endereco": "Endereco",
      "especie": "Especie"
    }
  }
}
```

Fonte: Autores

### **3.8 Licença de software**

O framework Jhipster utiliza a Licença Apache Versão 2.0, que é uma licença de software permissiva no modelo FOSS (Free and Open-Source Software) e pode ser classificada com ambas licenças de software livre e software open source, atribuindo liberdade para utilizar, copiar, estudar e modificar o software de qualquer maneira, e o código fonte é abertamente compartilhado para que sejam encorajadas as melhorias voluntárias no design do software. Por extensão, baseado no framework utilizado para o desenvolvimento, o software RGA – Registro Geral de Animais utiliza a Licença Apache Versão 2.0, disponível no endereço <http://www.apache.org/licenses/LICENSE-2.0>.

## 4 CONSIDERAÇÕES FINAIS

O presente trabalho obteve êxito em implementar os requisitos levantados para usabilidade do sistema, permitindo o registro dos animais, dos proprietários e do histórico de vacinas. Cada animal é vinculado a apenas um proprietário, o que permite atribuir-lhes as responsabilidades previstas para a posse ou guarda de animais de companhia. Foi possível também criar uma base de dados com o histórico de vacinas, onde cada animal pode ser vinculado a múltiplas doses de vacina, proporcionando informações que permitam as equipes de vigilância em saúde identificar animais que possam ser potenciais portadores de doenças zoonóticas.

Foram abordados e referenciados os temas pertinentes ao assunto, com ênfase no controle populacional de animais, controle de zoonoses, bem-estar animal e guarda responsável e a legislação brasileira existente, bem como o estudo de casos de programas de registro e identificação de animais em execução no exterior.

A utilização das tecnologias voltadas ao desenvolvimento web citadas no referente trabalho, propiciaram uma rápida curva de implementação, maior controle e organização do código e maior facilidade de manutenção.

A criação de uma aplicação web possibilitou maior disponibilidade de acesso, através da internet, e graças ao seu design responsivo, a viabilidade e ergonomia em dispositivos móveis. Também proporcionou um menor consumo de recursos de hardware, por ter uma infraestrutura centralizada e escalável.

Este sistema buscou se utilizar das tecnologias mais atuais e modernas, mantendo o foco em reusabilidade, ergonomia e acessibilidade, baseado nas deficiências desses elementos e componentes em sistemas existentes e na carência de sistemas desse porte, segundo a pesquisa bibliográfica.

Como todo software, este também, neste momento, não pode atender a todas necessidades que serão encontradas em seu período inicial de utilização. Porém, como a construção de software e a metodologia utilizada, nos confere a capacidade de melhoria, poderemos incrementar as funcionalidades necessárias e a adequação as legislações previstas na sua área de localização.

Para trabalhos futuros, pretende-se ampliar as funcionalidades existentes, como a implementação de módulo para intermediar a adoção de animais, ferramentas para gerenciamento de informações, como relatórios de dados estatísticos e remodelagem da interface visual.



## REFERÊNCIAS

AMARA, Renata Maria Albergaria. Bem-estar de cães e gatos. **Cadernos Técnicos de Veterinária e Zootecnia**, Belo Horizonte, v. 67, p.42-50, dez. 2012.

ARRONDISSEMENT DE VERDUN – **Réglement Sur Les Animaux**. Disponível em [http://ville.montreal.qc.ca/pls/portal/docs/PAGE/ARR\\_VER\\_FR/MEDIA/DOCUMENTS/RCA11\\_2100011.PDF](http://ville.montreal.qc.ca/pls/portal/docs/PAGE/ARR_VER_FR/MEDIA/DOCUMENTS/RCA11_2100011.PDF) Acesso em 02 mai. 2017.

BHARATHAN, Raghuram. **Apache Maven Cookbook**: Over 90 hands-on recipes to successfully build and automate development life cycle tasks following Maven conventions and best practices. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing Ltd., 2015. 272 p.

BOWER. **Bower: A package manager for the web**. 2017. Disponível em: <<https://bower.io/>>. Acesso em: 13 maio 2017.

BRASIL. Ministério da Saúde. **Casos de raiva humana por espécie agressora**, Brasil, 1986-2009. Comunicação feita pela Secretaria de Vigilância em Saúde. 2009a.

BRASIL. Ministério da Saúde. **Raiva humana Brasil**, 1986-2009. Comunicação feita pela Secretaria de Vigilância em Saúde. 2009c.

BRASIL. Ministério da Saúde. **Situação da raiva no Brasil**. Comunicação feita pela Secretaria de Vigilância em Saúde. 2009b.

BROOM, D.M. Animal Welfare: Concepts And Measurement. **Journal of Animal Science**, Savoy, v.69, 1991a.

CARDOSO NETO, Nicolau; RIMES, Sheila Olwen; SOBERÓN, Ruben. Proposta de modelo de sistema de registros de população de animais domésticos: comparação entre normas jurídicas brasileira e estrangeiras. **Cadernos Ibero-americanos de Direito Sanitário**, [s.l.], v. 5, n. 3, p.31-43, 30 set. 2016.

CITY OF ANDERSON – **Animal Control**. Disponível em <http://www.cityofhenderson.com/animal-control/home> Acesso em 02 de mai. De 2017.

FARIA, Thiago; NORMANDES JUNIOR,. **JPA e Hibernate**. Uberlândia - MG: Algaworks, 2015. 176 p.

FLORIANO, Francisco. PROJETO DE LEI N.º 4.564, DE 2016. 2016. Disponível em: <<http://www.camara.gov.br/sileg/integras/1441364.pdf>>. Acesso em: 30 jun. 2017.

FREEMAN, Adam. **The Definitive Guide to HTML5**. Springer Science Business Media, LLC., 233 Spring Street, 6th Floor, New York, Ny 10013: Apress, 2011. 1052 p.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2008.

GOLDSTEIN, Alexis; LAZARIS, Louis & WEYL, Estelle **HTML5 & CSS3 for the Real World**, SitePoint Pty. Ltd, 2011.

GOOGLE. **How does AngularJS support i18n/l10n?** 2010. Disponível em: <<https://docs.angularjs.org/guide/i18n>>. Acesso em: 08 maio 2017.

GOSLING, J. et al.: **The Java Language Specification, Java SE 8 Edition** (2015). Disponível em: <<http://docs.oracle.com/javase/specs/jvms/se8/html/index.html>><<http://docs.oracle.com/javase/specs/jls/se8/html/index.html>>. Acesso em: 20 de março de 2017.

GUTIERREZ, Felipe. **Pro Spring Boot**. New York, 233 Spring Street, 6th Floor, New York, Ny 10013: Springer Science, 2016. 373 p.

HERRON, David. **Node Web Development**. 2. ed. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing, 2013. 248 p.

IKKINK, Hubert Klein. **Gradle Effective Implementation Guide: Empower yourself to automate your build**. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing Ltd., 2012. 382 p.

INTERNATIONAL COMPANION ANIMAL MANAGEMENT COALITION - ICAM . **Humane dog population management guidance**. 2007. Disponível em [http://www.icam-coalition.org/downloads/Humane\\_Dog\\_Population\\_Management\\_Guidance\\_Portuguese.pdf](http://www.icam-coalition.org/downloads/Humane_Dog_Population_Management_Guidance_Portuguese.pdf) Acesso em 2 de maio de 2017.

JHIPSTER. **JHipster Domain Language (JDL)**. 2016. Disponível em: <<https://jhipster.github.io/jdl/>>. Acesso em: 08 maio 2017.

KAYAL, Dhrubojyoti. **Pro Java™ EE Spring Patterns: Best Practices and Design Strategies Implementing Java™ EE Patterns with the Spring Framework**. New York, Inc., 233 Spring Street, 6th Floor, New York, Ny 10013: Springer, 2008. 336 p.

KROL, Jason. **Web Development with MongoDB and Node.js**. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing, 2014. 294 p.

LALOU, Jonathan. **Apache Maven Dependency Management: Manage your Java and JEE project dependencies with ease with this hands-on guide to Maven**. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing Ltd., 2013. 158 p.

LANS, Rick F. van Der. **SQL for MySQL Developers: A Comprehensive Tutorial and Reference**. 75 Arlington Street, Suite 300 Boston, Ma: Pearson Education, 2007. 1031 p.

LINDHOLM, Tim et al.: **The Java Virtual Machine Specification**, Java SE 8 Edition (2015). Disponível em:

<<http://docs.oracle.com/javase/specs/jvms/se8/html/index.html>>. Acesso em: 20 de março de 2017.

LIQUIBASE. **Major Concepts**. 2017. Disponível em: <<http://www.liquibase.org/documentation/index.html>>. Acesso em: 02 maio 2017.

LUCKOW, Décio Heinzemann; MELO, Alexandre Altair. **PROGRAMAÇÃO JAVA PARA WEB: APRENDA A DESENVOLVER UMA APLICAÇÃO FINANCEIRA DO ZERO**. 2. ed. São Paulo, Sp: Novatec, 2015. 640 p.

MAYNARD, Travis. **Getting Started with Gulp**. Livery Place, 35 Livery Street, Birmingham B3 2pb, Uk. Isbn 978-1-78439-576-6: Packt Publishing Ltd., 2015. 251 p.

MCLAUGHLIN, Michael. **MySQL Workbench: Data Modeling and Development: Visually Model, Generate and Maning Databases**. New York: Mcgraw-hill, 2013. 793 p.

MITRA, Mainak. **Mastering Gradle: Master the technique of developing, migrating, and building automation using Gradle**. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing Ltd., 2015. 285 p.

MOUTINHO, Flavio Fernando Batista; NASCIMENTO, Elmiro Rosendo do; PAIXÃO, Rita Leal. **PERCEPÇÃO DA SOCIEDADE SOBRE A QUALIDADE DE VIDA E O CONTROLE POPULACIONAL DE CÃES NÃO DOMICILIADOS**. *Ciência Animal Brasileira*, [s.l.], v. 16, n. 4, p.574-588, dez. 2015.

ORGANIZAÇÃO MUNDIAL DA SAÚDE - OMS. **Joint FAO/WHO expert committee on zoonoses**. Third Report. Geneva: WHO, 1967. 127 p.

ORGANIZAÇÃO MUNDIAL DA SAÚDE - OMS. **Zoonoses**, 2015. Disponível em: <http://www.who.int/topics/zoonoses/en/>. Acesso em 02 mai. 2017.

ORGANIZAÇÃO MUNDIAL DA SAÚDE - OMS. **Guidelines for dog population management**. Genebra: Who, 1990. 116 p.

ORGANIZAÇÃO MUNDIAL PARA A SAÚDE ANIMAL - OIE. **Terrestrial Animal Health Code**, 2008. Disponível em: [http://www.oie.int/eng/normes/mcode/en\\_glossaire.htm#sous-chapitre-2](http://www.oie.int/eng/normes/mcode/en_glossaire.htm#sous-chapitre-2)

OTTINGER, Joseph B.; LINWOOD, Jeff; MINTER, Dave. **Beginning Hibernate: For Hibernate 5**. 4. ed. New York, 233 Spring Street, 6th Floor, New York, Ny 10013: Springer Science, 2016. 234 p.

PACHEV, Sasha. **Understanding MySQL Internals**. 1005 Gravenstein Highway North, Sebastopol, Ca: O'reilly Media, Inc., 2007.

PANDA, Sandeep. **AngularJS: Novice to Ninja**. Cambridge Street Collingwood Vic Australia 3066: Sitepoint Pty. Ltd., 2014. 304 p.

PASQUILI, Sandro. **Mastering Node.js**. Livery Place 35 Livery Street Birmingham B3 2pb, Uk: Packt Publishing, 2013. 346 p.

RAIBLE, Matt. **The JHipster Mini-Book**. United States: C4media, 2015. 131 p.

SCHMITZ, Daniel; LIRA, Douglas. **AngularJS na prática**. Victoria, British Columbia, Canada: Leanpub, 2016. 126 p.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. rev. atual. Florianópolis: UFSC, 2005. 138 p.

SILVANO, D et al. DIVULGAÇÃO DOS PRINCÍPIOS DA GUARDA RESPONSÁVEL: UMA VERTENTE POSSÍVEL NO TRABALHO DE PESQUISA. **Revista Eletrônica Novo Enfoque**, ano 2010, v. 09, n. 09, p. 64 – 86, disponível em: <http://www.castelobranco.br/sistema/novoenfoco/files/09/artigos/06.pdf>. Acesso em: 08 de maio de 2017.

SLATER, M. R.; DI NARDO, A.; PEDICONE, O.; VILLA, P. D.; CANDELORO, L.; ALESSANDRINI, B.; DEL PAPA, S. Cat and dog ownership and management patterns in central Italy. **Preventive Veterinary Medicine**, Amsterdam, v. 85, n. 3-4, 2008.

Somerville, Ian. **Software Engineering**. 9.. ed. Harlow, England: Addison-Wesley, 2010.

SPURLOCK, Jake. **Bootstrap – Responsive Web Development**. 1.ed. Sebastopol: O'Reilly Media, 2013.

TWITTER. Bootstrap. Disponível em: <<https://bootstrapdocs.com/v3.3.6/docs/>>. Acesso em: 05 maio 2017.

UNESCO. Universal Declaration of Animal Rights. 1978. Disponível em: <<http://www.esdaw.eu/unesco.html>>. Acesso em: 30 jun. 2017.

UNIÃO INTERNACIONAL PROTETORA DOS ANIMAIS - UIPA. **Guarda Responsável**. 2014. Disponível em: <<http://www.uipa.org.br/guarda-responsavel/>>. Acesso em: 02 maio 2017.

VIEIRA, A. M. L.; ALMEIDA, A. B.; MAGNABOSCO, C.; FERREIRA, J. C. P.; CARVALHO, J. L. B.; GOMES, L. H.; PARANHOS, N. T.; REICHMANN, M. L.; GARCIA; R. C.; LUNA, S. L. P.; NUNES, V. F. P.; CABRAL, V. B. Programa de controle de cães e gatos do Estado de São Paulo. **Boletim Epidemiológico Paulista**, São Paulo, v.3, suppl. 5, p. 1-139, 2006. Disponível em: [ftp://ftp.cve.saude.sp.gov.br/doc\\_tec/outros/suple5\\_cao.pdf](ftp://ftp.cve.saude.sp.gov.br/doc_tec/outros/suple5_cao.pdf) Acesso em 02 mai. 2017.

WAZLAWICK, Raul Sidnei. 2014. **Metodologia de Pesquisa para Ciência da Computação**. Elsevier, 146p.

WEST, Matt. **HTML5 Foundations**. The Atrium, Southern Gate, Chichester, West Sussex, Po19 8sq, United Kingdom: John Wiley & Sons Ltd, 2013. 386 p.

# ANEXO

## Anexo 1 – Modelo relacional do banco de dados

