

CENTRO PAULA SOUZA

ETEC LAURO GOMES

Desenvolvimento de Sistemas Integrado ao Ensino Médio

Kaick Kenithi Nishiya

Kauan Matos Lopes da Silva

Moacir José da Silva Filho

Nathan Enrico Romero

Pedro Goldoni Magri

Victor Rayan Souza Ramos

AMICÃO

São Bernardo do Campo

2022

**Kaick Kenithi Nishiya
Kauan Matos Lopes da Silva
Moacir José da Silva Filho
Nathan Enrico Romero
Pedro Goldoni Magri
Victor Rayan Souza Ramos**

AMICÃO

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da Etec Lauro Gomes, orientado pela Prof. Rosa Mitiko Shimizu, como requisito parcial para obtenção do título de técnico em desenvolvimento de sistemas.

**SÃO BERNARDO DO CAMPO, SÃO PAULO
2022**

DEDICATÓRIA

A conclusão deste trabalho resume-se em dedicação, dedicação que vi ao longo dos anos em cada um dos professores deste curso, a quem dedico este trabalho.

AGRADECIMENTOS

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos. Thomas Shelby, meu querido, sou muito grato por seu apoio e amor, sem você este TCC não teria chegado ao fim. Obrigado por sua gentileza e compreensão.

“Qualquer **tecnologia** suficientemente
avançada é equivalente à mágica”

Clarke

RESUMO

É notório que este estudo examina as dificuldades das pessoas que têm interesse em adotar um animal de estimação os quais servem como uma ótima companhia sendo considerados os melhores amigos do homem e relacionados a diversos benefícios pessoais, e realiza a implementação do um sistema contemplado por um aplicativo móvel e um site, cujos permitam facilitar o processo de adoção ou apadrinhamento de animais pelos usuários, através da construção de uma interface a qual fornece uma comunicação mais eficiente entre as instituições de abrigos de animais e os indivíduos que tem a pretensão em adota-los. Assim sendo, o estudo descreve dificuldades de acesso, segurança e qualidade dos serviços prestados por abrigos de animais. Portanto, foi utilizado o artificio da pesquisa de campo e revisão bibliográfica aplicado ao processo de investigação afirmando o posicionamento de que as pessoas têm o interesse de adotar animais que nas ruas estão recorrendo um sistema computacional, haja vista as inconveniências dos processos tradicionais que são a elevada burocracia e que abrigos não são tão facilmente encontrados.

Palavras-chave: Aplicativo-móvel; Site; Adoção; Animais; Apadrinhamento.

ABSTRACT

It is clear that this study examines the difficulties of people who are interested in adopting a pet, which serves as a great companion being considered man's best friend and related to various personal benefits, and performs the implementation of a system contemplated by a mobile application and a website, which facilitate the process of adoption or sponsorship of animals by users, through the construction of an interface which provides a more efficient communication between animal shelter institutions and individuals who intend to adopt them. Therefore, the study describes difficulties in access, security and quality of services provided by animal shelters. Therefore, the artifice of field research and bibliographical review applied to the investigation process was used, stating the position that people are interested in adopting animals that are using a computational system on the streets, given the inconveniences of traditional processes that are the high bureaucracy and that shelters are not so easily found.

Keywords: Mobile-app; Site; Adoption; Animals; Sponsorship.

LISTA DE ILUSTRAÇÕES

Figura 1 - Gráfico de idade dos entrevistados.....	17
Figura 2 - Efeitos econômicos da pandemia	18
Figura 3 - Índice de posse de animais.....	18
Figura 4 - Forma de obtenção do animal	19
Figura 5 - Lazer em convívio com animal.....	19
Figura 6 - Adoção de animal mau tratado	20
Figura 7 - Aptidão a cuidar de animais.....	20
Figura 8 - Burocracia na adoção	21
Figura 9 - Ideais de melhoria.....	22
Figura 10 - Viabilidade de aplicativo para adoção.....	22
Figura 11 - Justificativa	23
Figura 12 - Preferencia de idade	23
Figura 13 - Preferencia de adesão	24
Figura 14 - Adoção de "vira-latas"	24
Figura 15 - Instituto Luiza Mell	25
Figura 16 - Adotar.com.....	26
Figura 17 - Site RockBicho.....	27
Figura 18 - Tela inicial do site ou Home	42
Figura 19 - Contato	42
Figura 20 - Tela de Cadastro Institucional.....	43
Figura 21 - Sobre Nós	43
Figura 22 - Animais por filtro de Favoritos.....	44
Figura 23 - Informações do Animal	45
Figura 24 - Tela de Agendamento.....	46
Figura 25 - Sobre Nós	47
Figura 26 - Mapa do Front-end do site	48
Figura 27 – Mapa do back-end do site	49
Figura 28 - Diagrama de caso de uso do app	50
Figura 29 - Diagrama de caso de uso do site.....	50
Figura 30 - Diagrama de Pert.....	58
Figura 31 - Diagrama de Grant.....	59
Figura 32 - Diagrama de Entidade e Relacionamento.....	116
Figura 33 - Ciclo de vida do sistema	117
Figura 34 - Fluxograma do app 1	118
Figura 35 - Fluxograma do app 2	119
Figura 36 - Fluxograma do site.....	120
Figura 37 - Fluxograma do site 2.....	121
Figura 38 - Fluxograma do site 3.....	122

Figura 39 - Fluxograma do site 4.....	123
Figura 40 - Fluxograma do site 5.....	124
Figura 41 - Fluxograma 6	125
Figura 42 - Fluxograma 7	126
Figura 43 - Diagrama de Classes.....	127
Figura 44 - Plano de necócio Canvas	153
Figura 45 - Manual Pagina Sobre Nós	154
Figura 46 - Manual Pagina Download	154
Figura 47 - Manual página de contato.....	155
Figura 48 - Manual Pagina Administrativa.....	156
Figura 49 - Manual Tela de listagem de animais.....	157
Figura 50 - Manual Tela de informação de animais	158
Figura 51 - Tela de agendamento de animais.....	159
Figura 52 - Manual Tela de Favoritos.....	160

LISTA DE TABELAS

Elemento opcional. É a relação das tabelas contidas no trabalho. Estas devem vir em lista própria, de acordo com a ordem apresentada no texto, com cada item designado por seu nome específico, acompanhado do respectivo número da página.

Para atualizar a lista, clicar com o botão direito do mouse sobre o sumário em Atualizar campo e selecionar uma opção disponível de acordo com a necessidade.

Tabela 1 - Desempenho dos alunos na prova de conhecimentos específicos **Erro!**
Indicador não definido.

Tabela 2 - Situação da Educação Brasileira em 2002 – Ensino Médio**Erro!** **Indicador não definido.**

LISTA DE ABREVIATURAS

EAD	Ensino a Distância
SO	Sistemas Operacionais
JDK	Java Development Kit
IDE	Integrated Development (Ambiente de Desenvolvimento Integrado)
SDK	Software Development Kit
APK	Android Application Package
SVN	Apache Subversion
WWW	World Wide Web

LISTA DE SIGLAS

HTML	HyperText Markup Language
CSS	Cascading Style Sheet
SQL	Standart Query Language
MIT	Massachussts Institute of Technology
LXC	Linux Containers
CVS	Sistema de Versões Concorrentes
DVCS	Sistema de controle de Versão Distribuído

LISTA DE ACRÔNIMOS

PHP	HyperText Preprocessor
-----	------------------------

SUMARIO

1. INTRODUÇÃO	14
1.1 Problematização	15
1.2 Motivação.....	15
1.3 Desafio.....	15
1.4 Objetivo.....	16
1.4.1 Objetivo Geral	16
1.4.2 Objetivo Específico.....	16
2. REFERENCIAL TEÓRICO.....	17
2.1 Metodologias de Pesquisa	17
2.1.1 Site de Referência: Instituto Luisa Mell	25
2.1.2. Site de Referência Adotar.com.br	25
2.1.3. Site de Referência: Rockbicho	26
2.2 Tecnologias Utilizadas.....	28
2.2.1 Ferramentas de Banco de Dados.....	28
2.2.2 Ferramentas de Desenvolvimento.....	30
2.2.3 Ferramentas de Design	34
2.2.4 Ferramentas de Diagramação.....	34
2.2.5 Ferramentas de Documentação	35
2.2.6 Ferramentas de Modelagem	37
2.2.7 Ferramentas de Programação.....	38
3. ANÁLISE DO SISTEMA	42
3.1 Design do site.....	42
3.2 Design do app	44
3.3. MAPA DO SITE	48
3.3.1. Mapa do front-end	48
3.3.2. Mapa do back-end.....	49
3.4. Diagrama de caso de uso.....	50
3.4.1. Diagrama do app	50
3.4.2. Digrama do site	50
3.5. Dicionário de dados.....	51
3.6. Cronograma de Atividades	57
3.6.1 Diagrama de Pert	58
3.6.2 Diagrama de Grant.....	59
3.7 Requisitos Funcionais	60
3.7.1 Aplicativo	60
3.7.2 Site	63
3.8 Requisitos Não Funcionais.....	107
3.8.1 Aplicativo	107
3.8.2 Site	109
3.9 Diagrama de Entidade e Relacionamento	116

3.10. Ciclo de vida do sistema.....	117
3.11. Fluxograma	118
3.11.1 Aplicativo.....	118
3.11.2. Site	120
3.12. Diagrama de classes	127
4. DESENVOLVIMENTO	128
4.1.Aplicativo.....	128
4.2. Site	128
5. PLANO DE NEGÓCIO(CANVAS)	143
6. MANUAIS.....	153
6.1. Manual de usuário do site	153
6.1.1. Pagina Sobre Nós	154
6.1.2.Página de download.....	154
6.1.3. Pagina de contato	155
6.1.4.Pagina de Administração.....	155
6.2.Manual de usuário do Aplicativo Mobile.....	157
6.2.1. Tela de listagem de animais disponíveis	157
6.2.2. Tela de exibição das informações referentes aos animais	158
6.2.3.Tela de agendamento de animais	159
6.2.4. Tela de Animais filtrados por favoritamento.....	160
7. HORAS TRABALHADAS	161
7.1. Custos profissionais	161
7.2.Custo das ferramentas.....	161
7.3.Custo total.....	162
8. CONCLUSÃO	163
9. REFERÊNCIAS.....	164

1. INTRODUÇÃO

É notório que com o surgimento da pandemia, muitos cidadãos perderam o seu emprego, o que causou uma crise econômica generalizada por todo o nosso país, e devido a isso, muitas famílias vivenciam uma certa instabilidade econômica, o que por sua vez segundo Exame (2021), ocasionou um aumento de cerca de 60% em abandonos de animais domésticos devido a incapacidade de fornecer as devidas condições de vida para os mesmos.

Entretanto, pode-se observar, simultaneamente, de acordo com Veja Saúde (2021), que tal situação ocasionou um aumento de cerca de 50% na procura de animais domésticos para adoção devido à introdução da prática comum de estudos online (EAD) e trabalhos home office, que já eram executadas posteriormente, porém se intensificaram nesse período de pandemia.

Além disso, o elevado número de animais abandonados é um risco à saúde pública. Isto ocorre devido à possibilidade de esses animais não terem sido vacinados com sua devida regularidade, assim como os mesmos poderem ter os seus dejetos sendo descartados de forma irregular em ambientes urbanos públicos, contribuindo para o aumento da proliferação de doenças.

De acordo com pesquisas, observa-se a constante adesão aos serviços digitais móveis, a qual ocorre em função de sua versatilidade e conveniência. Desse modo, mediante Agência Brasil (2020), se equivale em 79,3% a população dos brasileiros com dez anos ou mais que possuem aparelhos celulares para uso pessoal, o que comprova que tais dispositivos móveis possuem elevado uso na atualidade.

Dessa maneira, a relevância do desenvolvimento desse aplicativo se define pela sua natureza de facilitar o processo de adoção de animais resgatados por abrigos e reduzir a grande quantidade de animais que vivem nas ruas, mitigando efetivamente os impactos desses problemas urbanos. Diante disto, nosso projeto possui como objetivo o desenvolvimento de uma plataforma que tenha a capacidade de atingir a essas perspectivas.

1.1 Problematização

Pode-se observar, a partir dos números estatísticos coletados em meio de nossas pesquisas que o número de animais que foram abandonados em função à pandemia, apresentaram um drástico aumento, ao passo que simultaneamente a procura pelos mesmos também havia aumentado, afetando desse modo a qualidade sanitária do espaço público.

1.2 Motivação

Nosso grupo observou o grande impacto econômico trazido por essa pandemia para o nosso Estado, e a suas mais diversas consequências que trouxe com ela, dentre eles observamos o fato de que muitas pessoas haviam optado por abandonar seus animais de estimação e que os mesmos não se encontravam em condições de vida dignas. Além desse fato, já havíamos observado o problema de questão sanitária envolvendo os dejetos de animais de rua e as doenças que os mesmos podem transmitir para os humanos no ambiente público.

1.3 Desafio

Os desafios que deveremos solucionar ao decorrer do desenvolvimento do nosso projeto se definem como a questão jurídica envolvida no processo de adoção de animais resgatados por abrigos e também a parceria dos mesmos para a utilização de nossa plataforma.

1.4 Objetivo

1.4.1 Objetivo Geral

Facilitar a burocracia envolvida no processo de adoção de animais resgatados por abrigos e instituições próprias.

1.4.2 Objetivo Específico

- Criar uma ferramenta para auxiliar a administração por parte dos abrigos de animais em relação ao processo de adoção dos mesmos;
- Criar uma interface com o intuito de facilitar o processo de adoção de animais por parte do cliente;
- Contribuir para a diminuição de animais abandonados na rua;
- Melhorar a qualidade da saúde pública em suas questões sanitárias relacionadas aos dejetos de animais e possíveis doenças.

2. REFERENCIAL TEÓRICO

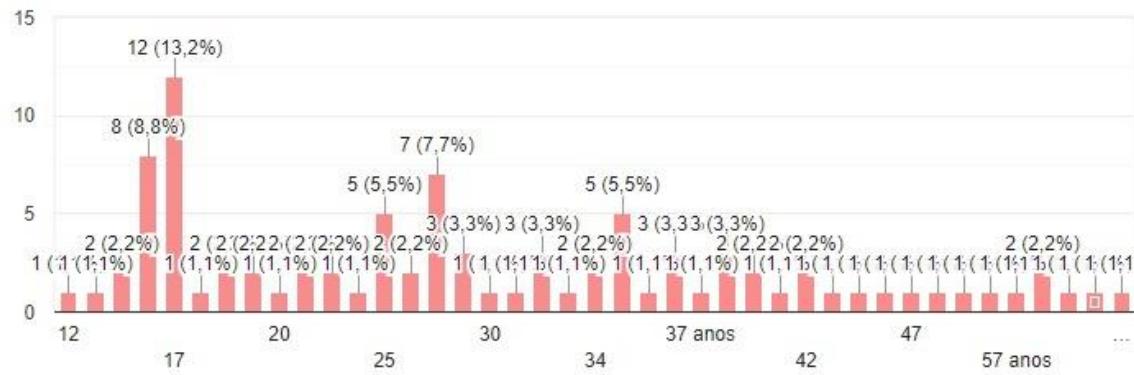
2.1 Metodologias de Pesquisa

Utilizando a plataforma Google Forms, realizamos uma pesquisa de campo entre os dias 13/04/2022 a 23/04/2022, sendo amplamente divulgada por meio da via de comunicação digital Whatsapp e obtendo um número de 91 respostas.

Questão 1: Qual é o seu e-mail?

Figura 1 - Gráfico de idade dos entrevistados

Fonte: Autoria própria.

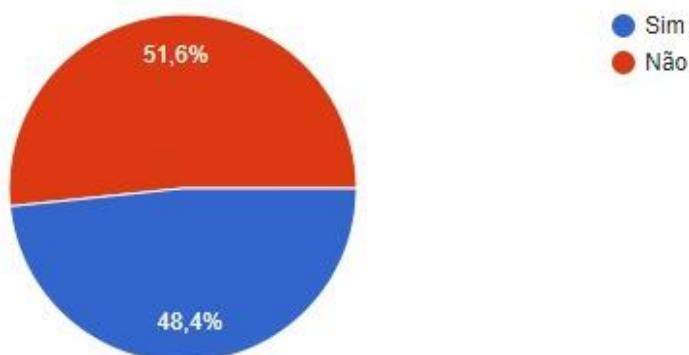


Questão 2: Qual é a sua idade?

Questão 3: Você sofreu os efeitos econômicos da pandemia? (Perda potencial de renda, etc)

Figura 2 - Efeitos econômicos da pandemia

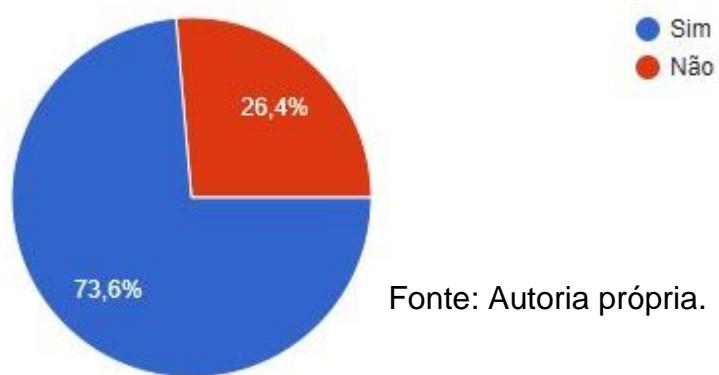
91 respostas



Fonte: Autoria própria.

Figura 3 - Índice de posse de animais

91 respostas



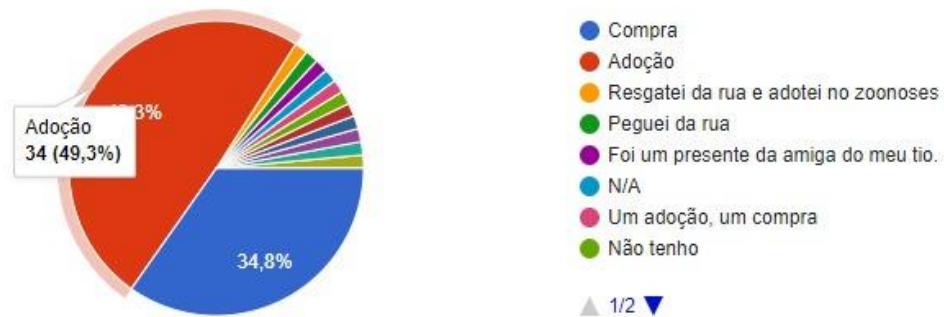
Fonte: Autoria própria.

Questão 4: Você possui um animal de estimação (pet)?

Questão 5: Em caso afirmativo, como você adquiriu seu pet?

Figura 4 - Forma de obtenção do animal

69 respostas



Fonte: Autoria própria.

- Compra
- Adoção
- Resgatei da rua e adotei no zoonoses
- Peguei da rua
- Foi um presente da amiga do meu tio.
- N/A
- Um adoção, um compra
- Não tenho

▲ 1/2 ▼



Fonte: Autoria própria.

▲ 2/2 ▼

Questão 6: Você acredita que um animal pode alegrar sua rotina?

Figura 5 - Lazer em convívio com animal

91 respostas



Fonte: Autoria própria.

Questão 7: Você adotaria um animal que já sofreu maus tratos?

Figura 6 - Adoção de animal mau tratado

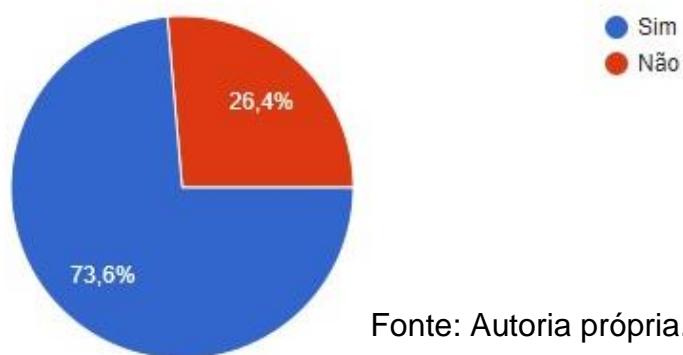
91 respostas



Fonte: Autoria própria.

Figura 7 - Aptidão a cuidar de animais

91 respostas

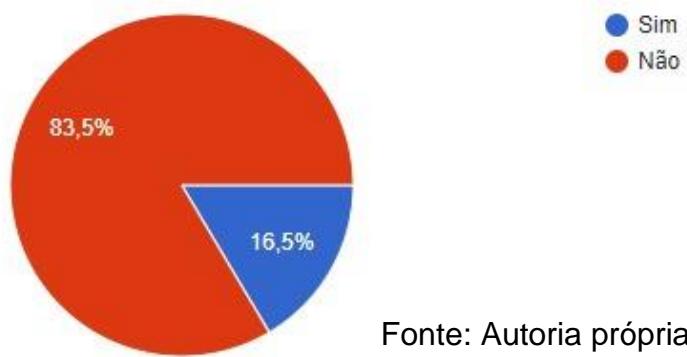


Fonte: Autoria própria.

Questão 8: Você se considera apto a cuidar de um animal de estimação (condição financeira, tempo disponível, etc)?

Figura 8 - Burocracia na adoção

91 respostas



Fonte: Autoria própria.

Questão 9: Você acha os processos burocráticos para adotar um animal muito demorados?

Figura 9 - Ideais de melhoria

Acreditar nas pessoas que se predispõem a cuidar de um animal sem contratos e sem termo de responsabilidade.

Agilidade na avaliação do novo dono

Não sei responder, nunca participei em um processo de adoção

Digitalizaria o processo

Investiria em um meio tecnológico que substituisse de forma digital toda a papelada necessária e agilizasse todos os processos envolvidos na adoção.

aumentar a divulgação sobre o assunto

Sinceramente, desconheço o processo.

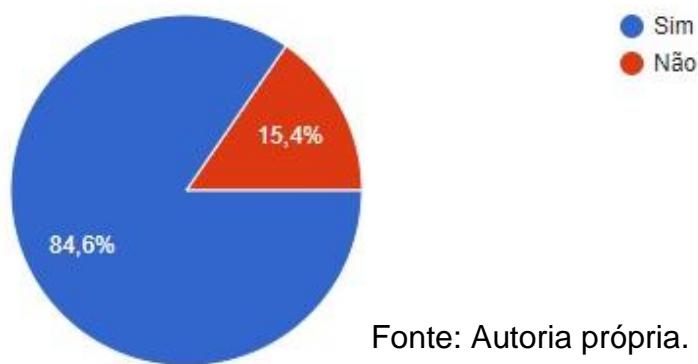
Se a burocracia de checagem de condições de adoção entre outros pudesse ser feita de forma mais rápida, certamente o processo de adoção seria melhor.

Questão 10: Em caso afirmativo, o que você faria para melhorar?

(Resposta dissertativa, alguns exemplos:)

Figura 10 - Viabilidade de aplicativo para adoção

91 respostas



Fonte: Autoria própria.

Questão 11: Você acha que seria viável o auxílio à adoção de um animal por meio de uma aplicação?

Figura 11 - Justificativa

Facilidade

Praticidade

Mais prático

poderia agilizar o processo e talvez teríamos um número menor de animais esperando .

Facilitaria o processo de adoção e ampliaria o seu acesso

Porque o acesso ao celular é muito grande, facilitando a divulgação e o processo de adoção

facilitaria a adoção tanto para os doadores quanto as que adotariam

Para facilitar o acesso e diminuir a burocracia

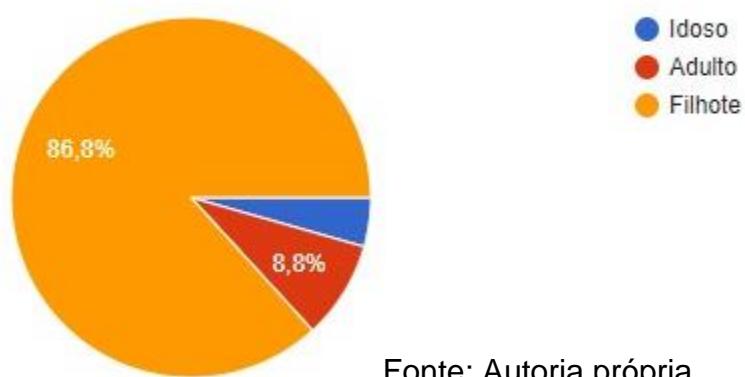
Sim

Questão 12: Porque você acha isso?

(Resposta dissertativa, alguns exemplos:)

Figura 12 - Preferencia de idade

91 respostas

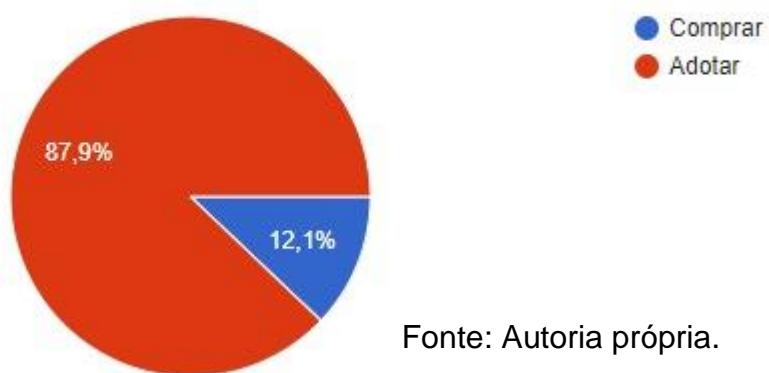


Fonte: Autoria própria.

Questão 13: Você prefere adotar um animal idoso, adulto ou filhote?

Figura 13 - Preferencia de adesão

91 respostas



Fonte: Autoria própria.

Questão 14: Você prefere comprar ou adotar animais de estimação?

Questão 15: Você adotaria um animal vira-lata (mistura entre duas raças)?

Figura 14 - Adoção de "vira-latas"

91 respostas



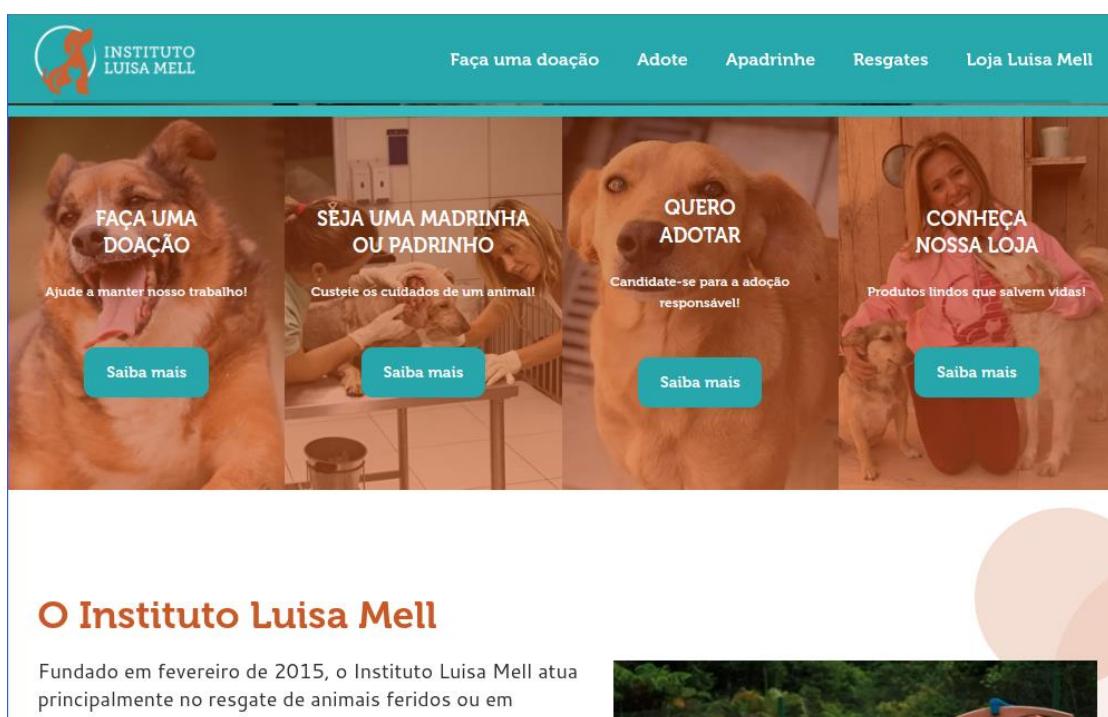
Fonte: Autoria própria.

2.1.1 Site de Referência: Instituto Luisa Mell

O Instituto Luisa Mell possui como principal objetivo a doação de animais que foram resgatados, sendo muitos deles feridos ou em situação de risco, trabalhando em seu cuidado e recuperação. Tal instituição se encontra localizada em Ribeirão Pires, na região metropolitana, e abriga cerca de 300 animais, cachorros e gatos, onde são protegidos e cuidados até terem a chance oportuna de serem adotados.

De acordo com eles, é necessário um investimento financeiro mensal de cerca de 300.000 reais, no qual se é possibilitado por meio de diversas doações e também de parcerias.

Figura 15 - Instituto Luiza Mell



Fonte: Instituto Luisa Mell

2.1.2. Site de Referência Adotar.com.br

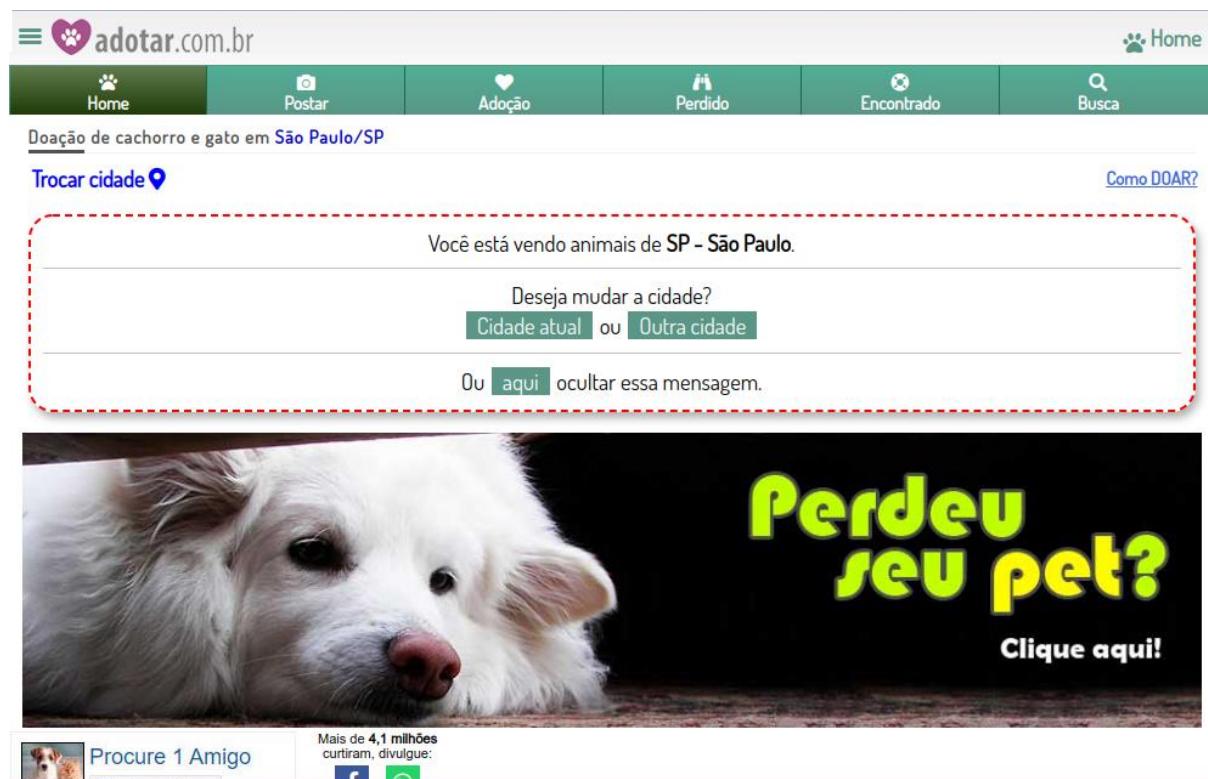
O site adotar.com.br consta com uma extensa área de atendimento, sendo possível a adoção de animais por meio deste site em diversos estados brasileiros. Ele possui

a função de adotar animais, informar sobre animais que estejam perdidos e animais que foram encontrados, além de contar com uma pequena seção de blog, onde são postadas matérias e artigos interessantes relacionados a cães e gatos, e uma seção com eventos relacionados a animais que são recomendados de acordo com a sua localização.

Pode-se destacar também que há uma aba especificando as instituições parceiras, a qual se é indicado o site oficial e uma breve descrição, sendo novamente indicado de acordo com a localização do usuário.

Figura 16 - Adotar.com

Fonte: Adotar.com



2.1.3. Site de Referência: Rockbicho

Rockbicho é uma organização não governamental, sem fins lucrativos, localizado em Belo Horizonte, que possui como objetivo promover a guarda responsável e a defesa dos direitos dos animais, possuindo um programa de adoção

que ajuda a reduzir a reprodução descontrolada nas ruas (castração) e a redução do abandono e sofrimento dos animais. Por meio do site é possível adotar diversos cães e gatos, sendo possível ao usuário realizar um anúncio de doação de animal. Eles possuem um pequeno blog onde são anunciados eventos e novidades, e uma loja com alguns produtos para ajudar no financiamento dos custos envolvidos no processo de cuidados com os animais.

O site conta com a ajuda de diversos órgãos públicos e instituições parceiras para o financiamento de suas atividades e de doações abertas para auditoria, sendo elas registradas mensalmente de modo público para demonstrar a transparência e a veracidade do bom uso delas. Além da ajuda financeira, os indivíduos podem contribuir por meio de um lar temporário, visto que a empresa não possui um abrigo próprio e necessita que pessoas dispostas forneçam abrigo para os animais até eles encontrarem um dono definitivo, sendo fornecidos todos os recursos necessários para tal feito, tais como ração, comedouros, caminha, e medicamentos.

Figura 17 - Site RockBicho

Fonte: Rock bicho



2.2 Tecnologias Utilizadas

2.2.1 Ferramentas de Banco de Dados

PostgreSQL

O PostgreSQL é uma ferramenta que atua como sistema de gerenciamento de bancos de dados relacionados. Seu foco é permitir implementação da linguagem SQL em estruturas, garantindo um trabalho com os padrões desse tipo de ordenação dos dados.

Nos últimos anos, o uso desse sistema tem crescido consideravelmente, muito por conta de sua praticidade e pela sua alta compatibilidade com diferentes padrões de linguagem. Seu funcionamento é desenvolvido para ser, na prática de grande suporte para que qualquer trabalho seja feito sem maiores dificuldades. Muitas empresas construíram produtos e soluções usando o PostgreSQL, algumas delas em destaque são a Apple, a Fujitsu, a Red Hat, a Cisco, a Juniper Network, etc.

Um de seus pontos principais é sua adequação em padrões de conformidade, ajudando a construir bancos de dados otimizados. Nesse trabalho, com suas qualidades principais, o PostgreSQL ajuda a armazenar informações de forma segura e, se necessário, restaurá-las sempre que houver solicitação de outras aplicações integradas.

O PostgreSQL é um sistema que lida bem com altos volumes de solicitações e com cargas de trabalho grandes, ou seja, funciona muito bem para sites com intensidade de acesso. E-commerce famosos, por exemplo, é um ótimo exemplo de estrutura que precisa desse sistema para ter um desempenho otimizado, devido ao alto número de acessos simultâneos recebidos.

O PostgreSQL é muito fácil de manter devido a sua estabilidade. Portanto, se você desenvolver aplicações baseados no PostgreSQL, o custo de desenvolvimento será baixo em comparação com outros sistemas de gerenciamento de banco de dados.

Docker

Docker é uma plataforma open source que facilita a criação e administração de ambientes isolados. Ele possibilita o empacotamento de uma aplicação ou ambiente dentro de um contêiner, se tornando portátil para qualquer outro host que contenha o Docker instalado. É possível criar, implantar, copiar e migrar de um ambiente para outro com maior flexibilidade. A ideia do Docker é subir apenas uma máquina, ao invés de várias. E, nessa única máquina, é possível rodar várias aplicações sem que haja conflitos entre elas.

O Docker é algo parecido com uma máquina virtual extremamente leve, mas não se trata de fato de uma máquina virtual. O Docker utiliza contêineres que possuem uma arquitetura diferente, permitindo maior portabilidade e eficiência. O contêiner exclui a virtualização e muda o processo para o Docker. Então, não podemos dizer que o Docker é uma máquina virtual.

Apesar do Docker ter sido desenvolvido inicialmente com base na tecnologia LXC (Linux Containers – sendo, portanto, mais associado aos contêineres Linux), hoje essa tecnologia tornou-se independente de sistema operacional, havendo a possibilidade de utilizar o Docker em ambientes Linux, Windows e até mesmo MacOS.

Dentre as mais diversas vantagens de se utilizar o Docker, podemos citar a economia de recursos, melhor disponibilidade do sistema (pelo compartilhamento do SO e de outros componentes), possibilidades de compartilhamento, simplicidade de criação e alteração da infraestrutura, manutenção simplificada (reduzindo o esforço e o risco de problemas com as dependências do aplicativo), entre muitas outras.

Laravel

Desenvolvido por Taylor B. Otwell, tendo sua primeira versão beta lançada em meados de junho de 2011, o Laravel é um Framework Open Source sob a licença MIT, criado com o propósito de ser uma alternativa mais avançada do CodeIgniter. Atualmente, se encontra na versão 5.8, tendo seu código-fonte hospedado no GitHub.

O Laravel possui um sistema de template (Blade) que facilita a criação da camada de visualização de dados (Páginas HTML). Com ele, podemos facilmente criar páginas simples e intuitivas de forma rápida e eficaz. Dentre alguns dos recursos do Blade, se destacam: Herança de layouts, sistema de tags, seções e uso de código PHP nos templates.

O Laravel possui, por padrão, um módulo de autenticação/autorização completo que provê todos os recursos para sua implementação, como: Autenticação de usuários, autorização de usuários, recuperação de senhas, logout, controle de sessão e cadastro de usuários.

Com o Laravel não precisamos criar códigos SQL para manipular ou criar tabelas no Banco de Dados. Todo processo é feito utilizando código PHP que, posteriormente, será convertido em instruções SQL. Implementa o padrão Active Record, onde cada model da aplicação representa uma tabela no banco de dados.

Dentre as vantagens de utilizar o Laravel, podemos citar que ele é simples e relativamente fácil de se usar, possui uma documentação completa, é amplamente utilizado, possui uma comunidade ativa e é, acima de tudo, gratuito.

2.2.2 Ferramentas de Desenvolvimento

Android Studio

O Android Studio é uma IDE (Ambiente de desenvolvimento integrado), criado pela Google e anunciado do Google I/O 2013 para auxiliar no desenvolvimento de aplicativos para Android. Ele pode ser instalado nos sistemas operacionais Windows, OSX e Linux e é recomendado pelo próprio Google que o hardware possua, no mínimo, 4 GB de memória e 1GB de espaço livre em disco. É necessário ter o Java instalado na máquina através do JDK (Java Development Kit) e não a JRE, como normalmente é instalado, pois para desenvolver em Android é necessário que todas as classes de desenvolvimento do Java estejam presentes na máquina.

A IDE é baseada em IntelliJ IDEA; um ambiente de desenvolvimento para programas que possui poderosas ferramentas de edição de código. Em termos de sua análise de código, ele destaca os erros imediatamente, a fim de dar uma solução mais rápida. Como ferramentas integradas para desenvolver ou construir programas no Android, contém uma interface de usuário previamente construída ou projetada, com vários modelos de tela, onde os elementos existentes podem ser movidos. Além disso, são incluídos depuradores para emuladores e a possibilidade de trabalhar com Logcat.

O Android Studio possui diferentes componentes que auxiliam na tarefa de construção de aplicativos; Sistema de compilação baseado em Gradle, compilação de variantes e vários arquivos APK, bem como modelos de código que auxiliam na construção de aplicativos. Um editor de layout completo com suporte para edição de arrastar e soltar de elementos de tema. Facilidade de uso e compatibilidade de versão, Code encolhe com ProGuard e cada vez menos consumo de recursos com Gradle. Por fim, suporte integrado para Google Cloud Platform, que facilita a integração de mensagens na nuvem do Google e o App Engine.

Visual Studio Code

Em 2015 foi lançado pela Microsoft um editor de código destinado ao desenvolvimento de aplicações web chamado de Visual Studio Code, ou simplesmente VSCode. Anunciada durante o Build, evento voltado a

desenvolvedores que ocorre nos Estados Unidos anualmente, trata-se de uma ferramenta leve e multiplataforma que está disponível tanto para Windows, quanto para Mac OS e Linux e atende a uma gama enorme de projetos, não apenas ASP.NET, como também Node.js. Adicionalmente, o editor possui suporte à sintaxe de diversas linguagens como Python, Ruby, C++.

Além de ser totalmente gratuito, ainda no segundo semestre do ano do lançamento, durante o evento Connect(), o editor foi anunciado como open source, tendo código disponibilizado no GitHub, o que permite à comunidade técnica contribuir com seu desenvolvimento e facilitando a criação de extensões e novas funcionalidades.

Dentre as vantagens de utilizar o Visual Studio Code podemos citar que ele é uma ferramenta simples de se utilizar, é open source, está disponível para os principais sistemas operacionais do mercado (Windows, Linux e Mac), possui uma arquitetura bem planejada, grande possibilidade de customização e diversas funcionalidades e extensões disponíveis.

Bootstrap

Bootstrap é um framework front-end que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Além disso, pode lidar com sites de desktop e páginas de dispositivos móveis da mesma forma.

Originalmente, o Bootstrap foi desenvolvido para o Twitter por um grupo de desenvolvedores liderados por Mark Otto e Jacob Thornton Logo e se tornou uma das estruturas de front-end e projetos de código aberto mais populares do mundo.

Antes de ser uma estrutura de código-fonte aberto, o Bootstrap era conhecido como Twitter Blueprint. Após alguns meses de desenvolvimento, o Twitter realizou sua primeira Hack Week: o projeto ganhou uma grande popularidade quando desenvolvedores de todos os níveis de habilidade usaram o framework sem

qualquer orientação externa. Após o evento, ele serviu como guia de estilo para o desenvolvimento de ferramentas internas na empresa por mais de um ano antes de seu lançamento se tornar público.

Hoje estima-se que cerca de 7 milhões de sites utilizem o Bootstrap como framework front-end. Entre suas vantagens, podemos citar a documentação farta e a comunidade muito ativa, a infinidade de componentes que podem ser facilmente chamados em suas aplicações, além da boa base de padrões estéticos, que permitem criar páginas belas e funcionais.

GitHub

GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Foi criado em 2007 e lançado em abril de 2008 por Tom Preston-Werner, Chris Wanstrath, P. J. Hyett e Scott Chacon e adquirido pela Microsoft em 2018. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.

O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux.

Tendo uma arquitetura distribuída, o Git é um exemplo de DVCS (Sistema de Controle de Versão Distribuído). Em vez de ter apenas um único local para o histórico completo da versão do software, como é comum em sistemas de controle de versão outrora populares como CVS ou Subversion (também conhecido como SVN), no Git, a cópia de trabalho de todo desenvolvedor do código também é um repositório que pode conter o histórico completo de todas as alterações. Além de ser distribuído, o Git foi projetado com desempenho, segurança e flexibilidade em mente.

2.2.3 Ferramentas de Design

Photoshop

Photoshop foi inventado em 1987 por dois irmãos, Thomas e John Knoll, que venderam a licença de distribuição para a Adobe em 1988. O produto era originalmente chamado Display.

Photoshop é um pilar para designers, desenvolvedores da web, fotógrafos, artistas gráficos e muitos outros profissionais criativos, bem como amadores. O software é usado para editar, criar e retocar imagens, bem como adicionar efeitos especiais. Os gráficos podem ser criados e exportados para outros programas.

O programa está disponível para computadores Windows e Mac pelo preço de R\$90,00 mensais, ou R\$43,00 mensais pelo pacote “Fotografia”, que inclui também os programas Lightroom e Lightroom Classic. É considerado um editor de gráficos raster, o que significa que os usuários podem criar e editar imagens e salvá-las em um dos vários formatos.

O Photoshop emprega um sistema de edição baseado em camadas que permite criar e alterar imagens com muitas sobreposições. As camadas podem ser usadas para criar sombras e outros efeitos e podem atuar como filtros que afetam as cores subjacentes. Ele possui também muitos recursos de automação e atalhos de teclado que ajudam a economizar tempo em tarefas repetitivas. Instale filtros e plug-ins, novos pincéis e texturas e outros extras úteis no Photoshop para aumentar continuamente sua funcionalidade.

2.2.4 Ferramentas de Diagramação

Anaconda

Anaconda é uma plataforma de ciência de dados para Python. Possibilita a instalação de diferentes versões da linguagem Python e a criação de ambientes de desenvolvimento específicos para diferentes necessidades. O Anaconda foi criado pela Continuum Analytics e vem configurado com uma pré-instalação das principais bibliotecas Python para ciências de dados.

A sua popularidade pode ser explicada porque ele traz as principais ferramentas usadas para ciências e análise de dados, aprendizado de máquina e inteligência artificial com apenas uma instalação. Assim, com uma configuração curta simples é possível já começar a trabalhar.

Além das bibliotecas, o Anaconda vem com o Conda, uma ferramenta para gerenciamento de projetos e ambientes operada exclusivamente a partir da linha de comando. A maioria dos projetos, pacotes e ambientes de ciências de dados do mundo real são gerenciados e ambientados no Conda. Você também pode usar o Anaconda para criar ambientes para isolar projetos construídos com versões diferentes do Python ou versões diferentes de pacotes.

Por já vir com os pacotes pré-instalados, o Anaconda é um download bastante grande, com cerca de 500 MB. Se você não quer ocupar todo este espaço no seu disco rígido, é possível baixar o Miniconda, uma distribuição menor que inclui apenas conda e Python, e instalar qualquer um dos pacotes disponíveis com conda.

2.2.5 Ferramentas de Documentação

Microsoft Word

O Word é um processador de textos desenvolvido pela empresa norte-americana Microsoft. Um software de processamento de textos é aquele que permite ao usuário criar, editar e modificar documentos que incluem texto formatado e outros objetos gráficos. O Word é uma ferramenta com mais de 30 anos de história. Foi lançada para MS-DOS em 1983. No momento roda nas últimas versões do Windows

e do macOS. Também tem aplicativos móveis, feitos para smartphones e tablets, com Android ou iOS como sistema operacional.

A principal função do Word é processar texto. Isso o diferencia dos editores de texto, como o bloco de notas do Windows ou Notepad++. Ao contrário desses dois últimos, o Word permite adicionar imagens, criar gráficos, incluir tabelas, modificar a fonte ou escrever textos em negrito, itálico ou sublinhados. Pensando no usuário, todas as modificações são feitas de forma visual. Os documentos podem ser exportados em muitos formatos, como PDF ou HTML. Porém, o mais comum é armazenar os projetos em arquivos DOCX. Esses últimos são compatíveis com uma infinidade de aplicativos e dispositivos.

Além disso, esta ferramenta possui diversas funcionalidades que visam melhorar a experiência do usuário. Por exemplo, permite salvar os arquivos criados na nuvem, graças à compatibilidade com OneDrive. Também é perfeita para compartilhar documentos com outros usuários. Em suas últimas atualizações foram incluídas funções como salvamento automático ou a edição simultânea entre várias pessoas do mesmo documento.

Microsoft PowerPoint

PowerPoint é de um programa dedicado à criação de apresentações, que podem ser usadas para os mais variados fins: reuniões, palestras, negociações comerciais, apresentações corporativas, etc. Essas apresentações são divididas em slides, que também podem ser chamados de telas. Ele apresenta ao usuário diversas ferramentas de edição de texto, inserção de imagens, vídeos, músicas e animações.

O PowerPoint foi criado por Robert Gaskins e Dennis Austin em uma empresa de software chamada Forethought, Inc. Lançado em 20 de abril de 1987, inicialmente ele rodava apenas em computadores da Apple. Três meses mais tarde, a Microsoft adquiriu os direitos do programa por US\$14 milhões.

Esta foi a primeira aquisição significativa da Microsoft, que montou uma nova unidade de negócios somente para o PowerPoint no Vale do Silício, onde a Forethought estava localizada. A partir de sua segunda versão, o programa se tornou oficialmente um componente do Pacote Microsoft Office e começou a se popularizar.

Uma pesquisa de 2003 apontou que o Powerpoint detinha quase 95% do mercado de apresentações e mostrou que, apesar da resistência e do preconceito que enfrenta, o software ainda tem muita relevância.

2.2.6 Ferramentas de Modelagem

ASTAH COMMUNITY

Astah foi desenvolvido no Japão na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui JVM (Máquina Virtual Java). JUDE (Ambiente para Desenvolvedores UML e Java) obteve o prêmio “Produto de Software do Ano 2006”, pela Agência de Promoção de Informação Tecnológica no Japão. Anteriormente conhecido como JUDE, ele funciona nas plataformas Windows, Mac e Linux.

Podemos citar as empresas famosas que usam Astah como Amazon, Google e Oracle. Pessoas que possuem diversas funções como: Engenheiros de Software, Engenheiros de Sistemas, Arquitetos, Desenvolvedores, Gerentes de Projeto e Analistas de Negócios. Estudantes, Professores, Empresas ou qualquer Indivíduo pode usar o Astah. Ele é fácil para um iniciante, mas robusto o suficiente para projetar sistemas complexos.

Na área de Engenharia de Software, a UML (Linguagem de Modelagem Unificada) é uma linguagem de modelagem que permite representar um sistema de forma padronizada. Astah é utilizado nos diagramas dinâmicos, essa ferramenta já é bastante consolidada, voltada para a modelagem de sistemas utilizando a UML,

utiliza como recurso adicional a modelagem MAS ML (Modelagem de um Sistema Multiagente).

2.2.7 Ferramentas de Programação

HTML

O HTML é uma linguagem de marcação. Estas linguagens são constituídas de códigos que delimitam conteúdo específicos, segundo uma sintaxe própria. O HTML tem códigos para criar páginas na web. Estes códigos que definem o tipo de letra, qual o tamanho, cor, espaçamento, e vários outros aspectos do site.

HTML foi criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB.

Para criar e editar códigos em HTML é necessário qualquer editor de texto comum, como bloco de notas. Para testar os códigos, basta salvar o arquivo em formato .HTML e executar. Para o teste é necessário ter um navegador configurado como padrão. Não é necessária internet, pois o arquivo com os códigos está na máquina onde está sendo executado.

CSS

O CSS é uma linguagem que determina a aparência (layout) de páginas para a Web. Este programa permite ao usuário criar páginas da Web com códigos mais fáceis de elaborar que os códigos HTML. Estes códigos permitem fazer as aplicações com facilidade.

O CSS foi proposto pela primeira vez em outubro de 1994, por Hakon Lie, que queria facilitar a programação de sites, que na época era muito mais complexa. As pessoas tinham que utilizar mais códigos para chegar a um resultado simples, como criar uma tabela.

Em 1995 o CSS1 foi desenvolvido pela W3C, um grupo de empresas do ramo da informática. A linguagem de estilos ganhou muito destaque entre 1997 e 1999, neste período ficou conhecido por grande parte dos programadores.

Este programa é utilizado pelos programadores em todo o mundo. O CSS controla as opções de margem, linhas, cores, alturas, larguras, imagens e posicionamento, sem necessidade de programar em HTML. O CSS tem também alguns códigos prontos, permitindo aos usuários pouparem tempo criando códigos muito comuns.

JAVASCRIPT

JavaScript, é uma das linguagens de programação mais populares e usadas no mundo. Ela é uma linguagem interpretada, de alto nível e multi-paradigma (orientado a objeto, funcional, imperativo e, protótipos). Com ela, é possível desenvolver desde páginas dinâmicas, aplicativos para smartphones, sistemas complexos e até jogos eletrônicos.

O JavaScript foi criado por Brendan Eich em 1995 durante seu tempo na Netscape Communications (para os geração Z, Netscape foi um dos primeiros navegadores), que, além de criar o JavaScript, foi também um dos fundadores da Mozilla Corporation. Inicialmente chamada de Mocha, suas primeiras versões eram de uso exclusivo da Netscape e seu desenvolvimento foi inspirado nas linguagens Java, Scheme e Self. No decorrer da história, a Netscape acabou fazendo uma parceria com a Sun, desenvolvedora do Java, que queria usar a tecnologia da Netscape para fortalecer a sua recém-criada linguagem (Java), começaram então a veicular a linguagem de script JavaScript como uma companheira do Java, sem notar a possível concorrência que as duas acabariam por ter.

JAVA

Java é uma linguagem de programação orientada a objetos que começou a ser criada em 1991, na Sun Microsystems. Teve inicio com o Green Project, no qual os mentores foram Patrick Naughton, Mike Sheridan, e James Gosling. Este projeto não tinha intenção de criar uma linguagem de programação, mais sim de antecipar a “próxima onda” que aconteceria na área da informática e programação. Os idealizadores do projeto acreditavam que em pouco tempo os aparelhos domésticos e os computadores teriam uma ligação.

Nesta época, a internet estava ficando cada vez mais popular, e a equipe do Green Project começou a pensar em aplicações para o Oak na internet, onde a palavra-chave é interação. Eles conseguiram adaptar a linguagem Oak para a internet, e em 1995 foi lançado o Java, que era uma versão atualizada do Oak para a internet.

A tecnologia Java teve uma enorme utilização, e logo grandes empresas como a IBM, anunciaram que estariam dando suporte ao Java, ou seja, os produtos destas empresas iriam rodar aplicativos feitos em Java. Estimativas apontam que a tecnologia Java foi a mais rapidamente incorporada na história da informática. Em 2003 o Java já tinha mais de 4 milhões de desenvolvedores. A ideia inicial do Green Project começou a se concretizar. A linguagem deles passou a ser utilizada em dezenas de produtos diferentes. Computadores, celulares, palmtops, e a maioria dos produtos da Apple.

Em 2006 muitas partes do Java estavam sendo passadas para a licença de Software libre, e a maioria já estava disponível para o público gratuitamente, tudo Sob licença GNU. O Java foi uma revolução na interatividade, sua utilização aumenta a cada dia. Java é uma linguagem relativamente simples e dinâmica, permite criar programas e aplicações para a Web sem depender de outra linguagem.

PYTHON

Python é uma linguagem de programação interpretada cuja filosofia enfatiza uma sintaxe favorecendo um código mais legível, além de ser “free”.

Python foi criado no ano de 1989 por Guido van Rossum no Centro de Matemática e Tecnológica da Informação (CWI, Centrum Wiskunde e Informatica), na Holanda, como sucessor da linguagem de programação ABC, capaz de lidar com exceções e interagir com o sistema operacional Amoeba.

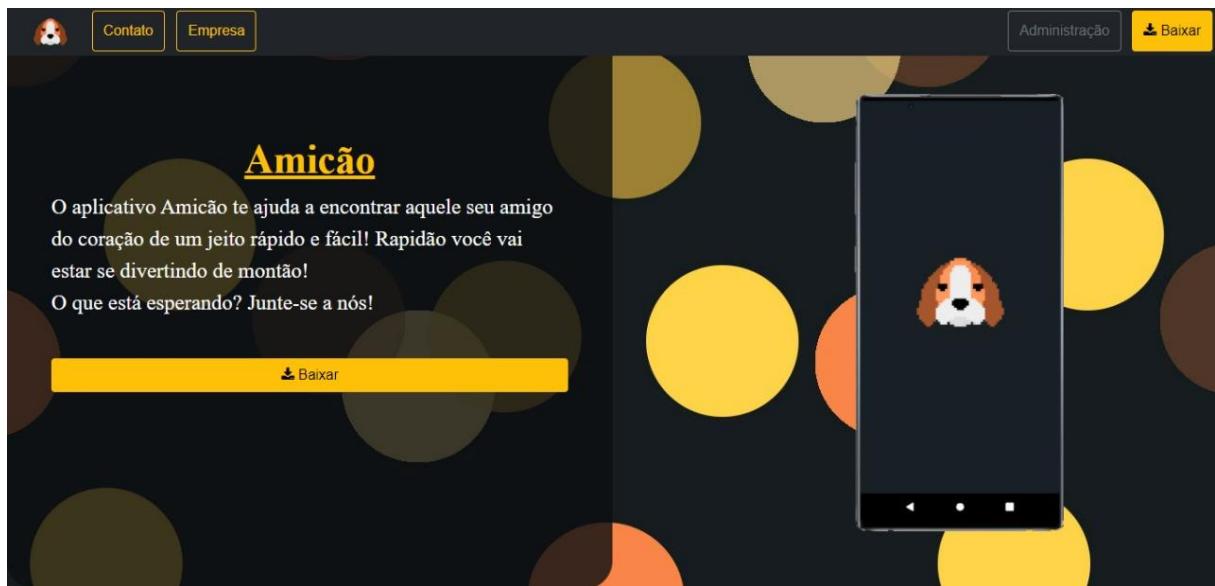
O nome da língua vem do gosto de seu criador pelos humoristas britânicos Monty Python. Van Rossum é o principal autor de Python, e seu papel central contínuo na decisão da direção de Python é reconhecido, referindo-se a ele como Ditador de Vida Benevolente.

Atualmente, é gerenciada pela Python Software Foundation. Possui uma licença de código aberto, chamada Python Software Foundation License, que é compatível com a GNU General Public License a partir da versão 2.1.1 e incompatível em certas versões anteriores.

3. ANÁLISE DO SISTEMA

3.1 Design do site

Figura 18 - Tela inicial do site ou Home



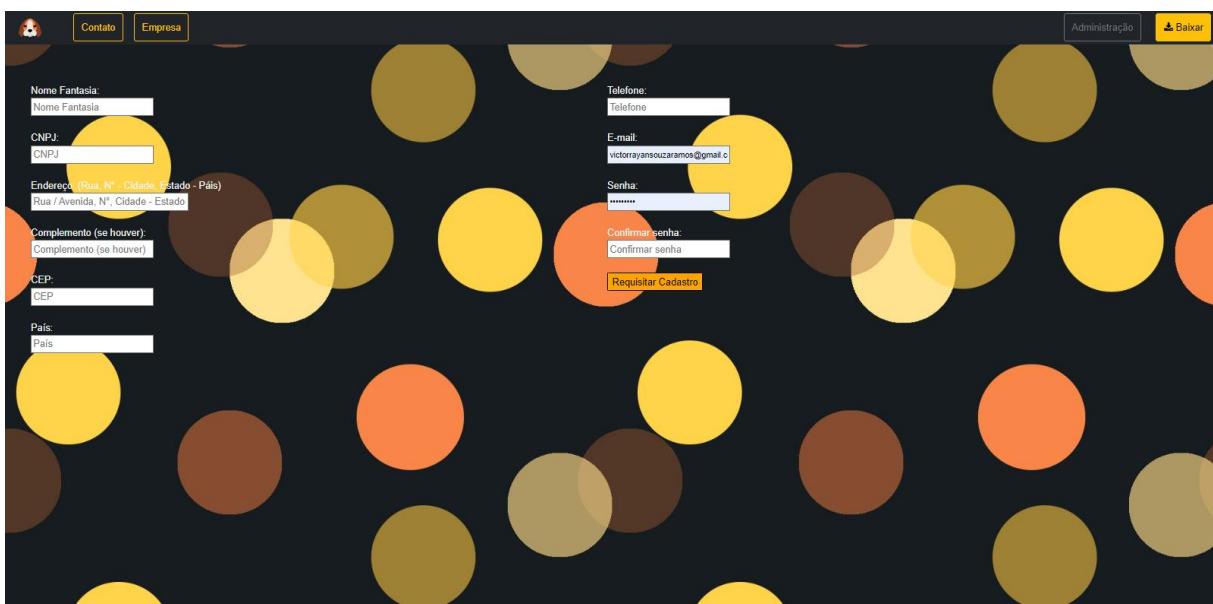
Fonte: Autoria própria.

Figura 19 - Contato

Nome completo:	<input type="text" value="Seu nome e sobrenomes"/>
Email para contato:	<input type="text" value="email@domínio.com"/>
Assunto:	<input type="text" value="Assunto"/>
Digite sua mensagem:	<input type="text" value="Mensagem..."/>

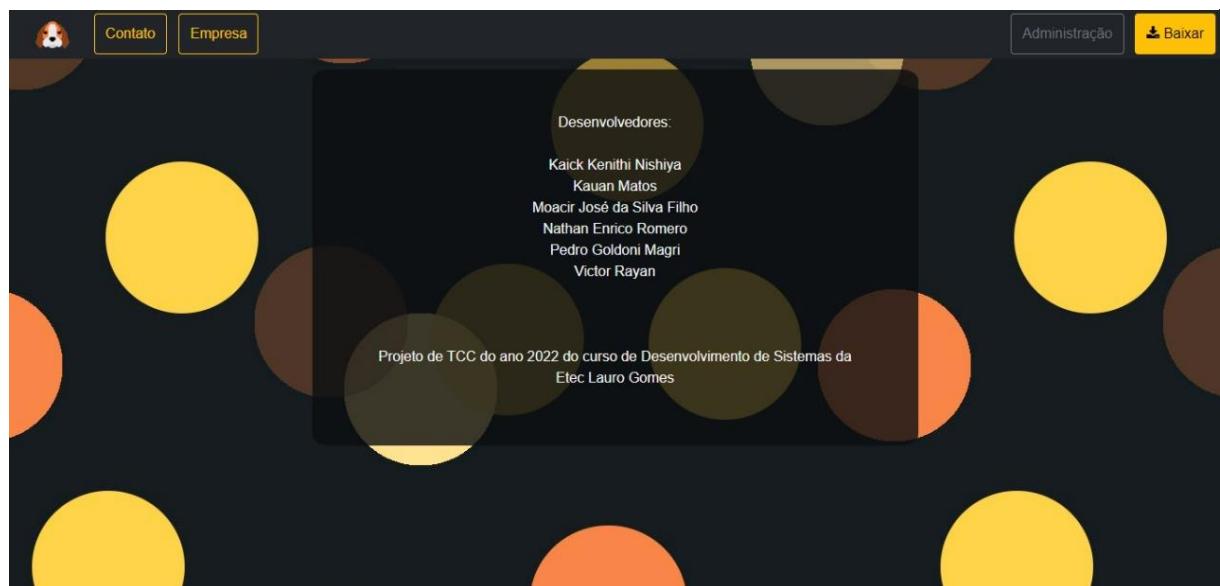
Fonte: Autoria própria.

Figura 20 - Tela de Cadastro Institucional



Fonte: Autoria própria.

Figura 21 - Sobre Nós



Fonte: Autoria própria.

Figura 22 - Animais por filtro de Favoritos



Fonte: Autoria própria.

Figura 23 - Informações do Animal



Fonte: Autoria própria.

Figura 24 - Tela de Agendamento

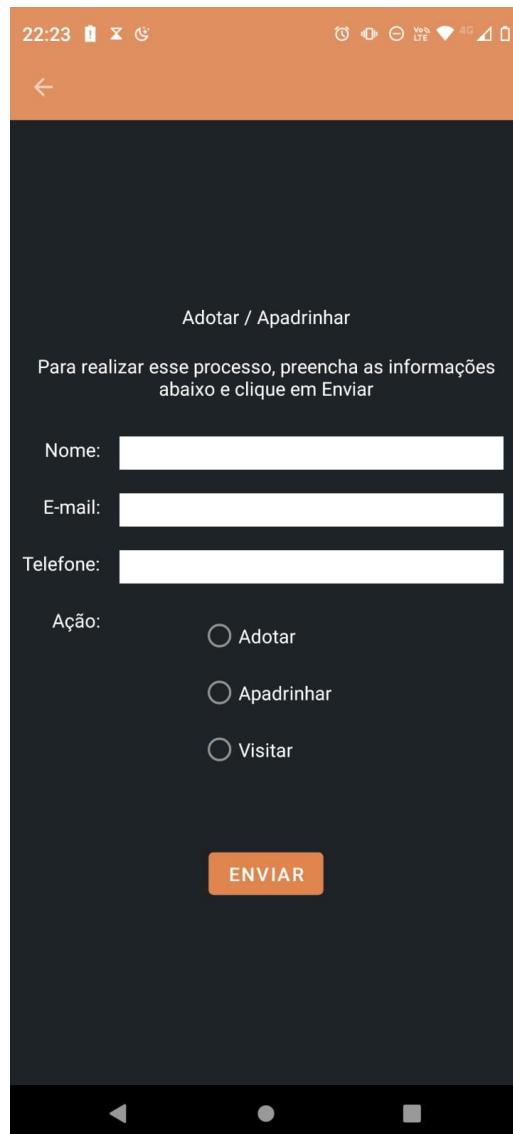
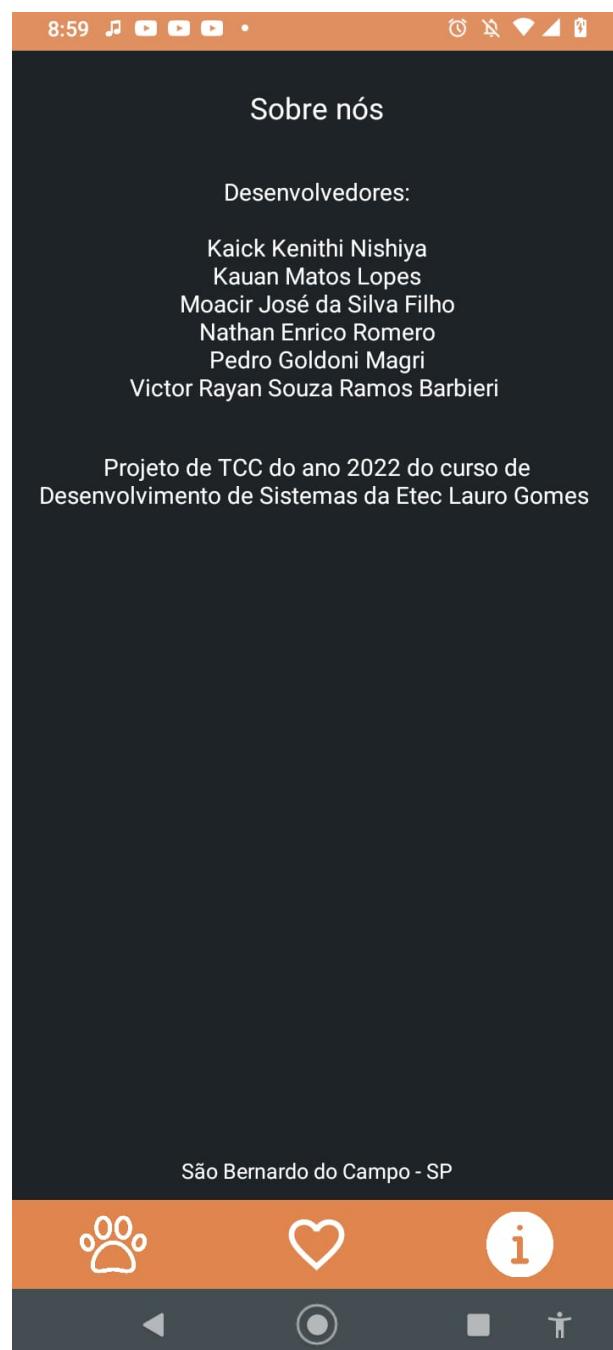


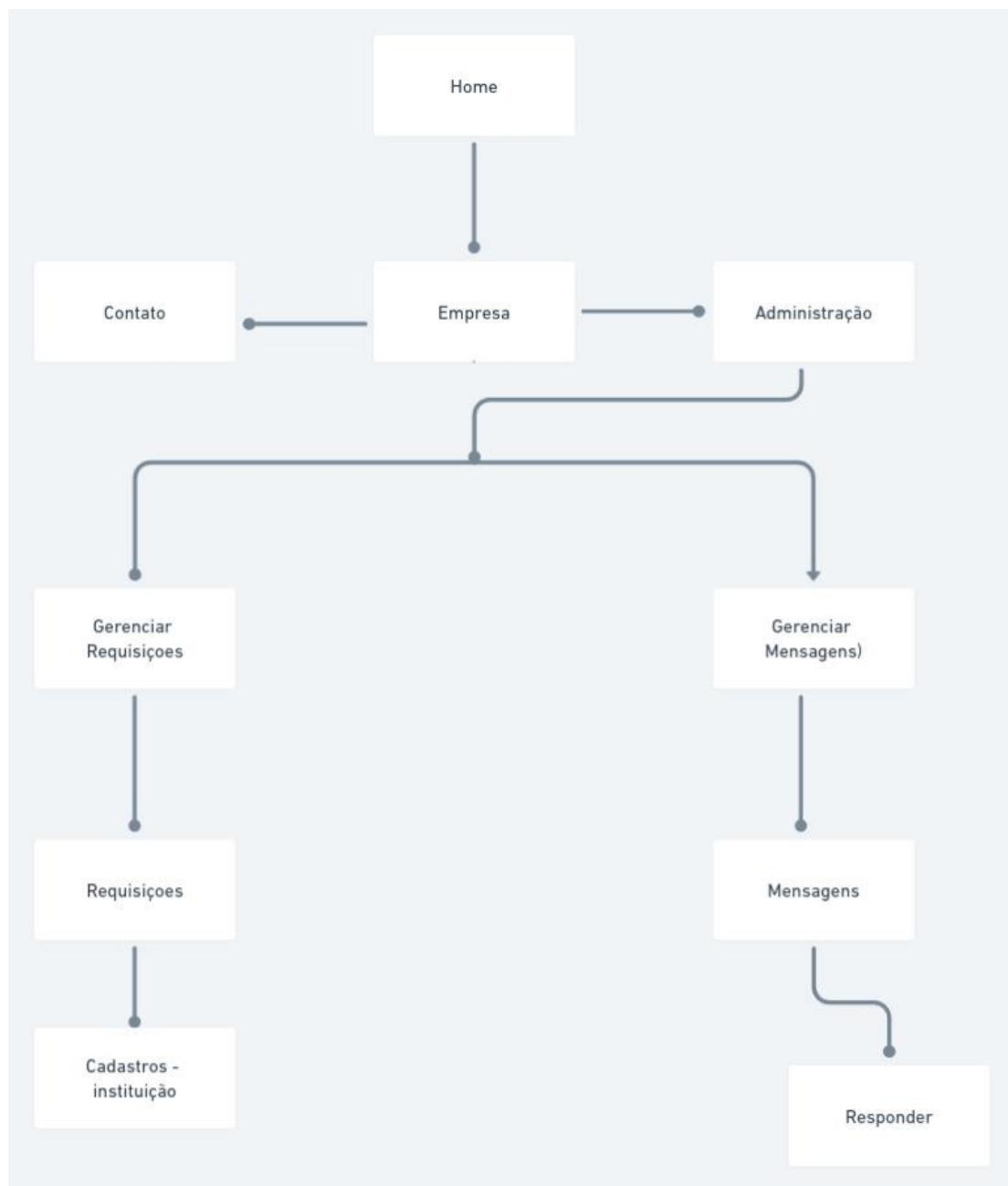
Figura 25 - Sobre Nós



Fonte: Autoria própria.

3.3. MAPA DO SITE

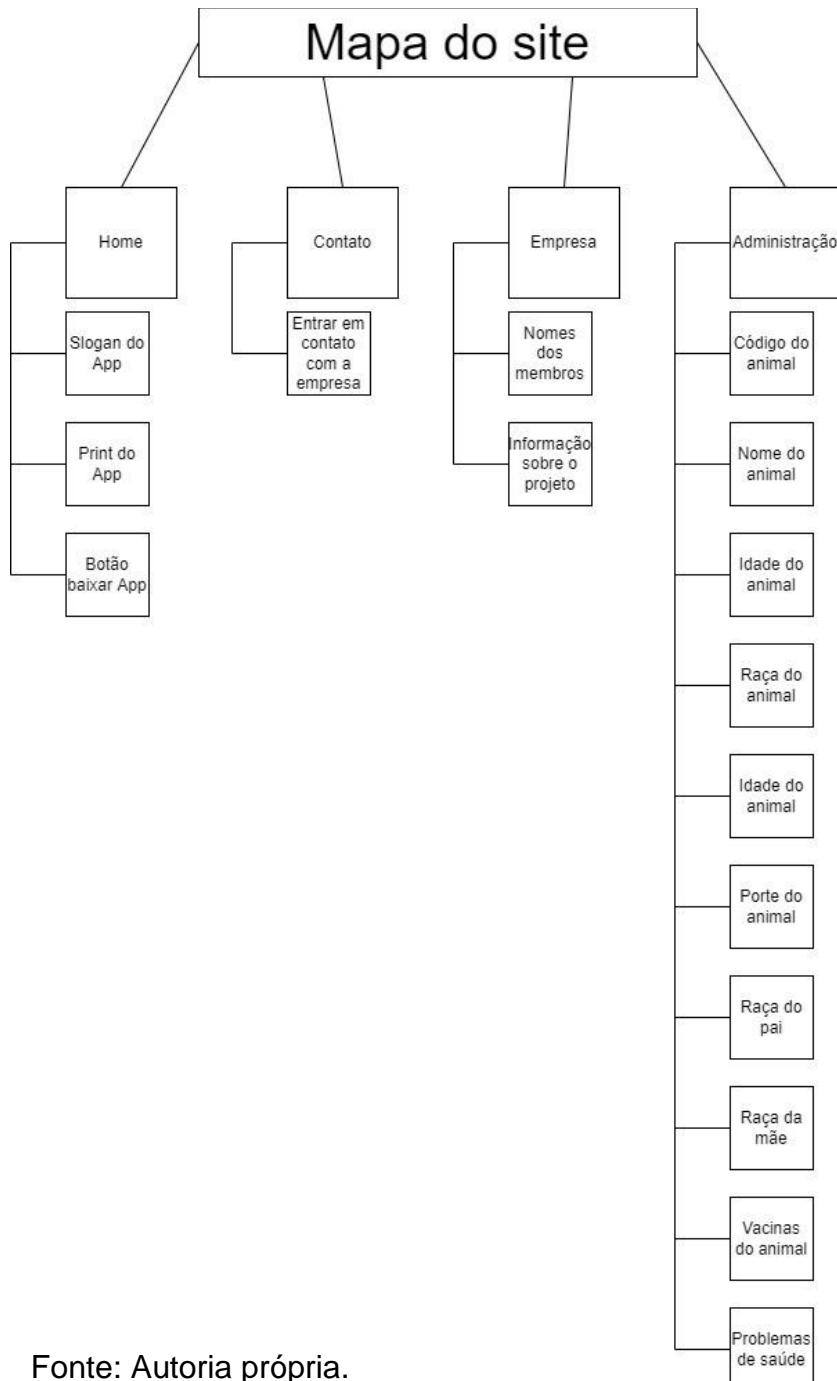
Figura 26 - Mapa do Front-end do site



Fonte: Autoria própria.

3.3.2. Mapa do back-end

Figura 27 – Mapa do back-end do site

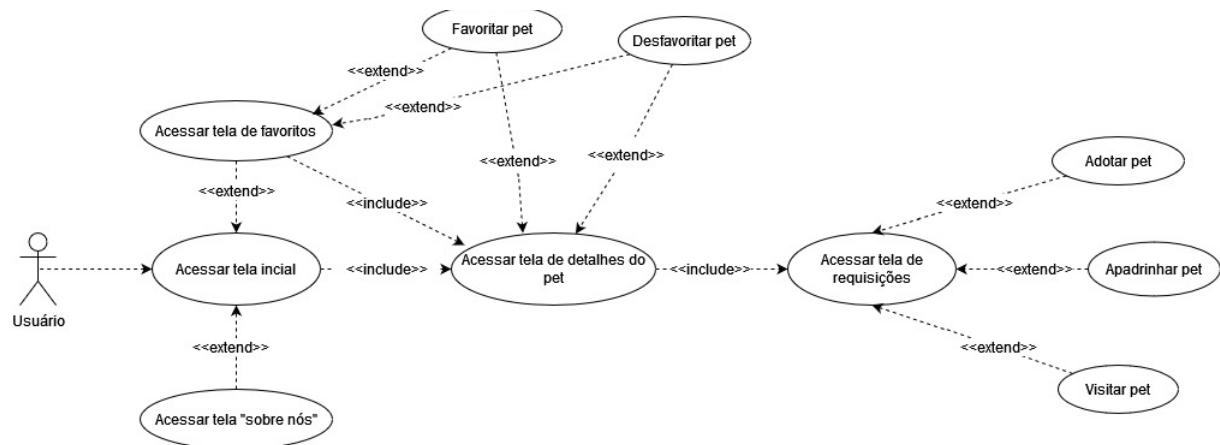


Fonte: Autoria própria.

3.4. Diagrama de caso de uso

3.4.1. Diagrama do app

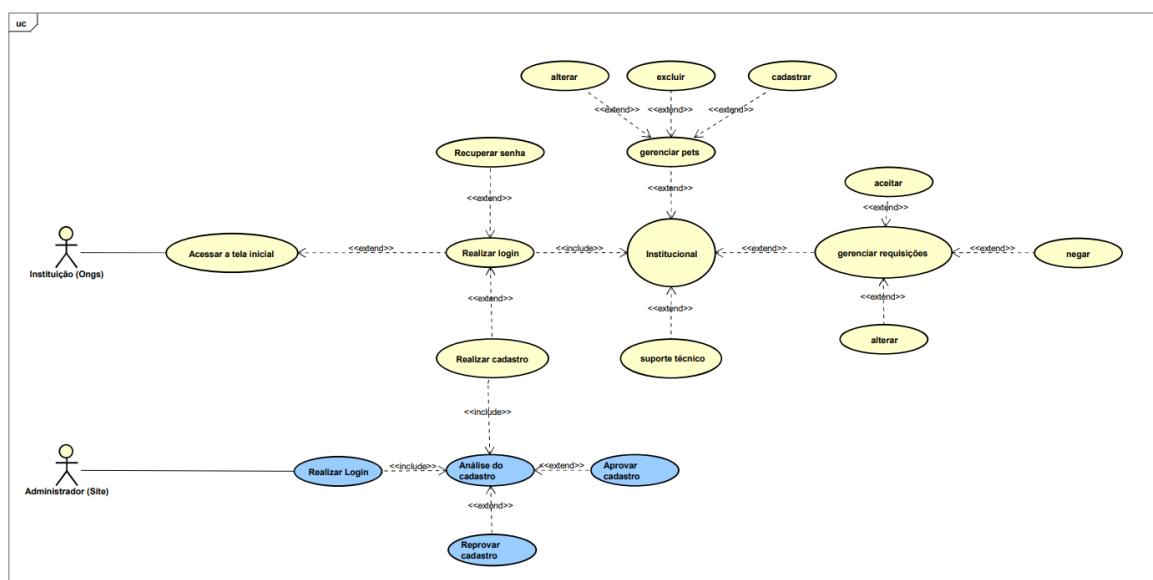
Figura 28 - Diagrama de caso de uso do app



Fonte: Autoria própria.

3.4.2. Digrama do site

Figura 29 - Diagrama de caso de uso do site



Fonte: Autoria própria.

3.5. Dicionário de dados

tb_pets							
---------	--	--	--	--	--	--	--

Campo	Tipo	Tamanho	Obrigatorio	Unico	Chave	Valor (Padrao)	Regras de Validação
Id	varchar	256	Sim	Sim	PK	-	-
nome	Varchar	100	Sim	Não	-	-	-
raca	Varchar	20	Não	Não	-	-	-
raca_pai	Varchar	20	Não	Não	-	-	-
raca_ma_e	Varchar	20	Não	Não	-	-	-
Nascimento	Datetim	23	Não	Não	-	-	-
idade	int	2	Sim	Não	-	-	-
status	Varchar	20	Sim	Não	-	-	-
comportamento	Varchar	500	Não	Não	-	-	-
genero	Varchar	9	Sim	Não	-	-	-
img_path	Varchar	520	Sim	Não	-	-	-
porte	Varchar	8	Sim	Não	-	-	-
vacinas_	Varchar	3	Sim	Não	-	-	-

essencia is							
saudade	Varchar	500	Não	Não	-	-	-

tb_reqs							
---------	--	--	--	--	--	--	--

Campo	Tipo	Tamanho	Obrigatório	Único	Chave	Valor (Padrão)	Regras de Validação
Id	varchar	256	Sim	Sim	PK	-	-
id_pet	varchar	256	Sim	Sim	FK	-	-
nome	varchar	100	Sim	Nao	-	-	-
doc_nun	int	20	Sim	Sim	-	-	-
phone	varchar	20	Sim	Sim	-	-	-
email	varchar	100	Sim	Sim	-	-	-
Endereço	varchar	100	Sim	Sim	-	-	-
Cep	varchar	20	Sim	Não	-	-	-
regiao	varchar	50	Sim	Não	-	-	-
Obs	varchar	500	Sim	Não	-	-	-
status	varchar	20	Sim	Não	-	-	-
req_type	varchar	20	Sim	Não	-	-	-

tb_org

Campo	Tipo	Tamanh o	Obrigato rio	Unico	Chave	Valor (Padrao)	Regras de Validaçã o
Id	varchar	256	Sim	Sim	-	-	-
cnpj	varchar	20	Sim	Sim	-	-	-
Cep	integer	20	Sim	Sim	-	-	-
Enderec o	varchar	100	Sim	Nao	-	-	-
phone	varchar	20	Sim	Não	-	-	-
email	varchar	100	Sim	Não	-	-	-
regiao	varchar	50	Sim	Nao	-	-	-

tb_auth_org

Campo	Tipo	Tamanho	Obrigatório	Único	Chave	Valor (Padrão)	Regras de Validação
Id	varchar	256	Sim	Sim	PK	-	-
id_org	varchar	256	Sim	Não	FK	-	-
email	varchar	100	Sim	Sim	-	-	-
Username	varchar	100	Sim	Não	-	-	-
Password	varchar	100	Sim	Não	-	-	-
rememberToken	varchar	256	Sim	Sim	-	-	-
phone	varchar	20	Sim	Sim	-	-	-

tb_users_faq							
--------------	--	--	--	--	--	--	--

Campo	Tipo	Tamanho	Obrigatório	Único	Chave	Valor (Padrão)	Regras de Validação
Id	varchar	256	Sim	Sim	PK	-	-
fullname	varchar	100	Sim	Não	-	-	-
email	varchar	100	Sim	Não	-	-	-
Mensagem	varchar	500	Sim	Nao	-	-	-
solicitation_status	varchar	20	Sim	Nao	-	-	-

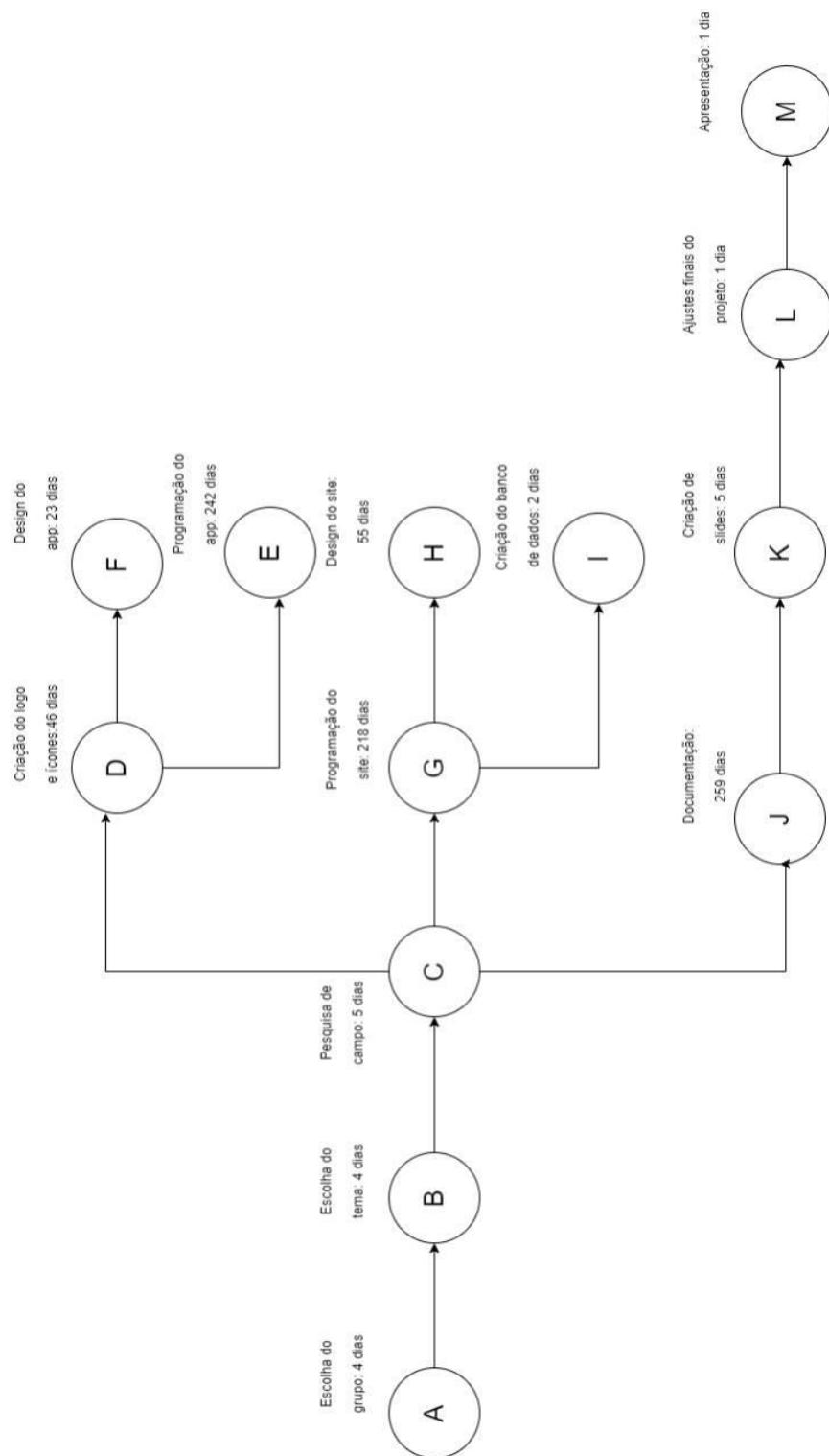
3.6. Cronograma de Atividades

Cronograma de Atividades

Nome da Tarefa	Tarefa	Duração	Início	Término	Predecessoras
A	Escolha do Grupo	2 dias	08/02/2022	10/02/2022	
B	Escolha do tema	14 dias	16/02/2022	02/03/2022	A
C	Pesquisa de campo	6 dias	09/03/2020	15/03/2022	B
D	Criação da Logo e dos Ícones	42 dias	16/03/2022	28/04/2022	C
E	Programação do Aplicativo	217 dias	28/04/2022	01/11/2022	F;D
F	Design do aplicativo	23 dias	28/04/2022	23/05/2022	D
G	Programação do Site	210 dias	28/04/2022	23/10/2022	D
H	Design do Site	20 dias	28/04/2022	20/05/2022	
I	Criação do banco de Dados	1 dia	20/05/2022	21/05/2022	G;E
J	Documentação	287 dias	10/02/2022	24/11/2022	E;G
K	Criação dos Slides para Apresentação Final	7 dias	17/11/2022	24/11/2022	J

3.6.1 Diagrama de Pert

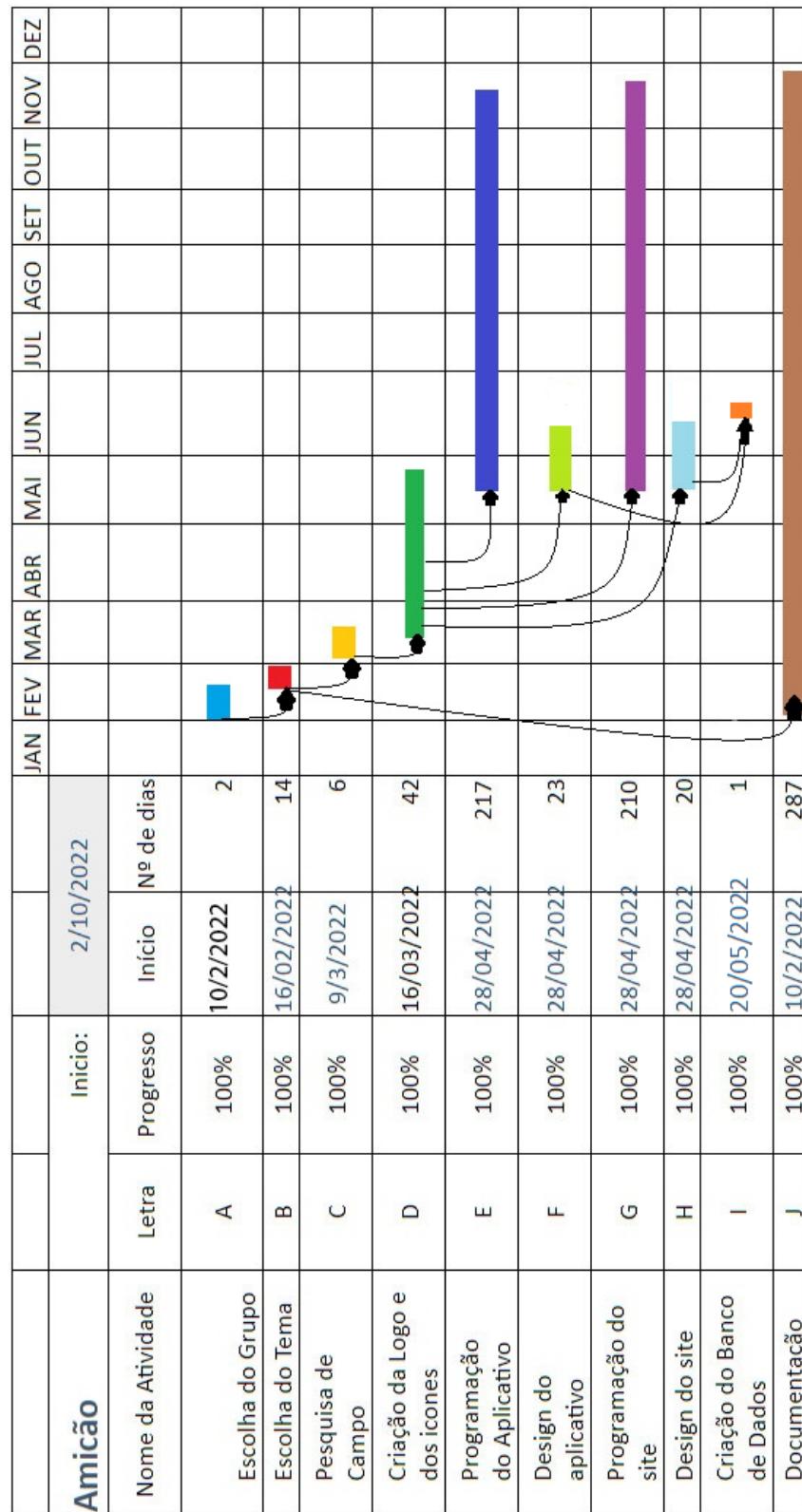
Figura 30 - Diagrama de Pert



Fonte: Autoria própria.

3.6.2 Diagrama de Grant

Figura 31 - Diagrama de Grant



Fonte: Autoria própria.

3.7 Requisitos Funcionais

3.7.1 Aplicativo

Código:	RF001
Classificação:	Obrigatória
Autor:	Usuário
Descrição:	Exibição de animais: Exibe todos os animais disponíveis para adoção que estão cadastrados no sistema
Pré-condições:	O usuário deve iniciar a aplicação
Pós-condições:	A tela que mostra as informações é exibida
Fluxo Principal:	1. O usuário inicia a aplicação e é direcionado a tela de animais disponíveis para a adoção.
Fluxo de Erro:	Não há
Fluxo Alternativo:	Não há

--	--

Código:	RF002
Classificação:	Optional
Autor:	Usuário
Descrição:	Gerenciar Animais: O sistema permite que o usuário selecione determinado animal, de modo que as informações sobre esse sejam exibidas, assim como as opções de selecionar o animal como favorito, adotar-lo ou apadrinhar-lo
Pré-condições:	O usuário deverá clicar no ícone do animal, cujo é exibido na tela de animais disponíveis para adoção
Pós-condições:	Serão exibidas as informações referentes ao animal selecionado e as devidas opções
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário inicia a aplicação e é direcionado a tela de animais disponíveis para a adoção. 2. O usuário seleciona o ícone do animal.
Fluxo de Erro:	Não há

Fluxo Alternativo:	Não há
---------------------------	---------------

Código:	RF003
Classificação:	Opcional
Autor:	Usuario
Descrição:	Favoritar: Essa funcionalidade permite ao usuário definir que os animais selecionados sejam exibidos em uma tela diferente, que possa ser acessada clicando em um ícone, onde estes serão exibidos.
Pré-condições:	O Usuário deverá iniciar o aplicativo, clicar no ícone do animal na tela de exibição dos animais disponíveis, de depois selecionar a opção “favoritar”
Pós-condições:	O animal será exibido na aba “favoritos”

Fluxo Principal:	<p>1. O usuário inicia a aplicação e é direcionado a tela de animais disponíveis para a adoção.</p> <p>2. O usuário seleciona o ícone do animal na tela de exibição de animais disponíveis e é direcionado a tela de gerenciamento de animais.</p> <p>3. O usuário seleciona a opção “favoritar” na aba de gerenciar animais.</p>
Fluxo de Erro:	Não há
Fluxo Alternativo:	Não há

3.7.2 Site

Código:	RF001
Classificação:	Opcional.
Autor:	Usuário.
Descrição:	Download do app mobile Android: o sistema deverá possibilitar ao usuário a realização do download do app através de um link.
Pré-Condições:	Não há.
Pós-Condições:	O usuário poderá instalar a aplicação em seu dispositivo Android.
Fluxo Principal:	O usuário acessa a página do site e clica no

	botão “Baixar” para que o sistema redirecione o usuário automaticamente ao link de download.
Fluxo de Erro:	Não há.
Fluxo Alternativo	Não há.

Código:	RF002
Classificação:	Obrigatório.
Autor:	Instituição.
Descrição:	Cadastro Institucional: o sistema deverá permitir que as instituições realizem cadastro na plataforma, para que possa cadastrar e gerenciar seus pets além de serem dispostos publicamente aos usuários que procuram adotar ou apadrinhar pets.
Pré-Condições:	Não há.
Pós-Condições:	A instituição deverá aguardar a análise e aprovação do cadastro.
Fluxo Principal:	<p>1 – Funcionário da instituição: acessa o site.</p> <p>2 – Funcionário da instituição: clica no botão “Institucional”.</p> <p>3 – Sistema: redireciona o usuário para página “Login”.</p> <p>4 – Funcionário da instituição: clica no botão “Realizar Cadastro”.</p>
Fluxo de Erro:	[FE001] Presença de dados em formato inválido: o sistema recarregará a página de cadastro com os dados já inseridos pelo usuário em exceção dos dados incorretos, além de retornar um

	<p>elemento na página informando ao funcionário os dados incorretos.</p> <p>[FE002] Ausência de dados: o sistema recarregará a pagina de cadastro com os dados já informados pelo usuário e um elemento informando a ausência de dados.</p>
Fluxo Alternativo	Funcionário da instituição acessa diretamente a página de “Cadastro” através da URL.

Código:	RF003
Classificação:	Opcional.
Ator:	Funcionário da plataforma.
Descrição:	Aprovação de cadastros: o sistema deverá fornecer ao funcionário da plataforma (cujo irá gerenciar as instituições) a possibilidade de poder aprovar a solicitação de cadastro das instituições após a análise voluntária do funcionário da plataforma diante dos dados submetidas pelas insituições durante seu cadastro.
Pré-Condições:	O funcionário da plataforma deverá estar autenticado com sua conta de funcionário, apenas se seguir pelo fluxo alternativo [FA001].
Pós-Condições:	O funcionário da instituição

	<p>recebe um e-mail informando a confirmação de seu cadastro no e-mail e então terá sua conta ativa para poder cadastrar gerenciar seus pets na plataforma.</p>
Fluxo Principal:	<p>1 – Funcionário da plataforma: acessa o website.</p> <p>2 – Funcionário da plataforma: clica no botão “Institucional”.</p> <p>3 – Sistema: redireciona para a página de login.</p> <p>4 – Sistema: verifica se há uma sessão de autenticação do usuário ativa.</p> <p>5 – Funcionário da plataforma: realiza login com sua conta de funcionário.</p> <p>6 – Sistema: redireciona para a página institucional com as funcionalidades disponíveis para o tipo de conta autenticada.</p> <p>7 – Funcionário da plataforma: clica no botão “Analizar cadastros”.</p> <p>8 – Sistema: redireciona para a página de “Cadastros das instituições”.</p> <p>9 – Funcionário da plataforma: realiza o filtro procurando apenas por cadastros em situação de análise.</p>

	<p>10 – Sistema: recarrega a página listando apenas cadastro em status de análise (não aprovados e/ou reprovados).</p> <p>11 – Funcionário da plataforma: seleciona um cadastro para inspecionar os dados.</p> <p>12 – Sistema: redireciona o usuário para uma página dinâmica com os dados da instituição selecionada, chamada “Instituição”.</p> <p>13 – Funcionário da instituição: clica em “Aprovar”.</p> <p>14 – Sistema: envia um e-mail para a instituição correspondente notificando a aprovação da conta institucional e a sua habilitação para poder usufluir da plataforma.</p>
Fluxo de Erro:	<p>[FE002] Caso não haja nenhuma instituição com cadastro efetuado, o sistema deverá recarregar a página com um elemento informando que não há instituições cadastradas na plataforma.</p> <p>[FE003] Caso após o filtro do funcionário da plataforma (passo 9 do Fluxo Principal deste requisito funcional) não houver</p>

	<p>nenhuma instituição correspondente, o sistema deverá recarregar a página com um elemento informando de que não há instituições em estágio de análise cadastral na plataforma.</p> <p>[FE004] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem, então o sistema redireciona para uma página de erro 404.</p> <p>[FE005] Caso quando o funcionário clicar no botão “Aprovar” a sessão de usuário estiver expirada ou inexistente, então o sistema redireciona para uma página de erro 404.</p>
Fluxo Alternativo	<p>[FA001] Funcionário da plataforma acessa qualquer uma das páginas intermediárias do Fluxo Principal diretamente através da URL.</p> <p>1 – Funcionário da plataforma: acessa uma página intermediária (Institucional, Cadastro das instituições ou Instituição).</p>

	<p>2 – Sistema: verifica se há alguma sessão ativa.</p> <p>3 – Sistema: verifica se a sessão de autenticação ativa é de uma conta tipo funcionário da plataforma.</p> <p>4 – Sistema: redireciona para a página em que o funcionário da plataforma estava requisitando no passo 1 deste Fluxo Alternativo, prosseguindo ao Fluxo Principal.</p>
--	---

Código:	RF004
Classificação:	Opcional.
Autor:	Funcionário da plataforma.
Descrição:	Reprovação de cadastros: o sistema deverá fornecer ao funcionário da plataforma (cujo irá gerenciar as instituições) a possibilidade de poder reprovar a solicitação de cadastro das instituições após a análise voluntária do funcionário da plataforma diante dos dados submetidas pelas instituições durante seu cadastro.
Pré-Condições:	O funcionário da plataforma deverá estar autenticado com

	sua conta de funcionário, apenas se seguir pelo fluxo alternativo [FA002].
Pós-Condições:	O funcionário da instituição recebe um e-mail informando a confirmação de seu cadastro no e-mail e então terá sua conta ativa para poder cadastrar gerenciar seus pets na plataforma.
Fluxo Principal:	<p>1 – Funcionário da plataforma: acessa o website.</p> <p>2 – Funcionário da plataforma: clica no botão “Institucional”.</p> <p>3 – Sistema: redireciona para a página de login.</p> <p>4 – Sistema: verifica se há uma sessão de autenticação do usuário ativa.</p> <p>5 – Funcionário da plataforma: realiza login com sua conta de funcionário.</p> <p>6 – Sistema: redireciona para a página institucional com as funcionalidades disponíveis para o tipo de conta autenticada.</p> <p>7 – Funcionário da plataforma: clica no botão “Analizar cadastros”.</p> <p>8 – Sistema: redireciona para a página de “Cadastros das instituições”.</p>

	<p>9 – Funcionário da plataforma: realiza o filtro procurando apenas por cadastros em situação de análise.</p> <p>10 – Sistema: recarrega a página listando apenas cadastro em status de análise (não aprovados e/ou reprovados).</p> <p>11 – Funcionário da plataforma: seleciona um cadastro para inspecionar os dados.</p> <p>12 – Sistema: redireciona o usuário para uma página dinâmica com os dados da instituição selecionada, chamada “Instituição”.</p> <p>13 – Funcionário da instituição: clica em “Reprovar”.</p> <p>14 – Sistema: redireciona para uma página de justificativa.</p> <p>15 – Funcionário da plataforma: submete uma justificativa.</p> <p>16 – Sistema: envia uma e-mail para a instituição correspondente informando a reprovação do cadastro e a justificativa.</p> <p>17 – Sistema: retorna à página de cadastro das instituições.</p>
Fluxo de Erro:	[FE006] Caso não haja nenhuma instituição com cadastro efetuado, o sistema deverá recarregar a página com um

	<p>elemento informando que não há instituições cadastradas na plataforma.</p>
	<p>[FE007] Caso após o filtro do funcionário da plataforma (passo 9 do Fluxo Principal deste requisito funcional) não houver nenhuma instituição correspondente, o sistema exibir a página com um elemento informando de que não há instituições em estágio de análise cadastral na plataforma.</p>
	<p>[FE008] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem, então o sistema redireciona para uma página de erro 404.</p>
	<p>[FE009] Caso quando o funcionário submeter a justificativa de reprovação, a sessão de usuário estiver expirada ou inexistente, então o sistema redireciona para uma página de erro 404.</p>
	<p>[FE010] Caso o funcionário</p>

	<p>submeta a justificativa em branco, o sistema recarrega a página com um elemento informando a ausência e a obrigatoriedade da justificativa para a efetivação da operação, retornando ao Fluxo Principal.</p>
Fluxo Alternativo	<p>[FA002] Funcionário da plataforma acessa qualquer uma das páginas intermediárias do Fluxo Principal diretamente através da URL.</p> <p>1 – Funcionário da plataforma: acessa uma página intermediária (Institucional, Cadastro das instituições ou Instituição).</p> <p>2 – Sistema: verifica se há alguma sessão ativa.</p> <p>3 – Sistema: verifica se a sessão de autenticação ativa é de uma conta tipo funcionário da plataforma.</p> <p>4 – Sistema: redireciona para a página em que o funcionário da plataforma estava requisitando no passo 1 deste Fluxo Alternativo, prosseguindo ao Fluxo Principal.</p>

Código:	RF005
Classificação:	Optional.
Autor:	Funcionário da plataforma.
Descrição:	SAU – Serviço de Atendimento ao Usuário (responder dúvidas, relatos ou requerimentos de usuário): o sistema deverá fornecer uma funcionalidade que possibilite ao funcionário da plataforma a visualizar e responder dúvidas, relatos ou requerimentos de usuários do website.
Pré-Condições:	O funcionário da plataforma deverá estar autenticado com sua conta de funcionário, apenas se seguir pelo fluxo alternativo [FA002].
Pós-Condições:	Os usuários receberam uma resposta no e-mail informado durante a submissão da mensagem.
Fluxo Principal:	<p>1 – Funcionário da plataforma: acessa o website.</p> <p>2 – Funcionário da plataforma: clica no botão “Institucional”.</p> <p>3 – Sistema: verifica se há uma sessão de autenticação do usuário ativa.</p> <p>4 – Sistema: redireciona para a página institucional com as funcionalidades pertinentes ao tipo de</p>

funcionário da plataforma.

5 – Funcionário da plataforma: clica no botão “Mensagens”.

6 – Sistema: redireciona para a páginas de mensagens, onde o sistema listará todas as mensagens dos usuários.

7 – Funcionário da plataforma: realiza o filtro para exibir apenas mensagens com status “não resolvido”.

8 – Sistema: recarrega a página listando apenas as mensagens correspondentes ao filtro.

9 – Funcionário da plataforma: clica em uma mensagem.

10 – Sistema: redireciona para uma página dinâmica onde serão inspecionados os dados da mensagem selecionada, chamada “Dados da mensagem”.

11 – Funcionário da plataforma: escreve uma resposta e seleciona o status da mensagem (resolvido ou em andamento) e submete as informações para o sistema.

12 – Sistema: envia um e-mail para o destino de e-mail fornecido pelo usuário da autoria da mensagem, contendo em seu corpo, a mensagem do funcionário da plataforma e seu status.

14 – Sistema: retorna à página de

	<p>mensagens com os dados dinâmicamente atualizados.</p>
Fluxo de Erro:	<p>[FE011] Caso não exista nenhuma mensagem de usuário, o sistema deverá exibir a página com um elemento apontando que ainda não há nenhuma mensagem.</p> <p>[FE012] Caso o usuário submeta a justificativa com ausência de dados, o sistema recarregará a página com um elemento notificando as informações ausentes e sua obrigatoriedade de serem preenchidas.</p> <p>[FE013] Caso após o filtro do funcionário na página de mensagens (passo 7 do Fluxo Principal deste requisito funcional), não haja nenhuma mensagem correspondente, o sistema deverá exibir um elemento na página notificando de que ainda não há nenhuma mensagem que atendam os critérios de busca.</p> <p>[FE0014] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem, então o sistema redireciona para uma página de erro 404.</p>

	<p>[FE0015] Caso quando o funcionário submeter a mensagem de resposta, a sessão de usuário estiver expirada ou inexistente, então o sistema redireciona para uma página de erro 404.</p>
Fluxo Alternativo	<p>[FA003] Funcionário da plataforma acessa qualquer uma das páginas intermediárias do Fluxo Principal diretamente através da URL.</p> <p>1 – Funcionário da plataforma: acessa uma página intermediária (Institucional, Mensagens ou Dados da mensagem).</p> <p>2 – Sistema: verifica se há alguma sessão ativa.</p> <p>3 – Sistema: verifica se a sessão de autenticação ativa é de uma conta tipo funcionário da plataforma.</p> <p>4 – Sistema: redireciona para a página em que o funcionário da plataforma estava requisitando no passo 1 deste Fluxo Alternativo, prosseguindo ao Fluxo Principal.</p>

Código:	RF006
Classificação:	Obrigatório.
Atores:	Funcionários da plataforma e instituições.
Descrição:	Login: o sistema deverá fornecer uma funcionalidade de login aos usuários (instituições e funcionários) para que possam criar uma sessão de autenticação ativa relativa ao seu tipo de conta e obter permissões especiais para a utilização de outras funcionalidades.
Pré-Condições:	Instituições: possuirem cadastro aprovado. Funcionários: possuirem cadastro de funcionário na plataforma.
Pós-Condições:	Funcionários: poderão utilizar as funcionalidades de SAU, gerenciamento de cadastros de instituições e alterações de dados da própria conta. Instituições: poderão utilizar as funcionalidades de gerenciamento de pets, gerenciamento de requisições de usuários e alterações de dados da própria conta.
Fluxo Principal:	1 – Usuário: acessa o website. 2 – Usuário: clica no botão “Institucional”.

	<p>3 – Sistema: verifica se há uma sessão de autenticação do usuário ativa, se caso houver, então o fluxo está finalizado.</p> <p>4 – Sistema: redireciona para a página de Login (caso não houver uma sessão ativa).</p> <p>5 – Usuário: insere suas credencias de login (e-mail e senha)e submete o formulário.</p> <p>6 – Sistema: verifica se as credencias conferem.</p> <p>7 – Sistema: cria uma sessão ativa de autenticação do usuário, com expiração de 7 dias, podendo ser destruida antes com logout ou exclusão da sessão do navegador do usuário.</p> <p>8 – Sistema: verifica o tipo de conta (se instituição ou usuário) e atribui as permissões correspondente às Pós-Condições.</p>
Fluxo de Erro:	<p>[FE016] As credenciais não conferem: o sistema irá recarregar a página de login com um elemento informando que as credenciais fornecidas são inválidas.</p> <p>[FE017] O usuário submete o formuário de login com algum campo em branco: o sistema irá recarregar a página com um elemento informando</p>

	<p>a obrigatoriedade do campo nulo e que deve ser preenchido.</p>
Fluxo Alternativo	<p>[FA004] O usuário chega à tela de login por meio de links diretos de outras páginas do site que requerem autenticação:</p> <p>1 – Usuário: acessa alguma página privada (qualquer página que requira autenticação) do website.</p> <p>3 – Sistema: verifica se há uma sessão de autenticação do usuário ativa, se caso houver, então o fluxo está finalizado.</p> <p>4 – Sistema: redireciona para a página de Login (caso não houver uma sessão ativa).</p> <p>5 – Usuário: insere suas credenciais de login (e-mail e senha) e submete o formulário.</p> <p>6 – Sistema: verifica se as credenciais conferem.</p> <p>7 – Sistema: cria uma sessão ativa de autenticação do usuário, com expiração de 7 dias, podendo ser destruída antes com logout ou exclusão da sessão do navegador do usuário.</p> <p>8 – Sistema: verifica o tipo de conta (se instituição ou usuário) e atribui as permissões correspondente às Pós-Condições.</p>

Código:	RF007
Classificação:	Optional.
Autor:	Funcionários da plataforma e instituições.
Descrição:	Recuperação de senha: o sistema deverá fornecer uma funcionalidade ao usuário que permita-o recuperar o acesso a sua conta (processo usualmente chamado de "Recuperação de senha", porém se trata de definir outra senha à conta).
Pré-Condições:	O usuário deverá ter acesso a sua conta de e-mail (a mesma utilizada na conta da plataforma).
Pós-Condições:	O usuário (funcionário ou instituição) terá acesso a sua conta através de sua nova senha.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 – Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: redireciona para página de login.</p> <p>5 – Usuário: clica no botão “Recuperar senha”.</p> <p>6 – Sistema: redireciona para a página de recuperação de senha.</p>

	<p>7 – Usuário: coloca seu e-mail e submete o formulário.</p> <p>8 – Sistema: gera e envia um link de recuperação de senha para o e-mail do usuário.</p> <p>9 – Usuário: acessa seu e-mail e acessa o link de recuperação de senha.</p> <p>10 – Sistema: verifica se o link é valido.</p> <p>11 – Sistema: verifica se o link expirou.</p> <p>12 – Sistema: redireciona para a página de definição de nova senha.</p> <p>13 – Usuário: insere a senha e submete o formulário.</p> <p>14 – Sistema: verifica novamente a integridade do link de recuperação de senha.</p> <p>15 – Sistema: atualiza a senha do usuário no banco de dados.</p> <p>16 – Sistema: redireciona usuário para a página principal.</p>
Fluxo de Erro:	<p>[FE018] Caso o usuário submeta o formulário de confirmação do e-mail em branco: o sistema irá recarregar a página com um elemento notificando a obrigatoriedade do campo em branco.</p> <p>[FE019] Caso o usuário submeta o formulário de confirmação de e-mail,</p>

	<p>com um e-mail inválido: o sistema irá recarregar a página com um elemento notificando de que o e-mail é inválido.</p> <p>[FE020] Caso o usuário submeta o formulário de confirmação de e-mail, com um e-mail que não corresponde a nenhum usuário.</p> <p>[FE021] Caso o link de recuperação não seja válido ou esteja expirado: o sistema redireciona para a página de erro 404.</p> <p>[FE022] Caso o usuário submeta o formulário de definição de nova senha em branco: o sistema irá recarregar a página com um elemento notificando o obrigatoriedade do preenchimento da senha.</p>
Fluxo Alternativo	<p>[FA005] O usuário acessa qualquer página do website que requira autenticação.</p> <p>1 – Usuário: acessa qualquer página do website que requira autenticação.</p> <p>2– Sistema: verifica se há uma sessão de usuário ativa.</p> <p>3 – Sistema: redireciona para página de login.</p> <p>4 – Usuário: clica no botão</p>

	<p>“Recuperar senha”.</p> <p>5 – Sistema: redireciona para a página de recuperação de senha.</p> <p>6 – Usuário: coloca seu e-mail e submete o formulário.</p> <p>7 – Sistema: gera e envia um link de recuperação de senha para o e-mail do usuário.</p> <p>8 – Usuário: acessa seu e-mail e acessa o link de recuperação de senha.</p> <p>9 – Sistema: verifica se o link é valido.</p> <p>10 – Sistema: verifica se o link expirou.</p> <p>11 – Sistema: redireciona para a página de definição de nova senha.</p> <p>12 – Usuário: insere a senha e submete o formulário.</p> <p>13 – Sistema: verifica novamente a integridade do link de recuperação de senha.</p> <p>14 – Sistema: atualiza a senha do usuário no banco de dados.</p> <p>15 – Sistema: redireciona usuário para a página principal.</p>
--	--

Código:	RF008
Classificação:	Opcional.
Autor:	Instituição.

Descrição:	Cadastro de pets: o sistema deverá fornecer uma funcionalidade que permita ao funcionário / usuário da instituição a cadastrar seus pets.
Pré-Condições:	O usuário deverá estar devidamente logado em uma conta do tipo institucional aprovada.
Pós-Condições:	O usuário poderá deletar e alterar informações sobre o pet cadastrado.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 - Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: verifica se a conta logada é do tipo institucional.</p> <p>5 – Sistema: redireciona para a página Institucional com as funcionalidades permissivas relacionadas ao tipo de conta.</p> <p>6 – Usuário: clica em “Gerenciar Pets”.</p> <p>7 – Sistema: redireciona para a página de gerenciamento de pets.</p> <p>8 – Usuário: clica em “Cadastrar pet”.</p> <p>9 – Sistema: redireciona para a página de cadastramento de pets.</p> <p>10 – Usuário: insere as informações do pet e submete o formulário de cadastramento.</p> <p>11 – Sistema: verifica se a sessão do</p>

	<p>usuário ainda está ativa.</p> <p>12 – Sistema: insere os dados no banco de dados.</p> <p>13 - Sistema: redireciona usuário para a página de gerenciamento de pets com um elemento notificando que o cadastramento do pet foi realizado com sucesso.</p>
Fluxo de Erro:	<p>[FE023] Caso a sessão de autenticação do usuário não esteja ativa ou esteja expirada no envio dos dados do pet para salvamento no banco de dados: o sistema cancelará a operação e redirecionará para uma página de erro 404.</p> <p>[FE024] Caso a sessão de autenticação do usuário não esteja ativa ou esteja expirada no momento de verificação da sessão pelo sistema: o sistema redirecionará para uma página de erro 404.</p> <p>[FE024] Caso o usuário submeta algum dado do pet em branco: o sistema irá recarregar a página com um elemento notificando a obrigatoriedade dos dados.</p> <p>[FE025] Caso o usuário submeta algum dado do pet em formato inválido: o sistema irá recarregar a</p>

	<p>página com um elemento notificando obrigatoriedade do preenchimento válido do dado.</p> <p>[FE026] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem: o sistema redireciona para uma página de erro 404.</p> <p>[FE027] Caso não haja uma sessão de usuário ativa quando o usuário acessar uma página privada (que requer autenticação):</p> <ol style="list-style-type: none"> 1 – Sistema: redireciona para a página de login. 2 – Sistema: redireciona para a página em que o usuário estava tentando acessar. (retorna ao Fluxo Principal).
Fluxo Alternativo	<p>[FA006] O usuário acessa a página de cadastramento de pets via link direto:</p> <ol style="list-style-type: none"> 1 – Usuário: clica em “Cadastrar pet”. 2 – Sistema: redireciona para a página de cadastramento de pets. 3 – Usuário: insere as informações do pet e submete o formulário de cadastramento. 4 – Sistema: verifica se a sessão do usuário ainda está ativa.

	<p>5 – Sistema: insere os dados no banco de dados.</p> <p>6 - Sistema: redireciona usuário para a página de gerenciamento de pets com um elemento notificando que o cadastramento do pet foi realizado com sucesso.</p>
--	---

Código:	RF009
Classificação:	Opcional.
Autor:	Instituição.
Descrição:	Deleção de pet: o sistema deverá fornecer uma funcionalidade que permita ao usuário / instituição a realizar a deleção de um pet.
Pré-Condições:	<p>1 – Usuário: estar devidamente logado em uma conta do tipo institucional aprovada.</p> <p>2 – Usuário: ter cadastrado um pet na plataforma.</p>
Pós-Condições:	Não há.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 - Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: verifica se a conta logada é do tipo institucional.</p> <p>5 – Sistema: redireciona para a página Institucional com as</p>

	<p>funcionalidades permissivas relacionadas ao tipo de conta.</p> <p>6 – Usuário: clica em “Gerenciar Pets”.</p> <p>7 – Sistema: redireciona para a página de gerenciamento de pets.</p> <p>8 – Usuário: clica em “Deletar” sobre o pet em que quer remover da plataforma.</p> <p>9 – Sistema: verifica se há uma sessão de autenticação do usuário ativa.</p> <p>10 – Sistema: verifica se o pet a ser deletado pertence a instituição logada.</p> <p>11 – Sistema: realiza a deleção do pet da base de dados.</p> <p>12 – Sistema: recarrega a página com um elemento notificando o pet foi deletado com sucesso.</p>
Fluxo de Erro:	<p>[FE028] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem, então o sistema redireciona para uma página de erro 404.</p> <p>[FE029] Caso não haja uma sessão de usuário ativa quando o usuário acessar uma página privada (que requer autenticação):</p>

	<p>1 – Sistema: redireciona para a página de login.</p> <p>2 – Sistema: redireciona para a página em que o usuário estava tentando acessar. (retorna ao Fluxo Principal).</p>
Fluxo Alternativo	<p>[FA007] Usuário acessa diretamente a página de gerenciamento de pets:</p> <p>1 – Sistema: redireciona para a página de gerenciamento de pets.</p> <p>2 – Usuário: clica em “Deletar” sobre o pet em que quer remover da plataforma.</p> <p>3 – Sistema: verifica se há uma sessão de autenticação do usuário ativa.</p> <p>4 – Sistema: verifica se o pet a ser deletado pertence a instituição logada.</p> <p>5 – Sistema: realiza a deleção do pet da base de dados.</p> <p>6 – Sistema: recarrega a página com um elemento notificando o pet foi deletado com sucesso.</p>

Código:	RF0010
Classificação:	Opcional.
Autor:	Instituição.
Descrição:	Alterar pet: o sistema deverá fornecer

	uma funcionalidade que permita ao usuário / instituição realizar a alteração de dados de um pet.
Pré-Condições:	<p>1 – Usuário: estar devidamente logado em uma conta do tipo institucional aprovada.</p> <p>2 – Usuário: ter cadastrado um pet na plataforma.</p>
Pós-Condições:	Não há.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 - Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: verifica se a conta logada é do tipo institucional.</p> <p>5 – Sistema: redireciona para a página Institucional com as funcionalidades permissivas relacionadas ao tipo de conta.</p> <p>6 – Usuário: clica em “Gerenciar Pets”.</p> <p>7 – Sistema: redireciona para a página de gerenciamento de pets.</p> <p>8 – Usuário: clica em “Alterar” no pet em que deseja alterar.</p> <p>9 – Sistema: redireciona para a página de alteração de pet com os dados do pet selecionado para alteração.</p> <p>10 – Usuário: altera as informações</p>

	<p>do pet e submete o formulário de alteração.</p> <p>11 – Sistema: verifica se a sessão do usuário ainda está ativa.</p> <p>12 – Sistema: atualiza os dados no banco de dados.</p> <p>13 - Sistema: redireciona usuário para a página de gerenciamento de pets com um elemento notificando que a alteração do pet foi realizada com sucesso.</p>
Fluxo de Erro:	<p>[FE028] Caso a sessão de autenticação do usuário não esteja ativa ou esteja expirada no envio dos dados do pet para atualização no banco de dados: o sistema cancelará a operação e redirecionará para uma página de erro 404.</p> <p>[FE029] Caso a sessão de autenticação do usuário não esteja ativa ou esteja expirada no momento de verificação da sessão pelo sistema: o sistema redirecionará para uma página de erro 404.</p> <p>[FE030] Caso o usuário submeta algum dado do pet em branco: o sistema irá recarregar a página com um elemento notificando a obrigatoriedade dos dados.</p>

	<p>[FE031] Caso o usuário submeta algum dado do pet em formato inválido: o sistema irá recarregar a página com um elemento notificando obrigatoriedade do preenchimento válido do dado.</p> <p>[FE032] Caso a sessão de autenticação do usuário ativa seja do tipo instituição (conta de uma instituição) quando o sistema realizar a checagem: o sistema redireciona para uma página de erro 404.</p> <p>[FE033] Caso não haja uma sessão de usuário ativa quando o usuário acessar uma página privada (que requer autenticação):</p> <ol style="list-style-type: none"> 1 – Sistema: redireciona para a página de login. 2 – Sistema: redireciona para a página em que o usuário estava tentando acessar. (retorna ao Fluxo Principal).
Fluxo Alternativo	<p>[FA008] Usuário acessa a página de alteração de pet diretamente via link:</p> <ol style="list-style-type: none"> 1 - Sistema: verifica se há uma sessão de usuário ativa. 2 – Sistema: verifica se a conta logada é do tipo institucional. 3 – Sistema: redireciona para a

	<p>página de alteração de pet com os dados do pet selecionado para alteração.</p> <p>4 – Usuário: altera as informações do pet e submete o formulário de alteração.</p> <p>5 – Sistema: verifica se a sessão do usuário ainda está ativa.</p> <p>6 – Sistema: atualiza os dados no banco de dados.</p> <p>7 - Sistema: redireciona usuário para a página de gerenciamento de pets com um elemento notificando que a alteração do pet foi realizada com sucesso.</p>
--	---

Código:	RF011
Classificação:	Opcional.
Autor:	Instituição e Funcionário da plataforma.
Descrição:	Alterar senha da conta: o sistema deverá ter uma funcionalidade que permite ao usuário a alterar a senha de sua conta.
Pré-Condições:	<p>1 – Usuário: estar devidamente logado em uma conta do tipo institucional aprovada.</p> <p>2 – Usuário: ter cadastrado um pet na plataforma.</p>
Pós-Condições:	1 – Usuário: poderá efetuar login com

	<p>a sua nova senha.</p> <p>2 – Sistema: a senha antiga do usuário não será mais considerada uma credencia do usuário.</p>
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 - Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: verifica se a conta logada é do tipo institucional.</p> <p>5 – Sistema: redireciona para a página Institucional com as funcionalidades permissivas relacionadas ao tipo de conta.</p> <p>6 – Usuário: clica no botão “Minha Conta”.</p> <p>7 – Sistema: redireciona o usuário para a página “Minha Conta”.</p> <p>8 – Usuário: preeche os campos de senha e confirmação de senha na seção de alteração de senha.</p> <p>9 – Usuário: clica no botão “Alterar senha”.</p> <p>10 – Sistema: atualiza a senha na base de dados.</p> <p>11 – Sistema: redireciona para a página de mensagem notificando a alteração da senha comm sucesso, com um botão OK.</p>
Fluxo de Erro:	[FE034] Caso o usuário clique no

	<p>botão “Alterar senha” com um dos campos de senha (Senha e Confirmação de Senha) em branco: o sistema recarregará a página com elemento notificando a obrigatoriedade de preenchimento dos campos, cancelando a operação.</p> <p>[FE035] Caso o usuário clique no botão “Alterar senha” com valores de senha e confirmação distintos: o sistema recarregará a página com um elemento notificando de que as credenciais não conferem, cancelando a operação.</p> <p>[FE036] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal, quanto na submissão da nova senha, cancelando a operação.</p>
Fluxo Alternativo	<p>[FA009] Usuário acessa a página “Minha Conta” diretamente através de link URL:</p> <p>1 – Usuário: acessa a página “Minha Conta” pela URL.</p> <p>2 – Sistema: verifica se há uma sessão de autenticação do usuário.</p> <p>3 – Sistema: redireciona o usuário</p>

	<p>para a página “Minha Conta”.</p> <p>4 – Usuário: preeche os campos de senha e confirmação de senha na seção de alteração de senha.</p> <p>5 – Usuário: clica no botão “Alterar senha”.</p>
--	---

Código:	RF012
Classificação:	Optional.
Autor:	Instituição.
Descrição:	Aceitar requisições: o sistema deverá fornecer uma funcionalidade que permita ao usuário a aceitar agendamentos / requisições de usuários do app com relação aos pets, sendo elas, por exemplo, agendamento do usuário para conhecer um pet, adotar ou apadrinhar, trâmites realizados pessoalmente.
Pré-Condições:	O funcionário da instituição deverá estar autenticado com sua conta de funcionário, apenas se seguir pelo fluxo alternativo.
Pós-Condições:	Não há.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 – Sistema: verifica se há uma sessão de usuário ativa.</p>

	<p>4 – Sistema: redireciona para a página “Institucional” com as funcionalidades permissivas relacionadas ao tipo de conta logada.</p> <p>5 – Usuário: clica em “Gerenciar requisições”.</p> <p>6 – Sistema: redireciona para a página de “Requisições”.</p> <p>8 – Usuário: realiza o filtro procurando apenas por requisições ainda não avaliadas.</p> <p>9 – Sistema: realiza a busca na base de dados e retorna os resultados.</p> <p>10 – Usuário: clica na opção “Aceitar” de uma determinada requisição.</p> <p>11 – Sistema: recarrega a página de gerenciamento de requisições com um elemento notificando que a requisição foi aprovada com sucesso, além de atualizar as informações na base de dados.</p>
Fluxo de Erro:	<p>[FE037] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p> <p>[FE038] Caso não haja resultados para o filtro do usuário: o sistema recarrega a página com um elemento notificando que de não há resultados</p>

	<p>que confiram com os critérios de busca.</p> <p>[FE039] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente no momento do click em “Aceitar” com relação a uma requisição: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p>
Fluxo Alternativo	<p>[FA010] Caso o usuário acesse diretamente a página de gerenciamento de requisições via url / link:</p> <ol style="list-style-type: none"> 1 – Sistema: verifica se há uma sessão de usuário ativa. 2 – Sistema: redireciona para a página de “Requisições”. 3 – Usuário: realiza o filtro procurando apenas por requisições ainda não avaliadas. 4 – Sistema: realiza a busca na base de dados e retorna os resultados. 5 – Usuário: clica na opção “Aceitar” de uma determinada requisição. 6 – Sistema: redireciona para a página de gerenciamento de requisições com um elemento notificando que a requisição foi aprovada com sucesso, além de atualizar as informações na base de

	dados.
--	--------

Código:	RF013
Classificação:	Optional.
Autor:	Instituição.
Descrição:	Recusar / Negar requisições: o sistema deverá fornecer uma funcionalidade que permita ao funcionário da instituição a recusar requisições relacionadas aos seus pets realizadas por usuários comuns, sendo elas, por exemplo, agendamento do usuário para conhecer um pet, adotar ou apadrinhar, trâmites realizados pessoalmente.
Pré-Condições:	O funcionário da instituição deverá estar autenticado com sua conta de funcionário, apenas se seguir pelo fluxo alternativo.
Pós-Condições:	Não há.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 – Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: redireciona para a página “Institucional” com as funcionalidades permissivas relacionadas ao tipo de conta logada.</p>

	<p>5 – Usuário: clica em “Gerenciar requisições”.</p> <p>6 – Sistema: redireciona para a página de “Requisições”.</p> <p>8 – Usuário: realiza o filtro procurando apenas por requisições ainda não avaliadas.</p> <p>9 – Sistema: realiza a busca na base de dados e retorna os resultados.</p> <p>10 – Usuário: clica na opção “Recusar” de uma determinada requisição.</p> <p>11 – Sistema: redireciona para a página de justificativa de recusa.</p> <p>12 – Usuário: preenche a justificativa e submete o formulário.</p> <p>13 – Sistema: redireciona para a página de gerenciamento de requisições com um elemento notificando que a requisição foi recusada com sucesso, além de atualizar as informações na base de dados.</p>
Fluxo de Erro:	<p>[FE040] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p> <p>[FE041] Caso não haja resultados para o filtro do usuário: o sistema</p>

	<p>recarrega a página com um elemento notificando que de não há resultados que confirmam com os critérios de busca.</p> <p>[FE042] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente no momento do click em “Recusar” com relação a uma requisição: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p> <p>[FE043] Caso o usuário submeta o formulário de justificativa em branco: o sistema recarrega a página com um elemento informando a obrigatoriedade do preenchimento do campo de justificativa.</p> <p>[FE043] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente no momento de submissão do formulário de justificativa: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p>
Fluxo Alternativo	[FA011] Caso o usuário acesse diretamente a página de gerenciamento de requisições via url /

	<p>link:</p> <p>1 – Sistema: verifica se há uma sessão de usuário ativa.</p> <p>2 – Sistema: redireciona para a página de “Requisições”.</p> <p>3 – Usuário: realiza o filtro procurando apenas por requisições ainda não avaliadas.</p> <p>4 – Sistema: realiza a busca na base de dados e retorna os resultados.</p> <p>5 – Usuário: clica na opção “Recusar” de uma determinada requisição.</p> <p>6 – Sistema: redireciona para a página de justificativa de recusa.</p> <p>7 – Usuário: preenche a justificativa e submete o formulário.</p> <p>8 – Sistema: redireciona para a página de gerenciamento de requisições com um elemento notificando que a requisição foi recusada com sucesso, além de atualizar as informações na base de dados.</p>
--	---

Código:	RF014
Classificação:	Opcional.
Autor:	Instituição.
Descrição:	Alteração de requisição: o sistema deverá fornecer uma funcionalidade que permita ao funcionário da instituição à alterar uma requisição (a

	mesma relacionada anteriormente nos dois últimos requisitos funcionais).
Pré-Condições:	O funcionário da instituição deverá estar autenticado com sua conta de funcionário, apenas se seguir pelo fluxo alternativo.
Pós-Condições:	Não há.
Fluxo Principal:	<p>1 – Usuário: acessa o website.</p> <p>2 – Usuário: clica no botão “Institucional”.</p> <p>3 – Sistema: verifica se há uma sessão de usuário ativa.</p> <p>4 – Sistema: redireciona para a página “Institucional” com as funcionalidades permissivas relacionadas ao tipo de conta logada.</p> <p>5 – Usuário: clica em “Gerenciar requisições”.</p> <p>6 – Sistema: redireciona para a página de “Requisições”.</p> <p>8 – Usuário: realiza o filtro procurando apenas por requisições ainda não avaliadas.</p> <p>9 – Sistema: realiza a busca na base de dados e retorna os resultados.</p> <p>10 – Usuário: clica na opção “Alterar” de uma determinada requisição.</p> <p>11 – Sistema: redireciona para a página de alteração da requisição.</p> <p>12 – Usuário: seleciona uma data e</p>

	<p>preenche uma justificativa de alteração.</p> <p>13 – Sistema: redireciona para a página de gerenciamento de requisições com um elemento notificando que a requisição foi alterada com sucesso, além de atualizar as informações na base de dados.</p>
Fluxo de Erro:	<p>[FE044] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p> <p>[FE045] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente no momento do click em “Alterar” com relação a uma requisição: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.</p> <p>[FE046] Caso o usuário submeta o formulário de alteração da requisição com algum dado em branco: o sistema recarrega a página com um elemento notificando a obrigatoriedade da preenchimento da informação faltante.s</p>

	[FE047] Caso o usuário esteja com sua sessão de autenticação expirada ou inexistente no momento de submissão do formulário de alteração: o sistema redireciona para a página de erro 404 tanto na verificação do sistema pelo Fluxo Principal.
Fluxo Alternativo	<p>[FA011] Caso o usuário acesse diretamente a página de gerenciamento de requisições via url / link:</p> <ol style="list-style-type: none"> 1 – Sistema: verifica se há uma sessão de usuário ativa. 2 – Sistema: redireciona para a página de “Requisições”. 3 – Usuário: clica na opção “Recusar” de uma determinada requisição. 4 – Sistema: redireciona para a página de justificativa de recusa. 5 – Usuário: preenche a justificativa e submete o formulário. 6 – Sistema: redireciona para a página de gerenciamento de requisições com um elemento notificando que a requisição foi recusada com sucesso, além de atualizar as informações na base de dados.

3.8 Requisitos Não Funcionais

3.8.1 Aplicativo

Código	Descrição	Tipo	Classificação
001	As informações contidas no banco de dados devem ser disponibilizadas para inclusão e modificação somente para os administradores via website.	Segurança	Obrigatório
002	O aplicativo deve ter a capacidade de exibir todos os registros existentes no servidor.	Escalabilidade	Obrigatório
003	O aplicativo deve ser compatível com todos os dispositivos android com a versão do sistema 5.0 ou superior (API	Portabilidade	Obrigatório

	level 21+).		
004	O aplicativo deve ter uma UI compatível com diversos tamanhos de tela.	Portabilidade	Desejável
005	A consulta dos dados de animais favoritados deve ocorrer em menos de 1 segundo.	Desempenho	Desejável
006	Os resultados de pesquisas de animais no servidor devem ocorrer em menos de 2 segundos.	Desempenho	Desejável
007	O usuário poderá utilizar um esquema de cores diferente.	Configurações	Desejável
008	Os registros de animais devem estar disponíveis na tela inicial por meio de uma	Interoperabilidade	Obrigatório

	conexão ao servidor Heroku.		
009	O aplicativo deverá ser desenvolvido na linguagem de programação java.	Implementação	Obrigatório

3.8.2 Site

Código	Descrição	Tipo	Classificação
RNF.001	O sistema deve permitir que seus clientes / usuários façam o download da aplicação mobile Android	Desempenho	Obrigatório
RNF.002	O sistema deve permitir que seus usuários alterem as suas informações após efetuarem seu cadastro, tê-lo aprovado e após ter efetuado login.	Configurabilidade	Obrigatório
RNF.003	O tempo de I/O	Desempenho	Desejável

	<p>de recuperação de dados de pets, cadastros não aprovados ou reprovados, e requisições, não devem ultrapassar 2 segundos para uma quantidade considerável de registros (cerca de 20 registros).</p>		
RNF.004	O website deve possuir compatibilidade com diferentes telas de dispositivos do cliente / usuário.	Portabilidade	Obrigatório
RNF.005	A visualização e alteração parcial dos dados da base de dados referentes aos pets, devem ser restritos exclusivamente para Usuários Instituições que contenham a autoria dos dados.	Segurança	Obrigatório

RNF.006	A visualização e alteração parcial dos dados da base de dados referentes aos cadastros não analisados de instituições, devem ser restringidos exclusivamente para os Usuários Funcionários da Plataforma.	Segurança	Obrigatório
RNF.007	Todo e qualquer acesso à páginas e rotas do website que sejam privadas, tais como, páginas posteriores ao fluxo de Institucional, e páginas dinâmicas específicas com parâmetros de recuperação de dados em rotas, devem ser protegidos por autenticação	Segurança	Obrigatório

	(Login) e as mesmas só devem ser servidas e/ou performadas pelos usuários, se a rota ou página realmente pertencer ao usuário.		
RNF.008	Toda resposta (I/O) de submissão de formulário por parte dos usuários do sistema, devem demorar no máximo 3 segundos.	Desempenho	Desejável
RNF.009	A visualização e alteração dos dados referentes às requisições, bem como, agendamentos para apadrinhamento, adoção, etc. devem ser devidamente restringidos exclusivamente	Segurança	Obrigatório

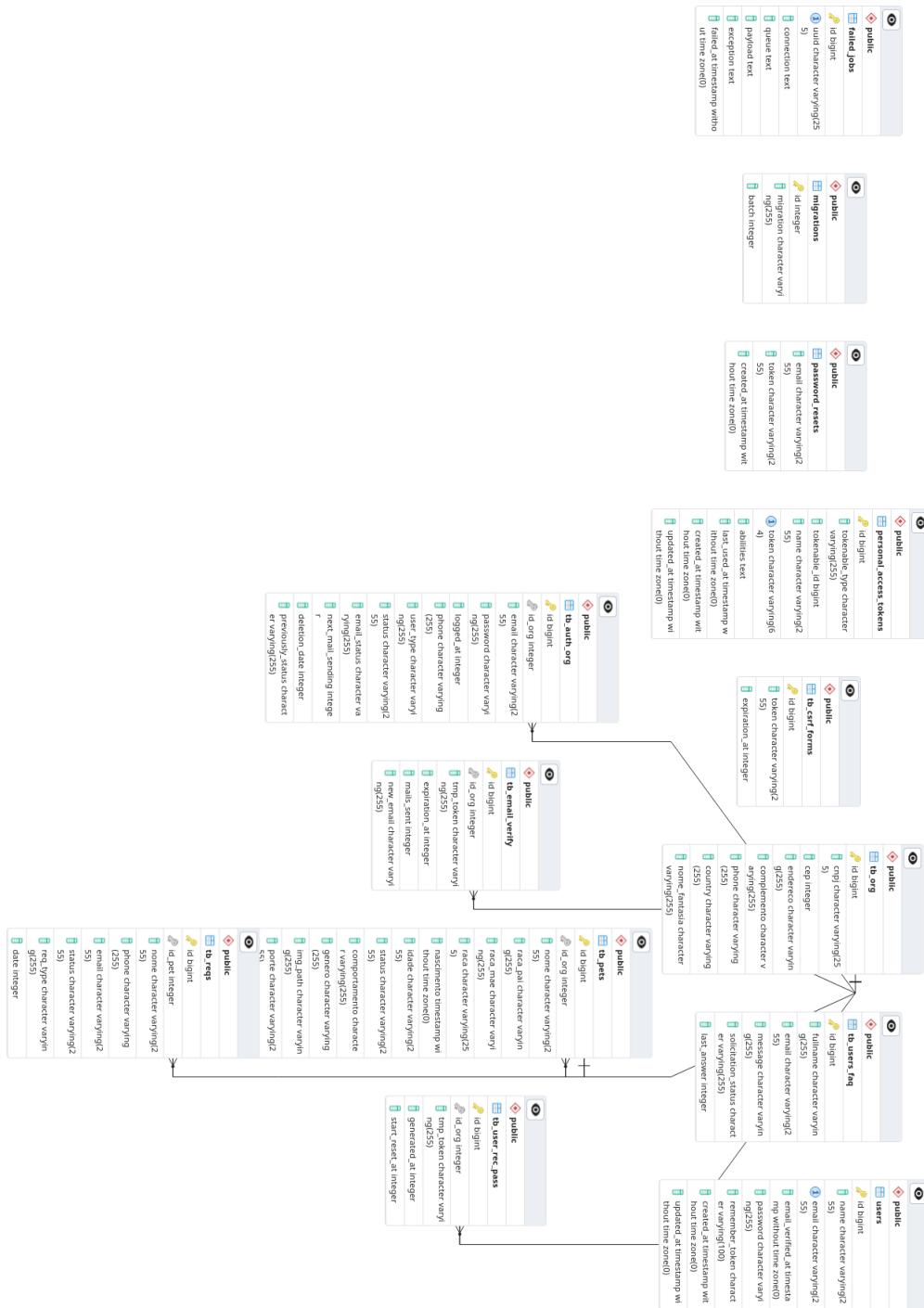
	para as instituições que contêm a propriedade dos dados referentes ao pet em que se foi realizada uma requisição por parte do usuário do app.		
RNF.010	A visualização das mensagens de contato deve ser restringida exclusivamente para os Usuários Funcionários da Plataforma.	Segurança	Obrigatório
RNF.011	Todo e qualquer formulário ou requisição submetida pelo usuário, deverá conter em seu cabeçalho de requisição, um token CSRF para proteção contra ataques de requisição forjada em uma sessão de autenticação de usuário ativa.	Segurança	Obrigatório

RNF.012	Links de redefinição de senha de um usuário específico deverá ser criptográfico com função de Hash resistente a colisões, deverá conter um tempo de expiração de 7 dias, e ser inacessível ou invalidado após a sua utilização.	Segurança	Obrigatório
RNF.013	O sistema deverá conter segurança contra-ataques XSS (Cross-site-scripting).	Segurança	Obrigatório
RNF.014	O sistema deverá conter segurança contra-ataques de SQL Injection, utilizando de framework Eloquent ou outras soluções de tratamento de entrada e abstração SQL para orientação à objetos, para	Segurança	Obrigatório

	manipulação de dados da base de dados.		
RNF.015	O sistema só poderá permitir a autenticação de Usuários Instituição, se somente se, os mesmos tiverem seus cadastros no status "ativo" após a análise dos Usuários Funcionários da Plataforma. Bem como os Usuários Instituição, só poderão usufluir da plataforma se estiverem autenticados.	Segurança	Obrigatório

3.9 Diagrama de Entidade e Relacionamento

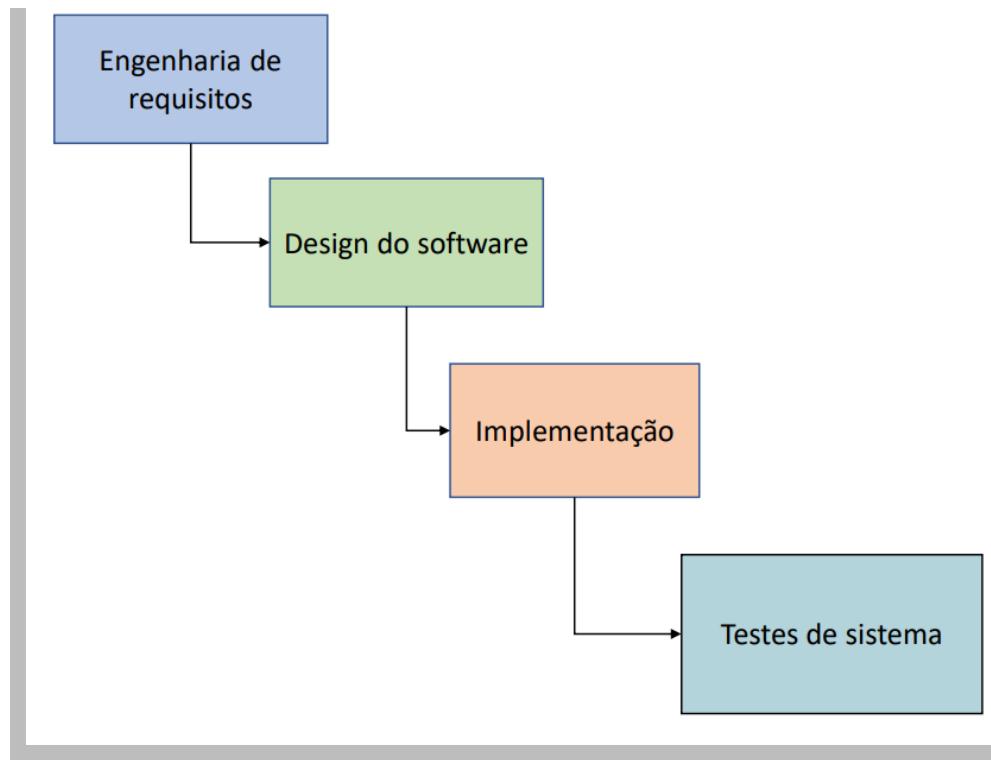
Figura 32 - Diagrama de Entidade e Relacionamento



Fonte: Autoria própria.

3.10. Ciclo de vida do sistema

Figura 33 - Ciclo de vida do sistema

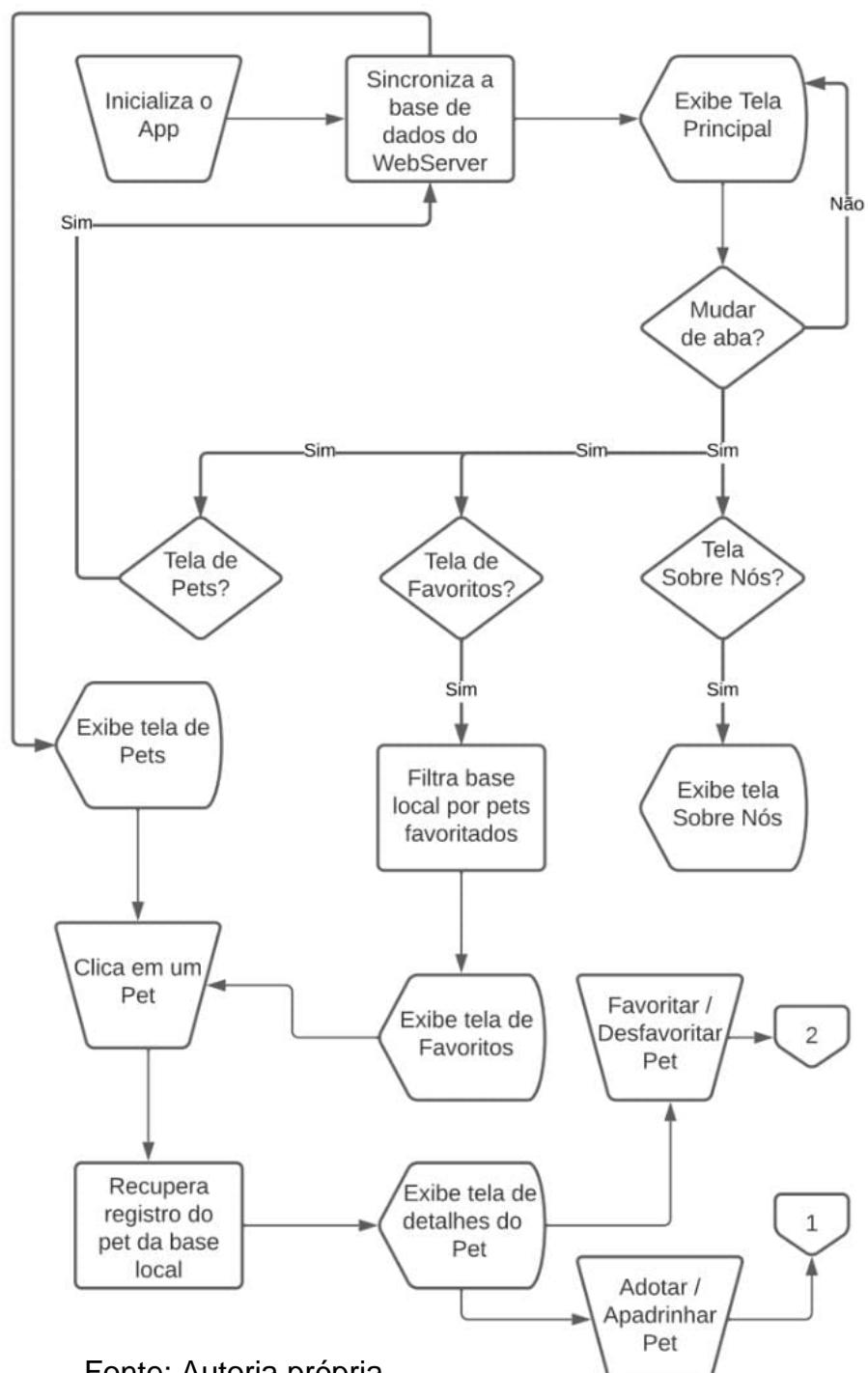


Fonte: Autoria própria.

3.11. Fluxograma

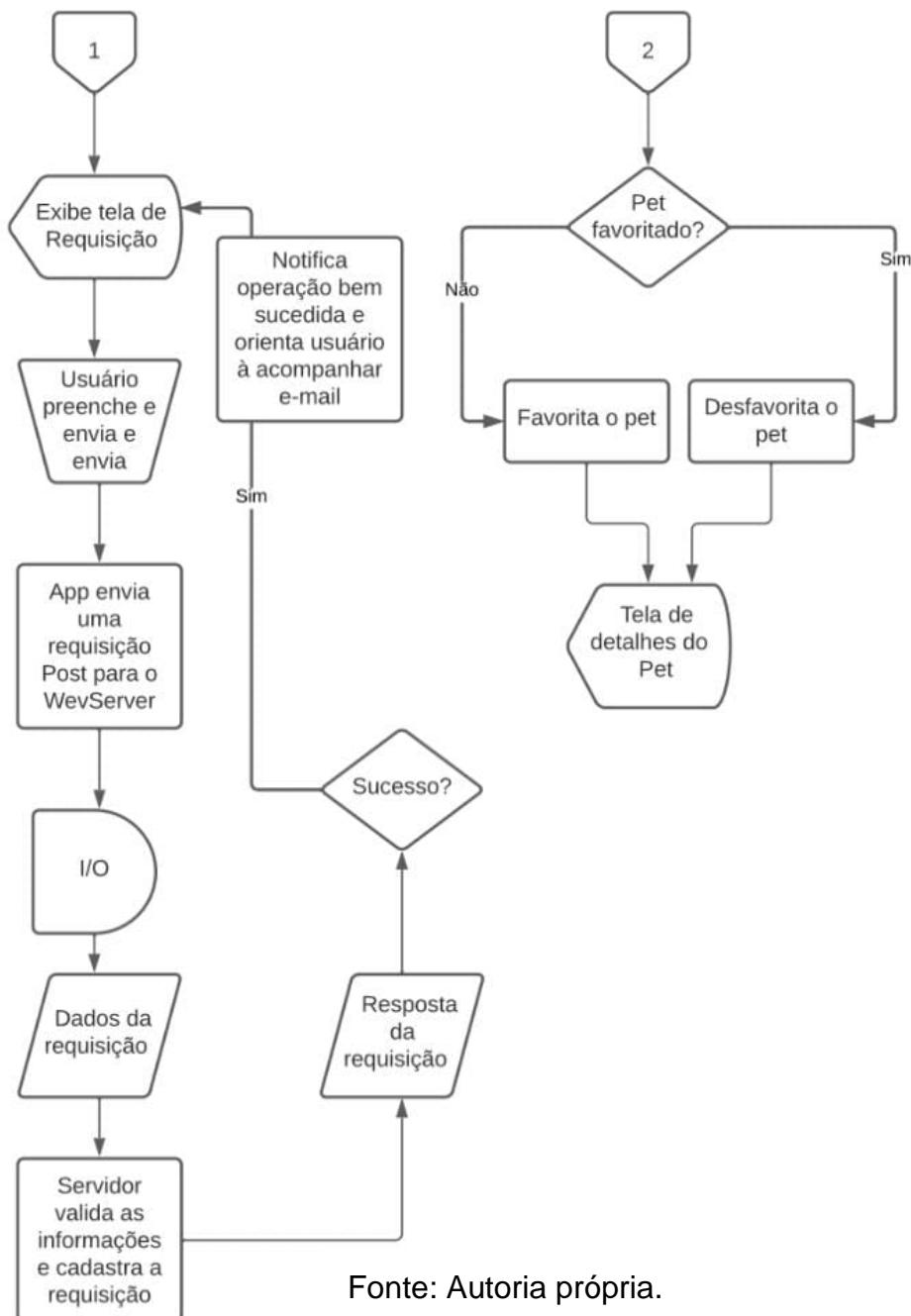
3.11.1 Aplicativo

Figura 34 - Fluxograma do app 1



Fonte: Autoria própria.

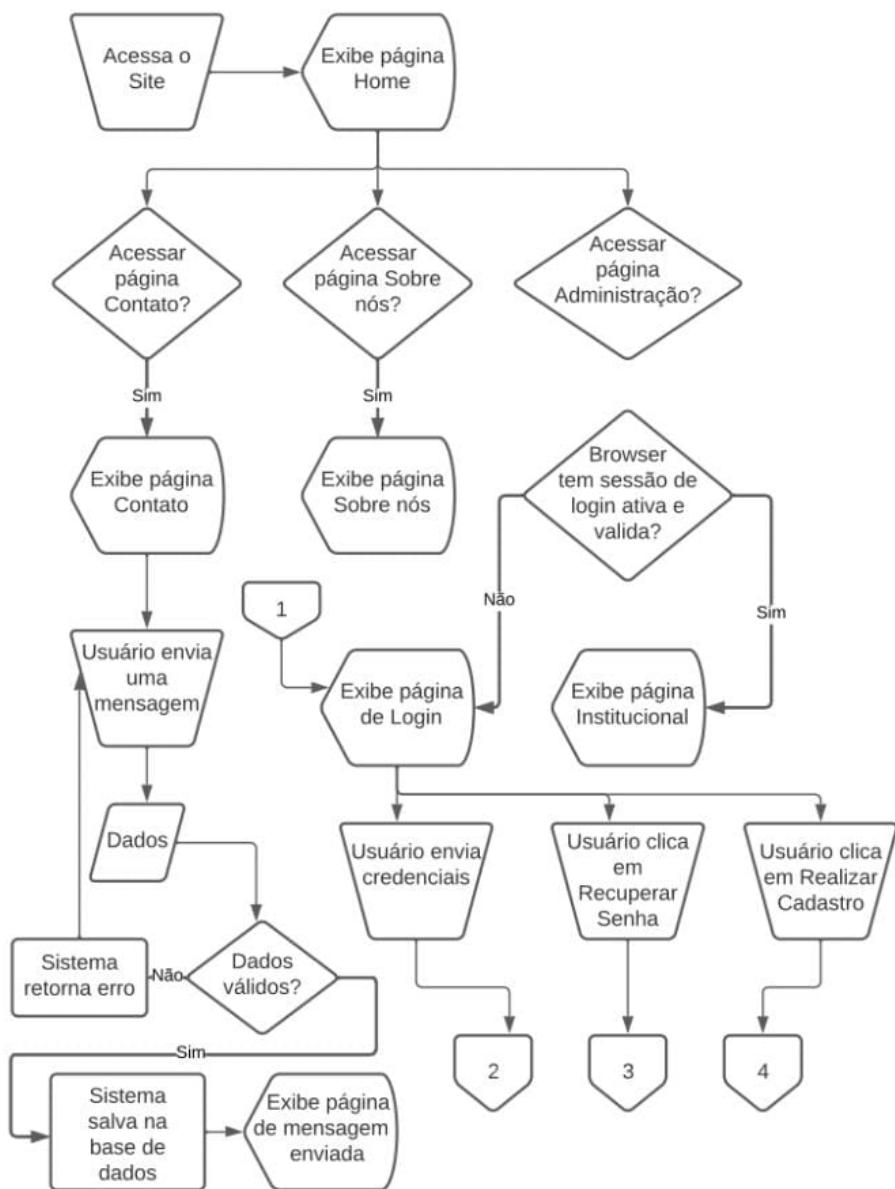
Figura 35 - Fluxograma do app 2



Fonte: Autoria própria.

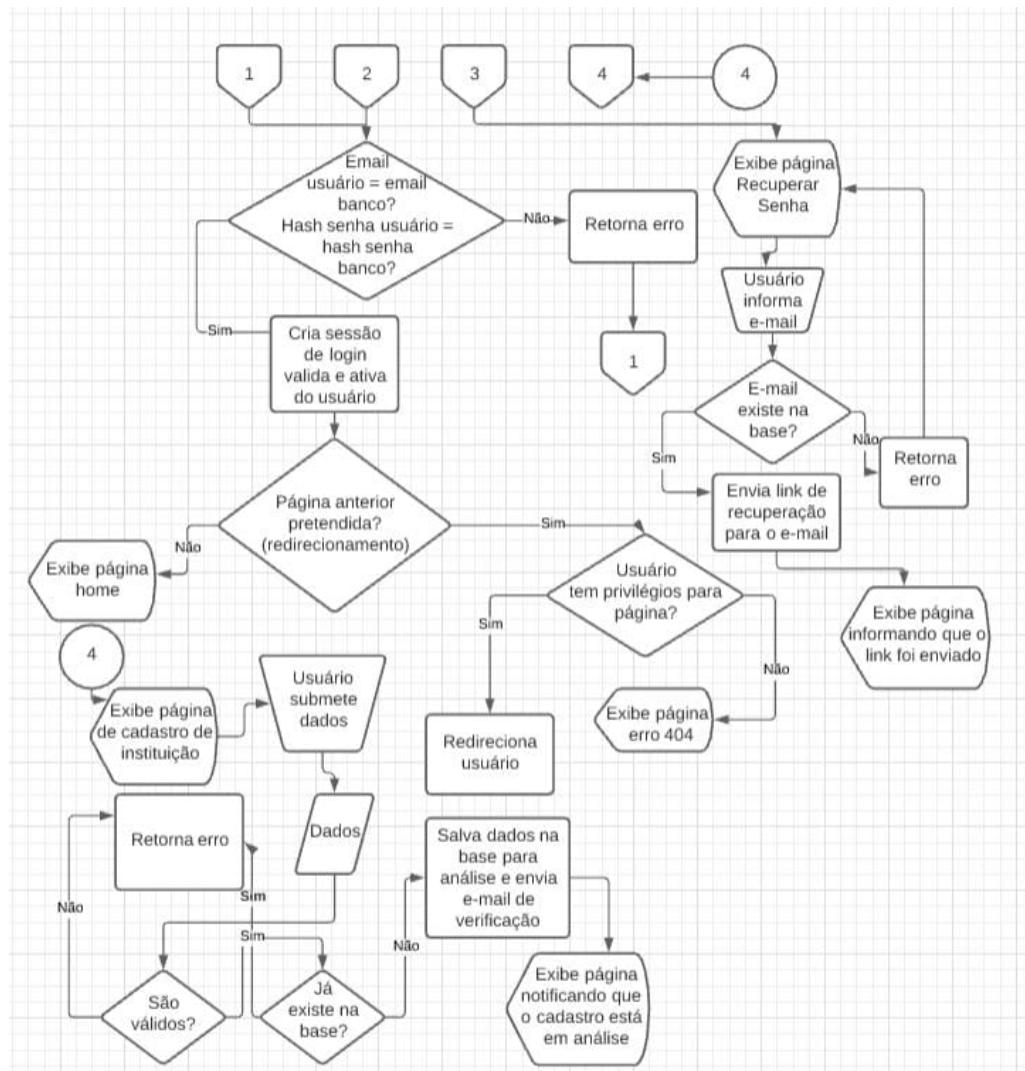
3.11.2. Site

Figura 36 - Fluxograma do site



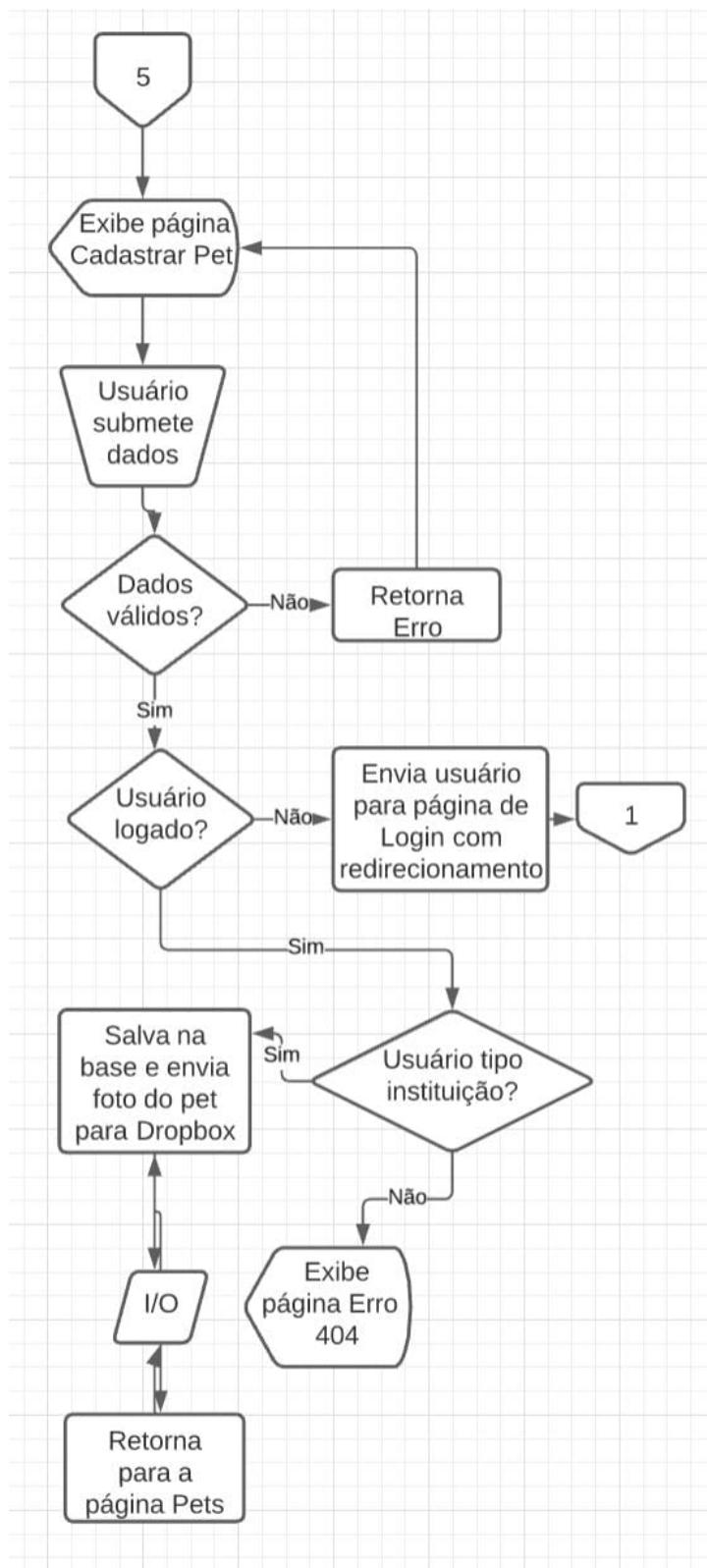
Fonte: Autoria própria.

Figura 37 - Fluxograma do site 2



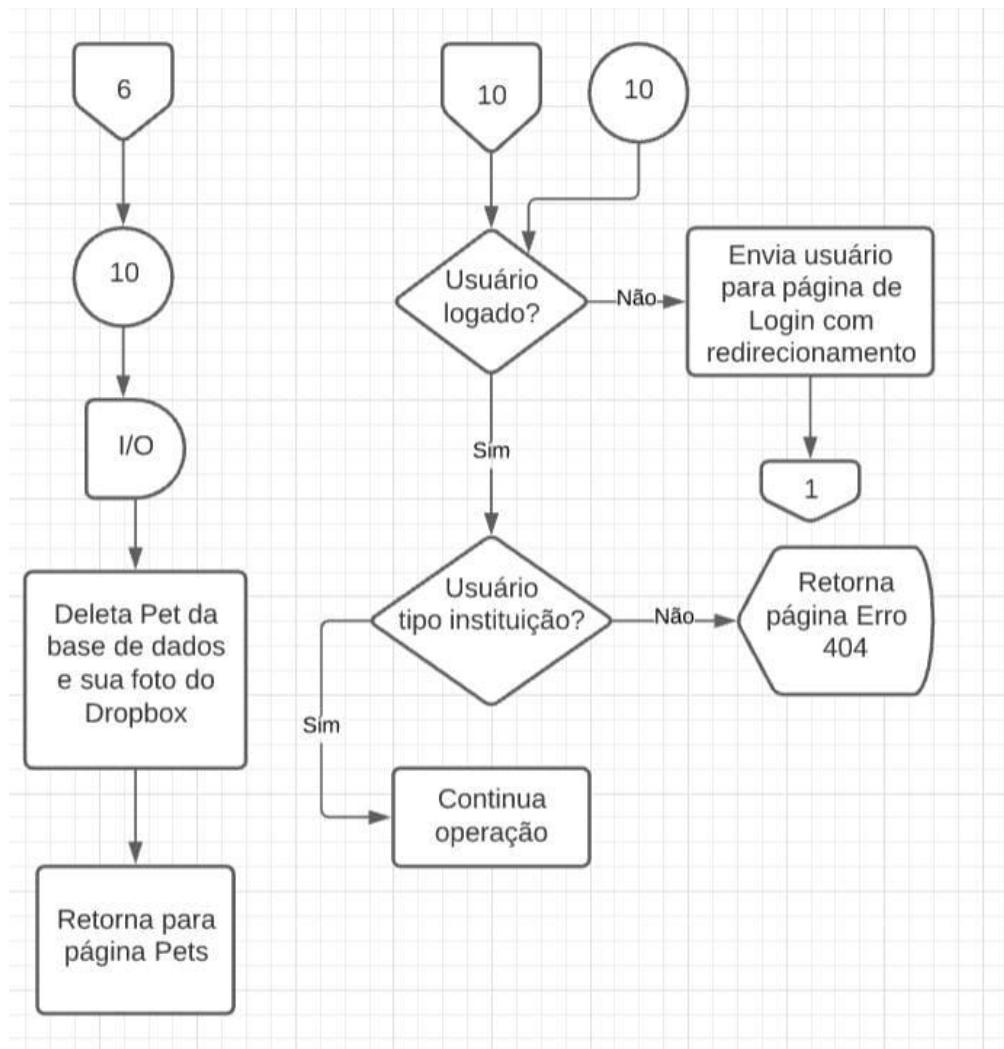
Fonte: Autoria própria.

Figura 38 - Fluxograma do site 3



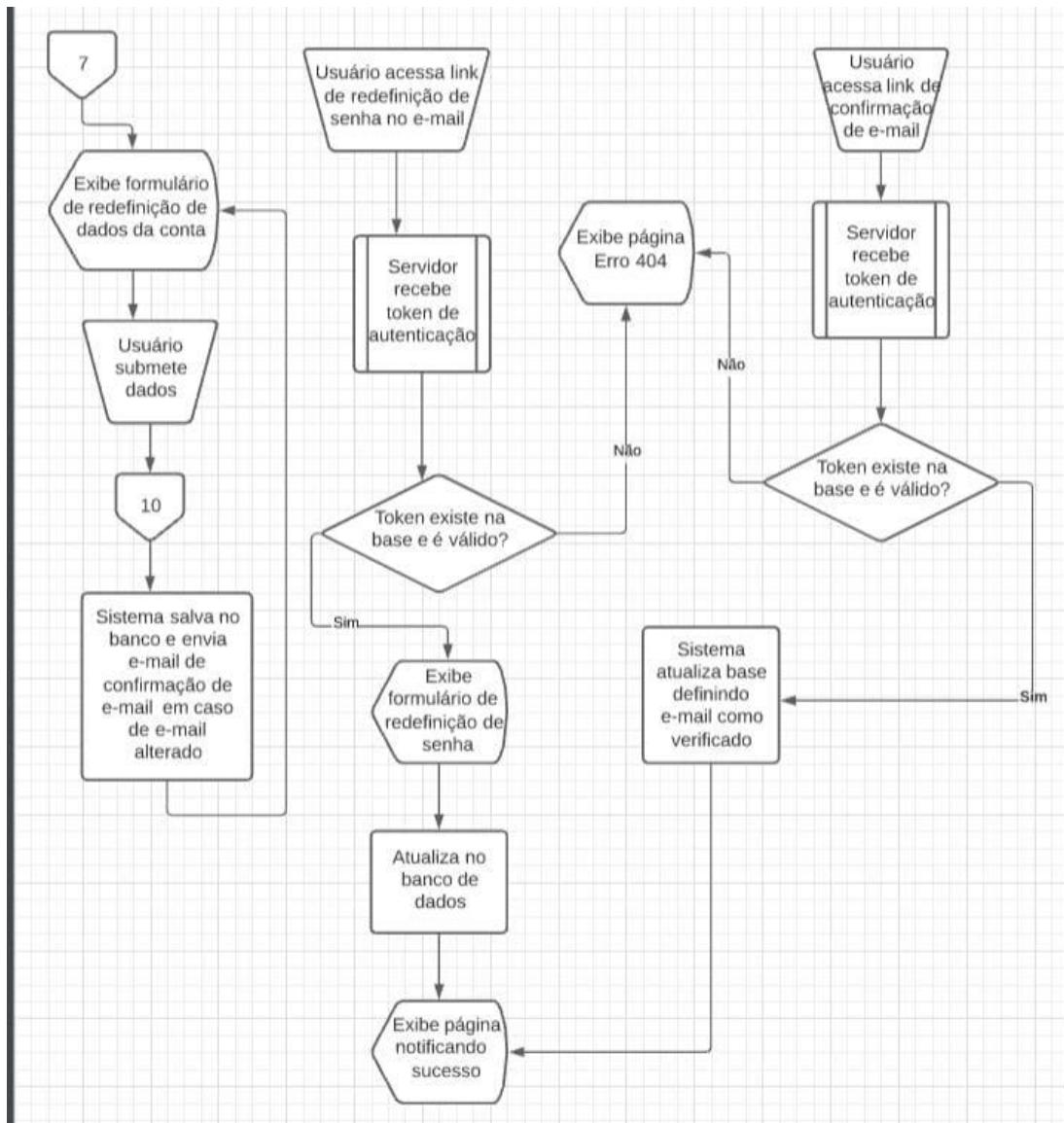
Fonte: Autoria própria.

Figura 39 - Fluxograma do site 4



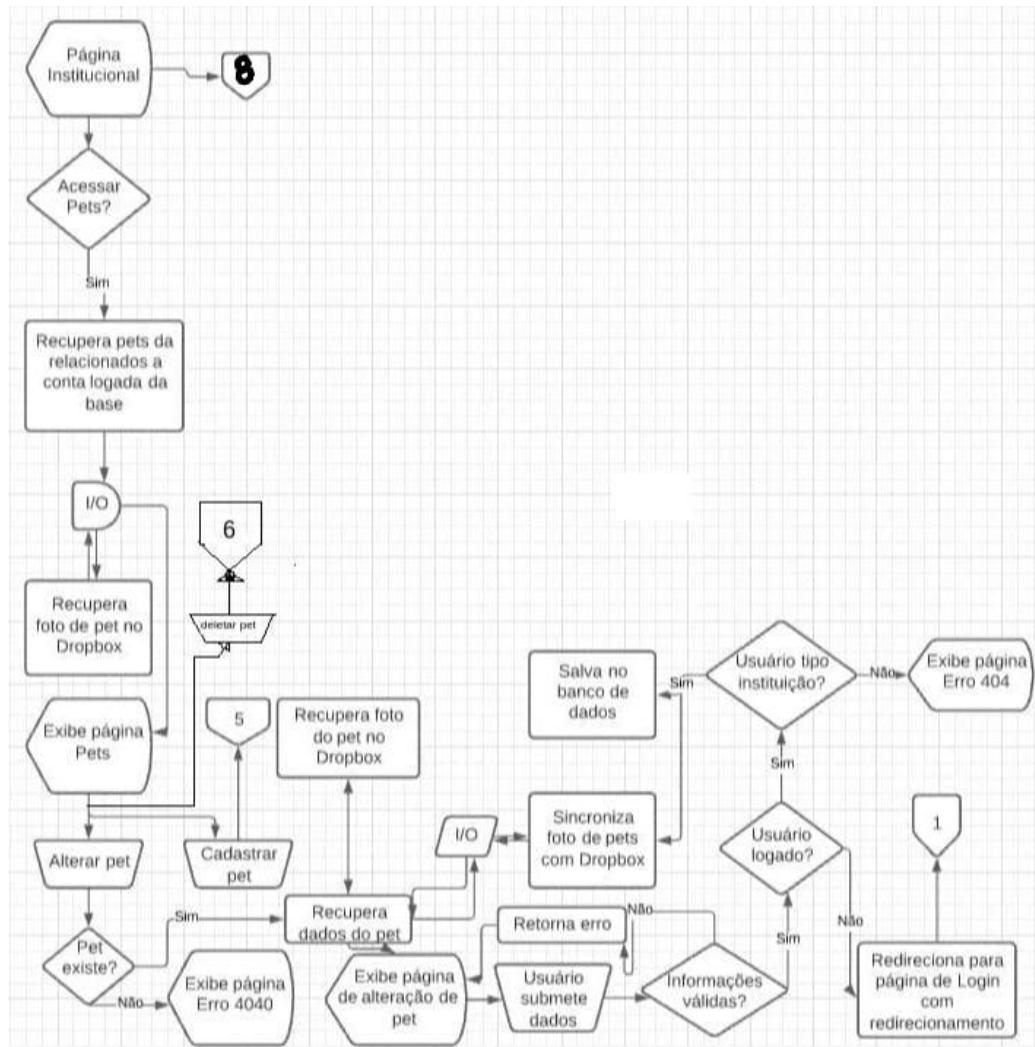
Fonte: Autoria própria.

Figura 40 - Fluxograma do site 5



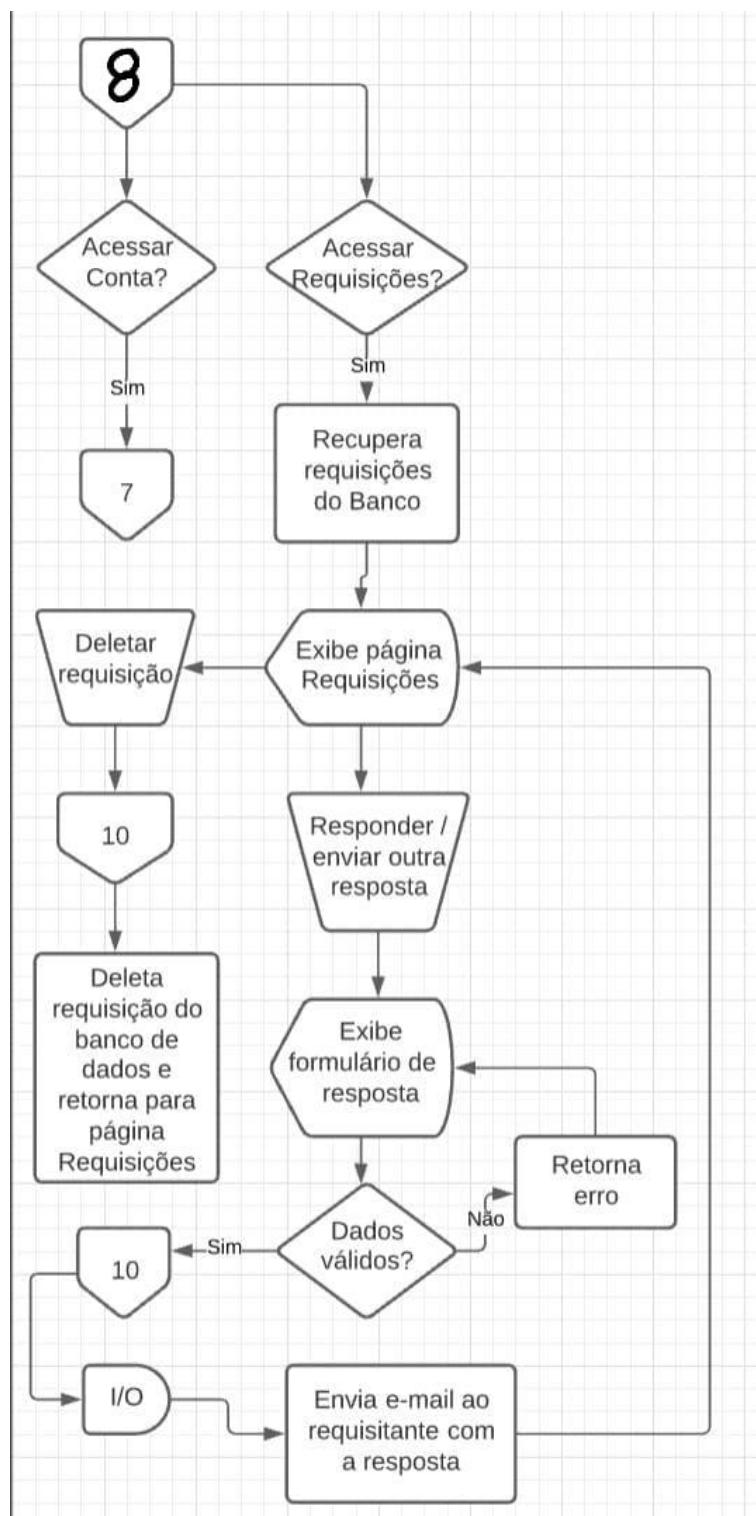
Fonte: Autoria própria.

Figura 41 - Fluxograma 6



Fonte: Autoria própria.

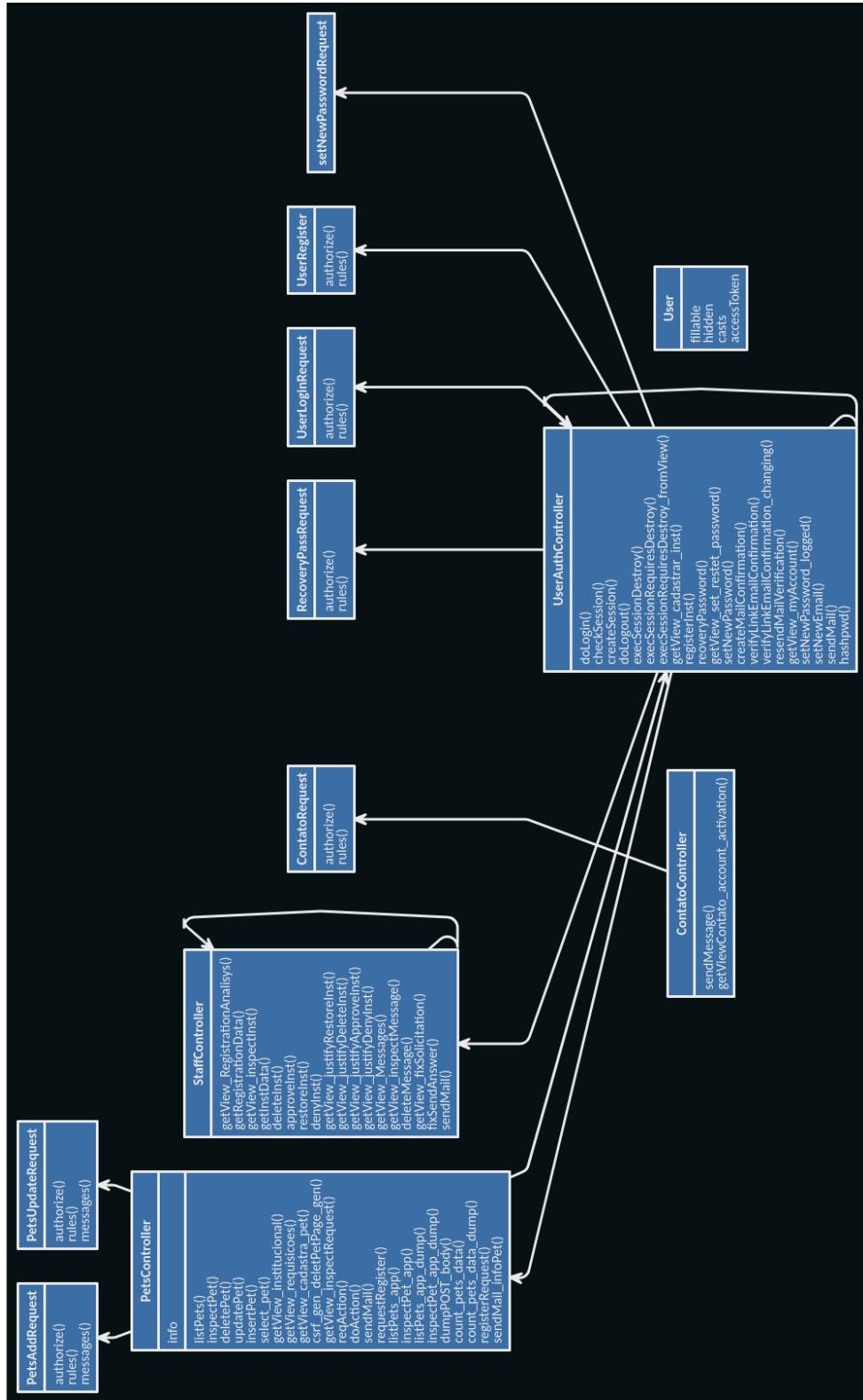
Figura 42 - Fluxograma 7



Fonte: Autoria própria.

3.12. Diagrama de classes

Figura 43 - Diagrama de Classes



4. DESENVOLVIMENTO

4.1. Aplicativo

4.1.1. Fragment_fav.java

```
package com.example.amicacina;
```

```
//Realiza as importações de bibliotecas necessárias para a execução de  
métodos da classe.
```

```
import android.annotation.SuppressLint;
```

```
import android.content.Intent;
```

```
import android.database.Cursor;
```

```
import android.os.Bundle;
```

```
import androidx.fragment.app.Fragment;
```

```
import android.view.LayoutInflater;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.widget.AdapterView;
```

```
import android.widget.GridView;
```

```
import android.widget.TextView;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class fragment_fav extends Fragment {
```

```
    GridView gridviewFav;
```

```
    TextView txtFav;
```

```
    public fragment_fav() {
```

```
        // Required empty public constructor
```

```
}
```

```
@SuppressLint("Range", "MissingInflatedId")  
@Override  
  
//Método de inicialização da classe.  
public View onCreateView(LayoutInflater inflater, ViewGroup container,  
    Bundle savedInstanceState) {  
  
    //Instancia o fragment de favoritos.  
    View view = inflater.inflate(R.layout.fragment_fav, container, false);  
  
    //Instancia widgets, objetos, arraylist e adapters para a alimentação do  
    fragment.  
    gridViewFav = (GridView) view.findViewById(R.id.GridViewFav);  
    txtFav = (TextView) view.findViewById(R.id.txtFavo);  
    ArrayList<GridModel> gridModelArrayList = new  
    ArrayList<GridModel>();  
    GridAdapter adapter = new GridAdapter(getContext(),  
    gridModelArrayList);  
  
    //Instancia a classe de Controller do banco de dados como objeto e  
    realiza a busca dos pets favoritados.  
    DatabaseController db = new DatabaseController(getContext());  
    Cursor cursor = db.retrieveFavPets();  
  
    if(cursor.getCount()==0){  
        MainActivity.null_fav=true;  
    }  
    else{  
        MainActivity.null_fav=false;  
    }  
  
    //Alimenta o adapter com os dados do banco de dados.
```

```
for(int i=0; i<cursor.getCount(); i++) {  
    gridModelArrayList.add(new GridModel(  
        cursor.getString(cursor.getColumnIndex("id")),
```

4.1.2.Fragment_home.java

```
package com.example.amicacina;
```

```
//Realiza as importações de bibliotecas necessárias para a execução de  
métodos da classe.
```

```
import android.annotation.SuppressLint;  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;
```

```
import androidx.fragment.app.Fragment;
```

```
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.GridView;  
import android.widget.TextView;  
import android.widget.Toast;
```

```
import com.google.gson.JsonArray;  
import com.google.gson.JsonObject;  
import com.koushikdutta.async.future.FutureCallback;  
import com.koushikdutta.ion.Ion;  
  
import java.util.ArrayList;
```

```
public class fragment_home extends Fragment {
```

```

GridView gridViewHome;

public fragment_home() {
    // Required empty public constructor
}

@SuppressLint("Range")
@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_home, container, false);

    // Instanciando o gridView com o elemento gráfico
    gridViewHome = (GridView) view.findViewById(R.id.gridViewHome);

    // Criação de uma lista de arrays, utilizando como modelo a classe
    GridModel, onde foi
    // determinada as variáveis necessárias e suas devidas posições
    ArrayList<GridModel> gridModelArrayList = new
    ArrayList<GridModel>();

    DatabaseController db = new DatabaseController(getContext());
    Cursor cursor = db.retrieveData();
    Log.d("data", String.valueOf(cursor.getCount()));
    for(int i=0 ; i < cursor.getCount(); i++){
        gridModelArrayList.add(new GridModel(
            cursor.getString(cursor.getColumnIndex("id")),
            cursor.getString(cursor.getColumnIndex("nome")),
            cursor.getString(cursor.getColumnIndex("raca")),
            cursor.getString(cursor.getColumnIndex("nascimento"))
        ));
    }
}

```

4.1.3.GridAdapter.java

```

package com.example.amicacina;

//Realiza as importações de bibliotecas necessárias para a execução de
métodos da classe.

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import com.squareup.picasso.Picasso;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import java.util.ArrayList;

public class GridAdapter extends ArrayAdapter<GridModel> {
    // Criando as variáveis para colocar as informações na activity_details.
    public static String nome,raca,nasc,idad,stat,gene,port,comp;

    // Método requerido para usar o modelo da grid
    public GridAdapter(@NonNull Context context, ArrayList<GridModel>
gridModelArrayList) {
        super(context, 0, gridModelArrayList);
    }

    //Método que retorna que preenche os elementos / widget com os dados
do adapter.

    @NonNull
    @Override

```

```

public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
    View listItemView = convertView;
    // Se não tiver os dados, ele preenche a view com os dados
    if (listItemView == null) {

        // Infla o layout para mostrar os items no GridView
        listItemView =
LayoutInflater.from(getContext()).inflate(R.layout.card_item, parent, false);
    }
    // Objeto do modelo da grid recebe o valor contido na posição definida
no grid
    GridModel gridModel = getItem(position);

    // Instanciando os campos de informações (textos e a foto)
    TextView nameTV = listItemView.findViewById(R.id.idTVname);
    TextView racaTV= listItemView.findViewById(R.id.idTVraca);
    TextView nascTV= listItemView.findViewById(R.id.idTVnasc);
    TextView idadTV= listItemView.findViewById(R.id.idTVidad);
    TextView statTV= listItemView.findViewById(R.id.idTVstat);
    TextView geneTV= listItemView.findViewById(R.id.idTVgene);
    TextView portTV= listItemView.findViewById(R.id.idTVport);
    TextView compTV= listItemView.findViewById(R.id.idTVcomp);
    ImageView petIV = listItemView.findViewById(R.id.GVimg);

    // Usa métodos existentes no GridModel para pegar os dados do banco
de dados e os posicionar em seus devidos campos
    nameTV.setText(gridModel.getpet_name());
    racaTV.setText(gridModel.getPet_raca());
    nascTV.setText(gridModel.getPet_nasc());
    idadTV.setText(gridModel.getPet_idad());
    statTV.setText(gridModel.getPet_stat());
    geneTV.setText(gridModel.getPet_gene());
    portTV.setText(gridModel.getPet_port());
}

```

```

        compTV.setText(gridModel.getPet_comp());

        // Usa a biblioteca Picasso para recuperar a imagem da web através da
        url do pet no banco de dados.

        Picasso.get().load(gridModel.getImgid()).into(petIV);

        return listItemView;
    }
}

```

4.1.4.GridModel.java

```

package com.example.amicacina;

public class GridModel {

    // Criando variáveis para armazenar os dados do pet
    private String pet_id, pet_name, pet_raca, pet_nasc, pet_idad, pet_stat,
    pet_gene, pet_port, pet_comp, imgid;

    // Instanciando as variáveis na ordem que deve ser preenchida, serve
    como um modelo onde todos os dados são incluidos
    // na exata ordem, como em um vetor
    public GridModel(String pet_id, String pet_name, String pet_raca, String
    pet_nasc, String pet_idad, String pet_stat,
                      String pet_gene, String pet_port, String pet_comp, String imgid)
    {
        this.pet_id = pet_id;
        this.pet_name = pet_name;
        this.pet_raca = pet_raca;
        this.pet_nasc = pet_nasc;
        this.pet_idad = pet_idad;
        this.pet_stat = pet_stat;
        this.pet_gene = pet_gene;
        this.pet_port = pet_port;
        this.pet_comp = pet_comp;
    }
}

```

```
    this.imgur_id = imgid;
}

// Métodos getters e setters, utilizado para proteger os dados
public String getpet_id() {
    return pet_id;
}

public void setpet_id(String id) {
    this.pet_id = id;
}

public String getpet_name() {
    return pet_name;
}

public void setpet_name(String pet_name) {
    this.pet_name = pet_name;
}

public String getImgid() {
    return imgid;
}

public void setImgid(String imgid) {
    this.imgur_id = imgid;
}

public String getPet_raca() {
    return pet_raca;
}

public void setPet_raca(String pet_raca) {
    this.pet_raca = pet_raca;
}
```

```
}
```

4.1.5.MainActivity.java

```
package com.example.amicacina;
```

```
//Realiza as importações de bibliotecas necessárias para a execução de  
métodos da classe.
```

```
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBar;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.fragment.app.Fragment;
```

```
import android.content.Context;  
import android.content.Intent;  
import android.database.SQLException;  
import android.database.sqlite.SQLiteDatabase;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;
```

```
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.gson.JsonArray;  
import com.google.gson.JsonNull;  
import com.google.gson.JsonObject;  
import com.koushikdutta.async.future.FutureCallback;
```

```

import com.koushikdutta.ion.Ion;

//Activity Principal, classe executada ao inicializar o app.
public class MainActivity extends AppCompatActivity {

    public static final String GET_DATA_URL =
"https://amicao.herokuapp.com/application_retrieve/pets";
    public static Fragment selectedFragment = null;
    public static int id_pet;
    public static int pos;
    public static Boolean from_fav;
    public static Boolean null_data;
    public static Boolean null_fav;
    public static Boolean internet_connection;
    public static String first_run;
}

```

//Método responsável por inicializar todas as definições de comportamento da activity toda vez em que a mesma é invocada.

@Override

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

//Oculta elemento visual desnecessário.

```

ActionBar actionBar = getSupportActionBar();
actionBar.hide();

```

//Inicializa as variáveis estáticas que definem a forma como o(s) pet(s) será(ão) buscado(s) no banco de dados.

```

null_data=false;
from_fav=false;

```

4.1.6.Activity_details.java

```
package com.example.amicacina;

//Realiza as importações de bibliotecas necessárias para a execução de
métodos da classe.

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.squareup.picasso.Picasso;

import org.w3c.dom.Text;

import java.util.Objects;

public class activity_details extends AppCompatActivity{

    ImageView imgFoto;
```

```

    TextView txtCidade, txtNome, txtRaca, txtNascimento, txtIdade, txtStatus,
    txtGenero, txtPorte, txtComportamento;
    String fav_action;
    Cursor cursor;

    @SuppressLint("Range")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);

        // Esconde a barra superior para fins estéticos
        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayHomeAsUpEnabled(true);
        actionBar.setTitle("");

        // Define ColorDrawable object and parse color
        // using parseColor method
        // with color hash code as its parameter
        ColorDrawable           colorDrawable           =           new
        ColorDrawable(Color.parseColor("#e08f60"));

        // Set BackgroundDrawable
        actionBar.setBackgroundDrawable(colorDrawable);

        // Instanciamento dos elementos responsáveis por receber os dados
        imgFoto = (ImageView) findViewById(R.id.imgFoto);
        txtNome = (TextView) findViewById(R.id.txtNome);
        txtRaca = (TextView) findViewById(R.id.txtRaca);
        txtNascimento = (TextView) findViewById(R.id.txtNascimento);
        txtIdade = (TextView) findViewById(R.id.txtIdade);
    }
}

```

4.1.7.Activity_info.java

```
package com.example.amicacina;
```

//Realiza as importações de bibliotecas necessárias para a execução de métodos da classe.

```
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import com.google.gson.JsonArray;
import com.google.gson.JsonObject;
import com.koushikdutta.async.future.FutureCallback;
import com.koushikdutta.ion.Ion;

import org.w3c.dom.Text;
```

```

import java.util.Calendar;
import java.util.Locale;

public class activity_info extends AppCompatActivity {
    //Define a URL em que o corpo POST da requisição de agendamento será
    enviada.
    public static final String REQUEST_SENDING_URL =
    "https://amicao.herokuapp.com/api/application_send/send_request";
    TextView txtMessage;

    /*
    O método abaixo inicializa a activity e oculta texto desnecessário da
    AppBar.
    */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info);

        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayHomeAsUpEnabled(true);
        actionBar.setTitle("");

    // Define ColorDrawable object and parse color
}

```

4.1.8.CreateDatabase.java

```

package com.example.amicacina;

//Realiza as importações de bibliotecas necessárias para a execução de
métodos da classe.
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

```

```

import android.util.Log;

public class CreateDatabase extends SQLiteOpenHelper {

    //Constantes que carregam as queries a serem executadas ao decorrer da
    classe, como criação do banco de dados e de suas tabelas.
    private static final String DATABASE_NAME = "db_amicao";
    private static final int DATABASE_VERSION = 1;
    public final String CREATE_TABLE = "create table if not exists tb_pets (id
    integer primary key, nome text, status text, comportamento text, endereco text,
    genero text, favoritado text, foto text, raca text, idade text, porte text, nascimento
    text);";
    public final String CREATE_TABLE_SYNC = "create table if not exists
    tb_sync(id integer primary key);";
    public final String CREATE_TABLE_SETTINGS = "create table if not
    exists tb_settings(id integer primary key, refresh_time integer);";

    //Instância a classe.
    public CreateDatabase(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    /*
    Método responsável por criar as tabelas necessárias para o banco de
    dados na primeira execução do app no
    smartphone do usuário. Bem como, tabelas de pets e de configuração do
    app.
    */
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TABLE);
        db.execSQL(CREATE_TABLE_SYNC); db.execSQL(CREATE_TABLE_SETTINGS);
    }
}

```

//O método abaixo é responsável por a atualização da estrutura de tabelas no banco de dados.

//Em caso de necessidades futuras, queries SQL podem ser atribuidas para serem executadas quanto à atualização da base.

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
}
}
```

4.2. Site

4.2.1.Classe controller

```
<?php
```

//Controller responsável por realizar operações relacionadas ao fluxo de contato.

```
namespace App\Http\Controllers; //define o escopo de acesso da classe.
```

```
use Illuminate\Http\Request; //importa classe de requisições.
```

```
use App\Http\Requests\ContatoRequest; //importa classe de validação do formulário de contato.
```

```
use Illuminate\Support\Facades\DB; //importa classe Eloquent de banco de dados.
```

```
use Symfony\Component\HttpFoundation\RequestStack; //importa classe de requisições do Simfony.
```

```
class ContatoController extends Controller //Incializa a classe e extende a classe Controller.
```

```
{
```

//Método que salva a mensagem do usuário no banco de dados para serem visualizadas pelo usuário administrador.

```
public function sendMessage(ContatoRequest $request){
```

```

//obtém dados do formulário de contato.

$fullname = $request->post('txtnome'); //adquire o nome completo
através de POST

$email = $request->post('txtemail'); //adquire o email através de POST
$message = $request->post('txtmsg'); //adquire a mensagem através de
POST

//Realiza a inserção dos dados de contato no banco de dados.

$insert = DB::insert('insert into tb_users_faq(
    fullname, email, message, solicitation_status
)
values(
    ?, ?, ?, ?
)', array($fullname, $email, $message, 'not_solved'));

//verifica o status da operação no banco de dados realizada com
Eloquent.

if($insert){
    $status = "sucess";
}
else{
    $status = "fail";
}

//retorna a view que notifica o usuário da mensagem enviada.
return view('contato.msg-sent')->with('status', $status);
}

//função que retorna a view em caso de contato do usuário para
reativação de conta institucional.

public function getViewContato_account_activation(Request $request){
    $email = $request->post('txtEmail');
}

```

```
    return view('contato')->with('info', 'account_activation_request')->with('email',
$email);
```

4.2.2.Controller

```
<?php
```

//Classe responsável por receber paramtros das rotas e redireciona-los para as views.

```
namespace App\Http\Controllers;
```

//Realiza a importação de componenetes necessários para o funcionamento da classe.

```
use Illuminate\Foundation\Auth\Access\AuthorizesRequests; //importa a classe de autorização de requisições.
```

```
use Illuminate\Foundation\Bus\DispatchesJobs; //importa classe de tarefas.
```

```
use Illuminate\Foundation\Validation\ValidatesRequests; //importa classe de validação de requisições.
```

```
use Illuminate\Routing\Controller as BaseController; //importa classe de roteamento.
```

```
class Controller extends BaseController //Inicializa a classe e extende a classe BaseController
```

```
{
```

```
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests; //define os componentes importados na classe.
```

```
}
```

4.2.3.Download App

```
<?php
```

//Controller responsável por realizar operações relacionadas ao download do app.

```

namespace App\Http\Controllers; //define o escopo de acesso da classe.

use Illuminate\Http\Request; //importa a classe de requisições.

class DownloadApp extends Controller //inicializa a classe e extende (herda)
a classe Controller

{
    //função que retorna a view / página de download do app com a URL de
    acesso ao arquivo interno.

    public function getView_download(){
        return view('download_app')->with('app_path',
url('/app_apk/amicao_app.apk'));
        //retorna a view com a url do app como parâmetro.
    }
}

```

4.2.4.PetsController

```

<?php

//Controller responsável por realizar operações de gerenciamento de pets.

namespace App\Http\Controllers; //define o escopo de acesso da classe.
(segurança)

use Illuminate\Support\Facades\DB; //importa classe de Eloquent (abstração
de banco de dados)

use Illuminate\Http\Request; //importa classe de Requisições

use Symfony\Component\HttpFoundation\RequestStack; //importa classe de
requisições do Symfony

use App\Http\Requests\PetsAddRequest; //importa classe de validação de
dados no processo de cadastramento de pets.

use App\Http\Requests\PetsUpdateRequest; //importa classe de validação
de dados no processo de atualização de pets.

use File; //importa a classe de manipulação de arquivo.

use Carbon\Carbon; //importa a classe de tempo do sistema.

```

```

use VictorRayan\DropboxRayanVrsrb\Dropbox_FileUpload; //importa a
classe de upload de arquivo para o dropbox da biblioteca da API do Dropbox.

use VictorRayan\DropboxRayanVrsrb\Dropbox_FileDeletion; //importa a
classe de deleção de arquivo da biblioteca da API do Dropbox.

use Illuminate\Support\Facades\Storage; //importa a biblioteca de
manipulação de armazenamento interno no servidor.

use VictorRayan\DropboxRayanVrsrb\Dropbox_AccessFile; //importa a
biblioteca de acesso à arquivos externos da biblioteca da API do Dropbox.

use App\Http\Controllers\UserAuthController; //importa o Controller de
Autenticação de usuário.

use DateTime; //importa a classe de manipulação de tempo em String.

use Mail; //importa a biblioteca de operações de mailing.

//Instancia a classe / Controller de Pets e extende a classe Controller.

class PetsController extends Controller
{
    public $info=""; //variável pública de log de operações.

    //método responsável por realizar a listagem de animais do banco de
dados em uma view.

    public function listPets(){

        //verifica se o usuário está autenticado e se a sessão é válida.

        if((new UserAuthController)->checkSession()){

            //verifica os privilégios do usuário logado.

            if(session('user_type') == 'inst'){

                //realiza a busca dos pets na base de dados através de Eloquent.

                $pets = DB::select('select * from tb_pets where id_org=?',
array(session('id_org')));

                //adquire path das imagens dos pets da base de dados e gera uma
URL de acesso temporário no Dropbox.

```

```

for($i=0;$i<count($pets);$i++){
    $img_path = $pets[$i]->img_path;
    $img_link = (new Dropbox_AccessFile)->getTemporaryLink($img_path);
    $pets[$i]->img_path=$img_link;
}

//gera um token CSRF para permitir a exclusão segura de pets.
$csrf_tk = (new PetsController)->csrf_gen_deletPetPage_gen();

//retorna a informação de operações realizadas com pets para a
view notificar ao usuário.
if($this->info!=null && $this->info!=""){

    //retorna uma view (página HTML + CSS + dados dinâmicos)
    para o navegador do usuário com a informação de operação realizada.
    return view('pets')->with('pets', $pets)->with('info', $this->info)->with('csrf_tk', $csrf_tk);
} else{
    //retorna uma view (página HTML + CSS + dados dinâmicos)
    para o navegador do usuário
    return view('pets')->with('pets', $pets)->with('csrf_tk', $csrf_tk);
}
else{
    //Caso o usuário logado tenha privilégios inadequados, é retornado
    uma view de erro 404.
    return view('error_404');
}

```

4.2.5.StaffController

<?php

//Classe Controller de funcionários / administradores da plataforma.

```

namespace App\Http\Controllers; //define o escopo de acesso da classe.
use Carbon\Carbon; //importa a classe de tempo Unix.

use Illuminate\Http\Request; //importa a classe de requisições.
use App\Http\Controllers\StaffController; //importa a própria classe para
invocação segura de objetos filhos.
use Illuminate\Support\Facades\DB; //importa a classe de Eloquent do banco
de dados.

use Mail; //importa a classe de operações de e-mails.

class StaffController extends Controller //inicializa a classe e extende
Controller.
{
    //função que retorna a view com os cadastros de instituições.
    public function getView_RegistrationAnalys(){

        //verifica se o usuário tem uma sessão de login ativa e válida no
        browser.
        if((new UserAuthController)->checkSession()){

            //verifica o tipo de usuário logado (se é administrador ou instituição)
            //apenas administrador tem acesso à view, instituição é redirecionado
            para view 404.
            if(session('user_type')=='staff'){

                //recupera através do Eloquent dados de registros de instituições
                da base de dados.
                $data = (new StaffController)->getRegistrationData();

                //caso não exista nenhum registro, retorna para a view com
                parametro de informação para
                //notificar o usuário da inexistência de instituições na plataforma.
                if($data!=null){

```

```

if(!empty(session('info_register_analisys'))){
    //retorna para a view com o objeto de dados, nulo.
    //com parâmetro informando a situação.
    return view('registration_analisys')->with('data', $data)-
>with('info', session('info_register_analisys'));
}

else{ //senão, apenas retorna para a view com os dados dos
registros.

    return view('registration_analisys')->with('data', $data);
}

}

else{

    //caso ocorra erro na operação do Eloquent, redireciona para
página de erro 404.

    return view('error_404');
}

}

else{

    //caso o usuário não seja administrador, é redirecionado para
página de erro 404.

    return view('error_404');
}

}

else{

    //Caso não exista sessão válida de usuário, então redireciona para
página de login com parâmetro de retorno
    //para a rota desse método.

    session(['required_route'=>'getView_RegistrationAnalisis',
'router_owner' => 'staff']);
}

```

4.2.6. UserAuthController

```

<?php

//Classe Controller de Autenticação de usuário.

namespace App\Http\Controllers; //define o escopo de acesso da classe.

use Illuminate\Http\Request; //importa a classe de requisições.

use App\Http\Requests\UserLoginRequest; //importa classe de validação do formulário de login.

use App\Http\Controllers\UserAuthController; //importa a própria classe para invocação segura de objetos filhos.

use Carbon\Carbon; //importa classe de tempo Unix.

use Illuminate\Support\Facades\DB; //importa classe do Eloquent ORM framework. (abstração de banco de dados)

use App\Http\Controllers\PetsController; //importa classe Controller de Pets.

use App\Http\Controllers\StaffController; //importa classe Controller de funcionários / administradores.

use App\Http\Requests\UserRegister; //importa classe de validação do formulário de cadastro de instituições.

use App\Http\Requests\RecoveryPassRequest; //importa classe de validação do formulário de recuperação de senha.

use App\Http\Requests\setNewPasswordRequest; //importa classe de validação do formulário de definição de nova senha.

use Mail; //importa classe de operações com e-mail.

//inicializa a classe de Controller do usuário e extende a classe Controller.

class UserAuthController extends Controller

{
    //método que realiza a login.

    public function doLogin(UserLoginRequest $request){

        $user_email = $request->post('txtEmail'); //adquire o e-mail informado pelo usuário via POST.
    }
}

```

```

$user_password = $request->post('txtPassword'); //adquire a senha
informada pelo usuário via POST.

$secret = "30/07/2003"; //segredo do servidor utilizado como salt em
função de hash.

$user_password_hash = hash('sha256', $user_password.$secret);
//gera hash da senha do usuário com o salt anterior.

$timestamp = Carbon::now()->timestamp; //importa timestamp.

(new UserAuthController)->execSessionDestroy(); //executa método de
destruição de sessões de usuário anteriores do browser e do banco.

//verifica se o hash da senha informada pelo usuário realiza match com
o hash da senha relacionada com o e-mail também informado, do banco.

$checkCredentials = DB::select('select * from tb_auth_org where
email=? and password=?',
array($user_email, $user_password_hash));

//adquire informações da conta para criar a sessão do usuário.

$accounts_data = DB::select('select id_org, deletion_date from
tb_auth_org where status=?', array('deleted'));

//verifica se a operação acima foi bem sucedida.

if($accounts_data){

```

5. PLANO DE NEGÓCIO(CANVAS)

Figura 44 - Plano de negócio Canvas

Business Model Canvas - Amicão



Fonte: Autoria própria.

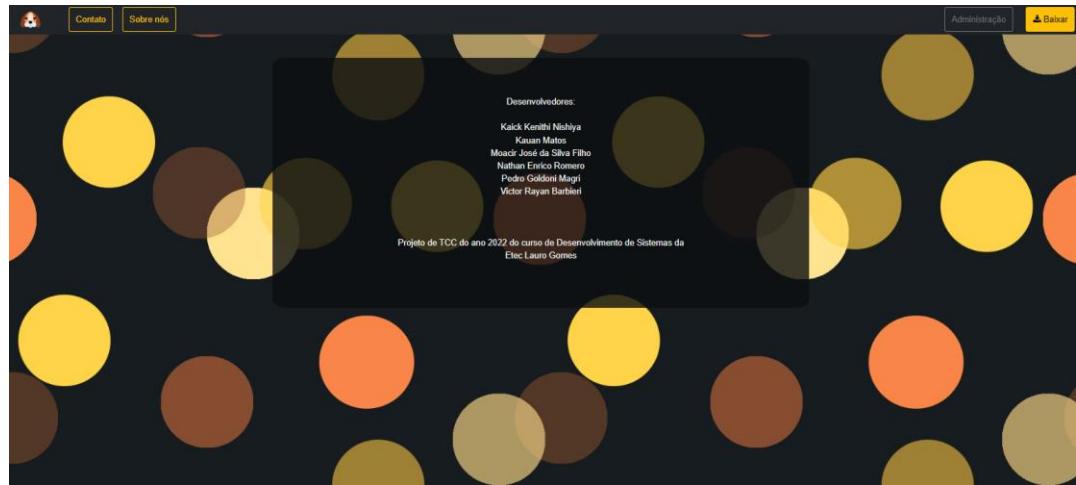
6. MANUAIS

6.1. Manual de usuário do site

O usuário acessa o site por meio do link “<https://amicao.herokuapp.com/>”. Nesta página inicial, ele pode obter acesso a outras páginas do site, como a de “sobre nós” e a de “download”, assim como a aba de administração, de modo que para que tal feito seja realizado basta que esse clique no ícone referente ao item desejado.

6.1.1. Pagina Sobre Nós

Figura 45 - Manual Pagina Sobre Nós

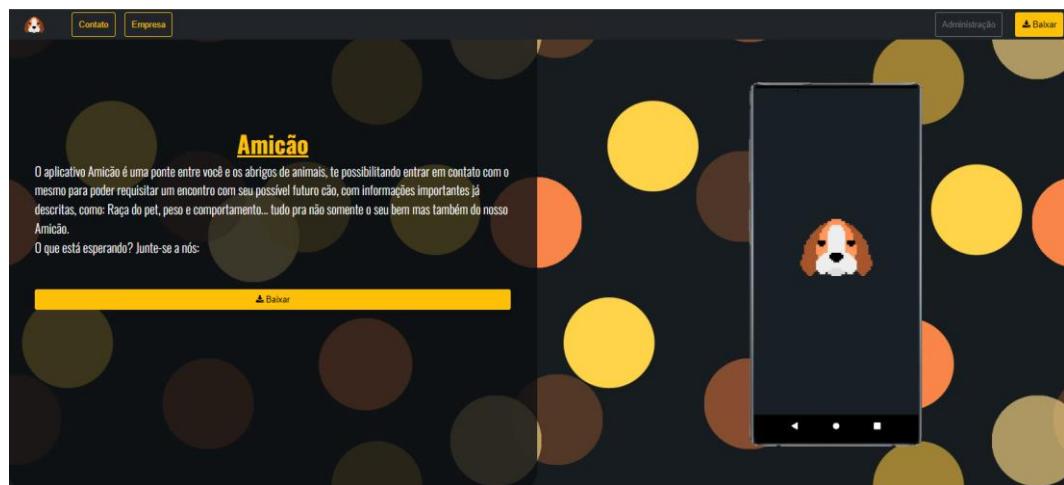


Fonte: Autoria própria.

Nessa pagina o usuario poderá analisar informações sobre os criadores do sistema tal como a sua concepção.

6.1.2. Página de download

Figura 46 - Manual Pagina Download

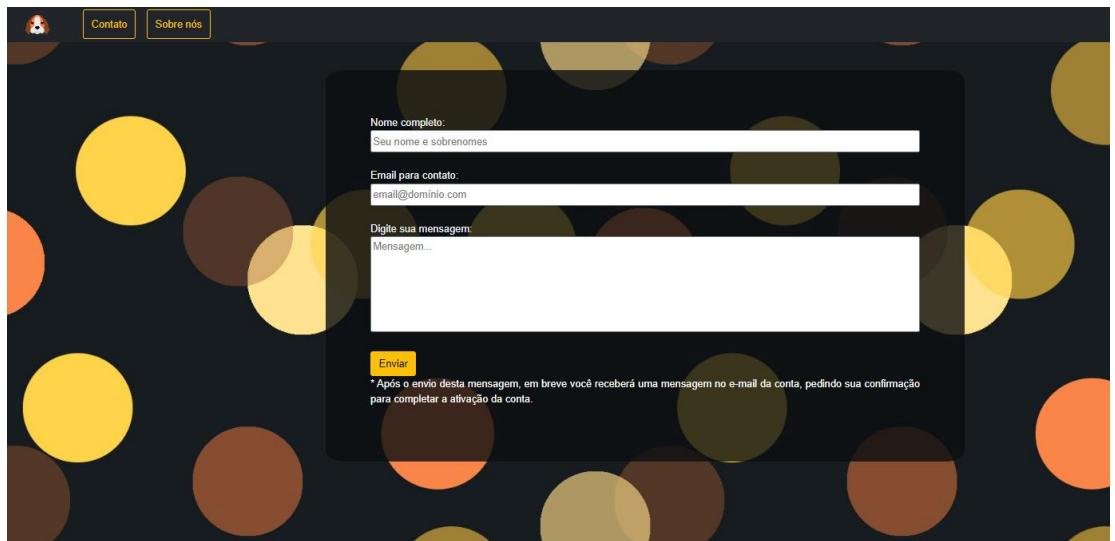


Fonte: Autoria própria.

Essa pagina podera ser acessada quando o usuario clica na opção de “baixar”, onde assim este executa a operaçao de fazer o download do pacote que contem o app Amicão.

6.1.3. Pagina de contato

Figura 47 - Manual página de contato



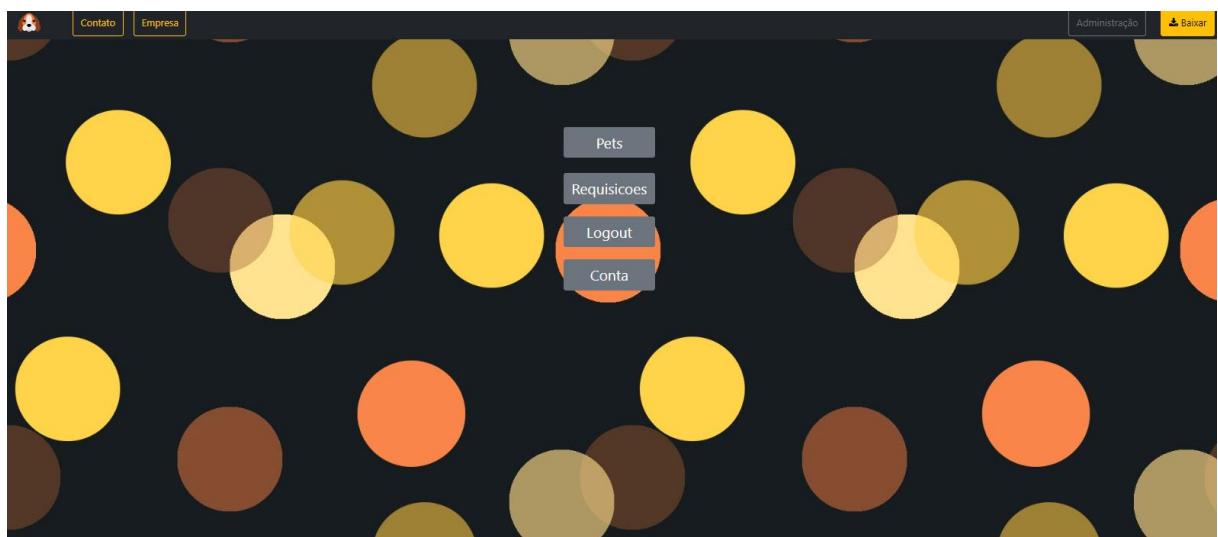
Fonte: Autoria própria.

Nessa tela o usuário deverá inserir o seu nome completo no campo “nome completo” e também o seu email no campo “email para contato” e adicionar no campo “digite sua mensagem” qualquer informação que deseje comunicar com a equipe de suporte do sistema.

6.1.4. Página de Administração

Essa página pode ser utilizada selecionando o botão “Administração” na tela principal, cujo direciona o usuário a tela que permite registrar uma instituição como fornecedora de animais para a adoção, ou seja, é uma página para ser usada somente pela equipe de administração das intuições que cuidam dos animais.

Figura 48 - Manual Pagina Administrativa



Fonte: Autoria própria.

Nessa página o administrador deverá selecionar a opção ‘Pets’ para registrar os animais da instituição, a opção “requisições” para verificar quais animais foram requisitados pelos usuários, a opção “logout” para cancelar o login e sair do modo administrador, e a opção “conta” para atualizar os dados cadastrais da instituição.

6.2. Manual de usuário do Aplicativo Mobile

6.2.1. Tela de listagem de animais disponíveis

Figura 49 - Manual Tela de listagem de animais



Fonte: Autoria própria.

Nesta tela o usuário terá acesso a todos os animais disponíveis para adoção listados de forma organizadas em seu ecrã, de modo que permite que ao selecionar o animal de vosso desejo lhe seja permitido acessar as informações referentes a esse elemento.

6.2.2. Tela de exibição das informações referentes aos animais

Figura 50 - Manual Tela de informação de animais

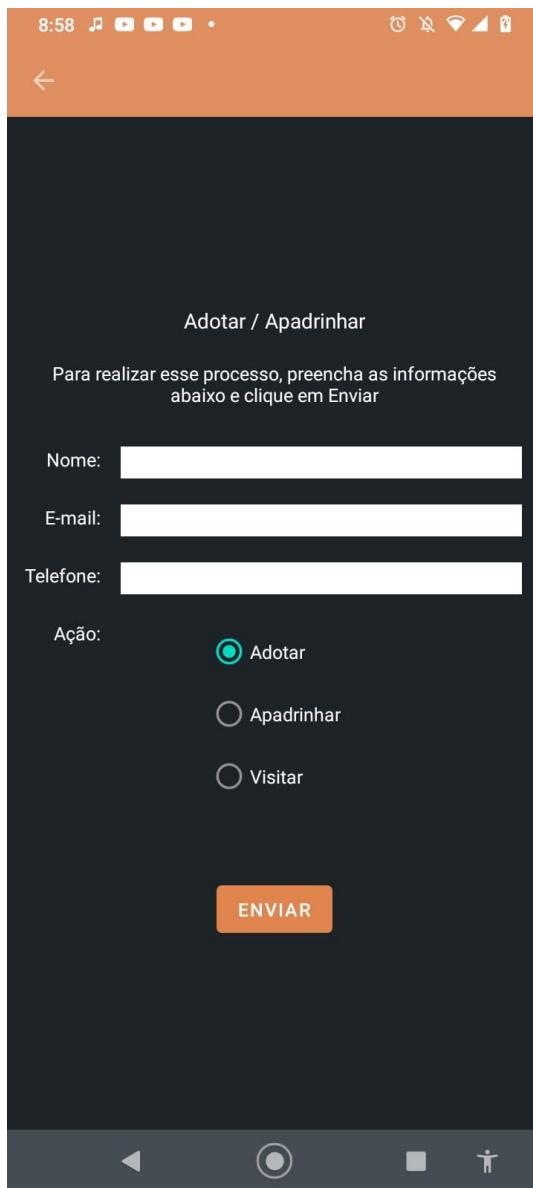


Fonte: Autoria própria.

Nessa tela há o nome dos animais e as informações de localização, idade, raça, sexo, porte físico assim como sua qualidade comportamental. Além disso ela dá ao usuário a opção de selecionar o animal como favorito através do botão “favorita”, e também permite acesso a realizar a requisição de adoção ou apadrinhamento, a aqui pode ser feita clicando no botão com essa denominação.

6.2.3. Tela de agendamento de animais

Figura 51 - Tela de agendamento de animais

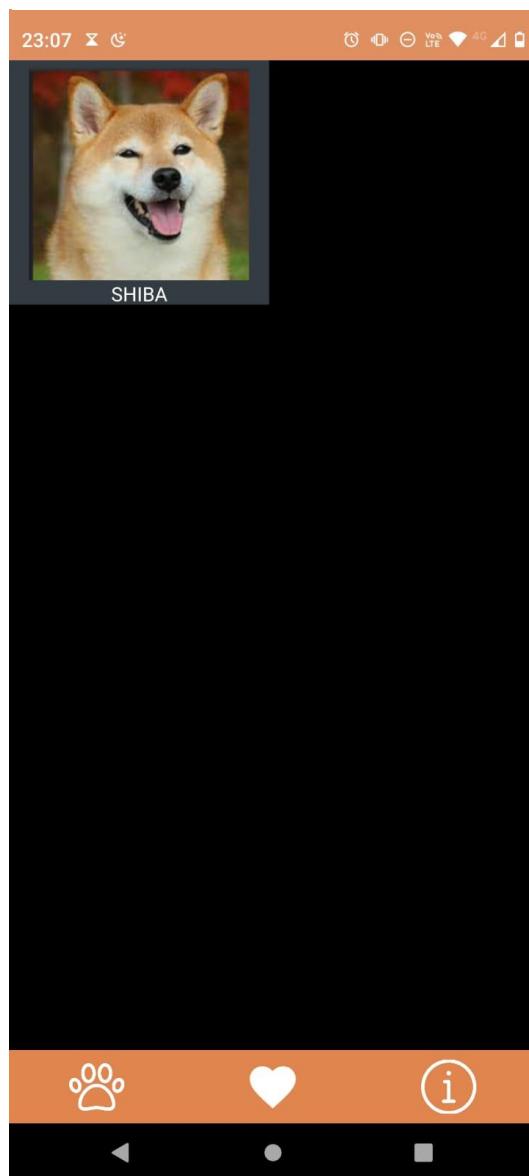


Fonte: Autoria própria.

Essa tela pode ser acessada através do botão “adotar/apadrinhar”, que está disponível na tela de informações sobre os animais. Dessa forma, nessa tela o usuário deverá inserir suas informações de contato para, e selecionar se deseja agendar uma adoção, um apadrinhamento ou uma visita para o animal selecionado, assim sendo isso pode ser realizado através do sistema de seleção de bolinhas, e depois clicando no botão de enviar, de modo que a equipe de suporte entrará em contato via email ou SMS confirmando o agendamento.

6.2.4.Tela de Animais filtrados por favoritamento

Figura 52 - Manual Tela de Favoritos



Fonte: Autoria própria.

Essa tela deverá ser acessada clicando no ícone de coração na barra “BottomNavigation”, a qual permite ao usuário visualizar os animais selecionados por ele como favoritos.

7. HORAS TRABALHADAS

7.1. Custos profissionais

| NOME | FUNÇÃO | VALOR HORA | HORAS TOTAIS | SALÁRIO |
|-----------------|---------------------------------|------------|--------------|-----------|
| KAICK KENITHI | Programador Mobile front-end | R\$40,00 | 560h | R\$22,400 |
| KAUAN MATOS | Documentação e Ciencia de Dados | R\$45,00 | 840h | R\$37,800 |
| NATHAN ROMERO | Programador Web Front-End | R\$20,00 | 560h | R\$11,200 |
| PEDRO GOLDONI | Programador Mobile Back-end | R\$74,00 | 840h | R\$62,160 |
| MOACIR JOSÉ | Designer de UI | R\$75,00 | 120h | R\$5,600 |
| VICTOR BARBIERI | Programador Web Back-end | R\$40,00 | 840h | R\$36,600 |
| TOTAL | R\$175.760 | | | |

7.2. Custo das ferramentas

| Ferramentas | Valor (por mês) |
|--------------------|------------------|
| Docker | Gratuito |
| Android Studio | Gratuito |
| Visual Studio Code | Gratuito |
| GitHub | Gratuito |
| Photoshop | R\$ 43,00 |
| Office 365 | R\$ 36,00 |
| ASTAH Community | Gratuito |
| Laravel | Gratuito |
| Android SDK | Gratuito |
| PostgreSQL | Gratuito |
| Apache | Gratuito |
| Nginx | Gratuito |
| PHP Composer | Gratuito |
| Heroku CLI | Gratuito |
| Total | R\$ 79,00 |

7.3. Custo total

| | |
|-----------------------------|---------------|
| Mão de Obra | R\$175.760,00 |
| Softwares e Licenças | R\$79,00 |
| Custo Total | R\$175,839,00 |

8. CONCLUSÃO

Temos que, é observável o alto índice de animais nas ruas, assim como a relação que este fenômeno tem com os aspectos econômicos da atualidade e as recentes crises sanitárias. Além disso, nota-se a clara necessidade de desenvolvimento da infraestrutura digital contemplada pelo sistema de adoção de animais idealizado, para facilitar a adoção e o apadrinhamento dos animais. Sendo assim, diante do exposto, é correto afirmar que o projeto cumpre com o objetivo estabelecido, haja vista que foi construída uma interface que permite as instituições de caridade realizarem o cadastro de seus animais, tal como permite aos usuários acessarem todos os animais disponíveis e realizarem o agendamento de visitas, apadrinhamento a adoção, de modo que o desenvolvimento desse sistema implica na redução da burocracia envolvida nas requisições de adoções facilitando o processo de forma realmente notável, assim como promove a diminuição de problemas derivados da higiene pública rudimentar referente aos animais abandonados nas áreas de habitação humana.

Conquanto, vemos a real relevância do desenvolvimento tecnológico da computação associada à manutenção da ordem social que se define pelo controle dos aspectos sanitários e da qualidade de vida dos indivíduos por meio do benefício que é ter acesso facilitado a animais para seu lazer e satisfação pessoal, além dos acréssimos econômicos gerados pela diminuição de gastos propiciados pela melhoria na eficiência do processo de relação entre os abrigos de animais e as pessoas.

9. REFERÊNCIAS

ANDRADE, Ana Paula. O que é Laravel? **Treina Web**, 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-laravel>. Acesso em: 21 abr. 2022.

CARACTERÍSTICAS e qualidades do Android Studio. **Do Linux**, s.d. Disponível em: <https://blog.desdelinux.net/pt/caracteristicas-y-cualidades-de-android-studio/>. Acesso em 21 abr. 2022.

EDSON. Introdução ao Visual Studio Code. **DevMedia**, 2016. Disonível em: <https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418>. Acesso em: 21 abr. 2022.

FERNANDES, Henrique Marques. O que é JavaScript, como funciona e para que serve? **Marques Fernandes**, 2020. Disponível em: <https://marquesfernandes.com/desenvolvimento/javascript-o-que-e-como-funciona-e-para-que-serve/>. Acesso em 21 abr. 2022.

FISCHBORN, Giovanna. **Abrigos lotados: abandono de animais cresce na pandemia**. Correio Braziliense, 2021. Disponível em: <https://www.correobraziliense.com.br/revista-do-correio/2021/08/4945175-abrigos-lotados-abandono-de-animais-cresce-na-pandemia.html>. Acesso em: 21 abr. 2022.

GUEDES, Kayobrussy. O que é e pra quê serve o Adobe Photoshop? **Top Gadget**, 2021. Disponível em: <https://www.topgadget.com.br/howto/graphs/o-que-e-e-pra-que-serve-o-adobe-photoshop.htm>. Acesso em: 21 abr. 2022.

GUEDES, Marylene. No final das contas: o que é o Docker e como ele funciona? **Treina Web**, 2019. Disponível em: <https://www.treinaweb.com.br/blog/no-final-das-contas-o-que-e-o-docker-e-como-ele-funciona>. Acesso em: 21 abr. 2022.

HANASHIRO, Akira. VS Code - O que é e por que você deve usar? **Treina Web**, 2021. Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em: 21 abr. 2022.

L, Andrei. O que é GitHub e Como Usá-lo. **Hostinger**, 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github>. Acesso em: 21 abr. 2022.

LIMA, Guilherme. Bootstrap - O que é, como e quando usar? **Alura**, 2021. Disponível em: <https://www.alura.com.br/artigos/bootstrap>. Acesso em: 21 abr. 2022.

O que é Anaconda para Python? **ByLearn**, 2020. Disponível em: <https://dojo.bylearn.com.br/python/o-que-e-anaconda-para-python/>. Acesso em: 21 abr. 2022.

O que é Git. **Atlassian**, s.d. Disponível em: <https://www.atlassian.com/br/git/tutorials/what-is-git>. Acesso em: 21 abr. 2022.

OTERO, Edgar. O que é o Word e para que serve. **Malavida**, 2022. Disponível em: <https://www.malavida.com.br/soft/microsoft-word/q/para-que-serve-o-word.html#gref>. Acesso em: 21 abr. 2022.

PACIEVITCH, Yuri. Cascading Style Sheets (CSS). **Info Escola**, 2022. Disponível em: <https://www.infoescola.com/informatica/cascading-style-sheets-css/>. Acesso em: 21 abr. 2022.

PACIEVITCH, Yuri. HTML. **Info Escola**, 2022. Disponível em: <https://www.infoescola.com/informatica/html/>. Acesso em 21 abr. 2022.

PACIEVITCH, Yuri. História do Java. **Info Escola**, 2022. Disponível em: <https://www.infoescola.com/informatica/historia-do-java/>. Acesso em 21 abr. 2022.

POWERPOINT: tudo o que você precisa saber e entender está aqui. **Smartalk**, 2021. Disponível em: <https://smartalk.com.br/blog/powerpoint/>. Acesso em: 21 abr. 2022.

SAIBA mais como o python surgiu e qual o seu cenário atual. **Vulpi**, 2018. Disponível em: <https://blog.vulpi.com.br/python-como-surgiu/>. Acesso em: 21 abr. 2022.

SCUDERO, Erick. Os 6 frameworks front-end mais amados no mundo (segundo o GitHub). **Becode**, 2018. Disponível em: <https://becode.com.br/frameworks-front-end-mais-amados-segundo-github/>. Acesso em: 21 abr. 2022.

SEABRA, Bruno. O que é Astah? **WorkStars**, 2021. Disponível em: <https://workstars.com.br/tie-business/guest-post/o-que-e-o-astah-posttecnico-por-bruno-seabra/>. Acesso em: 21 abr. 2022.

SOUZA, Ivan. PostgreSQL: saiba o que é, para que serve e como instalar. **Rockcontent**, 2020. Disponível em:<https://rockcontent.com.br/blog/postgresql/>. Acesso em: 21 abr. 2022.

TECNOLOGIA PostgreSQL. **DevMedia**, s.d. Disponível em: <https://www.devmedia.com.br/guia/tecnologia-postgresql/34328>. Acesso em 21 abr. 2022.

TUTORIAL de Android Studio. **DevMedia**, s.d. Disponível em: <https://www.devmedia.com.br/tutorial-de-android-studio/34003>. Acesso em: 21 abr. 2022.