



---

**Faculdade de Tecnologia de Americana – Ministro Ralph Biasi**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Adriano Benatti Junior  
Matheus Maldonado da Silva  
Maurílio Alexandro Lima

## **SOFTWARE PARA GESTÃO DE ADOÇÃO DE ANIMAIS**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas sob a orientação do (a) Prof.<sup>(a)</sup> Esp. Antônio Alfredo Lacerda.

**Americana, SP**

**2019**



---

**Faculdade de Tecnologia de Americana – Ministro Ralph Biasi**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

Adriano Benatti Junior  
Matheus Maldonado da Silva  
Maurílio Alexandro Lima

**SOFTWARE PARA GESTÃO DE ADOÇÃO DE ANIMAIS**

**Americana, SP**  
**2019**

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS**  
**Dados Internacionais de Catalogação-na-fonte**

B394s BENATTI JUNIOR, Adriano

Software para gestão de adoção de animais. / Adriano Benatti Júnior, Maurilio Alexandre Lima, Matheus Maldonado da Silva. – Americana, 2019.

98f.

Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Esp. Antônio Alfredo Lacerda

1 Sistemas orientados a objetos 2. Desenvolvimento de software I. LIMA, Maurílio Alexandre II. SILVA, Matheus Maldonado da III. LACERDA, Antônio Alfredo IV. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.3.04

681.3.05

Adriano Benatti Junior  
Matheus Maldonado da Silva  
Maurílio Alexandro Lima

## SOFTWARE PARA GESTÃO DE ADOÇÃO DE ANIMAIS

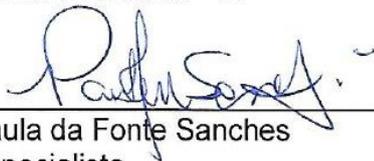
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Americana, 10 de Junho de 2019.

**Banca Examinadora:**



Antônio Alfredo Lacerda  
Especialista  
Fatec Americana - SP



Paula da Fonte Sanches  
Especialista  
Fatec Americana - SP



Alberto Martins Júnior  
Especialista  
Instituição de atuação:

## **AGRADECIMENTOS**

Primeiramente a Deus pela vida, por nos dar força, esperança e visão para iluminar os nossos caminhos.

As nossas famílias, que sempre acreditaram em nossos potenciais e nos deram toda a base para chegarmos até aqui, nos motivando e nos desafiando para que nossos sonhos fossem realizados.

Em especial ao professor especialista Antônio Alfredo Lacerda que nos orientou afim de que esse trabalho fosse concluído.

A todos os professores da Fatec Americana que contribuíram com o conhecimento adquirido, pelo incentivo e pela amizade.

Aos amigos de classe e todas as pessoas que fizeram parte da nossa formação.

## RESUMO

Tendo em vista as dificuldades de ONG's em divulgar os animais para adoções em seus eventos e o número de animais aumentando devido ao índice de animais abandonados, esse projeto se propôs a desenvolver um sistema web para gestão de adoção de animais. O objetivo foi criar uma plataforma digital e interativa onde os animais pudessem ser apresentados em uma página da internet para que qualquer pessoa que acessasse o sistema e tivesse interesse, conseguisse realizar a adoção de algum animal ou se voluntariar na ONG. A proposta foi promover o aumento das adoções por meio da plataforma digital e evitar o abarrotamento das baias, criando um canal de interação de voluntários e adotantes e auxiliando a gestão dos administradores por intermédio de relatórios. Para a realização do desenvolvimento desse sistema foram utilizadas técnicas de Engenharia de Software, a criação de diagramas UML e modelagem de dados do banco de dados, definição de arquitetura de software MVC (Model, View e Controller), linguagem de programação e de criação de interface gráfica e estrutura de servidor web, que possibilitassem o acesso simultâneo de várias máquinas a uma única aplicação e banco de dados. Optou-se pela utilização da metodologia ágil Kanban para monitoramento das atividades relacionadas ao desenvolvimento do sistema, por ser uma metodologia interativa e de fácil visualização, trazendo benefícios devido a compatibilidade com mudanças dentro do projeto e não dependendo de etapas concluídas para serem começadas. Com isso foi realizado a criação de um protótipo funcional de um sistema web para adoção de animais em um ambiente real, possibilitando implementar novas funcionalidades para atender ONG's de acordo com os seus requisitos, de forma mais abrangente, realizar a melhoria na gestão de adoção de animais, afim de aumentar o índice de adoções e voluntariado.

**Palavras Chave:** Gestão adoção de animais; ONG's; Sistema web.

## ABSTRACT

In view of the difficulties of NGO's (Non-governmental organization) in publicizing animals for adoptions in their events and the increasing number of animals related to the index of abandoned animals, the development of a web system for animal adoption management was started. The goal was to create a digital and interactive platform where animals would be exposed on a web page and anyone accessing the system could adopt animal adoption and volunteer at the NGO. The idea was to promote the increase of adoptions through the digital platform and avoid the crowding of the bays, creating a channel of interaction of volunteers and adopters and assisting the management of the administrators through reports. For the development of this system were used Software Engineering techniques the creation of UML (Unified Modeling Language) diagrams and data modeling of the database, definition of software architecture MVC (Model, View and Controller) as well as programming language and creation of graphical interface and web server structure, which enabled the simultaneous access of multiple machines to a single application and database. The agile Kanban methodology was used to monitor activities related to development, because it is interactive and easy to visualize, bringing benefits of being compatible with changes within the project and not depending on the steps completed to be started. This led to the creation of a functional prototype of a web system for adoption of animals in a real environment, making it possible to implement new functionalities to meet an NGO according to its requirements, in a more comprehensive way, to carry out the improvement in adoption management to increase adoption rates and volunteerism.

**Keywords:** Animal husbandry management; NGO; Web system.

## SUMÁRIO

INTRODUÇÃO .....	3
2. METODOLOGIA .....	5
2.1 Metodologia ágil .....	5
2.2 Metodologia Kanban .....	6
2.2.1 Cronograma e divisão do trabalho .....	8
2.3 Tecnologias empregadas .....	9
2.4 Banco de dados .....	10
2.5 Ferramenta PHPMyAdmin .....	11
2.6 Servidor <i>Web</i> Apache .....	12
2.7 Frameworks - Laravel.....	13
2.8 Bootstrap .....	15
3. ENGENHARIA DE SOFTWARE.....	16
3.1 Levantamento de requisitos .....	17
3.2 Diagrama de casos de uso.....	19
3.3. Diagrama de classes .....	22
3.4. Dicionário de dados.....	23
3.5. Diagrama de atividades.....	28
3.6. Diagrama de sequencias.....	29
3.7. Modelagem de dados.....	36
3.8. Arquitetura de software MVC .....	37
4. PROJETO.....	39
4.1 Descrição do projeto .....	39
4.2 Dificuldades encontradas .....	39
4.3 Prototipagem .....	40

4.3.1 Menu navegação – Home .....	40
4.3.2 Tela Conheça nossos animais .....	42
4.3.3 Tela Seja Voluntário .....	42
4.3.10 Tela Animais – Animais .....	49
4.3.11 Tela Animais – Novo .....	50
4.3.12 Tela Relatórios – Adoções .....	52
4.3.13 Tela Relatórios – Animais para Adoção .....	53
4.4 – Casos de testes.....	54
5. CONCLUSÃO .....	57
REFERÊNCIAS BIBLIOGRÁFICAS .....	58
ANEXO I .....	60
ATA 1: ENTREVISTA COM VOLUNTÁRIA DA ONG VIRA LATA VIRA VIDA.....	60
ANEXO II .....	63
ATA 2: ENTREVISTA COM CARLOS, PRESIDENTE DA AAANO. ....	63
ANEXO III.....	65
MANUAL DO USUÁRIO.....	65

## LISTA DE FIGURAS

Figura 1 - Quadro Kanban.....	7
Figura 2 - Quadro detalhado .....	8
Figura 3 - Cronograma Kanban.....	8
Figura 4 - Continuação cronograma Kanban.....	9
Figura 5 - Diagrama de caso de uso .....	20
Figura 6 - Diagrama de classe controller.....	23
Figura 7 - Diagrama de classe gerenciamento.....	23
Figura 8 - Diagrama de atividades .....	28
Figura 9 - Criar relatório .....	29
Figura 10 - Login .....	30
Figura 11 - Cadastrar animais.....	30
Figura 12 - Editar animais .....	31
Figura 13 - Excluir animais.....	31
Figura 14 - Cadastrar voluntário.....	32
Figura 15 - Editar voluntário .....	32
Figura 16 - Alterar dados do usuário .....	33
Figura 17 - Alterar status do usuário .....	33
Figura 18 - Pesquisar usuários.....	34
Figura 19 - Pesquisar animais.....	34
Figura 20 - Solicitar adoção.....	35
Figura 21 - Processos de aprovação.....	35
Figura 22 - DER .....	37
Figura 23 – Tela principal .....	41
Figura 24 - Tela conheça nossos animais.....	42
Figura 25 - Termo de responsabilidade.....	43
Figura 26 - Tela cadastro .....	43
Figura 27 - Tela contato .....	44
Figura 28 - Tela login .....	45
Figura 29 - Menu de navegação.....	45
Figura 30 - Tela adoções .....	46
Figura 31 - Visualizar cadastro.....	47
Figura 32 - Voluntários.....	47

Figura 33 - Excluir cadastro.....	48
Figura 34 - Tela gerenciar pessoas.....	48
Figura 35 - Tela animais.....	49
Figura 36 - Tela editar .....	49
Figura 37 - Tela animais.....	50
Figura 38 – Tela cadastro novo animal .....	51
Figura 39 – Tela relatório geral .....	52
Figura 40 - Relatório gerado.....	52
Figura 41 - Tela relatório animais.....	53
Figura 42 - Exemplo do relatório .....	53

## LISTA DE TABELAS

Tabela 1 - Divisão das etapas de desenvolvimento .....	9
Tabela 2 - Requisito funcional.....	18
Tabela 3 - Requisitos não funcionais do projeto.....	19
Tabela 4 - Animais .....	23
Tabela 5 - Endereços .....	24
Tabela 6 - Mensagens.....	24
Tabela 7 - Pessoas .....	25
Tabela 8 - Solicitações .....	25
Tabela 9 - Telefones .....	25
Tabela 10 - Usuários .....	26
Tabela 11 - Adoções .....	26
Tabela 12 - Migrations.....	26
Tabela 13 - Fotos .....	27
Tabela 14 - Agendas .....	27
Tabela 15 - Casos de teste .....	54

## INTRODUÇÃO

Em 2013, o Brasil ocupou a quarta colocação no ranking mundial, com 132,4 milhões de animais de estimação, e esse número cresce a cada ano. O número de cães e gatos representam juntos mais de 50% desse total (ABINPET, 2015). Em contrapartida o número crescente de animais abandonados nas vias públicas é cada vez maior, por motivo de mudanças, ou até mesmo viagens, sem ter com quem deixar os animais, seus donos acabam entregando-os em órgãos ou os abandonando nas ruas. Esses animais ficam sujeitos a contrair doenças, serem atropelados e sofrerem maus tratos (SANTANA e MARQUES, 2001). São motivos de abandono também, a falta de planejamento financeiro, o consumismo, que interferem no orçamento doméstico, uma vez que animais exigem cuidado e geram despesas e também a idade avançada do animal ou doenças (CARUSO, 2009).

Mesmo considerado maus tratos de animais e sendo crime, o abandono é uma prática facilitada, pela falta de fiscalização, pena não rigorosa, e pelos casos não denunciados, ou até mesmo pela falta de conhecimento da lei (GOMES, 2013). Consequentemente ocorre a superpopulação de animais errantes. A maioria desses animais não são castrados, e com uma gestação de aproximadamente 3 meses, a população cresce sem controle, levando milhares a ser sacrificados com a justificativa de eliminar doenças (SANTANA e MARQUES, 2001; GOMES, 2013).

São responsáveis pela captura de animais de rua, os centros de controle de zoonoses, com o objetivo de diminuir as doenças. No entanto, a falta de pessoal qualificado para o serviço, e a ausência de infraestrutura, levam muitos animais saudáveis a morte (DIAS, 2015).

Tentando resolver esse problema, surgem as associações de proteção aos animais, instituições sem fins lucrativos, responsáveis pelo resgate e tratamento dos animais, vítimas de maus tratos, recuperando-os até se tornarem novamente desejáveis pela sociedade (LAZARIN, 2014; GOMES, 2013). Embora muito relevante, essas instituições, sozinhas não conseguem atender toda a demanda, levando muitas instituições a abrigar mais animais do que a infraestrutura permite. Surgem então, os defensores de animais, que acabam transformando a própria casa em abrigo temporário, até os animais conseguirem um novo lar (LAZARIN, 2014).

Portanto, este trabalho buscará desenvolver um sistema, que auxilie as ONGs a gerenciar a adoção de animais abandonados, sob sua tutela, facilitando e

agilizando o processo de forma que possa atender um número maior de animais e consequentemente reduzir o número desses na rua, a proliferação de doenças e os maus tratos que sofrem. A partir de entrevista prévia com esse setor, o sistema buscará atender suas reais necessidades, e os interessados na adoção poderão fazer consulta online por espécie, por porte, encontrando o animal de sua preferência entre os cadastrados. Existem muitas informações relevantes a ser contempladas como vacinas, castração, se convive bem com outros animais, entre outras informações importantes ao adotante.

Uma hipótese desse trabalho é que o desenvolvimento de um sistema para gerenciar a adoção de animais em ONGs, pode facilitar os processos envolvidos, permitindo o atendimento a um número maior de animais e consequentemente, contribuindo para diminuir o número desses nas ruas, a proliferação de doenças e maus tratos. De acordo com a pesquisa levantada com as ONG's Vira-Lata Vira Vida contida na ATA1 e Associação dos Amigos dos Animais de Nova Odessa (AAANO) contida na ata2, as ONG's não possuem a quantidade de baias suficientes para manter os animais abrigados, ou seja, a ONG Vira-Lata Vira Vida só possui o espaço físico de armazenamento para 300 animais e a ONG AAANO possui a capacidade de 250 animais, mas atualmente está com mais de 600 animais para adoção. Por meio disso, o projeto do TCC irá contribuir para que sejam feitas feiras de adoção online de todos os animais cadastrados, com o intuito de diminuir a sobrecarga de animais em relação a capacidade de armazenamento das ONG's.

## 2. METODOLOGIA

### 2.1 Metodologia ágil

O desenvolvimento de *software* exige o uso de uma metodologia que gerencie seus processos e etapas. Considerando que o desenvolvimento não está imune a imprevistos, é necessário utilizar ferramentas que permitam resolvê-los através de mecanismos para ações corretivas. Existem algumas particularidades das metodologias ágeis, e uma delas é que são adaptativas e não preventivas, o que a diferencia da metodologia clássica na qual se pode criar um *software* inteiro e no final do projeto descobrir que houveram mudanças, as quais precisarão ser implementadas, o que se exigirá adaptações, fazendo com que o sistema tenha rupturas de *softwares* (SOARES, 2004).

Em contrapartida as metodologias ágeis permitem constante feedback, pois dessa forma quando houver alterações e mudanças, essas serão rapidamente implementadas nos requisitos afim de evitar o futuro retrabalho. Um dos pontos positivos das metodologias ágeis são os desenvolvimentos constantes, afim de possibilitar a visualização do *software* antes que esteja totalmente terminado, sendo que o cliente poderá dar a sua opinião durante o processo para que no final ele não se surpreenda com um *software* que não o atendesse. A fase de teste e integração do *software* são feitas de forma contínua durante o andamento do projeto, não necessitando de um tempo exclusivo para essa finalidade, pois elas já fazem parte do processo (SOARES, 2004)

As metodologias ágeis ficaram populares no ano de 2001, quando um grupo de especialistas dos EUA resolveram se unir para criar uma forma de melhorar a performance dos seus projetos, com isso foi criada a Aliança Ágil e o estabelecimento do Manifesto Ágil e esses possuem os conceitos e princípios comuns que fundamentaram o termo Desenvolvimento Ágil e passou a descrever alguns conceitos, como esses:

As interações entre os indivíduos são mais que processos e ferramentas;

Documentações são importantes, mas não mais que o *software* em funcionamento;

Colaboração com o cliente, em vez de pensar somente em negociações de contratos;

Atender as mudanças muito mais que seguir um plano (AGILE MANIFESTO, 2001).

As metodologias ágeis mais comuns são: *Extreme Programming* mais conhecida como XP, Scrum, *Feature-Driven Development (FDD)*, *Rational Unified Process (RUP)*, OpenUP e Kanban. Para o nosso projeto desenvolvimento de *software* escolhemos utilizar a metodologia Kanban que consiste em otimizar os processos e é orientada a eventos, sendo o seu foco é o trabalho em processo, sendo que as evoluções serão vistas de forma visual através dos quadros e seus respectivos status.

## 2.2 Metodologia Kanban

Entre as metodologias ágeis esse projeto optou por utilizar o método Kanban, conhecido desde 1940 quando a Toyota o desenvolveu aplicado a sua linha de produção, após observar o quanto as prateleiras de supermercados eram abastecidas segundo critérios visuais: o espaço disponível para os novos produtos, a quantidade que já saiu devido as vendas, quais estão parados e assim por diante, isso levou a uma nova forma de gerir seus processos de produção, que passou a utilizar um cartão visual (Kanban) para sinalizar etapas cumpridas ou não (FERNANDES E GODINHO FILHO, 2007)

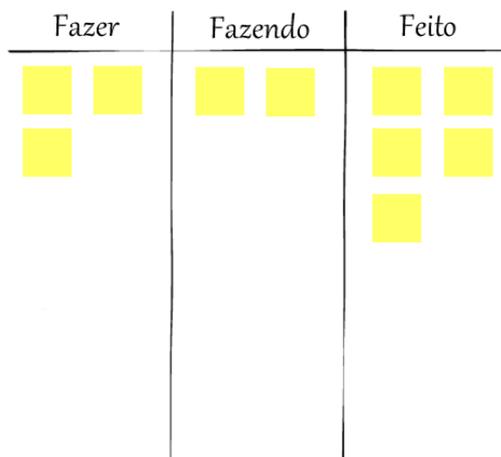
Na empresa o Kanban está atrelado ao conceito de *Pull Systems* (sistemas de produção puxados), contrário ao conceito de *Push Systems* (sistemas de produção empurrada), utilizado pela maioria das indústrias. Um sistema de produção empurrada é conhecido por iniciar a produção antes de ocorrer uma demanda, no caso, o processo recebe uma ordem de produção e executa, empurrando o resultado da operação atual para a operação seguinte. Um sistema de produção puxado se caracteriza por começar a produção quando um item é vendido, gerando demanda para a fabricação de outro, portanto cada parte do processo é alimentada pela demanda da etapa anterior (SLACK, 1999).

Para o desenvolvimento de *software*, o Kanban começou a ser utilizado em 2006 na Corbis (empresa fundada por Bill Gates). Decidiram que o fluxo de trabalho deveria ser visualizado, para isso utilizaram um quadro branco com alguns cartões colados pendurado na parede. A ideia havia surgido de um dos membros da equipe e foi bem aceita. Os resultados prévios do uso do Kanban na Corbis foram

apresentados nas conferências, “*Lean New Product Development*” e “*Agile 2007*”. Depois disso, o Kanban vem ganhando respeito na comunidade de desenvolvimento de *software*, devido a facilidade que permite a visualização das etapas de trabalho (ANDERSON, 2011).

Na prática o Kanban permite controlar o progresso das tarefas de forma visual. É, comumente, utilizado um quadro branco com *Post-it* colados, esses papéis representam as suas tarefas, ao concluí-las o papel é puxado para a etapa seguinte até que elas sejam finalizadas. Ao olhar para um quadro Kanban é fácil enxergar como o trabalho está, em qual etapa, permitindo não só comunicar o status, mas também dar e receber feedbacks. Um quadro Kanban simples:

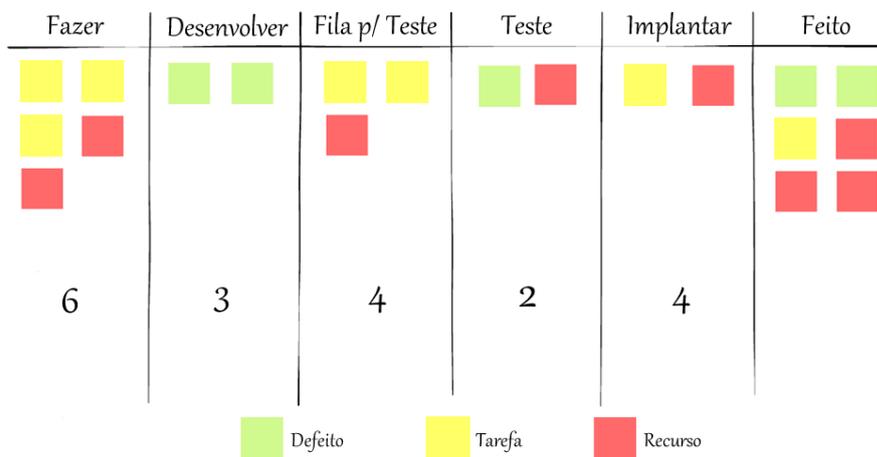
Figura 1 - Quadro Kanban



Fonte: <https://blog.mastertech.tech/negocios/kanban-a-metodologia-agil-mais-simples-e-presente-em-nossas-vidas/>

Um quadro mais detalhado:

Figura 2 - Quadro detalhado



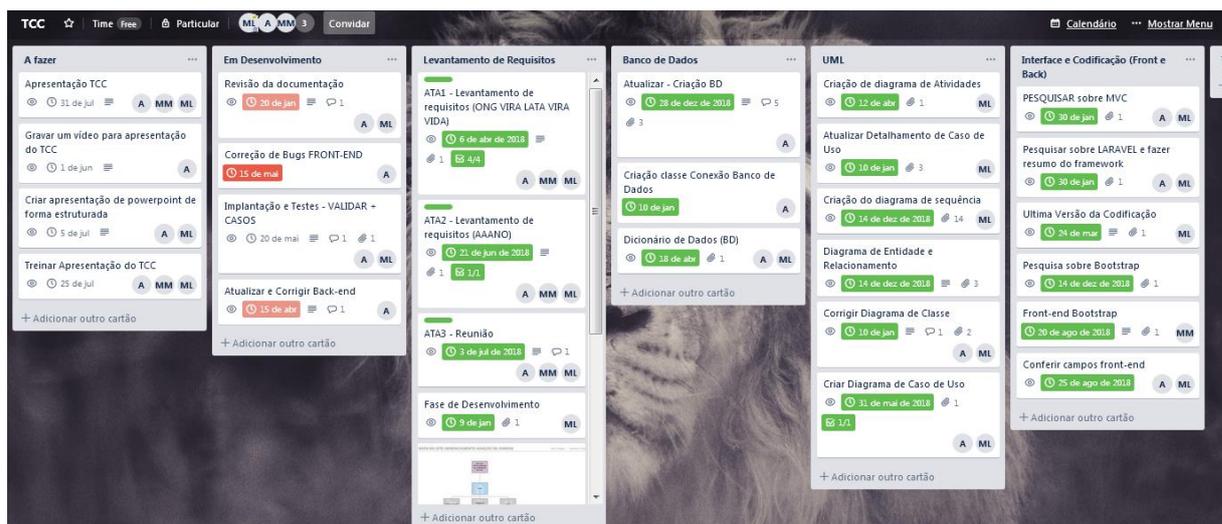
Fonte: <https://blog.mastertech.tech/negocios/kanban-a-metodologia-agil-mais-simples-e-presente-em-nossas-vidas/>

Na imagem acima nota-se que é possível dividir as tarefas por cores e por ordem de prioridade, o que vem em cima e antes deve ser realizado primeiro. Em cada coluna há um número que delimita a quantidade máxima de itens que podem ser adicionados para não sobrecarregar a equipe de trabalho.

Considerando os benefícios do Kanban nosso projeto utilizará essa metodologia para o desenvolvimento do *software* proposto.

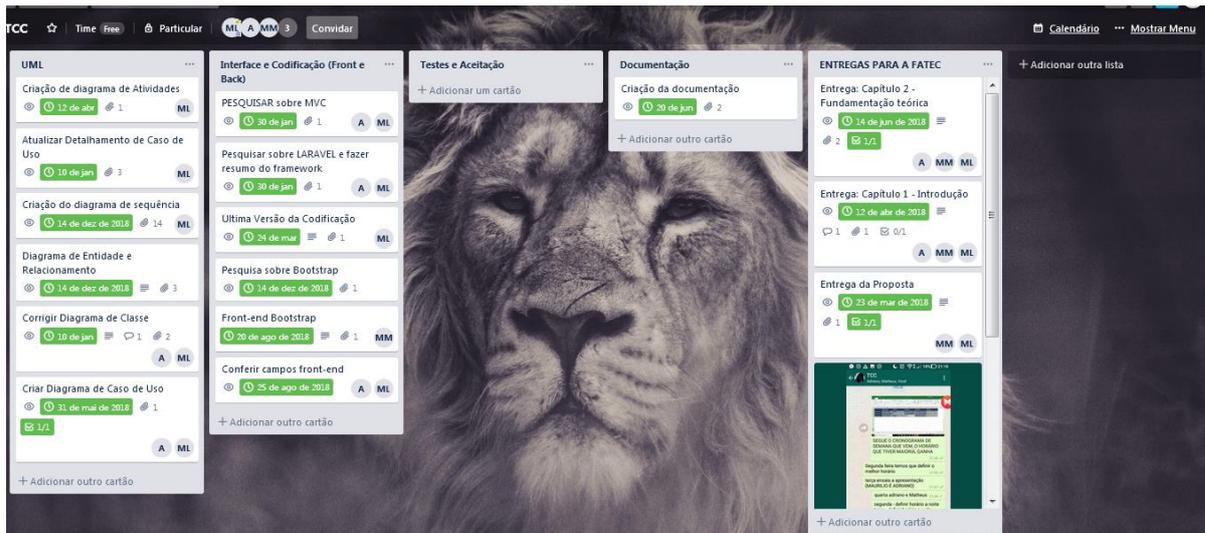
## 2.2.1 Cronograma e divisão do trabalho

Figura 3 - Cronograma Kanban



Fonte: Elaborado pelo autor (2018)

Figura 4 - Continuação cronograma Kanban



Fonte: Elaborado pelo autor (2018)

Tabela 1 - Divisão das etapas de desenvolvimento

Nome da tarefa	Q2 2018			Q3 2018			Q4 2018			Q1 2019			Q2 2019			Q3 2019		
	abr	mai	jun	jul	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul	ago	set
1 Criar Cronograma de Atividades																		
2 Levantamento de Requisitos																		
3 Criar Banco de Dados																		
4 Modelagem de Dados (UML)																		
5 Interface (Front End)																		
6 Codificação (Back End)																		
7 Testes de Aceitação																		
8 Documentação																		

Fonte: Elaborado pelo autor (2018)

## 2.3 Tecnologias empregadas

Optou-se pelo uso do PHP (*"Hypertext Preprocessor"*, originalmente *Personal Home Page*) para o desenvolvimento do sistema aqui proposto considerando que essa linguagem pode ser utilizada na maioria dos sistemas operacionais como o Microsoft Windows, Mac OS X e Linux, devido a suportar a maioria dos servidores, trabalhando tanto com módulo quanto como um processador CGI (*Computer Graphic Imagery*), não sendo limitado como outras linguagens que aceitam apenas HTML (*HyperText Markup Language*, expressão inglesa que significa "Linguagem de Marcação de Hipertexto") e CSS (*Cascading Style Sheets*), pois o PHP inclui imagens, tem suporte para o *Flash* e arquivos PDF, além de possuir amplo suporte para o banco de dados, ser conhecido por utilizar protocolos para o suporte de comunicação e funcionar bem para a maioria dos navegadores (NIEDERAUER, 2007).

Esse sistema fará uso também do HTML uma linguagem de marcação que possibilita organizar textos e imagens no desenvolvimento *web*, e criar *hiperlinks*, além de funcionar bem para a maior parte dos navegadores, ser considerada a melhor linguagem de estrutura de site, trabalhando com títulos, tabelas, textos, listas, fotos, e se relacionar muito bem com CSS e PHP (TREML, 2013).

Por fim será utilizado também o CSS que surgiu para auxiliar o HTML, quando esse começou a ficar confuso com a mistura de códigos e a necessidade de criar *sites* mais rápidos. O CSS traz a possibilidade de trabalhar com folhas de estilos na *web*, porém seu principal objetivo é agilizar e facilitar a leitura de páginas na *web*.

O CSS não é uma linguagem de programação nem uma linguagem de marcação. É uma sequência de declarações de propriedades e seus valores para a manipulação da forma com o conteúdo de uma página *web* será exibido (GOMES, 2010, pg. 11).

O CSS será utilizado, pois funciona bem para a maioria dos navegadores, será necessário para o padrão do HTML5 na estrutura do site, já que é a linguagem que o complementa melhor, e o uso dos dois fará com que a página do site fique mais leve e mais acessível. Além de que facilitará a acessibilidade e principalmente a separação em uma estrutura de documento, auxiliando a construção do site (QUIERELLE, 2012).

## 2.4 Banco de dados

Optou-se pelo uso do MySQL devido a sua facilidade de uso, interface simples e capacidade de rodar em vários sistemas operacionais. É propriedade da Oracle Corporation e distribuído em duas versões, uma fonte aberta *Community Edition* e outra paga pela *Standard Edition*, que é uma série de recursos aprimorados.

No entanto, é mais popular como fonte aberta, que pode ser baixado a partir do seu site. Sua edição tem todas as características necessárias para aplicativos seguros da *Web*. *Sites* como Google, Wikipedia, Facebook, YouTube e assim por diante usam o MySQL.

Suas principais vantagens são a sua natureza de código aberto, sendo livre pode ser usado por qualquer pessoa, sem qualquer licença ou autorização. Sob a GNU *General Public License* e o código fonte está disponível, o que permite aos desenvolvedores personalizar o código-fonte de acordo com as suas necessidades e

modificar o banco de dados para o seu uso, é fácil, rápido e de alto desempenho, possui compatibilidade cruzada, ou seja pode ser instalado em todos os principais sistemas operacionais, como UNIX, Solaris, Linux, bem como o Windows sem perda de desempenho. Ele também trabalha com vários desenvolvimentos *interfaces such* intransigentes como JDBC, ODBC, Pearl, Python, PHP, C ++, e muitos outros, devido ao desenvolvimento de API que é nele integrada.

Outro aspecto importante é o seu uso de memória, uma vez que evita vazamentos de memória, permitindo soluções eficazes de armazenamento. A única base de dados que pode conter até 8 Tera *Bytes*. Possui excelente segurança de dados, o que o torna um banco de dados seguro e confiável para soluções de nuvem populares como Microsoft Azure. Acessos não autorizados aos dados não é possível porque, ele é protegido por senha criptografada. Autoridade e direitos especiais são submetidos apenas a entrada autorizada. Os dados podem ser recuperados usando o *backup*, armazenamento em cache e arquivos no caso de serem danificados ou perdidos, portanto, a restauração de dados é garantida no MySQL (SANTOS, 2006).

## 2.5 Ferramenta PHPMyAdmin

Se escolheu para desenvolver esse sistema a ferramenta phpMyAdmin para gerenciar o MySQL na *web*, uma vez que o sistema a se desenvolver será feito em linguagem *Web*, pensando em uma comunicação mais rápida e fluente para manipulação de dados. É uma ferramenta de *software* gratuita escrita em PHP, destinada para manipulação SQL via *web*. O phpMyAdmin suporta operações para gerenciamento de banco de dados, como, tabelas, colunas, usuários, permissões etc.). A seguir, algumas das funcionalidades suportadas pelo phpMyAdmin:

- criar, navegar, editar e eliminar bancos de dados, tabelas, *views*, colunas e índices;
- criar, copiar, eliminar, renomear e alterar bancos de dados, tabelas, colunas e índices;
- manutenção de servidores, bancos de dados e tabelas, com propostas para configurações de servidor;
- executar, editar e marcar qualquer declaração SQL, até mesmo consultas em lote;
- carregar arquivos de texto em tabelas;

- importar dados e estruturas MySQL de planilhas *OpenDocument*, assim como de arquivos XML, CSV e SQL;
- administrar vários servidores;
- adicionar, editar e remover contas de usuário e privilégios do MySQL;
- criar gráficos PDF do layout do seu banco de dados;
- pesquisar em um banco de dados inteiro ou em apenas em uma parte;
- controlar alterações em bancos de dados, tabelas e visualizações;
- suporte a MySQL, a extensão melhorada do MySQL;
- criar, editar, chamar, exportar e apagar processos e funções armazenadas;
- criar, editar, exportar e apagar eventos e gatilhos;
- disponível em 80 linguagens diferentes (<https://www.phpmyadmin.net/>)

Considerando que essa ferramenta possui boas medidas de segurança e é de fácil manuseio, optou-se por seu uso.

## 2.6 Servidor *Web Apache*

Para o servidor *web* funcionar com o PHP é necessário o Apache. Para Davis e Phillips (2008, p. 4) “Apache é um servidor *web* que transforma requisições dos navegadores em resultados em páginas da *web* e sabe como processar o código PHP”. Conhecido também como um projeto de código aberto (*open source*), sem ele não seria possível utilizar o PHP, já que ele liga o cliente que faz requisições e a linguagem no servidor que processa, através de módulos próprios, um deles exclusivo para o PHP, além de armazenar todos os arquivos do *front-end* que será enviado ao cliente.

Como um projeto de código aberto, tem todas as vantagens que isso proporciona, especialmente para usuários avançados que manipulam o código fonte para sua utilização. Possui também com grande material didático e várias comunidades na internet.

Existem outras opções para substituir o Apache, mas ele tem a vantagem de não ter custo, e ter uma grande compatibilidade com sistemas operacionais diversos, o que permite que o servidor possa utilizar tanto um sistema operacional Linux como um Windows.

Para o desenvolvimento do protótipo utilizou-se a ferramenta XAMPP versão 3.2.2, que provê o serviço Apache e o MySQL, em um sistema operacional Windows 10

Professional versão 1709, o que não interfere em qual sistema a estação cliente utilizará, todos conseguirão acessar e fazer requisições ao servidor da mesma forma.

Este ambiente de desenvolvimento pode ser considerado um ambiente de teste, pois a ferramenta foi idealizada para funcionar tanto em servidor Windows quanto um Linux, porém apenas foi testada neste ambiente.

## 2.7 Framework - Laravel

Um *framework* é uma estrutura, ou plataforma, em que você pode escolher e desenvolver um *software*. Ao contrário de outros *frameworks*, o Laravel garante grande performance e escalabilidade.

Começando pela fase das ferramentas e compatibilidade, separamos os frameworks em três categorias: Vanguarda, Consagrados e Revelações. O grupo de Vanguarda contém aqueles *frameworks* com uma base de código mais madura, que são utilizados para o desenvolvimento há vários anos. Este *framework* é uma plataforma estável, e é compatível com as versões mais antigas de PHP. Parte de um projeto que começou em 2008, ele continua sendo desenvolvido, mas deve uma queda de popularidade devido as novas opções disponíveis na comunidade de desenvolvimento.

As opções consagradas são aquelas que possuem grande uso, uma comunidade de desenvolvimento ativa, base de códigos testada e são frequentemente escolhidos para novos *sites*. O Zend faz quase tudo, e mostra serviço. Muitas lojas e projetos comerciais são construídos por ele, já que ele é atualizado constantemente, além de testado.

Este framework tem um número extenso de referências de terceiros a seu dispor, como o *PHPUnit* (usado para testes), *SwiftMailer* (e-mail), e *Twig* (templates). Eles possuem um desenvolvimento de qualidade, e tudo é facilmente configurável. As revelações foram adotadas no desenvolvimento de grandes projetos há poucos anos, tem novas ferramentas PHP, e um desenvolvimento inteligente.

*FuelPHP* – A equipe de que construiu este framework adotou grandes números de ideias vindas do *CodeIgniter*, assim como muitas inspirações de outros frameworks. Como resultado ele se apresenta como um produto mais leve e modular. Ele possui uma boa documentação, e um grande número dos desenvolvedores que utilizavam o *CodeIgniter* migraram para este *framework*.

Laravel – Este *framework* já existe há algum tempo, mas parece que só se estabeleceu em sua terceira versão, e sempre é remodelado em versões subsequentes. Além de ser, inicialmente, inspirado pelo *CodeIgniter* e alguns outros *frameworks*, ele tomou um novo caminho em sua jornada de desenvolvimento.

O Laravel é um *framework* PHP avançado, que incorpora um ecossistema amplo com hospedagem instantânea e desenvolvimento. Uma das partes cruciais do desenvolvimento de qualquer aplicativo *web* é a autenticação, e os desenvolvedores costumam gastar muito tempo gerando estes códigos. O Laravel facilita este processo por organiza a lógica de autorização, enquanto organiza também o acesso aos recursos disponíveis. Além disso, um dos últimos *redesigns* do Laravel oferece um código de validação pré-construído, e que vai diminuir exponencialmente o tempo necessário para compor estes grandes códigos.

Por outro lado, a lógica de aplicação pode ser executada em qualquer aplicativo, através de controles ou diretamente, utilizando a sintaxe. Em resumo, o Laravel oferece flexibilidade para os desenvolvedores, assegurando um ecossistema adaptável, perfeito para a construção de aplicativos empresariais ou pequenos *sites*.

O *Blade* representa um modelo simples, porém dinâmico, de mecanismo de *template*, construído junto ao Laravel, para utilização de PHP puro. Os outros *frameworks* restringem que seus usuários utilizem códigos desta forma. Já o Laravel dá suporte a esta ferramenta.

O *Artisan* é uma interface disponibilizada pelo Laravel. Ela oferece vários comandos prontos, para ajudar com o avanço do desenvolvimento de seu projeto. Tipicamente, os desenvolvedores interagem com o *framework* utilizando linhas de comando manuais. Através do *Artisan*, ele pode criar várias tarefas tediosas da programação com facilidade e precisão.

O Laracast é um tutorial de aprendizado de grande qualidade, que incorpora vídeos grátis e pagos, que foram criados especificamente para ajudar os usuários a se acostumarem com os principais conceitos do Laravel. Os vídeos são criados por Jeffrey Way, e são claros e concisos, além de satisfazer o principal propósito.

Todos os desenvolvedores que utilizam o Laravel podem confirmar a existência de uma documentação detalhada. Ele utiliza o *Flysystem* para conseguir isso, e todas as opções de documentação podem ser organizadas em um *framework* remoto.

Quanto ao MVC (*Model View Controller*), o Laravel utiliza o padrão *model-view-controller*. Ela entrega várias funções já incorporadas, e faz com que seja ainda mais fácil desenvolver seus projetos.

O Laravel garante uma segurança extrema aos seus usuários, já que nunca salva suas senhas, no formato de texto, em sua base de dados, utilizando um algoritmo para a criação de uma senha criptografada. Além disso, a homologação do SQL garante uma segurança ainda mais avançada.

Manter o banco de dados em sincronia com as ferramentas de desenvolvimento pode ser uma tarefa cansativa, mas com as migrações de banco de dados do Laravel, isso pode ser feito sem muito esforço. Mais algumas vantagens:

- Facilidade e simplicidade de uso;
- Documentação completa e de fácil compreensão;
- Estruturação organizada do projeto;
- Aumento da produtividade e reaproveitamento de código;
- Baixa curva de aprendizado;
- Segue boas práticas de desenvolvimento.

O desenvolvimento através do Laravel posiciona este *framework* entre os melhores da atualidade, já que sempre consegue sucesso em satisfazer todas as principais demandas do desenvolvimento *web*. Qualquer empresa que trabalha com o desenvolvimento de aplicativos pode confirmar as contribuições das ferramentas citadas dentro de um grande projeto. Apesar disso, a escolha de um *framework* deve ir além de apenas ferramentas. Ela deve incorporar um conhecimento profundo do mercado, desafios, requisitos e mais.

## 2.8 Bootstrap

Ele foi desenvolvido por Jacob Thorton e Mark Otto, engenheiros do Twitter, como uma tentativa de resolver incompatibilidades dentro da própria equipe.

O intuito era otimizar o desenvolvimento de sua plataforma por meio da adoção de uma estrutura única. Isto reduziria inconsistências entre as diversas formas de se codificar, que variam de profissional para profissional. E a tentativa deu tão certo que eles perceberam o grande potencial da ferramenta, lançando-a no GitHub como um *software* livre.

Na prática, a principal aplicação do Bootstrap seria na criação de *sites* responsivos (*mobile*). Com o Bootstrap, o profissional já não tem mais que perder tanto tempo digitando toda uma linha de CSS novamente. Esta facilidade se deve ao fato de que ele possui vários plugins em JavaScript (jQuery) que tornam o seu dia-a-dia muito mais fácil. Com inúmeras bibliotecas prontas disponíveis, o trabalho que o desenvolvedor tem é de, praticamente, só as incluir em seus projetos.

As principais características que fazem o Bootstrap ser um diferencial, são:

- Biblioteca de componentes
- Reuso de código
- Padrão visual
- Responsividade

Talvez uma das características mais interessantes deste framework é o fato dele possuir uma extensa biblioteca de componentes, como ícones, caixas de texto, painéis e cores em *links*.

Assim como é objetivo de todo *framework*, o Bootstrap permite que menos código possa ser escrito, já que ele entrega uma série de formatações visuais prontas. A única necessidade do desenvolvedor é saber qual a classe ele deve chamar.

Além de padronizar uma série de parâmetros, o Bootstrap é uma das formas mais fáceis de deixar um *site* responsivo. Esse termo, significa que um *site* é adaptável a diferentes tipos de tela, como celular, *tablet* ou computador.

### 3. ENGENHARIA DE SOFTWARE

Este capítulo é destinado a apresentar o processo de Engenharia de *Software* do sistema aqui proposto. Define-se Engenharia de *Software* segundo Fritz Bauer (*apud* PRESSMAN, 2011, p. 39), como “*o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais*”.

Aqui então se incluem métodos, processos de especificação, técnicas, modelagem de dados, arquitetura, desenvolvimento, manutenção e documentação, para facilitar o desenvolvimento e assegurar a qualidade do *software*.

Na modelagem de dados e documentação optou-se na elaboração do sistema, pelo levantamento de requisitos, diagramas de casos de uso, classes e

atividades, e a modelagem de banco de dados utilizando o diagrama de entidade e relacionamento e o padrão de arquitetura de *software* MVC. Como ponto inicial escolheu-se o levantamento de requisitos, considerando que oferece uma visão geral das funcionalidades necessárias ao sistema, sem a necessidade de se especificar nesse momento, como essas funcionalidades serão implementadas.

Posteriormente ao levantamento dos requisitos, são apresentados os diagramas da linguagem UML, definidos por Guedes (2009, p. 19) como:

“[...] Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos. [...] linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de *software* [...]”.

A opção por se utilizar a orientação a objetos exige que se crie, no mínimo, os diagramas de casos de uso, classes e atividades, que serão base para a etapa de desenvolvimento. O sistema também é documentado, para facilitar a interpretação por parte de outros programadores, e permitir a manutenção do sistema. Alguns dos diagramas apresentados foram elaborados após o desenvolvimento do sistema, como por exemplo, o diagrama de sequência.

Na modelagem de dados utilizou-se o diagrama ER, considerado a estrutura que sustenta a aplicação, portanto é necessário que essa modelagem combine perfeitamente com o desenvolvimento da aplicação. Na sequência são apresentadas as definições do MVC, padrão de arquitetura de *software* adotado neste projeto, responsável na organização, especificação, desenvolvimento e manutenção do sistema.

No que se refere a metodologia de Engenharia de *Software*, optou-se pelo uso das metodologias ágeis, o que facilitou a equipe seguir as etapas necessárias ao desenvolvimento desse projeto.

A Engenharia de *Software* permitiu um desenvolvimento padrão, pensando nas manutenções posteriores e implementações futuras. A seguir as partes que compuseram esta etapa.

### **3.1 Levantamento de requisitos**

O início de um novo sistema, exige primeiramente o levantamento e análise de requisitos para buscar compreender o que será feito, a partir da análise da real necessidade, sobre essa fase Guedes (2009, p. 22) informa que:

As etapas de levantamento e análise de requisitos trabalham com o domínio do problema e tentam determinar ‘o que’ o *software* deve fazer e se é realmente possível desenvolver o *software* solicitado. Na etapa de levantamento de requisitos, o engenheiro de *software* busca compreender

as necessidades do usuário e o que ele deseja que o sistema a ser desenvolvido realize.

É um processo cíclico, no qual se volta buscando comparação do que foi definido e o que foi feito, adicionando novos requisitos ou retirando requisitos que já não são mais considerados parte do escopo do projeto.

A seguir foram separados os requisitos funcionais, aqueles que representam ações do sistema, e requisitos não funcionais, aqueles que apresentam características do sistema.

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do *software*. (SOMMERVILLE, 2007). A Tabela 1 apresenta os requisitos funcionais deste projeto.

Tabela 2 - Requisito funcional

Identificação	Requisito Funcional
RF001	Um dos requisitos funcionais é o cadastro de usuários, com o qual pode-se realizar através de uma pessoa que não tenha cadastro no sistema, ou através de um usuário com permissões de usuário administrador.
RF002	Ao realizar o cadastro, o usuário vai usar seu e-mail como <i>login</i> e vai ter uma senha própria para acessar o <i>software</i> . Em caso de perda dos dados de acesso, somente o usuário administrado/super administrador poderá resetar a senha. <i>Login</i> é um dos dados de segurança, que preserva a integração dos dados do usuário e as suas solicitações/histórico.
RF003	O sistema deve prover condições para que o usuário com o perfil voluntário escolha o animal que deverá ser adotado efetue a solicitação de adoção sem erro e após a análise do administrador do <i>site</i> o solicitante receba uma confirmação do <i>site</i> informando que a solicitação de adoção foi aprovada ou reprovada.
RF004	O usuário com o perfil de administrador/super administrador será responsável pelo gerenciamento dos cadastros de animais, cadastros de usuários e suas permissões, analisar solicitações de adoções de animais podendo aprovar ou rejeitar
RF005	O sistema deve prover a visualização de todos os animais cadastrados na página principal, para que qualquer pessoa visualize os animais e possuam interesse em realizar uma adoção.
RF006	Algumas alterações menores podem ser feitas nos dados dos animais e usuários. Esse requisito está diretamente relacionado com o requisito RF004.

Fonte: Elaborado pelo autor (2018)

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema.” (SOMMERVILLE, 2007). A Tabela 2 apresenta os requisitos não funcionais deste projeto.

Tabela 3 - Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria
RNF001	Cada senha de usuário é pessoal e intransferível, portanto, é importante não repassar senha e palavra passe de um usuário para o outro. Pois existem permissões de acessos diferentes para cada tipo de usuário e isso assegura a integração dos dados do sistema para que sejam fidedignos.	Segurança
RNF002	Como o usuário irá utilizar o celular ou computador, o programa deve ser ágil em todas as suas funcionalidades, evitando problemas de carregamento de fotos, acesso demorado, por isso um aplicativo leve e rápido será de grande importância para o usuário.	Desempenho
RNF003	Não são todas as pessoas que estão habituadas a usar a internet ou conhecem algum <i>site</i> de adoções de animais, por isso o <i>software web</i> deve ser bem intuitivo para que qualquer pessoa possa efetuar uma solicitação de adoção.	Usabilidade

Fonte: Elaborado pelo autor (2018)

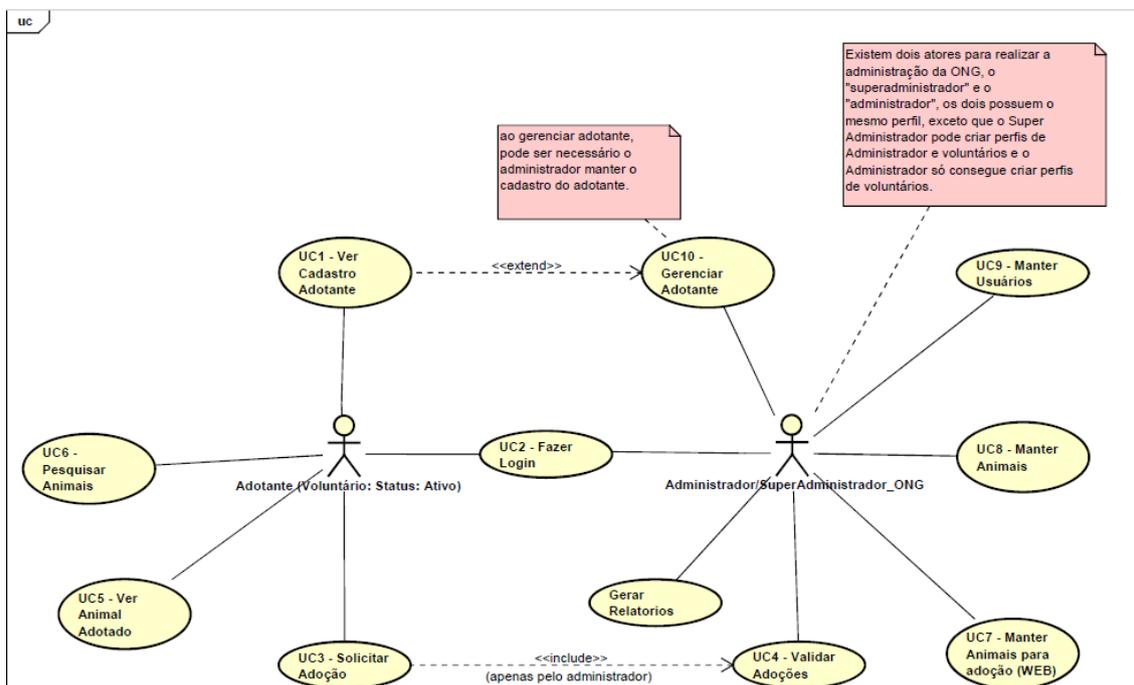
### 3.2 Diagrama de casos de uso

O diagrama de casos de uso “*mostra conceitualmente o conjunto de funções que o sistema deve executar para atender aos requisitos do cliente, servindo como um contrato entre o cliente e o desenvolvedor*” (LIMA, 2010, p. 57).

Na figura 5 temos o diagrama de casos de uso desenvolvido a partir dos requisitos funcionais levantados anteriormente, apresentando uma visão da interação dos usuários com o sistema. O diagrama foi desenvolvido com a ferramenta de criação de diagramas UML Astah *Community*.

## Diagrama de Caso de Uso - UC – Software para Gestão de Adoção de Animais

Figura 5 - Diagrama de caso de uso



Fonte: Elaborado pelo autor (2018)

### Especificação de Caso de Uso – UC

- Atores: Adotante (Voluntário: Status: Ativo) e Administrador e Super Administrador.

Adotante - Usuário no sistema que irá realizar o cadastro para realizar a solicitação de adoção de animais.

Administrador – Usuário no sistema que será o administrador do *software* com as permissões de realizar a validações das solicitações de adoções, manter animais para adoções (cadastro), manter animais para adoções *WEB* (no *site*), manter usuários como voluntários na ONG e gerenciar cadastro de adotante. Este perfil só poderá criar usuários do tipo “Voluntários”.

Super Administrador – este usuário terá todos os acessos do perfil Administrador e além disso poderá criar novos perfis do tipo: “Administrador”, “Super Administrador” e “Voluntários”.

**UC1 – Manter Cadastro Adotante**

Nesse caso de uso, serve para o adotante realizar o cadastro no sistema, podendo realizar a visualização do seu cadastro.

Obs.: O Administrador\_ONG poderá realizar a visualização e a manutenção dos cadastros, podendo cadastrar, editar e inativar.

**UC2 – Fazer Login**

Nesse caso de uso, serve para os usuários Adotante e Administrador\_ONG realizar o acesso ao sistema.

**UC3 - Solicitar Adoção**

Nesse caso de uso, serve para quando o usuário Adotante escolher um animal, enviar uma solicitação para que o Administrador\_ONG realize a validação.

**UC4 – Validar Adoções**

Nesse caso de uso, serve para realizar a aprovação ou a rejeição da solicitação de adoção realizada pelo Adotante, quem irá realizar a validação é o Administrador\_ONG.

**UC5 – Ver o Animal Adotado**

Nesse caso de uso, serve para o adotante verificar o animal que foi adotado.

**UC6 – Pesquisar Animais**

Nesse caso de uso, serve para o Adotante realizar pesquisas referente a características dos animais cadastrados no sistema.

**UC7 – Manter Animais para Adoção (WEB)**

Nesse caso de uso, serve para o Administrador\_ONG realizar o gerenciamento de animais no sistema, realizando a exclusão dos animais que já foram adotados e a inserção de novos animais para adoção.

### **UC8 – Manter Animais**

Nesse caso de uso, serve para o Administrador\_ONG realizar o cadastro dos animais na ONG, podendo também realizar funções com a finalidade de alterar, editar e excluir.

### **UC9 – Manter Usuários**

Nesse caso de uso, serve para o Administrador\_ONG realizar o cadastro de usuários na ONG, com as permissões necessárias de perfil.

### **UC10 – Gerenciar Adotante**

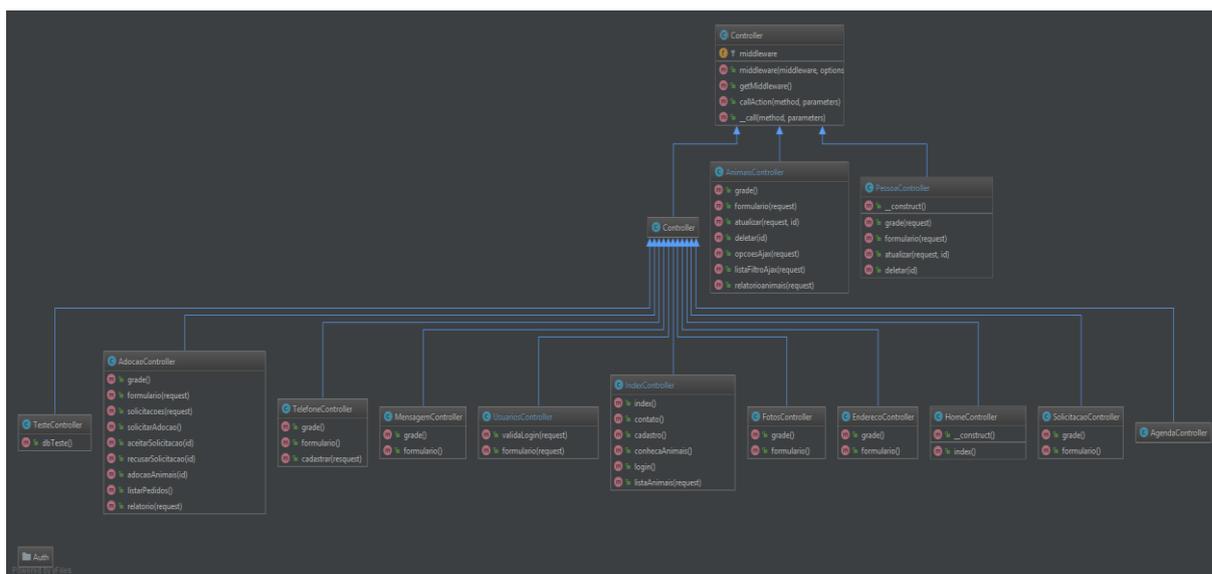
Nesse caso de uso, serve para o Administrador\_ONG realizar o gerenciamento de adotantes, podendo realizar as funções de cadastrar, editar, inativar e excluir adotante.

## **3.3. Diagrama de classes**

O diagrama de classes é específico para o uso de orientação a objeto, que tem como base as classes de objetos. Utilizado tanto anterior ao desenvolvimento, definindo as classes que serão implementadas, com seus atributos e métodos, quanto posteriormente para a documentação dessas classes, o que permitirá compreender as classes do sistema.

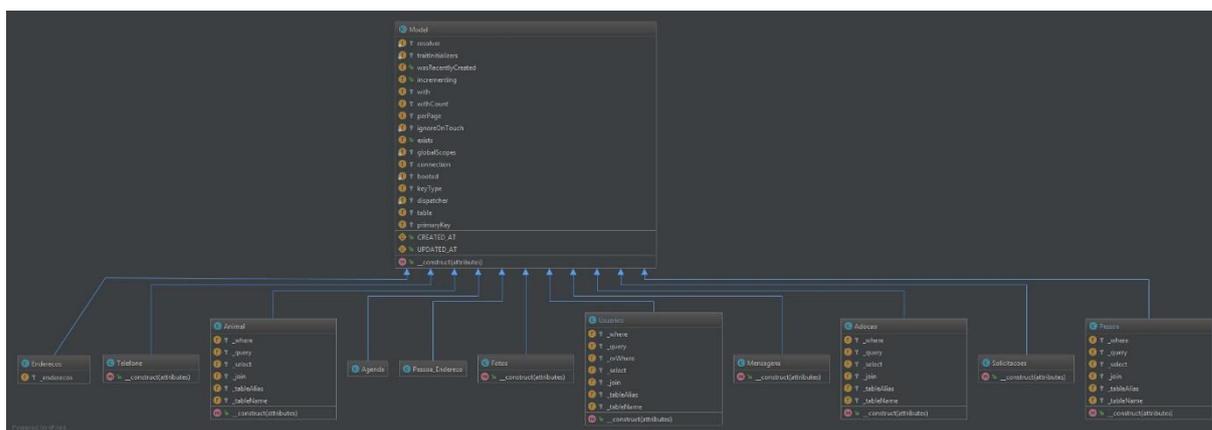
Guedes (2009, p. 106) entende sobre o diagrama de classes que esse é “[...] uma visão estática de como as classes estão organizadas, preocupando-se em como definir a estrutura lógica das mesmas. [...] serve ainda como base para a construção da maioria dos outros diagramas da linguagem UML”.

Figura 6 - Diagrama de classe controller



Fonte: Elaborado pelo autor (2018)

Figura 7 - Diagrama de classe gerenciamento



Fonte: Elaborado pelo autor (2018)

### 3.4. Dicionário de dados

O dicionário de dados é um documento no qual, estão as informações sobre as tabelas, seu conteúdo, bem como os relacionamentos entre os seus elementos.

Tabela 4 - Animais

Tabela: animais						
	CAMPO	DESCRIÇÃO	TIPO	TAM	DEC	DEC
PK	Id	Código do Animal	INT	10	-	NOT NULL AUTO_INCREMENT
	idade	Idade do Animal	INT	11	-	NOT NULL
	Raca	Raça do Animal	VARCHAR	50	-	NOT NULL
	nome	Nome do Animal	VARCHAR	100	-	NOT NULL
	Cor	Cor do Animal	VARCHAR	20	-	NOT NULL
	Porte	Porte do Animal	VARCHAR	20	-	NOT NULL

	Deficiencia	Deficiência do Animal	CHAR	1	-	NOT NULL
	Sexo	Sexo do Animal	CHAR	1	-	NOT NULL
	vacinado	Vacina do Animal	CHAR	1	-	NOT NULL
	dt_registro	Data de entrada do animal na ONG	DATE	-	-	NOT NULL
	Castrado	Se o animal é castrado ou não	CHAR	1	-	NOT NULL
	patologia	Detalhes da patologia	VARCHAR	200	-	NOT NULL
	Situação	Situação do animal	TEXT	-	-	NOT NULL
	Status	Adotado ou não	VARCHAR	1		NOT NULL
	Descrição	Descrição do animal	TEXT	-		NOT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 5 - Endereços

Tabela: enderecos						
	CAMPO	DESCRIÇÃO	TIPO	TAM	DEC	OBS
<b>PK</b>	Id	Código do endereço do adotante	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código da pessoa adotante	INT	10	-	NOT NULL
	bairro	Bairro do adotante	VARCHAR	50	-	NOT NULL
	numero	Número da residência do adotante	VARCHAR	5	-	NOT NULL
	rua	Rua do adotante	VARCHAR	100	-	NOT NULL
	Cidade	cidade do adotante	VARCHAR	50	-	NOT NULL
	estado	Estado do adotante	CHAR	2	-	NOT NULL
	resumo	Detalhes do endereço do adotante	VARCHAR	200	-	NOT NULL
	Cep	CEP do adotante	VARCHAR	9		NOT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL
	Tipomoradia	Tipo de lugar onde mora	VARCHAR	250	-	NOT NULL
	Temquintal	Se a moradia possui quintal	VARCHAR	250	-	NOT NULL

Fonte: Elaborado pelo autor (2018)

Tabela 6 - Mensagens

Tabela: mensagens						
	CAMPO	DESCRIÇÃO	TIPO	TAM	DEC	OBS
<b>PK</b>	Id	Código da mensagem	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_rem	Código do remetente	INT	10	-	NOT NULL (REFERENCES id_pessoa)

<b>FK</b>	id_dest	Código do destinatário	INT	10	-	NOT NULL (REFERENCES id_pessoa)
	mensagem	Descrição da mensagem	TEXT	-	-	NOT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 7 - Pessoas

<b>Tabela: pessoas</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código da pessoa	INT	10	-	NOT NULL AUTO_INCREMENT
	Nome	Nome da Pessoa	VARCHAR	255	-	NOT NULL
	Sexo	Sexo da pessoa	VARCHAR	1		
	Código	Código da pessoa	VARCHAR	255	-	NOT NULL
	dt_nasc	Data de Nascimento da Pessoa	DATE	-	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 8 - Solicitações

<b>Tabela: solicitacoes</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código da solicitação	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código do adotante	INT	10	-	DEFAULT NULL
	data	Data da solicitação da adoção	DATE	-	-	DEFAULT NULL
	descricao	Descrição da solicitação	VARCHAR	255	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 9 - Telefones

<b>Tabela: telefones</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código do Telefone	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código do adotante	INT	10	-	DEFAULT NULL
	numero	Número do telefone	VARCHAR	12	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 10 - Usuários

<b>Tabela: usuários</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código do usuário no site	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código do adotante ou funcionário	INT	10	-	DEFAULT NULL
	nivel	Nível de permissão de usuário	CHAR	2	-	DEFAULT NULL
	email	E-mail do usuário	VARCHAR	100	-	DEFAULT NULL
	senha	Senha do usuário	VARCHAR	20	-	DEFAULT NULL
	token	token do usuário (esqueceu a senha)	VARCHAR	255	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL
	Foto	Foto do usuário	VARCHAR	200	-	DEFAULT NULL
	Status	Usuário ativo ou não	VARCHAR	1	-	DEFAULT NULL

Fonte: Elaborado pelo autor (2018)

Tabela 11 - Adoções

<b>Tabela: adocaos</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código do usuário no site	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código do adotante ou funcionário	INT	10	-	DEFAULT NULL
<b>FK</b>	id_animal	Código do animal	INT	10	-	DEFAULT NULL
	Adotado	Se animal está adotado ou não	VARCHAR	1	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL
	data_adocao	Data da adoção do animal	DATE	-	-	DEFAULT NULL
	data_recusa	Data recusa para adotar animal	DATE	-	-	DEFAULT NULL
	data_pedido	Data solicitação adoção	DATE	-	-	DEFAULT NULL

Fonte: Elaborado pelo autor (2018)

Tabela 12 - Migrations

<b>Tabela: migrations</b>						
	<b>CAMPO</b>	<b>DESCRIÇÃO</b>	<b>TIPO</b>	<b>TAM</b>	<b>DEC</b>	<b>DEC</b>
<b>PK</b>	Id	Código de migração	INT	10	-	NOT NULL AUTO_INCREMENT
	Migration	Migração de dados	VARCHAR	255	-	DEFAULT NULL
	Batch	-	INT	11	-	DEFAULT NULL

Fonte: Elaborado pelo autor (2018)

Tabela 13 - Fotos

Tabela: fotos						
	CAMPO	DESCRIÇÃO	TIPO	TAM	DEC	DEC
<b>PK</b>	Id	Código da foto no sistema	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_animal	Código do animal	INT	10	-	DEFAULT NULL
	Caminho	Caminho da foto no sistema	VARCHAR	200	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL
	updated_at	Fez upload/alteração dos dados.	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

Tabela 14 - Agendas

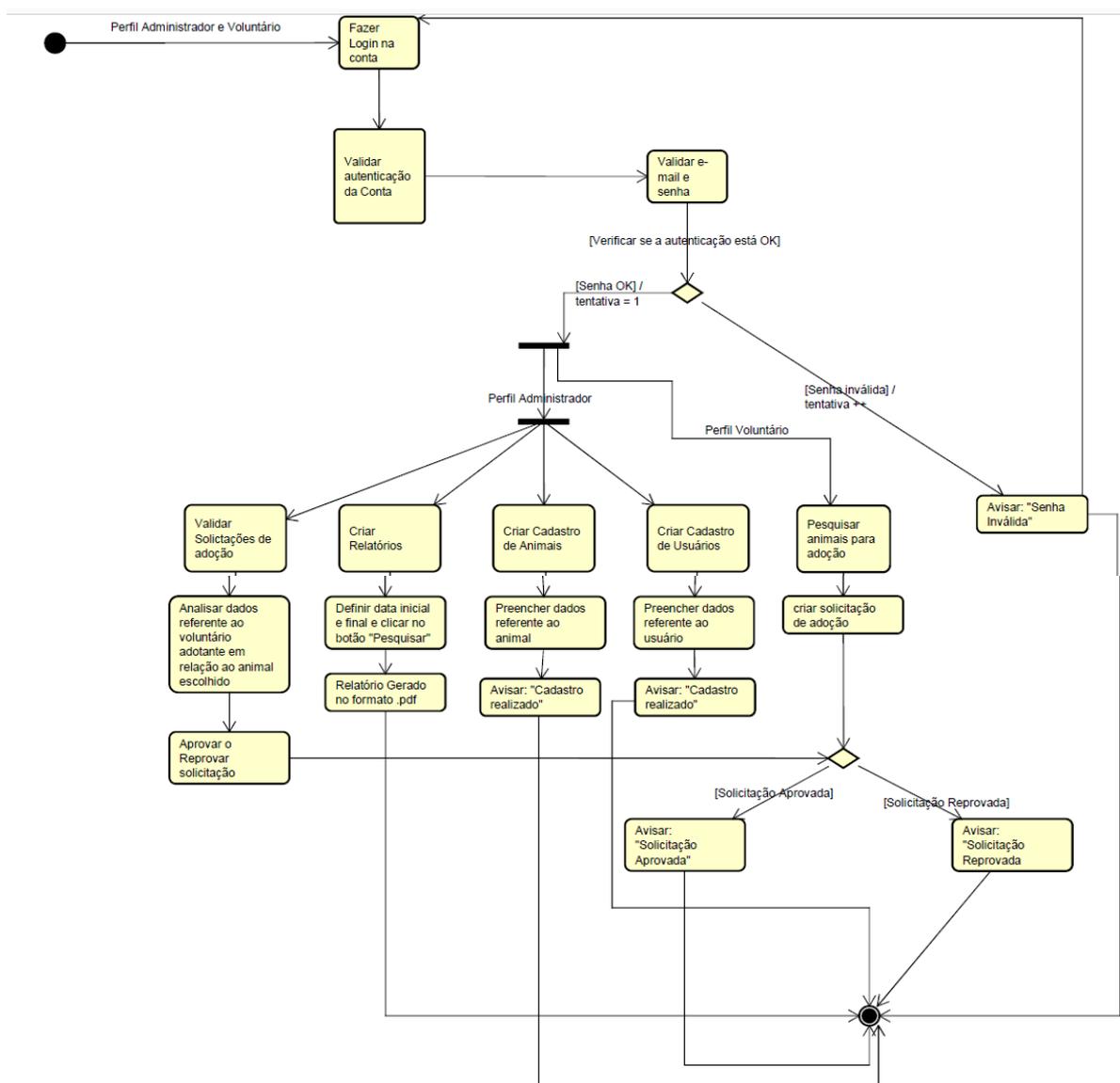
Tabela: agendas						
	CAMPO	DESCRIÇÃO	TIPO	TAM	DEC	DEC
<b>PK</b>	Id	Código da foto no sistema	INT	10	-	NOT NULL AUTO_INCREMENT
<b>FK</b>	id_pessoa	Código da pessoa	INT	10	-	DEFAULT NULL
	Date	Data agendada	DATE	-	-	DEFAULT NULL
	created_at	Criou os dados	TIMESTAMP	-	-	NULL

Fonte: Elaborado pelo autor (2018)

### 3.5. Diagrama de atividades

Segundo Guedes, esse diagrama é “utilizado, como o próprio nome já diz, para modelar atividades, que podem ser um método ou um algoritmo, ou mesmo um processo completo”. (GUEDES, 2009, p. 285).

Figura 8 - Diagrama de atividades

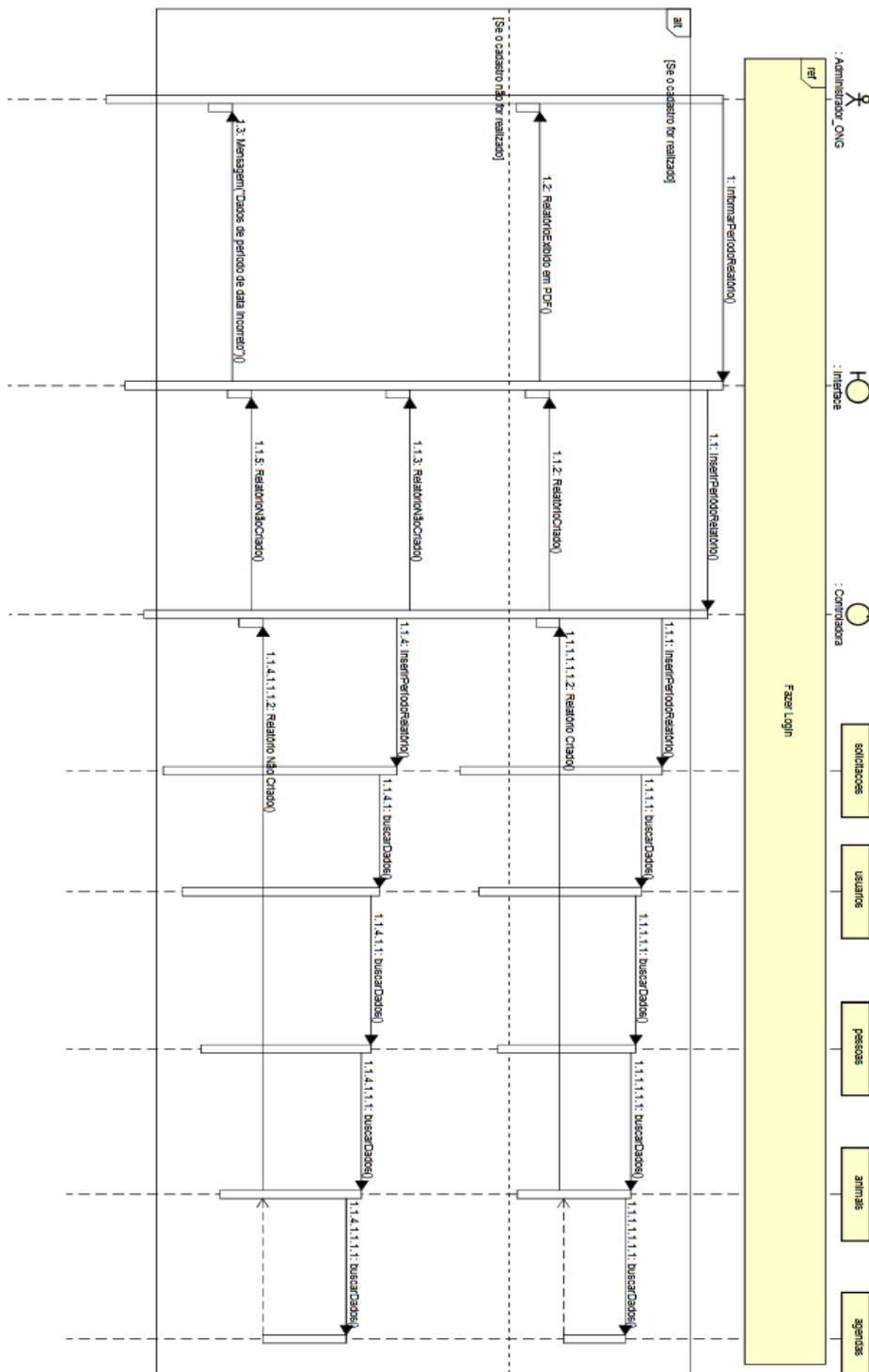


Fonte: Elaborado pelo autor (2018)

### 3.6. Diagrama de sequencias

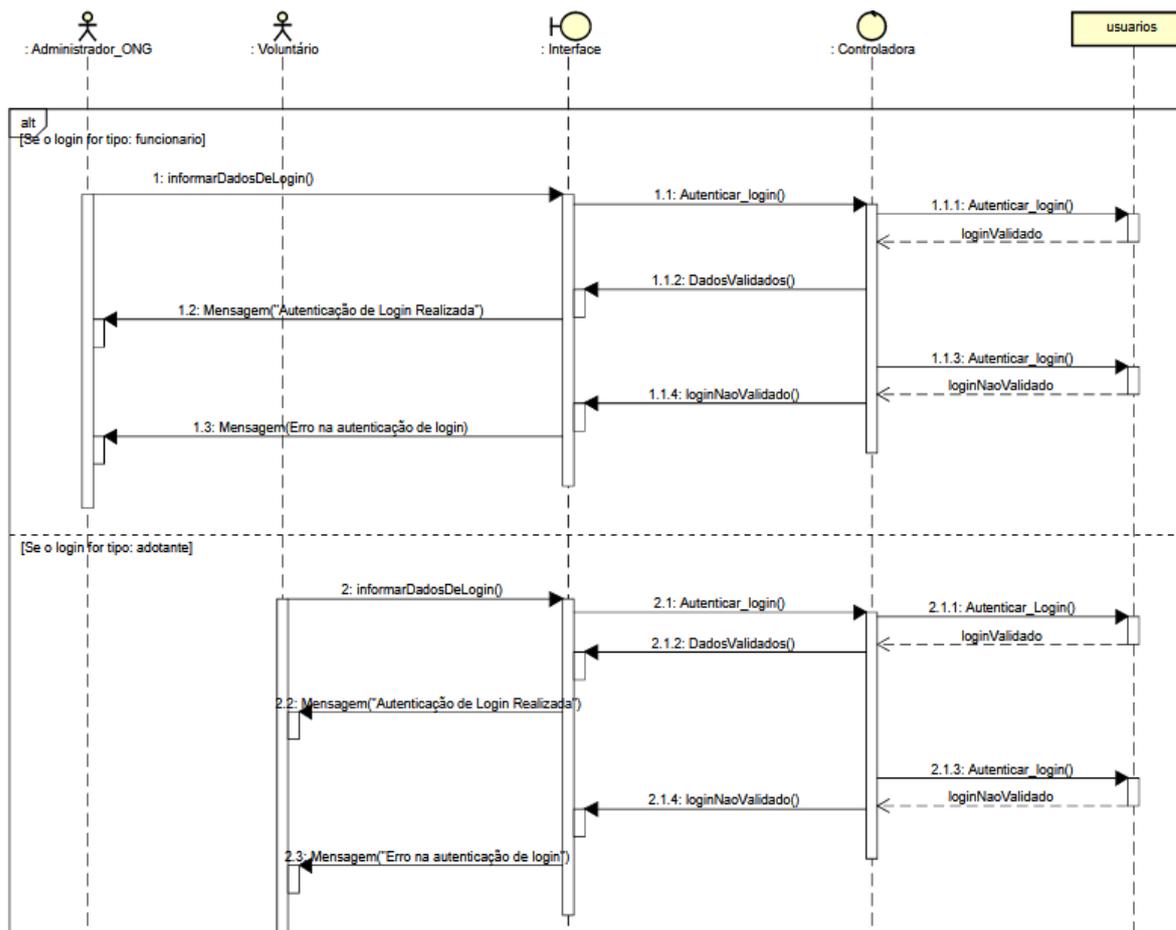
O objetivo desse diagrama é representar as interações entre objetos de forma detalhada entre instâncias de classes e faz parte das ferramentas UML. Em síntese sua função é ilustrar o funcionamento de um sistema.

Figura 9 - Criar relatório



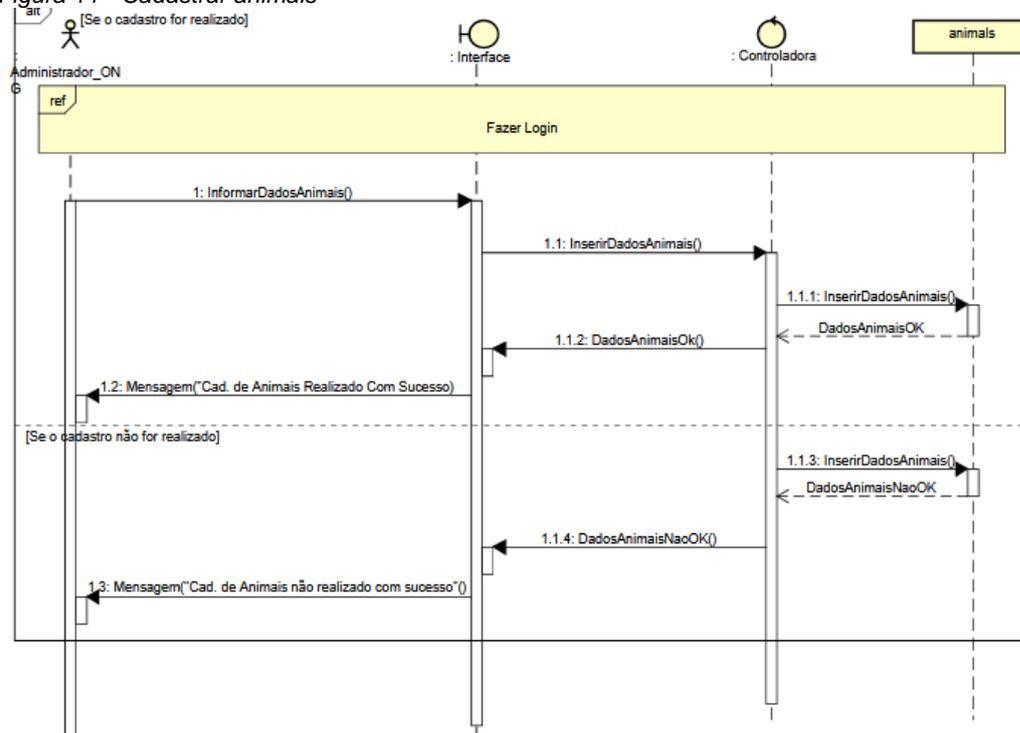
Fonte: Elaborado pelo autor (2018)

Figura 10 - Login



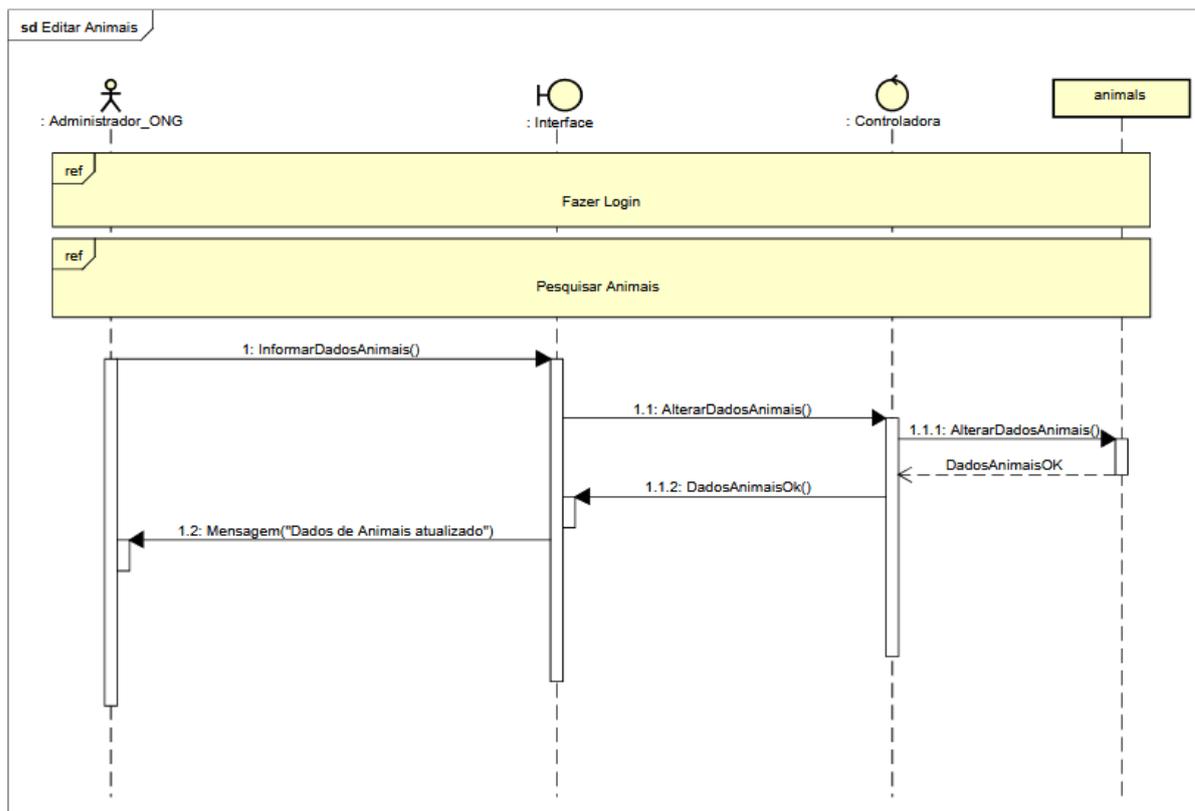
Fonte: Elaborado pelo autor (2018)

Figura 11 - Cadastrar animais



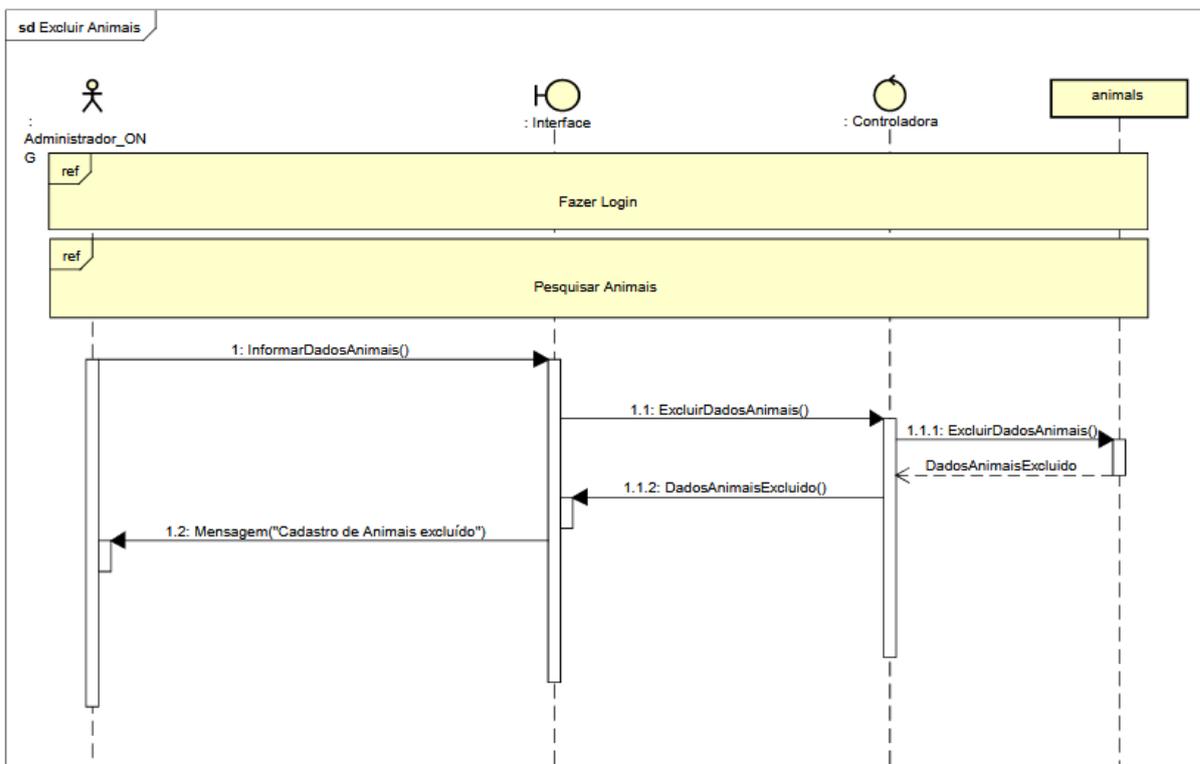
Fonte: Elaborado pelo autor (2018)

Figura 12 - Editar animais



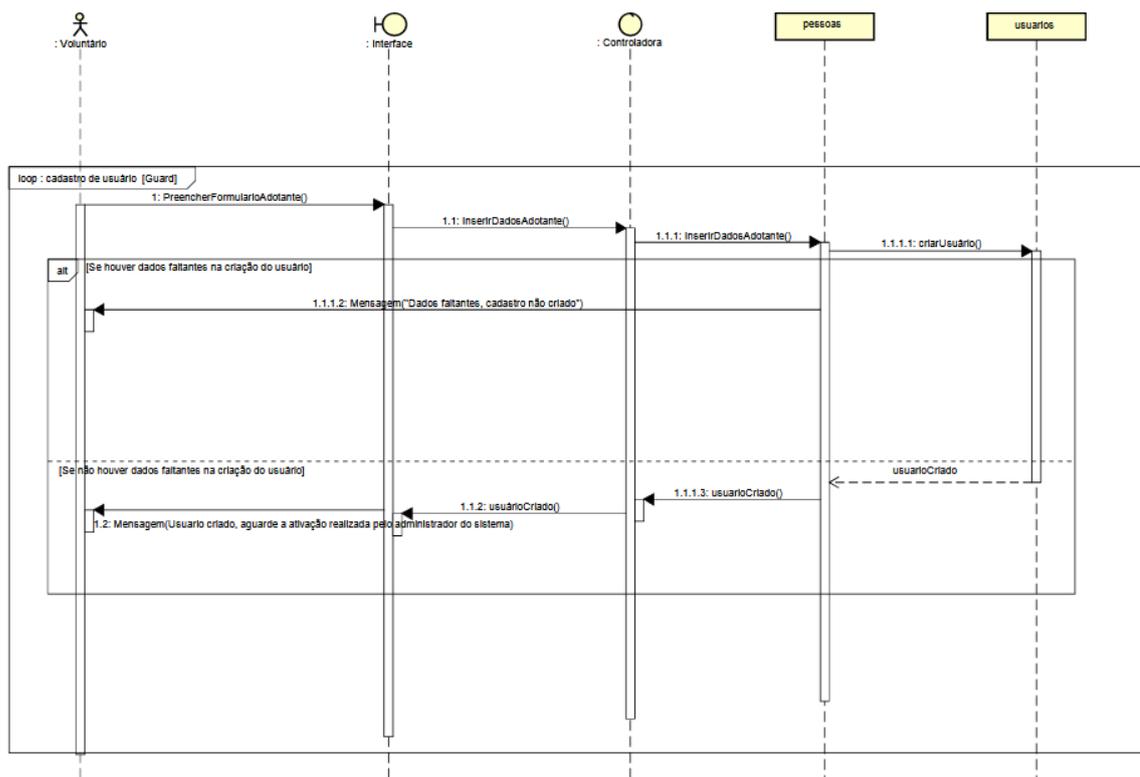
Fonte: Elaborado pelo autor (2018)

Figura 13 - Excluir animais



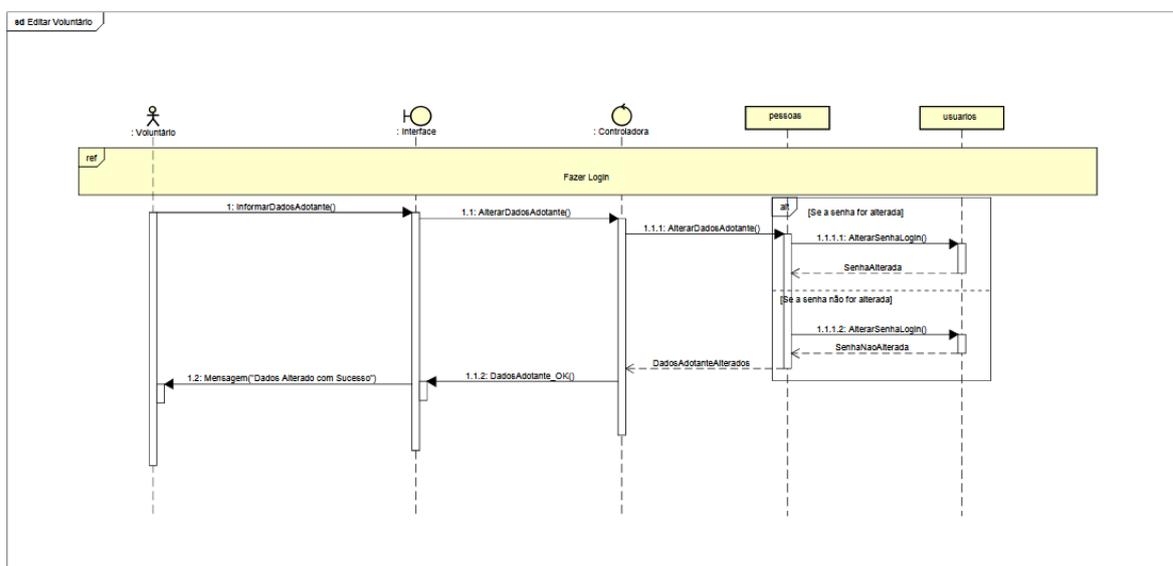
Fonte: Elaborado pelo autor (2018)

Figura 14 - Cadastrar voluntário



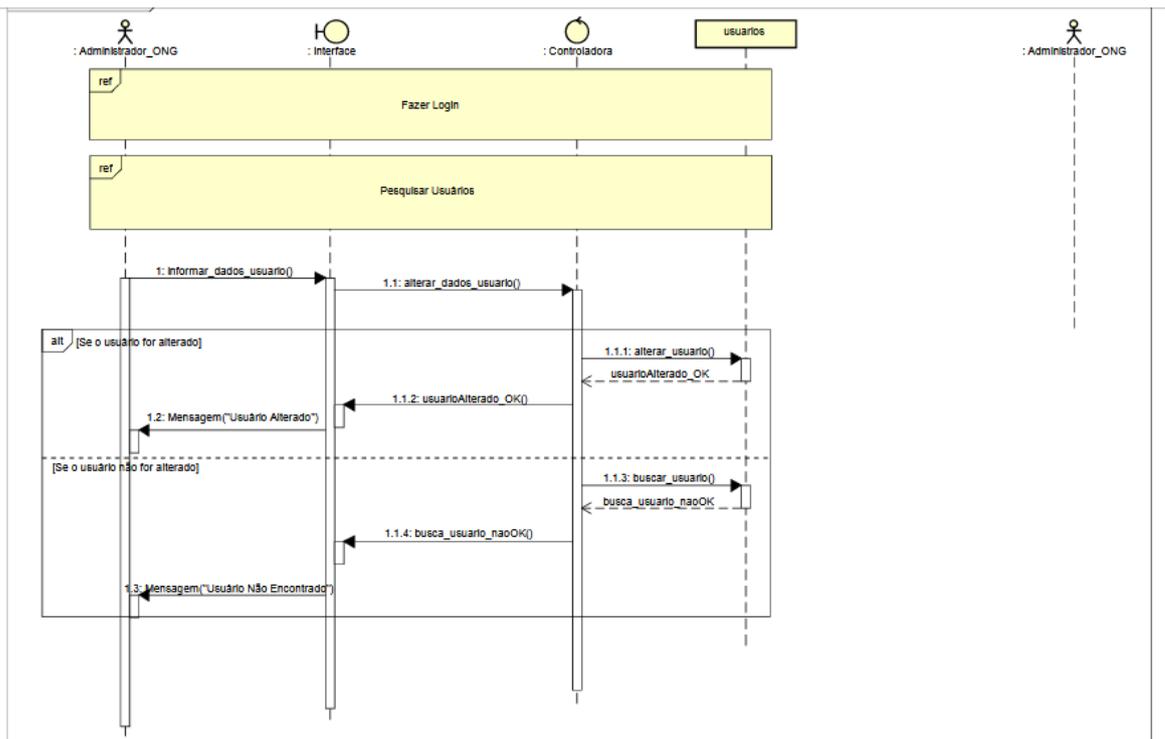
Fonte: Elaborado pelo autor (2018)

Figura 15 - Editar voluntário



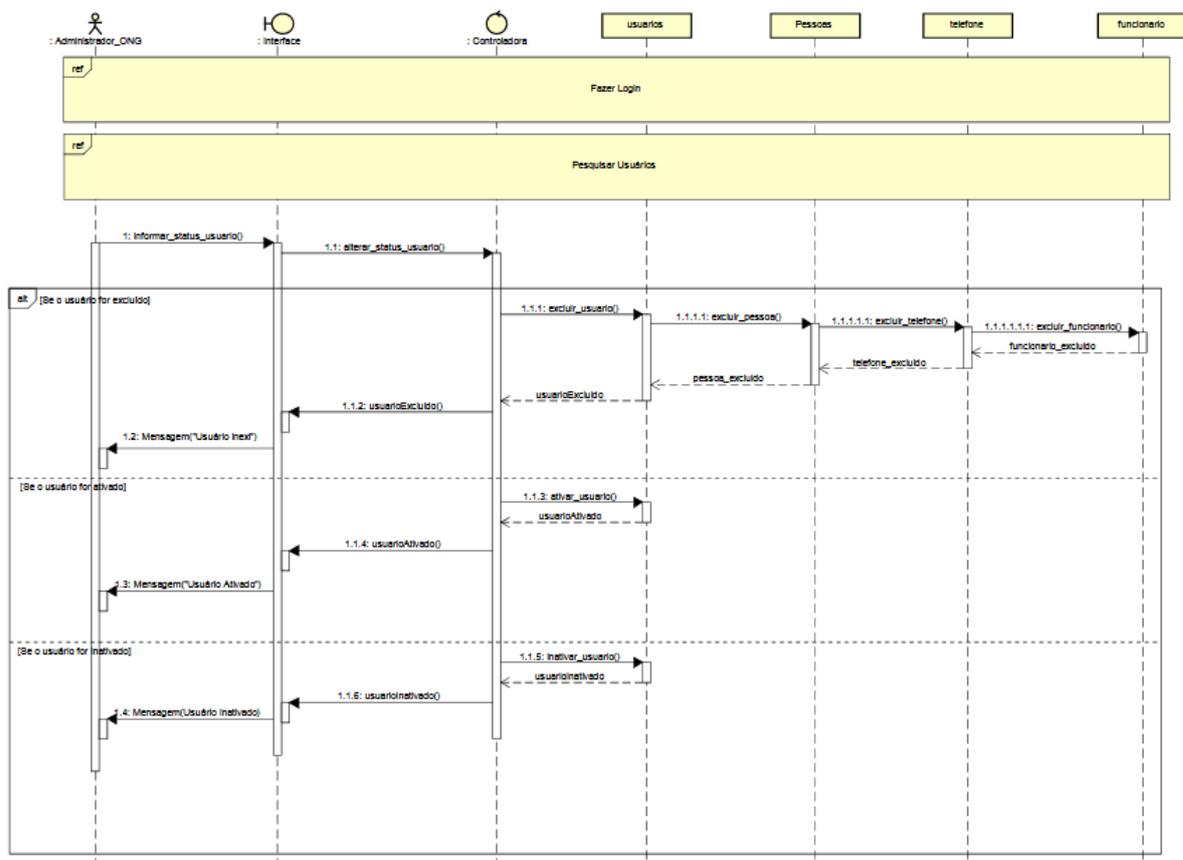
Fonte: Elaborado pelo autor (2018)

Figura 16 - Alterar dados do usuário



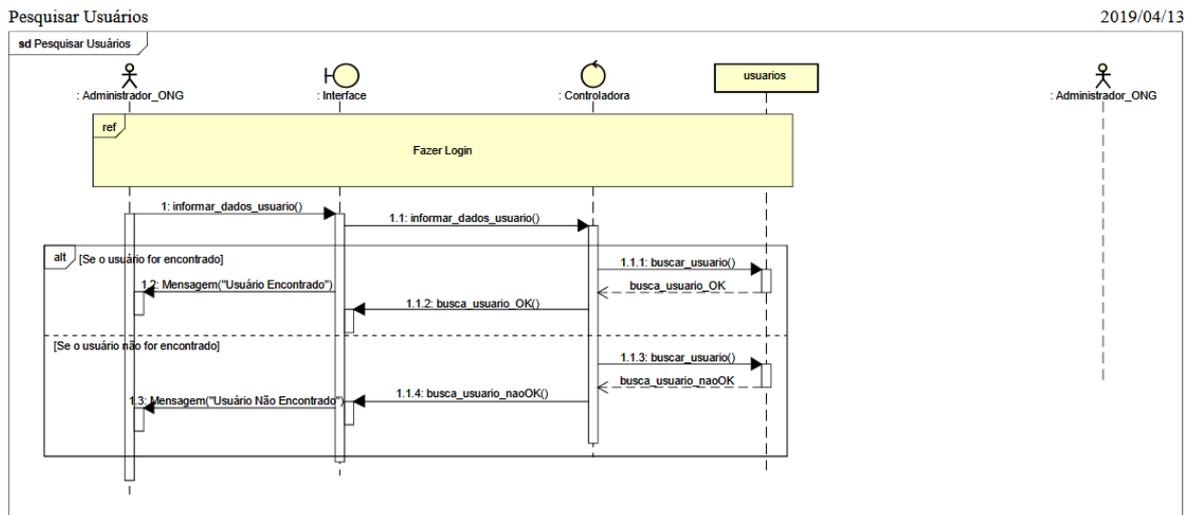
Fonte: Elaborado pelo autor (2018)

Figura 17 - Alterar status do usuário



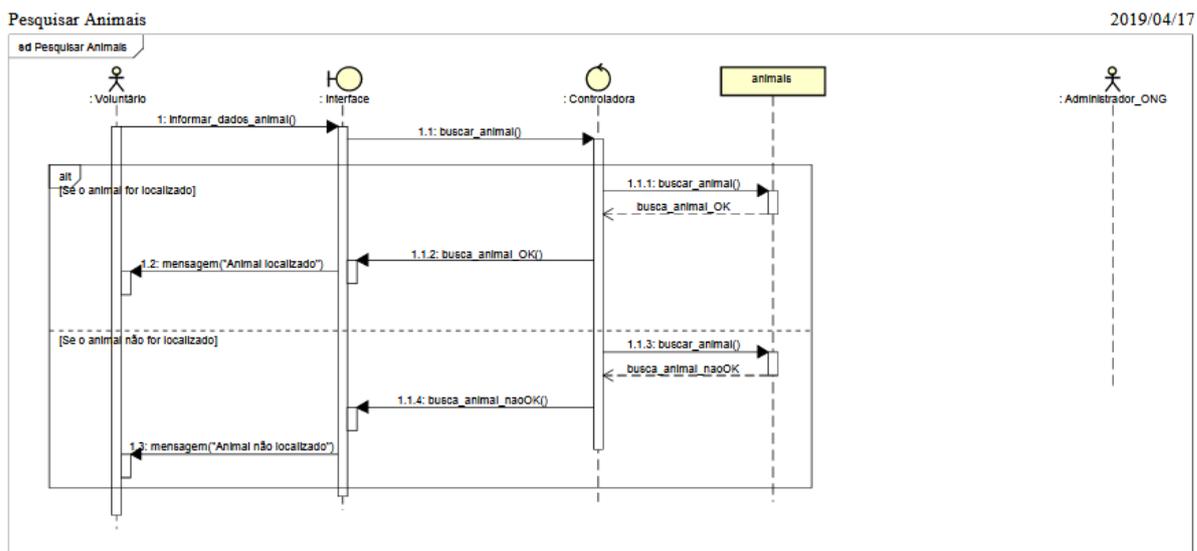
Fonte: Elaborado pelo autor (2018)

Figura 18 - Pesquisar usuários



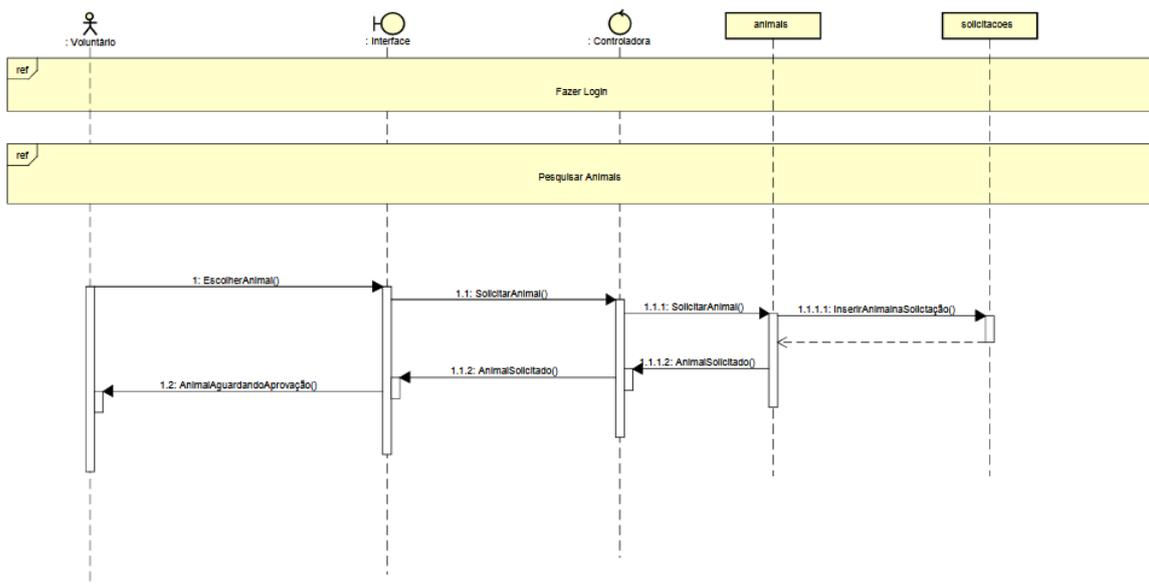
Fonte: Elaborado pelo autor (2018)

Figura 19 - Pesquisar animais



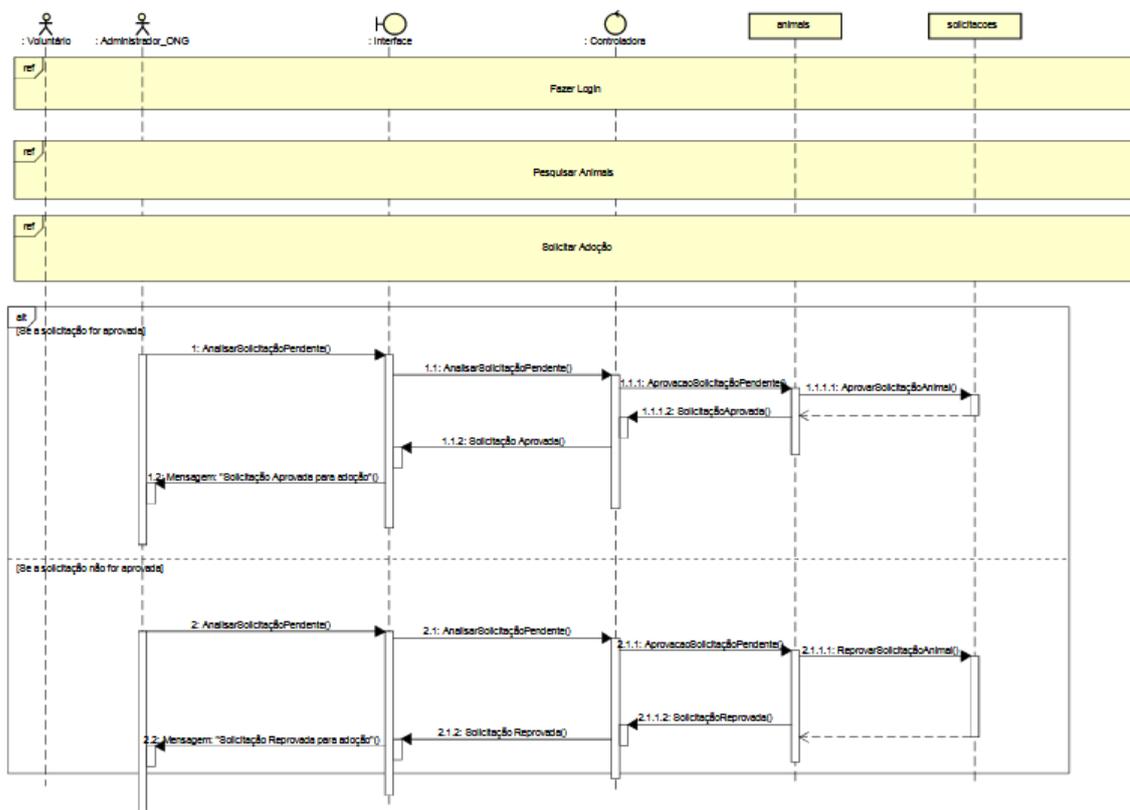
Fonte: Elaborado pelo autor (2018)

Figura 20 - Solicitar adoção



Fonte: Elaborado pelo autor (2018)

Figura 21 - Processos de aprovação



Fonte: Elaborado pelo autor (2018)

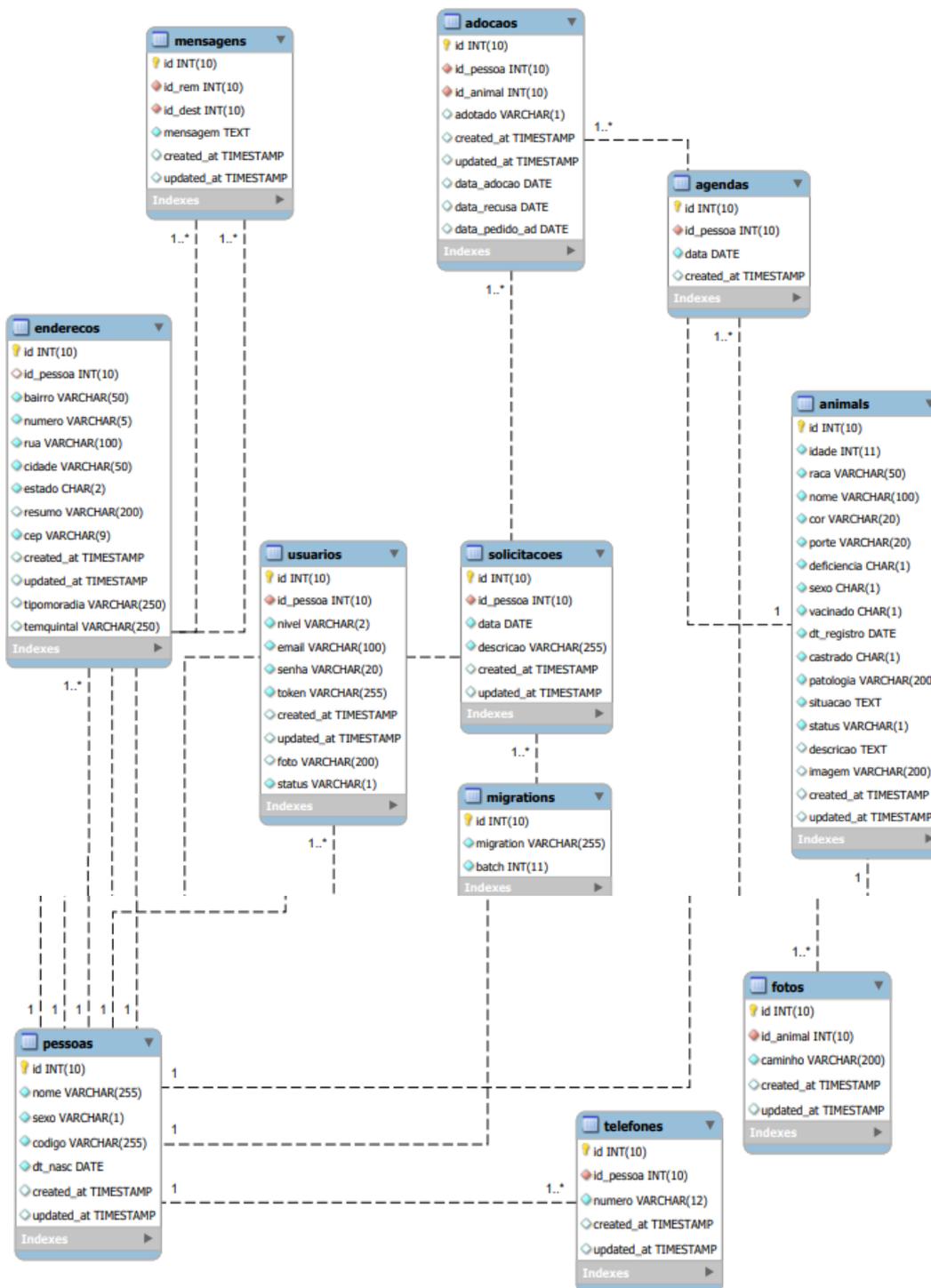
### 3.7. Modelagem de dados

Existem dois tipos de níveis de modelagem de dados: o modelo conceitual, que segundo Heuser (2004, p. 16) “*é uma descrição do banco de dados de forma independente de implementação em um SGBD*” e o modelo lógico que, para o autor (2004, p.17) “*é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD*”. É o processo de criação de modelos que definem entidades, relacionamentos, atributos e como esses dados serão armazenados em um banco de dados físico pelo SGBD (Sistema Gerenciador de Banco de Dados).

A principal ferramenta para a criação desses modelos é o DER (Diagrama de Entidade e Relacionamento), no qual as entidades são representadas por retângulos, com seus atributos detalhados no interior dessas formas, e as relações por losangos conectados as entidades por linhas e com indicações de cardinalidade.

Para o sistema proposto nesse projeto desenvolveu-se apenas o modelo lógico, devido ao uso da ferramenta de modelagem de banco de dados MySQL Workbench. Essa ferramenta criou o banco de dados físico para o sistema a partir do DER, apresentado na próxima página.

Figura 22 - DER



Fonte: Elaborado pelo autor (2018)

### 3.8. Arquitetura de software MVC

Engholm Junior (2013, p. 207) define arquitetura de software como “a estrutura ou estruturas do sistema, que incluem elementos do software, as propriedades externamente visíveis desses elementos e as relações entre eles”. Não é novidade esse conceito, portanto existem diversos modelos de arquitetura

ligados a *frameworks*, linguagens e técnicas de desenvolvimento, os quais consideram aspectos específicos na construção de sistemas, como ferramentas de ambiente de desenvolvimento, reaproveitamento de códigos, heranças, classes, entre outros.

Se destaca o padrão MVC de *model, view e controller*, ou modelo, visão e controle, entre esses modelos de arquitetura, definido por Pitt (2012, p. 1) como “*um modelo de arquitetura de software construído ao redor da interconexão de três componentes principais [...] com grande foco em programação orientada a objeto [...]*”. A característica desse modelo é dividir o desenvolvimento em três camadas separadas, que serão ligadas através da camada controle, o que facilita a modificação de uma camada sem interferir nas outras. Por exemplo, é possível substituir a parte gráfica da aplicação sem a necessidade de alterações significativas nas camadas do modelo e controle.

Essa arquitetura foi escolhida no desenvolvimento do sistema considerando os benefícios proporcionados, como organização de código, reaproveitamento de funções, organização estrutural de camadas, entre outros. A seguir são detalhadas as camadas do MVC.

## **MODEL**

A camada de modelo pode ser definida como “um objeto que representa as informações do domínio de negócios da aplicação” (DALL’OGLIO, 2011, p. 477). Pode ser entendido como uma classe, da programação orientada a objetos, que representa algo no mundo físico, como uma pessoa, usuário, produto, ativo, equipamento, peça, etc., e que traz dentro de si todas as regras de manipulação de dados, sendo responsabilidade dessa camada conectar-se com banco de dados diretamente para recuperar ou registrar informações. Essa camada tem todos os códigos que são necessários para fazer alterações requisitadas pela camada *view* na aplicação e na base de dados. Porém a *model* não decide quando realizar essas ações, ficando essa tarefa de acioná-la depois da requisição da *view* a cargo da camada *controller*.

## **VIEW**

Na visão Dall’Olgio (2011, p. 477) explica que “teremos a definição da interface com o usuário, como os campos serão organizados e distribuídos na tela”.

É a camada de apresentação, que trará os resultados para o usuário a partir das opções que ela também disponibilizará. Essa camada não tem contato com a base de dados, ela se limita apenas a enviar requisições para o controle e receber dele resultados, porém não sabe o que mostra e quando mostrar, quem dirá o que ela deverá exibir e quando será o controle. Para este projeto a *view* será constituída de arquivos HTML, com CSS e *JavaScript*, chamados de *front-end*.

## **CONTROLLER**

Na camada de controle Dall’Olgio (2011, p. 478) ainda diz que “teremos a manipulação dos inputs do usuário, sua interpretação e a execução das tarefas correspondentes”. Essa é a camada que fará o “meio de campo” entre a visão e o modelo, ela diz quando deve ser chamada uma visão e quando um modelo deve ser alterado, embora ela não faça alterações.

O controle receberá todas as requisições da visão e iniciará todos os modelos necessários para uma determinada ação, por exemplo, se o usuário quer imprimir na tela uma tabela com dados de funcionários, o controle acionará o modelo funcionários que trará todos os dados dos funcionários e esses dados serão entregues para a visão pelo controle, e a visão exibirá para o usuário aquilo que ele solicitou.

## **4. PROJETO**

### **4.1 Descrição do projeto**

Foi desenvolvido um sistema, com a finalidade de agilizar o processo de adoção, proporcionando consultar *online* os dados dos animais e posteriormente realizar pelo sistema o processo de adoção. O sistema trouxe medidores referentes a entrada e saída de animais. A implantação do sistema ajudará a otimizar a emissão de documentação, a facilitar o processo de adoção, a promover um canal online de adoção com exposição dos animais e permitirá um cadastro de voluntários que desejam atuar junto a ONG.

### **4.2 Dificuldades encontradas**

No projeto houve algumas dificuldades como:

- Lembrar UML (*Unified Modeling Language*) e todas as suas estruturas ao fazer os diagramas.

- Estudar o *framework* Laravel que foi um desafio para a equipe, mas acrescentou um amplo conhecimento sobre desenvolvimento.
- Criar filtros simultâneos para busca dos animais.
- Geração de relatórios passando os dados dos animais.
- Códigos para integrar juntos ao banco de dados, entender a estrutura dos *Modals* de conexão e seleção de dados.

### 4.3 Prototipagem

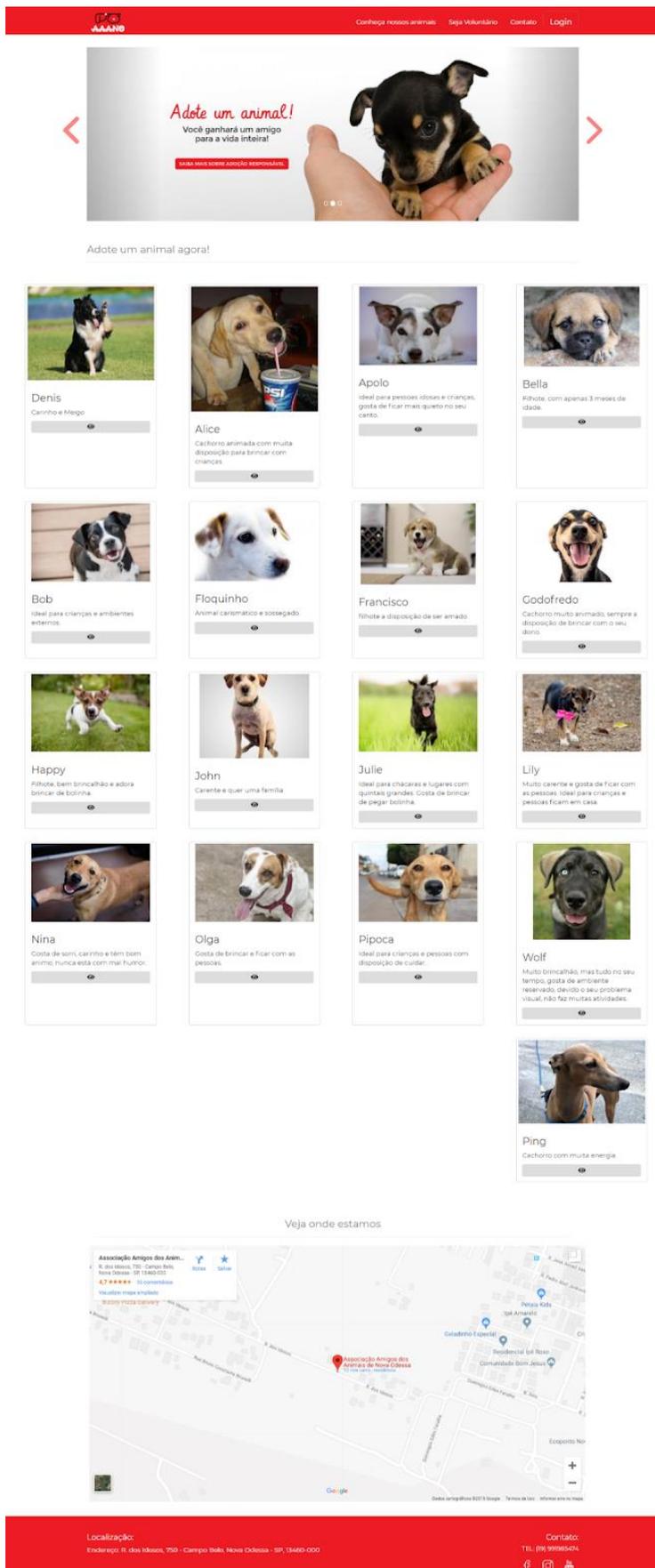
A função da prototipagem é identificar requisitos específicos que são necessários para que os recursos gerais dos aplicativos sejam eficientes, além de validar se a iteração do aplicativo com o usuário é intuitiva e fornecer um *feedback* que possibilite refinar os requisitos. (PRESSMAN, 2005).

#### 4.3.1 Menu navegação – Home

A figura 23 apresenta o protótipo da tela principal. A versão final da tela no sistema composta por:

- **Tela Conheça nossos animais:** Redireciona o usuário para a tela de pesquisa dos animais, onde o usuário poderá realizar buscas através de filtros;
- **Tela Seja Voluntário:** Redireciona o usuário para a tela de cadastro de voluntários;
- **Tela Contato:** Redireciona o usuário para a tela onde haverá informações sobre a localização da ONG, e-mail e telefone de contato;
- **Tela Login:** Redireciona o usuário para a tela de *login* para acessar o sistema com as suas credenciais;
- **Ícone logo da ONG:** Redireciona o usuário sempre para a tela Home.
- **Slider de Fotos:** Apresentação de imagens referente a ONG.
- **Adote um Animal Agora:** Relação de animais pendentes para adoção com a possibilidade de visualizar as suas características.

Figura 23 – Tela principal



Fonte: Elaborado pelo autor (2018)

### 4.3.2 Tela Conheça nossos animais

A figura 24 apresenta o protótipo da tela “Conheça nossos animais”. A versão final da tela no sistema composta por filtros simultâneos como:

- Filtro Raça: Opção de escolha da Raça;
- Filtro Idade: Opção de escolha da Idade;
- Filtro Cor: Opção de escolha da Cor;
- Filtro Porte: Opção de escolha do Porte
- Filtro Deficiência: Opção de escolha da Sim ou Não.
- Filtro Vacinado: Opção de escolha Sim ou Não.

Figura 24 - Tela conheça nossos animais

O protótipo da tela 'Conheça nossos animais' apresenta uma barra de navegação superior em vermelho com o logo 'AAAO' à esquerda e os links 'Conheça nossos animais', 'Seja Voluntário', 'Contato' e 'Login' à direita. Abaixo, há uma barra de filtros com sete campos de seleção: 'Raça', 'Idade', 'Cor', 'Porte', 'Deficiência' e 'Vacinado', cada um com uma seta para baixo. Abaixo dos filtros, há uma barra de busca em vermelho com o ícone de lupa e o texto 'Pesquisar'.

Fonte: Elaborado pelo autor (2018)

### 4.3.3 Tela Seja Voluntário

A figura 26 apresenta o protótipo da tela “Cadastrar Voluntário”. A versão final da tela no sistema é composta por:

- Campo RG/CPF: Para que o usuário digite seu RG/CPF;
- Campo Nome: Para que o usuário digite seu nome;
- Campo data de nascimento: Para que o usuário digite seu dia, mês e ano de nascimento;
- Campo Celular: Para que o usuário digite seu número de telefone preferido para contato;
- Campo Email: Para que o usuário digite seu endereço de e-mail;
- Campo Sexo: Para que o usuário escolha a opção Feminino ou Masculino;
- Campo Rua: Para que o usuário informe o nome da rua;
- Campo Número da casa: Para que o usuário informe número da casa;
- Campo Bairro: Para que o usuário informe o nome do bairro;
- Campo CEP: Para que o usuário informe o nome do CEP;
- Campo Cidade: Para que o usuário informe o nome da cidade;
- Campo Estado: Para que o usuário escolha a opção de estado;

- Campo Tipo de Moradia: Para que o usuário escolha a opção Casa, Apartamento ou Chácara;
- Campo tem quintal: Para que o usuário escolha a opção de Sim ou Não;
- Campo senha: Para que o usuário digite sua senha;
- Campo Confirmar senha: Para que o usuário digite novamente a senha inserida no campo senha;
- Botão Concordo com os termos e condições: Confirmação do termo de voluntariado.

Figura 25 - Termo de responsabilidade

### Termo de responsabilidade

Ciente que concordando com os termos estará aceitando fazer parte da ONG de livre e espontânea vontade. Ciente de que os serviços não serão remunerados e nem possui registro na carteira. Suas informações serão encaminhadas ao administrador para que o mesmo possa analisar os dados. Assim que os dados forem analisados será feito contato.

Fonte: Elaborado pelo autor (2018)

- Botão Solicitar: Executa a ação de realizar o cadastro do usuário como voluntário.

Figura 26 - Tela cadastro

Preencha o formulário abaixo para se cadastrar! Lembre-se que o cadastro está sujeito a aprovação do(a) Administrador(a).

**Nome**  
Qual seu nome? \*

**RG/CPF**  
Digite seu CPF \*

**Sexo**  
Selecione seu sexo ▼

**Data de Nascimento**  
dd/mm/aaaa

**Celular**  
Celular para contato \*

**Email**  
Digite seu email \*

**Senha**  
Digite sua senha \*

**Rua**  
Qual sua rua? \*

**Número da casa**  
Número da casa \*

**Bairro**  
Qual seu bairro? \*

**CEP**  
Qual seu cep? \*

**Cidade**  
Mora em qual Cidade? \*

**Estado**  
-- Qual seu estado? -- ▼

**Confirmar senha**  
Confirme sua senha \*

**Tipo de Moradia**  
-- Qual o tipo da sua moradia? -- ▼

**Tem Quintal**  
-- Em sua moradia tem quintal? -- ▼

Concordo com os termos e condições

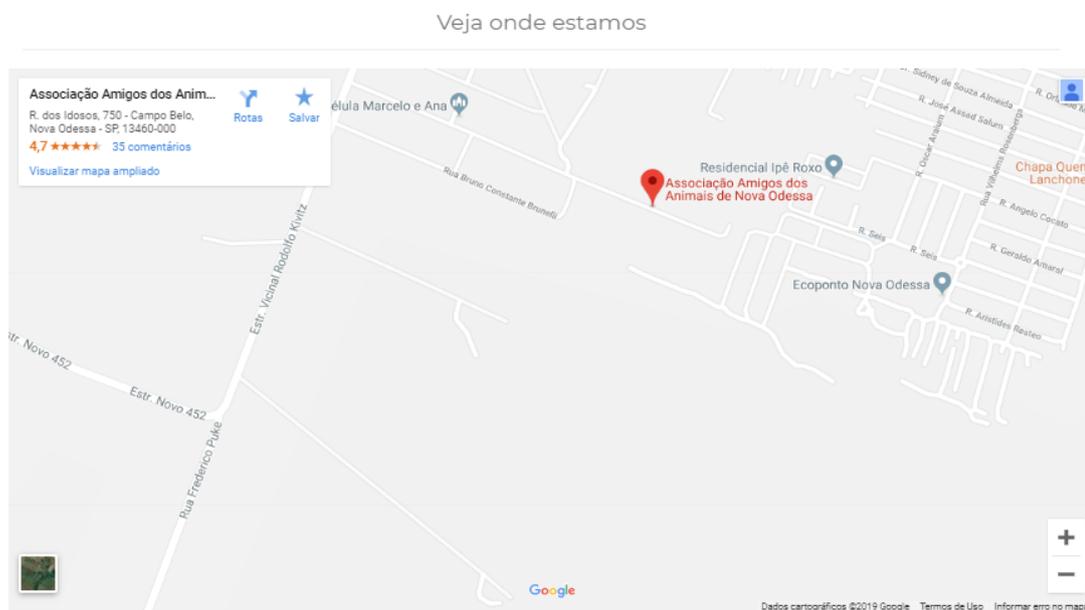
Fonte: Elaborado pelo autor (2018)

### 4.3.4 Tela Contato

A figura 27 apresenta o protótipo da tela “Contato”. A versão final da tela no sistema é composta por:

- Veja onde estamos: Localização exata do *google maps*;
- Informações da ONG: Endereço, telefone de contato e redes sociais.

Figura 27 - Tela contato



Fonte: Elaborado pelo autor (2018)

### 4.3.5 Tela de login

A figura 28 apresenta o protótipo da tela “Login”. A versão final da tela no sistema é composta por:

- Campo e-mail: Para que o usuário digite seu e-mail;
- Campo senha: Para que o usuário digite sua senha;
- Manter salvo: Para que o sistema traga o os dados preenchidos como e-mail e senha, dessa forma, o usuário não precisará digitar suas credenciais;
- Botão login: Realiza a ação de entrar na conta do usuário;

Figura 28 - Tela login

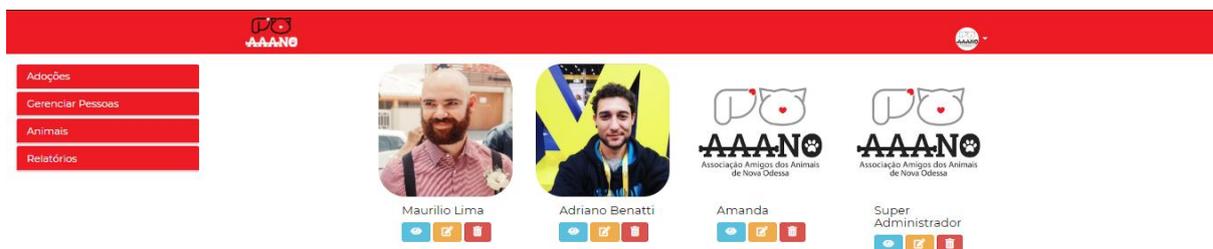
Fonte: Elaborado pelo autor (2018)

#### 4.3.6 Menu navegação – Administradores e Super Administradores

A figura 29 apresenta o protótipo da tela “Menu de navegação – Administradores e Super Administradores”, o qual possui todos os menus de navegações. A versão final da tela no sistema é composta por:

- Menu Adoções: Redireciona o usuário para a tela de solicitações de adoções, onde o usuário poderá realizar aprovações ou rejeições;
- Menu Gerenciar Pessoas: Redireciona o usuário para a tela de cadastro de voluntários, com permissões nas ações de edições, exclusões e visualizações de outros voluntários já existentes no sistema;
- Menu Animais: Redireciona o usuário para a tela de cadastro de animais, com permissões nas ações de edições, exclusões e visualizações dos animais já existentes no sistema.
- Menu Relatórios: Redireciona o usuário para a tela de Relatórios;
- Ícone da ONG: Redireciona o usuário sempre para a tela Home.
- Ícone da foto do usuário: Redireciona o usuário para fazer sair do sistema.

Figura 29 - Menu de navegação



Fonte: Elaborado pelo autor (2018)

### 4.3.7 Tela Adoções – Solicitações de Adoções

A figura 30 apresenta o protótipo da tela “Adoções” onde ficará todos os animais que estão separados para adoção. A versão final da tela no sistema é composta por:

-  : Representa visualizar os dados de quem solicitou a adoções referente ao animal;
-  : Representa aprovar a solicitação de adoção de animal;
-  : Representa reprovar a solicitação de adoção de animal;

Figura 30 - Tela adoções



Fonte: Elaborado pelo autor (2018)

### 4.3.8 Tela Gerenciar Pessoas – Voluntários

A figura 32 apresenta o protótipo da tela “Voluntários” onde ficará todos os usuários voluntários que são existentes no sistema, podendo realizar algumas ações, conforme ícones abaixo. A versão final da tela no sistema é composta por:

-  : Representa visualizar os dados do usuário cadastrado. Ao clicar é exibido a figura abaixo:

Figura 31 - Visualizar cadastro

Maurilio Lima	
Informações	
Nome:	Maurilio Lima
Email:	maurilio.lima@gmail.com
RG/CPF:	45660720889
Rua:	Bulgaria
Nº:	255
Bairro:	Jardim Europa
Cidade:	Americana
Morada:	Casa
Quintal:	Sim
Nível:	Administrador

Fonte: Elaborado pelo autor (2018)

- : Representa editar os dados do usuário. Ao clicar é exibido a figura abaixo:

Figura 32 - Voluntários




Altere os campos a serem editados

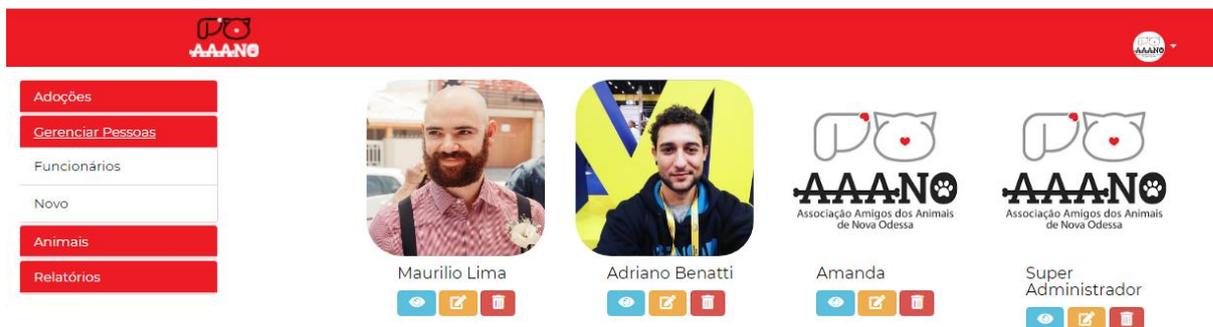
---

<p><b>Nome</b></p> <input type="text" value="Adriano Benatti"/>	<p><b>Rua</b></p> <input type="text" value="Rua de Verzeas"/>
<p><b>RG/CPF</b></p> <input type="text" value="45660720889"/>	<p><b>Número da casa</b></p> <input type="text" value="2470"/>
<p><b>Celular</b></p> <input type="text" value="2470"/>	<p><b>Bairro</b></p> <input type="text" value="Jardim Europa"/>
<p><b>Email</b></p> <input type="text" value="adriano.benatti@gmail.com"/>	<p><b>CEP</b></p> <input type="text" value="13455452"/>
<p><b>Senha</b></p> <input type="password" value=""/>	<p><b>Cidade</b></p> <input type="text" value="SBO"/>
<p><b>Status</b></p> <input type="text" value="Ativo"/>	<p><b>Estado</b></p> <input type="text" value="Acre"/>
<p><b>Tipo de usuário</b></p> <input type="text" value="Administrador"/>	<p><b>Confirmar senha</b></p> <input type="text" value="Confirme sua senha *"/>
<p><b>Escolha uma Imagem de perfil</b></p>  <p><a href="http://localhost:8000/storage/imagens/adriano-benatti.jpeg">http://localhost:8000/storage/imagens/adriano-benatti.jpeg</a></p> <input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado	<p><b>Tipo de Moradia</b></p> <input type="text" value="Casa"/>
	<p><b>Tem Quintal</b></p> <input type="text" value="Sim"/>

Fonte: Elaborado pelo autor (2018)

-  : Representa excluir o usuário cadastro. Ao clicar, exclui o usuário imediatamente.

Figura 33 - Excluir cadastro



Fonte: Elaborado pelo autor (2018)

#### 4.3.9 Tela Gerenciar Pessoas – Novo

A figura 34 apresenta o protótipo da tela “gerenciar pessoas” na qual será cadastrado um novo usuário. Essa tela é idêntica a “Tela Seja Voluntário”, a única diferença é que possui a opção de escolher uma imagem para o perfil do usuário. A versão final da tela no sistema é composta por:

Figura 34 - Tela gerenciar pessoas

<p><b>Nome</b></p> <input type="text" value="Qual seu nome? *"/>	<p><b>Rua</b></p> <input type="text" value="Qual sua rua? *"/>
<p><b>RG/CPF</b></p> <input type="text" value="Digite seu CPF *"/>	<p><b>Número da casa</b></p> <input type="text" value="Número da casa *"/>
<p><b>Sexo</b></p> <input type="text" value="Selecione seu sexo"/>	<p><b>Bairro</b></p> <input type="text" value="Qual seu bairro? *"/>
<p><b>Data de Nascimento</b></p> <input type="text" value="dd/mm/aaaa"/>	<p><b>CEP</b></p> <input type="text" value="Qual seu cep? *"/>
<p><b>Celular</b></p> <input type="text" value="Celular para contato *"/>	<p><b>Cidade</b></p> <input type="text" value="Mora em qual Cidade? *"/>
<p><b>Email</b></p> <input type="text" value="superadministrador@gmail.com"/>	<p><b>Estado</b></p> <input type="text" value="Acre"/>
<p><b>Senha</b></p> <input type="text" value="--"/>	<p><b>Confirmar senha</b></p> <input type="text" value="Confirme sua senha *"/>
<p><b>Tipo de usuário</b></p> <input type="text" value="Super Administrador"/>	<p><b>Tipo de Moradia</b></p> <input type="text" value="-- Qual o tipo da sua moradia ? --"/>
	<p><b>Tem Quintal</b></p> <input type="text" value="-- Em sua moradia tem quintal ? --"/>
	<p><b>Escolha uma imagem de perfil</b></p> <input type="text" value="Escolher arquivo"/> Nenhum arquivo selecionado

Fonte: Elaborado pelo autor (2018)

### 4.3.10 Tela Animais – Animais

A figura 35 apresenta o protótipo da tela “Animais” na qual, será exibido todos animais cadastrados no sistema (pendente para adoção). A versão final da tela no sistema é composta por:

- : Representa visualizar os dados do animal cadastrado. Ao clicar é exibido a figura abaixo:

Figura 35 - Tela animais

Denis	
Informações	
Nome:	Denis
Raça:	Vira-lata
Cor:	bege
Idade:	2
Sexo:	Masculino
Porte:	pequeno
Situação:	Ele é fofinho
Status:	Ativo
Vacinado:	Sim
Deficiência:	Não
Patologia:	Sim
Data de registro:	27/04/2019
Descrição:	Carinho e Meigo

Fonte: Elaborado pelo autor (2018)

- : Representa editar os dados do animal. Ao clicar é exibido a figura abaixo:

Figura 36 - Tela editar

AAANO
11:44

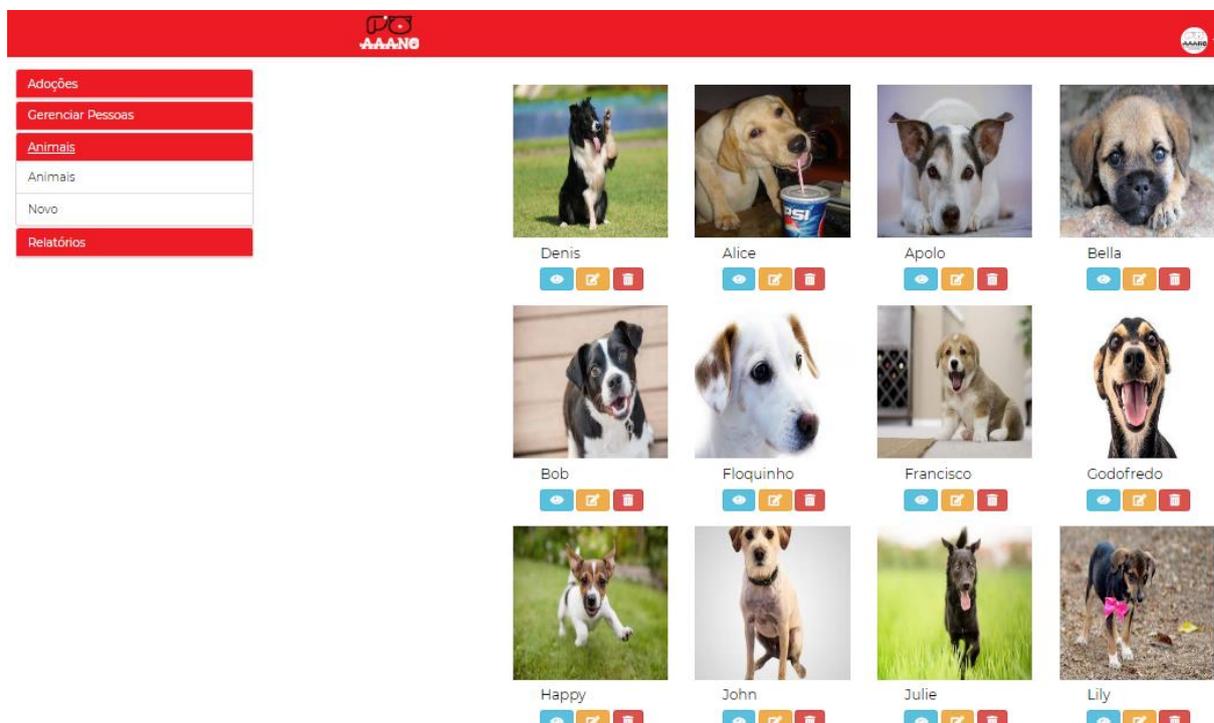
Atualize os dados um animal

<p><b>Nome</b> <input type="text" value="Denis"/></p> <p><b>Raça</b> <input type="text" value="Vira-lata"/></p> <p><b>Idade</b> <input type="text" value="2"/></p> <p><b>Cor</b> <input type="text" value="bege"/></p> <p><b>Porte</b> <input type="text" value="pequeno"/></p> <p><b>Sexo</b> <input type="text" value="Macho"/></p> <p><b>Situação</b> <input type="text" value="Ele é fofinho"/></p>	<p><b>Vacinado</b> <input type="text" value="Sim"/></p> <p><b>Deficiência</b> <input type="text" value="Não"/></p> <p><b>Data de registro</b> <input type="text" value="27/04/2019"/></p> <p><b>Castrado</b> <input type="text" value="Sim"/></p> <p><b>Patologia</b> <input type="text" value="Sim"/></p> <p><b>Escolha uma imagem</b>  <input type="text" value="http://localhost:8000/storage/imagens/denis.jpeg"/> <input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado</p> <p><b>Status</b> <input type="text" value="Ativo"/></p> <p><b>Descricao</b> <input type="text" value="Carinho e Meigo"/></p>
---	---

Fonte: Elaborado pelo autor (2018)

-  : Representa excluir o animal do cadastro. Ao clicar, exclui o animal imediatamente.

Figura 37 - Tela animais



Fonte: Elaborado pelo autor (2018)

#### 4.3.11 Tela Animais – Novo

A figura 38 apresenta o protótipo da tela “novo animal”, na qual será cadastrado um novo animal. A versão final da tela no sistema é composta por:

- Campo Nome: Para que o usuário digite o nome do animal;
- Campo Raça: Para que o usuário digite a raça do animal;
- Campo Idade: Para que o usuário digite a idade do animal;
- Campo Cor: Para que o usuário digite a cor do animal;
- Campo Porte: Para que o usuário digite o tamanho do animal;
- Campo Sexo: Para que o usuário escolha o sexo do animal (Macho ou Fêmea);
- Campo Vacinado: Para que o usuário escolha se o animal está vacinado (Sim ou Não);
- Campo Deficiência: Para que o usuário escolha se o animal possui algum tipo de deficiência (Sim ou Não);

- Campo Data de Registro: Para que o usuário escolha a data de entrada do animal na ONG;
- Campo Castrado: Para que o usuário escolha se o animal é castrado (Sim ou Não);
- Campo Patologia: Para que o usuário digite a patologia do animal;
- Campo Escolha uma imagem: Para que o usuário faça o carregamento de alguma imagem do animal;
- Campo Situação: Para que o usuário informe as situações atual do animal.
- Campo Descrição: Para que o usuário informe algumas características do perfil do animal.
- Botão Concordo com os termos e condições: Confirmação do termo de inserção do animal para habilitação da adoção.
- Botão Cadastrar: Executa a ação de realizar o cadastro do animal e incluir no sistema.

Figura 38 – Tela cadastro novo animal

The screenshot shows a web form for registering a new animal. The form is titled "Cadastre um animal" and is organized into two columns. The left column contains fields for "Nome", "Raça", "Idade", "Cor", "Porte", "Sexo", and "Situação". The right column contains dropdown menus for "Vacinado", "Deficiência", "Castrado", and "Patologia", a date field for "Data de registro", and an image upload section titled "Escolha uma Imagem". At the bottom of the form, there is a checkbox labeled "Concordo com os termos e condições" and a green button labeled "Cadastrar".

Fonte: Elaborado pelo autor (2018)

### 4.3.12 Tela Relatórios – Adoções

A figura 39 apresenta o protótipo da tela “relatório geral” onde será exibido o relatório com dados das solicitações de adoções. A versão final da tela no sistema é composta por:

- Campo Data de Início: Para que o usuário digite a data inicial;
- Campo Data de Fim: Para que o usuário digite a data final;
- Botão Pesquisar: Botão para executar a pesquisa e gerar o relatório no formato .pdf;

Figura 39 – Tela relatório geral

Fonte: Elaborado pelo autor (2018)

### Relatório gerado:

Figura 40 - Relatório gerado

Associação Amigos dos Animais de Nova Odessa				
Relatório			De 01/01/2019 até 18/05/2019	
<b>Adoções realizadas</b>				
Nome do adotante	Nome do animal	Raça	Data de adoção	Status animal
Adriano Benatti	Ben	biggle	06/02/2019	I
Adriano Benatti	horse	Cavalo	10/02/2019	I
Adriano Benatti	Thor	biggle	10/02/2019	I
Adriano Benatti	Ted	biggle	10/02/2019	I
Adriano Benatti	xxxxxxx	xxxxxx	10/02/2019	I
Adriano Benatti	Charlie	pastor Alemão	10/02/2019	I
Adriano Benatti	Cadela	Pincher	10/02/2019	I
Adriano Benatti	Amora	biggle	24/03/2019	I
Adriano Benatti	Bilie	dog	10/02/2019	I
Super Administrador	Bob	Vira-lata	17/04/2019	I
<b>Pedidos de Adoção</b>				
Nome do adotante	Nome do animal	Raça	Data do pedido de adoção	Status animal
<b>Pedidos de Adoção recusados</b>				
Nome do adotante	Nome do animal	Raça	Data de recusa	Status animal
Adriano Benatti	Amora	biggle	06/02/2019	I
Adriano Benatti	Bilie	dog	06/02/2019	I

Fonte: Elaborado pelo autor (2018)

### 4.3.13 Tela Relatórios – Animais para Adoção

A figura 41 apresenta o protótipo da tela “Relatório – Animais para Adoção” onde será exibido o relatório com dados dos animais pendentes para a adoção. A versão final da tela no sistema é composta por:

- Campo Data de Início: Para que o usuário digite a data inicial;
- Campo Data de Fim: Para que o usuário digite a data final;
- Botão Pesquisar: Botão para executar a pesquisa e gerar o relatório no formato .pdf;

Figura 41 - Tela relatório animais

O protótipo da tela 'Relatório Animais' apresenta um cabeçalho vermelho com o logo 'AAANO' e um menu lateral com as opções: Adoções, Gerenciar Pessoas, Animais e Relatórios. O título principal é 'Relatório Animais'. Abaixo dele, há dois campos de entrada para 'Data de início' (01/01/2019) e 'Data de fim' (18/05/2019), seguidos por um botão verde 'Pesquisar'.

Fonte: Elaborado pelo autor (2018)

Relatório gerado:

Figura 42 - Exemplo do relatório

O exemplo do relatório gerado apresenta um cabeçalho vermelho com o logo 'AAANO' e o nome 'Associação Amigos dos Animais de Nova Odessa'. Abaixo, há um subtítulo 'Relatório' e o período 'De 01/01/2019 até 18/05/2019'. A tabela principal, intitulada 'Animais esperando adoção', contém as seguintes informações:

Nome do animal	Raça	Cor	Idade	Descrição	Data de Registro
Denis	Vira-lata	bege	2	Carinho e Meigo	2019-04-27
Alice	Vira-lata	bege	2	Cachorro animada com muita disposição para brincar com crianças.	2019-04-25
Apolo	Vira-lata	mesclado	2	ideal para pessoas idosas e crianças, gosta de ficar mais quieto no seu canto.	2019-12-10
Bella	Vira-lata	bege	1	Filhote, com apenas 3 meses de idade.	2019-01-22
Bob	Vira-lata	mesclado	3	Ideal para crianças e ambientes externos.	2019-01-18
Floquinho	Vira-lata	branco	2	Animal carismático e sossegado.	2019-03-12
Francisco	Vira-lata	bege	1	filhote a disposição de ser amado	2019-03-17
Godofredo	Vira-lata	preto	1	Cachorro muito animado, sempre a disposição de brincar com o seu dono.	2018-10-10
Happy	Vira-lata	mesclado	1	Filhote, bem brincalhão e adora brincar de bolinha.	2019-04-25
John	Vira-lata	bege	1	Carente e quer uma família	2019-04-18
Julie	Vira-lata	preto	4	Ideal para chácaras e lugares com quintais grandes. Gosta de brincar de pegar bolinha.	2019-01-01

Fonte: Elaborado pelo autor (2018)

#### 4.4 – Casos de testes

Tabela 15 - Casos de teste

Caso de Teste	Nome do Teste	Descrição	Passos	Resultado Esperado
CT1	Criar cadastro de Voluntário pela home	Criar um cadastro para acesso ao sistema ao clicar no botão: "Seja Voluntário"	1- Clicar no botão "Seja Voluntário"	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma mensagem de conclusão.
			2- Preencher os dados solicitados	
			3- Clicar no botão "Concordo com os termos e condições"	
			4- Clicar no botão "Solicitar"	
CT2	Fazer Login (todos os perfis)	Acesso ao sistema.	1- Clicar no botão "Login"	Acesso a área do usuário de acordo com perfil.
			2- Digitar o e-mail e a senha.	
			3- Clicar no botão "Login".	
CT3	Fazer Logoff	Sair do sistema	1- Clicar no ícone da imagem do usuário.	Sair da área do usuário.
			2- Clicar no botão "Efetuar Logoff"	
CT4	Cadastro de Animal	Criar um cadastro de animal para adoção, dentro da área do administrador	1- Clicar no botão "Animais".	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma mensagem de conclusão.
			2- Clicar no botão "Novo".	
			3- Preencher os dados solicitados	
			4- Fazer upload da foto do animal.	
			5- Concordar com os termos e condições de inserção do animal para adoção.	

			6- Clicar no botão "Cadastrar" para finalizar o processo.	
<b>CT5</b>	Cadastro de Voluntário pelo menu administrador	Criar um cadastro de voluntários dentro da área do administrador.	1- Clicar no botão "Gerenciar Pessoas".	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma mensagem de conclusão.
			2- Clicar no botão "Novo".	
			3- Preencher os dados solicitados	
			4- Fazer upload da foto do perfil.	
			5- Em tipo de usuário, escolher a opção de perfil Voluntário.	
			6- Clicar no botão "Cadastrar" para finalizar o processo.	
<b>CT6</b>	Cadastro de Administrador	Criar um cadastro de administrador dentro da área do super administrador.	1- Clicar no botão "Gerenciar Pessoas".	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma mensagem de conclusão.
			2- Clicar no botão "Novo".	
			3- Preencher os dados solicitados	
			4- Fazer upload da foto do perfil.	
			5- Em tipo de usuário, escolher a opção de perfil Administrador.	
			6- Clicar no botão "Cadastrar" para finalizar o processo.	
<b>CT7</b>	Cadastro de super administrador	Criar um cadastro de super administrador dentro da área do super administrador.	1- Clicar no botão "Gerenciar Pessoas".	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma mensagem de conclusão.
			2- Clicar no botão "Novo".	
			3- Preencher os dados solicitados	
			4- Fazer upload da foto do perfil.	
			5- Em tipo de usuário, escolher a opção de perfil Administrador.	
			6- Clicar no botão "Cadastrar" para finalizar o processo.	
<b>CT8</b>	Visualizar animal, editar e excluir	Visualizar animais e realizar ações de editar e excluir.	1- Clicar no botão "Animais".	Conseguir visualizar os dados dos animais e fazer as ações de editar e excluir animal.
			2- Clicar no botão "Animais".	
			3- Visualizar animais no painel	
			4- Realizar ação de acordo com os botões de visualizar, editar e excluir.	

<b>CT9</b>	Visualizar Voluntários, editar e excluir	Visualizar animais e realizar ações de editar e excluir.	1- Clicar no botão "Gerenciar Pessoas".	Conseguir visualizar os dados dos voluntários e fazer as ações de editar e excluir voluntários.
			2- Clicar no botão "Voluntários".	
			3- Visualizar voluntários no painel.	
			4- Realizar ação de acordo com os botões de visualizar, editar e excluir.	
<b>CT10</b>	Criar Relatório de Adoção	Criar relatório com o status das solicitações de adoções.	1- Clicar no botão "Relatórios".	Exibir o relatório com as situações de adoções.
			2- Clicar no botão "Adoções"	
			3- Digitar ou escolher a Data de início e Data de fim	
			4- Clicar no botão "Pesquisar"	
<b>CT11</b>	Criar Relatório de Animais para adoção	Criar relatório com dados dos animais para adoção	1- Clicar no botão "Relatórios".	Exibir o relatório com os animais para adoções.
			2- Clicar no botão "Adoções"	
			3- Digitar ou escolher a Data de início e Data de fim	
			4- Clicar no botão "Pesquisar"	
<b>CT12</b>	Filtrar Animais na home	Realizar filtros simultâneos para encontrar animal desejado.	1- Clicar no botão "Conheça nossos animais"	Retornar animal de acordo com as características buscadas.
			2- Escolher os filtros necessários.	
			3- Clicar no botão "Pesquisar".	

Fonte: Elaborado pelo autor (2018)

Referente aos casos de testes citados acima, todos foram efetivamente concluídos com sucesso, nos navegadores: Internet Explorer, Mozilla Firefox e Google Chrome. Foram testados a sua responsividade e não houve alterações no front end.

## 5. CONCLUSÃO

Este trabalho teve como objetivo final o desenvolvimento de um sistema *web* que possibilitasse para uma ONG de adoção de animais, que qualquer pessoa na *web* pudesse realizar uma adoção de um animal e se inscrever através de voluntariado.

O sistema foi concluído e permite transmitir de maneira digital pesquisas sobre os animais da ONG, interação entre voluntários e funcionalidades de cadastros de novos usuários de diferentes tipos de perfis, edições e exclusões. Solicitações de animais baseadas na análise dos dados dos adotantes podendo realizar ações de aprovações e rejeições. Medidores também foram criados para melhorar a visualização de status dos animais, possibilitando ver quantos animais estão para serem adotados, pendentes e reprovados e o quantitativo de animais que ainda não foram solicitadas adoções.

A metodologia *Kanban* foi utilizada para organizar um cronograma, prever as tarefas que estavam em possíveis atrasos, possibilitando ações preventivas para que o prazo fosse cumprido. As reuniões sobre o projeto foram realizadas, por acesso remoto de forma semanal para analisar as dificuldades de cada integrante da equipe, essas reuniões foram de extrema importância para identificar as possíveis alterações nos requisitos e possibilitar o desenvolvimento de um sistema eficiente. Como o projeto foi desenvolvido para um cliente real, foi muito importante a experiência adquirida, pois foi necessário aprender com o cliente os processos e ajudar com melhorias, mesmo que ainda existam algumas melhorias que serão realizadas na etapa de implantação.

A implantação do *software* no cliente não foi possível devido o tempo limitado do desenvolvimento, mas o projeto será implementado, ajustes serão realizados se necessários, considerando a causa defendida.

Como equipe algumas dificuldades técnicas foram superadas através de estudos e pesquisas para relembrar tecnologias e metodologias aprendidas nos semestres anteriores. O estudo pessoal e pesquisa em fóruns e autores técnicos possibilitaram melhorar a resolução dos problemas que surgiram no desenvolvimento do sistema e isso contribuiu no amadurecimento da equipe durante o projeto, bem como no conhecimento adquirido na elaboração do projeto.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABINPET. **Faturamento do setor crescerá 7,4% e fechará em R\$ 17,9 Bilhões em 2015**. Associação Brasileira da Indústria de Produtos para Animais de Estimação, jun. 2015. Disponível em: <<http://abinpet.org.br/site/faturamento-do-setor-crescera-74-e-fechara-em-r-179-bilhoes-em-2015/>> Acesso em: 20 mar. 2018.

AGILE MANIFESTO, Disponível em <http://agilemanifesto.org/>. 2001.

ANDERSON, David J. **.Kanban: Mudança Evolucionária de Sucesso Para Seu Negócio de Tecnologia**. Front Cover. Blue Hole Press, 2011

CARUSO, A. **O desafio de conter a superpopulação e o abandono dos animais**. Clínica Veterinária, ano XIV, n. 81, p. 12, jul/ago. 2009. Disponível em: <<http://issuu.com/clinicavet/docs/clinica-veterinaria-n81/>> Acesso em: 20 mar. 2018.

DALL’OGLIO, P. (2011). **PHP programando com orientação a objetos**. 2ª edição. Editora: Novatec, Brasil.

DAVIS, Michele E.; PHILLIPS, Jon A. **Aprendendo PHP e MySQL**. Trad. Rita Sussekind. 2 ed. Rio de Janeiro: Alta Books, 2008. p. 2-5.

DIAS, J., **Raiva canina**. Tudo Sobre Cachorros, mar. 2015. Disponível em: <<http://tudosobrecachorros.com.br/2012/02/raiva-canina.html/>> Acesso em: 20 mar. 2018.

ENGHOLM, H. J. **Análise e Design - Orientados a Objetos**. 2013

FERNANDES, F. C. F; GODINHO FILHO, M. **Sistemas de coordenação de ordens: revisão, classificação, funcionamento e aplicabilidade**. Revista Gestão & Produção, São Carlos, v.14, n.2, 2007

GOMES, Ana Laura. **Xhtml/Css – Criação de páginas web**. Editora Senac, 2010.

GOMES, C. C. M., **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. Monografia (Curso de Medicina Veterinária) – Faculdade de Agronomia e Medicina Veterinária da Universidade de Brasília, Brasília.

GUEDES, Gilleanes T. A. **UML: uma abordagem prática**. São Paulo: Novatec Editora, 2009. p. 19, 22, 106, 285-230.

- HEUSER, C.A. **Projeto de Banco de Dados**. 6a Edição. 2004
- LAZARIN, L. R., **A Comunicação como Estratégia de Gestão em ONGs: informação, sensibilização e engajamento para sustentar e ampliar a garantia dos direitos dos animais**. 2014. Monografia (Curso de Publicidade e Propaganda) – Universidade Federal do Rio Grande do Sul, Porto Alegre.
- LIMA, Adilson da Silva. **UML 2.0: Do Requisito à Solução**. 4.ed. São Paulo: Érica Ltda., 2010. p. 57.
- NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2 ed. São Paulo: Novatec, 2007.
- PITT, Chris. **Pro PHP MVC**. 1.ed. New York: Apress, 2012. p. 1.
- PRESSMAN, Roger. **Engenharia de Software: Uma abordagem profissional**. 7ª edição. Editora: McGraw-Hill – Artmed, 2011. ISBN: 9788563308337.
- QUIERELLE, D. A. **Criando sites com HTML-CSS-PHP**. Construindo um projeto iniciante. São Paulo, 2012.
- SANTANA, L. R.; MARQUES, M. R., **Maus tratos e crueldade contra animais nos Centros de Controle de Zoonoses: aspectos jurídicos e legitimidade ativa do Ministério Público para propor Ação Civil Pública**. 2001. Disponível em: <[http://www.forumnacional.com.br/maus\\_tratos\\_CCz\\_de\\_Salvador.pdf](http://www.forumnacional.com.br/maus_tratos_CCz_de_Salvador.pdf) / > Acesso em: 20 mar. 2018.
- SANTOS, Adriano. **Entendendo um pouco do MySQL**. 2006. Disponível em: <https://www.devmedia.com.br/mysql-quem-e-voce/1752>.
- SLACK, N.; CHAMBERS, S.; HARDLAND, C.; HARRISON, A.; JOHNSTON, R. (1999). **Administração da Produção**: edição compacta. São Paulo: Atlas.
- SOARES, Michel dos Santos, **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Unipac - Universidade Presidente Antônio Carlos, Faculdade de Tecnologia e Ciências de Conselheiro Lafaiete, 2004.
- SOMMERVILLE, I. **Engenharia de Software**, Addison-Wesley Brasil, 8ª Edição, 2007
- TREML, Glauber. **Quais as vantagens do HTML5 para os usuários e desenvolvedores**. 2013. Disponível em:

<https://imasters.com.br/desenvolvimento/quais-as-vantagens-do-html5-para-os-usuarios-e-desenvolvedores/?trace=1519021197&source=single>

## **ANEXO I**

### **ATA 1: ENTREVISTA COM VOLUNTÁRIA DA ONG VIRA LATA VIRA VIDA**

Aos dias cinco do mês de abril de dois mil e dezoito às dez horas, na Fatec, na cidade de Americana – São Paulo, reuniram-se os alunos Maurílio Alexandre Lima e Adriano Benatti Junior do curso de Análise e Desenvolvimento de Sistemas para entrevista na ONG Vira Lata Vira Vida para compreensão de como funciona uma ONG em relação aos processos de entrada e saída de animais, desde a adoção até a captação do animal para ONG, afim de desenvolvimento do projeto da disciplina de TCC II do professor orientador Junior Lemos. A reunião foi presidida pelos alunos citados acima e contou com a participação da voluntária da ONG Paula e também professora da Fatec Americana da disciplina de Estatística. Os alunos Maurílio Alexandre Lima e Adriano Benatti Junior fizeram algumas perguntas como: Como funciona o processo de adoção de animal? E a voluntária Paula nos informou que é através da Zoonose da cidade e dos veterinários que trabalham naquele órgão, e com isso são escolhidos os animais que estão mais adequados para adoção, pois a ONG permite apenas o armazenamento de 300 cachorros, que ficam alocados em baias, e essas baias são definidas em uma quantidade de 10 cachorros por baias e são separadas por afinidades entre eles, cada baia é separada por grupos de cachorros, sendo que existe uma que é chamada de asilo, devido cachorros serem muito velhos e evitar futuros desentendimentos com outros, afim de poupá-los. Existem cachorros que ficam soltos, pois eles estão extremamente velhos e com isso, a ONG imagina que o animal não será adotado e está no fim da vida, por isso, de forma consciente em respeito ao animal, preserva a questão de liberdade. A mesma informou também, que para os animais entrarem na ONG é de forma obrigatória a castração, é feito pela própria ONG através dos voluntários que são veterinários e até mesmo fazem ação social para castração de animais da população. A voluntária Paula nos informou também sobre os Protetores Independentes, que são pessoas que não fazem parte da ONG, mas possui uma paixão excessiva por animais que vão resgatar em qualquer local e horário animais que estão abandonados, ou até mesmo sofrendo maltrato por “donos” ou sofrendo alguma crueldade, desde falta de alimentação, solidão e ruindade mesmo! A mesma citou que um cachorro foi usado para algum tipo de ritual onde os dentes foram

arrancados. Enfim, a mesma deixou bem claro, que os Protetores Independentes são pessoas que lutam pela causa e bem estar do animal, mas os animais resgatados não vão para a ONG, pois a ONG só aceita animais do Zoonose. A voluntária Paula, informou que se uma pessoa quer fazer a adoção do animal é necessário comparecer na ONG, preencher um questionário que contém as informações do futuro adotante para validação da ONG e se o mesmo possui condições de adotar o animal, com o objetivo que o animal não seja abandonado ou maltratado futuramente. A ONG vai até a casa do suposto adotante para verificar o local, para inspecionar e ver se é adequado para o animal viver de forma confortável e que não prejudique a saúde do animal. Quando um animal é aprovado para adoção o voluntário que assinou a adoção, o mesmo é responsável pela adoção, pois foi o mesmo que fez a validação anteriormente, se caso houver algum problema futuro o mesmo voluntário será questionado e tentará resolver, de acordo com as suas condições. O Maurílio Alexandre Lima perguntou sobre as leis de adoção no Brasil, a voluntária informou que existe leis de maltrato animal, mas não existe leis para o a adoção, e nos orientou irmos até a ONG e conversar com a coordenadora, pois ela possui mais informações específicas sobre. O Adriano Benatti Junior perguntou se a ONG possui algum site? A mesma respondeu que sim, e o site é o [www.viralataviravida.org.br](http://www.viralataviravida.org.br), tentamos acessar, mas não conseguimos devido a falta de conexão a internet. O Maurílio Alexandre Lima perguntou sobre quais funcionalidades iriam ajudar para um desenvolvimento de software de gestão de animais, para otimização dos processos em uma ONG de adoção, a mesma sugeriu um sistema que pudesse localizar os animais da região e em qual ONG está, a questão de perfis de cachorro com as suas histórias e fotos, filtros de pesquisas como: raça, idade do animal, cor do animal... A mesma disse que as histórias dos animais comovem muito a população e isso é uma das melhores maneiras de um animal ser adotado. Falamos também sobre funcionalidades como: preservar os históricos do animal no sistema, entrada e saída dos animais, exclusão do animal do sistema em caso de falecimento ou estado de “adotado” para cachorro adotado. Um cachorro já adotado pode ser devolvido, mas não seria o ideal, pois a validação para adoção tem que ser algo eficiente, se caso for devolvido o ideal é verificar o login que efetuou a validação. A voluntária Paula, nos convidou para dia cinco de maio que haverá um mutirão e nesse dia seria legal conhecermos a ONG, e a mesma passou seu e-mail para que a mesma pudesse nos enviar o questionário de adoção.

Então os alunos agradeceram pela oportunidade de aprender mais como funciona a ONG Enfim a entrevista houve a duração de 45 minutos, nada mais havendo a tratar, foi lavrada a presente ata, que vai assinada por mim, Maurílio Alexandro Lima, Adriano Bennatti e pela voluntária Paula.

Adriano Benatti Junior

---

Maurílio Alexandro Lima

---

Paula – Voluntária

---

## **ANEXO II**

### **ATA 2: ENTREVISTA COM CARLOS, PRESIDENTE DA AAANO.**

Sábado dia 21 de abril de 2018, Maurílio teve um encontro com o presidente da ONG Carlos em Nova Odessa para levantamento de requisitos, expondo a ideia do grupo e desenvolvimento juntamente com as informações captadas na entrevista. A entrevista se deve em função do desenvolvimento do trabalho de conclusão pela Faculdade de Tecnologia de Americana. Na primeira parte da entrevista Maurílio expõe a ideia do grupo que é realizar o desenvolvimento de web site para auxiliar na adoção dos animais e como divulgação da ONG. Referente a adoção o usuário que sentir vontade de adotar poderá escolher as características do animal, entrará em web site, fará a realização de um cadastro, que será avaliada. Caso seja aprovada será agendado um dia com a visita do adotante. Maurílio explica para Carlos referente a ONG que a professora Paula (ATA1) é voluntária. Carlos explica que em Nova Odessa é diferente, pois a AAANO possui canil, e é uma das únicas na região, por este motivo tem muitos abandonos em Nova Odessa. Maurílio pergunta como funciona o recolhimento dos animais. Carlos explica que é feito o resgate e levado para ONG apenas os animais em estado emergencial. Carlos fala exemplo de um caso que aconteceu há poucos dias. Maurílio pergunta quais bairros da cidade acontecem mais abandono, Carlos diz que são os bairros mais afastados da cidade, menos urbanizados como Guarapari e Fazenda Velha. Maurílio explica que não será feito um sistema ERP, que não iremos desenvolver por exemplo estoque de ração, as contabilidades da ONG. O sistema será voltado para adoção, com permissões hierárquicas dependendo função de cada voluntário dentro da ONG. Carlos explica que no sistema será focado apenas os cachorros adultos, pois são os mais difíceis de serem adotados. Os cachorros filhotes saem para adoção com facilidade. Maurílio pergunta para Carlos como funciona a identificação do cachorro, e Carlos explica que uma voluntária sabe o nome de todos os animais dentro da memória. Maurílio explica que pensando em sistema teremos que criar uma maneira de identificação, como por exemplo, uma fita na pata. Pois para registro em histórico no sistema será necessário a identificação do cachorro. Carlos explica que não são todos os animais que serão catalogados, pois existem animais muito agressivos, outros animais que estão com a idade muito avançada e caso o adotante não tenha foco de cem por cento para com os animais o mesmo virá a falecer. Carlos que tomará a decisão de quais animais cadastrar e quais não. Maurílio pergunta como

funciona, por exemplo, eu tenho um animal na minha casa e quero me desfazer por N motivos. Carlos informa que não será feito o recolhimento do animal, apenas com animais da rua em estados emergenciais. Carlos explica que no momento que o adotante assinou um termo e fez a adoção ele tem que arcar com a responsabilidade, porém caso o adotante realmente não queira mais dar paradeiro ao animal é aconselhado a divulgar em redes sociais. Maurílio pergunta como funciona referente aos protetores independentes, Carlos informa que a ONG possui advogado e que tudo é feito dentro da lei, porém entende que os protetores independentes fazem o que “dão na cabeça”, que pensam na vida do animal acima da lei. Carlos explica que em Nova Odessa não possui muitos protetores independentes. Carlos diz também que não é necessária uma campanha para protetores independentes pois a ONG nas redes sociais dá dicas de como cuidar dos animais, quando precisam de lar temporário, e Maurílio questiona sobre. Para lar temporário o adotante fará um cadastro respondendo a alguns requisitos, dizendo disponibilidade de quantos dias ficará com o animal. É feita uma entrevista, um reconhecimento na casa do solicitante e divulgando o animal nas redes sociais até encontrar o dono fixo. O voluntário da ONG que faz o intermédio para devolução do animal. Maurílio pergunta como funciona o processo de adoção de um animal que acabou de chegar na ONG, qual a primeira coisa a se fazer após o resgate. Carlos explica que é feita uma avaliação do veterinário, condições de saúde do animal e castração. Após o animal ou fica no veterinário na clínica aguardando ou vai para o canil. Carlos explica sobre o canil coletivo em que os animais ficam juntos. Maurílio explica que terá que no site terá uma opção para fazer login no próprio web site, pois a pessoa que tiver o primeiro contato com o animal terá de realizar o cadastro. Carlos explica que o cadastro terá que ser feito somente pela ONG, pois é lá que é feito. Maurílio faz perguntas sobre qual tipo de informações será cadastrada. Carlos explica que o veterinário não fará nenhum cadastro, ele passará as informações para os voluntários da ONG e o cadastro será feito somente lá. Maurílio explica sobre o cadastro dos animais por baia, para poder ter maior organização dos animais, em quais baias eles estariam ou não. Carlos explica que alguns animais têm baia fixa, porém tem animais que fazem um rodizio em função de rotatividade e tamanho, algumas vezes brigas entre os próprios animais. Maurílio explica que servirá como um controle para saber quais animais foram adotados. Carlos explica que tem vontade de realizar o controle e organização para melhor uso do sistema.

Pois com o sistema funcionando com baias numeradas e alto controle, a funcionalidade e organização do lugar irá melhorar. Carlos explica que os voluntários vão ao canil todo domingo, Maurílio fala que pode ser a longo prazo a organização das baias pois entende que é complicado mudar a maneira como eles trabalham na ONG. Maurílio questiona agora a parte da adoção. Carlos informa que é realizada uma entrevista verbal, onde tem um formulário preenchido, junto da documentação do adotante. Carlos explica sobre o termo de adoção que agora é chamado de contrato, pois com a ajuda de uma advogada foi realizado um termo jurídico mais rigoroso. Carlos explica que a ONG possui uma equipe de visita pós adoção para saber se o adotante está cuidando do animal de acordo com o termo. Os dois presentes na reunião discutem sobre adotantes que não poderão mais adotar, por mal tratos ao animal. Carlos explica que é feito uma "lista negra". Maurílio explica que seria legal atrelar se o adotante está apto a adoção pelo CPF. O sistema trará o motivo do bloqueio pelo qual não conseguirá realizar a adoção. Carlos explica que possui um grupo na região no qual são espalhadas essas pessoas que realizam mal tratos e não poderá adotar. Maurílio diz que teve a ideia de um sistema regional que trará as informações pelo CPF dos adotantes. Carlos explica que pensam em uso de microchips para quando os animais saem da ONG, pois assim teria melhor acompanhamento, porém como Nova Odessa é uma cidade de baixa renda a ONG não tem muita ajuda para bancar tal tecnologia. Maurílio explica que teria como realizar através da identificação do animal, porém Carlos informa que no microchip vai a informação do adotante e não os dados do animal. Maurílio explica para Carlos referente a habilidade dos alunos explicando que não temos grande experiência, Carlos entende e diz que acha muito legal, e que nosso trabalho será muito bem visto pela banca do TCC. Maurílio explica sobre os TCC1 e TCC2, como funciona os trabalhos na faculdade. Diz o porquê escolhemos a AAANO pois o local está precisando de muita ajuda pois está mais precário. Os dois presentes na entrevista falam sobre as outras ONGS e como funciona, Carlos explica que em Nova Odessa a Zoonoses que deveria realizar o trabalho que eles fazem, porém, a ONG que acaba recebendo todo o trabalho, que na verdade seria dever do estado. Eles falam sobre o vereador Leitinho que sempre dedicou sua causa ajudar os animais, porém como é vereador da oposição seus projetos de leis dificilmente são aprovados. Carlos conta como briga contra a prefeitura por mais direitos na causa. Maurílio diz que escolhemos a causa, explica que teremos o prazo até o final do ano para

desenvolver. Maurílio comenta sobre o estágio voluntário para podermos nos formar e ganharmos horas complementares. Maurílio fala sobre a documentação de estágio e Carlos diz que assinará o estágio. Maurílio explica sobre os outros dois alunos que não estavam presentes não entrevista. Os dois presentes conversam sobre os cursos da FATEC e ETEC, Carlos inclusive diz que cursou técnico em informática. Maurílio diz que usaremos linguagem PHP. Os dois presentes também concordam que é mais viável o sistema ser web, porém com cadastro e login, desenvolvendo um sistema. Maurílio diz que estamos na fase de levantamento de requisitos.

Maurílio Alexandro Lima

---

Carlos – Presidente AAANO

---

# Manual de Usuário

(Software Gestão de Adoção de Animais)



---

## **OBJETIVO**

---

Através deste material registrar de forma clara, os processos para utilização do sistema, auxiliando no processo de aprendizagem na utilização do mesmo.

---

## **SUMÁRIO**

---

<b>1 – ACESSO AO SISTEMA</b> .....	<b>22</b>
<b>2 – PERMISSÕES DE ACESSO NO SISTEMA</b> .....	<b>22</b>
<b>2.1 – REGRAS PARA UTILIZAÇÃO DO SISTEMA</b> .....	<b>22</b>
<b>3 – CADASTROS DE USUÁRIOS</b> .....	<b>22</b>
<b>3.1 – CADASTRAR USUÁRIO</b> .....	<b>22</b>
<b>3.2 – VISUALIZAR USUÁRIO</b> .....	<b>22</b>
<b>3.3 – EDITAR USUÁRIO</b> .....	<b>22</b>
<b>3.4 – EXCLUIR USUÁRIO</b> .....	<b>22</b>
<b>4 – CADASTROS DE ANIMAIS</b> .....	<b>22</b>
<b>4.1 – CADASTRAR ANIMAIS</b> .....	<b>22</b>
<b>4.2 – VISUALIZAR ANIMAIS</b> .....	<b>22</b>
<b>4.3 – EDITAR ANIMAIS</b> .....	<b>22</b>
<b>4.4 – EXCLUIR ANIMAIS</b> .....	<b>22</b>
<b>5 – SOLICITAÇÃO DE ADOÇÃO</b> .....	<b>22</b>
<b>5.1 – CRIAR SOLICITAÇÃO DE ADOÇÃO</b> .....	<b>22</b>
<b>5.2 – APROVAR SOLICITAÇÃO DE ADOÇÃO</b> .....	<b>22</b>
<b>5.3 – REPROVAR SOLICITAÇÃO DE ADOÇÃO</b> .....	<b>22</b>
<b>6 – CADASTRO DE VOLUNTÁRIO – SEJA VOLUNTÁRIO</b> .....	<b>22</b>
<b>7 – CONHEÇA NOSSOS ANIMAIS</b> .....	<b>22</b>
<b>7 – RELATÓRIOS</b> .....	<b>22</b>
<b>8 – DIAGRAMA ( MAPA DO SITE )</b> .....	<b>22</b>

---

### 3 – CADASTROS DE USUÁRIOS

O Cadastro de usuário pode ser feito de duas formas, ou o usuário acessa o sistema e clica no menu “Seja Voluntário” e aguarda alguém da ONG realizar a ativação do acesso para o site ou algum administrador da ONG realiza o cadastro do usuário.

#### 3.1 – CADASTRAR USUÁRIO

- Realizar acesso ao sistema com o perfil de “Super Administrador ou Administrador” e ir na aba “Gerenciar Pessoas”, conforme demarcação em vermelho.



- Após clicar em Gerenciar Pessoas, para visualizar os voluntários já cadastrados, clique em “Funcionários”, para cadastrar um novo usuário é necessário clicar em “Novo” e aparecerá o formulário para criação do usuário. O Administrador deve digitar os dados e confirmar, clicando no botão de “Cadastrar”.

**Nome**  
Qual seu nome? \*

**RG/CPF**  
Digite seu CPF \*

**Sexo**  
Selecione seu sexo

**Data de Nascimento**  
dd/mm/aaaa

**Celular**  
Celular para contato \*

**Email**  
Digite seu email \*

**Senha**  
Digite sua senha \*

**Confirmar senha**  
Confirme sua senha \*

**Tipo de usuário**  
Super Administrador

**Rua**  
Qual sua rua? \*

**Número da casa**  
Número da casa \*

**Bairro**  
Qual seu bairro? \*

**CEP**  
Qual seu cep? \*

**Cidade**  
Mora em qual Cidade? \*

**Estado**  
Acre

**Escolha uma imagem de perfil**  
Escolher arquivo Nenhum arquivo selecionado

**Tipo de Moradia**  
-- Qual o tipo da sua moradia? --

**Tem Quintal**  
-- Em sua moradia tem quintal? --

**Se voluntariar como:**  
Voluntário da ONG

✓ Cadastrar

### Dados para serem preenchidos:

- ✓ Nome: É o nome da pessoa que terá acesso ao sistema.
- ✓ RG/CPF: Identificação da pessoa.
- ✓ Sexo: Masculino/Feminino.
- ✓ Endereço: (Rua, Número da casa, Bairro, CEP, Cidade, Estado).
- ✓ Data de Nascimento: É a data de nascimento da pessoa.
- ✓ Celular: É o telefone de contato da pessoa.
- ✓ Senha: Senha para acesso ao sistema.
- ✓ Confirmação de Senha: para alguma alteração no cadastro da pessoa, sempre deverá digitar a senha e confirmar, para validação dos dados.
- ✓ Tipo de Usuário: é a permissão de acesso no sistema.
- ✓ Tipo de Moradia: tipo de residência do usuário.
- ✓ Tem Quintal: Se o voluntário tem quintal.
- ✓ Se voluntariar como: Refere-se ao tipo de voluntariado (adotante ou voluntário da ONG).

*Obs.: Quando é realizado o cadastro de usuário, é necessário realizar a ativação dele para que o usuário consiga ter acesso ao sistema.*

### 3.2 – VISUALIZAR USUÁRIO

- Para realizar a visualização de um cadastro de usuário do sistema, o administrador deverá clicar em “Gerenciar Pessoas” e depois “Funcionários” e procurar o usuário desejado clicando no botão azul, abaixo da foto de perfil:

- Após clicar no botão azul, será aberto o formulário que contém as informações do usuário para visualização:

Adriano Benatti

---

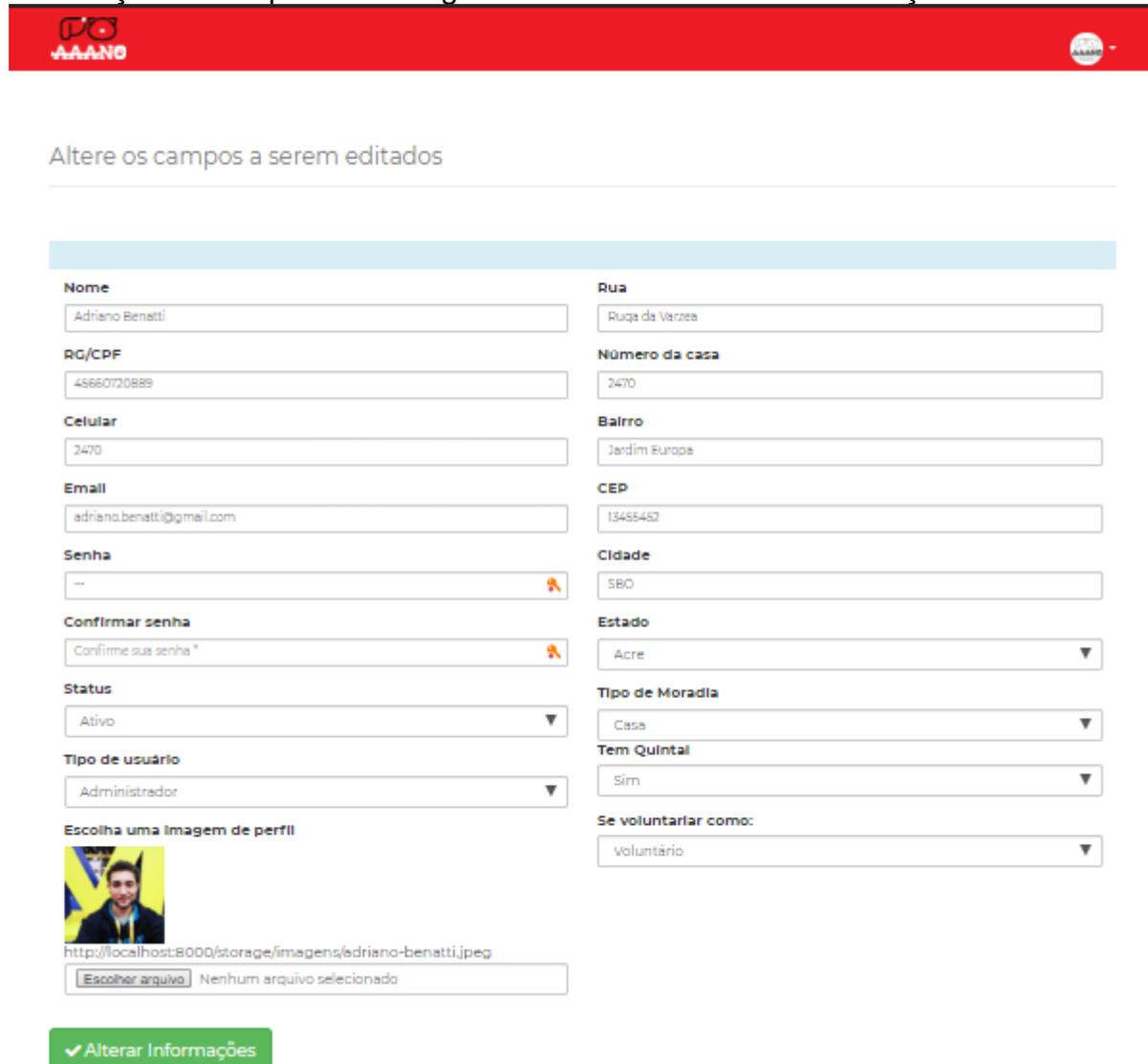
Informações

Nome:	Adriano Benatti
Email:	adriano.benatti@gmail.com
RG/CPF	45660720889
Rua:	Ruça da Varzea
Nº:	2470
Bairro:	Jardim Europa
Cidade:	SBO
Moradia:	Casa
Quintal:	Sim
Nível:	Administrador
Participação na ONG:	Voluntário

### 3.3 – EDITAR USUÁRIO

- Para realizar a edição de um cadastro de usuário do sistema, o administrador deverá clicar em “Gerenciar Pessoas” e depois “Funcionários” e procurar o usuário desejado clicando no botão amarelo, abaixo da foto de perfil:

- Após clicar no botão amarelo, será aberto o formulário que contém as informações do usuário, basta verificar o que deseja realizar e clicar no botão “Alterar Informações”. Sempre será obrigatório atualizar senha e confirmação de senha.



AAANO

Altere os campos a serem editados

<b>Nome</b> <input type="text" value="Adriano Benatti"/>	<b>Rua</b> <input type="text" value="Rua da Varzea"/>
<b>RG/CPF</b> <input type="text" value="48660720889"/>	<b>Número da casa</b> <input type="text" value="2470"/>
<b>Celular</b> <input type="text" value="2470"/>	<b>Bairro</b> <input type="text" value="Jardim Europa"/>
<b>Email</b> <input type="text" value="adriano.benatti@gmail.com"/>	<b>CEP</b> <input type="text" value="13465452"/>
<b>Senha</b> <input type="password" value="..."/>	<b>Cidade</b> <input type="text" value="SBO"/>
<b>Confirmar senha</b> <input type="password" value="Confirme sua senha *"/>	<b>Estado</b> <input type="text" value="Acre"/>
<b>Status</b> <input type="text" value="Ativo"/>	<b>Tipo de Moradia</b> <input type="text" value="Casa"/>
<b>Tipo de usuário</b> <input type="text" value="Administrador"/>	<b>Tem Quintal</b> <input type="text" value="Sim"/>
<b>Escolha uma imagem de perfil</b>  <small>http://localhost:8000/storage/imagens/adriano-benatti.jpeg</small> <input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado	<b>Se voluntariar como:</b> <input type="text" value="Voluntário"/>

✓ Alterar Informações

### 3.4 – EXCLUIR USUÁRIO

- Para realizar a exclusão do usuário, será permitido somente pelo usuário com o perfil “Administrador”, podendo excluir usuários “Voluntário” e “Super Administrador” podendo excluir todos os outros perfis. Para realizar a exclusão o “Administrador” deverá clicar em “Gerenciar Pessoas” e depois “Funcionários” e procurar o usuário desejado clicando no botão vermelho, abaixo da foto de perfil:

AAANO

Adoções

Gerenciar Pessoas

Voluntários

Novo

Animais

Relatórios



Maurilio Lima



Adriano Benatti



Amanda



Super Administrador

***Após a exclusão o usuário não aparecerá mais no quadro de funcionários.***

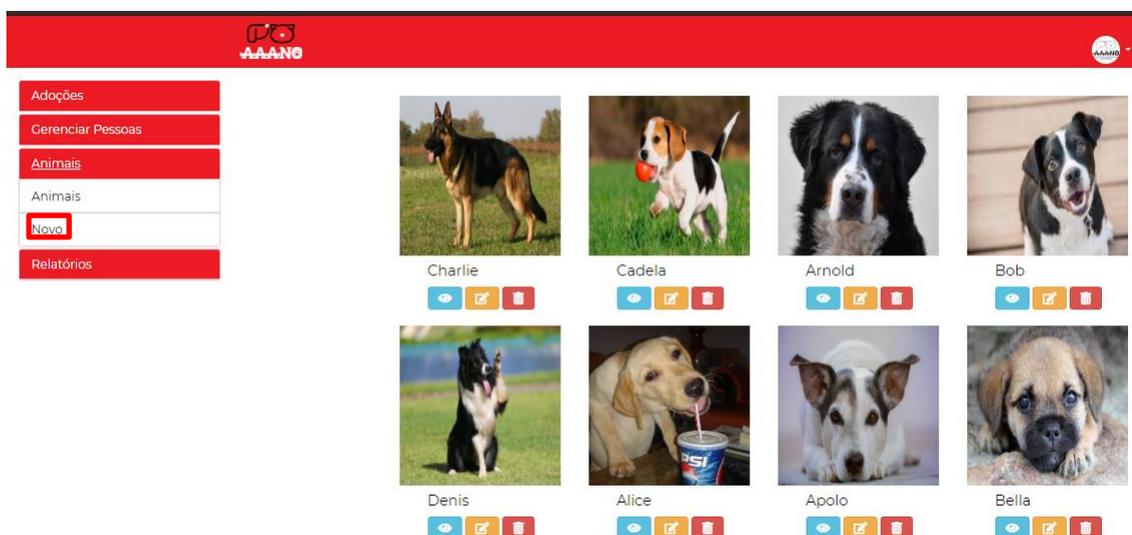
---

## 4 – CADASTROS DE ANIMAIS

Cadastros de animais só serão realizados através do perfil “Administrador” e “Super Administrador”. Quando o animal é cadastrado, automaticamente ele já aparece disponível para a realização da solicitação de adoção.

### 4.1 – CADASTRAR ANIMAIS

- Realizar acesso ao sistema com o perfil de “Super Administrador ou Administrador” e ir na aba “Animais” e depois em “Novo” conforme demarcação em vermelho. Quando é clicado em “Animais” será exibido todos os animais cadastrados no sistema.



- Após clicar em “Novo” será exibido o formulário para cadastro dos animais para a adoção.

Cadastre um animal

<b>Nome</b> Nome *	<b>Vacinado</b> Selecione uma opção
<b>Raça</b> Raça *	<b>Deficiência</b> Não
<b>Idade</b> Idade *	<b>Data de registro</b> dd/mm/aaaa
<b>Cor</b> Cor *	<b>Castrado</b> Selecione uma opção
<b>Porte</b> Porte *	<b>Patologia</b> Selecione uma opção
<b>Sexo</b> Selecione uma opção	<b>Escolha uma Imagem</b> Escolhar arquivo Nenhum arquivo selecionado
<b>Situação</b>	<b>Descrição</b>

Concordo com os termos e condições

[Cadastrar](#)

- Após o preenchimento de todos os dados acima, o animal já estará cadastrado e disponível para a realização da solicitação de adoção.

#### 4.2 – VISUALIZAR ANIMAIS

- Para realizar a visualização de um cadastro de animal no sistema, o administrador deverá clicar em “Animais” e depois “Animais” e procurar o animal desejado clicando no botão azul, abaixo da foto de perfil:

AAANG

Adoções

Adoções

Gerenciar Pessoas

Animais

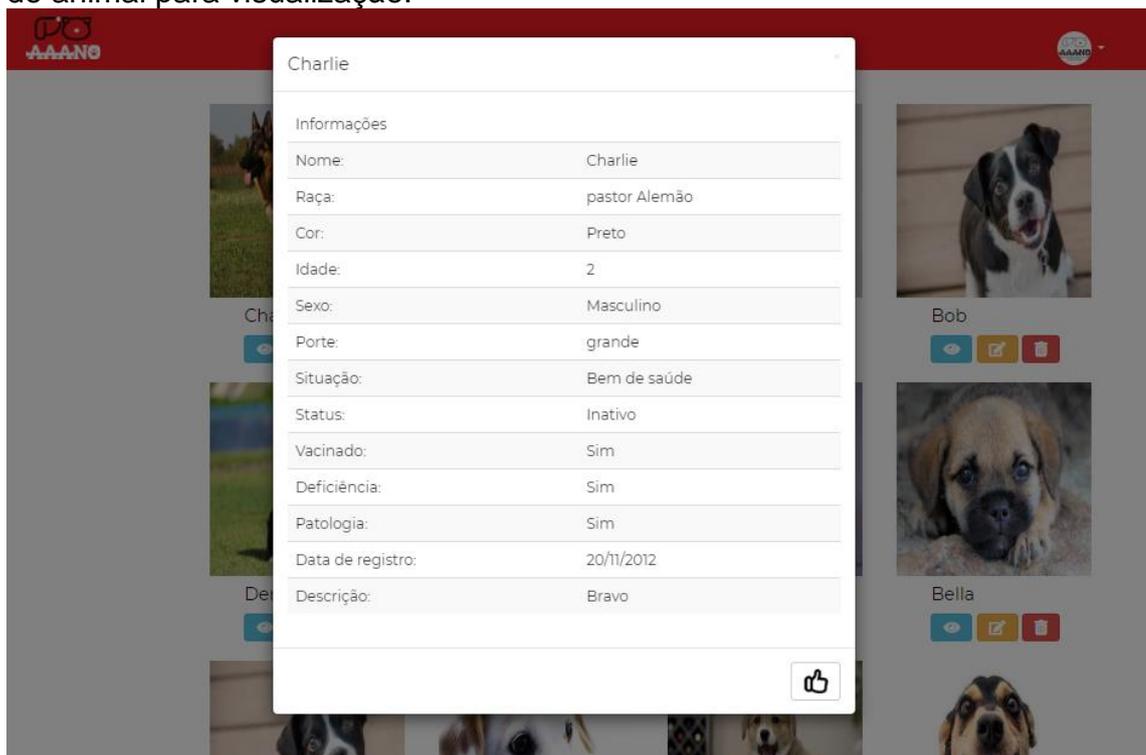
Animais

Novo

Relatórios

			
Charlie	Cadela	Arnold	Bob
			

- Após clicar no botão azul, será aberto o formulário que contém as informações do animal para visualização.



### 4.3 – EDITAR ANIMAIS

- Para realizar a edição de um cadastro de animais do sistema, o administrador deverá clicar em “Animais” e depois “Animais” e procurar o animal desejado clicando no botão amarelo, abaixo da foto de perfil:



- Após clicar no botão amarelo, será aberto o formulário que contém as informações do animal, basta verificar o que deseja realizar e clicar no botão “Alterar Informações”. Sempre será obrigatório atualizar senha e confirmação de senha.

Atualize os dados um animal

<p><b>Nome</b></p> <input type="text" value="Charlie"/>	<p><b>Vacinado</b></p> <input type="text" value="Sim"/>
<p><b>Raça</b></p> <input type="text" value="pastor Alemão"/>	<p><b>Deficiência</b></p> <input type="text" value="Sim"/>
<p><b>Idade</b></p> <input type="text" value="2"/>	<p><b>Data de registro</b></p> <input type="text" value="20/11/2012"/>
<p><b>Cor</b></p> <input type="text" value="Preto"/>	<p><b>Castrado</b></p> <input type="text" value="Sim"/>
<p><b>Porte</b></p> <input type="text" value="grande"/>	<p><b>Patologia</b></p> <input type="text" value="Sim"/>
<p><b>Sexo</b></p> <input type="text" value="Macho"/>	<p><b>Escolha uma imagem</b></p>  <p><a href="http://localhost:8000/storage/imagens/zadriano.jpeg">http://localhost:8000/storage/imagens/zadriano.jpeg</a></p> <input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado
<p><b>Situação</b></p> <input type="text" value="Bem de saúde"/>	<p><b>Status</b></p> <input type="text" value="Inativo"/>
	<p><b>Descricao</b></p> <input type="text" value="Bravo"/>

Atualizar dados

## 4.4 – EXCLUIR ANIMAIS

- Para realizar a exclusão o “Administrador” deverá clicar em “Animais” e depois “Animais” e procurar o usuário desejado clicando no botão vermelho, abaixo da foto de perfil:




Adoções

Gerenciar Pessoas

**Animais**

Relatórios



Charlie



Cadela



Arnold



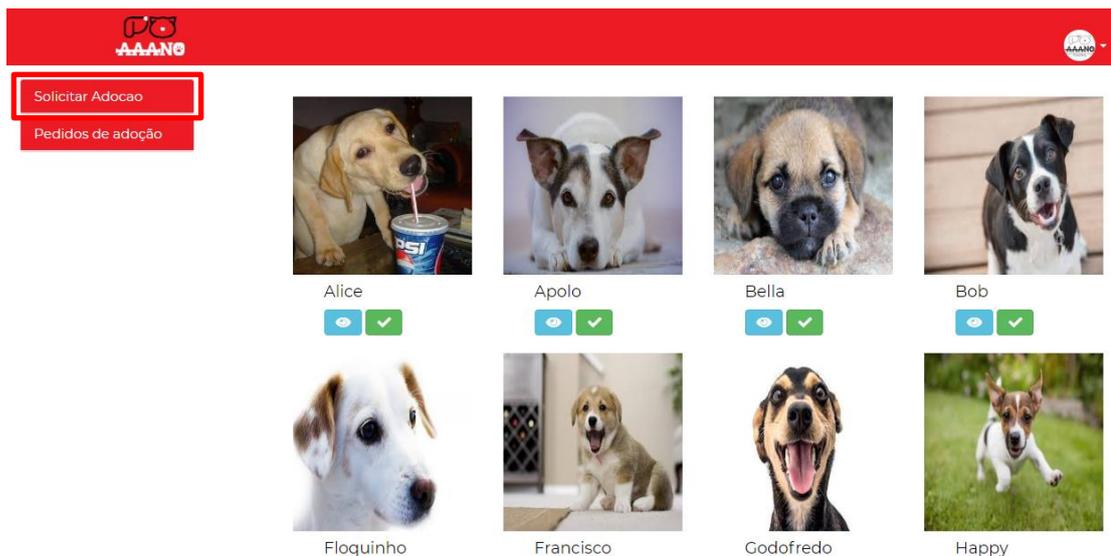
Bob

**Após a exclusão o animal não aparecerá mais no quadro de animais para adoção.**

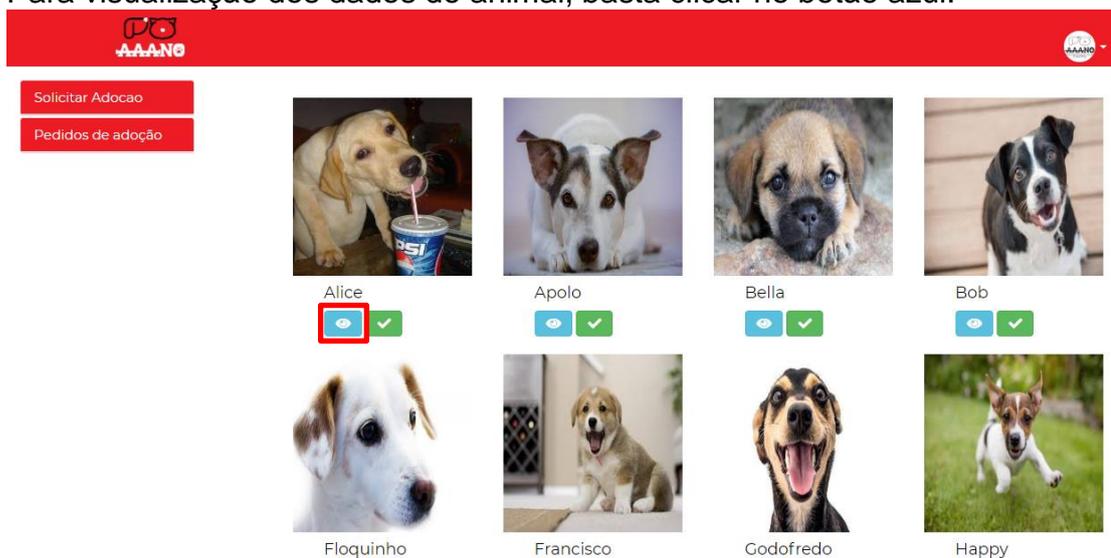
## 5 – SOLICITAÇÃO DE ADOÇÃO

### 5.1 – CRIAR SOLICITAÇÃO DE ADOÇÃO

- A solicitação de adoção só é permitida mediante ao cadastro de usuário com o tipo “Voluntário”, pois assim o voluntário deverá realizar o login e buscar os animais que estão disponíveis para adoção, isso pode ser feito clicando na aba “Solicitar Adoção”, conforme demarcação em vermelho da imagem abaixo:



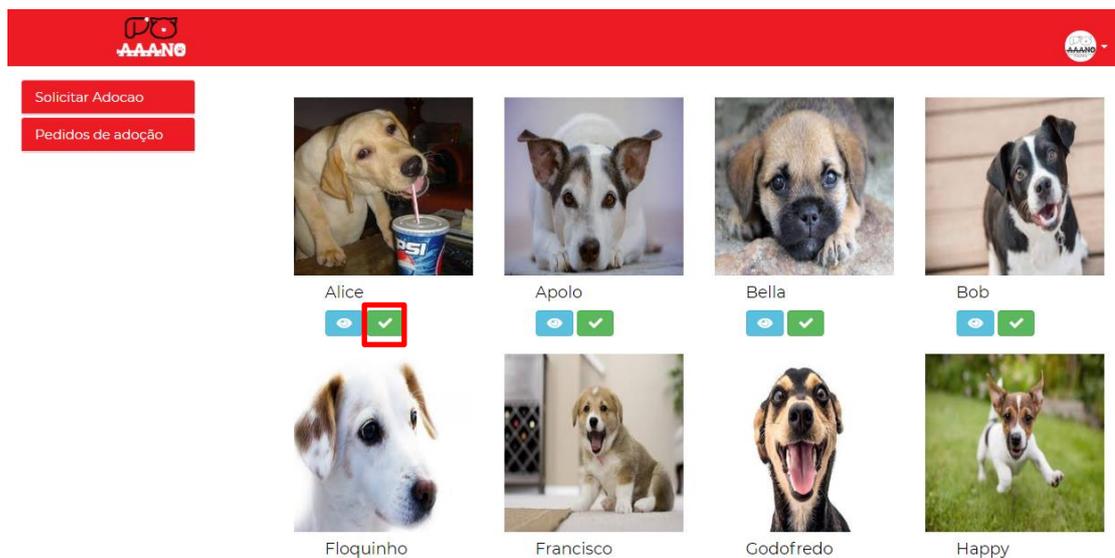
- Para visualização dos dados do animal, basta clicar no botão azul.



- Após clicar no botão azul, será exibido um formulário com todas as características do animal.

Informações	
Nome:	Alice
Idade:	2
Raça:	Vira-lata
Cor:	bege
Porte:	1
Deficiência:	Não
Sexo:	Fêmea
Registro de Cadastro:	25/04/2019
Patologia:	Sim
Situação do Animal:	Encontrada em Nova Odessa no bairro no Jardim Santa Rosa, com ferimentos nas patas.
Vacinado:	Sim
Data de Registro:	25/04/2019

- Para solicitar a adoção, basta fechar o formulário aberto com as características do animal e clicar no botão verde.



- Após clicar no botão verde, será exibida a imagem abaixo informando que a solicitação de adoção foi criada e deverá aguardar a aprovação do animal por intermédio do usuário “administrador”.

	
Denis	Alice
Adoção esperando aprovação!	Adoção esperando aprovação!
22/05/2019	25/05/2019

## 5.2 – APROVAR SOLICITAÇÃO DE ADOÇÃO

- Para realizar a aprovação de adoção o usuário “Administrador ou Super Administrador” precisa estar logado no sistema. Após logado, basta clicar na aba “adoções” que será exibida as solicitações de adoções. Abaixo tem duas solicitações para realizar aprovação, uma da voluntária “Wanessa” e outra do voluntário com o nome de “Voluntário”, para aprovar a solicitação basta clicar no botão verde.

- Após clicar no botão verde, será exibida a imagem abaixo informando que a solicitação de adoção foi aprovada e atualizará o status na área do voluntário solicitante e o animal sumirá do quadro de solicitações de adoções.

- Após a aprovação da solicitação de adoção for aprovada/rejeitada o voluntário deverá acessar o sistema e entrar na aba “Pedidos de Adoção” e verificar o status, se estiver como “Adoção Liberada” significa que o voluntário pode adotar o animal.

Denis	Alice
Adoção esperando aprovação!	Adoção liberada
22/05/2019	25/05/2019

### 5.3 – REPROVAR SOLICITAÇÃO DE ADOÇÃO

- Para realizar a reprovação de adoção o usuário “Administrador ou Super Administrador” precisa estar logado no sistema. Após logado, basta clicar na aba “adoções” que será exibida as solicitações de adoções. Abaixo tem duas solicitações para realizar aprovação, uma da voluntária “Wanessa” e outra do voluntário com o nome de “Voluntário”, para reprovar a solicitação basta clicar

no botão vermelho.

AAANG

Adoções

Gerenciar Pessoas

Animais

Relatórios

### Solicitações de adoção

Amanda

AAANG

- Após clicar no botão vermelho, será exibida a imagem abaixo informando que a solicitação de adoção foi reprovada e atualizará o status na área do voluntário solicitante e o animal sumirá do quadro de solicitações de adoções.

AAANG

Adoções

Gerenciar Pessoas

Animais

Relatórios

### Solicitações de adoção

Não há solicitações no momento

AAANG

- Após a aprovação da solicitação de adoção for aprovada/rejeitada o voluntário deverá acessar o sistema e entrar na aba “Pedidos de Adoção” e verificar o status, se estiver como “Adoção Liberada” significa que o voluntário pode adotar o animal.

AAANG

Solicitar Adocao

Pedidos de adoção

	
Denis	Alice
Adoção recusada	Adoção liberada
25/05/2019	25/05/2019

AAANG

## 6 – CADASTRO DE VOLUNTÁRIO – SEJA VOLUNTÁRIO

- No menu “Seja Voluntário” qualquer pessoa terá acesso para se voluntariar a ONG, o cadastro é idêntico ao [3.1 – CADASTRAR USUÁRIO](#). Ao finalizar o cadastro, a equipe administrativa da ONG analisará o cadastro e verificará o tipo de voluntariado que a pessoa escolheu, se for “Voluntário da ONG” não é necessário realizar a ativação do cadastro no sistema, pois esse tipo refere-se a voluntariado na ONG. Se for “Adotante” será necessário ativar o cadastro, pois esse tipo de voluntário deseja realizar a adoção de animal.

Preencha os dados para se cadastrar

Preencha o formulário abaixo para se cadastrar! Lembre-se que o cadastro está sujeito a aprovação do(a) Administrador(ine).

<b>Nome</b> Qual seu nome? *	<b>Rua</b> Qual sua rua? *
<b>RG/CPF</b> Digite seu CPF *	<b>Número da casa</b> Número da casa *
<b>Sexo</b> Selecione seu sexo	<b>Bairro</b> Qual seu bairro? *
<b>Data de Nascimento</b> dd/mm/aaaa	<b>CEP</b> Qual seu cep? *
<b>Celular</b> Celular para contato *	<b>Cidade</b> More em qual cidade? *
<b>Email</b> Digite seu email *	<b>Estado</b> -- Qual seu estado? --
<b>Tipo de Moradia</b> -- Qual o tipo de sua moradia? --	<b>Senha</b> Digite sua senha *
<b>Tem Quintal</b> -- Em sua moradia tem quintal? --	<b>Confirmar senha</b> Confirme sua senha *
<b>Se voluntariar como:</b> Voluntário da ONG Voluntário da ONG Adotante	
<input type="button" value="Solicitar"/>	

## 7 – CONHEÇA NOSSOS ANIMAIS

- Esse menu serve para realizar buscas de animais dentro do sistema, para isso você deve escolher os critérios através dos filtros:



- Após escolher os critérios basta clicar no botão pesquisar para buscar um animal com as características escolhidas.



- Para verificar as informações dos animais basta clicar no botão azul.

Selecione o filtro para escolher as características do animal desejado!

Raça ▾ Idade ▾ Cor ▾ Porte ▾ [Pesquisar](#)



Charlie

[Verificar Informações](#)

- Será aberta uma janela com os detalhes do animal, conforme imagem abaixo:

Charlie

Informações

Nome:	Charlie
Idade:	2
Raça:	pastor Alemão
Cor:	Preto
Porte:	grande
Deficiência:	Sim
Sexo:	Masculino
Patologia:	Sim
Situação do Animal:	Bem de saúde
Vacinado:	Sim

Localização: Contato:

## 7 – RELATÓRIOS

- Em relatórios será onde o administrador poderá verificar quais são os animais que foram adotados, quais não foram e os animais que estão sem solicitações de adoções. Para isso foi realizado 2 relatórios e servirá para a realização do gerenciamento administrativo da ONG.
- Relatório Adoções:  
Na tela abaixo, o administrador deverá inserir a data de início e a data de fim e clicar em Pesquisar.

AAANO

Adoções

Gerenciar Pessoas

Animais

Relatórios

Adoções

Animais para adoção

### Relatório Geral

Data de início: 01/01/2019

Data de fim: 26/05/2019

**Pesquisar**

- Após clicar em pesquisar será exibido o relatório com todos os dados:

relatorio 1 / 1

AAANO Associação Amigos dos Animais de Nova Odessa

Relatório De 01/01/2019 até 26/05/2019

#### Adoções realizadas

Nome do adotante	Nome do animal	Raça	Data de adoção	Status animal
Adriano Benatti	Charlie	pastor Alemão	10/02/2019	I
Adriano Benatti	Cadela	Pincher	10/02/2019	I
Super Administrador	Bob	Vira-lata	17/04/2019	I
Amanda	Alice	Vira-lata	25/05/2019	I

#### Pedidos de Adoção

Nome do adotante	Nome do animal	Raça	Data do pedido de adoção	Status animal
------------------	----------------	------	--------------------------	---------------

#### Pedidos de Adoção recusados

Nome do adotante	Nome do animal	Raça	Data de recusa	Status animal
Adriano Benatti	Cadela	Pincher	10/02/2019	I
Adriano Benatti	Cadela	Pincher	10/02/2019	I
Amanda	Denis	Vira-lata	25/05/2019	A

- Relatório Animais para Adoção:  
Na tela abaixo, o administrador deverá inserir a data de início e a data de fim e clicar em Pesquisar.

AAANO

Adoções

Gerenciar Pessoas

Animais

Relatórios

### Relatório Animais

Data de início: 01/01/2019

Data de fim: 26/05/2019

**Pesquisar**

- Após clicar em pesquisar será exibido o relatório com todos os dados:

**Associação Amigos dos Animais de Nova Odessa**

**Relatório** De 01/01/2019 até 26/05/2019

Animais esperando adoção									
Nome do animal	Raça	Cor	Idade	Deficiência	Castrado	Sexo	porta	patologia	Data de Registro
Dennis	Vira-lata	bege	2	N	S	M	pequeno	S	27/04/2019
Apolo	Vira-lata	mechado	2	N	S	M	2	S	10/12/2019
Bella	Vira-lata	bege	1	N	N	F	2	N	22/01/2019
Bub	Vira-lata	mechado	3	N	S	M	pequeno	S	18/01/2019
Floresinha	Vira-lata	branco	2	N	S	M	2	S	12/03/2019
Francisco	Vira-lata	bege	1	N	N	M	1	S	17/03/2019
Godofredo	Vira-lata	preto	1	N	S	M	2	S	10/10/2018
Happy	Vira-lata	mechado	1	N	N	F	1	S	25/04/2019
Jaha	Vira-lata	bege	1	S	S	M	1	N	18/04/2019
Julie	Vira-lata	preto	4	N	S	F	3	S	01/01/2019
Lily	Vira-lata	preto	1	N	N	F	1	S	17/04/2019
Nina	Vira-lata	bege	2	N	S	F	2	S	01/01/2019
Olga	Vira-lata	branco	2	N	S	F	2	N	12/02/2019
Pipoca	Vira-lata	bege	2	N	N	F	2	S	28/02/2019
Wolf	Vira-lata	preto	2	S	N	M	3	S	22/03/2019
<b>Total animais disponíveis</b>								<b>15</b>	

## 8 – DIAGRAMA ( MAPA DO SITE )

