



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR DE ANGICOS
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLOGIA DA INFORMAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALESCIA BEZERRA FERNANDES

**CENTRAL DO PET: UMA PLATAFORMA WEB PARA UNIFICAR E DIVULGAR
INFORMAÇÕES SOBRE CÃES E GATOS**

ANGICOS – RN

2018

ALESCIA BEZERRA FERNANDES

**CENTRAL DO PET: UMA PLATAFORMA WEB PARA UNIFICAR E DIVULGAR
INFORMAÇÕES SOBRE CÃES E GATOS**

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof^ª. M.^a Luana Dantas Chagas

ANGICOS – RN

2018

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

F363c Fernandes, Alescía Bezerra.
CENTRAL DO PET: UMA PLATAFORMA WEB PARA
UNIFICAR E DIVULGAR INFORMAÇÕES SOBRE CÃES E GATOS
/ Alescía Bezerra Fernandes. - 2018.
44 f. : il.

Orientadora: Luana Dantas Chagas.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Sistemas de
Informação, 2018.

1. Central do pet. 2. Adoção de animais. 3.
Ferramenta web. I. Chagas, Luana Dantas, orient.
II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

ALESCIA BEZERRA FERNANDES

**CENTRAL DO PET: UMA PLATAFORMA *WEB* PARA UNIFICAR E DIVULGAR
INFORMAÇÕES SOBRE CÃES E GATOS**

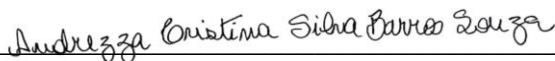
Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

APROVADA EM: 20/09/2018

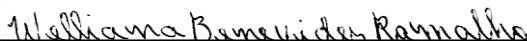
BANCA EXAMINADORA



Prof.^a M.^a Luana Dantas Chagas (UFERSA)
Presidente



Prof.^a M.^a Andreza Cristina da Silva Barros Souza (UFERSA)
Membro Examinador



Prof.^a M.^a Welliana Benevides Ramalho (UFERSA)
Membro Examinador

RESUMO

O abandono de animais é um problema cotidiano: além de ser um ato cruel, pode trazer complicações à sociedade. Ao serem abandonados, os animais passam a viver em más condições, com medo, passando fome, sede, ficando vulneráveis a maus-tratos e doenças, podendo inclusive arriscar a qualidade da saúde pública. Tendo em vista esse cenário, o objetivo desta monografia é criar uma ferramenta *web* gratuita de colaboração sobre animais em situação de abandono, de modo a agregar informações sobre esses animais e facilitar a comunicação sobre questões como doações, adoções e divulgação de campanhas/eventos, tudo isso em prol da diminuição do número de animais abandonados. Para tanto, foi realizado um levantamento de características sobre os animais que a ferramenta deveria abranger e sobre as tecnologias e ferramentas já existentes, considerando seus pontos fortes e fracos. A partir disso, foi desenvolvido a Central do Pet, que consiste em um ambiente de colaboração para adotantes, ONGs, CCZs, entidades públicas e qualquer outro que demonstre interesse sobre o abandono de animais. Ao realizar o estudo foi possível perceber que o abandono de animais não é um problema que se resolverá rapidamente e espera-se que a Central do Pet atue como meio para diminuir os índices de abandono. Futuramente poderão ser acrescentadas funcionalidades para abranger outros campos de interesse aos donos de animais, como serviços veterinários, informações sobre cuidados com animais, localização de serviços de pet shop e hotelaria, entre outros que se julguem interessantes para o público alvo.

Palavras-chave: Central do Pet. Adoção de animais. Ferramenta web.

ABSTRACT

The abandonment of animals is an everyday problem: besides being a cruel act, it can bring complications to society. When they are abandoned, the animals begin to live in poor conditions, with fear, going hungry, thirsty, becoming vulnerable to maltreatment and diseases, and even risking the quality of public health. In view of this scenario, the objective of this monograph is to create a free collaborative web tool about abandoned animals, in order to aggregate information about these animals and to facilitate communication on issues such as donations, adoption and dissemination of campaigns / events, all in favor of reducing the number of abandoned animals. To do so, a survey was carried out on the characteristics of the animals that the tool should cover and on the already existing technologies and tools, considering their strengths and weaknesses. From this, the Central do Pet was developed, which consists of a collaborative environment for adopters, NGOs, ZCCs, public entities and any other that shows interest in the abandonment of animals. In the study, it was possible to perceive that the abandonment of animals is not a problem that will be resolved quickly and the Central do Pet is expected to act as a device to decrease the rates of abandonment. In the future, functionalities may be added to cover other areas of interest to pet owners, such as veterinary services, information on animal care, location of pet shop and hotel services, among others that may be of interest to the target audience.

Keywords: Central do Pet. Pets adoption. Web tool.

LISTA DE FIGURAS

Figura 1 – Fases da vida de cães e gatos.....	14
Figura 2 – Combinações de cores de pelos mais comuns em cães e gatos	15
Figura 3 – Estrutura cliente-servidor	16
Figura 4 – Fluxo da metodologia de desenvolvimento do projeto	21
Figura 5 – Diagrama de Caso de Uso – conta	24
Figura 6 – Diagrama de Caso de Uso – sistema	24
Figura 7 – Método POST da aplicação	26
Figura 8 – Conexão do banco de dados	27
Figura 9 – Estruturas dos arquivos do mongoDB.....	28
Figura 10 – Exemplo de busca no banco de dados.....	28
Figura 11 – Comparativo entre Angular e JQuery.....	29

LISTA DE TABELAS

Tabela 1 – Definição de porte de acordo com o peso do animal.....	13
---	----

LISTA DE ABREVIATURAS E SIGLAS

ANDA	Agência de Notícias de Direitos Animais
CCZ	Centro de Controle de Zoonoses
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hyper Text Markup Language</i>
IHC	Interação Humano Computador
OMS	Organização Mundial da Saúde
ONG	Organização Não Governamental
UIPA	União Internacional Protetora dos Animais
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1 INTRODUÇÃO	09
1.1 JUSTIFICATIVA	10
1.2 OBJETIVOS	11
1.2.1 Objetivo geral	11
1.2.2 Objetivos específicos	12
2 REFERENCIAL TEÓRICO	13
2.1 CARACTERÍSTICAS DOS ANIMAIS	13
2.2 DESENVOLVIMENTO <i>WEB</i>	15
3 CENTRAL DO PET	21
3.1 DESCOBERTA DE REQUISITOS	22
3.2 ELABORAÇÃO DE CASOS DE USO	23
3.3 ELABORAÇÃO DO <i>DESIGN</i> DE TELAS	25
3.4 DESENVOLVIMENTO DAS FUNCIONALIDADES	26
4 CONSIDERAÇÕES FINAIS	30
BIBLIOGRAFIA	31
APÊNDICE A – Requisitos do sistema	35
APÊNDICE B – Design de telas	41

1 INTRODUÇÃO

O abandono de animais é um problema cotidiano em nosso país: basta andar um pouco pelas ruas de qualquer cidade para se deparar com cães e gatos abandonados, muitas vezes doentes ou machucados. Segundo uma estimativa realizada pela Organização Mundial da Saúde (OMS), no Brasil, no ano de 2015, já existiam 30 milhões de animais abandonados, sendo em torno de 10 milhões de gatos e 20 milhões de cães (ANDA, 2015). Essa situação gera preocupações pois, sem controle, os animais se reproduzem muito rápido, criando assim superpopulação de animais abandonados, que ficam à mercê de doenças e maus tratos. Além disso, pode-se gerar ambientes propícios à propagação de doenças, inclusive para os seres humanos, as chamadas zoonoses.

No intuito de lidar com algumas dessas questões, há os Centros de Controle de Zoonoses (CCZ). Tratam-se de unidades de saúde pública que foram criadas com o objetivo de controlar e prevenir as zoonoses, adotando medidas de controle de população desses animais por meio da prestação de serviços como esterilização cirúrgica (castração) e tomando providências a respeito de animais infectados (tratamentos ou até mesmo abates), além de fornecer trabalhos educativos à população acerca do tema.

Nesse mesmo contexto, há também Organizações Não Governamentais (ONG) em prol da defesa dos animais que buscam, em sua maioria, resgatar e cuidar dos animais de rua, fornecendo alimento, abrigo e cuidados médicos, até que apareça algum adotante. Os recursos para arcar com estes gastos têm origem, em sua grande parte, nas doações da população, seja em forma de rações, medicamentos ou dinheiro.

Apesar do esforço dessas entidades, é possível verificar um cenário onde existem muito mais chamados do que conseguem atender. Em Shimozako *et al.* (2006), é feito um levantamento das chamadas recebidas e atendidas pelo CCZ de São Paulo em um intervalo de 33 meses (de janeiro de 2001 a setembro de 2003) e os dados encontrados mostram uma média aproximada de 1.539 chamados recebidos e 232 atendidos por mês. Sendo assim, os chamados atendidos correspondem a um pouco mais de 15% do total de chamados recebidos.

As ONGs também enfrentam situações desfavoráveis: superlotação e dificuldade de arcar com os custos de cuidar dos animais acolhidos são problemas recorrentes. Segundo relatos da ONG Quintal de São Francisco (GOMES, 2013), a superpopulação de animais abandonados gera superlotação nas ONGs, que ficam impossibilitadas de resgatar mais animais. Um exemplo dessa situação é que, no ano de 2011, a ONG Clube dos Vira Latas

abrigou 500 cães, quando sua capacidade era de 300, e a ONG Humanimal, que abrigou 100 cães, quando sua capacidade era de apenas 30 (MONTEIRO, 2011).

Tendo em vista este cenário, é possível desenvolver ferramentas que auxiliem na divulgação dos animais disponíveis para adoção, criando um ambiente propício também para divulgação de campanhas de vacinação, castração e diversas outras ações em prol dos animais abandonados. Assim, esse trabalho propõe o desenvolvimento de uma ferramenta *web*, denominada Central do Pet, para atender CCZs, ONGs, adotantes e voluntários de forma gratuita. Para os CCZs e as ONGs, o objetivo da ferramenta será principalmente possibilitar a divulgação de campanhas como vacinação, feiras de adoção, entre outras. Já para os adotantes e voluntários, será uma central de informações sobre os animais e como ajudá-los.

1.1 JUSTIFICATIVA

Abandono e maus-tratos de animais domésticos são considerados crimes, previstos no artigo 164 da Lei nº 2.848 de 07 de dezembro de 1940 (BRASIL, 1940), que trata do abandono de animais, com pena entre 15 dias a 6 meses, ou aplicação de multa, e na Lei nº 9.605 de 12 de fevereiro de 1998 (BRASIL, 1998), que trata das sanções penais e administrativas derivadas de condutas e atividades lesivas ao meio ambiente.

Apesar da existência dessas legislações, no Brasil, no ano de 2015 já haviam milhões de animais abandonados nas ruas. Segundo a Agência de Notícias de Direitos Animais (ANDA), nas grandes cidades do país, para cada cinco habitantes há um cachorro (ANDA, 2015), destes, 10% estão abandonados. No interior podemos encontrar números similares. Em Araçatuba – SP, por exemplo, são mais de 35 mil animais, destes 2,6 mil estão abandonados, aproximadamente 7,43% do total (ANDA, 2015). A cidade de Marília – SP conta com mais de 60 mil animais e a estimativa é de quase três mil em situação de abandono, que representa 5% do total (ANDA, 2015).

Este cenário contribui também para a propagação de várias doenças entre os animais de rua. Em especial podemos citar as zoonoses, que são doenças transmissíveis dos animais para os seres humanos, como a raiva, sarnas, bicho geográfico, toxoplasmose, leishmaniose, entre outras (MUTIIS, 2013).

Para tentar amenizar este cenário, várias ONGs e CCZs criaram sites para divulgar os animais em disponibilidade para adoção, porém muitos estão inacabados ou não disponibilizam informações completas a respeito dos animais. Na União Internacional Protetora dos Animais (UIPA), a ONG mais antiga do Brasil, o processo de adoção é apenas

presencial, não havendo compartilhamento das informações sobre eles online. Os animais do Instituto Luisa Mell, uma das ONGs mais bem-conceituadas e entre as 100 melhores do país (BARBOZA, 2017), são adotados somente em eventos, não havendo também o compartilhamento das informações. Já a Rockbicho.org apresenta problemas de otimização: todas as imagens dos animais são carregadas de uma única vez, acarretando em um carregamento lento, além da dificuldade de encontrar um animal em uma cidade em específico, sendo necessário clicar em cada anúncio e verificar a cidade. A página Cãopanhiaero Curitiba deixa a desejar em relação as informações divulgadas sobre os animais: muitos não possuem informações sobre castração, vermifugação e idade. E por fim, a Ampara Animal, uma das maiores do país, apesar de ter um sistema robusto de adoção e compartilhamento de informações, atua somente nas cidades de São Paulo e Rio de Janeiro. Essa regionalização foi constatada em todos os sites analisados, não sendo encontrado um sistema de nível nacional.

Outro desafio, considerando as ferramentas já existentes, está na quantificação dos dados: como cada organização desenvolveu seu próprio mecanismo, é inviável saber quantidades exatas de animais disponíveis nas ONGs e CCZs, pois para conseguir tais números seria necessário entrar em contato com cada uma das organizações e fazer a solicitação dos dados.

Levando em consideração as informações aqui apresentadas, este projeto oferecerá uma ferramenta de colaboração sobre animais em situação de abandono, de modo a agregar informações e facilitar a comunicação sobre questões como doações, adoções e divulgação de campanhas/eventos. Ademais, o projeto ainda contribuirá com a unificação, padronização e divulgação dos dados sobre animais disponíveis para adoção, visto que nenhuma dessas tarefas é bem atendida por nenhuma ferramenta disponível atualmente.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Desenvolvimento de um sistema *web* para atuar como um ambiente de colaboração sobre animais em situação de abandono, de modo a agregar informações e facilitar a comunicação sobre questões como adoções, doações e divulgação de campanhas/eventos.

1.2.2 Objetivos específicos

1. Elicitar os requisitos por meio de análise de ferramentas similares já existentes e informações disponíveis em sites de ONGs e CCZs.
2. Elaborar diagramas de casos de uso para cada funcionalidade do sistema (seguindo as normas definidas na UML - *Unified Modeling Language*).
3. Integrar o sistema a um banco de dados noSql.
4. Desenvolver as funcionalidades do sistema.
5. Criação de *designs* interativos e simplificados para as telas do sistema, respeitando fundamentos da IHC - Interação Humano-Computador.
6. Disponibilizar a ferramenta para uso da comunidade, especialmente para as ONGs, CCZs ou entidades similares, como secretarias de saúde.

2 REFERENCIAL TEÓRICO

Homens e animais compartilham uma relação que já dura milhares de anos, sendo possível encontrar vestígios dessa relação em pinturas de cavernas, seja ajudando em atividades de transporte e agricultura ou para proteger moradias. Nos dias de hoje, esse vínculo permanece com grande força: apesar de os animais ainda serem usados para auxiliar em tarefas de transporte e agricultura, animais como cães, gatos e pássaros ganharam um lugar especial no nosso cotidiano. Muitas pessoas veem nesses animais a companhia ideal, especialmente as que moram sozinhas. Vale ressaltar também que a companhia de um bichinho traz benefícios à saúde, como alívio de estresse e combate à depressão (SCHOLZE, 2013) e, apesar dos benefícios dessa relação, o abandono de animais é um problema constante e milhares de animais estão à espera de um novo lar em ONGs e CCZs em todo país.

2.1 CARACTERÍSTICAS DOS ANIMAIS

Para ocorrer a adoção dos animais disponíveis nos CCZs e ONGs é necessário apresentar algumas características desses animais para os possíveis adotantes, como porte, idade, cor da pelagem e outras características relativas à saúde e segurança do animal, como castração, vermifugação e microchipagem, pois é com base nelas que o processo de adoção é efetivado ou não pelo adotante. No que diz respeito ao porte dos animais, não foram encontrados estudos ou documentos oficiais que padronizem uma classificação. Sendo assim, serão utilizadas as definições do Turismo4Patas, que são apresentadas na Tabela 1.

Tabela 1 - Definição de porte de acordo com o peso do animal

Classificação	Peso
Mini ou Toy	até 6 kg
Pequeno ou Anão	de 6,001 a 15 kg
Médio	de 15,001 a 25 kg
Grande	de 25,001 a 45 kg
Gigante	de 45,001 kg acima
Fonte: Turismo4Patas (2014)	

Outra característica passível de subjetividade são as fases da vida desses animais e o tempo que cada uma dura, por esse motivo foi utilizada a classificação do portal BayerPet. Nele, as fases da vida de cães e gatos podem ser divididas em quatro estágios: filhote, adolescente, adulto e terceira idade. Esses estágios são mostrados na Figura 1.

Figura 1 - Fases da vida de cães e gatos



Fonte: Portal Bayer.net (2018)

Uma das características mais notórias em um animal é a cor de seus pelos. Cães e gatos possuem padrões de cores de pelos muito parecidos, pois ambos variam entre preto, branco, cinza e laranja/caramelo. As combinações de cores de pelo mais comuns são ilustradas na Figura 2.

Em relação à saúde do animal, dois aspectos são relevantes para o sistema: se o animal é castrado e se o animal é vermifugado. A castração consiste em um procedimento cirúrgico cujo objetivo é a esterilização do animal. Já a vermifugação consiste no ato de medicar os animais contra vermes, sendo necessária nova dosagem a cada 4 ou 6 meses, de acordo com o médico veterinário Marcelo Quinzani (SONCELA, 2016).

Em relação ao monitoramento dos animais, a estratégia mais efetiva atualmente é a microchipagem, que consiste na inserção de um microchip, do tamanho de um grão de arroz, no corpo do animal. Nesse microchip é registrado um número único e intransferível, que estará cadastrado em um banco de dados, onde as informações sobre aquele animal são armazenadas. Desse modo é possível fazer um monitoramento minucioso sobre o animal, abrangendo idade, informações sobre o dono, localização e outras informações pertinentes. A

microchipagem também é um ótimo método para prevenção do abandono desses animais, sendo que muitas vezes os animais que são acolhidos por abrigos são animais que se perderam, e o uso do microchip possibilita a localização de seus donos. Hoje muitos CCZs já realizam a microchipagem nos animais que são recolhidos, por isso é uma informação pertinente para o processo de adoção.

Figura 2 - Combinações de cores de pelos mais comuns em cães e gatos

Cães	Gatos
 Cinza	 Cinza
 Preto	 Preto
 Branco	 Branco
 Caramelo	 Laranja
 Preto e Branco	 Preto e Branco
 Caramelo e Branco	 Laranja rajado com Branco
 Caramelo e Preto	 Cinza rajado com Branco
 Tricolor	 Tricolor
 Pintas	 Siamês

Fonte: Autoria própria (2018)

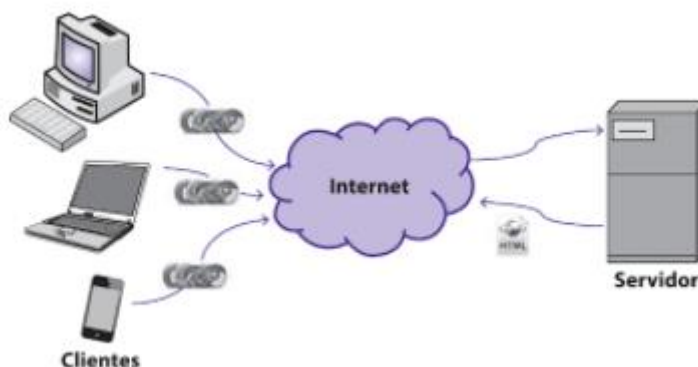
2.2 DESENVOLVIMENTO WEB

Desenvolver para a *web* consiste em criar sistemas que serão executados de forma remota através do navegador do usuário. O bom desenvolvimento desses sistemas requer conhecimento de fundamentos teóricos e tecnologias utilizados. Esses recursos são utilizados em diversas tarefas do desenvolvimento de projetos, como facilitar a comunicação do desenvolvedor com o usuário, estabelecer diretrizes para a criação de um ambiente de navegação agradável e de fácil aprendizado para o usuário ou criar documentos que servirão de base para os desenvolvedores criarem as funcionalidades do sistema. Podemos citar as

definições e características de sistemas *web*, a UML - *Unified Modeling Language*, e a IHC - Interação Humano-Computador como fundamentos para esse tipo de desenvolvimento.

Um sistema *web* é caracterizado como um sistema onde o processamento é feito de forma distribuída: parte do processamento é executado na máquina do cliente e parte é executado em outra máquina, chamada servidor. Essa estrutura é denominada estrutura cliente-servidor, e é ilustrada na Figura 3. A máquina cliente executa o sistema por meio de um navegador, fazendo com que a instalação direta do *software* no computador do usuário se torne desnecessária, e o acesso ocorre de forma remota através da Internet ou da intranet (MILETTO; BERTAGNOLLI, 2014).

Figura 3 - Estrutura cliente-servidor



Fonte: MILETTO; BERTAGNOLLI (2014)

É possível elencar 4 principais áreas no desenvolvimento:

- Codificação no cliente, que consiste na codificação da interface do usuário, abrangendo criação de botões, criação de formulários, envio dos campos preenchidos ao servidor e diversos outros mecanismos que atuam diretamente com o usuário, essa codificação é também chamada de *front-end*;
- Codificação no servidor, que consiste na codificação das funcionalidades oferecidas pelo sistema, como tratamento e modificação dos dados recebidos, conexão com o banco de dados, entre outras funções que são executadas de forma oculta ao usuário, essa codificação é também chamada de *back-end*;
- Bancos de dados, que são conjuntos de arquivos relacionados entre si e onde os registros gerados no sistema são armazenados;

- Áreas interdisciplinares que podem, de alguma forma, contribuir para o funcionamento do sistema, como *design* gráfico para criação do *design* das telas e regras de acessibilidade adicionadas à interface do usuário como forma de inclusão digital para minorias.

Para modelar um sistema *web* é necessária a criação e análise das estruturas e comportamentos que o sistema apresentará e para representá-las podemos utilizar a UML. UML - *Unified Modeling Language* (em português, Linguagem de Modelagem Unificada) - é a linguagem-padrão para a elaboração da estrutura de projetos de *software* e é essencial para um bom entendimento do sistema a ser desenvolvido. Seu principal objetivo é facilitar e padronizar a comunicação dos aspectos técnicos que serão executados no projeto de *software* para o usuário-cliente (que é o usuário que solicitou o desenvolvimento do sistema) e para a equipe de desenvolvimento, uma vez que seus diagramas são facilmente compreendidos pois a UML emprega recursos visuais para representar fluxos e comportamentos.

Os diagramas da UML podem ser divididos em três grupos: diagramas estruturais, utilizados nas representações das estruturas relativamente estáveis do sistema, como classes, objetos e pacotes; diagramas comportamentais, utilizados nas representações dos comportamentos do sistema, como os casos de uso e estados do sistema; e diagramas de interação, utilizados nas representações interativas do sistema, como sequências e comunicações estabelecidas dentro do sistema (RUMBAUGH; JACOBSON; BOOCH, 2010).

Um sistema *web* deve também apresentar uma interface que ofereça boa comunicação com seu usuário, sem confundi-lo ou tornando o uso do sistema difícil. O estudo desses aspectos interativos com o usuário é tratado pela IHC - Interação Humano-Computador - que é uma área que estuda como ocorrem as interações humanas com os sistemas computacionais. Uma vez que seus estudos fornecem uma forte base em quesito de usabilidade dos sistemas desenvolvidos, torna-se necessária a correta aplicação dos mesmos. Como exemplo, pode-se citar estudos sobre qualidade de uso e *affordances*.

A qualidade de uso possui 4 critérios:

- Usabilidade, que é definida por Nielsen (1993) como um conjunto de fatores compreendidos em facilidade de aprendizado, que trata do quão fácil é o usuário aprender a utilizar o sistema; facilidade de recordação, que consiste na capacidade dos usuários de reutilizar o sistema, mesmo após um longo período de tempo; eficiência, que trata da eficiência do sistema sendo que este deve apresentar altos níveis de produtividade; segurança no uso, que trata da quantidade de erros apresentada pelo sistema, que deve ser o menor

possível e erros graves ou sem solução não devem ocorrer; e satisfação do usuário, que consiste na satisfação do usuário em utilizar o sistema;

- Experiência do usuário, que é avaliada de acordo com os sentimentos que o usuário desenvolve ao utilizar o sistema, envolvendo aspectos positivos e negativos, como satisfação, entretenimento, cansaço ou frustração, gerando assim uma experiência positiva ou negativa (NIELSEN, 1993);

- Acessibilidade, que diz respeito ao quão acessível é o sistema para todos os tipos de usuário, possuindo limitações de capacidade ou não (NIELSEN, 1993);

- Comunicabilidade, que permite ao usuário tirar maior proveito das funcionalidades do sistema, uma vez que o próprio sistema comunica como determinada ferramenta ou funcionalidade é utilizada (NIELSEN, 1993).

É importante salientar que nem sempre será possível atender a todos esses critérios de forma satisfatória, logo a equipe de desenvolvimento deve averiguar quais serão mais importantes para o seu sistema. Na Central do Pet foram abrangidos os critérios de Usabilidade, Experiência do usuário e Comunicabilidade.

Já as *affordances*, segundo Norman (1988), são as características de um objeto capazes de revelar aos seus usuários as operações e manipulações que eles podem fazer com ele, em outras palavras, os objetos devem comunicar corretamente quais são suas funções, as características de um campo de inserção de texto, por exemplo, são diferentes das características de um botão. Este princípio não é utilizado apenas no *design* de *software*, qualquer objeto que tente comunicar suas funções está à mercê dele, como maçanetas de portas, lacres de garrafas, entre outros. É importante obedecer às *affordances* para não gerar confusão no usuário e acabar tornando o sistema difícil de se usar.

Esses princípios serão necessários para elaborar uma interface de sistema de fácil interação e um ambiente agradável para os usuários.

São necessárias também tecnologias para desenvolver um sistema *web*, como linguagens de programação, *frameworks*, banco de dados entre outras. Essas tecnologias são o que usamos para, de fato, construir o sistema.

Para que não haja perda dos registros gerados pelo sistema é necessário armazená-los em algum lugar. Para tal tarefa são utilizados bancos de dados. Podemos elencar duas abordagens para bancos de dados: bancos de dados relacionais e bancos de dados não-relacionais.

Nos bancos de dados relacionais, os registros são inseridos em tabelas, nas quais cada linha se refere a uma instância e cada coluna se refere a determinado atributo dessas

instâncias. Os bancos de dados relacionais são ótimos para trabalhar “verticalmente”, ou seja, em uma única tabela com uma grande quantidade de registros. Porém, quando é feita uma consulta que envolva muitas tabelas, são realizadas muitas operações, aumentando muito os esforços de processamento e memória.

Já os bancos de dados não-relacionais (também conhecidos como noSql) são bancos de dados em que os registros não são feitos em tabelas, mas sim em documentos, grafos, modelos chave/valor e outras abordagens. Um de seus principais atrativos é a escalabilidade, pois são projetados para aumentar a escala "horizontalmente", usando conjuntos de máquinas distribuídas de baixo custo para aumentar a transferência sem aumentar a latência.

Considerando que os computadores não entendem a nossa linguagem, se faz necessário o uso de algum mecanismo que converta nossas instruções em algo que ele possa executar: esse é o papel das linguagens de programação. Elas atuam como comunicadores de instruções para o computador, fazendo uso de regras sintáticas e semânticas mais compreensíveis ao ser humano. Ao longo da história da computação, diversas linguagens de programação surgiam, cada uma introduzindo facilidades e recursos que foram tornando a tarefa de programar mais fácil e menos suscetível a erros (GUDWIN, 1997).

Segundo Gudwin (1997), as linguagens de programação podem ser classificadas em diversas categorias, de acordo com suas características. São elas:

- Linguagens de Baixo Nível

São as linguagens que mais se assemelham a linguagem natural da máquina, que são conhecidas de forma unificada como “Linguagem Assembly”, sendo que cada processador possui um conjunto de instruções diferente e é necessária uma Linguagem Assembly diferente;

- Linguagens Não-Estruturadas

São linguagens um pouco mais flexíveis que as de Baixo Nível, pois seus comandos são mais simples para a compreensão humana, e não estão vinculados ao processador, tornando possível seu uso em diferentes plataformas. Podemos citar como exemplo as linguagens Cobol e Basic;

- Linguagens Procedurais

São linguagens que se diferenciam das Não-Estruturadas pela existência de estruturas de controle, que realizam teste de condições, repetição de blocos de código, seleção de alternativas e dividem o código em módulos, chamados de funções ou procedimentos. Alguns exemplos desse tipo de linguagem são o C e o Pascal;

- Linguagens Funcionais

As linguagens de programação Funcionais se caracterizam pela avaliação de expressões, ao invés da execução de comandos, expressões essas que são formadas utilizando funções para combinar valores básicos. Nesta categoria estão as linguagens Prolog e LISP;

– Linguagens Orientadas a Objeto

As linguagens Orientadas a Objeto foram criadas a partir da ideia fundamental de tentar simular o mundo real no computador, para isso foi utilizado o conceito de objetos. Esses objetos possuem estado e comportamento e interagem entre si, por meio do envio de mensagens, sendo que o programador deve definir quais objetos poderão receber que tipo de mensagens e as ações que os objetos devem realizar ao receber cada mensagem. Java e C++ são exemplos desse tipo de linguagem;

– Linguagens Específicas a Aplicações

Essas linguagens são, como o nome sugere, específicas para certas aplicações, como linguagens para bancos de dados, linguagens para simulação, linguagens de scripts e linguagens de formatação de texto, cada uma com sua sintaxe própria. Dentro das linguagens de formatação de texto temos HTML - *Hyper Text Markup Language* - que é uma linguagem de marcação de páginas *web*. É através dela que criamos os textos, formulários, botões e qualquer outro item que irá ser mostrado no *front-end* para o usuário. Por ser apenas uma linguagem de marcação as estilizações são as mínimas possíveis, como itálico, negrito, sublinhado e tamanho da fonte. Sendo assim, uma página criada apenas com HTML pode ser visualmente desagradável. Para contornar isso, existe o CSS - *Cascading Style Sheets* - que é um mecanismo utilizado para adicionar estilo às páginas criadas com HTML, possibilitando assim criar *designs* variados, tornando as páginas mais atrativas para o usuário.

A escolha da linguagem de programação mais adequada será baseada no tipo e na finalidade do sistema que será desenvolvido, levando em consideração também os recursos disponíveis para o desenvolvimento. Para um sistema *web* é comum o uso de linguagens Funcionais e Orientadas a Objeto, juntamente com a linguagem Específica HTML.

3 CENTRAL DO PET

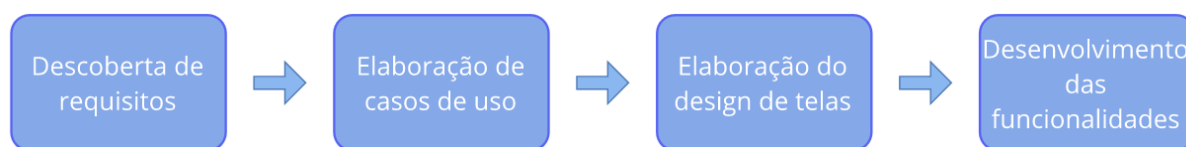
A partir dos estudos realizados, notou-se que o abandono de animais é um problema que não pode ser ignorado pois, além do sofrimento dos animais, ainda se trata de uma questão de saúde pública, uma vez que esses bichos podem trazer doenças para o ser humano. Assim, faz-se necessário o desenvolvimento de soluções que colaborem para diminuir o número de animais nessa condição.

Durante o desenvolvimento desse trabalho, além da dificuldade em conectar as pessoas que necessitam que seus bichos sejam adotados com as pessoas que querem adotar, notou-se também a dificuldade em quantificar com precisão o número de animais disponíveis em ONGs e abrigos. Isso ocorre devido as entidades atuarem de forma regional e não existir um compartilhamento efetivo desses dados. Desse modo, as soluções que focam em abandono de animais necessitam também possibilitar a geração de estatísticas, permitindo descrever em números a situação de animais abandonados.

Com o intuito de amenizar esses problemas foi desenvolvido a Central do Pet. Trata-se de um sistema *web* que tem como objetivo criar um ambiente de colaboração sobre animais em situação de abandono, agregando informações sobre eles e facilitando a comunicação sobre questões como doações, adoções e divulgação de campanhas/eventos.

O desenvolvimento do projeto foi executado conforme descrito na Figura 4. Como é possível observar na figura, foram planejadas quatro grandes etapas: descoberta de requisitos, elaboração de casos de uso, elaboração do *design* de telas e desenvolvimento das funcionalidades.

Figura 4 - Fluxo da metodologia de desenvolvimento do projeto



Fonte: Autoria própria (2018)

Na etapa de descoberta de requisitos foi realizada a descoberta e levantamento dos requisitos que o sistema deveria atender. Já na etapa de elaboração de casos de uso, foram elaborados casos de uso para modelar o funcionamento do sistema, usando como base os requisitos encontrados na fase de descoberta de requisitos. A partir dos casos de uso, foram

estudadas as telas que o sistema precisaria e foram elaborados *designs* agradáveis e interativos. Na última etapa, desenvolvimento das funcionalidades, todas as informações de etapas anteriores foram consideradas, sendo realizada a construção das funções e telas do sistema.

3.1 DESCOBERTA DE REQUISITOS

No desenvolvimento de um sistema é preciso considerar os requisitos funcionais que tratam das funções que o sistema precisa desempenhar. Por exemplo, em um sistema de consultas, seriam requisitos funcionais a realização de cadastros, emissão de relatório, entre outros. Esses requisitos são de extrema necessidade no processo de desenvolvimento de *software*, pois é a partir deles que as funcionalidades (tarefas que o sistema realizará) serão desenvolvidas. Vale salientar que não necessariamente teremos apenas um requisito funcional por funcionalidade.

Dito isso, a primeira tarefa de desenvolvimento da Central do Pet a ser realizada foi a descoberta e análise dos requisitos do sistema, utilizando a técnica de Reuso de requisitos. Para tal, utilizou-se como base o estudo de seis ferramentas similares já disponíveis na *web*: AdoteBicho, Adote um Focinho, Ampara Animal, Cãopanheiro Curitiba, Patinhas Online e Rockbicho. A partir delas, identificou-se 16 requisitos que estão relacionados ao gerenciamento das contas e registros que serão inseridos no sistema. São eles:

- Escolher cidade e estado;
- Criação de conta usuário-ONG;
- Criação de conta usuário-simples;
- Criação de conta usuário-CCZ-SAÚDE;
- Confirmação de conta de usuário;
- Editar conta usuário-simples;
- Editar conta usuário-ONG;
- Editar conta usuário-CCZ-SAÚDE;
- Excluir contas de usuário;
- Incluir animal;
- Editar animal;
- Arquivar animal;
- Incluir evento;

- Editar evento;
- Exibir eventos e animais na página inicial da cidade escolhida pelo usuário.

Os detalhes de cada um desses requisitos podem ser vistos no apêndice A deste documento.

Para a utilização do sistema pelos diferentes usuários, foi determinado a criação de 3 tipos de usuários com conta: usuário-simples, usuário-ONG e usuário-CCZ-SAÚDE. A diferença entre os tipos de conta está nas informações que cada um deve apresentar. Por exemplo, dados bancários para recebimento de doações no caso de ONGs e autoridades responsáveis.

No que se diz respeito ao cadastro de animais e eventos no sistema, foram elencados os dados necessários de cada, como foto, espécie e idade para animais e *banner*, data e tipo de ação para eventos.

3.2 ELABORAÇÃO DE CASOS DE USO

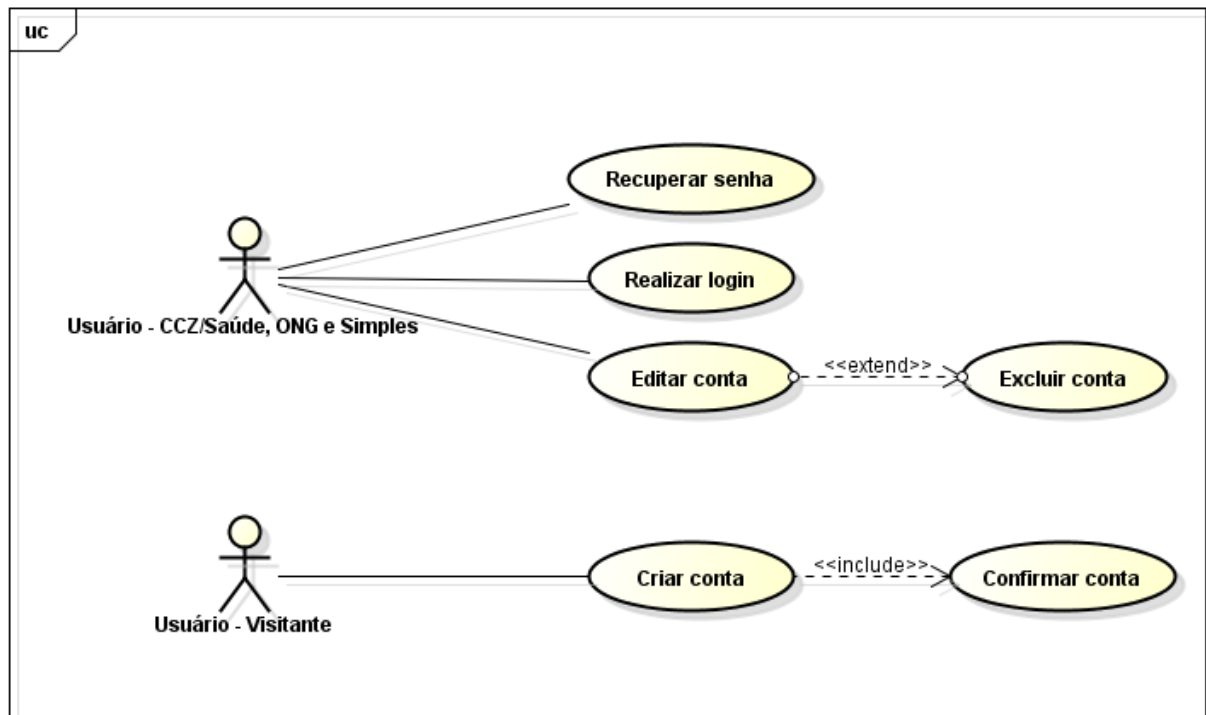
Com base nos requisitos, foram criados casos de uso para caracterizar as funcionalidades e o comportamento do sistema, representados através dos Diagramas de Caso de Uso da UML. A utilização de diagramas UML é importante pois estes permitem um bom entendimento do sistema a ser desenvolvido, uma vez que seus diagramas são facilmente compreendidos pois se utilizam de recursos visuais para representar fluxos e comportamentos do sistema.

Para desenhar os diagramas foi utilizada a ferramenta Astah Community. A escolha dessa ferramenta se deu devido sua facilidade de uso, bem como o fato de ser disponibilizada gratuitamente.

Para o desenvolvimento da Central do Pet foram elaborados 2 diagramas de Caso de Uso: o diagrama de conta de usuário e o diagrama de funcionalidades do sistema.

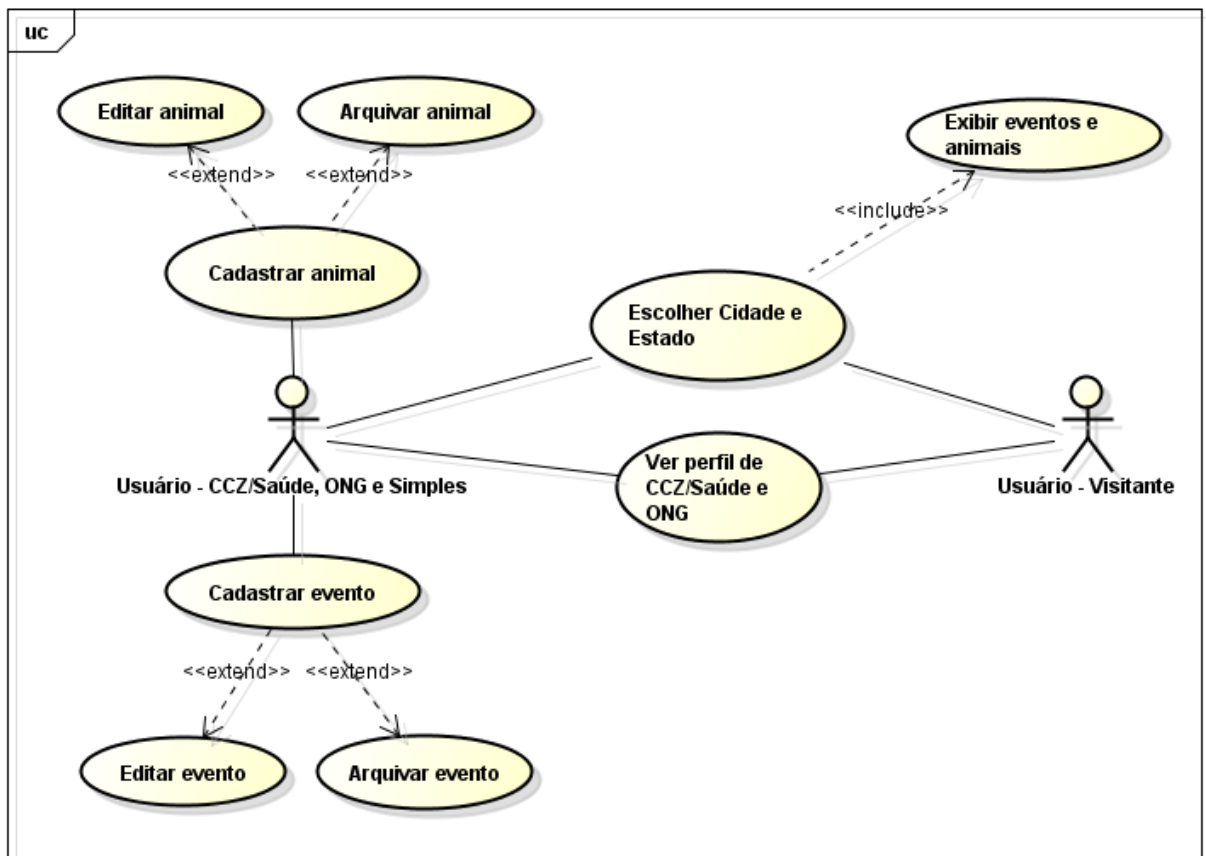
No diagrama de conta, apresentado na Figura 5, foram representados comportamentos e funcionalidades das contas de usuário. Foi possível identificar 2 tipos de ator nesse contexto: os usuários CCZ-SAÚDE, ONG e Simples, que já possuem conta, teriam acesso a funcionalidades de gerenciamento e uso de conta, como realizar login, recuperar senha, editar e excluir conta; e os usuários visitantes que não possuem conta no sistema, que teriam acesso a funcionalidade de criação e confirmação de conta.

Figura 5 – Diagrama de Caso de Uso - conta



Fonte: Autoria própria (2018)

Figura 6 – Diagrama de Caso de Uso – sistema



Fonte: Autoria própria (2018)

No diagrama de sistema, apresentado na Figura 6, foram representados comportamentos e funcionalidades relacionados ao sistema, que consistem em quais tarefas os usuários podem realizar no sistema, quais atividades são dependentes de outras e que tipo de usuário tem acesso a determinada funcionalidade. Sendo assim, foi possível identificar atividades exclusivas de usuários com conta como o cadastro de eventos e animais disponíveis para adoção, podendo também editá-los e arquivá-los conforme necessário; e atividades que qualquer usuário tem acesso dentro do sistema, como explorar as páginas de exibição de animais e eventos classificados por cidade, e ver perfis de entidades públicas, como CCZs e ONGs.

3.3 ELABORAÇÃO DO *DESIGN* DE TELAS

Com os Diagramas de Caso de Uso, foi possível ter uma ideia mais clara de como ocorrem os processos de interação com o usuário, e assim, foi dado início a identificação das telas necessárias para a interface do sistema.

Na criação do *design* das telas de interface com o usuário foi utilizado como base os princípios de qualidade de uso da IHC, que tratam de características como usabilidade, experiência do usuário, acessibilidade e comunicabilidade do sistema, e respeitando as *affordances*, que são características de um objeto que comunicam ao usuário quais suas funções, para não gerar confusão ao usuário enquanto estiver utilizando o sistema. Como exemplo de *affordance*, podemos citar que um campo de texto deve ter características visuais de um campo de texto (caixa vazia, com bordas finas, ponto de inserção e algum texto informativo como “Digite aqui”, e um tom claro na cor do texto).

Os *designs* foram criados com o auxílio da ferramenta Gravit Designer, que é uma ferramenta de *design* focada em simplicidade. A tela inicial do sistema, apresentada no design 1 do apêndice B deste documento, tenta construir algum grau de comunicabilidade e usabilidade com o usuário, por meio das imagens de setas direcionais e um título com mais destaque. Nos designs 2 e 3 do apêndice B deste documento são mostradas a tela principal do sistema e a tela de detalhe do evento, que usam ícones com o intuito de melhorar a comunicabilidade do sistema, uma vez que atuam como simplificadores de *ideias* (OZONO, 2017). Como aplicação de *affordances*, os *inputs* e botões dessas páginas também apresentam características que são comuns a esses recursos, como uma cor de fundo clara, no caso dos *inputs*, e uma cor forte para dar destaque e formato de retângulo arredondado, no caso dos botões.

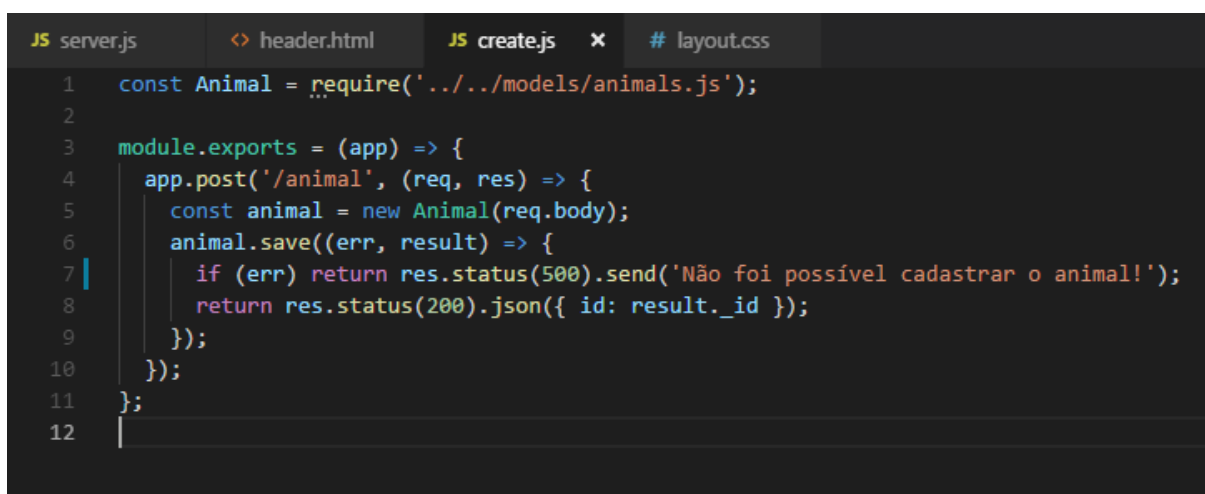
É possível notar ainda que o sistema faz uso de cores suaves que não causam irritação visual ao usuário, melhorando a experiência do usuário (DEVANER, 2016).

3.4 DESENVOLVIMENTO DAS FUNCIONALIDADES

A fase de codificação do sistema ocorreu de forma incremental, cada nova funcionalidade foi incrementada as já existentes. A linguagem definida para desenvolvimento foi o JavaScript, pois se trata de uma linguagem leve, interpretada e procedural, executada na máquina do cliente. Além de ser a linguagem *client-side* mais utilizada, permite a utilização de vários efeitos visuais em sites, tornando o ambiente mais dinâmico para o usuário. Outra vantagem do JavaScript é que podemos utilizá-la tanto no *front-end*, quanto no *back-end*, graças ao Node.js, um interpretador executado no servidor da aplicação. Portanto, utilizar Node.js permitiu usar uma única linguagem durante todo o projeto.

Ainda assim, algumas tarefas do desenvolvimento *web* não são suportadas diretamente pelo Node.js, como por exemplo GET, POST, DELETE. Para evitar a necessidade de escrever o código destas por inteiro, foi utilizado um *framework*, que é uma biblioteca de classes específicas para realizar determinada tarefa. Para tal, foi escolhido o Express, pois é atualmente o mais popular de Node.js e já está consolidado no mercado. O Express atua na aplicação gerenciando os objetos de solicitação e de resposta e toda a comunicação entre o servidor, cliente e banco de dados. Na Figura 7 é mostrado o código de cadastro de animal (POST), onde podemos observar as variáveis responsáveis por enviar e receber as solicitações no Express (req e res).

Figura 7 – Método POST da aplicação



```
JS server.js  <> header.html  JS create.js  x  # layout.css
1  const Animal = require('../models/animals.js');
2
3  module.exports = (app) => {
4    app.post('/animal', (req, res) => {
5      const animal = new Animal(req.body);
6      animal.save((err, result) => {
7        if (err) return res.status(500).send('Não foi possível cadastrar o animal!');
8        return res.status(200).json({ id: result._id });
9      });
10   });
11 };
12 |
```

Fonte: Autoria própria (2018)

Para o banco de dados foi utilizada uma abordagem noSql, pois o sistema proposto visa abranger todo o Brasil, logo é necessária uma tecnologia robusta para gerir grandes quantidades de informação. A ferramenta utilizada foi o mongoDB, um banco de dados orientado a documentos, pois além do formato orientado a documentos ser mais leve, as estruturas de consulta são simples em comparação a outros do mesmo ramo.

A conexão ao banco de dados é feita através do Mongoose, que é uma biblioteca do Node.js. Essa biblioteca mapeia os objetos do mongoDB, traduzindo os dados do banco de dados para objetos JavaScript para que possam ser usados na aplicação. Na Figura 8 é mostrado como a conexão do Mongoose com mongoDB ocorre.

Figura 8 – Conexão do banco de dados

```
const db = Mongoose.connection;
db.on('error', (err) => {
  console.error(err);
});

db.once('open', () => {
  console.log('Conectado ao MongoDB');
});
// conectando ao banco de dados
const dataBaseUrl = 'mongodb://admin:Yfwyjr46@ds255262.mlab.com:55262/centraldopet';
mongoose.connect(dataBaseUrl);
```

Fonte: Autoria própria (2018)

A estrutura do banco de dados está baseada em modelos, em que cada um representa uma instância salva, que no caso da Central do Pet, são os usuários e seus tipos, os animais e os eventos que são registrados pelos usuários. Cada um desses modelos possui informações que os arquivos salvos podem apresentar (como nome, idade e espécie). Na figura 9 são exibidas as estruturas desses arquivos.

A consulta ao banco de dados é feita por meio do método *find()*, sendo possível também utilizar a busca com paginação, como demonstra a Figura 10, o que torna o carregamento de arquivos mais leve, pois o mesmo é feito em “blocos”.

Figura 9 – Estruturas dos arquivos do mongoDB

```

JS events.js JS animals.js JS simple.js JS events.js JS animals.js JS simple.js JS events.js JS animals.js JS simple.js JS
1 const mongoose = require('mongoose'); 1 const mongoose = require('mongoose'); 1 const mongoose = require('mongoose');
2 2 2
3 const Schema = mongoose.Schema; 3 const Schema = mongoose.Schema; 3 const Schema = mongoose.Schema;
4 4 4
5 const Ong = mongoose.model('Ong', { 5 const Ccz = mongoose.model('Ccz', { 5 const Event = mongoose.model('Event', {
6   name: String, 6   name: String, 6   name: String,
7   password: String, 7   password: String, 7   avatar: String,
8   socialName: String, 8   avatar: String, 8   startDate: Date,
9   avatar: String, 9   state: String, 9   endDate: Date,
10  state: String, 10  city: String, 10  timetable: String,
11  city: String, 11  district: String, 11  state: String,
12  district: String, 12  cep: Number, 12  city: String,
13  cep: Number, 13  street: String, 13  street: String,
14  street: String, 14  number: String, 14  number: String,
15  number: String, 15  phone: Number, 15  type: String,
16  phone: Number, 16  email: String, 16  description: String,
17  email: String, 17  responsibleEmail: String, 17  servicesOffered: String,
18  cnpj: Number, 18  responsiblePhone: Number, 18  cost: String,
19  maxCapacity: Number, 19  requesterEmail: String, 19  phone: Number,
20  currentCapacity: Number, 20  requesterPhone: Number, 20  email: String,
21  }); 21  }); 21  });
22 22 22
23 module.exports = Ong; 23 module.exports = Ccz; 23 module.exports = Event;

```

```

JS events.js JS animals.js JS simple.js JS JS events.js JS animals.js JS simple.js JS
1 const mongoose = require('mongoose'); 1 const mongoose = require('mongoose');
2 2
3 const Schema = mongoose.Schema; 3 const Schema = mongoose.Schema;
4 4
5 const Animal = mongoose.model('Animal', { 5 const Simple = mongoose.model('Simple', {
6   name: String, 6   name: String,
7   avatar: String, 7   password: String,
8   age: String, 8   avatar: String,
9   size: String, 9   state: String,
10  species: String, 10  city: String,
11  color: String, 11  street: String,
12  castrated: Boolean, 12  number: String,
13  dewormed: Boolean, 13  phone: Number,
14  microchip: Boolean, 14  email: String,
15  }); 15  });
16 16
17 module.exports = Animal; 17 module.exports = Simple;

```

Fonte: Autoria própria (2018)

Figura 10 – Exemplo de busca no banco de dados

```

JS index.js
1 const Animal = require('../models/animals.js');
2
3 module.exports = (app) => {
4   app.get('/animal', (req, res) => {
5     const page = req.query.page || 1;
6     Animal.find({}).limit(20).skip((page - 1) * 20).exec((err, animals) => {
7       if (err) return res.status(500).send();
8       return res.status(200).json(animals);
9     });
10  });
11 };
12

```

Fonte: Autoria própria (2018).

Assim como para o *back-end*, foi utilizado um *framework* para o *front-end*. No caso, foi utilizado o Angular, um *framework* focado em produtividade e reuso de código. Com o intuito de demonstrar a capacidade de redução do tamanho do código, Miško Hevery, um dos criadores do Angular, reescreveu um projeto da Google que possuía 17.000 linhas em apenas 1.500 utilizando seu *framework* (BRANAS, 2014).

Com o Angular foi possível evitar esforço na escrita do código, por ser um *framework* com diversos recursos e funções já implementadas. Na Figura 11 é apresentado um exemplo de comparação entre código JQuery e o código do Angular. É possível notar uma diminuição de linhas entre os dois trechos de código que tem a mesma finalidade, além de ser mais limpo e simples, sempre que é escrito menos código, tem-se uma facilidade maior na manutenção e produtividade maior no desenvolvimento.

Figura 11 – Comparativo entre Angular e JQuery



Fonte: GARBIN (2014)

4 CONSIDERAÇÕES FINAIS

Esta monografia propôs, como objetivo geral, desenvolver um sistema *web* para atuar como um ambiente de colaboração sobre animais em situação de abandono. A Central do Pet foi o resultado de todos os processos de desenvolvimento apreciados neste documento: foram realizados levantamento de requisitos, analisando ferramentas já existentes; criação de casos de uso, utilizando as normas definidas pela UML; integração do sistema a um banco de dados noSql, desenvolvimento de funcionalidades do sistema e elaboração de *designs* interativos e intuitivos.

A Central do Pet ainda é uma aplicação básica, podendo ser a base de um projeto muito mais abrangente, contemplando muitos outros serviços como veterinária, hotelaria, *pet shops*, denúncias online, curiosidades e informações sobre animais e muitos outros que podem ser sugeridos pela comunidade.

Vale reforçar uma vez mais que a Central do Pet é um meio para tentar diminuir os índices de abandono, visto que o abandono de animais é um problema complexo e que não se resolverá rapidamente. Cabe a todos ajudar de alguma forma, seja adotando um animal de rua, compartilhando informações sobre animais abandonados com as entidades responsáveis ou mesmo com doações para abrigos de sua cidade.

BIBLIOGRAFIA

AGÊNCIA DE NOTÍCIAS DE DIREITOS ANIMAIS. **Cresce para 30 milhões o número de animais abandonados no Brasil.** Disponível em: <<https://anda.jusbrasil.com.br/noticias/154729183/cresce-para-30-milhoes-o-numero-de-animais-abandonados-no-brasil>> Acesso em: 06 abr. 2018.

AMAZON. **O que é banco de dados relacional?** Disponível em: <<https://aws.amazon.com/pt/relational-database>> Acesso em: 06 abr. 2018

AMAZON. **O que é NoSQL?** Disponível em: <<https://aws.amazon.com/pt/nosql/>> Acesso em: 06 abr. 2018.

BARBOSA, S. D. J.; SILVA, B. S. **Interação Humano Computador.** Campus-Elsevier, 2010.

BARBOZA, Mariana Queiroz. **As 100 melhores ONGs do Brasil.** Disponível em: <<https://epoca.globo.com/brasil/noticia/2017/08/100-melhores-ongs-do-brasil.html>> Acesso em: 12 set. 2018.

BRANAS, Rodrigo. **AngularJS Essentials.** Packt Publishing, 2014.

BRASIL. DECRETO LEI N° 2.848, DE 07 DE DEZEMBRO DE 1940. **Código Penal.** Disponível em: <<https://www.jusbrasil.com.br/topicos/10618135/artigo-164-do-decreto-lei-n-2848-de-07-de-dezembro-de-1940>> Acesso em: 17 abr. 2018.

BRASIL. LEI N° 9.605, DE 12 DE FEVEREIRO DE 1998. **Lei de Crimes Ambientais.** Brasília, DF, fev 1998. Disponível em: <<https://presrepublica.jusbrasil.com.br/legislacao/104091/lei-de-crimes-ambientais-lei-9605-98>> Acesso em: 17 abr. 2018.

CURSOS CTP. **Linguagem de Programação JavaScript:** as principais vantagens. Disponível em: <<https://www.cpt.com.br/cursos-informatica-desenvolvimentodesoftwares/artigos/linguagem-de-programacao-javascript-as-principais-vantagens>> Acesso em: 06 abr. 2018.

DANDARA, Mariana. **Dívidas e superlotação impedem ONGs de resgatar animais.** Disponível em: <<https://digitaispuccampinas.wordpress.com/2015/10/06/dividas-e-superlotacao-impedem-ongs-de-resgatar-animais/>> Acesso em: 10 abr. 2018.

DEVANER, Marcos. **Interação Humano Computador (Aula 9) – Uso de Cores.** Disponível em: <<https://pt.slideshare.net/devaner/acessibilidade-na-web-ihc-aula-09>> Acesso em: 12 set. 2018.

GARBIN, Wolmir C. **O que é AngularJS.** Disponível em: <<https://wolmirgarbin.wordpress.com/javascript/angularjs/>> Acesso em: 17 set. 2018.

GOMES, Caroline Cavalcante Maia. **Guarda responsável de animais de companhia:** um estudo sobre a responsabilidade civil dos proprietários e a entrega de cães e gatos na Diretoria de Vigilância Ambiental do Distrito Federal. 2013. 70 f., il. Monografia (Bacharelado em Medicina Veterinária) — Universidade de Brasília, Brasília, 2013.

GUDWIN, R.R.; **Linguagens de Programação.** Universidade Estadual de Campinas, 1997.

JÚNIOR, Luiz Fernando Duarte. **Por que Stanford trocou Java por JavaScript?** Disponível em: <<https://imasters.com.br/desenvolvimento/por-que-stanford-trocou-java-por-javascript/?trace=1519021197&source=single>> Acesso em: 06 abr. 2018.

JUSTEN, William. **A importância do Github para Desenvolvedores.** Disponível em: <<https://willianjusten.com.br/a-importancia-do-github-para-desenvolvedores>> Acesso em: 17 abr. 2018

KRAUSE, J. **Introducing Web Development.** Springer, 2016

MANTOVANI, Gustavo. **“Microchipagem”:** segurança para seu pet. Disponível em: <<http://www.revistapulodogato.com.br/materias/ler-materia/86/microchipagem-seguranca-para-seu-pet>> Acesso em: 21 abr. 2018.

MEUS ANIMAIS. **O microchip para cães e gatos.** Disponível em: <<https://meusanimais.com.br/o-microchip-para-caes-e-gatos/>> Acesso em: 21 abr. 2018.

MILETTO, Evandro Manara; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de Software II:** Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP - Eixo: Informação e Comunicação - Série Tekne. Porto Alegre: Bookman Editora, 2014.

MONTEIRO, Elder. **Superlotação preocupa ONGs que cuidam de animais no ABC.** Disponível em: <<http://www.metodista.br/rronline/noticias/cidades/2011/08/superlotacao-preocupa-ongs-que-cuidam-de-animais-no-abc>> Acesso em: 10 abr. 2018.

MOZILLA et. al. **Express/Node introduction.** Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction> Acesso em: 06 abr. 2018.

MUTIIS, Fabiana de. **Abandono de cães e gatos se torna problema de saúde pública em AL.** Disponível em: < <http://g1.globo.com/al/alagoas/noticia/2013/09/abandono-de-caes-e-gatos-se-torna-problema-de-saude-publica-em-al.html>> Acesso em: 12 set. 2018.

NATALINO, Alex. **Abandonar animais é crime previsto em Lei.** Disponível em: <<https://carollinasalle.jusbrasil.com.br/noticias/395206911/abandonar-animais-e-crime-previsto-em-lei>> Acesso em: 17 abr. 2018.

NIELSEN, J.; **Usability Engineering.** Morgan Kaufmann Inc, 1993.

NORMAN, D. A.; **The Design Of Everyday Things.** Doubleday, 1988.

OZONO, Letícia. **Como melhorar a usabilidade com ícones.** Disponível em: <<http://blog.alura.com.br/icones-design-e-usabilidade-na-web/>> Acesso em: 12 set. 2018.

PORTAL TUDO SOBRE CÃES. **Turismo 4 Patas.** Disponível em: <http://www.turismo4patas.com.br/htms/pop_porte.htm?&iiframe=true&width=600&height=340> Acesso em: 21 abr. 2018.

RAGGETT, D.; LE HORS, A.; JACOBS, I. **HTML 4.01 Specification.** W3C recommendation, 1999.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **Unified Modeling Language Reference Manual.** Addison-Wesley, 2010.

SCHOLZE, Marianne. **10 benefícios dos pets à saúde e ao bem-estar humano.** Disponível em: <<https://saude.ig.com.br/minhasaude/2013-04-20/10-beneficios-dos-pets-a-saude-e-ao-bem-estar-humano.html>> Acesso em 12 set. 2018.

SHIMOZAKO et al. Uso de sistemas de informação geográfica como ferramentas de auxílio na análise de casos de apreensão de animais no município de São Paulo. **Revista Ciência em Extensão**, v.2 n.2, 2006.

SONCELA, Paula. **A importância da vermifugação para a saúde do pet**. Disponível em: <<http://portalmelhoresamigos.com.br/a-importancia-da-vermifugacao-para-a-saude-do-pet/>> Acesso em: 21 abr. 2018.

APÊNDICE A – Requisitos do sistema

Requisito funcional 001

Identificador	RF001		
Nome	Escolher cidade e estado		
Módulo	N/A		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Ao acessar o sistema, o usuário visitante deverá escolher cidade e estado, e os dados exibidos deverão ser somente da localidade escolhida.		

Requisito funcional 002

Identificador	RF002		
Nome	Criação de conta usuário-ONG		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Na criação de conta do usuário-ONG deverão constar as seguintes informações: nome fantasia, endereço (rua, número, bairro, cidade, estado, CEP), informações para contato (e-mail, telefone), CNPJ, razão social, foto de perfil, capacidade máxima do abrigo, quantidade de animais abrigados, senha.		

Requisito funcional 003

Identificador	RF003		
Nome	Criação de conta usuário-simples		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Na criação de conta do usuário-simples deverão constar as seguintes informações: nome, foto de perfil, informações para contato (e-mail, telefone), endereço, senha.		

Requisito funcional 004

Identificador	RF004		
Nome	Criação de conta usuário-CCZ-SAÚDE		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Na criação de conta do usuário-CCZ-SAÚDE deverão constar as seguintes informações: nome, foto de perfil, endereço, informações para contato do CCZ ou secretaria de saúde (e-mail, telefone, ramal), capacidade máxima do abrigo, quantidade de animais abrigados, CPF do responsável pelo CCZ ou secretaria de saúde, informações para contato com o responsável pelo CCZ ou secretaria de saúde (e-mail, telefone), nome completo do solicitante da criação da conta, informações de contato do solicitante da conta, senha.		

Requisito funcional 005

Identificador	RF005		
Nome	Confirmação de conta de usuário		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Deverá ocorrer o processo de verificação por e-mail para efetivação da criação da conta.		

Requisito funcional 006

Identificador	RF006		
Nome	Editar conta usuário-simples		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	O usuário-simples poderá editar suas informações: nome, endereço (rua, número, bairro, cidade, estado, CEP), informações para contato (e-mail, telefone), descrição, senha.		

Requisito funcional 007

Identificador	RF007		
Nome	Editar conta usuário-ONG		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	O usuário-ONG poderá editar suas informações: nome fantasia, endereço (rua, número, bairro, cidade, estado, CEP), informações para contato (e-mail, telefone, ramal), CNPJ, razão social, descrição, dados bancários para recebimento de doações, horário de funcionamento, capacidade máxima do abrigo, quantidade de animais abrigados, senha.		

Requisito funcional 008

Identificador	RF008		
Nome	Editar conta usuário-CCZ-SAÚDE		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	O usuário-CCZ-SAÚDE poderá editar suas informações: nome, endereço (rua, número, bairro, cidade, estado, CEP), informações para contato (e-mail, telefone, ramal), descrição, horário de funcionamento, capacidade máxima do abrigo, quantidade de animais abrigados.		

Requisito funcional 009

Identificador	RF009		
Nome	Excluir contas de usuário		
Módulo	Usuários		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Deverá ser mostrada, nas configurações de conta, a possibilidade de exclusão de qualquer tipo de conta (com confirmação de ação e aviso de ação irreversível).		

Requisito funcional 010

Identificador	RF010		
Nome	Incluir animal		
Módulo	Registro de animais		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Para mostrar um animal para adoção, é necessário criar o registro no banco de dados. Esse registro deverá incluir: nome (se houver), idade (filhote, adolescente, adulto ou idoso), porte (mini/toy, pequeno/anão, médio, grande ou gigante), espécie, cor dos pelos (para cães: cinza, preto, branco, caramelo, preto e branco, caramelo e branco, caramelo e preto, tricolor ou pintas; para gatos: cinza, preto, branco, laranja, preto e branco, laranja rajado com branco, cinza rajado com branco, tricolor ou siamês), castrado, vermifugado, microchipagem, descrição do comportamento e foto.		

Requisito funcional 011

Identificador	RF011		
Nome	Editar animal		
Módulo	Registro de animais		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Deve ser possível a edição dos dados de um animal disponível para adoção. Nome, idade (filhote, adolescente, adulto ou idoso), porte (mini/toy, pequeno/anão, médio, grande ou gigante), espécie, cor dos pelos (para cães: cinza, preto, branco, caramelo, preto e branco, caramelo e branco, caramelo e preto, tricolor ou pintas; para gatos: cinza, preto, branco, laranja, preto e branco, laranja rajado com branco, cinza rajado com branco, tricolor ou siamês), castrado, vermifugado, microchipagem, descrição do comportamento, foto e checkbox de adoção.		

Requisito funcional 012

Identificador	RF012		
Nome	Arquivar animal		
Módulo	Registro de animais		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Importante
Descrição	Após a adoção, o registro do animal deverá ser mantido para fins de análise de históricos quantitativos, porém o registro deverá ser movido para uma área de “arquivados” do perfil do usuário.		

Requisito funcional 013

Identificador	RF013		
Nome	Incluir evento		
Módulo	Registro de eventos		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Para exibir um evento ou ação comunitária deverá ser criado um registro no banco de dados. Esse registro deverá conter: banner da ação, data (podendo ser mais de um dia), horário, local, tipo de ação (campanha veterinária, feira de adoção, castração comunitária, campanha de vacinação), descrição do evento, serviços oferecidos durante o evento (castração, adoção, microchipagem, vacinação) e custos (de entrada, dos serviços oferecidos, caso haja a cobrança).		

Requisito funcional 014

Identificador	RF014		
Nome	Arquivar evento		
Módulo	Registro de eventos		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Após o evento, o registro da ação deverá ser mantido para fins de análise de históricos quantitativos, porém o registro deverá ser movido para uma área de “arquivados” do perfil do usuário.		

Requisito funcional 015

Identificador	RF015		
Nome	Editar evento		
Módulo	Registro de eventos		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Deverá ser possível a edição dos eventos registrados. Banner da ação, data (podendo ser mais de um dia), horário, local, tipo de ação (campanha veterinária, feira de adoção, castração comunitária, campanha de vacinação), descrição do evento, serviços oferecidos durante o evento (castração, adoção, microchipagem, vacinação) e custos (de entrada, dos serviços oferecidos, caso haja a cobrança).		

Requisito funcional 016

Identificador	RF016		
Nome	Exibir eventos e animais na página inicial da cidade escolhida pelo usuário		
Módulo	N/A		
Data de criação	25/06/2018	Autor	Alescia Fernandes
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Essencial
Descrição	Após a escolha da cidade e estado pelo usuário visitante, deverá ser mostrada uma página inicial com alguns eventos e alguns animais disponíveis naquela cidade, focando nos banners e fotos dos animais. Além de mostrar separadamente as ONGs e CCZs atuantes naquela cidade.		

APÊNDICE B – Design de telas

Design 1 – Tela inicial do sistema



Design 2 – Tela principal do sistema



Design 3 – Tela de detalhe do evento

Menu

Central do Pet

Entrar ou Criar conta

Pesquisar

Angicos - RN

Cuide bem do seu bichinho

13 a 25 de agosto

Caros amigos, a execução dos pontos do programa auxilia a preparação e a composição do sistema de participação geral. Todas estas questões, devidamente ponderadas, levantam dúvidas sobre se a expansão dos mercados mundiais cumpre um papel essencial na formulação dos níveis de motivação.

Resumo do evento

Período	Horário
13 a 25 de agosto	Das 8:00 às 17:00
Local	
Hospital veterinário de Angicos Rua das Alamedas, 38 Bairro Asa Norte	
Ação principal	
Feira de adoção	
Organizador	
Hospital Veterinário de Angicos	
Serviços oferecidos	Custos
Adoção	Entrada: R\$ 0,00
Vacinação	Adoção: R\$ 0,00
Castração	Vacinação: R\$ 0,00
	Castração: R\$ 30,00

Design 4 – Tela de detalhe do pet

Menu

Central do Pet

Entrar ou Criar conta

Pesquisar

Angicos - RN

Informações do pet

Nome	Espécie	Sexo
Sem nome	Gato	Fêmea
Idade	Cor	
Filhote - até 1 ano	Laranja rajado com branco	
Raça	Porte	
SDR - Sem raça definida	Mini/toy - até 6kg	
Castrado	Vermifugado	Microchip
Não	Não	Não
Descrição do Comportamento		
Calma, carinhosa, preguiçosa. Tem medo de carros.		

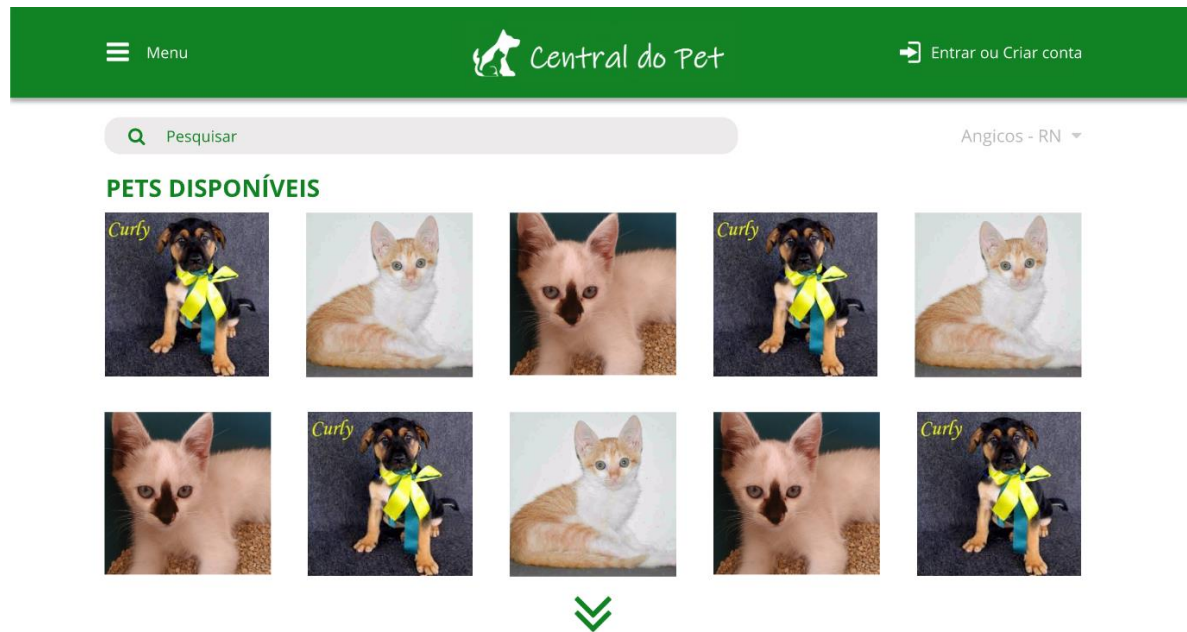
Responsável

Júlio Augusto da Silva

Contato

julio123@gmail.com

Design 5 – Tela de listagem de pets disponíveis



Design 6 – Tela de login/criação de conta

