

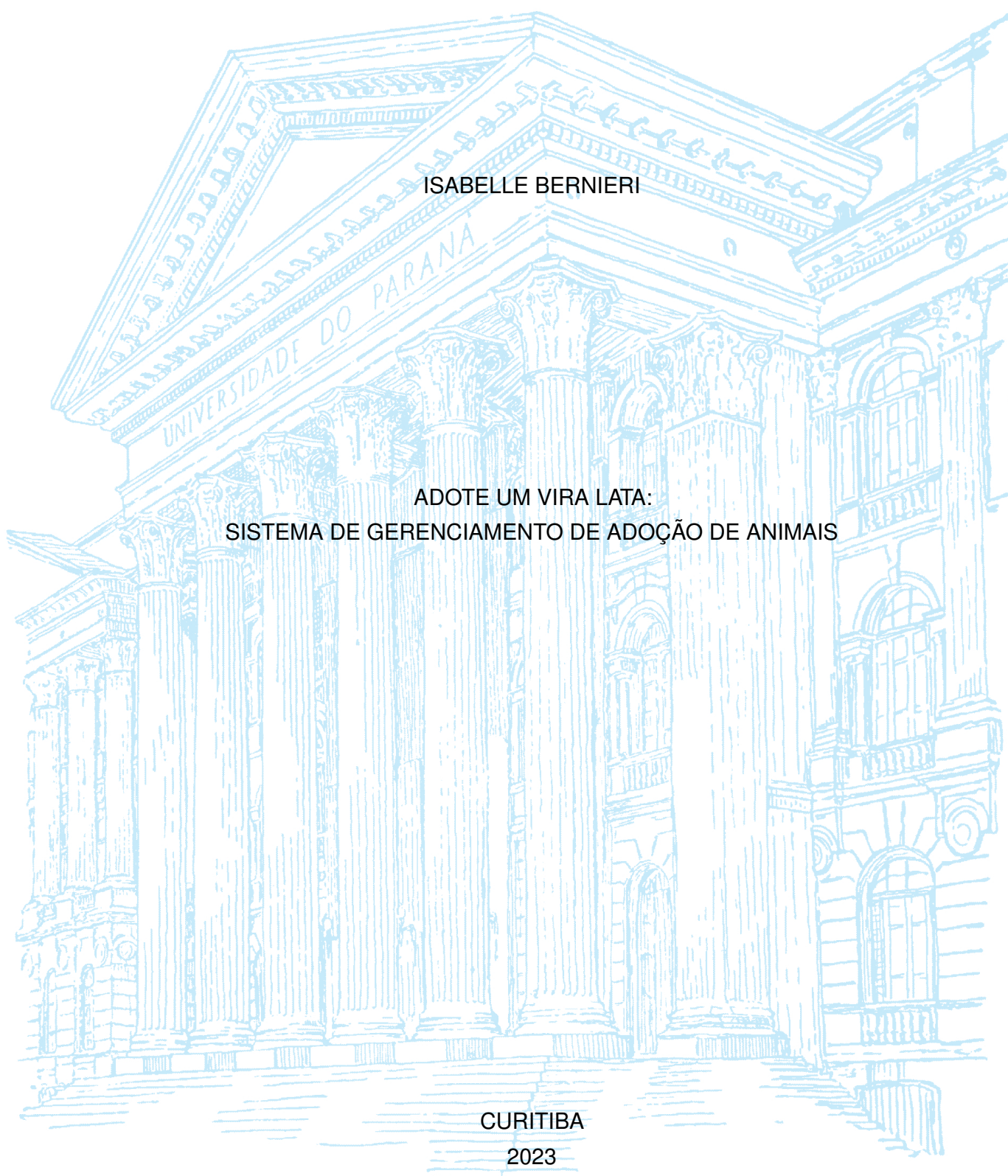
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ISABELLE BERNIERI

ADOTE UM VIRA LATA:
SISTEMA DE GERENCIAMENTO DE ADOÇÃO DE ANIMAIS

CURITIBA

2023



ISABELLE BERNIERI

ADOTE UM VIRA LATA:
SISTEMA DE GERENCIAMENTO DE ADOÇÃO DE ANIMAIS

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Alexander Robert Kutzke

CURITIBA

2023



UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR, CEP 81520-260
Telefone: 3360-5000 - <http://www.ufpr.br/>

Ata de Reunião

TERMO DE APROVAÇÃO

ISABELLE BERNIERI

ADOTE UM VIRA LATA: SISTEMA DE GERENCIAMENTO DE ADOÇÃO DE ANIMAIS

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Dr. Alexander Robert Kutzke

Orientador – SEPT/UFPR

Prof. Dr. João Eugênio Marynowski

SEPT/UFPR

Prof. Dr. Luiz Antonio Pereira Neves

SEPT/UFPR

Curitiba, 6 de dezembro de 2023.



Documento assinado eletronicamente por **ALEXANDER ROBERT KUTZKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 06/12/2023, às 18:53, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **JOAO EUGENIO MARYNOWSKI, PROFESSOR DO MAGISTERIO SUPERIOR**, em 06/12/2023, às 18:56, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **LUIZ ANTONIO PEREIRA NEVES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 11/12/2023, às 09:53, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **6253203** e o código CRC **E0554023**.

Agradecimentos

Ao professor Alex e a todos os familiares, amigos e companheiros que me deram suporte durante esta jornada.

Resumo

A presença de animais de estimação nos lares brasileiros está em constante crescimento. O Brasil possui a quarta maior população de animais de estimação no mundo. No entanto, é preocupante o número de animais abandonados registrados no país, que atualmente chega a 30 milhões. Para combater essa negligência, a cultura da adoção tem sido promovida como uma medida efetiva. Com o objetivo de incentivar essa prática, foi desenvolvido o sistema Adote Um Vira Lata, que centraliza a divulgação e a adoção de animais abandonados ou em situação de risco por meio da interação social. Esse projeto combina o crescente interesse pela interação virtual na sociedade com os benefícios de abordar o problema do abandono, muitas vezes subestimado. A concepção desta aplicação baseou-se nos princípios da UML, resultando na criação de diagramas de Classe e Casos de Uso. No contexto das operações de back-end, optou-se pelo desenvolvimento de uma API utilizando Node.js, em conjunto com um banco de dados MongoDB hospedado na nuvem, fornecido pelo serviço Heroku. Quanto à implementação das interfaces e às requisições da API, escolheu-se o framework React.js devido à sua característica de componentização. Por fim, os dados quantitativos e qualitativos ilustram uma transformação positiva na adoção de animais, destacando como o sistema Adote Um Vira-Lata promove o bem-estar dos animais e mobiliza a comunidade em prol da proteção animal.

Palavras-chave: Animais de estimação. Adoção. Abandono. Estimação. Node.js. React.js. Componentização.

Abstract

The presence of pets in Brazilian homes is constantly growing. Brazil has the fourth largest population of pets in the world. However, the number of abandoned animals registered in the country is worrying, which currently stands at 30 million. To combat this neglect, the culture of adoption has been promoted as an effective measure. With the aim of encouraging this practice, the Adote Um Vira Lata system was developed, which centralizes the dissemination and adoption of abandoned or at-risk animals through social interaction. This project combines the growing interest in virtual interaction in society with the benefits of addressing the often underestimated problem of abandonment. The design of this application was based on UML principles, resulting in the creation of Class and Use Case diagrams. In the context of back-end operations, it was decided to develop an API using Node.js, in conjunction with a Cloud-hosted MongoDB database powered by Heroku service. Regarding the implementation of interfaces and API requests, the React.js framework was chosen due to its componentization characteristic. Finally, the quantitative and qualitative data illustrate a positive transformation in animal adoption, highlighting how the Adote Um Vira-Lata system promotes animal welfare and mobilizes the community in favor of animal protection.

Keywords: Pets. Adoption. Abandonment. Estimation. Node.js. React.js. Componentization.

Lista de ilustrações

Figura 1 – UTILIZAÇÃO DO SISTEMA WEB	22
Figura 2 – PREFERÊNCIAS POR FILTRO DOS ANIMAIS	23
Figura 3 – PREFERÊNCIAS DAS INFORMAÇÕES DOS ANIMAIS	23
Figura 4 – NECESSIDADE DE FORMULÁRIO PARA ADOTANTES	23
Figura 5 – GERENCIAMENTO DE VOLUNTÁRIOS	24
Figura 6 – FUNCIONALIDADE ADICIONAL	24
Figura 7 – SITE AMIGO NÃO SE COMPRA	26
Figura 8 – SITE ROCK BICHO	27
Figura 9 – PLANO DE ATIVIDADE I	44
Figura 10 – PLANO DE ATIVIDADES II	45
Figura 11 – ARQUITETURA DO SISTEMA	52
Figura 12 – DIAGRAMA DE TELAS USUÁRIO	53
Figura 13 – DIAGRAMA DE TELAS VOLUNTARIO	54
Figura 14 – DIAGRAMA DE TELAS ADMINISTRADOR	54
Figura 15 – TELA DE CADASTRO	55
Figura 16 – TELA DE LOGIN	55
Figura 17 – TELA DE VISUALIZAÇÃO DE PETS	56
Figura 18 – TELA DE DETALHES DE PETS	57
Figura 19 – MODAL DE RESPONSABILIDADES ADOÇÃO	58
Figura 20 – MODAL DE FORMULÁRIO DE ADOÇÃO	59
Figura 21 – TELA DE VISUALIZAÇÃO DE ADOÇÕES	60
Figura 22 – MODAL DETALHES ADOÇÃO	60
Figura 23 – TELA DE VISUALIZAÇÃO DE ADOÇÕES	61
Figura 24 – TELA DE AGENDAMENTO DE VISITAS	62
Figura 25 – TELA DO ACOMPANHANTE DA VISITA	63
Figura 26 – TELA DO ACOMPANHANTE DA VISITA	63
Figura 27 – TELA DE VISUALIZAR VISITAS	64
Figura 28 – TELA DE SER VOLUNTÁRIO	65
Figura 29 – TELA DE PERFIL	65
Figura 30 – TELA DE NOTIFICAÇÕES - ABA NÃO VISTO	66
Figura 31 – TELA DE NOTIFICAÇÕES - ABA VISTO	66
Figura 32 – TELA GESTÃO DE PETS	67
Figura 33 – MODAL DE VISUALIZAÇÃO DE PET	67
Figura 34 – MODAL DE CONFIRMAÇÃO DE EXCLUSÃO	68
Figura 35 – TELA DE EDITAR PET	68
Figura 36 – TELA DE ADICIONAR PET	69
Figura 37 – TELA DE GESTÃO DE ADOÇÃO	70

Figura 38 – MODAL DE DETALHES DO ADOTANTE	70
Figura 39 – MODAL DE DETALHES DO ADOTANTE	71
Figura 40 – TELA GESTÃO DE ADOÇÕES	71
Figura 41 – CONTRATO	72
Figura 42 – TELA GESTÃO DE VISITAS - CALENDARIO	73
Figura 43 – TELA GESTÃO DE VISITAS - AGENDA	73
Figura 44 – TELA GESTÃO DE VISITAS - TABELA	74
Figura 45 – TELA GESTÃO DE USUÁRIOS - ABA ADOTANTE	75
Figura 46 – TELA GESTÃO DE USUÁRIOS - ABA VOLUNTÁRIOS	75
Figura 47 – DIAGRAMA DE CASOS DE USO	82
Figura 48 – DIAGRAMA DE CLASSES	83
Figura 49 – ESTRUTURA DO BANCO DE DADOS	85
Figura 50 – DS01 - EFETUAR LOGIN	108
Figura 51 – DS02 - EFETUAR CADASTRO	109
Figura 52 – DS03 - EDITAR PERFIL	110
Figura 53 – DS04 - VISUALIZAR PETS	110
Figura 54 – DS05 - SOLICITAR ADOCAO	111
Figura 55 – DS06 - AGENDAR VISITA	111
Figura 56 – DS07 - VISUALIZAR NOTIFICACOES	112
Figura 57 – DS08 - SER VOLUNTARIO	113
Figura 58 – DS09 - CADASTRAR PET	113
Figura 59 – DS10 - EDITAR PET	114
Figura 60 – DS11 - EXCLUIR PET	115
Figura 61 – DS12 - GERENCIAR ADOCOES	115
Figura 62 – DS13 - GERENCIAR VOLUNTARIO	116
Figura 63 – FLUXOGRAMA PROCESSO ADOÇÃO	117

Lista de tabelas

Tabela 1 – Tabela de Requisitos funcionais	37
Tabela 2 – Tabela de Requisitos não funcionais	39
Tabela 3 – Tabela Casos de Uso	40

Lista de abreviaturas e siglas

ABONG	Associação Brasileira de ONGs
ANDA	Agência de notícias de direitos animais
API	Application Programming Interface
CSS	Cascading Style Sheet
ECMA	European Computer Manufacturers Association
HTTP	HiperText Transference Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
IBP	Instituto Pet Brasil
IDE	Integrated Development Environment
JS	JavaScript
JSON	JavaScript Object Notation
OMS	Organização Mundial de Saúde
ONG	Organização não-governamental
ORM	Object-Relational Mapping
SQL	Structures Query Language
UFPR	Universidade Federal do Paraná
UML	Unified Modeling Language
WBS	Works Breakdown Structure

Sumário

1	INTRODUÇÃO	14
1.1	PROBLEMA	14
1.2	JUSTIFICATIVA	15
1.3	OBJETIVOS	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	16
1.4	ESTRUTURA DO DOCUMENTO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO	18
2.1.1	Abandono de animais	18
2.1.2	Organizações não governamentais	19
2.1.3	Adoção de animais	20
2.1.4	Adote um Vira Lata	21
2.2	PESQUISA COM PÚBLICO-ALVO	22
2.2.1	Questionário para Adote um Vira Lata	22
2.3	SOFTWARES SEMELHANTES	25
2.4	FUNDAMENTAÇÃO TEÓRICA DAS METODOLOGIAS	28
2.4.1	Métodos ágeis	28
2.4.2	Scrum	28
2.4.3	Kanban	29
2.5	FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS	30
2.5.1	JavaScript	30
2.5.2	React	30
2.5.3	Node.js e Express.js	31
2.5.4	MongoDB	32
2.5.5	Git	32
2.5.6	REST	33
2.5.7	UML	34
2.5.8	Cloud	35
3	MATERIAIS E MÉTODOS	36
3.1	PROCESSO DA ENGENHARIA DE SOFTWARE	36
3.1.1	Adaptação do Scrum para o desenvolvimento individual	36
3.1.2	Adaptação do Kanban para o desenvolvimento individual	36
3.2	REQUISITOS	37
3.2.1	Levantamento de Requisitos	37

3.2.2	Especificação de Requisitos	37
3.3	ANÁLISE - MODELAGEM DO SISTEMA	39
3.3.1	Diagrama de Casos de Uso	40
3.3.2	Diagrama de Classes	42
3.3.3	Diagrama de Sequência	42
3.3.4	Diagrama da Estrutura do Banco de Dados	42
3.4	PLANO DE ATIVIDADES	43
3.5	TECNOLOGIAS UTILIZADAS	45
3.5.1	Astah	45
3.5.2	Github	46
3.5.3	Postman	46
3.5.4	Visual Studio Code	46
3.5.5	Mern Stack	46
3.5.6	Ant Design	47
3.5.7	Cloudinary	47
3.5.8	Heroku	48
3.5.9	Hardware	48
3.6	DESENVOLVIMENTO DO PROJETO	48
3.6.1	Sprint 1	49
3.6.2	Sprint 2	49
3.6.3	Sprint 3	49
3.6.4	Sprint 4	49
3.6.5	Sprint 5	49
3.6.6	Sprint 6	50
3.6.7	Sprint 7	50
3.6.8	Sprint 8	50
3.6.9	Sprint 9	50
3.6.10	Sprint 10	50
4	APRESENTAÇÃO DO ADOTE UM VIRA LATA	52
4.1	ARQUITETURA DO SISTEMA	52
4.2	APRESENTAÇÃO DAS FUNCIONALIDADES	52
4.2.1	Cadastro	54
4.2.2	Login	55
4.2.3	Visualização Pets	56
4.2.4	Detalhes Pets	56
4.2.5	Adotar	57
4.2.6	Visualizar Adoções	59
4.2.7	Agendar Visitar	61

4.2.8	Acompanhante de Visita	62
4.2.9	Visualizar Visitas	64
4.2.10	Ser Voluntário	64
4.2.11	Perfil	65
4.2.12	Notificações	66
4.2.13	Gestão Pets	66
4.2.14	Gestão Adoção	69
4.2.15	Gestão Visita	72
4.2.16	Gestão Usuários	74
5	CONSIDERAÇÕES FINAIS	76
5.1	RECOMENDAÇÕES PARA TRABALHOS FUTUROS	76
5.1.1	Criação de um ChatBox	76
5.1.2	Acompanhamento Pós-Adoção	77
5.1.3	Responsividade e Identidade Visual	77
	Referências	78
	APÊNDICES	81
	APÊNDICE A – DIAGRAMA DE CASOS DE USO	82
	APÊNDICE B – DIAGRAMA DE CLASSES	83
	APÊNDICE C – DIAGRAMA DA ESTRUTURA DO BANCO DE DADOS	85
	APÊNDICE D – ENDPOINTS PARA AS REQUISIÇÕES	86
	APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA	108
	APÊNDICE F – FLUXOGRAMA PROCESSO ADOÇÃO	117

1 INTRODUÇÃO

O número de animais de estimação, também conhecidos como pets, no Brasil aumenta a cada ano. Em 2021, havia mais de 58 milhões de cães e 27 milhões de gatos, um aumento de 4,53% sobre os do ano anterior (BRASIL, 2022) . O Instituto Pet Brasil (IBP) constatou que esse aumento no número de animais de estimação está ligado à procura das pessoas e famílias por companhia.

A crescente demanda por animais de estimação resulta em um aumento do número de animais abandonados. A Organização Mundial da Saúde estima que no Brasil existam 30 milhões de animais abandonados , sendo 20 milhões de cães e 10 milhões de gatos. Esses animais ficam sujeitos a contrair doenças, serem atropelados e sofrerem maus tratos (ANDA, 2013) .

O abandono e maus tratos de animais é considerado um crime, porém é facilitado por falta de fiscalização, penalidades inadequadas e casos não denunciados, ou até mesmo por falta de conhecimento da lei (GOMES, 2013). A captura de animais de rua é realizada pelos centros de controle de zoonoses, visando reduzir a propagação de doenças. No entanto, a escassez de pessoal qualificado e a falta de infraestrutura resultam na morte de muitos animais saudáveis (DIAS, 2015).

Para abordar esse problema, surgem as associações de proteção aos animais, organizações sem fins lucrativos responsáveis pelo resgate e tratamento dos animais vítimas de maus-tratos, ajudando a recuperá-los até que se tornem adotáveis novamente pela sociedade. Apesar de sua importância, essas instituições não conseguem atender toda a demanda por si só, resultando em muitas delas abrigando mais animais do que sua infraestrutura permite.

1.1 PROBLEMA

Uma pesquisa realizada pelo Cobasi Cuida Brasil (COBASI, 2022) , com 57 organizações não governamentais (ONGs) e protetores independentes, revela que o abandono de animais ocorre em uma proporção maior do que se pode imaginar. A gravidade desse ato é evidenciada pelo fato de que 100% dos entrevistados afirmaram ter registrado casos de abandono ou devolução de animais. Além disso, foi constatado que a maioria dos casos de abandono está diretamente relacionada à falta de planejamento e problemas de posse responsável.

No Brasil, existem aproximadamente 185 mil animais que foram abandonados ou resgatados de situações de maus-tratos e estão sob os cuidados de ONGs e grupos de protetores. Os números do levantamento feito pelo IBP ressaltam a dimensão do problema do abandono e dos maus-tratos de animais no Brasil. As ONGs desempenham um papel fundamental na proteção e no resgate desses animais, oferecendo-lhes cuidados médi-

cos, alimentação, abrigo e buscando encontrar lares responsáveis por meio de adoções conscientes (IBP, 2022).

Devido à escassez de recursos, muitas dessas entidades dependem de doações de terceiros e parcerias com empresas para manter suas atividades. Aquelas com mais recursos podem contar com maior visibilidade através de mídias de impacto social ou até mesmo com seus próprios websites. No entanto, a maioria das entidades precisa buscar outros meios para chamar a atenção de potenciais financiadores e/ou adotantes. Atualmente, as principais ferramentas utilizadas por essas entidades para promover a adoção são os perfis em redes sociais, como Instagram e Facebook. No entanto, essas plataformas não são especificamente voltadas para esse propósito, o que dificulta muitas vezes a divulgação, já que as postagens podem se perder entre tantas outras interações dos usuários.

Uma das maiores entidades protetora de animais em Curitiba é a Adote um Vira Lata. O projeto é uma organização curitibana sem fins lucrativos dedicada à proteção e defesa dos animais. Eles realizam resgates, tratamentos, adoções e promovem a conscientização sobre a causa animal. Criada em dezembro de 2017, por Leticia Carolina Tavares, a Adote realiza todas as interações com os usuários através das redes sociais, que conta com 20 mil seguidores, tornando o processo de adoção mais demorado.

Dessa forma, o uso de um sistema web seria uma maneira de expandir as ações fornecidas pela entidade de proteção Adote um Vira Lata, agilizando o processo de adoção de maneira online. Sendo possível centralizar e automatizar o gerenciamento de todo o processo. Isso inclui a criação de perfis de animais disponíveis para adoção, o registro de interessados, a análise de candidatos e agendamento de visitas, entre outros. A automatização dessas tarefas reduziria a burocracia e agilizaria todo o processo, facilitando o acolhimento responsável e rápido dos animais.

1.2 JUSTIFICATIVA

Tendo em vista, como já mencionado, o grande número de animais abandonados e a pesquisa feita com a fundadora da Adote um Vira Lata, nasceu a ideia de desenvolvimento do projeto em questão.

O objetivo social deste trabalho é desenvolver um sistema web para adoção de animais que aborde os problemas mencionados anteriormente. O sistema terá como objetivo principal facilitar e agilizar o processo de adoção, proporcionando uma experiência mais satisfatória tanto para os adotantes em potencial quanto para os animais em busca de um lar.

Para tanto, propõe-se um sistema com objetivo de ser uma ferramenta que centralize as informações sobre animais disponíveis para adoção, facilitando para os interessados encontrar um animal compatível com suas preferências e condições. A ideia é que o site agilize processo de adoção, eliminando etapas burocráticas desnecessárias e possibilitando

a realização de pré-avaliações online, agendamentos de visitas e outras atividades por meio da plataforma. Os interessados em adoção poderão fazer buscas online, filtrando por espécie e porte, para encontrar o animal que mais se adequa às suas preferências entre os cadastrados. Além disso, o sistema fornece informações relevantes, como histórico de vacinação, castração e compatibilidade com outros animais, entre outros detalhes importantes para o adotante.

Desta forma, a ferramenta pode ser utilizada pelo projeto Adote um Vira Lata. Tendo assim, um sistema que possibilite: (1) estimular e facilitar o acesso aos animais para adoção; (2) auxiliar na conscientização das pessoas acerca da importância da adoção responsável; (3) ajudar na arrecadação de doações para o projeto.

1.3 OBJETIVOS

Dado a problemática apresentada anteriormente, são apresentados os objetivos gerais e específicos nas subseções a seguir.

1.3.1 Objetivo Geral

Desenvolver uma aplicação web, que tem como propósito centralizar e facilitar a divulgação de animais disponíveis para adoção, ao mesmo tempo em que promove a conscientização social sobre o problema do abandono de animais.

1.3.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Desenvolver uma aplicação web que facilite a divulgação de adoção de animais;
- Permitir que a entidade protetora “Adote um Vira Lata” tenha acesso à plataforma para gerenciar a divulgação de todos os animais disponíveis para adoção, incluindo informações detalhadas sobre cada animal;
- Possibilitar a visualização e acompanhamento dos animais que foram adotados por meio da plataforma, permitindo a atualização do status de adoção;
- Possibilitar o agendamento de visitas ao abrigo de animais para conhecer pessoalmente os animais disponíveis para adoção, proporcionando uma experiência interativa e facilitando a tomada de decisão no processo de adoção;
- Armazenar de maneira segura e eficiente os formulários de adoção responsável preenchidos pelos interessados, garantindo a integridade dos dados;

- Promover a usabilidade da plataforma, garantindo que seja intuitiva e fácil de usar, contribuindo para uma experiência positiva;
- Integrar a aplicação web com ferramentas de comunicação, como e-mail ou notificações, para manter os interessados informados sobre atualizações relacionados à adoção de animais;
- Documentar todo o sistema desenvolvido;

1.4 ESTRUTURA DO DOCUMENTO

Este documento está organizado em quatro capítulos adicionais, conforme descrito a seguir.

O Capítulo 2 pretende fornecer a fundamentação teórica do sistema, abordando de forma aprofundada os temas relevantes para a construção deste software. São discutidos assuntos como abandono de animais, organizações não governamentais, adoção e a entidade de proteção Adote um Vira Lata. Além disso, são introduzidos de forma concisa aspectos técnicos relacionados às tecnologias e metodologias utilizadas no desenvolvimento do projeto.

O Capítulo 3 aborda os materiais e métodos empregados, com suas definições técnicas, necessários para realizar o desenvolvimento do projeto de forma eficiente e aproveitando ao máximo os recursos físicos e tecnológicos disponíveis.

No Capítulo 4, são apresentados detalhes do software desenvolvido, incluindo informações sobre seu funcionamento e uma descrição das funcionalidades implementadas no sistema. Também são exibidas imagens que ilustram o produto final, oferecendo uma melhor compreensão do sistema descrito.

O Capítulo 5 apresenta as considerações finais sobre o projeto, analisando se os objetivos propostos foram alcançados de maneira satisfatória, identificando quaisquer tópicos que não obtiveram sucesso e delineando possíveis melhorias ou trabalhos a serem realizados em oportunidades futuras.

Ao final do documento, é possível encontrar os artefatos gerados durante o processo de definição do sistema, bem como a especificação técnica completa do projeto desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir, são abordados os tópicos relacionados aos temas e tecnologias relevantes para o desenvolvimento deste projeto. Além disso, serão apresentadas referências bibliográficas essenciais que embasam o desenvolvimento do trabalho. Com o objetivo de proporcionar uma compreensão aprofundada dos assuntos abordados nas seções e subseções a seguir, os temas foram organizados em categorias distintas, abrangendo os aspectos de Negócio, Metodologia e Tecnologia.

2.1 FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO

A seguir, serão explorados os temas relacionados ao âmbito do negócio, abordando as questões sociais e suas implicações no contexto do abandono de animais.

2.1.1 Abandono de animais

O abandono de animais é uma triste realidade que afeta milhões de animais em todo o mundo, incluindo o Brasil. De acordo com dados do Instituto Brasileiro de Geografia e Estatística (IBGE, 2013), estima-se que existam cerca de 30 milhões de animais abandonados no país. Esses animais enfrentam situações de extrema vulnerabilidade, sujeitos a doenças, fome, maus-tratos e até mesmo a morte (IBP, 2019).

Existem diversas causas que contribuem para o abandono de animais. Uma das principais é a falta de conscientização e responsabilidade por parte de alguns tutores. Muitas pessoas adquirem animais de estimação impulsivamente, sem considerar adequadamente as responsabilidades e cuidados necessários para criá-los. Com o tempo, essas pessoas podem se cansar da responsabilidade ou enfrentar dificuldades financeiras, resultando no abandono dos animais (ANJO, 2022).

Outro fator que contribui para o abandono é a falta de esterilização. A reprodução descontrolada de animais de estimação leva ao aumento da população e, conseqüentemente, ao aumento do número de animais abandonados. Uma amostragem realizada pela ARCA Brasil, ONG de proteção e bem-estar animal, aponta que no Brasil apenas 10% dos cachorros e gatos são castrados, contribuindo para a superpopulação e a propagação do abandono (ARCA, 2014).

Além disso, mudanças nas circunstâncias familiares, como divórcios, mudanças para locais onde animais não são permitidos ou a chegada de um novo membro na família, também podem levar ao abandono. Algumas pessoas consideram os animais como descartáveis, sem compreender o impacto emocional e físico que o abandono causa neles.

2.1.2 Organizações não governamentais

As Organizações Não Governamentais (ONGs), também conhecidas como entidades da sociedade civil, são entidades de natureza jurídica privada criadas pela iniciativa da sociedade. Elas desempenham um papel fundamental na abordagem de questões sociais que afetam a sociedade como um todo. Elas surgiram como uma resposta às demandas sociais não atendidas pelo Estado e pelo setor privado, atuando em diversas áreas, como assistência social, saúde, meio ambiente, educação, direitos humanos, entre outras. De acordo com a Associação Brasileira de Organizações Não Governamentais (ABONG, 2007), são consideradas ONGs,

[...] as entidades que, juridicamente constituídas sob a forma de fundação, associação e sociedade civil, todas sem fins lucrativos, notadamente autônomas e pluralistas, tenham compromisso com a construção de uma sociedade democrática, participativa e com o fortalecimento dos movimentos sociais de caráter democrático, condições estas, atestadas pelas suas trajetórias institucionais e pelos termos dos seus estatutos.

O surgimento das ONGs teve origem nas décadas de 1960 e 1970, quando houve um aumento significativo de associações, entidades e instituições que buscavam representar as demandas da população junto ao Estado. Esse movimento incluiu o notável trabalho das Comunidades Eclesiais de Base (CEBs), associações de pequenos produtores, cooperativas rurais e outras organizações. O impacto dessas organizações foi notável na promoção de uma cultura de ação não governamental, que, por sua vez, impulsionou o rápido crescimento e diversificação das ONGs nas décadas subsequentes (GALVAO; ARAUJO, 2014).

As ONGs desempenham um papel crucial na compreensão da crescente complexidade da sociedade brasileira, introduzindo novos processos e identidades na agenda política, simbólica e prática. Elas têm sido pioneiras em abrir caminhos para novas formas de relações sociais e métodos de produção, resultando na substituição das ações dos movimentos sociais por redes de trabalho e experiências concretas. Isso é mediado por parcerias e interações facilitadas pelas ONGs (ANDRADE, 2015).

Dessa forma, as ONGs surgiram como resposta ao desejo da sociedade civil de enfrentar problemas e desafios que não estavam sendo adequadamente abordados pelas instituições governamentais. Com frequência, atuam como fiscalizadoras, pressionando o Estado a cumprir suas obrigações e a garantir direitos.

Essas organizações obtêm financiamento de diversas fontes, como doações de pessoas físicas, empresas, fundações nacionais e internacionais, além de recursos provenientes de editais e convênios com o poder público. Essa variedade de fontes de financiamento contribui para a independência e autonomia das ONGs, permitindo que atuem de forma mais independente e crítica.

No contexto brasileiro, as ONGs dedicadas à adoção de animais desempenham um papel crucial na proteção e no cuidado desses seres vivos. Trabalhando de maneira voluntária e sem fins lucrativos, essas organizações se esforçam incansavelmente para

encontrar lares amorosos e responsáveis para animais que foram abandonados, maltratados ou estão em situação de risco. Seu compromisso em assegurar o bem-estar desses animais é fundamental para promover uma convivência harmoniosa entre seres humanos e animais na sociedade (CATLAND, 2023).

2.1.3 Adoção de animais

No Brasil, a legislação referente aos processos de adoção de animais, tanto domésticos quanto silvestres, ainda não é especificamente regulamentada por uma lei própria. No entanto, existem normas e regulamentos que abordam o comércio desses animais, visando controlar as operações de compra e venda, bem como restringir o comércio de animais silvestres em risco de extinção ou mesmo animais domésticos.

Em relação à Curitiba, foi aprovada e sancionada a Lei nº 13.914, que proíbe a criação comercial de animais em todo o município, exceto para aqueles que residem em áreas rurais (CURITIBA, 2011). Essa medida tem como objetivo regulamentar e restringir a criação de animais com finalidade comercial em áreas urbanas, garantindo a proteção e o bem-estar animal.

A proibição da criação comercial de animais em Curitiba busca combater a exploração irresponsável e descontrolada desses seres vivos, garantindo que eles não sejam criados em condições inadequadas, negligenciados ou submetidos a situações de maus-tratos. Além disso, a lei contribui para a conscientização da população sobre a importância de adotar animais em vez de comprá-los, promovendo a valorização da vida animal e incentivando a busca por opções de adoção responsável.

O processo de adoção de animais segue as seguintes etapas:

- **Entrevista:** No primeiro contato, o tutor passa por uma entrevista, na qual a ONG ou protetor busca obter informações sobre a experiência prévia do tutor com animais, se a residência possui espaço e segurança adequados para o animal, e se há crianças ou outras pessoas na casa. Essas perguntas têm o objetivo de garantir que o animal se adapte bem à nova família;
- **Entrega:** Em muitos casos, um voluntário acompanha a entrega do animal ao seu novo lar. Esse acompanhamento tem a finalidade de observar a reação do animal em seu novo ambiente e avaliar sua adaptação à casa e à família adotante;
- **Contrato de adoção:** No momento da adoção, é comum que o tutor receba um contrato ou formulário de adoção. Esse documento estabelece as responsabilidades do tutor em relação ao animal adotado. Nele, são especificadas as necessidades do animal e o tutor se compromete a cuidar dele ao longo de toda a sua vida. Caso o tutor não possa mais cuidar do animal em algum momento ou surja algum imprevisto,

ele se compromete a informar o doador original para que seja encontrada uma nova família para o animal;

Essas etapas são essenciais para garantir que a adoção ocorra de forma responsável e que o animal seja acolhido em um ambiente adequado e seguro. O objetivo é assegurar o bem-estar do animal adotado, bem como a conscientização do tutor sobre a importância de cuidar dele durante toda a sua vida. Além disso, o contrato de adoção ajuda a estabelecer uma rede de apoio entre o doador e o novo tutor, caso surjam quaisquer dificuldades ou necessidades futuras relacionadas ao animal.

2.1.4 Adote um Vira Lata

A Adote um Vira-Lata, ou apenas Adote, atua em Curitiba e conta com uma rede de voluntários e colaboradores dedicados ao bem-estar animal. A organização, situada na Rua Expedicionário José Hipólito 304, promove campanhas de conscientização sobre a importância da adoção, da esterilização e da posse responsável. Além disso, resgata animais em situação de abandono ou maus-tratos, oferece cuidados veterinários, abrigo temporário e busca encontrar lares amorosos para os animais por meio de feiras de adoção. A entidade também realiza trabalhos de educação, visando disseminar informações sobre os cuidados básicos com os animais, como alimentação adequada, vacinação, higiene e a importância do carinho e do respeito.

Em 2023, a entidade completa 6 anos, durante esse tempo, já resgatou mais de mil animais e encaminhou 800 para adoção. O objetivo do projeto é diminuir o número de animais abandonados e conscientizar a população sobre a importância da adoção responsável.

Com uma equipe de voluntários comprometidos e uma rede de parceiros e apoiadores, a Adote desempenha um papel fundamental na promoção da adoção responsável e na conscientização sobre a posse responsável de animais de estimação. Além do resgate e cuidado direto aos animais, a Adote também desenvolve programas de castração e esterilização, como uma forma de controle populacional e prevenção de doenças. Essas ações visam criar um impacto positivo e duradouro na vida dos animais, reduzindo o número de animais abandonados e garantindo uma vida saudável para aqueles sendo adotados.

A ONG também desempenha um papel importante na educação da comunidade, promovendo campanhas de conscientização sobre a importância da adoção responsável, dos cuidados veterinários, da vacinação e da alimentação adequada. Através de eventos, palestras e mídias sociais, eles buscam disseminar informações valiosas e inspirar ações positivas em prol dos animais.

Com sua dedicação, compromisso e amor pelos animais, a ONG Adote um Vira Lata tem se destacado como uma referência no cenário das organizações de proteção animal

no Brasil. Seu trabalho transforma a vida de inúmeros animais, proporcionando-lhes uma segunda chance de encontrar um lar amoroso e uma vida digna.

2.2 PESQUISA COM PÚBLICO-ALVO

Para validar a ideia do projeto, compreender melhor o ecossistema e mapear funcionalidades mais relevantes para o sistema foi realizada uma pesquisa com a entidade protetora em questão, como mostra o questionário a seguir.

2.2.1 Questionário para Adote um Vira Lata

No questionário foram feitas as seguintes perguntas:

1. Como um sistema web poderia agilizar o processo de adoção de animais no projeto?
2. Como o sistema web poderia facilitar a busca e filtragem de animais disponíveis para adoção?
3. Quais informações sobre os animais você considera como essenciais de serem apresentadas no sistema para auxiliar os adotantes na tomada de decisão?
4. O sistema web deveria incluir um formulário de inscrição para os adotantes?
5. Como você imagina que o sistema poderia auxiliar no gerenciamento de voluntários?
6. Além da gestão da adoção dos animais, qual funcionalidade específica você gostaria?

O questionário foi respondido em conjunto pela fundadora do projeto e por mais duas voluntárias envolvidas na iniciativa, como mostra a FIGURA 1 à FIGURA 6.

Figura 1 – COMO UM SISTEMA WEB AJUDARIA NO PROJETO

Como um sistema web poderia agilizar o processo de adoção de animais no projeto?

1 resposta

Um sistema web poderia agilizar o processo de adoção de animais ao permitir que os interessados visualizem e preencham formulários de adoção online. Isso economizaria tempo, pois as informações seriam enviadas diretamente ao nosso sistema, permitindo que a equipe avalie os candidatos de forma mais eficiente.

A Autora (2023)

Figura 2 – COMO UM SISTEMA WEB AJUDARIA NO FILTRO

Como o sistema web poderia facilitar a busca e filtragem de animais disponíveis para adoção?

1 resposta

O sistema web poderia fornecer uma interface amigável que permita aos adotantes em potencial buscar e filtrar animais com base em critérios como espécie, raça, idade, porte e localização. Isso facilitaria a visualização dos animais compatíveis com as preferências dos adotantes, agilizando o processo de seleção.

A Autora (2023)

Figura 3 – QUAIS INFORMAÇÕES SÃO IMPORTANTES

Quais informações sobre os animais você considera como essenciais de serem apresentadas no sistema para auxiliar os adotantes na tomada de decisão?

1 resposta

Informações essenciais incluem a descrição detalhada do animal, sua história de resgate, temperamento, compatibilidade com outros animais e crianças, necessidades especiais de cuidados, histórico médico, vacinação e esterilização. Fotos também são importantes para permitir aos adotantes visualizarem os animais.

A Autora (2023)

Figura 4 – FORMULÁRIO DE INSCRIÇÃO

O sistema web deveria incluir um formulário de inscrição para os adotantes?

1 resposta

Sim, um formulário de inscrição online seria uma funcionalidade valiosa do sistema web. Isso permitiria que os interessados preenchessem seus dados pessoais, fornecessem informações sobre sua experiência com animais de estimação anteriores e respondam a perguntas relacionadas à adoção responsável.

A Autora (2023)

Figura 5 – GERENCIAMENTO DE VOLUNTÁRIOS

Com você imagina que o sistema poderia auxiliar no gerenciamento de voluntários?

1 resposta

Um sistema poderia facilitar a administração dos voluntários, permitindo o cadastro e aprovação. Isso ampliaria a participação de mais pessoas no projeto, possibilitando maior engajamento e contribuição para a causa.

A Autora (2023)

Figura 6 – FUNCIONALIDADE ADICIONAL

Além da gestão de adoção dos animais, qual funcionalidade específica você gostaria?

1 resposta

Uma funcionalidade adicional que seria útil é a possibilidade de agendar visitas ao abrigo através do sistema web. Isso permitiria que os interessados conhecessem os animais pessoalmente e interagissem com eles, facilitando o processo de escolha e adoção.

A Autora (2023)

A partir das respostas, observou-se que um sistema web ajudaria no processo de adoção de animais. Através da tecnologia e da conexão entre adotantes e animais disponíveis, é possível tornar o processo mais eficiente, transparente e alcançar resultados mais positivos para o bem-estar animal. Dessa maneira, o presente projeto busca como foco principal facilitar o processo de acolhimento animal, assim o sistema terá 4 funcionalidades principais.

Primeiramente, o sistema contemplaria a divulgação dos animais disponíveis para adoção. A plataforma permitiria que a Adote um Vira Lata cadastrasse informações detalhadas sobre os animais prontos para adoção, apresentando dados específicos sobre cada um deles. Essa abordagem visa facilitar a visualização e busca por potenciais adotantes, proporcionando uma visão abrangente e transparente dos animais que aguardam um lar.

A segunda funcionalidade permite um processo de adoção simplificado online. Os interessados preenchem um formulário na plataforma, fornecendo informações sobre seu estilo de vida e experiência com animais. Esses dados auxiliam na análise para garantir uma correspondência mais precisa entre os adotantes e os animais disponíveis. Após o preenchimento, a plataforma realiza uma avaliação para possibilitar que os animais sejam adotados por lares adequados e responsáveis.

A terceira funcionalidade seria a facilitação do agendamento de visitas através da plataforma. Isso permitiria que os futuros adotantes agendem visitas aos abrigos de forma conveniente e organizada, possibilitando encontros presenciais com os animais que desejam adotar. Essa interação prévia entre o animal e o adotante em potencial é fundamental para a escolha responsável e afetuosa de um novo membro para a família.

Por fim, o sistema englobaria a possibilidade de aplicação para novos voluntários. Seria desenvolvida uma área para gestão e aplicação de novos voluntários interessados em auxiliar no abrigo. Essa funcionalidade contribuiria para ampliar a equipe de apoio, fortalecendo o suporte necessário para cuidar dos animais e manter as operações do abrigo de forma eficaz.

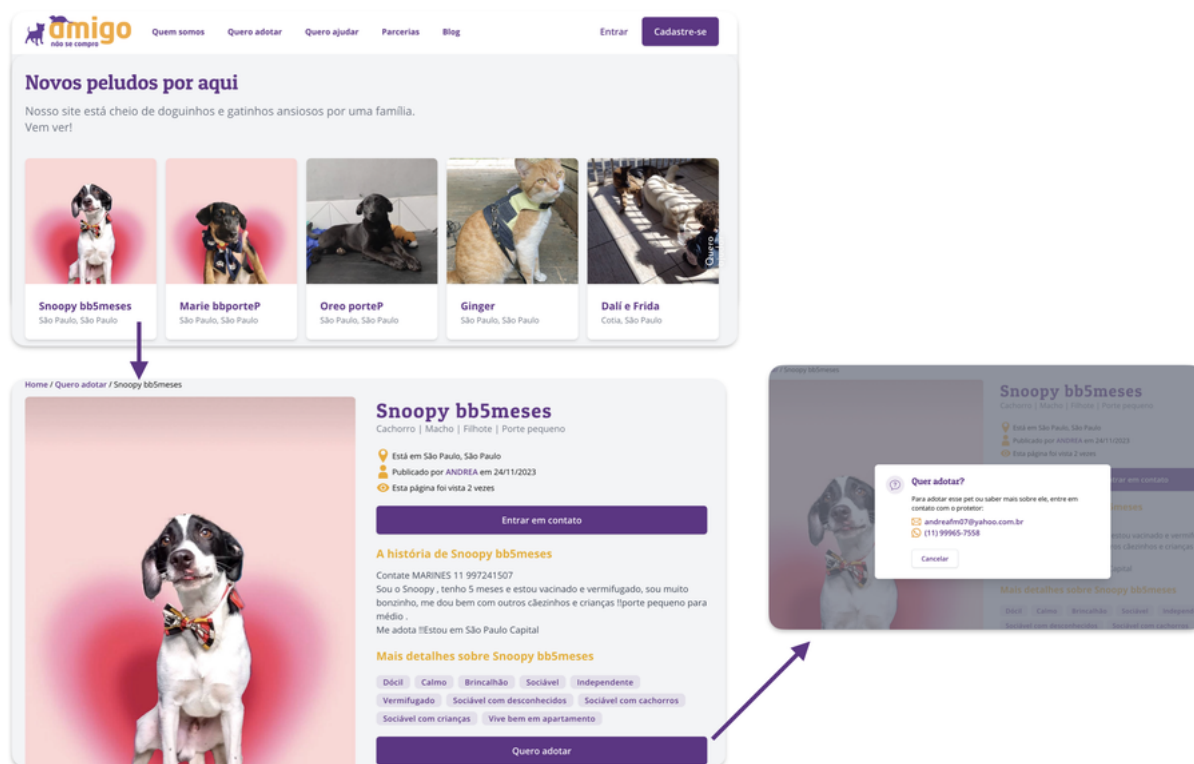
2.3 SOFTWARES SEMELHANTES

Com o intuito de compreender as tendências atuais da área e contribuir para a definição dos requisitos do sistema, foi conduzida uma análise de sites semelhantes ao escopo do projeto. A amostra dessa análise engloba dois sites que atuam na divulgação de animais disponíveis para adoção.

As amostras escolhidas foram:

a) Amostra 1 - Amigo Não Se Compra (<https://www.amigonaosecompra.com.br>): Plataforma dedicada à adoção de animais, oferecendo um espaço onde pessoas interessadas em encontrar um novo pet podem explorar uma variedade de animais disponíveis para adoção. Os usuários têm a chance de se registrar na plataforma e divulgar animais resgatados ou à procura de um novo lar. Se um usuário se interessar por um animal, é necessário contatar o responsável pelo cuidado do mesmo para dar sequência ao processo de adoção via WhatsApp, como mostra a FIGURA 7.

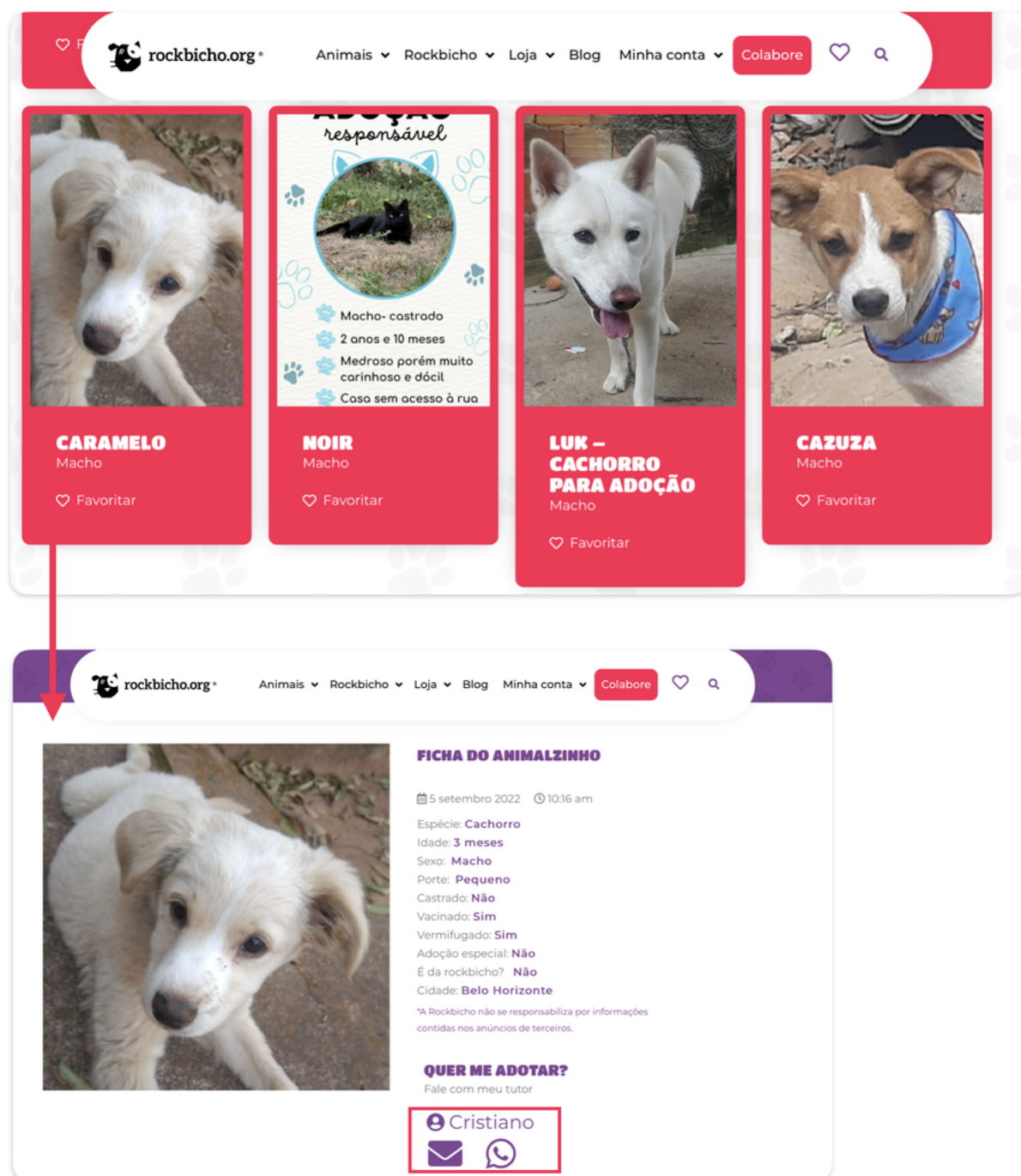
Figura 7 – SITE AMIGO NÃO SE COMPRA



Amigo não se compra (2023)

b) Amostra 2 - Rock Bicho (<https://rockbicho.org>): O site da Associação de Proteção Animal Rockbicho.org concentra-se na promoção da guarda responsável e na defesa dos direitos dos animais. Através dessa plataforma, é viável visualizar os animais resgatados à procura de um lar. Quando encontrar um animal desejado, o adotante pode dar início ao processo de adoção ao enviar um e-mail ao tutor responsável, como mostra na FIGURA 8.

Figura 8 – SITE ROCK BICHO



Rock bicho (2023)

Após analisar os sites similares ao propósito do projeto, identificou-se uma lacuna significativa em relação à adequação dos fluxos às necessidades específicas da Adote um Vira-Lata. A maioria dos softwares encontrados se baseava no uso predominante do WhatsApp ou do Email para administrar o fluxo de adoção, não oferecendo uma solução online dedicada e personalizada para entidade protetora. Isso ressalta a importância de criar um site customizado, alinhado às necessidades e aos objetivos específicos da organização

para tornar o processo de adoção mais eficiente e adaptado à realidade da Adote.

2.4 FUNDAMENTAÇÃO TEÓRICA DAS METODOLOGIAS

Nesta sessão estão presentes a fundamentação teórica por trás das metodologias utilizadas no desenvolvimento deste projeto.

2.4.1 Métodos ágeis

Os métodos ágeis são uma coleção de abordagens, princípios e práticas que visam agilizar e aprimorar a entrega de produtos. No ano de 2001, um grupo composto por 17 programadores, incluindo renomados profissionais como Robert Martin, Jeff Sutherland e Martin Fowler, elaborou o “Manifesto Ágil” (MANIFESTO, 2001). Esse documento estabelece quatro valores fundamentais:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

As metodologias ágeis procuram criar produtos com rapidez e qualidade, por meio de iterações frequentes em colaboração com os clientes (SCHWABER; HUNDHAUSEN; STARR, 2015). Esses valores e princípios provocaram grandes transformações na forma como o software é desenvolvido, resultando no surgimento de novos frameworks, como Scrum e Kanban.

Os métodos ágeis se espalharam rapidamente entre os profissionais de desenvolvimento de software e, conseqüentemente, em muitas empresas de tecnologia ao redor do mundo. Isso se deve aos vários benefícios claramente evidenciados pela prática ágil. As entregas se tornam mais ágeis e frequentes, aumentando a produtividade da equipe. Com a entrega constante de valor, há um maior engajamento com o cliente, permitindo feedbacks frequentes e reduzindo retrabalhos. Além disso, um processo de desenvolvimento mais direcionado permite previsões mais precisas de cronogramas e uma redução significativa nos custos.

2.4.2 Scrum

O Scrum, um framework amplamente utilizado na gestão do desenvolvimento de software, teve sua origem em 1993, criado por Jeff Sutherland e Ken Schwaber. As raízes

do Scrum remontam a um artigo pioneiro na Harvard Business Review, em 1986, intitulado “The New New Product Development Game”, escrito por Hirotaka Takeuchi e Ikujiro Nonaka.

Amplamente adotado nos anos 2000, o Scrum emergiu como a metodologia ágil mais utilizada no desenvolvimento de software. Ao combinar conceitos já existentes do mercado de forma organizada e eficaz, este framework reúne práticas já conhecidas e difundidas. O método Scrum tem como objetivo fragmentar grandes entregas em etapas menores, essa abordagem visa assegurar maior qualidade, agilidade e redução de retrabalho (SABBAGH, 2013).

No contexto do Scrum, a equipe efetua melhorias e incorpora novas funcionalidades em sprints, que são intervalos cíclicos, frequentemente realizados em períodos de duas semanas, durante as sprints, ocorrem quatro eventos principais. A Sprint Planning, na qual a equipe prioriza as principais tarefas a serem realizadas durante a sprint. A Daily Meeting, uma reunião diária de até 15 minutos, destinada a atualizar a equipe sobre o progresso e os obstáculos enfrentados pelos membros. A Review, que envolve a revisão do trabalho concluído durante a sprint, apresentado aos stakeholders somente o que foi finalizado. E, por último, a Retrospective, realizada entre os membros da equipe para analisar os pontos críticos e de sucesso, buscando melhorias para as próximas sprints (SCHWABER, 2004).

O Scrum é amplamente adotado devido à sua capacidade de promover a colaboração, a transparência, a flexibilidade e a rápida entrega de valor. Atualmente, é utilizado em empresas como Yahoo, Google, Microsoft, Globo, Adobe, etc (AWARI, 2023b). Ele permite uma melhor adaptação às mudanças nos requisitos do projeto e proporciona maior visibilidade do progresso para todas as partes interessadas.

2.4.3 Kanban

O Kanban, criado por Taiichi Ohno na década de 1960, é um método de organização originado da necessidade de controlar todas as etapas da produção na Toyota, uma empresa japonesa, com o objetivo de eliminar o desperdício de materiais (GIL, 2019).

O propósito do Kanban é proporcionar uma visualização clara das tarefas a serem realizadas, em andamento e concluídas. Isso é alcançado por meio de um quadro no qual cartões representam cada atividade, e esses cartões são organizados em colunas que indicam o status de cada tarefa. O Kanban é projetado para ser ágil, permitindo que os envolvidos gerenciem suas tarefas de maneira eficaz, e visual, de modo que toda a equipe tenha conhecimento das tarefas planejadas (BOEG, 2012).

Embora o Kanban seja frequentemente comparado a uma lista de tarefas (To-do List), ele vai além, enfatizando o acompanhamento do estado das tarefas, a definição de responsáveis por cada uma e a priorização, muitas vezes utilizando cores para tornar esses aspectos visíveis (GIL, 2019).

2.5 FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS

Além dos temas abordados no contexto Negócio, também são relevantes e igualmente significativos aqueles relacionados às tecnologias empregadas no desenvolvimento deste projeto. A seguir, são apresentadas as definições dessas vertentes tecnológicas.

2.5.1 JavaScript

O JavaScript (JS) é uma linguagem de programação que foi desenvolvido em 1995 por Brendan Eich para o navegador Netscape 2 e, em 1997, tornou-se o padrão ECMA-262, ES1. Atualmente, todos os navegadores oferecem suporte ao padrão de 2016, conhecido como ES6. De acordo com a pesquisa “2022 Developer Survey” do Stack Overflow (2022), o JavaScript continua sendo a linguagem mais utilizada pelos desenvolvedores, com cerca de 65% de toda a comunidade de programação ao redor do mundo (STACK OVERFLOW, 2022).

O JavaScript é considerado uma linguagem multiparadigma, pois permite a utilização de diferentes estilos de programação para resolver uma variedade de problemas. Ele oferece suporte à programação orientada a objetos, programação estruturada, programação funcional, entre outros (MOZILLA, 2021, 2021).

Independentemente da forma como é utilizado, o JavaScript pode ser executado tanto no lado do servidor (server-side) quanto no lado do cliente, no navegador (client-side). Um dos frameworks JavaScript pioneiros para execução no servidor é o Node.js. Já no lado do cliente, os principais frameworks utilizados incluem AngularJS, Vue e React (MOZILLA, 2021, 2021). Esses frameworks fornecem uma estrutura robusta para o desenvolvimento de aplicações interativas e dinâmicas no navegador.

Com sua ampla adoção e versatilidade, o JavaScript se tornou a escolha preferida dos desenvolvedores para criar uma variedade de aplicações web, desde sites simples até aplicativos complexos. Sua capacidade de executar no servidor e no cliente, juntamente com os diversos frameworks disponíveis, torna o JavaScript uma linguagem poderosa e essencial no mundo da programação.

2.5.2 React

React.js é uma biblioteca JavaScript de código aberto amplamente utilizada para criar interfaces de usuário interativas e dinâmicas. Foi desenvolvida pelo Facebook e lançada pela primeira vez em 2013. Desde então, ganhou popularidade entre desenvolvedores de todo o mundo devido à sua eficiência, flexibilidade e desempenho.

A criação do React.js foi liderada por Jordan Walke, um engenheiro de software do Facebook. Ele desenvolveu o React originalmente para resolver os desafios enfrentados pelo Facebook ao lidar com uma grande quantidade de código complexo e em constante

mudança. A ideia era criar uma solução que permitisse atualizações eficientes da interface do usuário, sem recarregar a página inteira.

O React.js utiliza um conceito chamado “componentes” como base para o desenvolvimento de interfaces de usuário. Os componentes são blocos de construção reutilizáveis que encapsulam o código e o estado relacionado a uma parte específica da interface. Isso permite que os desenvolvedores dividam a interface em componentes menores e independentes, facilitando a construção e a manutenção de aplicativos complexos (REACT, 2023).

Uma das principais características do React.js é a sua abordagem baseada em “Virtual DOM” (Documento de Objeto Virtual). O Virtual DOM é uma representação em memória da estrutura de elementos da interface do usuário. O React compara eficientemente as alterações no Virtual DOM com o DOM real, determina as diferenças e atualiza apenas as partes afetadas, minimizando a necessidade de renderizações completas da página.

Outra vantagem do React.js é sua compatibilidade com outras bibliotecas e frameworks. Ele pode ser combinado com bibliotecas como o Redux para gerenciamento de estado ou o React Router para navegação entre páginas. Além disso, o React Native, uma extensão do React.js, permite o desenvolvimento de aplicativos móveis nativos para iOS e Android usando JavaScript.

Existem vários frameworks e bibliotecas relacionados ao React, como o Next.js, Gatsby e o Create React App, estes foram criados para resolver desafios comuns no desenvolvimento de aplicativos React e para facilitar o processo de criação de interfaces de usuário reativas, escaláveis e de alto desempenho. Eles fornecem recursos adicionais, como roteamento, renderização no servidor, geração estática de páginas e ferramentas de desenvolvimento, que ajudam a simplificar o desenvolvimento de aplicativos React e melhorar a experiência do desenvolvedor (SILVEIRA, 2023).

O React.js tornou-se uma escolha popular para o desenvolvimento web moderno devido à sua facilidade de uso, eficiência, modularidade e grande comunidade de desenvolvedores. Através do React.js, os desenvolvedores podem criar interfaces de usuário interativas, responsivas e de alto desempenho, proporcionando uma experiência de usuário excepcional em aplicativos da web.

2.5.3 Node.js e Express.js

O Node.js é um ambiente de execução que utiliza o motor V8 do Google Chrome para interpretar e executar código JavaScript. Sua principal característica é a capacidade de ser não-bloqueante, ou seja, operar de forma assíncrona, executando em um servidor em uma única thread, o que implica que os recursos computacionais são alocados de maneira eficiente (ESTRELLA, 2023).

Esse projeto open-source (código aberto) foi concebido por Ryan Dahl em 2009,

com a visão de possibilitar o uso da mesma linguagem de programação no front-end e no back-end.

Por ser uma aplicação server-side, o Node.js oferece a flexibilidade de criar aplicações JavaScript que podem ser executadas independentemente de um navegador, diferenciando-se do cenário tradicional. Grandes empresas do setor de tecnologia, como Netflix, Uber e até o LinkedIn, adotaram o Node.js em suas infraestruturas, contribuindo para a consolidação e prestígio dessa ferramenta (ESEME, 2023) .

O Express.js, por sua vez, é um framework web construído sobre o Node.js, oferecendo suporte essencial para a organização de aplicações no servidor. Este framework apresenta estruturas para roteamento e facilita o gerenciamento de requisições e respostas HTTP (MONGODB, 2023a).

2.5.4 MongoDB

O MongoDB é um banco de dados orientado a documentos e não relacional. Os documentos, que consistem em dados declarados na estrutura chave-valor, são armazenados em formato BSON (Binary JSON) e organizados em coleções dentro de uma base de dados (MONGODB, 2023b).

A modelagem de dados no MongoDB é adaptável às necessidades específicas de cada projeto. A documentação enfatiza a importância de considerar os dados necessários para a aplicação, frequentemente encorajando o agrupamento de dados em um único documento e redundância de informações. Essas práticas visam melhorar a velocidade das consultas no banco em comparação com esquemas relacionais, pois reduzem a necessidade de buscar em outras coleções (MONGODB, 2023b).

2.5.5 Git

O Git é um sistema de controle de versão desenvolvido por Linus Torvalds, com o objetivo de facilitar o desenvolvimento colaborativo em seu projeto principal, o Kernel do Linux. Antes do Git, Torvalds e sua equipe utilizavam outra ferramenta de controle de versão, que passou a ter políticas de pagamento. Diante disso, Torvalds viu a necessidade de criar sua própria ferramenta, incorporando os aprendizados adquiridos com o uso de outras ferramentas (GIT, 2023).

Ao criar o Git, Torvalds definiu cinco princípios fundamentais para a ferramenta: velocidade, simplicidade, desenvolvimento não linear, distribuição e eficiência no gerenciamento de projetos grandes. Desde sua criação em 2002, o Git tem se mantido fiel a esses ideais e, ao longo do tempo, tem se tornado ainda mais simples de utilizar.

O Git funciona organizando um repositório, que é um diretório controlado pelo Git, com uma ou mais ramificações (branches) do código original. A cada nova versão, são geradas cópias apenas das alterações feitas, permitindo manter um histórico completo

das modificações. Isso possibilita a comparação entre as versões, a realização de merges (combinação de duas ramificações em uma única), a reversão de alterações e a possibilidade de baixar o código em diferentes máquinas (GIT, 2023).

Com o Git, os desenvolvedores podem trabalhar de forma mais eficiente, garantindo um melhor controle das versões do código, facilitando a colaboração em equipe e permitindo um acompanhamento detalhado das alterações ao longo do tempo.

2.5.6 REST

A arquitetura de software REST (Representational State Transfer) é um estilo arquitetural amplamente utilizado para projetar e desenvolver sistemas distribuídos na web, proposto por Roy Thomas Fielding em 2000 (GUPTA, 2023). O REST é baseado em princípios e restrições que permitem a criação de serviços web escaláveis, interoperáveis e de fácil manutenção.

O REST utiliza o protocolo HTTP (Hypertext Transfer Protocol) como base de comunicação entre cliente e servidor. Ele define um conjunto de operações, conhecidas como verbos HTTP, para interagir com recursos. Os principais verbos são GET, POST, PUT e DELETE, que correspondem às operações de recuperar, criar, atualizar e excluir recursos, respectivamente.

A arquitetura REST segue alguns princípios fundamentais (AWARI, 2023a). O primeiro é a utilização de recursos identificados por URLs (Uniform Resource Locators) únicas. Cada recurso possui uma representação, geralmente em formato JSON ou XML, que é transmitida entre o cliente e o servidor.

Outro princípio importante do REST é a utilização de um conjunto uniforme de interfaces. Isso significa que as operações são padronizadas e consistentes em todos os recursos. Isso simplifica o desenvolvimento, a manutenção e a integração de diferentes componentes do sistema.

Uma característica essencial do REST é o estado transferido entre o cliente e o servidor. O servidor não mantém nenhum estado da sessão do cliente, todas as informações necessárias para processar uma requisição devem ser incluídas na própria requisição. Isso permite que os serviços REST sejam altamente escaláveis e independentes de estado.

A arquitetura REST é amplamente adotada para a construção de APIs (Application Programming Interfaces) web. Ela permite que diferentes sistemas e plataformas se comuniquem de maneira eficiente e interoperável. Além disso, o REST facilita a evolução do sistema, pois cada recurso pode ser desenvolvido, testado e atualizado independentemente dos demais.

2.5.7 UML

O desenvolvimento de software busca proporcionar uma forma eficiente de criar um sistema de sucesso. Essa jornada envolve diversas etapas, como a análise de requisitos, especificação e construção do sistema. Para auxiliar nesses processos, a Unified Modeling Language (UML) é uma linguagem gráfica que permite modelar e visualizar os procedimentos envolvidos (AHMED, CARY E., 2002). A UML teve sua formalização e unificação dos padrões de notação estabelecidos a partir dos anos 90. Com base em abordagens anteriores, como Booch e OMT, a UML incorpora os principais recursos desses padrões de análise (BOOCH; RUMBAUGH; JACOBSON, 2006).

De acordo com Larman (2005), a UML pode ser aplicada de três maneiras: como rascunho, onde os diagramas são esboçados à mão; como planta de software, onde os diagramas mais detalhados podem ser usados para engenharia reversa ou codificação do sistema; e como linguagem de programação, onde uma especificação completa do sistema em UML é gerada como um executável, contemplando toda a lógica e comportamento analisados.

A UML é uma ferramenta abrangente para modelar visualmente sistemas de software. Ela oferece maneiras de representar não apenas os elementos mais simples de construção, mas também recursos para ilustrar as relações mais complexas entre eles (AHMED, CARY E., 2002). A UML vai além de apenas símbolos e gráficos. Cada símbolo utilizado na notação UML possui uma semântica bem definida. Isso significa que um desenvolvedor pode usar a UML para escrever seu modelo, e qualquer outro desenvolvedor ou ferramenta será capaz de interpretá-lo sem ambiguidades (BOOCH; RUMBAUGH; JACOBSON, 2006).

As particularidades de um sistema, sejam elas estáticas ou dinâmicas, são ilustradas em diagramas. Existem vários tipos de representação, agrupados em áreas focais específicas da modelagem visual, como:

- Diagrama de Caso de Uso: Identifica as funcionalidades exigidas no sistema a partir do ponto de vista do usuário;
- Diagrama de Classe: Demonstra as relações estáticas entre um grupo de classes e/ou interfaces do software;
- Diagrama de Sequência: Modela a troca de mensagens realizada entre os objetos de um sistema;

Na UML, os objetos gerados são tratados de maneira distinta, devido às práticas de desenvolvimento. Esses artefatos não são apenas partes do projeto a serem entregues, mas também são fundamentais para gerenciar, preservar e disseminar um sistema durante seu desenvolvimento e após sua implantação (BOOCH; RUMBAUGH; JACOBSON, 2006).

2.5.8 Cloud

Cloud, ou computação em nuvem, é um modelo que disponibiliza serviços de computação pela internet. Ao invés de manter aplicativos, dados e recursos em servidores locais ou em hardware físico, a computação em nuvem permite acesso e gestão desses recursos remotamente, utilizando servidores e infraestrutura de data centers espalhados globalmente. É a execução de tarefas computacionais em ambientes de TI conhecidos como nuvens, que abstraem, agrupam e compartilham recursos escaláveis por meio de uma rede. (HAT, 2023)

A computação em nuvem oferece uma série de benefícios significativos para usuários e empresas. Em primeiro lugar, ela proporciona flexibilidade e escalabilidade, permitindo que os recursos sejam dimensionados de acordo com a demanda, aumentando ou reduzindo a capacidade conforme necessário.

Além disso, a computação em nuvem oferece maior disponibilidade e confiabilidade, uma vez que os dados e aplicativos são distribuídos em vários servidores e locais geográficos. Isso garante que, mesmo em caso de falha em um servidor ou data center, os serviços possam ser continuamente disponibilizados a partir de outros pontos da rede.

Entre os principais provedores de serviços em nuvem estão Amazon Web Services (AWS), Microsoft Azure, IBM Cloud e Vercel. Esses provedores oferecem uma ampla gama de serviços e soluções para atender às necessidades de empresas de todos os tamanhos e setores.

3 MATERIAIS E MÉTODOS

Para o desenvolvimento deste trabalho, sustentou-se em uma diversidade de ferramentas, metodologias e planejamentos. Elas são documentadas a seguir.

3.1 PROCESSO DA ENGENHARIA DE SOFTWARE

Durante a execução do projeto, foram aplicadas as metodologias Kanban e Scrum como base para os processos de documentação e desenvolvimento. No entanto, foram realizadas adaptações específicas com o objetivo de aprimorar a produtividade.

3.1.1 Adaptação do Scrum para o desenvolvimento individual

Embora o Scrum seja uma metodologia ágil voltada para o trabalho em equipe, é possível adaptá-la e utilizá-la de maneira individual para o gerenciamento de projetos.

Durante o desenvolvimento deste projeto, as práticas cerimoniais do Scrum foram adaptadas para se adequarem a um ambiente de trabalho individual. Portanto, toda quinta-feira era dedicada ao encerramento da sprint. O backlog sempre continha tarefas planejadas para a sprint em andamento. As retrospectivas eram realizadas ao final de cada sprint, enquanto as reviews eram conduzidas com a orientação do supervisor.

É importante ressaltar que, ao trabalhar individualmente com o Scrum, o foco estará na gestão pessoal do projeto e no cumprimento dos prazos e metas estabelecidos.

3.1.2 Adaptação do Kanban para o desenvolvimento individual

O Kanban é uma metodologia ágil que pode ser adaptada e utilizada de maneira individual para o gerenciamento de tarefas e fluxo de trabalho pessoal. Ao aplicar o Kanban individualmente, é possível visualizar e controlar as atividades de forma mais eficiente.

O Kanban desempenhou um papel crucial na documentação e desenvolvimento deste projeto, fornecendo orientações claras por meio de cartões. Para implementar o Kanban, utilizou-se o Trello, uma ferramenta da Atlassian que permite a personalização dos cartões e das colunas do quadro Kanban. As colunas utilizadas foram “Backlog”, “Em Progresso”, “Concluído” e “A Fazer”. Esta última refere-se as tarefas adicionais caso houvesse tempo disponível durante a sprint.

Cada cartão foi numerado para indicar sua ordem de prioridade. Além disso, foram incorporadas etiquetas nos cartões para representar diferentes contextos das tarefas. Foram criadas quatro etiquetas, cada uma com uma cor específica: roxa para documentação, amarela para front-end, azul para back-end.

3.2 REQUISITOS

Durante a execução do projeto, foram aplicadas metodologias para os processos de levantamento de requisitos, como entrevista e questionários aplicados para os voluntários do projeto.

3.2.1 Levantamento de Requisitos

Por meio da entrevista e questionário feitos com a fundadora da iniciativa foram levantados os primeiros requisitos. Com a entrevista foi possível perceber as principais necessidades existentes em relação ao processo de adoção de animais. Assim, os principais requisitos levantados foram: estimular e facilitar o acesso aos animais para adoção; auxiliar na conscientização das pessoas acerca da importância da adoção responsável; ajudar na arrecadação de doações para o projeto.

Após esta entrevista, foi elaborado um questionário, aplicado à fundadora e duas voluntárias do projeto. Com este questionário foi possível entender melhor as necessidades da entidade protetora e se elas usariam a ferramenta proposta. O questionário é composto de 7 perguntas abertas que estão presentes na seção 2.2.1, em que o foco é saber quais seriam as principais funcionalidades necessárias em um sistema web para adoção de animais. A maioria das respostas focou na necessidade do sistema promover a adoção responsável.

3.2.2 Especificação de Requisitos

A especificação de requisitos é um processo crucial que abrange o estudo minucioso das necessidades do sistema, visando alcançar uma definição precisa e completa dos requisitos do software.

Para uma compreensão mais clara e efetiva, os requisitos são classificados em duas categorias principais: funcionais e não funcionais. Os requisitos funcionais delineiam as funcionalidades desejadas do software, descrevendo as ações e operações que o sistema deve ser capaz de executar (SOMMERVILLE, 2007).

Tabela 1 – Tabela de Requisitos funcionais

Número Requisito	Nome Requisito	Descrição
------------------	----------------	-----------

Número Requisito	Nome Requisito	Descrição
RF001	Cadastro de animais	Permitir o cadastro completo de animais disponíveis para adoção, incluindo informações como espécie, raça, idade, sexo, porte e descrição do comportamento.
RF002	Pesquisa e filtragem	Possibilitar aos usuários pesquisar e filtrar animais com base em critérios como espécie, idade e porte, para encontrar animais adequados às suas preferências.
RF003	Visualização de perfis	Exibir perfis individuais para cada animal, com fotos, informações detalhadas sobre sua personalidade, necessidades especiais, histórico médico e qualquer outra informação relevante para ajudar os adotantes na tomada de decisão.
RF004	Formulário de inscrição	Incluir um formulário de inscrição para que os interessados em adotar um animal possam se cadastrar, fornecendo informações pessoais, experiência prévia com animais de estimação e condições de moradia.
RF005	Processo de adoção	Implementar um processo de adoção com etapas definidas, permitindo que os adotantes avancem nas etapas à medida que cumprem os requisitos necessários, como formulário, visitas e assinatura do contrato.
RF006	Gerenciamento de adoções	Registrar e gerenciar as adoções concluídas, mantendo um histórico atualizado dos animais adotados, adotantes e datas de adoção.
RF007	Gerenciamento de voluntários	Permitir o cadastro e gerenciamento de voluntários no sistema.
RF008	Notificações	Enviar notificações para manter os usuários informados e garantir a organização na plataforma.
RF009	Agendamento de visitas	Permitir que os adotantes agendem visitas presenciais para conhecer os animais e interagir com eles antes de tomar a decisão final de adoção.

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

Tabela 2 – Tabela de Requisitos não funcionais

Número Requisito	Nome Requisito	Descrição
RF001	Usabilidade	O sistema deve ser intuitivo, de fácil navegação e compreensão, possibilitando que usuários de diferentes níveis de habilidade possam utilizá-lo sem dificuldades.
RF002	Desempenho	O sistema deve ter um desempenho rápido e responsivo, garantindo tempos de carregamento mínimos e processamento eficiente das operações, mesmo com grande volume de dados.
RF003	Segurança	O sistema deve possuir medidas robustas de segurança para proteger informações sensíveis dos usuários, como dados pessoais, registros de adoção e informações financeiras.
RF004	Manutenibilidade	O sistema deve ser facilmente mantido e atualizado, permitindo a incorporação de novas funcionalidades, correção de erros e ajustes conforme as necessidades da entidade protetora e dos usuários.
RF005	Documentação	O sistema deve possuir uma documentação abrangente, clara e atualizada, que descreva todas as funcionalidades, fluxos de trabalho e procedimentos operacionais, facilitando o entendimento e auxiliando na manutenção do sistema.

3.3 ANÁLISE - MODELAGEM DO SISTEMA

A modelagem desempenha um papel crucial na concepção e desenvolvimento de sistemas de software de qualidade. Segundo os renomados autores Booch, Rumbaugh e Jacobson (2005), uma empresa de destaque é aquela que oferece sistemas de alta qualidade, atendendo precisamente às necessidades de seus usuários.

A representação visual ampla e estruturada do sistema em desenvolvimento desempenha um papel crucial na orientação das atividades durante sua construção. Essa visualização é essencial para o entendimento e a execução eficaz do projeto (BOOCH; RUMBAUGH; JACOBSON, 2006).

A escolha da UML para o desenvolvimento do sistema Adote um Vira Lata foi motivada pela sua significativa relevância no processo de análise e modelagem de software, fornecendo uma base sólida para representar e comunicar a estrutura e o comportamento do sistema de forma clara e eficaz. Nesse contexto, a modelagem do sistema envolveu a

criação dos seguintes diagramas UML:

- Diagrama e especificação de Casos de Uso
- Diagrama de Classe
- Diagrama de Sequência

3.3.1 Diagrama de Casos de Uso

O diagrama de casos de uso é uma representação comportamental integrante da linguagem UML, sendo responsável por ilustrar requisitos funcionais e descrever como um sistema se comporta na perspectiva de um ator (SILVA; MARTINS; DINIZ, 2017).

Em conformidade com os requisitos identificados na seção 3.2.2, foi desenvolvido o Diagrama de Casos de Uso do projeto, cuja visualização está disponível no APÊNDICE A.

Após coletar os requisitos funcionais do sistema foi feita a criação do diagrama e a especificação dos casos de uso (TABELA 3). Essa etapa permitiu identificar a necessidade de três atores para o funcionamento do aplicativo:

- (1) Usuário: pode efetuar registro no sistema, fornecendo os dados necessários; efetuar login através de email e senha; pesquisar e filtrar animais; solicitar adoção; visualizar os animais adotados e efetuar o agendamento de visitas.
- (2) Voluntário: pode gerenciar os adotantes; cadastrar animais; gerenciar animais; validar adoções; gerar contrato e gerenciar visitas.
- (3) Administrador: gerencia os voluntários e usuários do projeto.

Tabela 3 – Tabela Casos de Uso

Caso de Uso	Descrição
UC001 - Efetuar Cadastro	O usuário pode se cadastrar no sistema, fornecendo as informações necessárias, como nome, e-mail e senha, que será criptografada. Após o cadastro, o usuário terá acesso às funcionalidades do sistema.

Caso de Uso	Descrição
UC002 - Efetuar Login	O usuário pode fazer login no sistema utilizando suas credenciais de acesso, como e-mail e senha. Após a autenticação, o usuário poderá acessar as funcionalidades do sistema de acordo com seu perfil.
UC003 - Visualizar Pets	O usuário pode visualizar os pets disponíveis para adoção na plataforma, acessando informações como fotos, descrições, idade, raça e estado de saúde. Isso oferece uma visão detalhada para facilitar a escolha do animal mais adequado às preferências do adotante.
UC004 - Solicitar Adoção	O usuário tem a possibilidade de solicitar a adoção de um animal específico através da plataforma. Para isso, ele preenche um formulário contendo informações cruciais para a adoção responsável, como motivação para adoção, detalhes sobre o ambiente de moradia e a disponibilidade para cuidar do animal.
UC005 - Visualizar Adoções	O usuário tem acesso à visualização dos pets adotados por ele anteriormente, bem como àqueles para os quais solicitou a adoção e ainda aguarda a aprovação.
UC006 - Agendar Visita	O usuário pode agendar uma visita para conhecer um pet disponível para adoção. O usuário selecionará o animal desejado, escolherá uma data e horário disponível e fará a confirmação do agendamento.
UC007 - Editar Perfil	O usuário pode modificar e atualizar as informações presentes em seus perfis na plataforma. Isso inclui a capacidade de alterar dados pessoais, como nome, informações de contato e preferências específicas.
UC008 - Visualizar Notificações	O usuário tem a opção de verificar e gerenciar as notificações enviadas pelo sistema. Isso inclui detalhes sobre o status da adoção, confirmação de visitas e outros avisos pertinentes. Além disso, o usuário poderá marcar as notificações como vistas ou deletá-las conforme necessário.
UC009 - Ser Voluntário	O usuário pode se candidatar para ser voluntário na plataforma. Ele pode enviar sua inscrição para participar ativamente como voluntários em atividades relacionadas ao cuidado e suporte aos animais, eventos de conscientização e outras iniciativas promovidas por abrigos e organizações parceiras.
UC010 - Gerenciar Adoções	O voluntário pode administrar os adotantes registrados no sistema, o que inclui visualizar informações, aprovar solicitações de adoção e acessar o histórico completo das adoções realizadas por esses adotantes

Caso de Uso	Descrição
UC011 - Cadastrar Pets	O voluntário poderá cadastrar novos pets disponíveis para adoção. Serão fornecidas informações sobre o animal, como nome, espécie, raça, idade, características, entre outras.
UC012 - Gerenciar Pets	O voluntário poderá gerenciar os pets cadastrados no sistema. Isso inclui atualizar informações, remover animais do sistema, entre outras ações relacionadas.
UC013 - Gerenciar Visitas	O voluntário poderá gerenciar as visitas agendadas para conhecer os animais disponíveis para adoção. Isso inclui visualizar, confirmar ou cancelar visitas, conforme necessário.
UC014 - Gerenciar Voluntários	O administrador pode gerenciar os voluntários registrados no sistema, o que envolve aprovar ou recusar solicitações, acessar informações dos voluntários e outras ações pertinentes ao gerenciamento desses voluntários.
UC015 - Gerenciar Usuários	O administrador pode gerenciar os usuários registrados no sistema.

3.3.2 Diagrama de Classes

O diagrama de classes, uma representação estrutural da linguagem UML, descreve as classes presentes no sistema, destacando seus atributos, métodos e relações (SILVA; MARTINS; DINIZ, 2017). O diagrama referente a este projeto está disponível no APÊNDICE B.

3.3.3 Diagrama de Sequência

O diagrama de sequência, pertencente à categoria de diagramas de interação da linguagem UML, tem como propósito representar a troca de mensagens entre distintos objetos em um cenário, destacando a ordem temporal dos eventos durante a ação representada (LARMAN, 2007). Os diagramas de sequência do projeto foram desenvolvidos de acordo com as situações delineadas no diagrama de casos de uso. Os respectivos diagramas estão disponíveis para consulta no APÊNDICE E.

3.3.4 Diagrama da Estrutura do Banco de Dados

Na criação do sistema optou-se por utilizar o banco de dados MongoDB, conforme discutido na Seção 2.2.4 deste documento. Ao estruturar as coleções do sistema, levou-se em consideração as informações destinadas às interfaces e os relacionamentos delineados no diagrama de casos de uso.

Os dados são armazenados no MongoDB em documentos, que se assemelham a estruturas de dados flexíveis e são formatados como objetos JSON. Cada documento, por

padrão, contém um identificador exclusivo chamado ObjectId, gerado automaticamente pelo banco de dados.

Esses documentos, com seus ObjectId únicos, são organizados em coleções, uma abstração similar às tabelas encontradas em bancos de dados relacionais. Essa estrutura de coleções e documentos oferece uma flexibilidade notável na modelagem de dados, permitindo formatos variados para cada documento dentro de uma mesma coleção

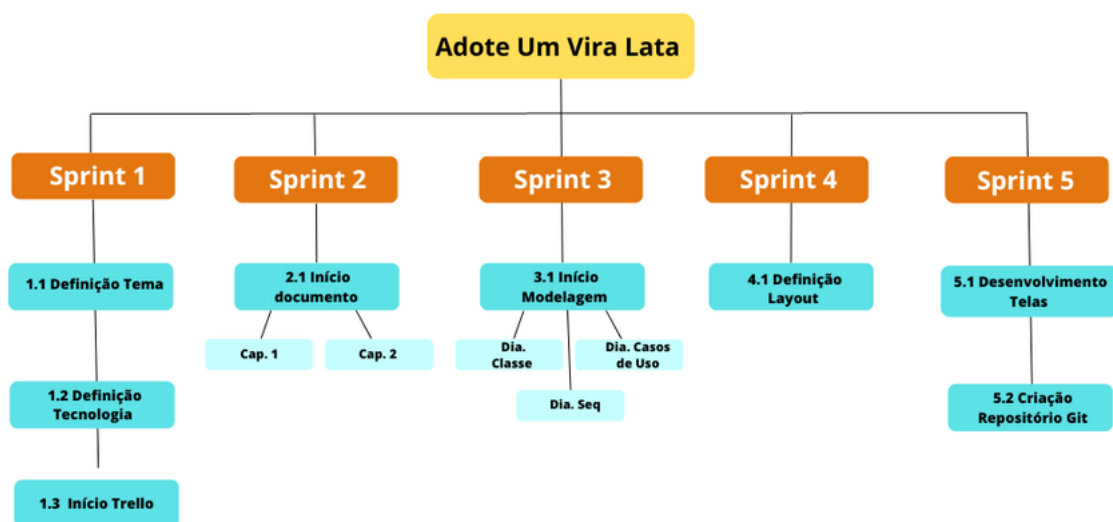
Quando um documento faz referência a um campo de outra coleção, esse campo é armazenado como um atributo do tipo ObjectId, como exemplificado pelo campo “idReferênciaColeção”. Essa referência possibilita a consulta dos dados desse atributo em outras coleções.

O APÊNDICE C apresenta o diagrama estrutural do banco, fornecendo uma representação visual dos modelos das coleções e suas inter-relações.

3.4 PLANO DE ATIVIDADES

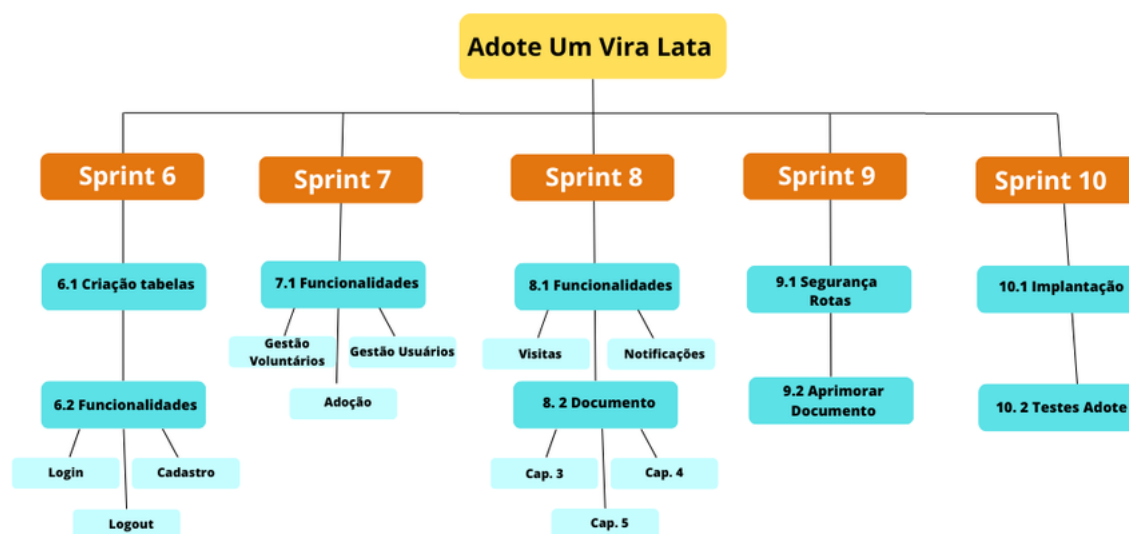
As atividades foram subdivididas em subatividades e organizadas em uma estrutura de divisão do trabalho (WBS, do inglês Work Breakdown Structure), conforme representado na Figura 8 e Figura 9. Essa ferramenta tem como objetivo decompor o projeto em partes menores e hierárquicas, permitindo uma visualização mais detalhada das atividades em cada nível.

Figura 9 – WBS: SPRINT 1 - SPRINT 5



A AUTORA (2023)

Figura 10 – WBS: SPRINT 6 - SPRINT 10



A AUTORA (2023)

3.5 TECNOLOGIAS UTILIZADAS

Para a documentação e desenvolvimento do trabalho, utilizou-se diversas tecnologias, as quais estão definidas a seguir.

3.5.1 Astah

O software escolhido para a modelagem dos dados neste projeto foi o Astah. Essa ferramenta oferece várias versões adequadas para modelar desde sistemas pequenos até os mais complexos, contando com uma ampla variedade de diagramas (ASTAH, 2023).

Para criar os diagramas apresentados neste trabalho, optamos pela versão Astah UML for Students, que concede acesso à versão mais completa da ferramenta mediante cadastro como estudante. A exceção foi o diagrama físico do banco de dados, para o qual utilizamos a versão Astah Professional.

3.5.2 Github

O Github é uma plataforma que hospeda e gerencia sistemas de controle de versão, o git. Assim sendo, o Github promove uma solução remota para a utilização do git, além de proporcionar a possibilidade de documentação de aplicações, comunicação de bugs e até deploys automáticos. É a principal plataforma de hospedagem de código fonte no mundo e conta com mais de 330 milhões de repositórios e 100 milhões de usuários (GITHUB, 2023).

3.5.3 Postman

O Postman é uma API client, ou seja, uma ferramenta que permite a simulação de requisições como cliente. Ele permite aos desenvolvedores testar, compartilhar e até documentar requisições a APIs (Interface de Programação de Aplicação). Ele facilita o desenvolvimento do back-end ao oferecer uma interface gráfica simples e rápida para acessar endpoints (pontos de comunicação) do projeto (POSTMAN, 2021). Ao final de cada requisição o usuário recebe uma resposta do servidor, com informações como status e corpo do retorno. Ele oferece suporte à aplicações RESTful e também ao GraphQL, uma linguagem de consulta de APIs. Também é possível realizar autenticações e habilitar ou desabilitar cookies, pequenos arquivos de texto que podem ser salvos nos navegadores (POSTMAN, 2023).

3.5.4 Visual Studio Code

O Visual Studio Code, ou simplesmente VS Code, é um editor de textos muito leve que oferece suporte para diversas linguagens e arquivos. O VS Code foi utilizado no projeto para o desenvolvimento do back-end e do front-end. Sua grande vantagem é a vasta variedade de extensões, que melhoram a produtividade dos desenvolvedores e rodam em processos paralelos, o que não compromete a velocidade do editor (VSCODE, 2023).

3.5.5 Mern Stack

MERN é uma sigla que representa MongoDB, Express, React e Node, em homenagem às quatro tecnologias essenciais que compõem essa pilha.

O funcionamento desse conjunto envolve o React, responsável pela composição do front-end e pelo envio de requisições via REST no lado do cliente. O Express.js opera em um servidor Node.js para construir o back-end, enquanto o MongoDB é o banco de dados NoSQL utilizado para armazenar os dados em formato JSON (MONGODB, 2023a).

Independentemente da variante escolhida, a abordagem ME(RVA)N é considerada ideal para lidar de forma abrangente com JavaScript e JSON (MONGODB, 2023a).

No projeto de adoção de animais, essa abordagem foi crucial em diferentes áreas. O MongoDB, um banco de dados NoSQL, foi escolhido por sua capacidade de armazenar dados em formato JSON, facilitando o gerenciamento e a recuperação de informações diversas, ideais para os dados de adoção de animais.

O Express.js, aliado ao Node.js, foi empregado para criar o servidor e desenvolver APIs e rotas necessárias para o sistema. Essa combinação permitiu a construção de um back-end sólido, capaz de lidar com as solicitações e a lógica do sistema de adoção.

Por sua vez, o React, uma biblioteca JavaScript, desempenhou um papel fundamental na construção do front-end. Ele possibilitou a criação de uma interface de usuário dinâmica, responsiva e intuitiva para os usuários finais. O React simplificou o desenvolvimento de componentes reutilizáveis, otimizando o processo de desenvolvimento e manutenção do sistema.

Essas tecnologias foram integradas para criar um sistema de adoção de animais moderno, eficiente e com uma experiência de usuário aprimorada tanto para os adotantes quanto para os administradores do sistema.

3.5.6 Ant Design

O Ant Design é uma renomada biblioteca de componentes de interface de usuário (UI) destinada ao React, oferecendo uma extensa gama de elementos prontos para uso no desenvolvimento web. É uma escolha frequente na criação de interfaces de alta qualidade e responsivas, fornecendo uma variedade de componentes bem projetados e comumente utilizados, como botões, formulários, layouts e modais.

Além de disponibilizar componentes prontos para uso, o Ant Design segue diretrizes de design consistentes, promovendo uma estética visual moderna e limpa para os aplicativos. Sua alta customização permite ajustes visuais que se adequam ao estilo específico de cada projeto (ANTD, 2023).

Devido à sua extensa variedade de componentes e uma comunidade ativa, o Ant Design se tornou uma escolha popular entre desenvolvedores que buscam criar interfaces de usuário elegantes e consistentes em aplicativos React. Essa biblioteca não só agrega eficiência ao processo de desenvolvimento, mas também contribui para a padronização e consistência visual, aspectos relevantes em ambientes de desenvolvimento de software para a construção de interfaces de qualidade.

3.5.7 Cloudinary

Foi decidido o uso do Cloudinary para o armazenamento das imagens dos animais registrados no sistema. Essa escolha foi feita devido à eficiência, escalabilidade e flexibilidade oferecidas por essa plataforma, tornando-a uma opção popular para o gerenciamento

de recursos visuais. Isso garante que as imagens permaneçam acessíveis mesmo em situações de alto tráfego ou demanda intensa.

Adicionalmente, o Cloudinary oferece uma variedade de recursos avançados para o gerenciamento de dados visuais, permitindo manipulações, redimensionamentos e otimizações em tempo real, adaptando-se às necessidades específicas do sistema (CLOUDINARY, 2023).

3.5.8 Heroku

Para hospedar o back-end da aplicação de forma simplificada e garantir a robustez e disponibilidade da sua API, optou-se por utilizar o serviço em nuvem Heroku Platform.

A Heroku Platform é uma Plataforma-como-um-Serviço (PaaS) baseada em nuvem, oferecendo um ambiente completo para o desenvolvimento, hospedagem e manutenção de aplicações. Sua vantagem está na eliminação dos custos diretos e na redução da complexidade e inflexibilidade associadas à construção e manutenção de plataformas localmente (IBM, 2021).

A capacidade da Heroku Platform de oferecer escalabilidade, robustez e alta disponibilidade, juntamente com seu suporte a várias tecnologias, a tornaram uma escolha popular na comunidade de desenvolvimento de software, contando atualmente com mais de 13 milhões de aplicativos hospedados (HEROKU, 2023).

A integração com o GIT e o GitHub possibilita a atualização em tempo real da aplicação, permitindo que as melhorias e alterações sejam implementadas rapidamente, beneficiando os usuários finais (HEROKU, 2023).

3.5.9 Hardware

O sistema Adote um Vira Lata foi desenvolvido utilizando um Macbook Pro M1. Este hardware proporcionou um ambiente de desenvolvimento ágil e eficiente, aproveitando a potência e recursos do processador M1 da Apple para a execução das atividades de programação, teste e depuração do sistema.

3.6 DESENVOLVIMENTO DO PROJETO

Com base na estimativa de esforço estabelecida para o desenvolvimento deste projeto, foram definidas 5 sprints, cada uma com duração de duas semanas. No entanto, é importante mencionar que houve uma pausa devido o recesso de julho. Nesta seção, apresentarei os artefatos construídos e as decisões tomadas ao longo da implementação do sistema proposto. Faremos uma descrição detalhada das etapas e dos resultados obtidos durante o processo de construção.

3.6.1 Sprint 1

A primeira Sprint ocorreu entre 01/06/2023 e 15/06/2023. Durante esse período, dediquei esforços à definição do tema central do sistema, analisando as tecnologias mais adequadas para sua implementação. Além disso, essa fase inicial envolveu a criação e organização do documento principal do projeto, bem como a estruturação do quadro no Trello, onde registrei as tarefas e etapas a serem desenvolvidas.

3.6.2 Sprint 2

Na segunda Sprint, que abrangeu o período de 15/06/2023 a 29/06/2023, o foco concentrou-se na elaboração dos capítulos 1 e 2 do documento. Durante esse tempo, detalhei informações sobre a introdução e o embasamento teórico do sistema proposto, abordando aspectos conceituais e as bases teóricas que sustentam o desenvolvimento do projeto.

3.6.3 Sprint 3

Na terceira Sprint, de 10/08/2023 a 24/08/2023, dediquei-me à modelagem do sistema. Nessa fase, foquei na criação de diagramas e especificações para entender a estrutura e as interações do sistema. Defini requisitos detalhados, identificando entidades, fluxos de dados e relações cruciais para o projeto. Essa etapa foi essencial para mapear os componentes centrais, proporcionando uma visão mais clara do escopo e das funcionalidades a serem desenvolvidas.

3.6.4 Sprint 4

Durante a quarta Sprint, que compreendeu o período de 24/08/2023 a 07/09/2023, concentrei meus esforços na criação do layout do sistema por meio do Whimsical. Essa etapa foi crucial para transformar os requisitos e diagramas em protótipos visuais interativos. Utilizei o Whimsical para esboçar as telas, definir fluxos de navegação e proporcionar uma representação tangível das funcionalidades do sistema. Esse processo facilitou a visualização e a validação das ideias, preparando o terreno para a implementação eficiente das interfaces do projeto.

3.6.5 Sprint 5

Durante a quinta Sprint, entre 07/09/2023 e 21/09/2023, iniciei o desenvolvimento das primeiras telas do sistema e estabeleci o repositório no Git para gerenciar o código-fonte

do projeto. Esse período marcou o começo da tradução dos protótipos do Whimsical para interfaces funcionais e a organização da base do código.

3.6.6 Sprint 6

Na sexta Sprint, entre 21/09/2023 e 05/10/2023, concentrei-me na criação das primeiras tabelas no MongoDB para armazenamento de dados, estabelecendo a base do banco para o sistema. Além disso, desenvolvi as funcionalidades de login, cadastro e logout, possibilitando que os usuários se autentiquem e se inscrevam na plataforma.

3.6.7 Sprint 7

Na sétima Sprint, entre 05/10/2023 e 19/10/2023, meu foco esteve na implementação da funcionalidade central do sistema: a adoção. Dediquei-me ao desenvolvimento de todas as etapas do processo de adoção de animais, desde a visualização dos pets disponíveis até o registro e confirmação da adoção pelos usuários. Além disso, trabalhei na implementação da gestão de usuários pelo administrador, proporcionando um controle mais eficiente e detalhado dos perfis na plataforma.

3.6.8 Sprint 8

Na oitava Sprint, entre 19/10/2023 e 02/11/2023, concentrei esforços na funcionalidade de agendamento de visitas aos animais disponíveis para adoção. Desenvolvi a interação com o calendário, o registro das visitas agendadas e a confirmação pela instituição responsável pela adoção. Essa adição aprimorou a interação na plataforma.

3.6.9 Sprint 9

Durante a nona Sprint, entre 02/11/2023 e 16/11/2023, concentrei meus esforços na melhoria da segurança das rotas da aplicação. Reforcei a proteção das rotas sensíveis, implementando medidas adicionais de segurança para garantir que apenas usuários autorizados pudessem acessar determinadas partes do sistema. Além disso, reservei um tempo para revisar e aprimorar o documento do TCC, garantindo sua consistência e alinhamento com as últimas atualizações e evoluções do projeto.

3.6.10 Sprint 10

Durante a décima Sprint, entre 16/11/2023 e 01/12/2023, meu foco esteve nos ajustes finais e na verificação minuciosa do sistema. Realizei testes finais, refinando o layout e a experiência do usuário, além de revisar o código para garantir qualidade e corrigir

possíveis falhas. Esta fase também englobou a implantação do sistema, preparando-o para a entrega final.

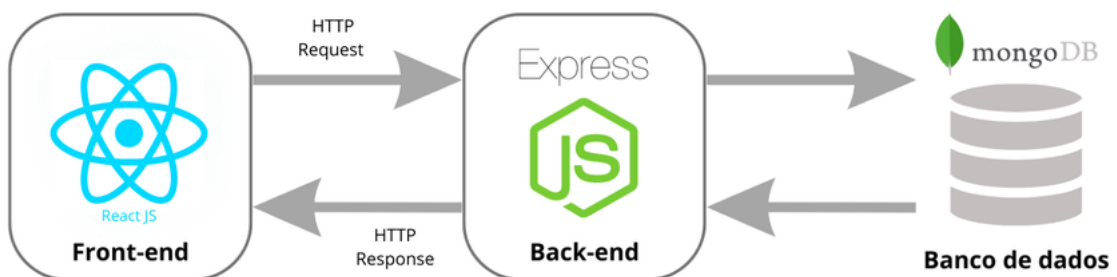
4 APRESENTAÇÃO DO ADOTE UM VIRA LATA

Esta seção apresenta a arquitetura básica do sistema Adote um Vira Lata e as funcionalidades implementadas, por meio das telas desenvolvidas e seus fluxos.

4.1 ARQUITETURA DO SISTEMA

O sistema foi desenvolvido em conformidade com os diagramas de modelagem do sistema e adotou a MERN Stack como conjunto de ferramentas. O front-end foi implementado em React, enquanto o back-end foi construído utilizando Node.js em conjunto com o Express.js. O armazenamento de dados foi realizado no MongoDB. A FIGURA 7 ilustra a arquitetura fundamental do sistema.

Figura 11 – ARQUITETURA DO SISTEMA



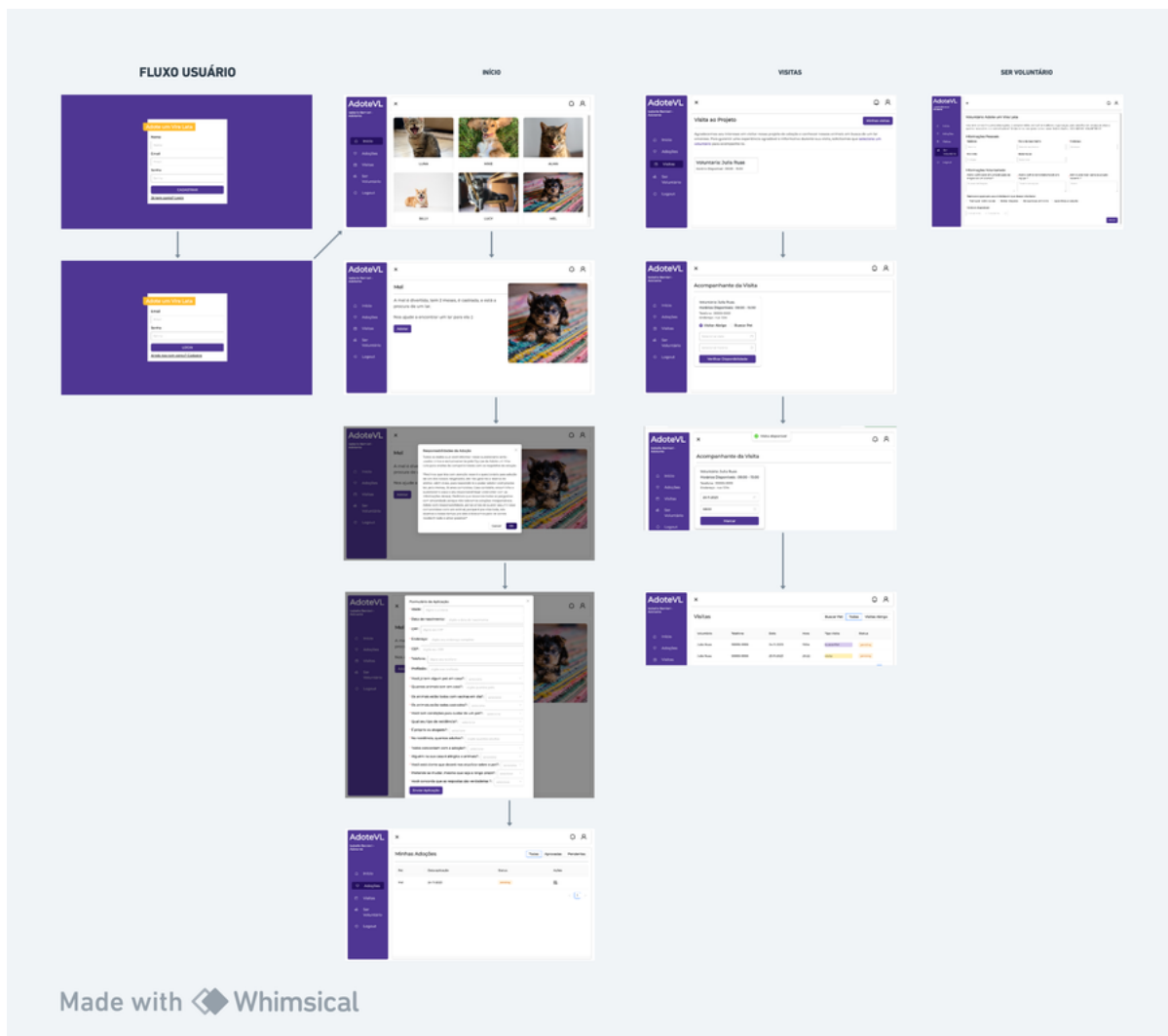
A AUTORA (2023)

4.2 APRESENTAÇÃO DAS FUNCIONALIDADES

O sistema “Adote um Vira-Lata” é uma plataforma online dedicada à administração e promoção de adoção de animais, conforme ilustrado no fluxograma do APÊNDICE F. Nele, destacam-se três principais participantes: os Voluntários da Entidade Protetora, responsáveis pelo cadastro dos animais disponíveis para adoção; os Potenciais Adotantes, que se inscrevem para adotar um animal; e, por fim, os Administradores, responsáveis por validar os voluntários e atualizar outros elementos do sistema.

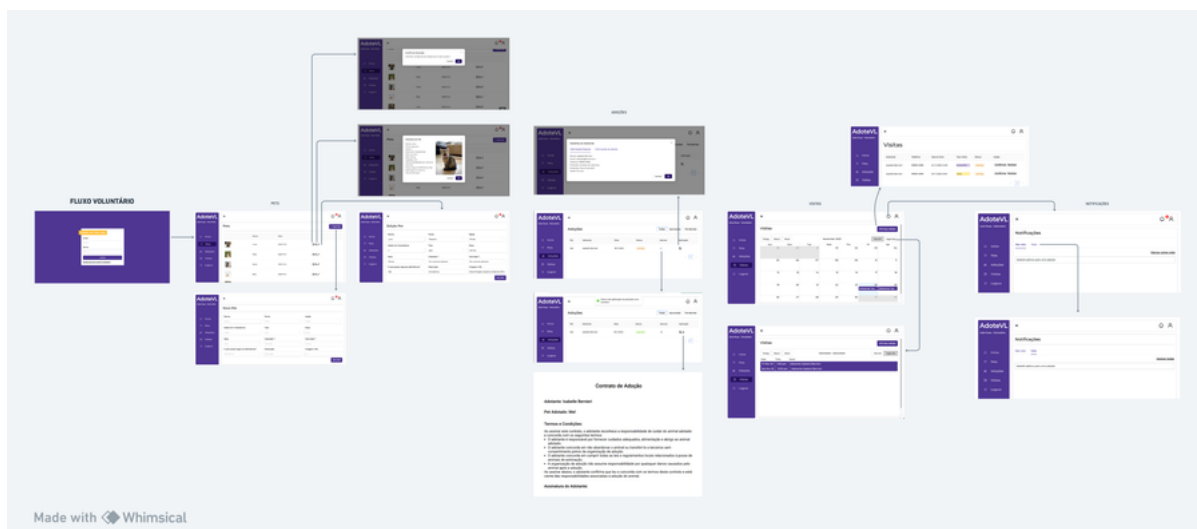
Em complemento, as FIGURAS 12 a 14 têm o propósito de exibir os diagramas de tela do sistema, representando a visão do usuário, voluntário e administrador, respectivamente.

Figura 12 – DIAGRAMA DE TELAS USUÁRIO



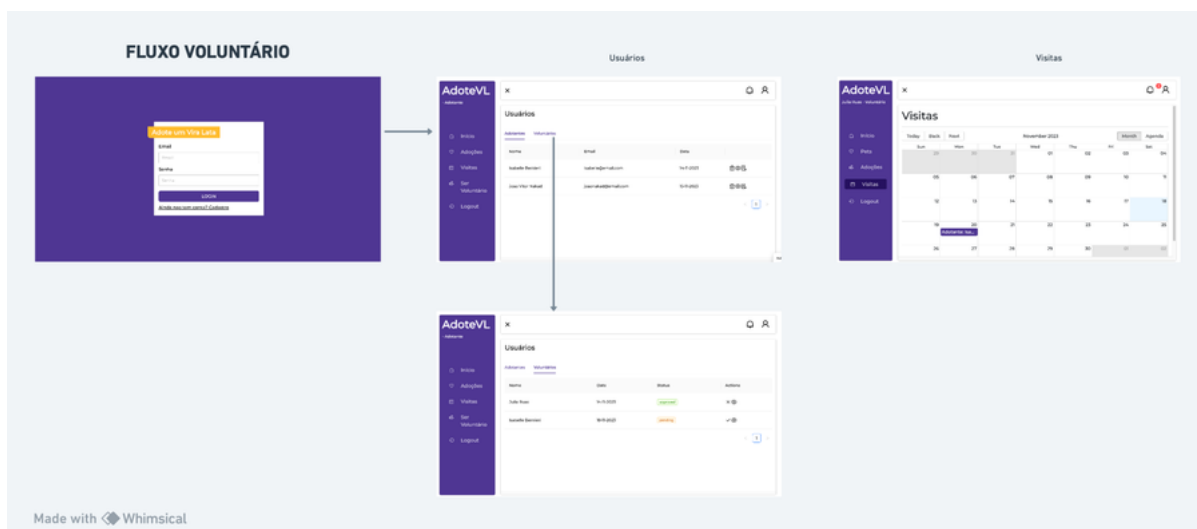
A AUTORA (2023)

Figura 13 – DIAGRAMA DE TELAS VOLUNTARIO



A AUTORA (2023)

Figura 14 – DIAGRAMA DE TELAS ADMINISTRADOR



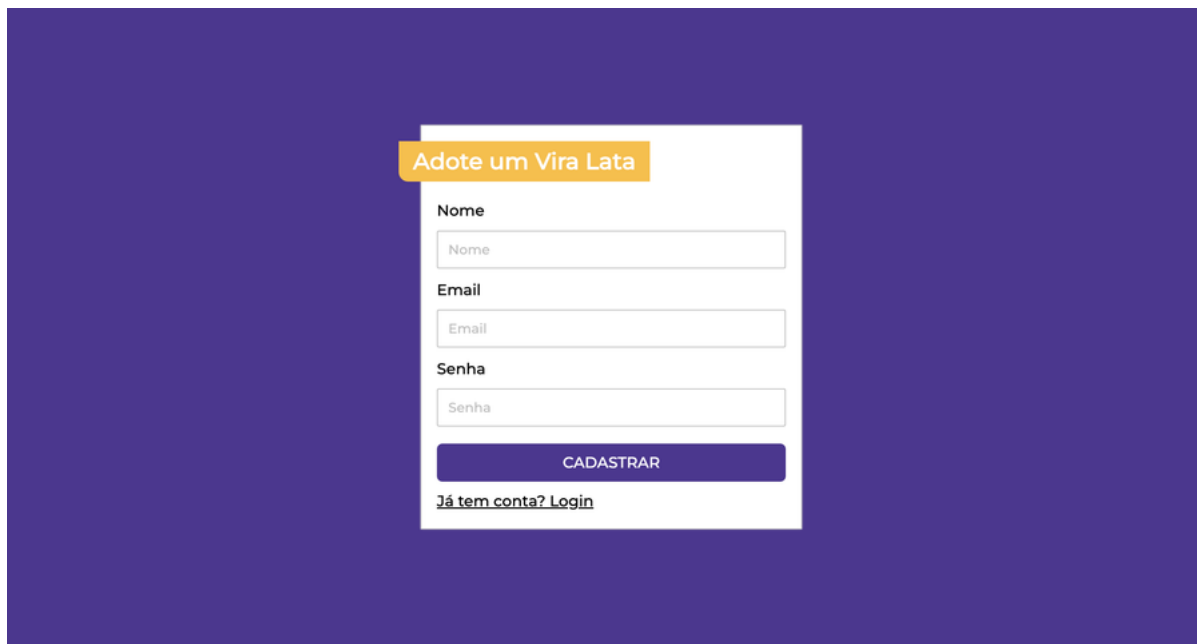
A AUTORA (2023)

A seguir, são detalhadas as funcionalidades do sistema, alinhadas aos requisitos e à modelagem do projeto.

4.2.1 Cadastro

Ao acessar o endereço do sistema, caso o usuário não esteja logado, ele apresentará a tela de cadastro, como mostra a FIGURA 15. A tela contém um formulário para que o usuário preencha com as informações de Nome, Email e Senha.

Figura 15 – TELA DE CADASTRO



The image shows a registration form titled "Adote um Vira Lata" on a dark purple background. The form is white and contains the following fields and elements:

- Nome**: A text input field with the placeholder "Nome".
- Email**: A text input field with the placeholder "Email".
- Senha**: A text input field with the placeholder "Senha".
- CADASTRAR**: A dark purple button with white text.
- Já tem conta? Login**: A link in blue text below the button.

A AUTORA (2023)

4.2.2 Login

Caso o cadastro seja bem-sucedido, o sistema redirecionará para a tela de login, como mostra a FIGURA 16. O usuário deverá preencher com Email e Senha para que consiga acessar as outras funcionalidades do sistema.

Figura 16 – TELA DE LOGIN



The image shows a login form titled "Adote um Vira Lata" on a dark purple background. The form is white and contains the following fields and elements:

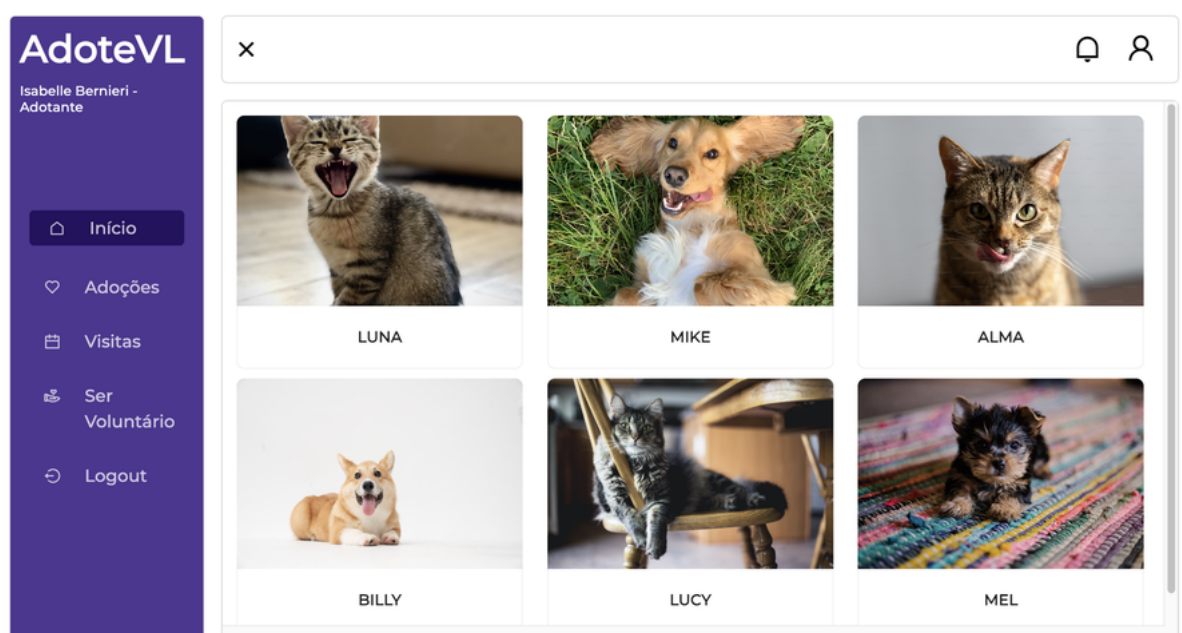
- Email**: A text input field with the placeholder "Email".
- Senha**: A text input field with the placeholder "Senha".
- LOGIN**: A dark purple button with white text.
- Ainda nao tem conta? Cadastro**: A link in blue text below the button.

A AUTORA (2023)

4.2.3 Visualização Pets

Se o login for bem-sucedido, o sistema direcionará automaticamente para a tela inicial correspondente a listagem dos pets disponíveis, como mostra a FIGURA 17.

Figura 17 – TELA DE VISUALIZAÇÃO DE PETS

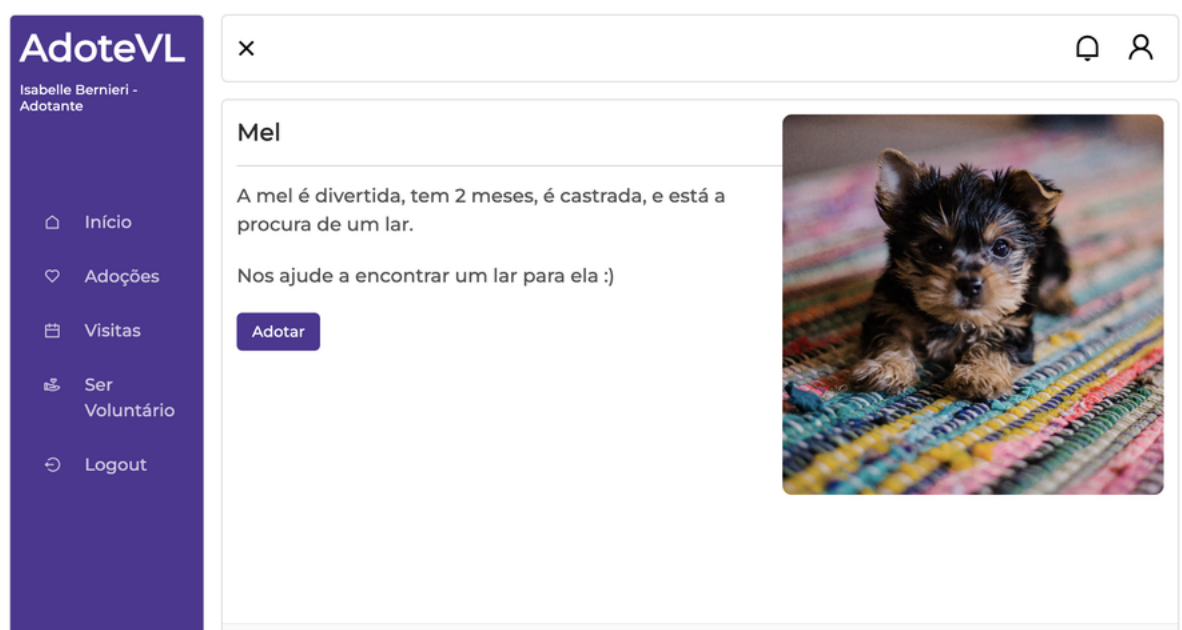


A AUTORA (2023)

4.2.4 Detalhes Pets

Caso o usuário clique em algum pet listado, a tela de detalhes de pets será exibida e mostrará informações específicas sobre o pet clicado. Isso incluirá detalhes como nome, idade, raça, descrição e outras características relevantes do animal disponível para adoção, conforme a FIGURA 18.

Figura 18 – TELA DE DETALHES DE PETS



A AUTORA (2023)

4.2.5 Adotar

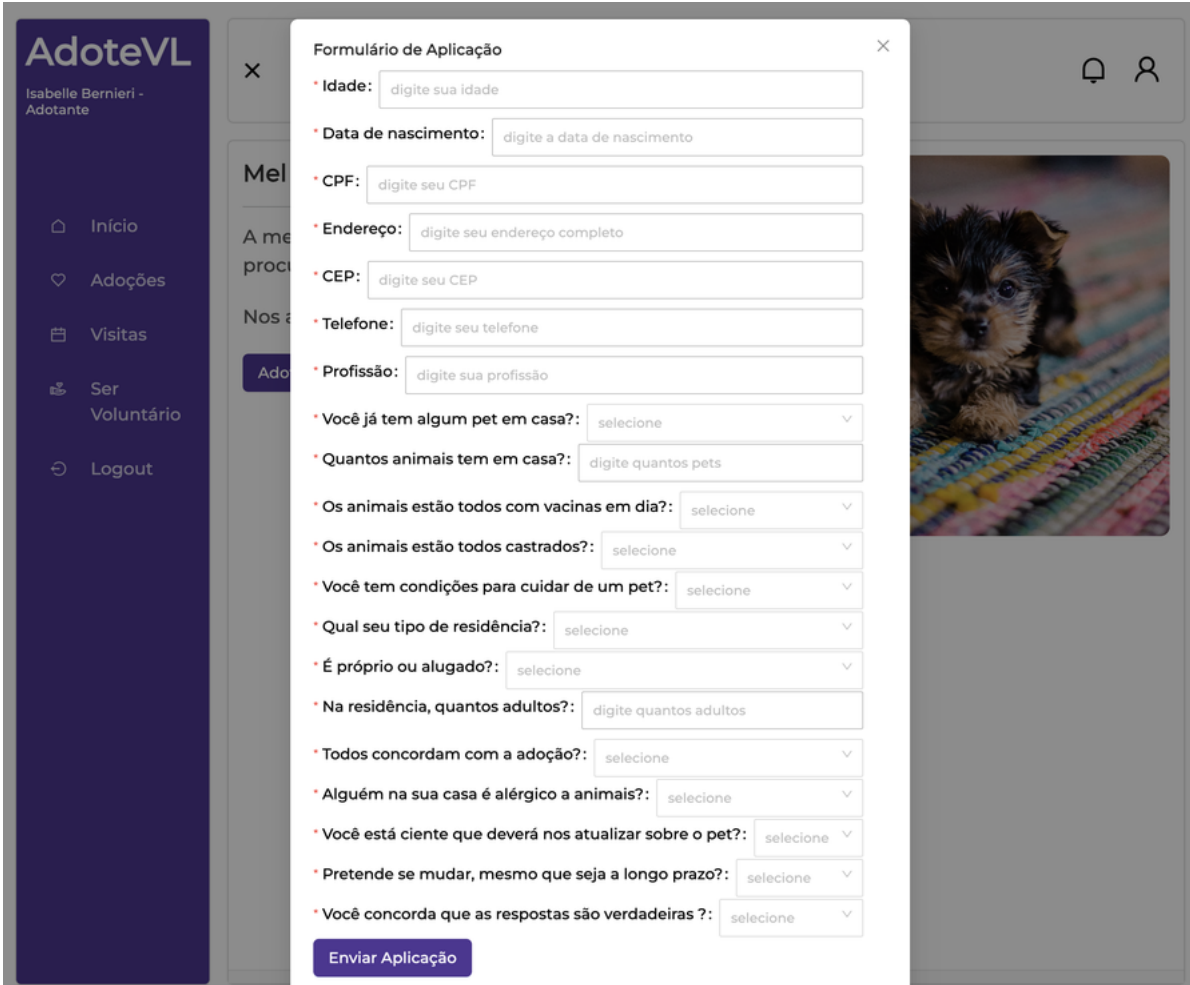
Quando o usuário opta por adotar um animal, é exibido um modal de adoção, dando início ao processo. Esse modal apresentará as responsabilidades da adoção (FIGURA 19), e fornecerá campos para preenchimento de informações essenciais (FIGURA 20). Entre elas, dados pessoais, questões sobre o ambiente de moradia e critérios para a adoção do animal selecionado.

Figura 19 – MODAL DE RESPONSABILIDADES ADOÇÃO



A AUTORA (2023)

Figura 20 – MODAL DE FORMULÁRIO DE ADOÇÃO



Formulário de Aplicação

* Idade:

* Data de nascimento:

* CPF:

* Endereço:

* CEP:

* Telefone:

* Profissão:

* Você já tem algum pet em casa?:

* Quantos animais tem em casa?:

* Os animais estão todos com vacinas em dia?:

* Os animais estão todos castrados?:

* Você tem condições para cuidar de um pet?:

* Qual seu tipo de residência?:

* É próprio ou alugado?:

* Na residência, quantos adultos?:

* Todos concordam com a adoção?:

* Alguém na sua casa é alérgico a animais?:

* Você está ciente que deverá nos atualizar sobre o pet?:

* Pretende se mudar, mesmo que seja a longo prazo?:

* Você concorda que as respostas são verdadeiras?:

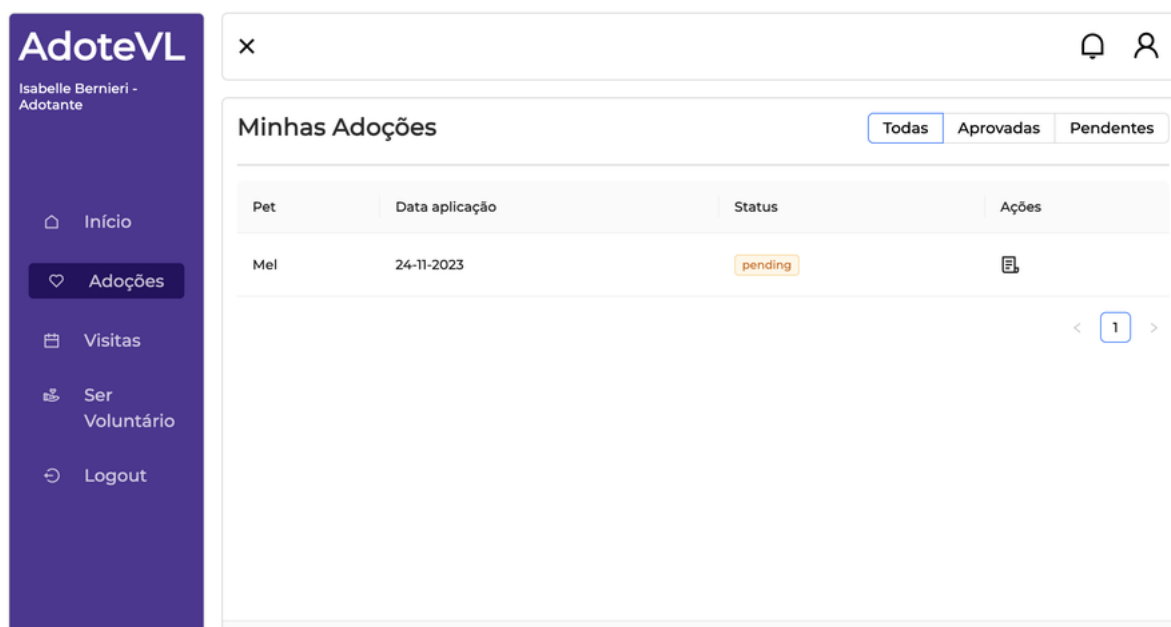
Enviar Aplicação

A AUTORA (2023)

4.2.6 Visualizar Adoções

Na tela de “Minhas Adoções”, o usuário pode visualizar o status de suas solicitações de adoção. Isso inclui adoções pendentes, recusadas ou aprovadas. Cada pet terá um indicativo claro do status da solicitação para que o usuário acompanhe facilmente o andamento de cada uma delas, como a FIGURA 21.

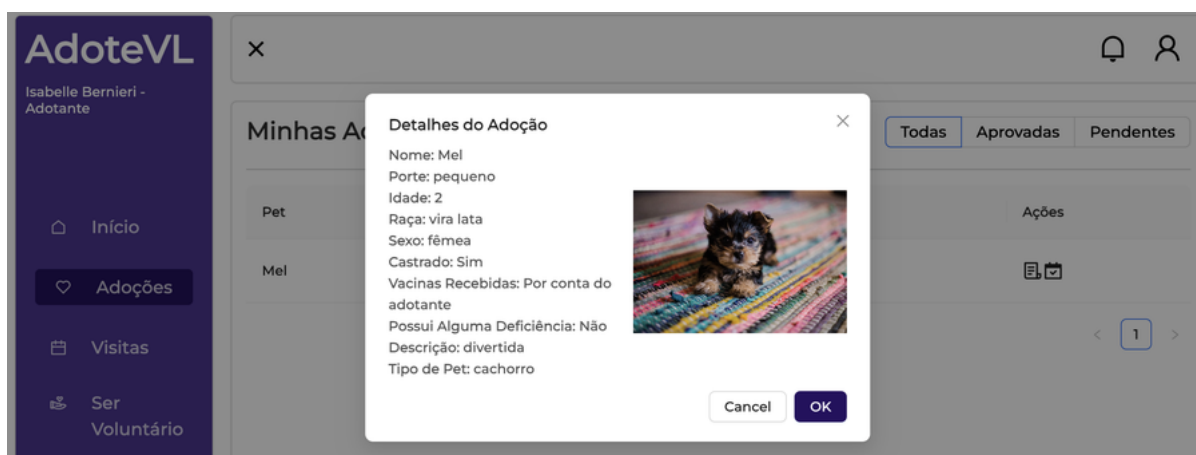
Figura 21 – TELA DE VISUALIZAÇÃO DE ADOÇÕES



A AUTORA (2023)

Se o usuário clicar no ícone na coluna de ações, terá acesso aos detalhes da adoção por meio de um modal, como mostra a FIGURA 22.

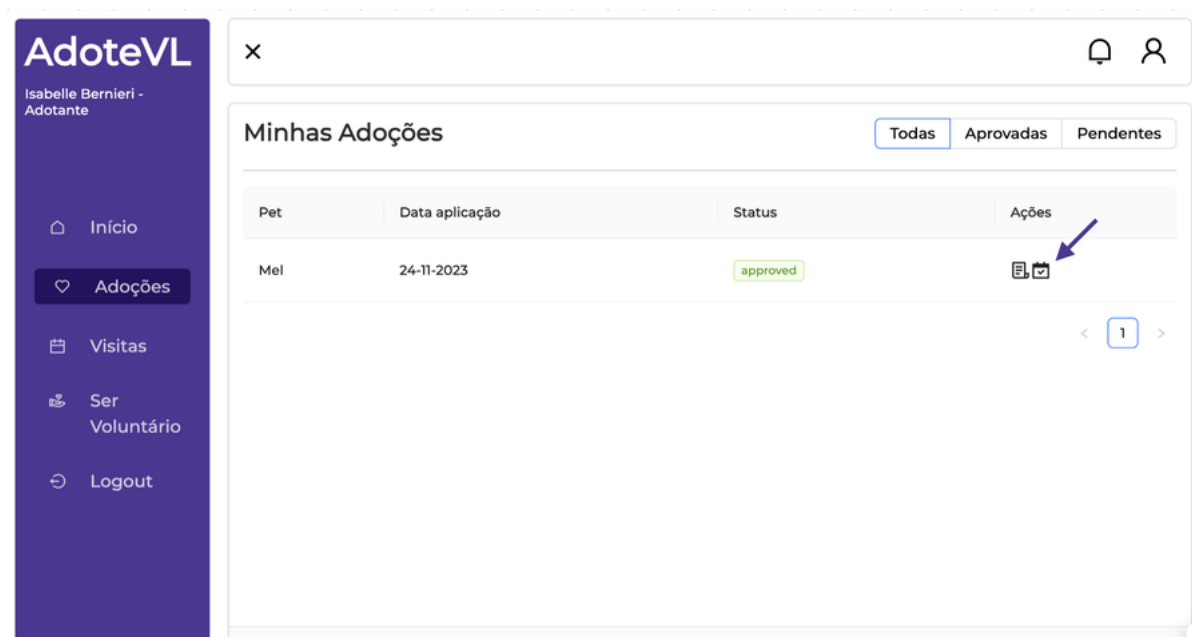
Figura 22 – MODAL DETALHES ADOÇÃO



A AUTORA (2023)

Depois de receber a aprovação para a adoção, o usuário pode acessar esta seção para agendar o momento específico de encontrar e levar seu novo companheiro para casa. Ao clicar no ícone do calendário, será direcionado para a tela de visitas, onde poderá selecionar a data desejada para buscar o pet adotado, como mostra a FIGURA 23.

Figura 23 – TELA DE VISUALIZAÇÃO DE ADOÇÕES

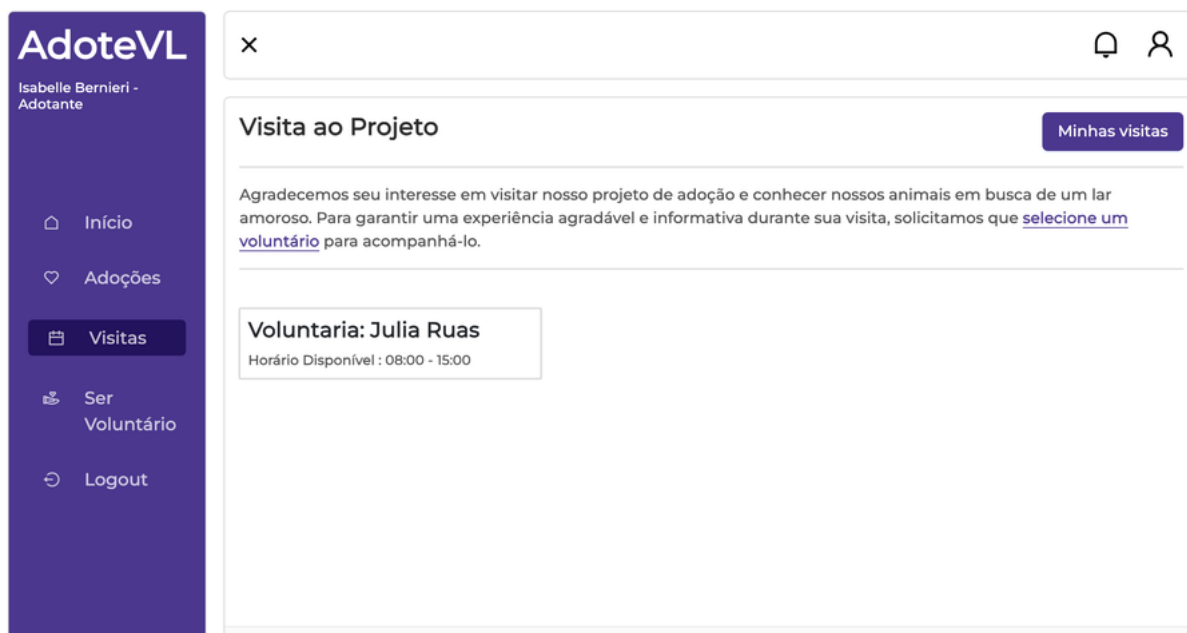


A AUTORA (2023)

4.2.7 Agendar Visitar

Na tela de “Visita ao Projeto”, os usuários podem selecionar um voluntário disponível para acompanhá-los na visita ao abrigo da entidade protetora. Esta tela (FIGURA 24) exibirá uma lista de voluntários disponíveis, com detalhes sobre cada um deles, como nome e disponibilidade. Os usuários poderão selecionar o voluntário de sua preferência e agendar a visita conforme a disponibilidade listada.

Figura 24 – TELA DE AGENDAMENTO DE VISITAS

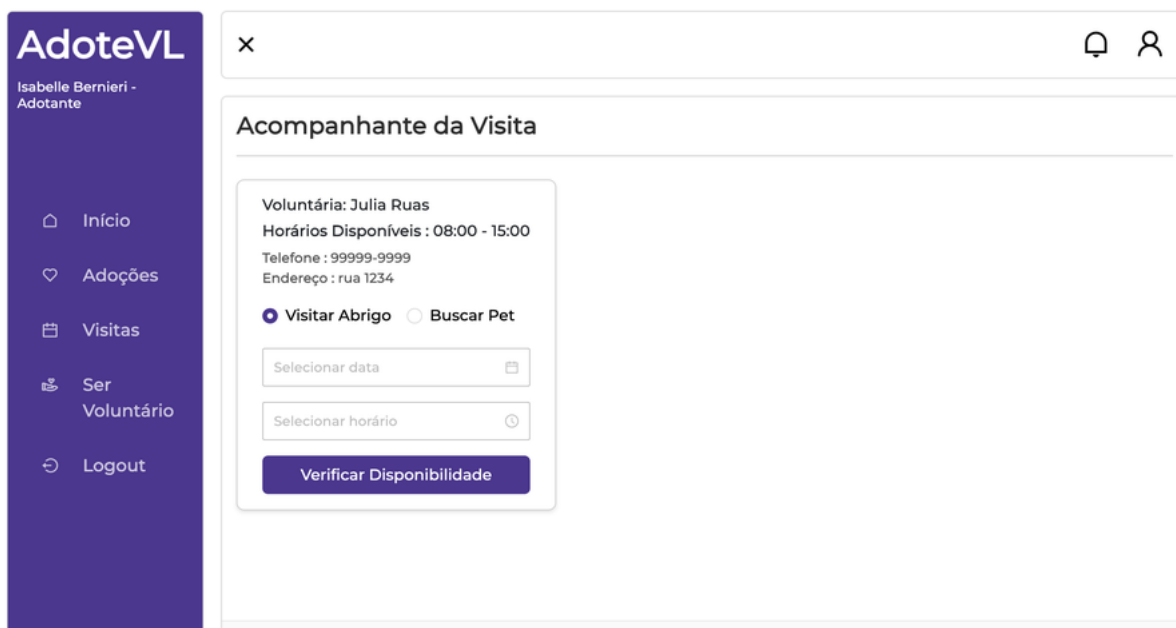


A AUTORA (2023)

4.2.8 Acompanhante de Visita

Ao selecionar o voluntário para acompanhamento, os usuários serão direcionados à tela de “Acompanhante de Visita” (FIGURA 25). Neste espaço, terão acesso a informações detalhadas sobre o voluntário escolhido para acompanhar a visita ao abrigo da entidade protetora. A tela apresentará dados como nome, informações de contato e permitirá a seleção de data e horário para a visita. Além disso, o usuário deverá selecionar se irá visitar o abrigo ou buscar o pet que foi adotado.

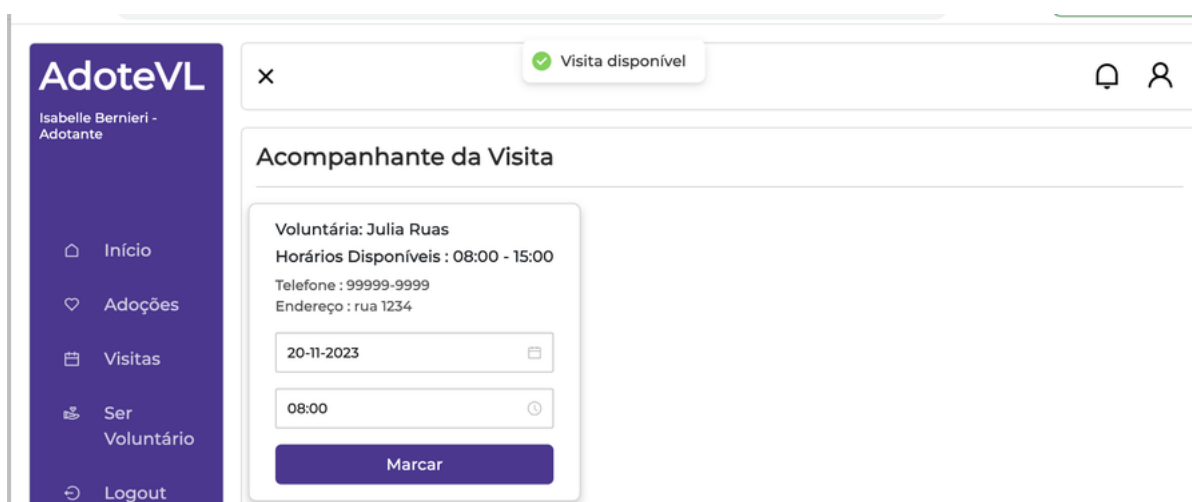
Figura 25 – TELA DO ACOMPANHANTE DA VISITA



A AUTORA (2023)

Ao clicar no botão “Verificar Disponibilidade”, o sistema verificará a agenda do voluntário e, se houver disponibilidade, um pop-up surgirá alertando sobre a disponibilidade e permitindo a confirmação do agendamento da visita (FIGURA 26).

Figura 26 – TELA DO ACOMPANHANTE DA VISITA

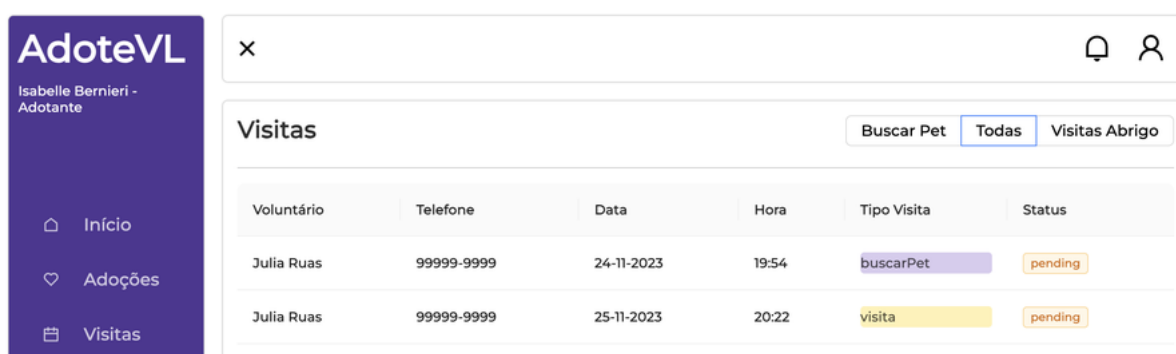


A AUTORA (2023)

4.2.9 Visualizar Visitas

Na tela de visualização de visitas, os usuários podem visualizar e gerenciar suas visitas agendadas. Eles têm a opção de filtrar entre as visitas agendadas para buscar o animal adotado e aquelas com o objetivo de conhecer o abrigo dos pets. Essa funcionalidade permite um acompanhamento mais organizado e específico das atividades planejadas, como mostra a FIGURA 27.

Figura 27 – TELA DE VISUALIZAR VISITAS



Voluntário	Telefone	Data	Hora	Tipo Visita	Status
Julia Ruas	99999-9999	24-11-2023	19:54	buscarPet	pending
Julia Ruas	99999-9999	25-11-2023	20:22	visita	pending

A AUTORA (2023)

4.2.10 Ser Voluntário

Na tela de “Ser voluntário”, os usuários podem encontrar informações abrangentes sobre como se tornar um voluntário na plataforma. Aqui, deverão preencher o formulário de inscrição e assim aguardar a análise do administrador para aprovação, como mostra a FIGURA 28.

Figura 28 – TELA DE SER VOLUNTÁRIO

The screenshot shows the 'Ser Voluntário' (Be a Volunteer) form in the AdoteVL application. The interface includes a purple sidebar with the AdoteVL logo and navigation options: Início, Adoções, Visitas, Ser Voluntário (highlighted), and Logout. The main content area is titled 'Voluntário Adote um Vira Lata' and contains a motivational message: 'Você tem tempo livre, ama cães e gatos, é comprometido, pontual, tem atitude, organização, sabe trabalhar em equipe, se adapta quando necessário e se comunica bem? Então venha nos ajudar com o nosso lindo trabalho... VEM SER UM VOLUNTÁRIO!!'. The form is divided into two main sections: 'Informações Pessoais' and 'Informações Voluntariado'. The 'Informações Pessoais' section includes fields for 'Telefone', 'Data de nascimento', 'Endereço', 'Profissão', and 'Rede Social'. The 'Informações Voluntariado' section includes three text areas: 'Como você lidaria em uma situação de resgate de um animal?', 'Como você se sente trabalhando em equipe?', and 'Motivo para fazer parte do projeto AdoteVL?'. Below these are checkboxes for 'Assinale a opção para as atividades em que deseja voluntariar' with options: 'Postagem redes sociais', 'Retirar doações', 'Recepcionar visitantes', and 'Apoio feira de adoção'. At the bottom, there is a 'Horários disponíveis' section with a date and time range selector. A blue 'enviar' button is located at the bottom right of the form.

A AUTORA (2023)

4.2.11 Perfil

A tela do perfil do usuário apresenta informações personalizadas, como nome e email. Isso permite ao usuário editar esses detalhes e clicar no botão “Salvar” para confirmar as alterações, como mostra na FIGURA 29.

Figura 29 – TELA DE PERFIL

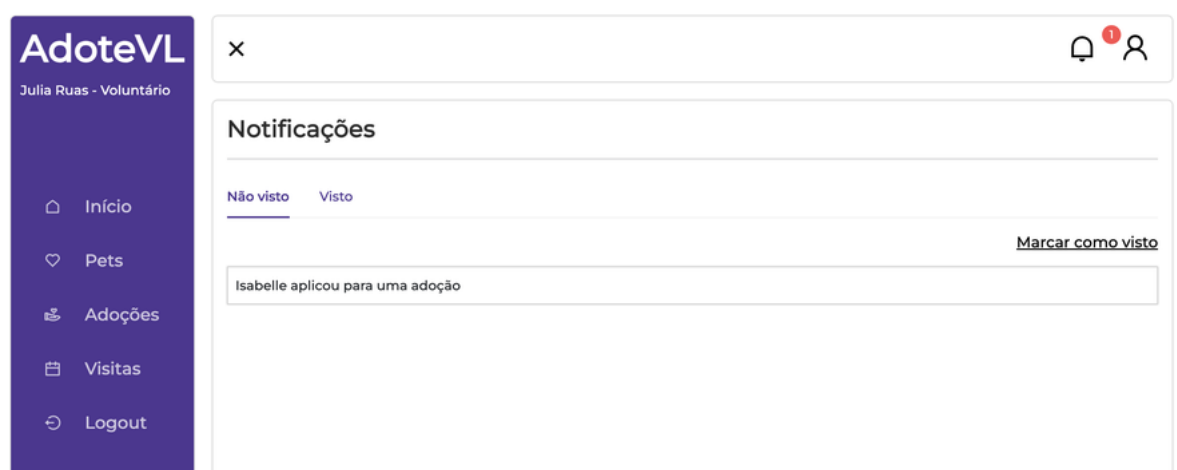
The screenshot shows the 'Perfil' (Profile) page in the AdoteVL application. The interface includes the same purple sidebar as Figure 28. The main content area is titled 'Perfil' and contains a section for 'Informações Pessoais'. This section includes two input fields: 'Nome' (Name) with the value 'Isabelle Bernieri' and 'Email' with the value 'isabelle@email.com'. A blue 'salvar' button is located at the bottom right of the form.

A AUTORA (2023)

4.2.12 Notificações

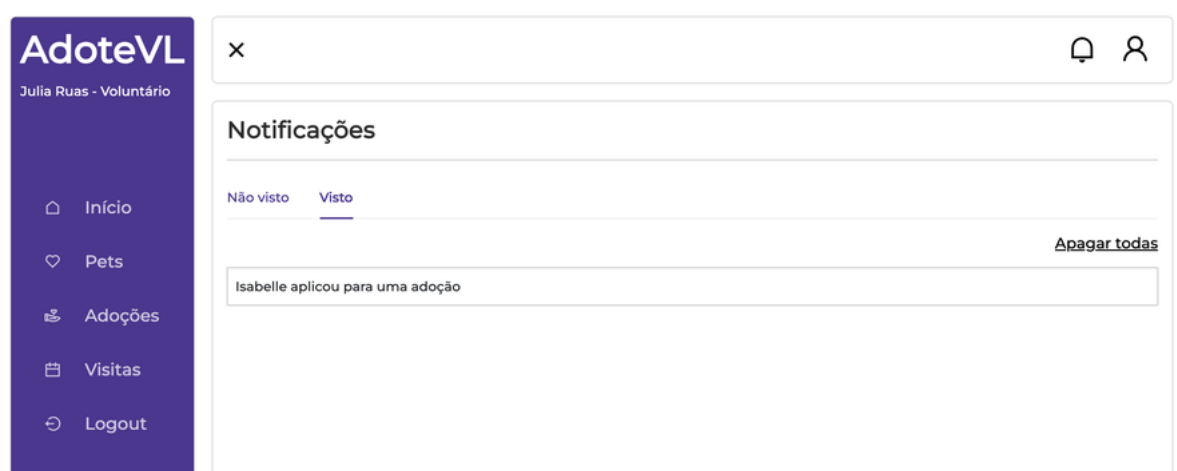
A tela de notificações exibe abas separadas para notificações não vistas (FIGURA 30) e vistas (FIGURA 31). Os usuários podem marcar as não vistas como vistas e deletá-las conforme necessário para organizar suas interações na plataforma.

Figura 30 – TELA DE NOTIFICAÇÕES - ABA NÃO VISTO



A AUTORA (2023)

Figura 31 – TELA DE NOTIFICAÇÕES - ABA VISTO



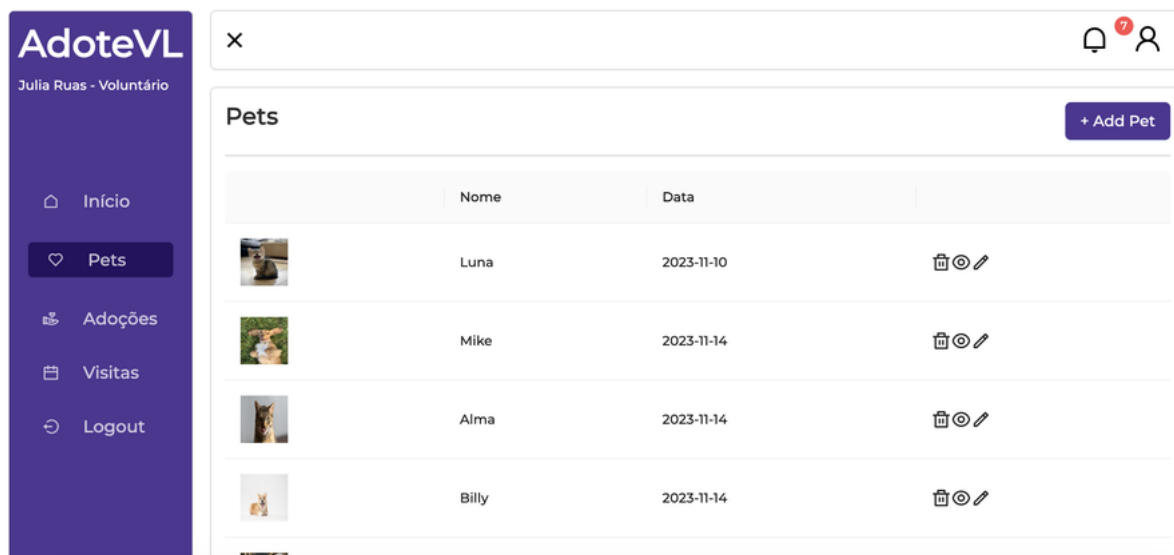
A AUTORA (2023)

4.2.13 Gestão Pets

A tela de “Gestão de Pets” é onde os voluntários podem visualizar, adicionar, editar ou excluir pets cadastrados. Eles têm acesso a uma lista de pets disponíveis para adoção e

podem realizar ações como visualizar detalhes, adicionar novos pets, editar informações existentes ou remover pets do sistema, como mostra a FIGURA 32.

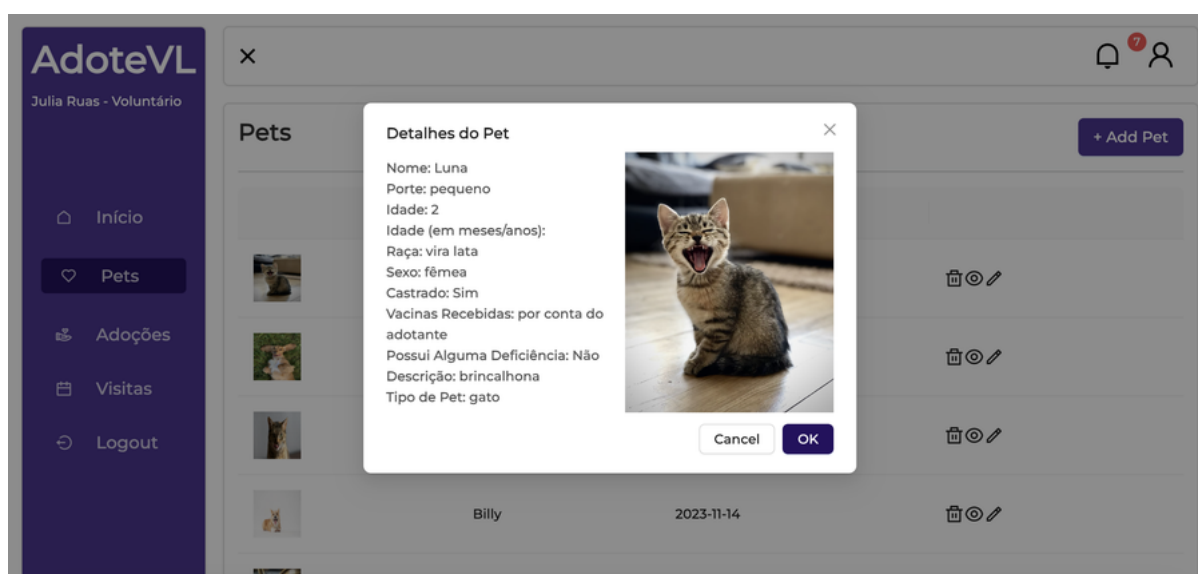
Figura 32 – TELA GESTÃO DE PETS



AUTORA (2023)

Ao clicar no ícone de visualizar, um modal será aberto (FIGURA 33), exibindo informações detalhadas sobre o pet selecionado. Essas informações incluirão detalhes como nome, idade, raça, descrição e outras características relevantes do animal disponível para adoção.

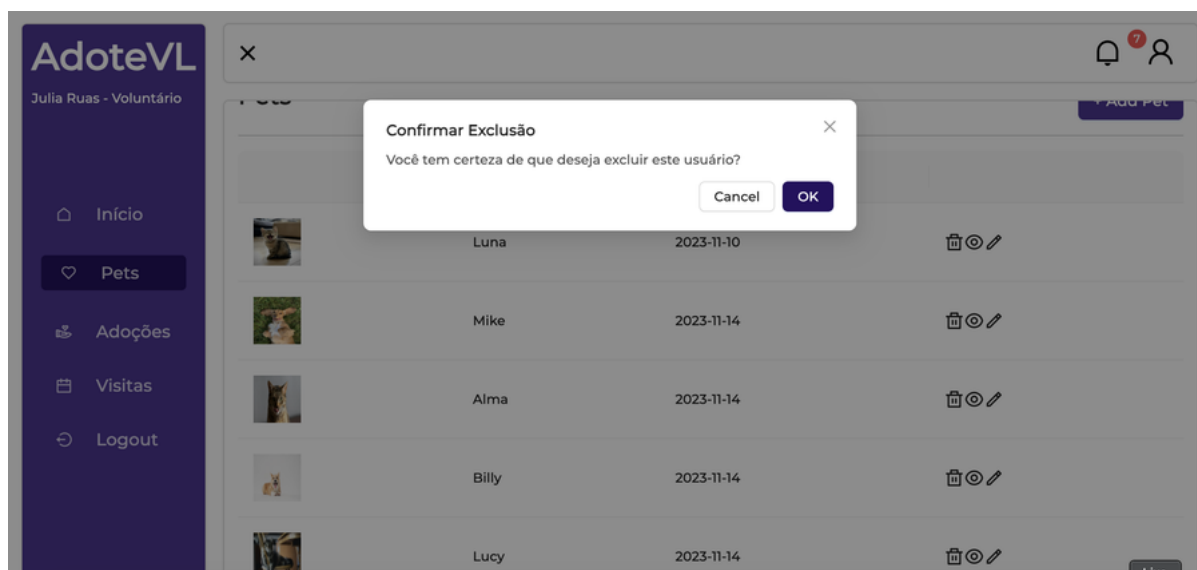
Figura 33 – MODAL DE VISUALIZAÇÃO DE PET



A AUTORA (2023)

Ao clicar no ícone de deletar o pet, um modal de confirmação (FIGURA 34) será exibido para garantir a exclusão do animal selecionado. Esse modal solicitará a confirmação do usuário antes de prosseguir com a exclusão do registro do pet.

Figura 34 – MODAL DE CONFIRMAÇÃO DE EXCLUSÃO



A AUTORA (2023)

Ao clicar no ícone de editar, o usuário será direcionado para a tela de edição (FIGURA 35), onde encontrará um formulário preenchido com as informações atuais do pet. Isso permitirá fazer modificações nos detalhes do animal, como nome, idade, raça, descrição e outras características relevantes.

Figura 35 – TELA DE EDITAR PET

A imagem mostra a tela de edição de um pet no sistema AdoteVL. O formulário é intitulado 'Edição Pet' e contém os seguintes campos: 'Nome' (Luna), 'Porte' (Pequeno), 'Idade' (Filhote), 'Idade em meses/anos' (2), 'Tipo' (gato), 'Raça' (vira lata), 'Sexo' (Fêmea), 'Castrado?' (Por conta do adotante), 'Vacinado?' (Por conta do adotante), 'O pet possui alguma deficiência?' (Não) e 'Descrição' (brincalhona). O campo 'Imagem URL' contém o endereço: https://images.unsplash.com/photo-15741. No canto inferior direito, há um botão 'SALVAR'.

A AUTORA (2023)

Ao clicar no ícone de adicionar pet, o usuário será direcionado para a tela de adição (FIGURA 36), onde encontrará um formulário vazio para preencher com informações sobre um novo animal a ser cadastrado. Isso incluirá campos para inserir nome, idade, raça, descrição e outras características relevantes do novo pet.

Figura 36 – TELA DE ADICIONAR PET

A captura de tela mostra a interface de usuário para adicionar um novo pet. No topo, há o logotipo "AdoteVL" e o nome do usuário "Julia Ruas - Voluntário". À esquerda, há um menu lateral com opções: Início, Pets, Adoções, Visitas e Logout. O formulário principal, intitulado "Novo Pet", contém os seguintes campos:

- Nome:** Campo de texto para o nome do pet.
- Porte:** Menu suspenso para selecionar o porte.
- Idade:** Menu suspenso para selecionar a idade.
- Idade em meses/anos:** Campo de texto para a idade em meses/anos.
- Tipo:** Menu suspenso para selecionar o tipo de pet.
- Raça:** Campo de texto para a raça.
- Sexo:** Menu suspenso para selecionar o sexo.
- Castrado?:** Menu suspenso para selecionar se o pet é castrado.
- Vacinado?:** Menu suspenso para selecionar se o pet foi vacinado.
- O pet possui alguma deficiência?:** Menu suspenso para selecionar se o pet possui alguma deficiência.
- Descrição:** Campo de texto para a descrição do pet.
- Imagem URL:** Campo de texto para o URL da imagem do pet.

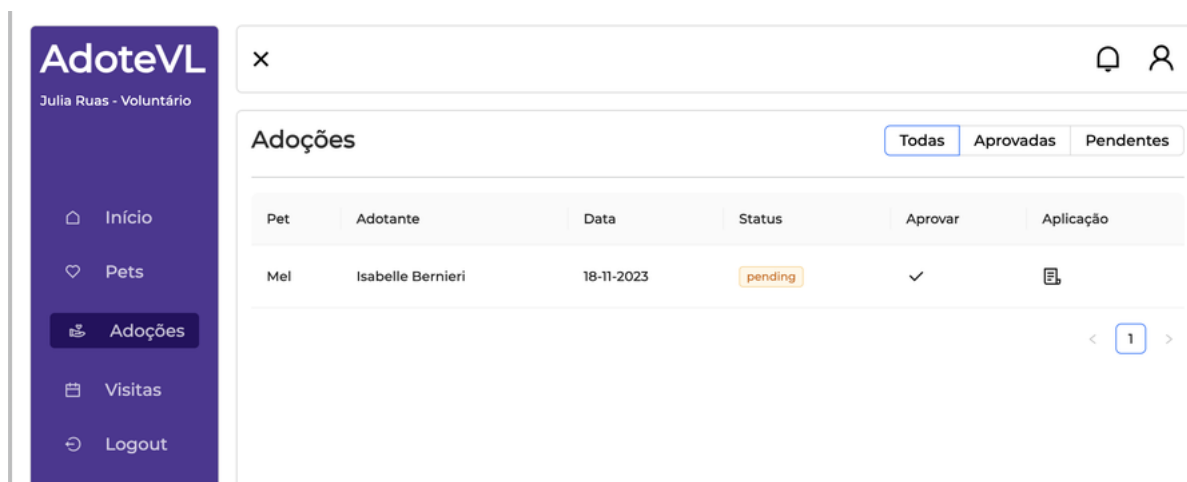
Um botão "SALVAR" em azul está localizado no canto inferior direito do formulário.

A AUTORA (2023)

4.2.14 Gestão Adoção

Na tela de gestão de adoções, o voluntário terá acesso a todas as solicitações de adoção. Ele poderá filtrar entre as opções "todas", "aprovadas" e "pendentes", para visualizar apenas as adoções que correspondem a esses critérios, como mostra a FIGURA 37.

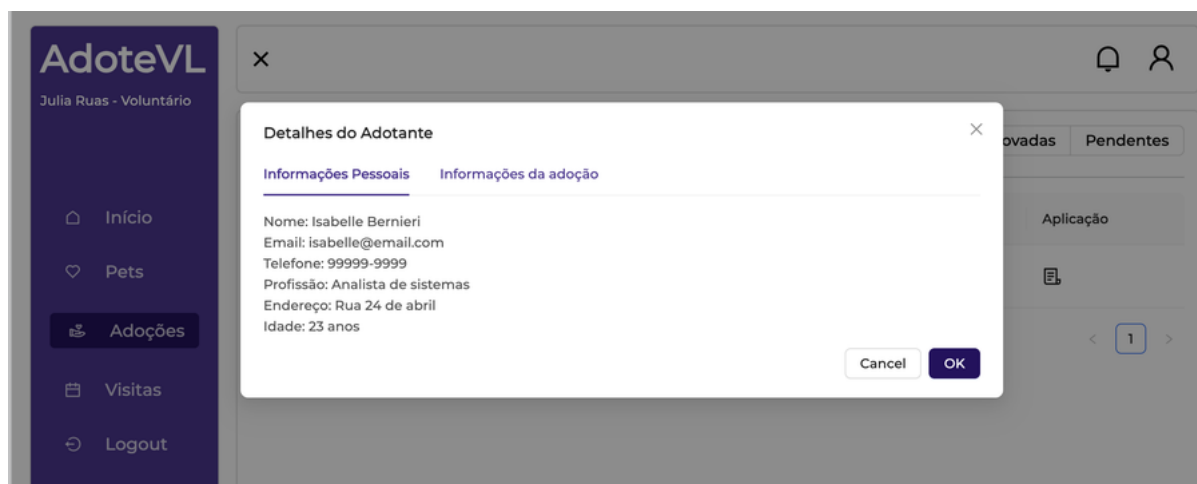
Figura 37 – TELA DE GESTÃO DE ADOÇÃO



A AUTORA (2023)

Ao selecionar uma solicitação de adoção específica, um modal detalhado será exibido (FIGURA 38 e 39), mostrando os dados do adotante, incluindo informações pessoais fornecidas durante a inscrição para adoção.

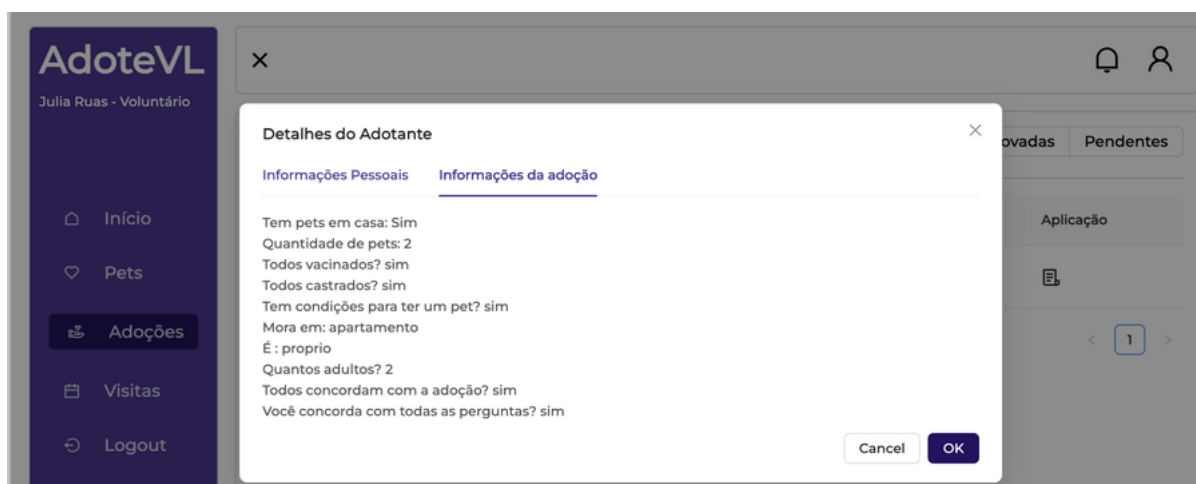
Figura 38 – MODAL DE DETALHES DO ADOTANTE



A AUTORA (2023)

Essa visualização detalhada permitirá que o voluntário avalie cuidadosamente as informações e tome a decisão apropriada em relação à aprovação ou rejeição da solicitação de adoção.

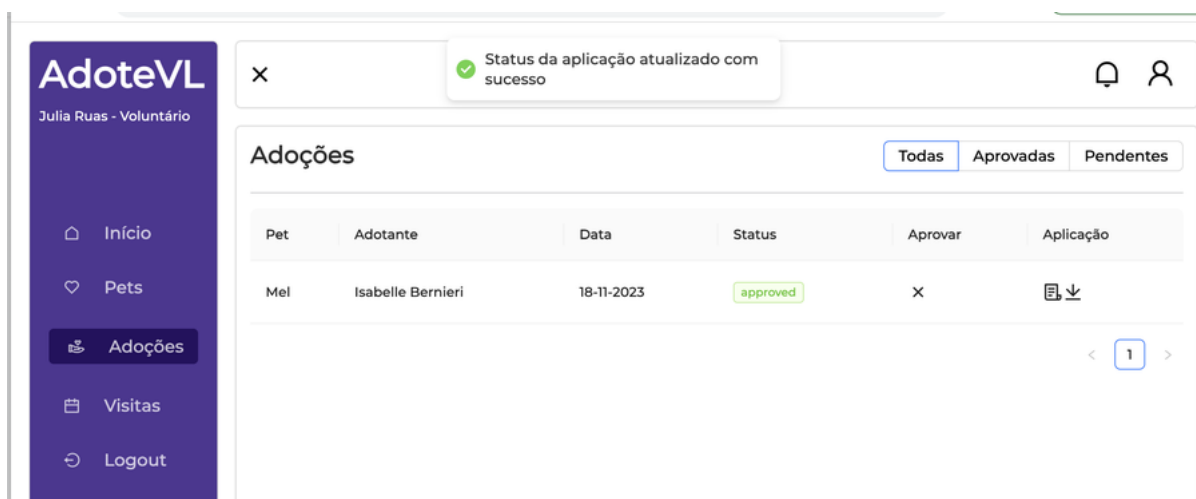
Figura 39 – MODAL DE DETALHES DO ADOTANTE



A AUTORA (2023)

O voluntário terá a capacidade de revisar as informações fornecidas pelo adotante e, com base nesses dados, poderá aprovar ou negar a solicitação de adoção. Ao aprovar, o pet será marcado como adotado e o status da adoção como aprovado, encerrando o processo de adoção para aquele pet específico, como mostra a FIGURA 40.

Figura 40 – TELA GESTÃO DE ADOÇÕES



A AUTORA (2023)

Quando uma adoção é aprovada, um ícone para geração de PDF estará disponível. Ao clicar nele, o voluntário pode baixar o contrato de adoção em formato PDF, facilitando o processo de obtenção desse documento essencial para a conclusão da adoção, conforma a FIGURA 41.

Figura 41 – CONTRATO

Contrato de Adoção

Adotante: Isabelle Bernieri

Pet Adotado: Mel

Termos e Condições:

Ao assinar este contrato, o adotante reconhece a responsabilidade de cuidar do animal adotado e concorda com os seguintes termos:

- O adotante é responsável por fornecer cuidados adequados, alimentação e abrigo ao animal adotado.
- O adotante concorda em não abandonar o animal ou transferi-lo a terceiros sem consentimento prévio da organização de adoção.
- O adotante concorda em cumprir todas as leis e regulamentos locais relacionados à posse de animais de estimação.
- A organização de adoção não assume responsabilidade por quaisquer danos causados pelo animal após a adoção.

Ao assinar abaixo, o adotante confirma que leu e concorda com os termos deste contrato e está ciente das responsabilidades associadas à adoção do animal.

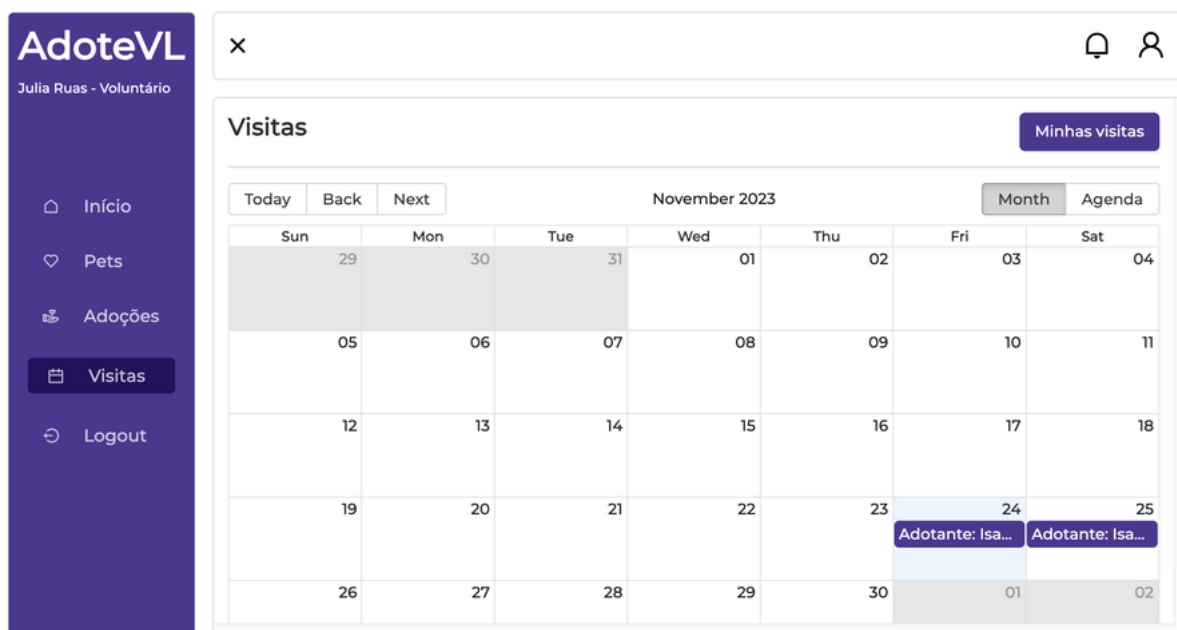
Assinatura do Adotante:

A AUTORA (2023)

4.2.15 Gestão Visita

Na tela de gestão de visitas (FIGURA 42), o voluntário poderá visualizar, por meio de um calendário, todas as visitas já agendadas em que está designado para acompanhar os interessados na adoção. Isso permitirá que ele tenha uma visão clara das datas ocupadas, ajudando na organização de sua agenda e no planejamento para atender aos potenciais adotantes durante as visitas ao abrigo.

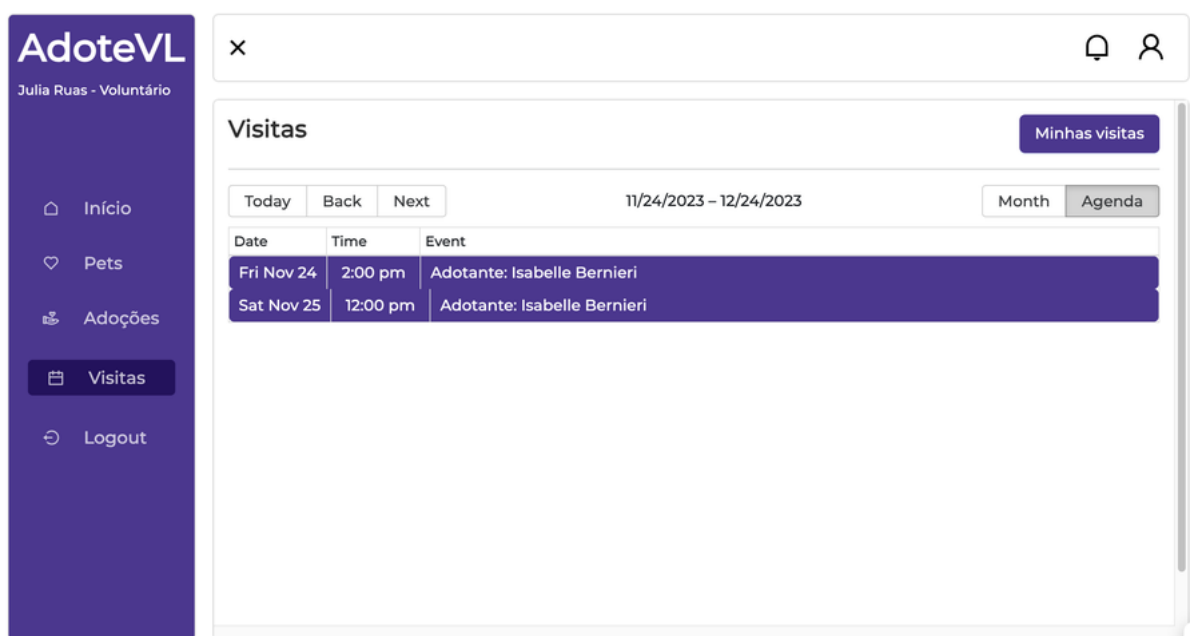
Figura 42 – TELA GESTÃO DE VISITAS - CALENDARIO



A AUTORA (2023)

Na tela de gestão de visitas (FIGURA 43), além do calendário, o voluntário terá a opção de visualizar as visitas em formato de agenda. Isso proporcionará uma visão mais detalhada das informações, permitindo organizar e analisar os dados de maneira mais estruturada e facilitando o acompanhamento das visitas agendadas.

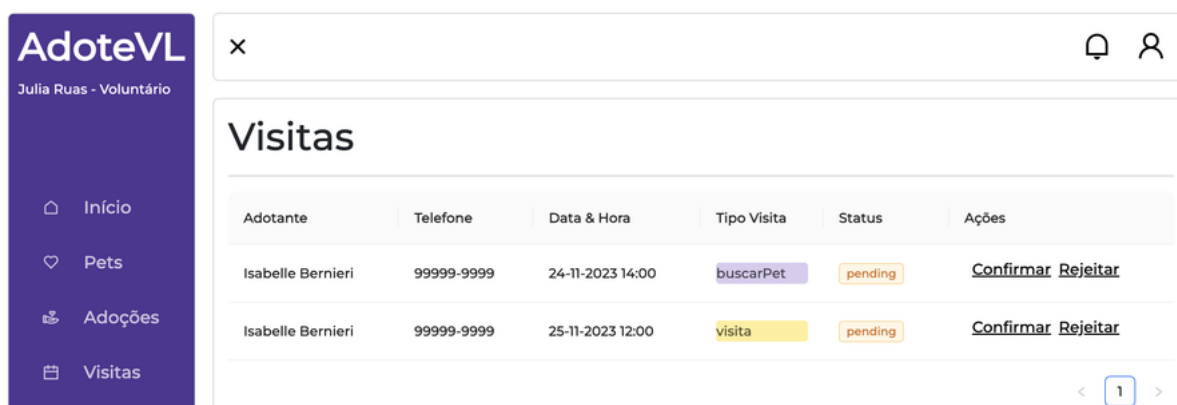
Figura 43 – TELA GESTÃO DE VISITAS - AGENDA



A AUTORA (2023)

Se o voluntário clicar no botão “Minhas Visitas”, terá uma visão mais detalhada das visitas, podendo aprová-las ou recusá-las conforme necessário., como mostra a FIGURA 44.

Figura 44 – TELA GESTÃO DE VISITAS - TABELA



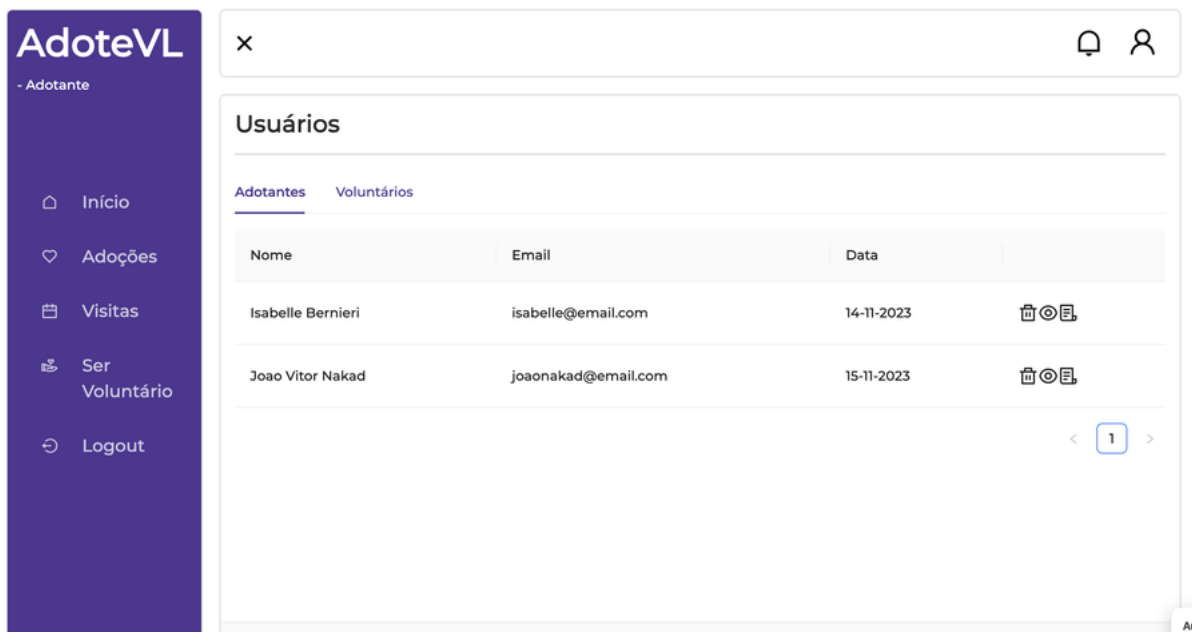
Adotante	Telefone	Data & Hora	Tipo Visita	Status	Ações
Isabelle Bernieri	99999-9999	24-11-2023 14:00	buscarPet	pending	Confirmar Rejeitar
Isabelle Bernieri	99999-9999	25-11-2023 12:00	visita	pending	Confirmar Rejeitar

A AUTORA (2023)

4.2.16 Gestão Usuários

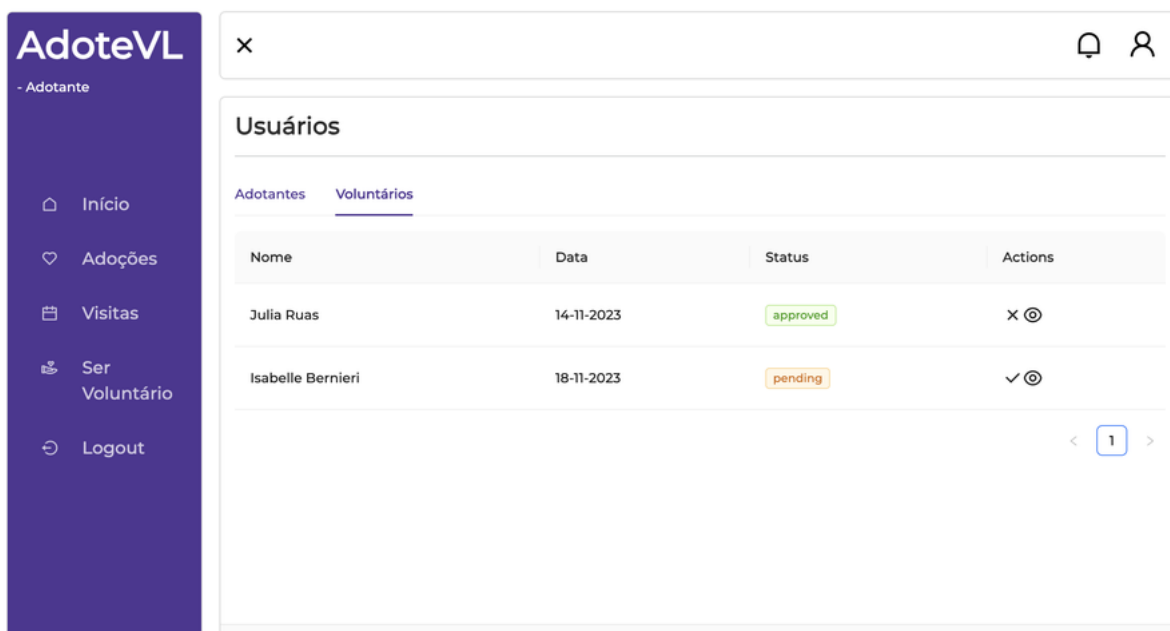
Na tela de gestão de usuários, destinada ao administrador, há duas abas principais: uma para a gestão de adotantes (FIGURA 45) e outra para a gestão de voluntários (FIGURA 46). O administrador terá a capacidade de revisar e tomar ações específicas em relação aos perfis de adotantes e voluntários registrados na plataforma. Isso inclui a visualização detalhada de informações, revisão de solicitações pendentes e outras operações relacionadas à administração dos usuários da plataforma.

Figura 45 – TELA GESTÃO DE USUÁRIOS - ABA ADOTANTE



A AUTORA (2023)

Figura 46 – TELA GESTÃO DE USUÁRIOS - ABA VOLUNTÁRIOS



A AUTORA (2023)

5 CONSIDERAÇÕES FINAIS

Ao final deste projeto, destaca-se o compromisso com a evolução contínua no cenário de adoção de animais. O sistema é concebido com a missão de simplificar e facilitar o processo de adoção, conectando o adotante a animais em busca de um lar amoroso.

Os objetivos principais são proporcionar uma plataforma intuitiva e acessível, promovendo a interação entre o adotante e animais disponíveis para adoção. Apesar das restrições temporais, centraliza-se na implementação de funcionalidades que priorizam a transparência, agilidade e segurança no processo de adoção.

O projeto concentra-se na criação de uma documentação abrangente para seus requisitos e estrutura, ao mesmo tempo em que avaliava a viabilidade do sistema projetado. São conduzidas pesquisas com público-alvo, estabelecido um cronograma para o trabalho e desenvolvidos Diagrama de Casos de Uso, Especificação dos Casos de Uso, Diagrama de Classes, Diagrama de Sequência e Diagrama de Banco de Dados. Adicionalmente, são delineados os conceitos relacionados a metodologias aplicadas e tecnologias a serem empregadas.

Para além da documentação, o sistema é construído para demonstrar suas principais funcionalidades e permitir uma análise mais profunda da aplicação. Foram adotados os frameworks React, Node e Express, juntamente com o banco de dados MongoDB.

Por fim, a perspectiva para os objetivos futuros é expandir o alcance do sistema, ampliando parcerias e promovendo uma experiência mais inclusiva. A meta é continuar aprimorando e oferecendo ferramentas que atendam não apenas às necessidades atuais, mas também às demandas emergentes no universo da adoção responsável de animais.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Após a conclusão deste projeto, foram identificadas oportunidades para melhorias futuras. Alguns recursos mencionados foram inicialmente reavaliados quanto à sua prioridade, a fim de assegurar que os demais requisitos fossem atendidos dentro do prazo estabelecido. Os itens relacionados a essas melhorias estão descritos na próxima subseção:

5.1.1 Criação de um ChatBox

Para futuros trabalhos, a implementação de um chatbox seria uma adição valiosa, simplificando a comunicação entre a organização e os potenciais adotantes. Essa ferramenta permitiria uma interação mais direta e instantânea, esclarecendo dúvidas, fornecendo orientações e agilizando o processo de adoção, promovendo uma experiência mais próxima e facilitando o engajamento dos interessados.

5.1.2 Acompanhamento Pós-Adoção

Como consideração final, uma das principais melhorias futuras sugeridas para este projeto seria a criação de um programa de acompanhamento pós-adoção. Infelizmente, devido a restrições de tempo e tamanho da equipe, essa funcionalidade não pode ser incluída no escopo. No entanto, ao desenvolver esse recurso permitiria aos adotantes compartilhar atualizações, fotos e histórias do animal pós a adoção, formando uma comunidade dentro da plataforma.

5.1.3 Responsividade e Identidade Visual

Nos próximos projetos, está planejada a criação de um site responsivo para atender às diversas necessidades dos usuários em diferentes dispositivos. Além disso, será mantida e acompanhada a identidade visual da ONG para garantir consistência e alinhamento com os valores e a missão da organização, refletindo as cores e o logotipo da Adote no site.

Referências

- ABONG. **Um novo marco legal para ONGs no Brasil**. 2007. Acesso em: 30/10/2023.
- ANDA. **Brasil tem 30 milhões de animais abandonados**. 2013. Disponível em: <https://www.jusbrasil.com.br/noticias/brasil-tem-30-milhoes-de-animais-abandonados/100681698>. Acesso em: 30/10/2023.
- ANDRADE, P. B. **O papel das ongs nas questões públicas**. 2015. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/1800/1/PBA18082017.pdf>. Acesso em: 30/10/2023.
- ANJO, P. **Abandono de animais: entenda qual é o impacto na vida do pet**. 2022. Disponível em: <https://petanjo.com/blog/paredeabandonar-conscientizacao-sobre-abandono-de-animais/>. Acesso em: 30/06/2023.
- ANTD. **ANT Design**. 2023. Disponível em: <https://ant.design/docs/react/getting-started>. Acesso em: 30/06/2023.
- ARCA. **Levantamento mostra que apenas 10% dos cachorros e gatos no Brasil são castrados**. 2014. Disponível em: <https://www2.jornalcruzeiro.com.br/materia/545910/levantamento-mostra-que-apenas-10-dos-cachorros-e-gatos-no-brasil-sao-castrados>. Acesso em: 30/06/2023.
- ASTAH. **About Astah & Change Vision**. 2023. Disponível em: <https://astah.net/about/>. Acesso em: 30/10/2023.
- AWARI. **Arquitetura REST**. 2023a. Disponível em: [https://awari.com.br/arquitetura-rest-representational-state-transfer-principios-e-melhores-praticas-na-construcao-de-apis/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Arquitetura%20REST%20\(Representational%20State%20Transfer\):%20Princípios%20e%20Melhores%20Práticas%20na%20Construção%20de%20APIs#:~:text=Princípios%20Fundamentais%20da%20Arquitetura%20REST,-1.&text=Ela%20define%20que%20as%20informações,JSON%2C%20HTML%2C%20entre%20outros](https://awari.com.br/arquitetura-rest-representational-state-transfer-principios-e-melhores-praticas-na-construcao-de-apis/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Arquitetura%20REST%20(Representational%20State%20Transfer):%20Princípios%20e%20Melhores%20Práticas%20na%20Construção%20de%20APIs#:~:text=Princípios%20Fundamentais%20da%20Arquitetura%20REST,-1.&text=Ela%20define%20que%20as%20informações,JSON%2C%20HTML%2C%20entre%20outros). Acesso em: 30/10/2023.
- AWARI. **Empresas que utilizam Scrum**. 2023b. Disponível em: <https://awari.com.br/empresas-que-utilizam-scrum-aprenda-com-os-melhores/>. Acesso em: 30/08/2023.
- BOEG, J. **Kanban em 10 Passos**. InfoQ Brasil, 2012. Disponível em: <https://www.infoq.com/br/minibooks/priming-kanban-jesper-boeg/>.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: Guia do Usuário**. 1. ed. Rio de Janeiro: Elsevier, 2006. 552 p.
- BRASIL, I. P. **Censo Pet IPB: com alta record de 6% em um ano, gatos lideram crescimento de animais de estimação no Brasil**. 2022. Disponível em: <https://institutopetbrasil.com/fique-por-dentro/amor-pelos-animais-impulsiona-os-negocios-2-2/>. Acesso em: 10/06/2023.
- CATLAND. **CATLAND**. 2023. Disponível em: <https://catland.org.br>. Acesso em: 30/10/2023.

CLOUDINARY. **Cloudinary Documentation**. 2023. Disponível em: <https://cloudinary.com/documentation>. Acesso em: 10/12/2023.

COBASI. **Cobasi Cuida lança estudo inédito sobre abandono de animais no Brasil**. 2022. Disponível em: <https://blog.cobasi.com.br/cobasi-cuida-estudo-abandono-de-animais-no-brasil/>. Acesso em: 30/06/2023.

CURITIBA. **LEI N 13.914**. 2011. Disponível em: https://protecaoanimal.curitiba.pr.gov.br/images/legislacao/LEI_13914_2011.pdf. Acesso em: 30/06/2023.

DIAS, J. **Raiva canina**. 2015. Disponível em: <https://tudosobrecachorros.com.br/raiva-canina/>. Acesso em: 30/06/2023.

ESEME, S. **Os 10 Tipos mais Populares de Aplicativos Node.js em 2023**. 2023. Disponível em: <https://kinsta.com/pt/blog/aplicativos-node-js/>. Acesso em: 30/10/2023.

ESTRELLA, C. **O Que é Node.js**. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-node-js>. Acesso em: 30/10/2023.

GALVAO, P.; ARAUJO, C. M. M. **ONGs no Brasil**. 2014. Disponível em: [http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1677-11682018000100007#:~:text=As%20primeiras%20ONGs%20no%20Brasil,2005%3B%20Souza%2C%202009\).](http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1677-11682018000100007#:~:text=As%20primeiras%20ONGs%20no%20Brasil,2005%3B%20Souza%2C%202009).) Acesso em: 30/10/2023.

GIL, R. **Kanban: Guia completo para entender o sistema de gestão visual**. 2019. Disponível em: <https://artia.com/kanban/>. Acesso em: 30/06/2023.

GIT. **Git Documentation**. 2023. Disponível em: <https://git-scm.com/doc>. Acesso em: 30/20/2023.

GITHUB. **GITHUB**. 2023. Disponível em: <https://github.com>. Acesso em: 30/10/2023.

GOMES, C. C. M. **Guarda responsável de animais de companhia**. 2013. Monografia (Medicina Veterinária) — Faculdade de Agronomia e Medicina Veterinária da Universidade. Disponível em: <https://bdm.unb.br/handle/10483/5985>.

GUPTA, L. REST API Tutorial. What is REST. novembro 2023. Disponível em: <https://restfulapi.net>. Acesso em: 10 nov 2023.

HAT, R. **Introdução à cloud computing**. 2023. Disponível em: <https://www.redhat.com/pt-br/topics/cloud>. Acesso em: 30/10/2023.

HEROKU. **Abou Heroku**. 2023. Disponível em: <https://www.heroku.com/about>. Acesso em: 30/10/2023.

IBGE. **População de animais de estimação no Brasil**. 2013. Acesso em: 30/06/2023.

IBM. **PaaS (Platform-as-a-Service)**. 2021. Disponível em: <https://www.ibm.com/topics/paas>. Acesso em: 30/10/2023.

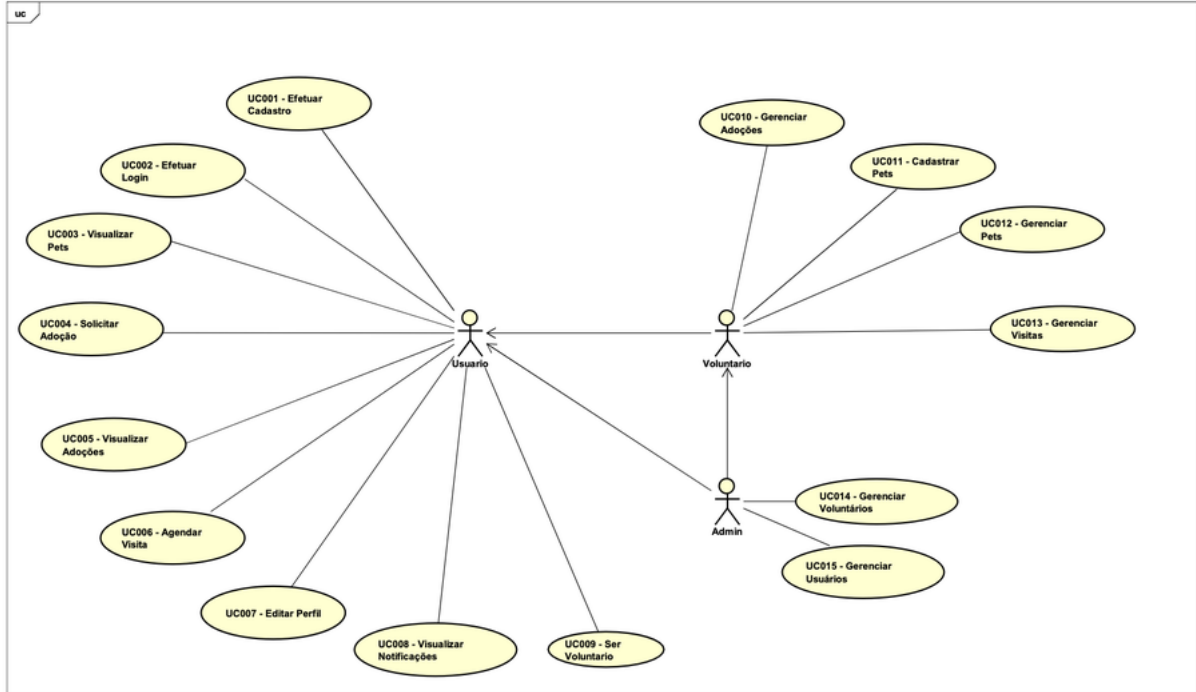
IBP. **Brasil tem quase 185 mil animais resgatados por ONGs**. 2022. Disponível em: <https://www.cnnbrasil.com.br/nacional/brasil-tem-quase-185-mil-animais-resgatados-por-ongs-diz-instituto/>. Acesso em: 30/06/2023.

- IBP. **Censo Pet: 139,3 milhões de animais de estimação no Brasil**. 2019. Disponível em: <https://institutopetbrasil.com/imprensa/censo-pet-1393-milhoes-de-animais-de-estimacao-no-brasil/>.
- LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. Terceira. Porto Alegre: Bookman, 2007.
- MANIFESTO, A. **Manifesto Agil**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 30/06/2023.
- MONGODB. **MERN Stack**. 2023a. Disponível em: <https://www.mongodb.com/mern-stack>. Acesso em: 30/10/2023.
- MONGODB. **MongoDB Documentation**. 2023b. Disponível em: <https://www.mongodb.com/docs/>. Acesso em: 30/10/2023.
- MOZILLA. **Javascript**. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 30/10/2023.
- POSTMAN. **POSTMAN**. 2023. Disponível em: <https://www.postman.com>. Acesso em: 30/10/2023.
- REACT. **React**. 2023. Disponível em: <https://react.dev>. Acesso em: 30/06/2023.
- SABBAGH, R. **Scrum: Gestão ágil para projetos de sucesso**. 2013. Acesso em: 30/06/2023.
- SCHWABER, K. **Agile Project Management with Scrum**. 2004. Acesso em: 30/06/2023.
- SCHWABER, K.; HUNDHAUSEN, R.; STARR, D. **Agile Project Management with Scrum**. 2. ed. [S.l.]: Microsoft Press, 2015. ISBN 978-0735696938.
- SILVA, R. O. da; MARTINS, B. R.; DINIZ, W. G. A COMPLEXIBILIDADE DA UML E SEUS DIAGRAMAS. **Tecnologias em Projeção**, v. 8, n. 1, p. 86 – 99, out 2017.
- SILVEIRA, A. **React Descomplicado**. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>. Acesso em: 30/06/2023.
- SOMMERVILLE. **Engenharia de Software**. 8. ed. [S.l.]: Pearson Addison Wesley, 2007. Acesso em: 30/06/2023.
- STACK OVERFLOW. **2022 Developer Survey**. 2022. Disponível em: <https://survey.stackoverflow.co/2022>. Acesso em: 30/06/2023.
- VSCODE. **Visual Studio Code**. 2023. Disponível em: <https://code.visualstudio.com>. Acesso em: 30/10/2023.

Apêndices

APÊNDICE A – DIAGRAMA DE CASOS DE USO

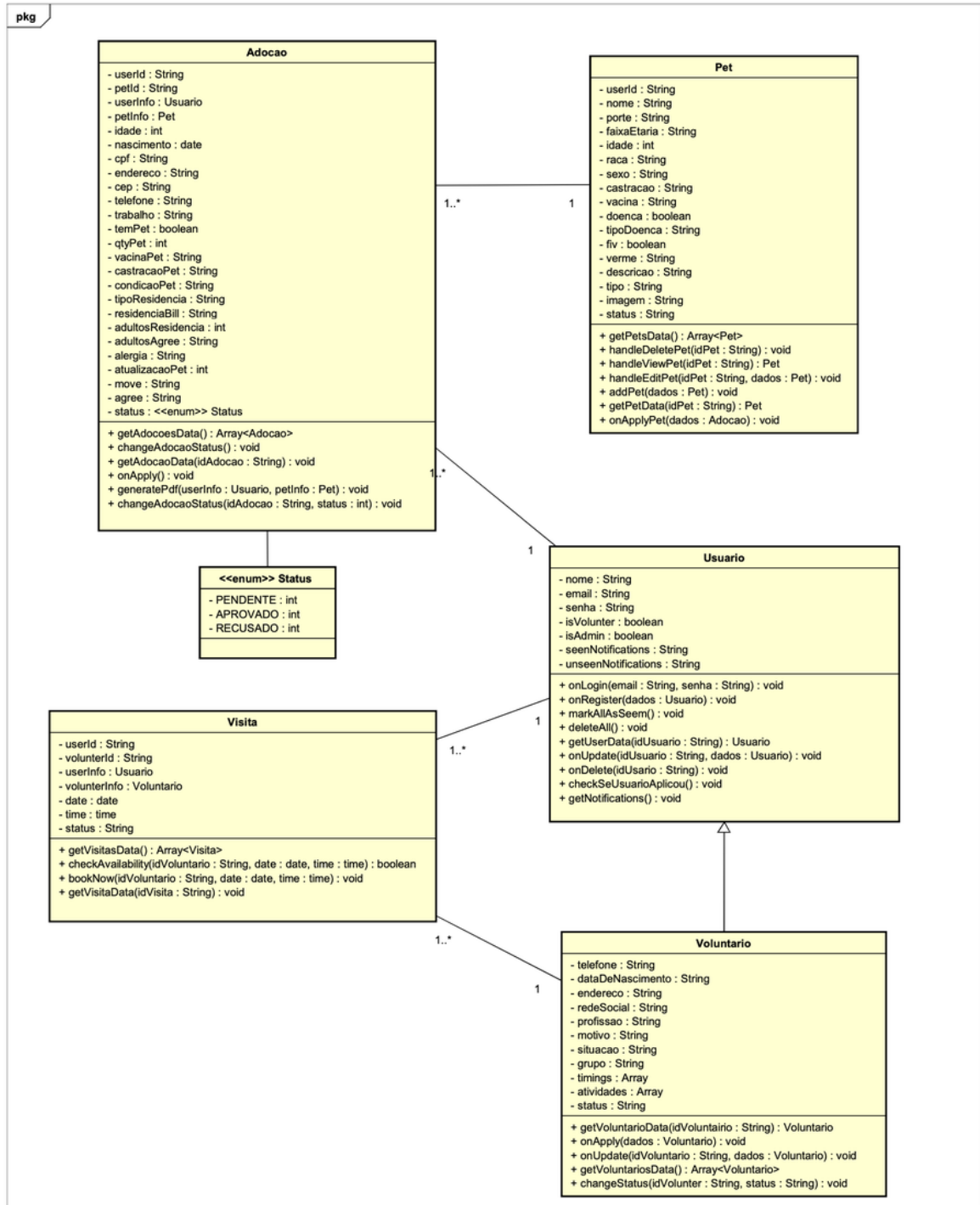
Figura 47 – DIAGRAMA DE CASOS DE USO



A AUTORA (2023)

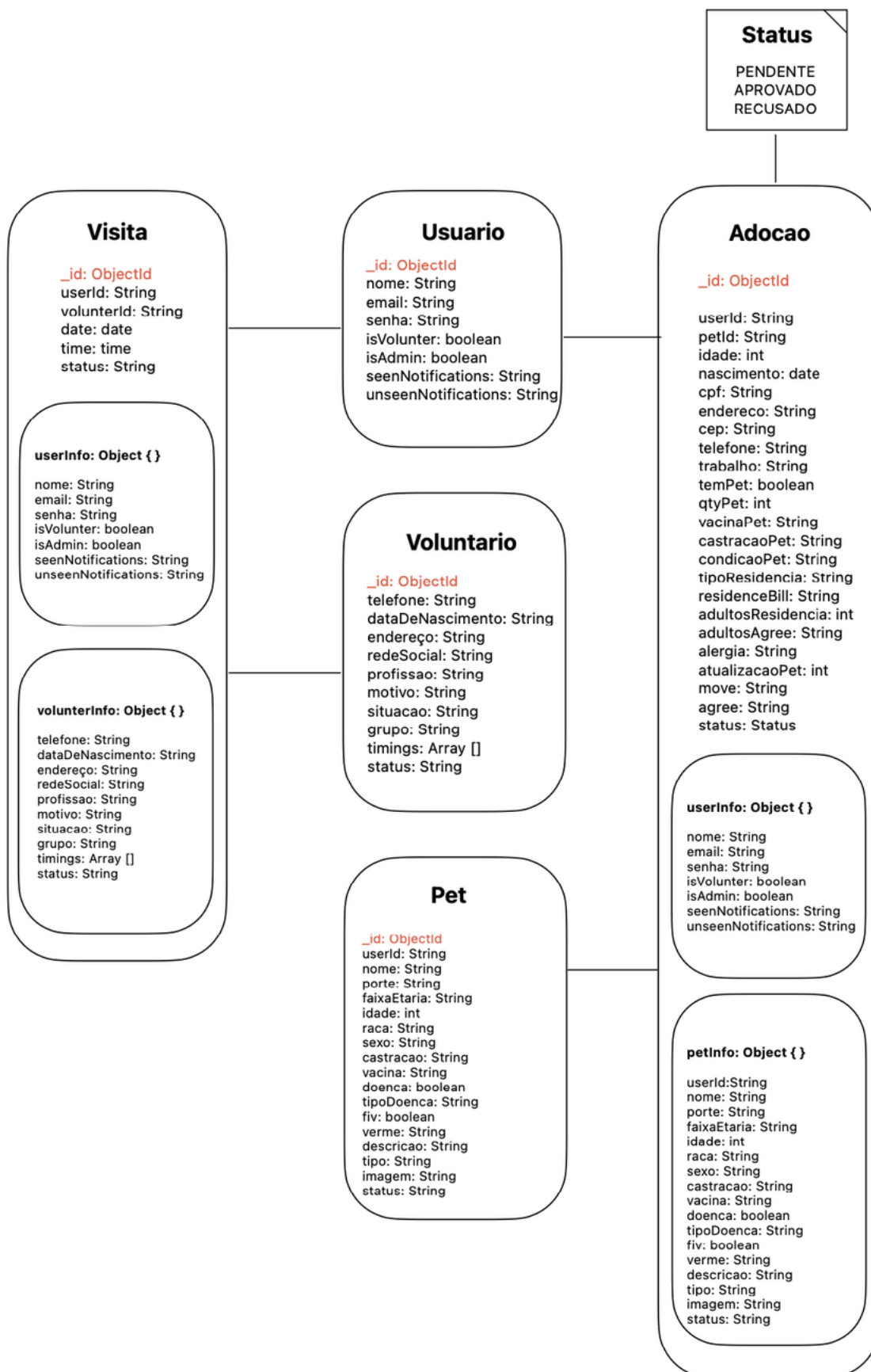
APÊNDICE B – DIAGRAMA DE CLASSES

Figura 48 – DIAGRAMA DE CLASSES



APÊNDICE C – DIAGRAMA DA ESTRUTURA DO BANCO DE DADOS

Figura 49 – ESTRUTURA DO BANCO DE DADOS



APÊNDICE D – ENDPOINTS PARA AS REQUISIÇÕES

API Adote Um Vira Lata

[Base URL: localhost:3000/]

Documentação da API Adote Um Vira Lata

HTTP

Admin

Documentação das Rotas de administrador:

1. Rota: `GET /api/admin/get-all-volunteers`

- **Descrição:** Obtém todos os voluntários cadastrados.
- **Parâmetros:** Nenhum
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Voluntários obtidos com sucesso",
  "data": [ // Lista de voluntários ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter voluntários",
  "success": false,
  "error": "Descrição do erro"
}
```

2. Rota: `GET /api/admin/get-all-users`

- **Descrição:** Obtém todos os usuários cadastrados.

- **Parâmetros:** Nenhum
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Usuários obtidos com sucesso",
  "data": [ // Lista de usuários ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter usuários",
  "success": false,
  "error": "Descrição do erro"
}
```

3. Rota: `POST /api/admin/change-volunter-account-status`

- **Descrição:** Atualiza o status da conta de um voluntário.
- **Parâmetros:**
 - Corpo da solicitação contendo `volunterId` e `status`.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Status do voluntário atualizado com sucesso",
  "data": { // Dados atualizados do voluntário }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao atualizar status do voluntário",
  "success": false,
}
```

```
"error": "Descrição do erro"
}
```

4. Rota: **DELETE** /api/admin/delete-user/:id

- **Descrição:** Exclui um usuário com base no ID.
- **Parâmetros:**
 - **id** (Path Parameter): ID do usuário a ser excluído.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Usuário deletado com sucesso",
  "data": { // Dados do usuário excluído }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao deletar usuário",
  "success": false,
  "error": "Descrição do erro"
}
```

Volunter

Documentação das Rotas de voluntário:

1. Rota: **GET** /api/volunter/get-volunter-info-by-user-id/:id

- **Descrição:** Obtém informações sobre um voluntário com base no ID do usuário.
- **Parâmetros:**
 - **id** (Path Parameter): ID do usuário associado ao voluntário.

- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Voluntário obtido com sucesso",
  "data": { // Dados do voluntário }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter voluntário",
  "success": false,
  "error": "Descrição do erro"
}
```

2. Rota: `GET /api/volunter/get-volunter-info-by-id/:id`

- **Descrição:** Obtém informações sobre um voluntário com base no ID do voluntário.
- **Parâmetros:**
 - `id` (Path Parameter): ID do voluntário.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Voluntário obtido com sucesso",
  "data": { // Dados do voluntário }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter voluntário",
  "success": false,
}
```

```
"error": "Descrição do erro"
}
```

3. Rota: `PUT /api/volunter/update-volunter-profile`

- **Descrição:** Atualiza o perfil de um voluntário com base no ID do usuário.
- **Parâmetros:**
 - Corpo da solicitação contendo os dados a serem atualizados.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Perfil atualizado com sucesso",
  "data": { // Dados atualizados do voluntário }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao atualizar perfil",
  "success": false,
  "error": "Descrição do erro"
}
```

4. Rota: `GET /api/volunter/get-appointments-by-volunter-id/:id`

- **Descrição:** Obtém todos os compromissos associados a um voluntário com base no ID do voluntário.
- **Parâmetros:**
 - `id` (Path Parameter): ID do voluntário.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
```

```
"message": "Agendamentos obtidos com sucesso",
"data": [ // Lista de compromissos associados ao voluntário ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter agendamentos",
  "success": false,
  "error": "Descrição do erro"
}
```

5. Rota: `PUT /change-application-status`

- **Descrição:** Atualiza o status de uma aplicação.
- **Parâmetros:**
 - Corpo da Requisição (JSON):
 - `applicationId` (String): ID único da aplicação a ser atualizada.
 - `status` (String): Novo status a ser atribuído à aplicação.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Status da aplicação atualizado com sucesso",
  "success": true,
  "data": { // Informações atualizadas da aplicação }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao atualizar o status da aplicação",
  "success": false,
  "error": "Descrição do erro específico"
}
```

6. Rota: `PUT /api/volunter/change-appointment-status`

- **Descrição:** Atualiza o status de um compromisso.
- **Parâmetros:**
 - Corpo da solicitação contendo `appointmentId` e `status`.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Status do agendamento atualizado com sucesso"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao atualizar status do agendamento",
  "success": false, "error": "Descrição do erro"
}
```

7. Rota: `DELETE /api/volunter/delete-pet/:id`

- **Descrição:** Exclui um animal de estimação com base no ID.
- **Parâmetros:**
 - `id` (Path Parameter): ID do animal de estimação a ser excluído.
- **Middleware de Autenticação:** Sim (authMiddleware)
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Pet deletado com sucesso",
  "data": { // Dados do animal de estimação excluído }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao deletar Pet",
  "success": false,
  "error": "Descrição do erro"
}
```

8. Rota: `PUT /api/volunter/update-pet`

- **Descrição:**

- Atualiza o perfil de um pet com base no ID.

- **Parâmetros:**

- `petId` (String): ID único do animal de estimação a ser atualizado.
- Corpo da Requisição (JSON):
 - Propriedades opcionais que devem ser atualizadas no animal de estimação.

- **Middleware de Autenticação:** Sim (authMiddleware)

- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Pet updated successfully",
  "data": { // Informações atualizadas do animal de estimação }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Error getting pet info",
  "success": false,
  "error": "Descrição do erro específico"
}
```

User

Documentação das Rotas de usuário:

1. Rota: **POST /api/user/register**

- **Descrição:** Registra um novo usuário.
- **Parâmetros:** Nenhum
- **Middleware de Autenticação:** Não
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true, "message": "Usuário criado com sucesso"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false, "message": "Usuário já existe"
}
```

```
{
  "success": false,
  "message": "Erro ao criar usuário",
  "error": "Detalhes do erro"
}
```

2. Rota: **POST /api/user/login**

- **Descrição:** Realiza o login de um usuário.
- **Parâmetros:**
 - **email** (string): E-mail do usuário.
 - **password** (string): Senha do usuário.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Login bem-sucedido",
  "data": "TOKEN"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Usuário não existe"
}
```

```
{
  "success": false,
  "message": "Senha incorreta"
}
```

```
{
  "success": false,
  "message": "Erro ao fazer login",
  "error": "Detalhes do erro"
}
```

3. Rota: `GET /api/user/get-user-info-by-id/:id`

- **Descrição:** Obtém as informações de um usuário por ID.
- **Parâmetros:**
 - `id` (string): ID do usuário.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
```

```
"data": { // Informações do usuário (exceto a senha) }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Usuário não existe"
}
```

```
{
  "success": false,
  "message": "Erro ao obter informações do usuário",
  "error": "Detalhes do erro"
}
```

4. Rota: **POST /api/user/apply-volunter-account**

- **Descrição:** Aplica para uma conta de voluntário.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:**
 - `name` (string): Nome do voluntário.
 - `outroParametro` (tipo): Outra descrição do parâmetro, se houver.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Conta de voluntário aplicada com sucesso"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Erro ao aplicar conta de voluntário",
}
```

```
"error": "Detalhes do erro"
}
```

5. Rota: **POST /api/user/mark-all-notifications-as-seen**

- **Descrição:** Marca todas as notificações como vistas.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:**
 - `userId` (string): ID do usuário.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Todas as notificações marcadas como vistas",
  "data": { // Dados do usuário atualizados }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Erro ao marcar as notificações como vistas",
  "error": "Detalhes do erro"
}
```

6. Rota: **DELETE /api/user/delete-all-notifications**

- **Descrição:** Apaga todas as notificações de um usuário.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:**
 - `userId` (string): ID do usuário.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
```

```
"message": "Todas as notificações apagadas",
"data": { // Dados do usuário atualizados }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Erro ao apagar as notificações",
  "error": "Detalhes do erro"
}
```

7. Rota: **GET /api/user/get-all-approved-volunteers**

- **Descrição:** Obtém todos os voluntários aprovados.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:** Nenhum
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Voluntários obtidos com sucesso",
  "data": [ // Lista de voluntários aprovados ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "message": "Erro ao obter voluntários",
  "error": "Detalhes do erro"
}
```

8. Rota: **GET /api/user/check-is-admin**

- **Descrição:** Verifica se o usuário possui permissão de administrador.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

- **Parâmetros:** Nenhum
- **Exemplo de Resposta de Sucesso:**

```
{
  "isAdmin": true // ou false
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "error": "Erro ao verificar status de administrador",
  "isAdmin": false
}
```

9. Rota: **POST /api/user/pets**

- **Descrição:** Cria um novo registro de animal de estimação.
- **Middleware de Autenticação:** Sim (**authMiddleware**).
- **Parâmetros:**
 - **body** (JSON): Detalhes do animal de estimação.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Animal de estimação criado com sucesso",
  "data": { // Detalhes do novo animal de estimação }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Detalhes do erro"
}
```

10. Rota: **GET /api/user/pets**

- **Descrição:** Obtém todos os registros de animais de estimação.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:** Nenhum
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Animais de estimação obtidos com sucesso",
  "data": [ // Lista de animais de estimação ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao buscar animais de estimação",
  "success": false,
  "error": "Detalhes do erro"
}
```

11. Rota: `GET /api/user/get-pet-info-by-id/:id`

- **Descrição:** Obtém informações de um animal de estimação pelo ID.
- **Middleware de Autenticação:** Sim (`authMiddleware`).
- **Parâmetros:**
 - `id` (Corpo da requisição): ID do animal de estimação.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Informações do animal de estimação obtidas com sucesso",
  "data": { // Detalhes do animal de estimação }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter informações do animal de estimação",
  "success": false,
  "error": "Detalhes do erro"
}
```

12. Rota: **POST /api/user/applications**

- **Descrição:** Cria uma nova aplicação para adoção de um animal.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

Parâmetros:

- `nome` : Nome do aplicante.
- `petId` : ID do animal que está sendo aplicado para adoção.
- `userId` : ID do usuário que está fazendo a aplicação.

- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Aplicação criada com sucesso",
  "data": { // Detalhes da aplicação }
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao criar aplicação",
  "error": "Detalhes do erro"
}
```

13. Rota: **GET /api/user/check-application**

- **Descrição:** Verifica se uma aplicação para adoção já foi feita por um usuário para um determinado animal.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

- **Parâmetros:**

- `petId`: ID do animal.
- `userId`: ID do usuário.

- **Exemplo de Resposta de Sucesso:**

```
{
  "applied": true
}
```

```
{
  "applied": false
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "applied": false,
  "error": "Detalhes do erro"
}
```

14. Rota: `GET /api/user/user-adoptions/:id`

- **Descrição:** Obtém todas as aplicações de adoção feitas por um usuário específico.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

Parâmetros:

- `id`: ID do usuário.

- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Aplicações obtidas com sucesso",
  "success": true,
}
```

```
"data": [ // Lista de aplicações ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "error": "Detalhes do erro"
}
```

15. Rota: **GET /api/user/all-applications**

- **Descrição:** Obtém todas as aplicações de adoção registradas no sistema.
- **Middleware de Autenticação:** Sim.
- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "data": [ // Lista de todas as aplicações ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "success": false,
  "error": "Detalhes do erro"
}
```

16. Rota: **POST /api/user/book-appointment**

- **Descrição:** Registra um novo agendamento.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

Parâmetros:

- Dados do agendamento (date, time, userId, volunteerInfo, etc.).

- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Agendamento realizado com sucesso",
  "success": true
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao realizar o agendamento",
  "success": false,
  "error": "Detalhes do erro"
}
```

17. Rota: **POST /api/user/check-booking-availability**

- **Descrição:** Verifica a disponibilidade para um novo agendamento em uma determinada data e horário.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

Parâmetros:

- `date` : Data do agendamento.
- `time` : Horário do agendamento.
- `volunterId` : ID do voluntário.

- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Agendamentos disponíveis",
  "success": true,
  "fromTime": "Hora de início do intervalo disponível"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao verificar a disponibilidade de agendamento",
  "success": false,
  "error": "Detalhes do erro"
}
```

18. Rota: `GET /api/user/get-appointments-by-user-id/:id`

- **Descrição:** Obtém todos os agendamentos feitos por um usuário específico.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

Parâmetros:

- `id`: ID do usuário.

- **Exemplo de Resposta de Sucesso:**

```
{
  "message": "Agendamentos obtidos com sucesso",
  "success": true,
  "data": [ // Lista de agendamentos ]
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao obter os agendamentos",
  "success": false,
  "error": "Detalhes do erro"
}
```

19. Rota: `PUT /api/user/update-user-profile`

- **Descrição:** Rota para atualizar o perfil de um usuário existente.
- **Middleware de Autenticação:** Sim (`authMiddleware`).

- **Parâmetros:**

- Corpo da solicitação contendo os dados a serem atualizados.

- **Exemplo de Resposta de Sucesso:**

```
{
  "success": true,
  "message": "Usuario editado com sucesso",
  "data": "Objeto contendo informações atualizadas do usuário"
}
```

- **Exemplo de Resposta de Erro:**

```
{
  "message": "Erro ao recuperar informacao do usuário",
  "success": false,
  "error": "Detalhes do erro ocorrido durante a atualização"
}
```

APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA

Figura 50 – DS01 - EFETUAR LOGIN

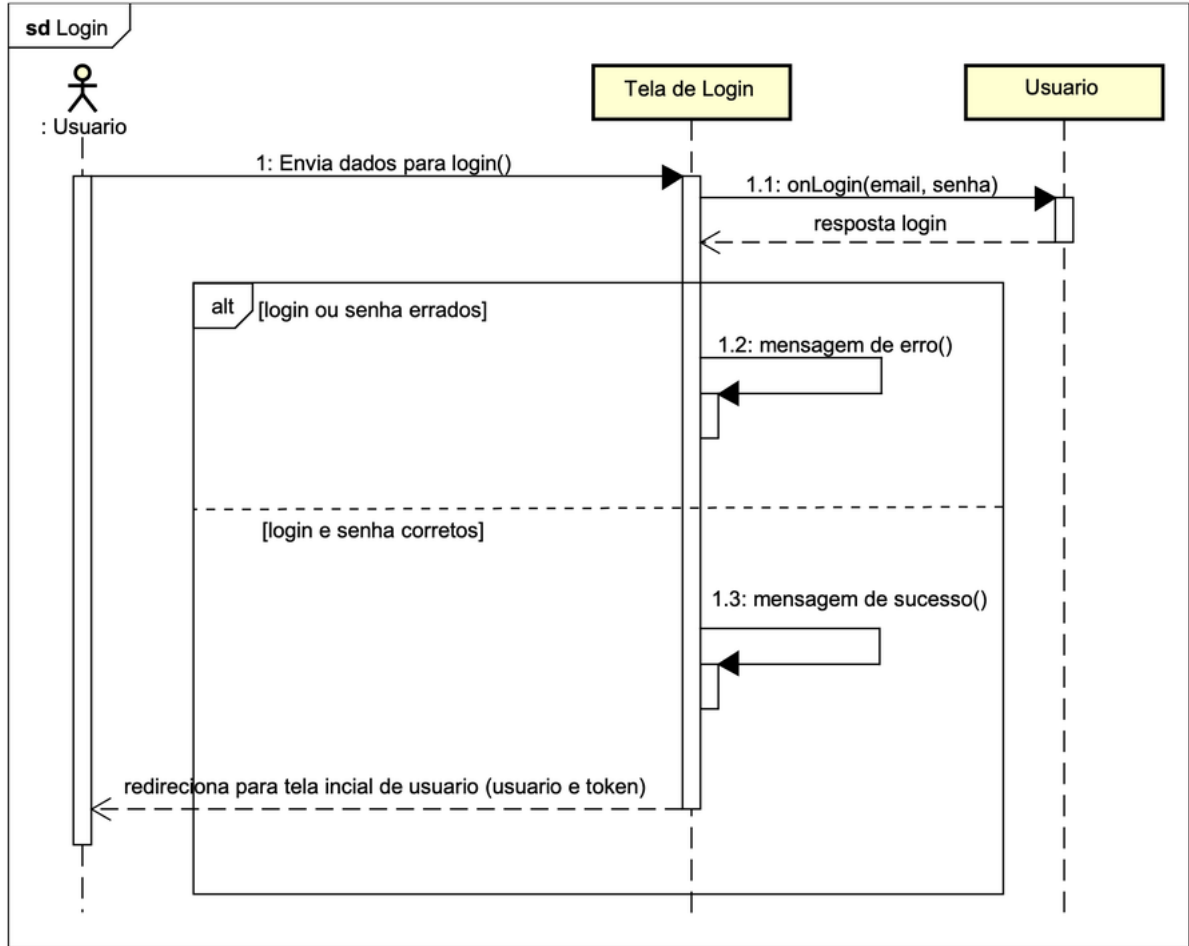
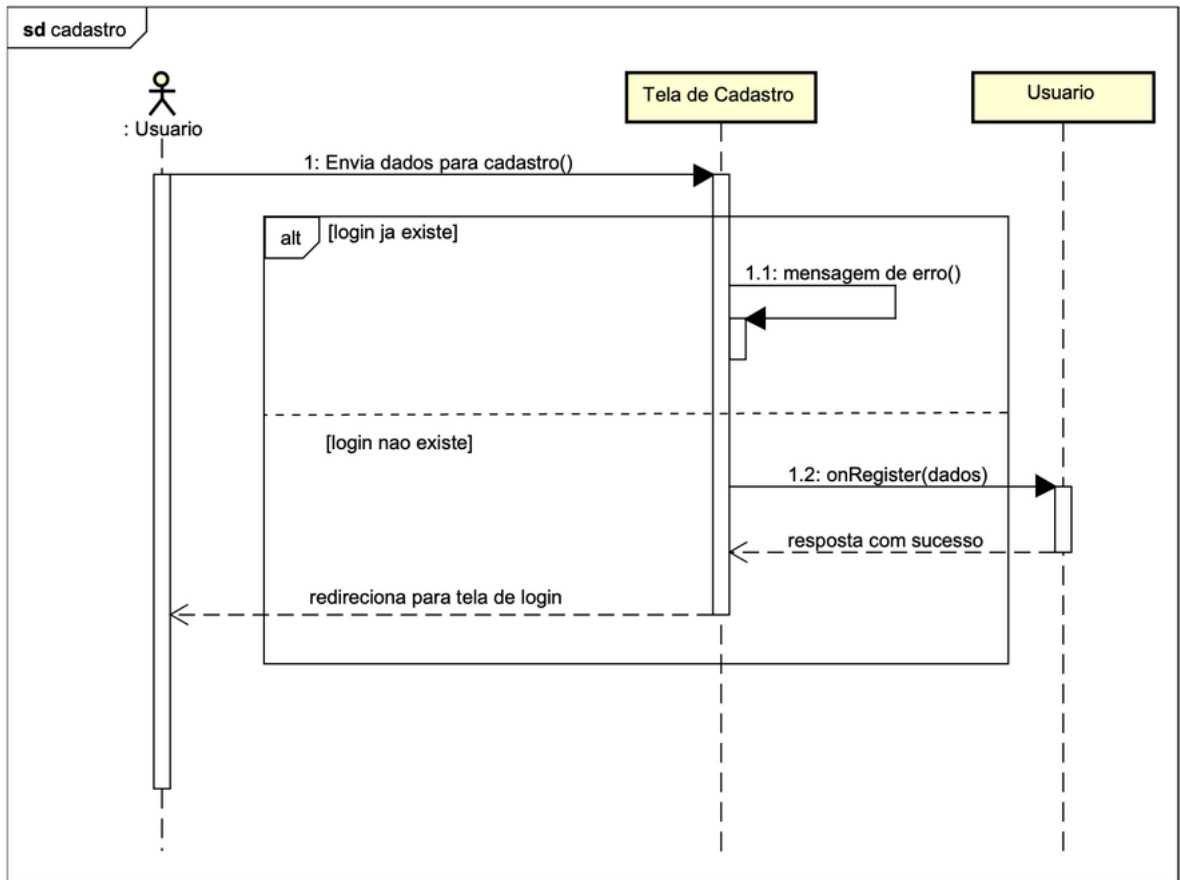
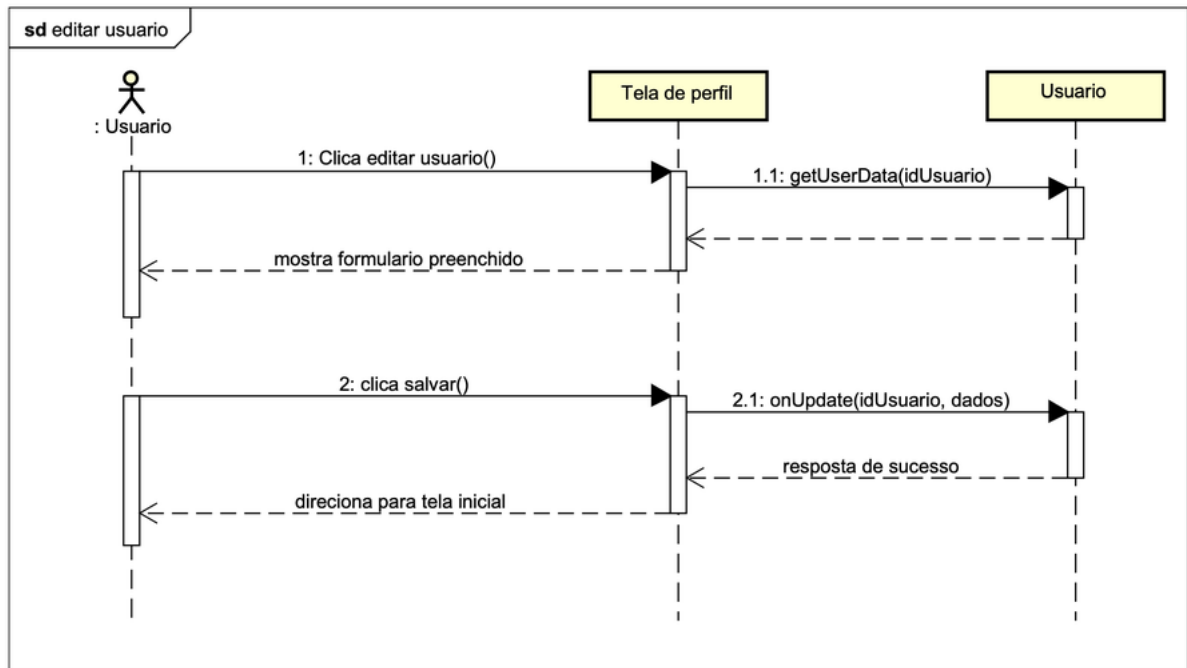


Figura 51 – DS02 - EFETUAR CADASTRO



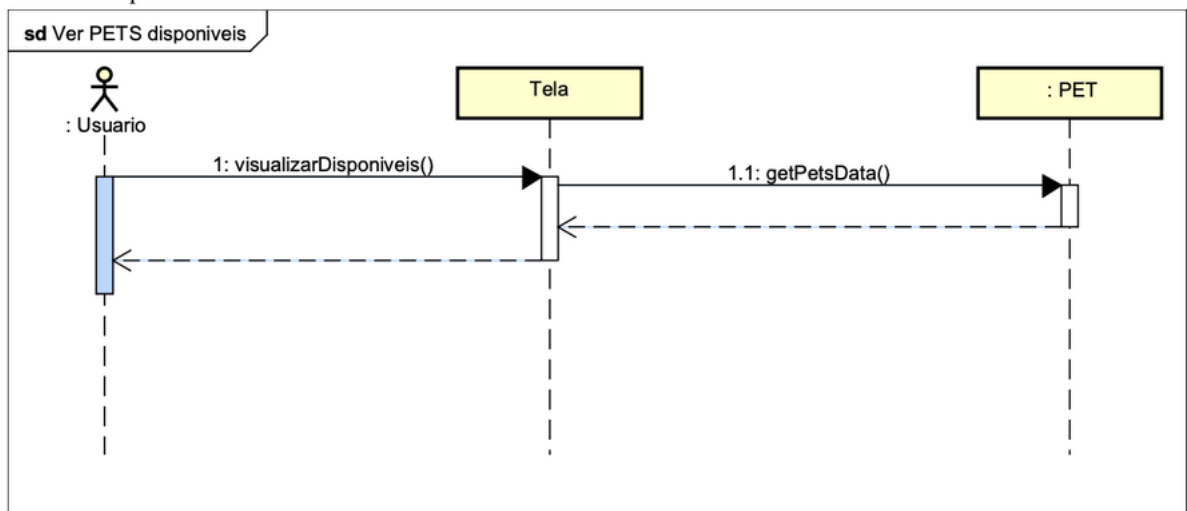
A AUTORA (2023)

Figura 52 – DS03 - EDITAR PERFIL



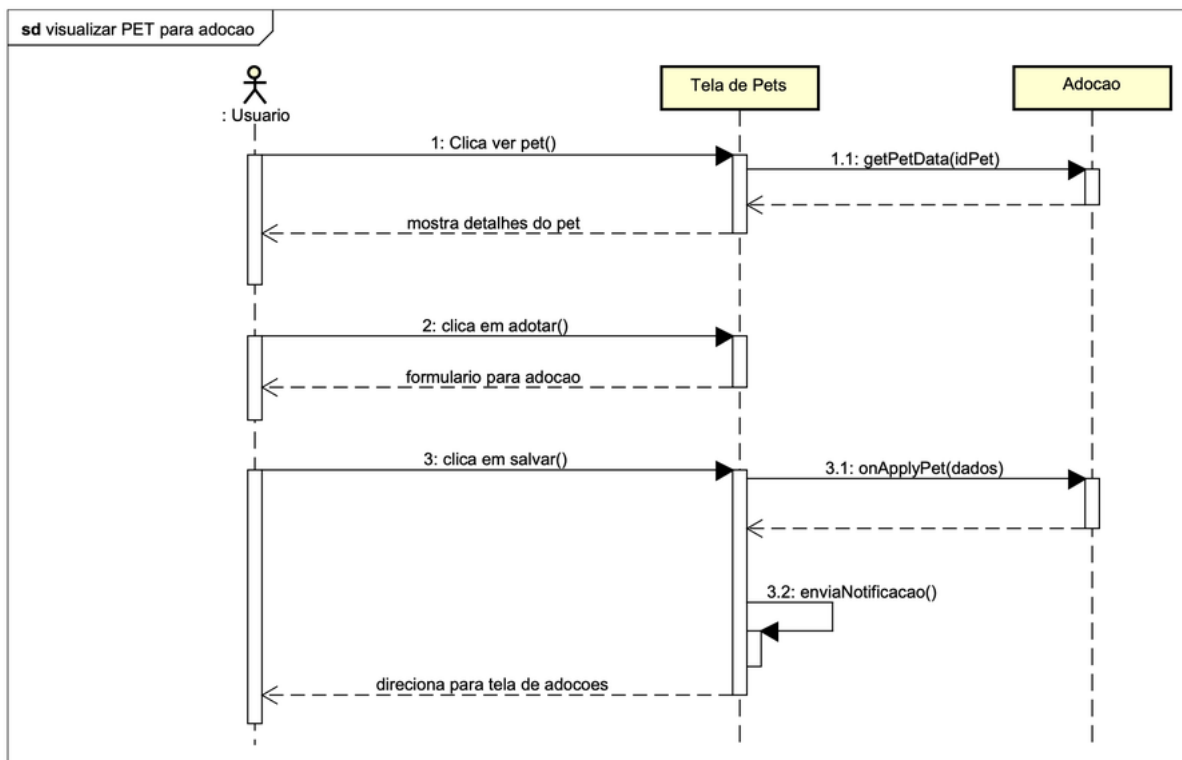
A AUTORA (2023)

Figura 53 – DS04 - VISUALIZAR PETS



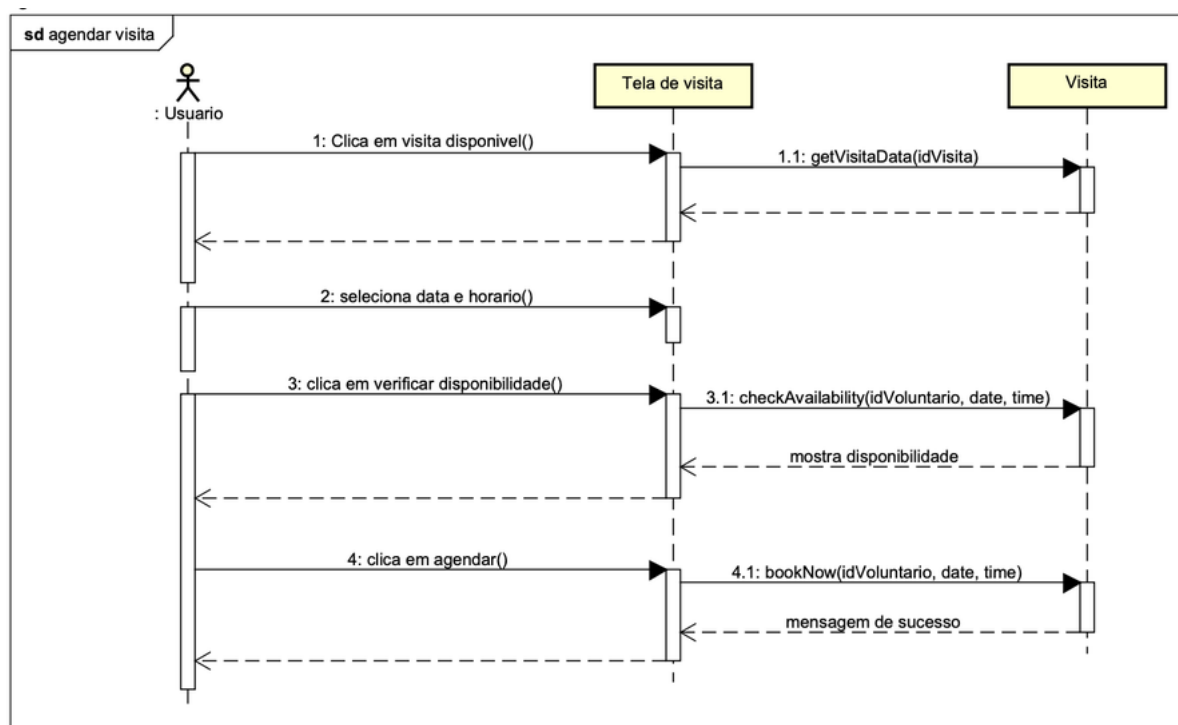
A AUTORA (2023)

Figura 54 – DS05 - SOLICITAR ADOCAO



A AUTORA (2023)

Figura 55 – DS06 - AGENDAR VISITA



A AUTORA (2023)

Figura 56 – DS07 - VISUALIZAR NOTIFICACOES

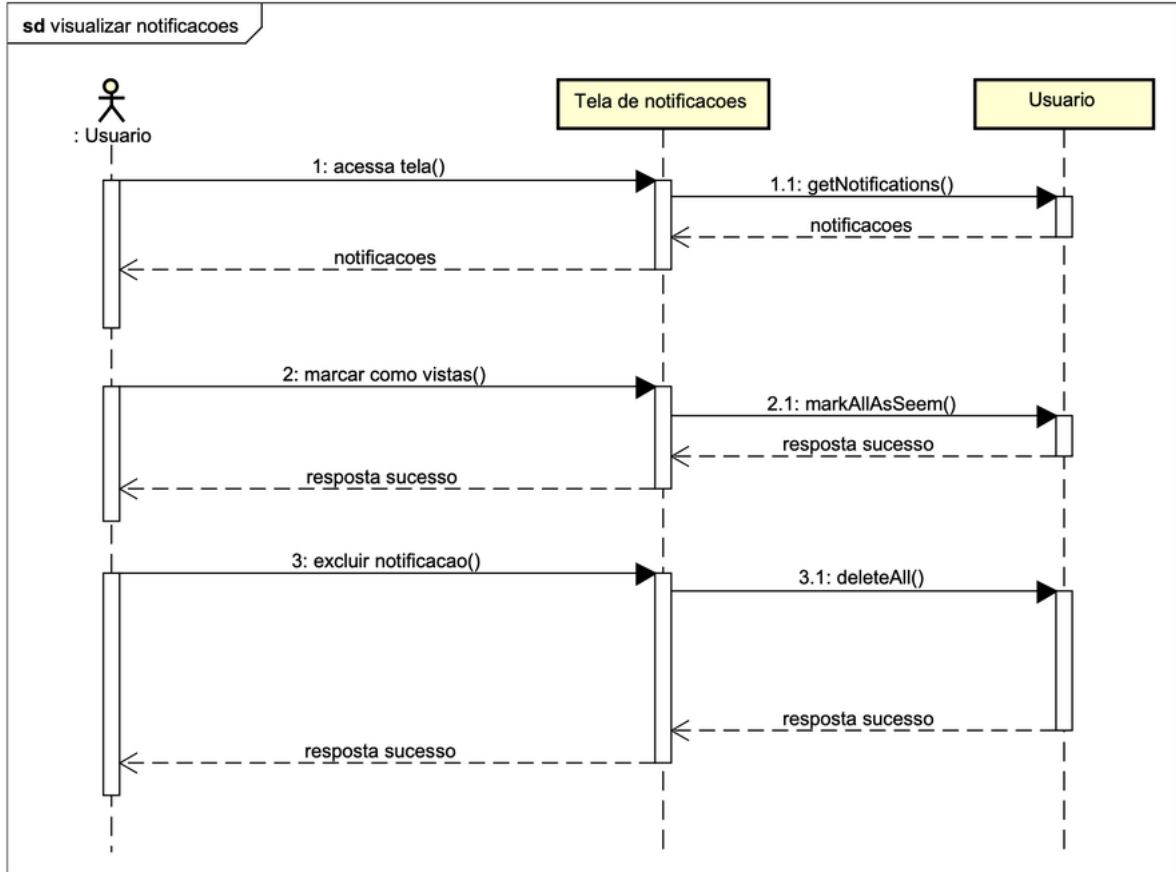
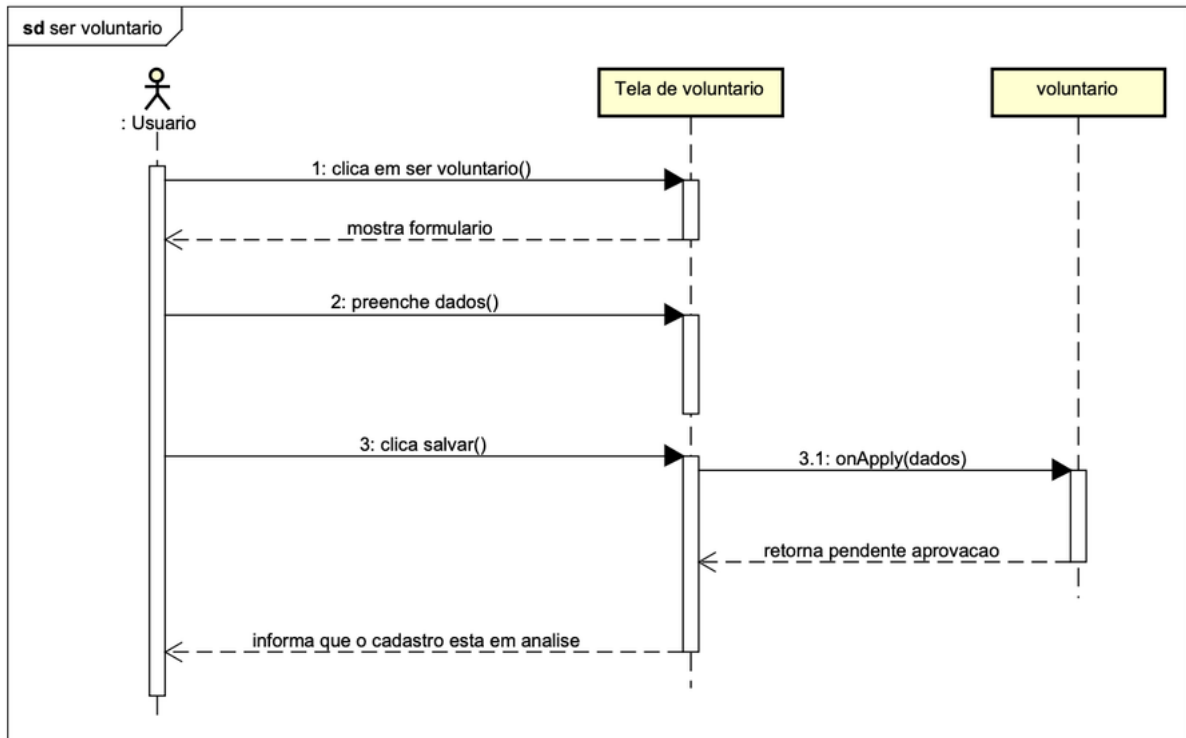
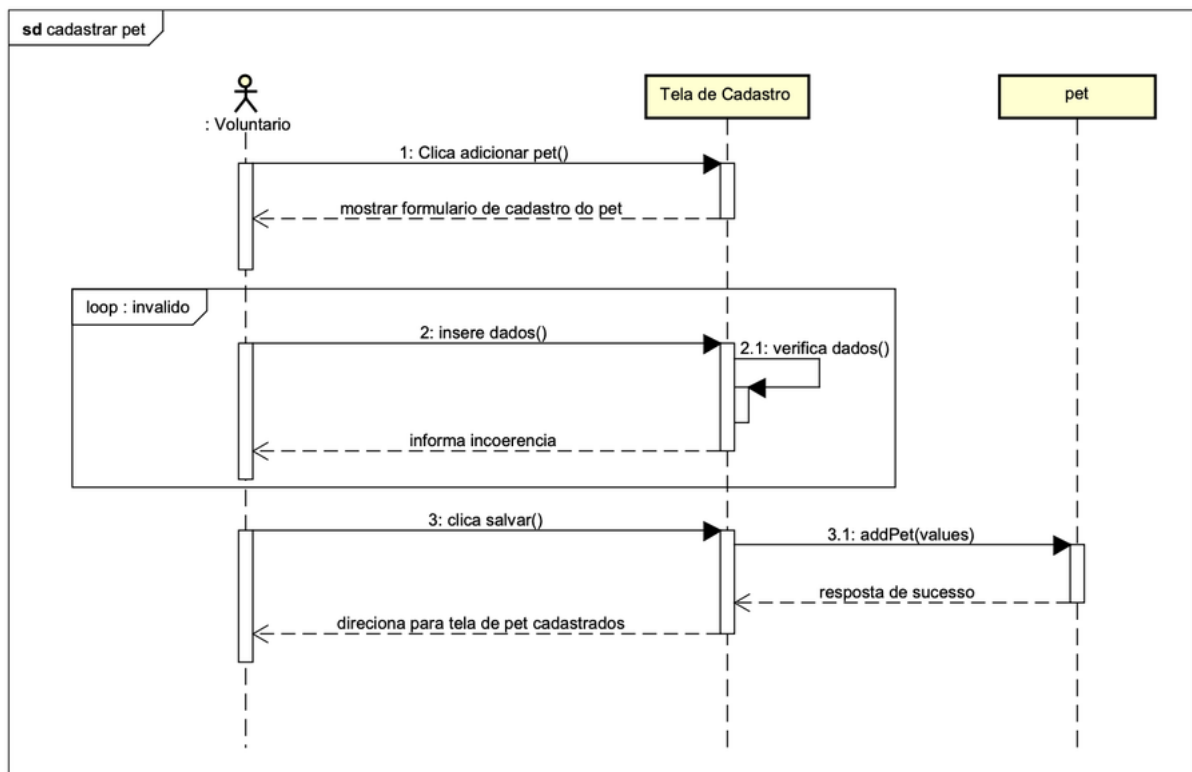


Figura 57 – DS08 - SER VOLUNTARIO



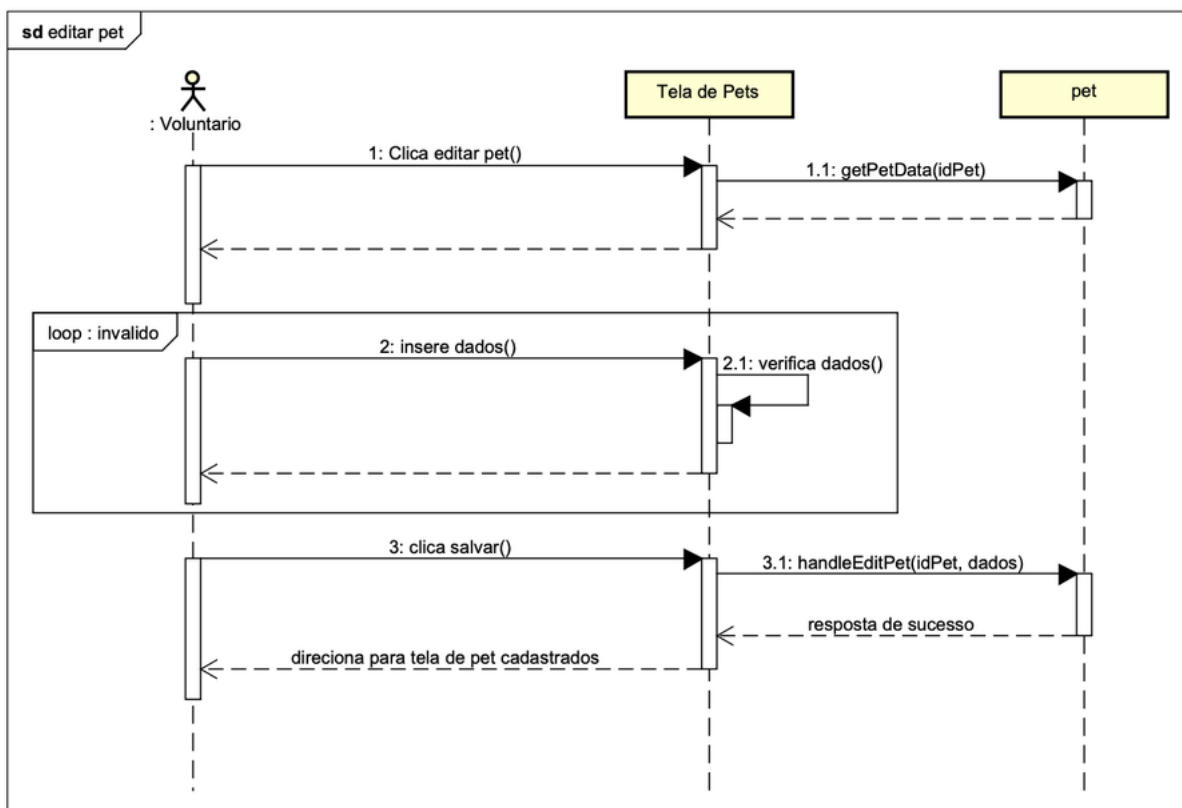
A AUTORA (2023)

Figura 58 – DS09 - CADASTRAR PET



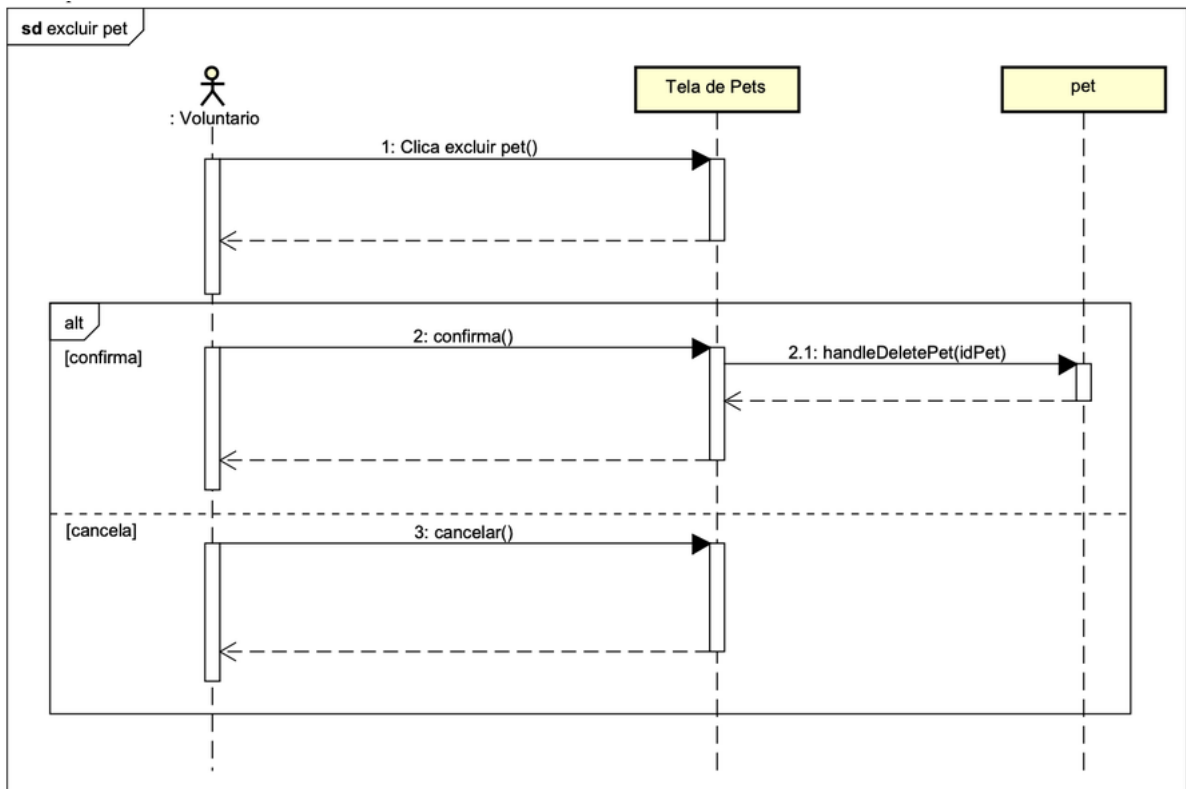
A AUTORA (2023)

Figura 59 – DS10 - EDITAR PET



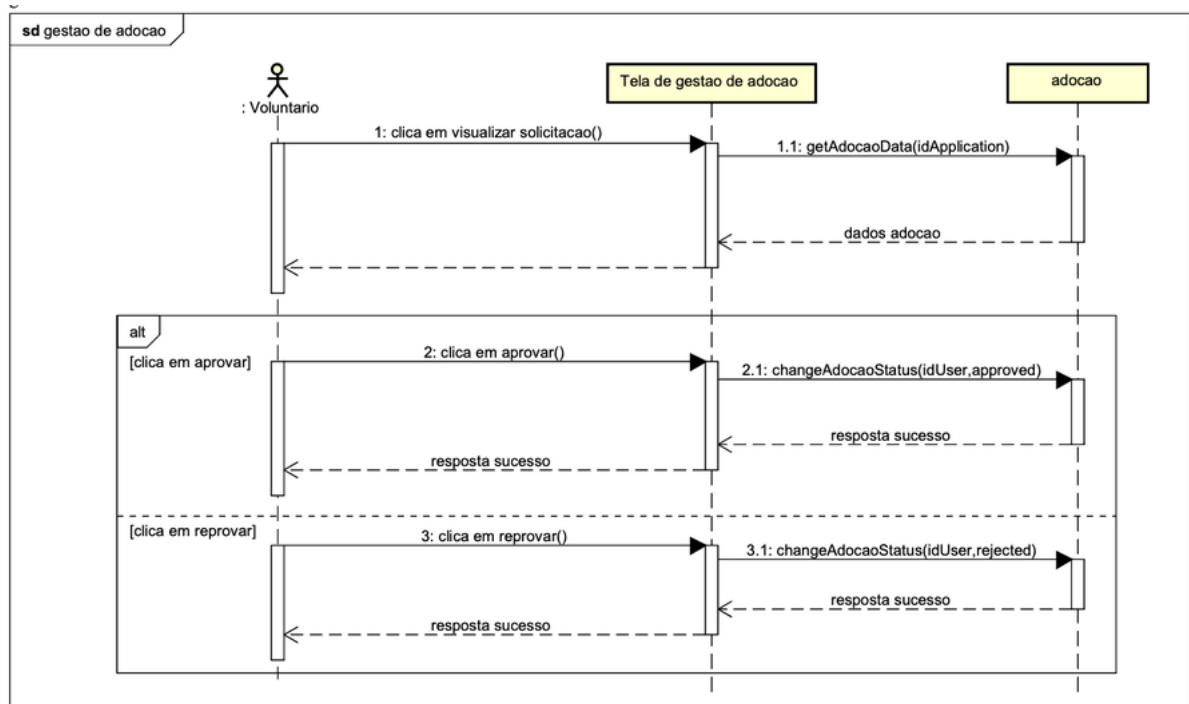
A AUTORA (2023)

Figura 60 – DS11 - EXCLUIR PET



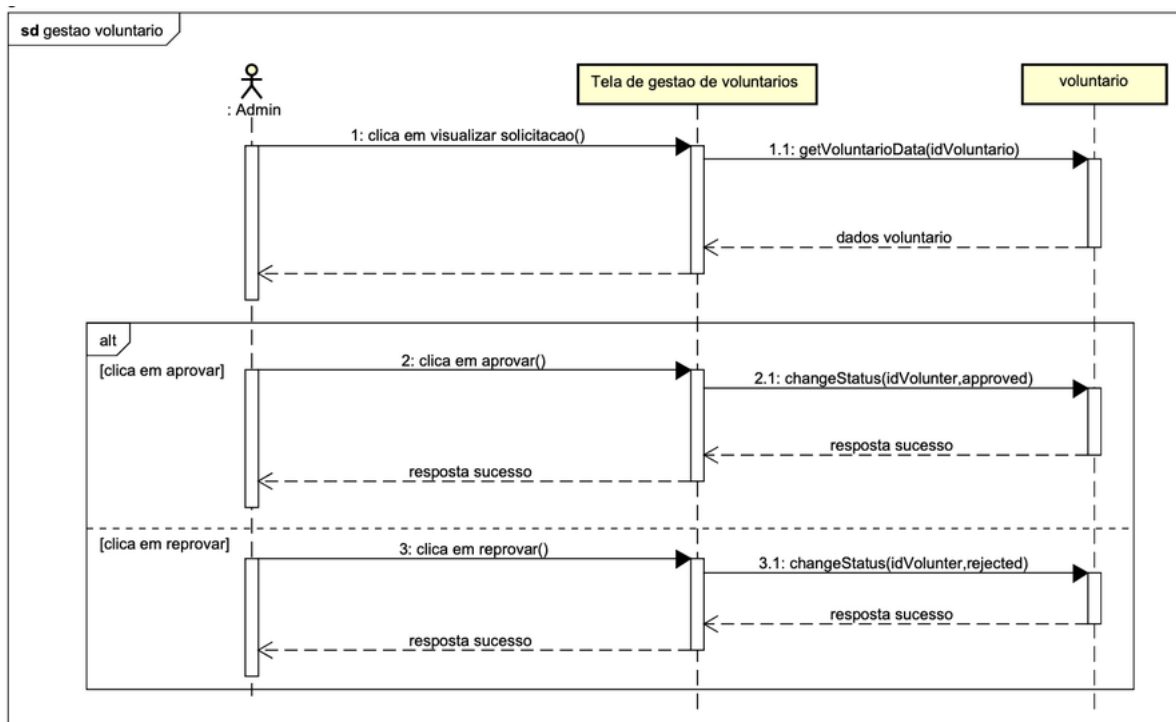
A AUTORA (2023)

Figura 61 – DS12 - GERENCIAR ADOCOES



A AUTORA (2023)

Figura 62 – DS13 - GERENCIAR VOLUNTARIO

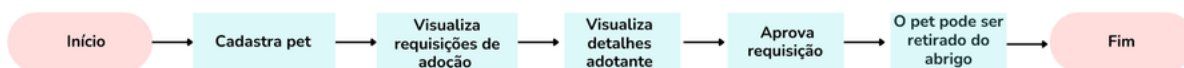


A AUTORA (2023)

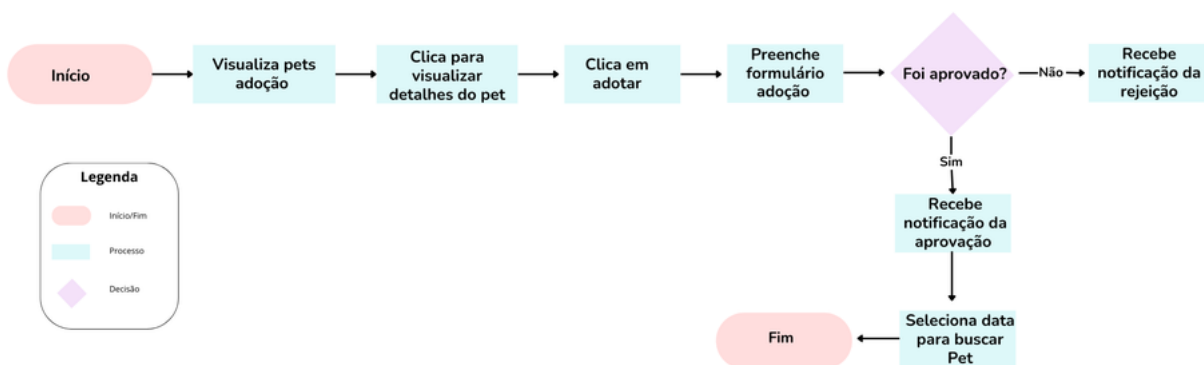
APÊNDICE F – FLUXOGRAMA PROCESSO ADOÇÃO

Figura 63 – FLUXOGRAMA PROCESSO ADOÇÃO

FLUXOGRAMA ADOÇÃO VOLUNTÁRIO



FLUXOGRAMA ADOÇÃO ADOTANTE



A AUTORA (2023)