

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**GABRIEL DE MORAIS TAVEIRA
GUILHERME CÉSAR BATISTA SOARES**

**MEAUDOTE: PROPOSTA DE UM WEBSITE PARA ADOÇÃO DE
ANIMAIS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Leonardo Henrique Raiz

FRANCA/SP

2024

MEAUDOTE: PROPOSTA DE UM WEBSITE PARA ADOÇÃO DE ANIMAIS

GABRIEL DE MORAIS TAVEIRA¹

GUILHERME CÉSAR BATISTA SOARES²

Resumo

A criação do projeto surgiu de forma natural, especialmente após observarmos diversos Trabalhos de Graduação em nossa turma. Ao refletirmos sobre o impacto desses projetos, decidimos focar em algo voltado para o público geral, e não apenas para o âmbito empresarial. Assim nasceu o "Meaudote," uma plataforma dedicada à adoção de animais. A escolha dessa plataforma foi impulsionada pela crescente preocupação com o abandono de animais em nossa cidade, além do nosso amor pelos animais. Com isso em mente, estruturamos o projeto para que seja acessível a todos, totalmente gratuito e com a intenção de integrar as ONGs locais que atuam na causa. Nosso objetivo foi desenvolver um software de fácil acesso, intuitivo e simples de usar, de modo que qualquer pessoa possa utilizá-lo sem dificuldades. Com essas metas bem definidas, chegamos à formulação do projeto apresentado a seguir como nosso Trabalho de Graduação.

Palavras-chave: Acessibilidade. Abandono de Animais. Adoção de Animais. Desenvolvimento de Sistemas. Intuitividade. ONGs.

Abstract

The creation of this project emerged naturally, especially after observing various Graduation Projects in our class. Reflecting on the impact these projects could have, we decided to focus on something directed at the public, rather than just the corporate world. This is how "Meaudote" was born—a platform dedicated to animal adoption. The choice of this platform was driven by the increasing concern over animal abandonment in our city, along with our love for animals. With this in mind, we structured the project to be accessible to everyone, completely free, and designed to integrate local NGOs working in this cause. Our goal was to develop software that's easy to access, intuitive,

and simple to use, ensuring that anyone can navigate it without difficulty. With these objectives clearly defined, we present this project as our Graduation Project.

¹ Graduando em Análise e Desenvolvimento de Sistema pela Fatec Dr Thomaz Novelino – Franca/SP.

Endereço eletrônico:gabriel.taveira@fatec.sp.gov.br.

² Graduando em Análise e Desenvolvimento de Sistema pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: guilhermecrs163@gmail.com

³ Professor da Fatec Dr Thomaz Novelinho – Franca/SP. Endereço eletrônico:

Leonardo.raiz@fatec.sp.gov.br.

Keywords: Accessibility. Animal Abandonment. Animal Adoption. Systems Development. Intuitiveness. NGOs.

1 INTRODUÇÃO

A crescente incidência de abandono e maus-tratos de animais é um problema que se intensifica com o passar dos anos em nossa cidade. Canis locais registram até 40 ocorrências de abandono de animais por mês, e esse número representa apenas as situações oficialmente notificadas. Além disso, por meio de pesquisas de campo, constatamos um número significativo de animais vivendo em condições de abandono nas ruas, evidenciando a gravidade da questão. Para entender melhor as dinâmicas de adoção, investigamos os meios mais comuns utilizados para esse fim. Observamos que canis e feiras de adoção são métodos tradicionais, mas plataformas digitais como Facebook, Instagram e WhatsApp também têm sido amplamente utilizadas para divulgar e facilitar a adoção de animais. Essas redes sociais abrigam diversas páginas dedicadas ao tema, tornando-se ferramentas importantes nesse processo. Diante desse cenário, propomos a criação de uma plataforma digital unificada que integre todas as publicações relacionadas à adoção de animais, atualmente dispersas em várias redes sociais. A ideia é centralizar as informações em um único ambiente, aumentando as chances de adoção e contribuindo para a redução do número de animais abandonados. Além de ser acessível, a plataforma terá um papel transformador, oferecendo a cães e gatos a oportunidade de encontrar um lar definitivo. O objetivo é contribuir para a diminuição do abandono de animais na cidade local.

O software foi concebido como parte do Trabalho de Graduação (TG), mas com potencial de evolução contínua após o término do curso. A versão inicial disponibilizada estará próxima da definitiva, com espaço para atualizações futuras, possibilitando o contato direto com os usuários e o aperfeiçoamento da experiência

com base no feedback. O foco foi não apenas na acessibilidade, mas também na usabilidade, garantindo que a plataforma seja intuitiva e de fácil acesso para todos.

Ferramentas e tecnologias estudadas no curso de Análise e Desenvolvimento de Sistemas foram utilizadas, como o JavaScript com o framework ReactJs na implementação do front-end, assim como o Java com o framework SpringBoot na implementação do back-end. A aplicação utiliza um banco de dados PostgreSQL, que é utilizado para o armazenamento de informações de usuário e dos pets registrados, a infraestrutura da aplicação será levantada utilizando o Terraform, uma tecnologia IAC (Infrastructure as Code) e será hospedada na Google Cloud Platform, trazendo segurança e escalabilidade dos recursos.

1.1 Termo de Abertura do Projeto (TAP)

Segundo o Guia PMBOK, principal referência em gerenciamento de projetos, existem várias práticas necessárias para se gerenciar um projeto de maneira eficiente, dentre elas, a elaboração do termo de abertura do projeto (TAP). O propósito do TAP é formalizar e estabelecer o início do empreendimento. O principal enfoque do projeto é a criação de um ambiente dedicado à adoção de animais. Para alcançar este objetivo, um sistema está sendo desenvolvido para possibilitar tanto (ONGs) quanto indivíduos a efetuarem doações e adoções de animais de estimação. A elaboração do site envolveu a utilização de diversas ferramentas, além da busca ativa por potenciais clientes interessados em aderir ao nosso sistema.

1.2 História da empresa

A ONG "Só Patinhas de Rua" tem sido uma presença constante na cidade de Franca, São Paulo, dedicada à proteção e cuidado dos animais de rua. Sua história é uma prova do poder da compaixão e do compromisso em fazer a diferença na vida dos animais mais vulneráveis.

Fundada pela Lindsay Cardoso apaixonada por animais, a "Só Patinhas de Rua" teve início como uma iniciativa de resgate e cuidado para os animais abandonados e maltratados que vagavam pelas ruas de Franca. Com um desejo compartilhado de fazer uma mudança positiva na comunidade, esses voluntários

dedicaram seu tempo, recursos e energia para oferecer esperança e ajuda aos animais desabrigados.

Desde o início, a missão da "Só Patinhas de Rua" tem sido clara: resgatar, cuidar e encontrar lares amorosos para animais em situação de rua. Seus objetivos incluem fornecer cuidados médicos, abrigo temporário, esterilização e adoção responsável, além de promover a conscientização sobre o bem-estar animal e a importância da esterilização.

Ao longo dos anos, a "Só Patinhas de Rua" expandiu suas operações e serviços para melhor atender às necessidades dos animais e da comunidade de Franca, São Paulo. Suas principais atividades incluem:

Resgate e Reabilitação: A ONG resgata animais em situação de abandono, maus-tratos ou perigo, fornecendo cuidados médicos, alimentação, abrigo temporário e reabilitação física e emocional.

Programas de Adoção: A ONG facilita a adoção responsável de animais resgatados, promovendo feiras de adoção, eventos de conscientização e campanhas nas redes sociais para encontrar lares amorosos para cada animal.

Educação e Sensibilização: A ONG realiza programas educacionais em escolas, empresas e eventos comunitários para promover o respeito pelos animais, incentivar a esterilização e vacinação, e conscientizar sobre os direitos dos animais.

Parcerias e Colaborações: A ONG trabalha em parceria com outras organizações locais, veterinários, empresas e voluntários para ampliar seu impacto e recursos, além de participar de iniciativas conjuntas em prol do bem-estar animal.

Ao longo de sua jornada, a "Só Patinhas de Rua" enfrentou diversos desafios, como falta de financiamento, infraestrutura limitada e resistência da comunidade. No entanto, com o apoio contínuo da comunidade e a dedicação incansável de seus voluntários, a ONG tem alcançado muitas conquistas significativas, salvando inúmeras vidas e inspirando outros a se envolverem na causa animal.

A história da "Só Patinhas de Rua" é um testemunho do poder da compaixão e do compromisso em fazer a diferença na vida dos animais mais necessitados. Com sua dedicação inabalável e trabalho incansável, a ONG continua a ser uma força positiva na comunidade de Franca, São Paulo, e um farol de esperança para todos os animais que cruzam seu caminho.

1.3 Objetivo do projeto

A falta de confiabilidade em um sistema próprio e gratuito para doações de animais é um problema recorrente. É notável que muitas plataformas suportam doações de animais, porém, não estão centralizadas em um site específico, sendo divulgadas em redes sociais, como Facebook e Instagram.

O objetivo é desenvolver uma aplicação que unifique tanto pessoas interessadas em doar quanto (ONGs) dedicadas à doação de animais. Além disso, também há uma busca em envolver pessoas que identificam animais em situação de abandono, oferecendo uma solução abrangente em uma única plataforma.

1.4 Justificativa do projeto

Após o levantamento de questões, não apenas com indivíduos, mas também com ONGs e pet shops, a ideia foi incentivada e considerada excelente. Não apenas simplificaria a vida das ONGs, mas também beneficiaria pessoas que desejam doar animais, seja em situação de abandono ou não. Ao discutir o que seriam funcionalidades cruciais no projeto, foi mencionado que seria interessante incluir uma opção de visita no site, juntamente com fotos e descrições dos animais. Foi demonstrado um entusiasmo com a ideia. A intenção é desenvolver um software gratuito com a missão de disseminar amplamente a divulgação deste serviço, visando consolidar um único ponto de referência para a adoção de animais. O cerne desta iniciativa é proporcionar às pessoas a oportunidade de estabelecerem contato direto com as organizações responsáveis pelo processo de adoção. Acreditamos que ao concentrar essas atividades em um local centralizado, conseguiremos otimizar e facilitar o processo de adoção de animais de estimação, promovendo uma abordagem mais eficiente e eficaz.

1.5 Descrição do projeto

A crescente preocupação com o bem-estar animal tem impulsionado a necessidade de soluções eficazes para promover a adoção responsável de animais de estimação. Neste contexto, o desenvolvimento de um sistema de adoção de animais surge como uma ferramenta crucial para conectar animais resgatados ou em abrigos a potenciais adotantes, proporcionando uma plataforma intuitiva e eficiente

para facilitar o processo de adoção, não apenas facilitando mais deixando em apenas um sistema ao invés de fragmentado o processo de adoção, sendo também 100% gratuito, para que qualquer um tenha acesso.

1.6 Funcionalidades

O programa inclui diversas funcionalidades para gerenciar o processo de adoção, proporcionando uma experiência simples e organizada para usuários e administradores. Uma das principais funcionalidades é o registo dos animais, que permite aos operadores inserir informações detalhadas sobre os animais, como raça, tipo, idade e tamanho, informações que ajudam a desenvolver um perfil para cada animal. Outra parte importante é a criação de perfis de usuário, onde é possível aumentar a compatibilidade entre adotante e animal criando um perfil pessoal que fornece informações importantes sobre o processo de adoção, incluindo histórico de adoção, localização e espaço disponível. Para facilitar, o sistema conta com uma função de busca avançada, permitindo aos usuários filtrarem os animais disponíveis com base em critérios específicos, como raça, espécie, idade e localização, podendo encontrar rapidamente um animal que atenda às suas necessidades e desejos. A opção de enriquecer o perfil também fornece uma maneira de completar perfis de animais, incluindo fotos, descrições detalhadas e outras informações relevantes, permitindo aos usuários obter uma visão completa do animal com antecedência. Esses recursos foram projetados para trazer uma navegação mais fácil e aumentar a segurança na hora da adoção do animal, e abaixando o arrependimento após adoção.

1.7 Riscos

Durante o desenvolvimento e implementação do programa de adoção de animais, foram identificados fatores de risco que poderiam impactar a implementação e o sucesso do programa. Esses riscos são divididos em quatro categorias: segurança da informação, operações, gestão e tecnologia.

Um dos principais riscos é a segurança da informação, que inclui a possibilidade de exposição de informações pessoais dos usuários, como dados de contato, histórico de login e outras informações confidenciais. Foram tomadas fortes medidas de segurança para proteger essas informações, como criptografia durante o armazenamento e transmissão, e diversas outras medidas para acesso ao sistema.

Essas precauções foram projetadas para garantir a privacidade e segurança do usuário e proteger as informações contra acesso não autorizado.

Outro risco importante é o desempenho, especialmente o risco de fraude. Podem haver fraudes ou erros no processo de adoção, como uma adoção que não requer supervisão adequada ou um processo abusivo, para evitar casos como este, foi implementado um sistema de monitoramento dos adotantes e acompanhamento pós-adoção, para garantir que atendam às condições necessárias para manter os animais seguros.

Também foram identificados riscos de conflitos de interesses entre as unidades de gestão, operadores de sistemas e ONGs parceiras. Esse risco se manifesta, por exemplo, quando administradores priorizam a adoção de determinados animais de estimação e favorecem certas ONGs mais do que outras. Para evitar isso, foram definidas diretrizes claras e imparciais de governança que assegurarão tratamento justo e equitativo para todas as ONGs e adotantes.

Em relação aos riscos técnicos, há a possibilidade de falhas no sistema, como interrupções no serviço, lentidão ou mau funcionamento, o que pode comprometer a experiência do usuário e a eficiência do processo de adoção. Para minimizar esses riscos, foram adotadas práticas de manutenção contínua e monitoramento proativo do sistema, além de criar um plano de contingência para falhas potenciais. A infraestrutura em nuvem também será utilizada para garantir alta disponibilidade e escalabilidade, reduzindo o impacto das interrupções de serviço.

Por fim, há o risco de falhas na comunicação entre os potenciais adotantes e as ONGs, o que pode resultar em adoções malsucedidas ou desentendimentos durante o processo. Para resolver esse problema, foi pensado em um sistema de notificações automatizadas e um sistema de comunicação direta na plataforma para tornar a comunicação entre os participantes mais clara e eficiente, ainda não está presente nessa versão, a funcionalidade será adicionada mais à frente.

Esses riscos foram cuidadosamente considerados durante o processo de planejamento do projeto, e medidas foram tomadas para minimizar seu impacto. Com esses sistemas de gestão de risco, é esperado que o processo de adoção possa ser conduzido de forma segura e eficiente, proporcionando uma experiência satisfatória para todos os usuários e, acima de tudo, ajudando a proteger os animais envolvidos no processo de adoção.

1.8 Cronograma

A seguir o cronograma usado para organização de tarefas na Tabela abaixo:

Tabela 1- Cronograma de Tarefas

FEVEREIRO	QUESTÃO PROBLEMA SOLUÇÃO SISTÊMICA MISSÃO/ VISÃO E VALORES SEGMENTO PERGUNTAS - QUESTÕES
MARÇO	5W2H SWOT EAP TAP BPMN
ABRIL	DOCUMENTAÇÃO DE REQUISITOS DIAGRAMAS EM GERAL PROTOTIPÇÃO DE TELAS
JUNHO	MATRIZ DE RASTREABILIDADE DOCUMENTAÇÃO DE PORTABILIDADE MÉTRICAS PROPOSTA COMERCIAL

Fonte: Autoria própria.

1.9 Custos

Não foram necessários custos com a plataforma, foram usados todos softwares e aplicativos de forma gratuita. Foram usados apenas computadores dos próprios responsáveis e plataformas disponibilizadas pela internet e aprendidas durante o curso de Análise e Desenvolvimento de Sistemas.

2 CANVAS DE NEGÓCIO (BUSINESS MODEL CANVAS - BMC)

O BMC, conhecido como "Modelo Canvas", é uma metodologia utilizada para projetar um modelo de negócios, tornando claros os pontos fortes da empresa ou ideia e identificando áreas que podem ser aprimoradas. Esse modelo é amplamente utilizado por diversas empresas, pois é eficaz ao fornecer uma visão abrangente do negócio. É composto por nove componentes essenciais para definir e avaliar se o modelo de negócios está sendo bem aplicado à ideia. Esses componentes são:

Proposta de Valor, Segmento de Clientes, Canais, Relacionamento com Clientes, Fontes de Receita, Recursos Chave, Atividades Chave, Parcerias Chave e Estrutura de Custos. Com esses nove elementos, é possível determinar se o negócio pode ser viável e promissor, sendo uma excelente ferramenta tanto para empresas estabelecidas quanto para startups.

“O Business Model Canvas é uma linguagem comum para descrever, visualizar, avaliar e alterar modelos de negócios. Ele descreve a lógica de criação, entrega e captura de valor por uma organização.” (OSTERWALDER; PIG-NEUR, 2011, p. 15)

A Figura 1 ilustra do modelo de negócios BMC:

Figura 1 - Modelo Canvas



Fonte: Autoria própria.

3 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos, como o próprio nome sugere, é uma das partes mais importantes da elicitação, nesse momento que é possível moldar um grande projeto começando com uma estrutura simples. Tudo depende das perguntas feitas e de como o cliente é compreendido. Muitas vezes os usuários não sabem exatamente o que querem; como expressar isso. Portanto, é fundamental adaptar as perguntas para chegar onde os clientes realmente querem.

3.1 Elicitação e Especificação

Os alunos envolvidos no projeto perceberam que, em muitos trabalhos de outros universitários o foco principal era o desenvolvimento de softwares voltados para o setor empresarial. Diante disso, sentiram a necessidade de criar algo que fosse direcionado ao público em geral, algo que trouxesse impacto social, fosse de fácil acesso e, acima de tudo, gratuito. A partir dessa ideia, surgiu a proposta de desenvolver uma plataforma voltada para a adoção de animais, com o intuito de preencher uma defasagem observada nesse segmento.

Atualmente, a adoção de animais é amplamente divulgada através de redes sociais, como Facebook, Instagram e WhatsApp. No entanto, o uso dessas plataformas, embora seja bastante ativo, se fragmenta em diversas páginas e grupos, o que dificulta a centralização das informações e ações voltadas à causa animal. Muitos anúncios de adoção e resgate acabam perdidos ou pouco acessados, uma vez que não existe um ponto central específico para reunir e divulgar tais iniciativas de forma organizada, ficando muitas vezes posts em apenas uma plataforma, não sendo feita a mesma divulgação em outra.

Com base nessa observação, os alunos iniciaram uma pesquisa aprofundada e buscaram contato com pessoas diretamente envolvidas na causa de proteção animal na cidade de Franca, São Paulo. O primeiro contato foi estabelecido com a responsável pela ONG "Cão que Mia", uma organização local que trabalha no resgate e cuidado de animais abandonados. Durante a conversa, a responsável pela ONG ressaltou que levantar fundos para a causa é uma tarefa difícil e que, geralmente, o trabalho é feito por amor aos animais, sem esperar retorno financeiro. Ela destacou que o principal objetivo da ONG é proporcionar um lar adequado e amoroso para os animais, trazendo felicidade para suas novas famílias.

Quando perguntada sobre como uma plataforma poderia auxiliar na causa, especialmente em relação à adoção de animais, a resposta foi positiva. Ela destacou que uma ferramenta dedicada a esse propósito seria de grande ajuda, pois poderia aumentar a visibilidade da ONG e agilizar o processo de adoção. Com uma plataforma centralizada, o tempo em que os animais ficariam em situação de abandono poderia ser reduzido, já que a divulgação alcançaria mais pessoas interessadas em adotar. Além disso, a plataforma poderia ser usada não apenas para adoções, mas também

para informar e conscientizar a população sobre os cuidados com os animais e as formas de apoiar a causa.

Após essa entrevista inicial, os alunos buscaram contato com outra pessoa influente na causa animal em Franca, Lindsey Cardoso, responsável pela ONG "Só Patinhas de Rua". Lindsey contou que tem trabalhado na proteção animal há muitos anos e que essa é uma luta constante, enfrentando diversas dificuldades, como a falta de recursos e o grande número de animais abandonados ou maltratados. Ela relatou que administra canis onde cuida de dezenas de animais e que, em média, cada canil abriga mais de 40 cães resgatados. Além disso, Lindsey relatou que, frequentemente, encontra animais em situações de maus-tratos e abandono nas ruas, o que aumenta ainda mais a demanda por recursos e cuidados.

A partir da conversa com Lindsey, os alunos apresentaram a proposta da plataforma de adoção, buscando entender como essa ferramenta poderia impactar o trabalho das ONGs. Lindsey mostrou-se bastante receptiva à ideia, destacando que uma plataforma específica para adoções não apenas traria mais visibilidade para as ONGs e para a causa animal, mas também facilitaria o processo de adoção e conscientização. Segundo Lindsey, quanto mais pessoas acessarem e conhecerem a plataforma, maior será a chance de os animais encontrarem lares, o que poderia, por sua vez, reduzir o número de animais em situação de abandono. Além disso, a plataforma poderia ser uma ferramenta importante para conectar as ONGs com outras empresas e voluntários que desejam apoiar a causa, promovendo parcerias e fortalecendo a rede de proteção animal.

Lindsey também mencionou que a plataforma poderia servir como um espaço para que as pessoas que precisam de ajuda com seus animais, como em casos de doações ou resgates, pudessem encontrar apoio de forma rápida e eficiente. Ela ressaltou que uma ferramenta centralizada e bem estruturada poderia melhorar significativamente a visibilidade das ONGs e facilitar o contato entre pessoas interessadas em adoção e as organizações que cuidam dos animais.

Ao final da conversa, Lindsey convidou os alunos a visitarem um dos canis administrados por ela, para que pudessem conhecer de perto a realidade do trabalho de resgate e cuidado com os animais. No entanto, devido a conflitos de agenda, a visita não pôde ser realizada naquele momento, mas os alunos se comprometeram a encontrar uma nova data para a visita, a fim de aprofundar seu entendimento sobre

as necessidades da ONG e explorar mais formas de a plataforma atender essas demandas.

Com essas entrevistas e pesquisas realizadas, os alunos confirmaram que há uma grande demanda por uma plataforma que centralize informações e facilite o processo de adoção de animais, além de promover maior visibilidade para a causa animal. As conversas com as ONGs reforçaram a ideia de que a criação de uma plataforma gratuita e acessível não só ajudaria as ONGs a divulgarem seu trabalho, mas também tornaria mais fácil para as pessoas encontrarem informações sobre adoção e sobre como apoiar a causa, fortalecendo o engajamento em torno da proteção animal.

3.2 BPMN (Business Process Model and Notation)

O BPMN é uma notação gráfica que facilita a visualização do fluxo de processos de um sistema, beneficiando tanto pessoas com conhecimento técnico quanto aquelas com experiência empresarial. Permite que o processo seja apresentado de forma intuitiva e clara, possibilitando até mesmo que indivíduos sem familiaridade técnica compreendam o funcionamento do sistema.

Simplifica o planejamento do sistema, indicando o número de telas e descrevendo as ações que cada usuário realizará. Dessa forma, torna-se uma ferramenta prática tanto para profissionais da área quanto para a apresentação a pessoas sem grande conhecimento técnico. Embora similar a um fluxograma, é mais completo e detalhado, cabendo ao programador aplicá-lo de maneira clara para facilitar a compreensão de todos.

Não deve ser visto como uma limitação para a empresa, mas sim como uma ferramenta explicativa que pode ajudar a reduzir processos e gastos desnecessários.

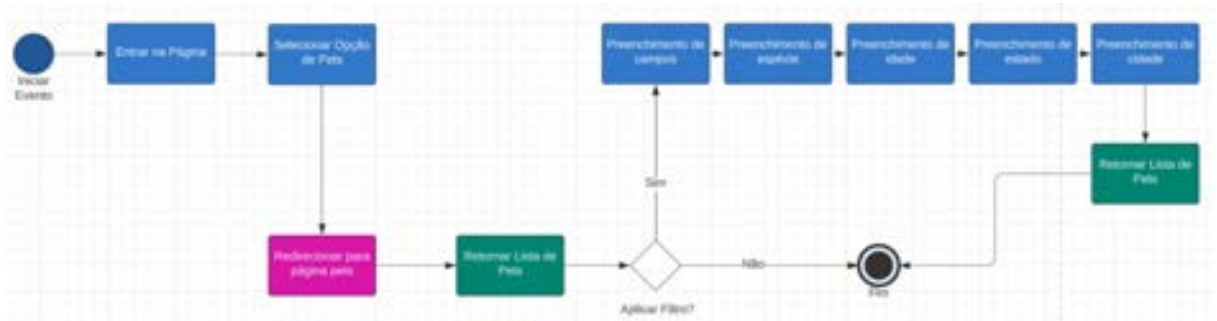
"Uma linguagem amplamente utilizada para modelagem de processos, proporcionando uma visão clara dos fluxos e atividades dentro de uma organização, o que facilita a implementação de melhorias". (CAMPOS, André L. N. Modelagem de Processos com BPMN. 2ª ed. Rio de Janeiro: Brasport, 2014, p. 15.)

Figura 2 - BPMN Autenticação



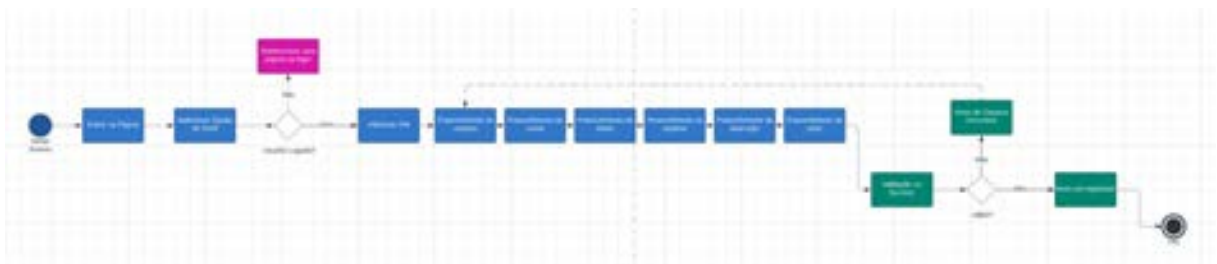
Fonte: Autoria própria.

Figura 3 - BPMN Pets



Fonte: Autoria própria.

Figura 4 - BPMN Registrar Pet



Fonte: Autoria própria.

3.3 Requisitos Funcionais

Requisitos funcionais são parte fundamental do processo de elicitação, que identifica os problemas e as necessidades que o software ou serviço deve atender. Nesta etapa, é necessário definir com precisão todas as funções e ações esperadas do sistema, assegurando que ele ofereça as funcionalidades que atendem às demandas dos usuários e do negócio. Cada função descrita deve incluir detalhadamente o que a ação realiza, quais são as entradas e saídas esperadas, e

como ela contribuí para o objetivo geral do sistema. Essa especificação minuciosa reduz riscos de erros durante o desenvolvimento, aumenta a qualidade final e aprimora a segurança do sistema, uma vez que suas funcionalidades são claramente compreendidas e podem ser testadas com precisão.

Pressman afirma que "os requisitos funcionais descrevem as funções que o sistema deverá realizar, como processar dados, reagir a entradas, executar cálculos e interagir com outros sistemas ou usuários. Esses requisitos são fundamentais para a concepção do sistema e para a comunicação entre desenvolvedores e usuários. "Ele também destaca que os requisitos devem ser específicos, testáveis e compreensíveis para todas as partes envolvidas. (PRESSMAN, 2010, p. 63).

Quadro 1 - Requisitos Funcionais do Sistema

ID: RF001	Nome do Requisito: Cadastrar Usuário
Descrição →	Registro de novos usuários no sistema.
Categoria: Evidente	Prioridades: Essencial.
Informações →	<ul style="list-style-type: none"> • CPF: Caixa de texto com máscara de CPF e validação de formato. • Nome de Usuário: Caixa de texto simples. • Senha: Caixa de texto de senha. • Confirmar Senha: Caixa de texto de senha com validação de igualdade com o campo Senha. • Email: Caixa de texto de email com validação de formato. • Celular: Campo numérico com validação de formato e máscara de celular. • Estado: Campo de seleção com opções pré-definidas. • Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo "Estado". • Sou ONG: Campo com caixa de seleção opcional.
Regras do Negócio:	<ul style="list-style-type: none"> • CPF é um valor único. • Email é um valor único. • Nome de usuário é um valor único. • Celular é um valor único.
ID: RF002	Nome do Requisito: Autenticar Usuário.
Descrição →	O usuário já cadastrado pode se autenticar no sistema.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Email: Campo de texto simples com validação de e-mail. • Senha: Campo de senha.
Regras do Negócio:	<ul style="list-style-type: none"> • Email e senha devem corresponder a credenciais de um usuário existente.
ID: RF003	Nome do Requisito: Editar Perfil.
Descrição →	O usuário já cadastrado pode editar seu perfil.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Nome de Usuário: Caixa de texto simples. • Celular: Campo numérico com validação de formato e máscara de celular. • Estado: Campo de seleção com opções pré-definidas.

	<ul style="list-style-type: none"> • Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo “Estado”. • Sou ONG: Campo com caixa de seleção opcional. • Imagem de Perfil: Campo de seleção de arquivo com validação de formato. • Imagem de Capa: Campo de seleção de arquivo com validação de formato.
Regras do Negócio:	<ul style="list-style-type: none"> • Usuário deve estar autenticado para salvar o registro. • Imagem com tamanho limite de 5mb.
ID: RF004	Nome do Requisito: Registrar Chave Pix.
Descrição →	O usuário pode registrar uma chave pix para receber pagamentos.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Tipo: Caixa de seleção com opções cpf/cnpj, email, celular, ou chave aleatória. • Chave: Caixa de texto.
Regras do Negócio:	<ul style="list-style-type: none"> • Todos os campos são obrigatórios. • Validação de chave baseada no tipo escolhido.
ID: RF005	Nome do Requisito: Editar Chave Pix.
Descrição →	O usuário pode editar uma chave pix já cadastrada.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Chave: Caixa de texto.
Regras do Negócio:	<ul style="list-style-type: none"> • Todos os campos são obrigatórios. • Validação de chave baseada no tipo escolhido.
ID: RF006	Nome do Requisito: Excluir Chave Pix.
Descrição →	O usuário pode excluir uma chave pix já cadastrada.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Seleção de chave e janela de confirmação de exclusão.
Regras do Negócio:	<ul style="list-style-type: none"> • É necessário estar logado
ID: RF007	Nome do Requisito: Criar Página de Doação.
Descrição →	O usuário pode criar uma página para arrecadar doações.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Título: Caixa de texto simples. • Descrição: Caixa de texto simples. • Valor Desejado: Campo Numérico.
Regras do Negócio:	<ul style="list-style-type: none"> • Todos os valores são obrigatórios.
ID: RF008	Nome do Requisito: Editar Página de Doação.
Descrição →	O usuário pode editar uma página de doações já registrada.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Título: Caixa de texto simples. • Descrição: Caixa de texto simples. • Valor Desejado: Campo Numérico.
Regras do Negócio:	<ul style="list-style-type: none"> • Todos os valores são obrigatórios.
ID: RF009	Nome do Requisito: Excluir Página de Doação.
Descrição →	O usuário pode excluir uma página de doação já cadastrada.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Seleção de página e janela de confirmação de exclusão.
Regras do Negócio:	<ul style="list-style-type: none"> • É necessário estar logado.
ID: RF010	Nome do Requisito: Listar Páginas de Doação.
Descrição →	O usuário pode ver um feed com diversas páginas de doação cadastradas por usuários.
Categoria: Evidente	Prioridades: Importante.

Informações →	<ul style="list-style-type: none"> Mostrar feed com páginas de doação.
Regras do Negócio:	<ul style="list-style-type: none"> É necessário estar logado.
ID: RF011	Nome do Requisito: Listar Páginas de Doação.
Descrição →	O usuário pode ver um feed com diversas páginas de doação cadastradas por usuários.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> Mostrar feed com páginas de doação.
Regras do Negócio:	<ul style="list-style-type: none"> É necessário estar logado.
ID: RF012	Nome do Requisito: Registrar Animal para Adoção.
Descrição →	O usuário cadastrado pode registrar um animal para adoção.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> Nome de Usuário: Caixa de texto simples. Descrição: Caixa de texto simples. Idade: Campo numérico. Imagem: Campo de seleção de arquivo com validação de formato. Estado: Campo de seleção com opções pré-definidas. Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo "Estado". Peso: Campo numérico. Espécie: Campo de seleção com opções pré-definidas.
Regras do Negócio:	<ul style="list-style-type: none"> Usuário deve estar autenticado para registrar o animal. Imagem com tamanho limite de 5mb.
ID: RF013	Nome do Requisito: Editar Animal para Adoção.
Descrição →	O usuário cadastrado pode editar uma página de adoção.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> Nome de Usuário: Caixa de texto simples. Descrição: Caixa de texto simples. Idade: Campo numérico. Imagem: Campo de seleção de arquivo com validação de formato. Estado: Campo de seleção com opções pré-definidas. Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo "Estado". Peso: Campo numérico. Espécie: Campo de seleção com opções pré-definidas.
Regras do Negócio:	<ul style="list-style-type: none"> Usuário deve estar autenticado para registrar o animal. Imagem com tamanho limite de 5mb.
ID: RF014	Nome do Requisito: Marcar Como Adotado
Descrição →	O usuário cadastrado pode editar uma página de adoção.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> Nome de Usuário: Caixa de texto simples. Descrição: Caixa de texto simples. Idade: Campo numérico. Imagem: Campo de seleção de arquivo com validação de formato. Estado: Campo de seleção com opções pré-definidas. Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo "Estado". Peso: Campo numérico. Espécie: Campo de seleção com opções pré-definidas.
Regras do Negócio:	<ul style="list-style-type: none"> Usuário deve estar autenticado para registrar o animal. Imagem com tamanho limite de 5mb.

ID: RF015	Nome do Requisito: Marcar Como Adotado
Descrição →	O usuário cadastrado pode editar uma página de adoção.
Categoria: Evidente	Prioridades: Importante.
Informações →	<ul style="list-style-type: none"> • Nome de Usuário: Caixa de texto simples. • Descrição: Caixa de texto simples. • Idade: Campo numérico. • Imagem: Campo de seleção de arquivo com validação de formato. • Estado: Campo de seleção com opções pré-definidas. • Cidade: Campo de seleção com opções pré-definidas baseadas na escolha do campo “Estado”. • Peso: Campo numérico. • Espécie: Campo de seleção com opções pré-definidas.
Regras do Negócio:	<ul style="list-style-type: none"> • Usuário deve estar autenticado para registrar o animal. • Imagem com tamanho limite de 5mb.

Fonte: Autoria Própria

3.4 Requisitos Não Funcionais

Os requisitos não funcionais são aqueles que se relacionam a forma como o software se tornando realidade, enquanto os funcionais estão focados no sistema no que será feito, os não funcionais descrevem como serão feitos, esses requisitos atendem as necessidades e especificações do usuário, sendo um complemento e continuação dos requisitos funcionais tendo características como segurança, desempenho e escalabilidade sendo importantes para o bom funcionamento do sistema.

Quadro 2 - Requisitos Não Funcionais

ID: RNF001	Nome do Requisito: Banco de dados
Descrição →	O sistema necessitará de banco de dados PostgreSQL para armazenar os dados registrados.
Categoria: Externo	Prioridades: Essencial.
Informações →	<ul style="list-style-type: none"> • Cadastrar: Inserção de dados. • Consultar /Modificar: Atualização de dados. • Consultar /Excluir: Exclusão de dados. • Consultar /Ler: Exibição de dados.
Regras do Negócio:	Não há.
ID: RNF002	Nome do Requisito: Infraestrutura
Descrição →	O sistema necessitará de uma infraestrutura para poder receber e servir a aplicação web
Categoria: Produto	Prioridades: Essencial.
Informações →	<ul style="list-style-type: none"> • Serviço do Google Cloud Run. • Vercel. • Instância de banco de dados. • Serviços de CI/CD para trazer facilidade no desenvolvimento. • Cloud Storage para armazenamento de imagens.

Regras do Negócio:	Não há.
ID: RNF003	Nome do Requisito: Aplicação WEB
Descrição →	O sistema será acessado através do navegador.
Categoria: Organizacional	Prioridades: Essencial.
Informações →	<ul style="list-style-type: none"> • Navegador WEB. • Conexão com a internet. • Computador ou celular em funcionamento.
Regras do Negócio:	O sistema terá compatibilidade mobile e desktop.

Fonte: A autoria própria.

3.5 Casos de Uso

O modelo de caso de uso descreve a maneira como cada usuário interage com o sistema de forma intuitiva e gráfica, representando o comportamento necessário para que ele atinja suas metas ao utilizar o software. Esse modelo consiste em um conjunto de elementos importantes para descrever os “casos de uso”, os “atores” e as relações entre eles.

O diagrama de caso de uso é um recurso visual que organiza subconjuntos do modelo, facilitando a comunicação não apenas entre os desenvolvedores, mas também com stakeholders externos. Ele mantém uma estrutura consistente e de fácil visualização, permitindo que as interações e funcionalidades do sistema sejam rapidamente compreendidas.

Além de proporcionar uma visão geral descritiva, o diagrama é um recurso intuitivo que ilustra as funções e interações com o sistema, ajudando a definir a complexidade do projeto. Dessa forma, ele permite analisar quais interações e funções serão implementadas de maneira mais eficaz e ajuda a estimar o custo do projeto de forma mais precisa. Isso torna mais simples a comunicação com empresas e clientes, que podem, por exemplo, entender um caso de uso para transações bancárias com facilidade.

O diagrama de caso de uso é útil também na resolução de problemas complexos, permitindo que possíveis soluções sejam visualizadas de forma clara. Ele facilita o acompanhamento do desenvolvimento do sistema, identificando quais funções estão funcionando adequadamente e quais precisam de ajustes. Essa análise contínua ajuda a aprimorar o projeto, corrigir eventuais problemas e implementar melhorias de acordo com as necessidades do usuário.

Rupert define os casos de uso como "uma técnica utilizada para capturar os requisitos de um sistema por meio da descrição de suas interações com os usuários e outros sistemas. Cada caso de uso descreve uma sequência de ações que o sistema deve realizar para atingir um objetivo específico do usuário". Ele também enfatiza que os casos de uso ajudam a tornar os requisitos mais compreensíveis para todas as partes envolvidas, servindo como um ponto de partida para o desenvolvimento. (RUPERT, 2016, p. 102).

Índice de casos de uso:

UC001: Cadastrar Usuário

UC002: Autenticar Usuário

UC003: Adotar um Animal

UC004: Ver Animais Para Adoção

UC005: Editar Perfil do Animal

UC006: Doar Animal

UC007: Cadastrar Animal

Figura 5 - Diagrama de Caso de Uso



Fonte: Autoria Própria

Os Quadros 3 ao 9 representam os Casos de Uso feitos para o desenvolvimento do sistema.

Quadro 3 - Use Case - Cadastrar Usuário

Caso de Uso - Cadastrar Usuário	
ID	UC 001
Descrição	Este caso de uso tem o objetivo de validar o cliente no sistema UC001
Ator Primário	Cliente

Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema solicita ao usuário seu nome. 2. O sistema solicita ao usuário seu cpf. 3. O sistema solicita ao usuário seu telefone. 4. O sistema solicita ao usuário seu estado. 5. O sistema solicita ao usuário sua cidade. 6. O sistema solicita ao usuário seu email. 7. O sistema solicita ao usuário sua senha. 8. O sistema busca no banco de dados se o cliente é cadastrado. 9. O sistema valida os dados cadastrados.
Pós-Condição	Encaminha para o informar o serviço.
Cenário Alternativo	a- CPF já cadastrado no sistema.
Inclusão	UC 002
Extensão	NÃO POSSUI

Fonte: Autoria própria.

Quadro 4 - Use Case - Autenticar Usuário

Caso de Uso - Autenticar Usuário	
ID	UC 002
Descrição	Este caso de uso tem o objetivo de validar o cliente no sistema UC002
Ator Primário	Cliente
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema solicita ao usuário seu email. 2. O sistema solicita ao usuário sua senha. 3. O sistema busca no banco de dados se o cliente é cadastrado. 1. O sistema valida os dados cadastrados.
Pós-Condição	Encaminha para a página do usuário.
Cenário Alternativo	b- Credenciais Inválidos.
Inclusão	Não possui
Extensão	Não possui

Fonte: Autoria própria.

Quadro 5 - Use Case - Adotar Animal

Caso de Uso - Adotar Animal	
ID	UC 003
Descrição	Este caso de uso tem como objetivo o usuário navegar pelos perfis dos animais para solicitar a adoção de algum UC003.
Ator Primário	Usuário
Pré-condição	Ser Cadastrado no sistema, estar autenticado.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário pesquisa e filtra os animais disponíveis para adoção. 2. O usuário seleciona um animal para adotar. 3. O usuário acessa o botão de comunicação com o dono do pet escolhido.
Pós-Condicion	Não possui
Cenário Alternativo	Cancelar a adoção
Inclusão	Não Possui
Extensão	Não possui

Fonte: Autoria própria.

Quadro 6 - Use Case - Ver Animais Para Adoção

Caso de Uso - Ver Animais Para Adoção	
ID	UC0004
Descrição	Este caso de uso tem um objetivo de entrar na aba de doações e adoções de animais UC0004.
Ator Primário	Cliente
Pré-condição	NÃO POSSUI
Cenário Principal	<ol style="list-style-type: none"> 1. Na página de adoções e doações o sistema vai mostrar uma página inteira de animais. 2. O usuário pode clicar nos animais para ver suas descrições. 3. O usuário poderá clicar em filtros para achar um animal de sua escolha.
Pós-Condicion	NÃO POSSUI
Cenário Alternativo	c- O cliente pode retornar a página principal.
Inclusão	NÃO POSSUI
Extensão	NÃO POSSUI

Fonte: Autoria própria.

Quadro 7 - Use Case - Editar Perfil do Animal

Caso de Uso - Editar Perfil do Animal
--

ID	UC005
Descrição	Este caso de uso tem um objetivo de editar seu perfil do animal 005
Ator Primário	Cliente
Pré-condição	Animal ser cadastrado no sistema, usuário deve estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. Na aba perfil o sistema mandará opções de editar as suas informações 2. Perguntará se tem certeza de que deseja salvar as informações alteradas
Pós-Condição	NÃO POSSUI
Cenário Alternativo	d- O cliente pode recusar as mudanças feitas no seu perfil
Inclusão	NÃO POSSUI
Extensão	NÃO POSSUI

Fonte: Autoria própria.

Quadro 8 - Use Case - Doar Animal

Caso de Uso - Doar um Animal	
ID	UC0006.
Descrição	Este caso de uso tem como objetivo o usuário doar um animal UC0006.
Ator Primário	Usuário.
Pré-condição	Ter um cadastro no sistema, estar autenticado na aplicação.
Cenário Principal	<ol style="list-style-type: none"> 1. O doador acessa o pet que será doado. 2. Altera seu estado para "adotado". 3. O sistema valida as informações fornecidas. 4. O sistema registra a doação do animal no sistema.
Pós-Condição	Redirecionado para a aba de Pets.
Cenário Alternativo	NÃO POSSUI
Inclusão	NÃO POSSUI
Extensão	NÃO POSSUI

Fonte: Autoria própria.

Quadro 9 - Use Case - Cadastrar Animal

Caso de Uso - Cadastrar Animal	
ID	UC 007

Descrição	Este caso de uso tem o objetivo de Cadastrar um Animal no sistema UC007
Ator Primário	Cliente
Pré-condição	Ter cadastro no sistema, estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema irá solicitar o preenchimento de um formulário de cadastramento do Animal 2. Irá pedir uma foto do animal 3. Irá pedir se deseja realmente salvar o animal no sistema.
Pós-Condção	Salva no seu perfil
Cenário Alternativo	NÃO POSSUI
Inclusão	NÃO POSSUI
Extensão	NÃO POSSUI

Fonte: Autoria própria.

3.6 Protótipos de Tela

Uma das partes mais importantes e cruciais para o desenvolvimento do sistema é ter o esboço de como serão as telas do projeto, permitindo visualizar como elas podem e devem ser. Esse esboço ajuda a avaliar se o layout, as cores, a navegação e a interação dos usuários estão alinhadas com a versão final planejada e os diagramas já estudados. Essa fase não serve apenas para identificar melhorias, mas também para deixar um protótipo praticamente pronto, onde a programação será feita diretamente sobre essa base visual.

A Figura 6 ilustra a tela inicial com um design minimalista e moderno, pensado para não apenas agradar aos olhos dos usuários, mas também permitir que eles possam passar horas navegando sem sentir cansaço visual. Essa tela concentra a maior parte das informações do site de forma organizada, evitando sobrecarregar o usuário. Além disso, foi projetada para incentivar uma navegação intuitiva e fácil, proporcionando uma experiência agradável e funcional.

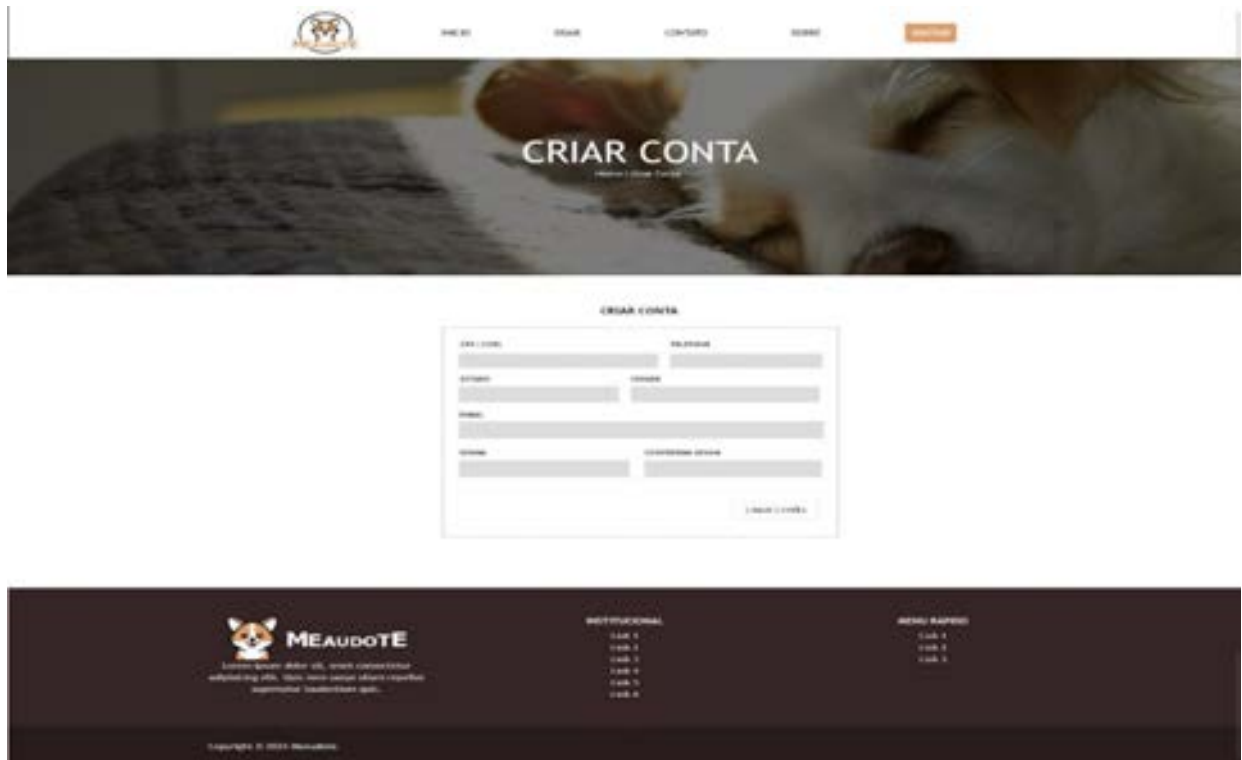
Figura 6 - Tela Inicial



Fonte: Autoria própria.

A Figura 7 se trata da tela de login dos usuários e criação de conta uma das partes mais importantes é a partir dela que você possui seu cadastro e próprio perfil dentro da plataforma.

Figura 7 - Tela de Criação de conta



Fonte: Autoria própria.

A Figura 8 trata de uma parte importante após a criação de sua senha, sendo possível caso o esquecimento de sua senha para o login a possível recuperação e mudança de senha para não ocorrer na perda de sua conta.

Figura 8 - Alteração de senha



Fonte: Autoria própria.

A Figura 9 representa a tela do processo de criação de conta.

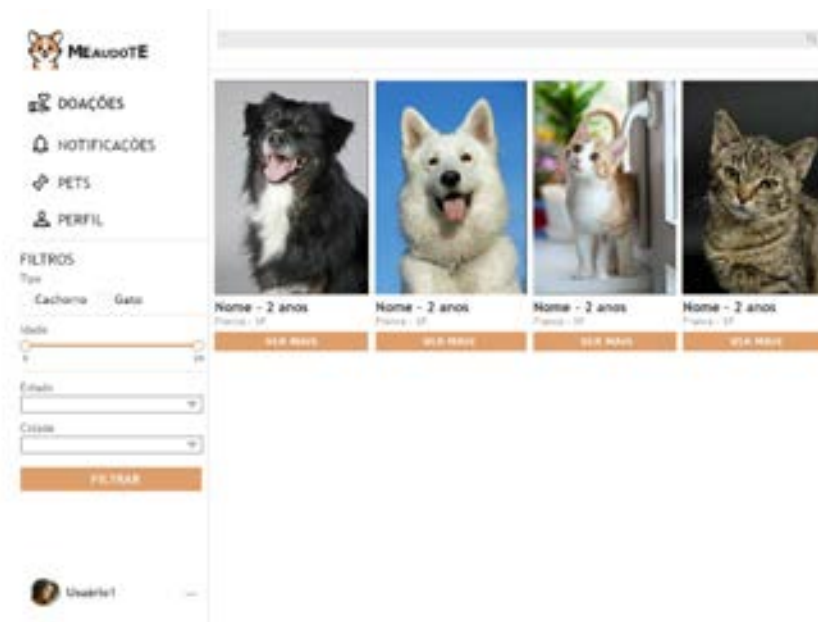
Figura 9 - Tela de criação de conta



Fonte: Autoria própria

A figura 10 representa a tela de catálogo de animais que estão disponíveis para adoções possuindo foto dos animais, informações adicionais sobre os mesmos e como principal seu nome idade e cidade destacados, deixando de uma forma fácil de ser vista e com filtros para ficar mais eficaz a pesquisa do usuário.

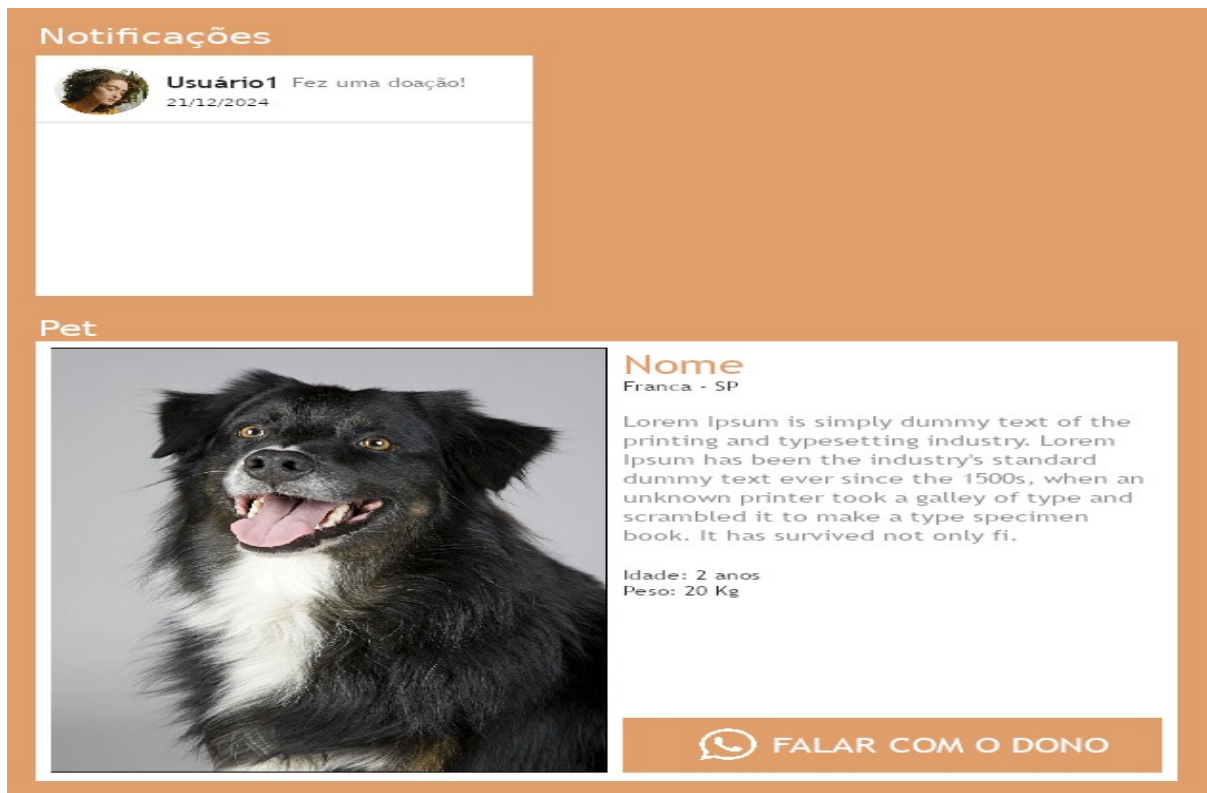
Figura 10 - Catálogo de animais



Fonte: Autoria própria.

A figura 11 se trata da parte após decisão de qual animal gostará de adotar e irá ser encaminhado para o chat com a pessoa responsável pelo animal doado sendo possível perguntar mais informações, possivelmente uma visita para vê-lo e finalizar adotando o animal.

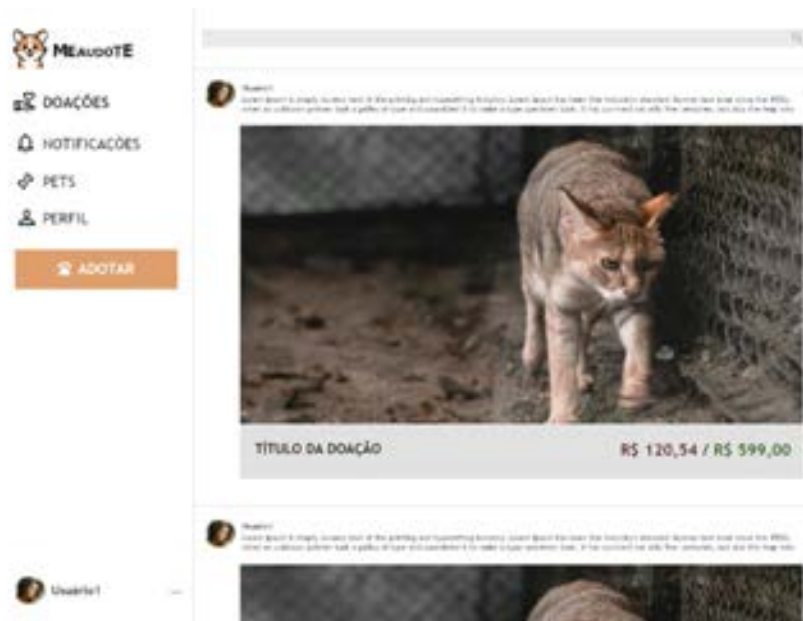
Figura 11 - Tela de chat de adoção



Fonte: Autoria própria.

A figura 12 representa a página de “Vaquinha” do sistema onde constam não só possíveis causas que você pode doar, mas também ocorrências de animais em estado de abandono ou em situação de maus-tratos, sendo como se fosse uma “timeline” de redes sociais, tendo notificações e doações já feitas.

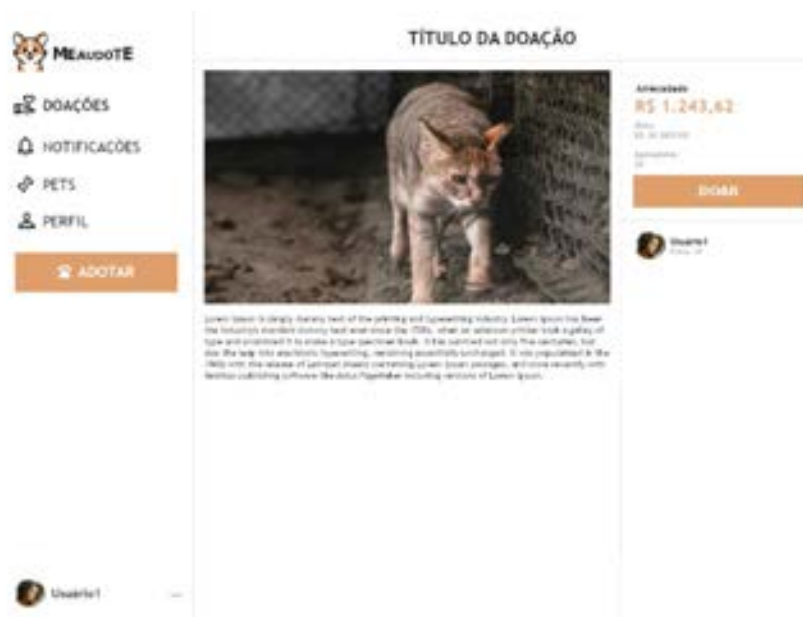
Figura 12 - Tela da página "Vaquinha"



Fonte: Autoria própria.

A figura 13 representa a tela após escolher uma doação a ser feita, tendo uma breve descrição do animal e informações do que está ocorrendo e a necessidade das doações, além do valor necessário para alcançar a meta.

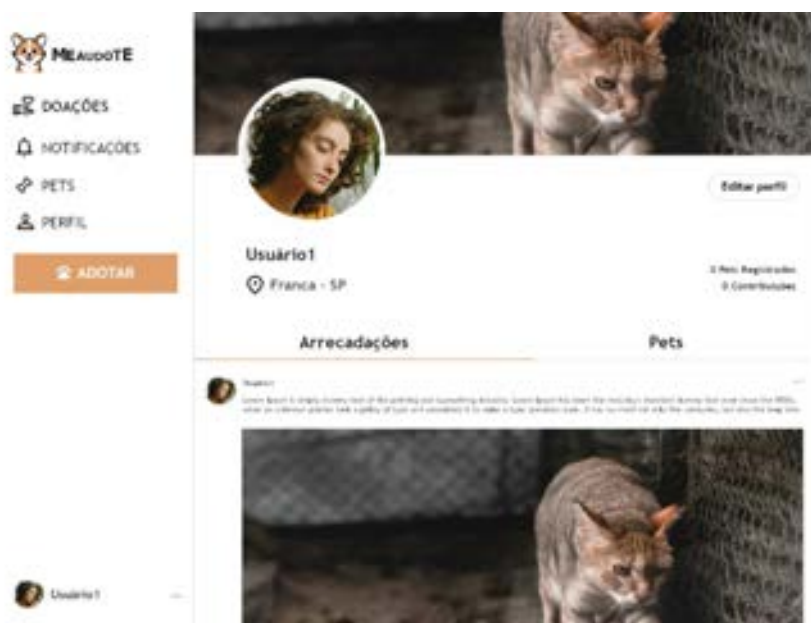
Figura 13 - Aba de Doação



Fonte: Autoria própria.

A figura 14 representa a tela de perfil do usuário podendo ser editada como se fosse em redes sociais, as diferenças são as publicações referentes a ocorrências e doações de animais além de ter animais adotados ou doados, sendo uma rede sociais que trata apenas deste assunto.

Figura 14 - Perfil do usuário



Fonte: Autoria própria.

A figura 15 representa a parte de Blog sendo uma página destinada a pessoas que estão pensando ou em processo de adotar um animal, tendo diversas dicas de como é ter um pet, seus cuidados e pontos a se pensar antes de adotar, foi pensado nesta aba para usuários que ainda tem suas dúvidas na hora de adotar o seu pet, para que avaliem se realmente é possível dar um lar para este animal.

Figura 15 - Tela de Blog



Fonte: Autoria própria.

4 FERRAMENTAS E MÉTODOS

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade com documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil o processo de implementação. A escolha dessas ferramentas também foi influenciada pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas. As licenças das ferramentas são de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais. Os repositórios oficiais para cada um dos artefatos gerados são:

BPMN: Lucidchart – [Diagramação com inteligência](#)

4 DESENVOLVIMENTO

A escolha das tecnologias para o desenvolvimento da aplicação foi orientada pela busca por eficiência e facilidade de manutenção. No front-end, foi escolhido o React com TypeScript, pois oferece uma biblioteca robusta e flexível para construção de interfaces dinâmicas, enquanto o TypeScript adiciona tipagem estática, proporcionando maior segurança e previsibilidade no código. No back-end, o Spring Boot foi utilizado por ser uma solução consolidada para a construção de APIs RESTful, com forte integração ao ecossistema Java e suporte a boas práticas de segurança e desempenho. O banco de dados utilizado foi um PostgreSQL por sua simplicidade e sua integração nativa com Enums, recurso que facilita o processo de desenvolvimento e validação de dados. A documentação da API com Swagger facilita a comunicação entre os desenvolvedores e a validação do uso correto dos endpoints. A infraestrutura foi construída na Google Cloud Platform (GCP), com o backend hospedado no Cloud Run, que oferece escalabilidade automática e gerenciamento simplificado de contêineres. O armazenamento de imagens foi delegado a um bucket, aproveitando a segurança e o baixo custo do Google Cloud Storage. A rede privada garante a segurança e controle de acesso, enquanto o frontend foi hospedado na Vercel pela sua integração nativa com React, pipelines automatizados de deploy, além da alta disponibilidade global, garantindo uma experiência de usuário rápida e eficiente. O uso de Terraform possibilitou a automação e reprodutibilidade da infraestrutura, facilitando a gestão do ambiente e permitindo escalar rapidamente conforme a demanda.

4.1.1 Back-end

No backend foram definidas as variáveis para a criação do usuário root, que terá todas as permissões para acessar a API e o método de inicialização. A api após iniciar, será entregue por um tomcat webservice. Por se tratar de uma aplicação web restful, é uma aplicação que retorna informação através dos controllers.

Figura 17 - Admin Controller

```
@RestController
@RequestMapping("/api/v1/admin")
@CrossOrigin("*")
public class AdminController {
    private final AppUserService appUserService;
    private final AppRoleService appRoleService;
    private final AdoptionAnimalService adoptionAnimalService;
    private final ProfileService profileService;

    @Autowired
    public AdminController(AppUserService appUserService, AppRoleService appRoleService, AdoptionAnimalService adoptionAnimalService, ProfileService profileService) {
    }

    @GetMapping("/users")
    @ResponseStatus(HttpStatus.OK)
    public Response<Page<AppUserDto>> getAll() {
    }

    @PostMapping("/users/{id}")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<AppUserDto> update(@PathVariable("id") String id, @RequestBody AppUserDto appUserDto) throws NotFoundException {
    }

    @GetMapping("/roles")
    @ResponseStatus(HttpStatus.OK)
    public Response<List<AdminAppRoleDto>> getAll() {
    }

    @PostMapping("/roles")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<AppRoleDto> saveRole(@RequestBody AdminAppRole appRole) throws FieldInvalidException {
    }

    @PostMapping("/roles/{name}")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<AppRoleDto> saveRole(@PathVariable("name") String name, @RequestBody AdminAppRole appRole) throws FieldInvalidException, NotFoundException {
    }
}
```

Fonte: Autoria própria.

Figura 18 - Adoption Animal Controller

```
public class AdoptionAnimalController {
    private final AdoptionAnimalService adoptionAnimalService;
    private final AppUserService appUserService;

    @Autowired
    public AdoptionAnimalController(AdoptionAnimalService adoptionAnimalService, AppUserService appUserService) {
        this.adoptionAnimalService = adoptionAnimalService;
        this.appUserService = appUserService;
    }

    @GetMapping("/{id}")
    @ResponseStatus(HttpStatus.OK)
    public Response<AdoptionAnimalDto> findById(@PathVariable("id") UUID id) {
    }

    @PostMapping("/{id}/users")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<AppUserDto> findById(@PathVariable("id") UUID id) throws NotFoundException {
    }

    @GetMapping("/{id}")
    @ResponseStatus(HttpStatus.OK)
    public Response<Page<AdoptionAnimalDto>> findById(@PathVariable("id") UUID id) {
    }

    @PostMapping
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<Page<AdoptionAnimalDto>> findById(@RequestBody AdoptionAnimal adoptionAnimal) throws AppUserNotFoundException, NotFoundException {
    }

    @PostMapping("/{id}")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<AdoptionAnimalDto> update(@PathVariable("id") UUID id, @RequestBody AdoptionAnimal adoptionAnimal) throws AppUserNotFoundException, NotFoundException {
    }

    @DeleteMapping("/{id}")
    @ResponseStatus(HttpStatus.ACCEPTED)
    public Response<String> delete(@PathVariable("id") UUID id) throws AppUserNotFoundException, NotFoundException {
    }
}
```

Fonte: Autoria própria.

Figura 19 - App User Controller

```

@RestController
@RequestMapping("/api/v1/users")
@CrossOrigin("")
public class AppUserController {

    private final AppUserService appUserService;
    private final AdaptionUserService adaptionUserService;
    private final PkakeyService pkakeyService;

    @Autowired
    public AppUserController(AppUserService appUserService, AdaptionUserService adaptionUserService, PkakeyService pkakeyService) {
    }

    @GetMapping("/current")
    public ResponseEntity<Response<AppUserDto>> getCurrent(@Principal @RequestParam String loginId) throws AppUserNotLoggedInException, NotFoundException {
    }

    @GetMapping("/{username}")
    public ResponseEntity<Response<AppUserDto>> getUser(@PathVariable("username") String username) throws NotFoundException {
    }

    @GetMapping
    public ResponseEntity<Response<Page<AppUserDto>>> getAll() {
    }

    @GetMapping("/current/pets")
    public ResponseEntity<Response<Page<AdaptionServiceDto>>> getAdaptationService() {
    }

    @GetMapping("/{username}/pets")
    public ResponseEntity<Response<Page<AdaptionServiceDto>>> getAdaptationServiceByUser() {
    }

    @GetMapping("/current/pkakeys")
    public ResponseEntity<Response<List<PkakeyDto>>> getPkakeys(@Principal @RequestParam String loginId) throws NotFoundException, AppUserNotLoggedInException {
    }

    @GetMapping("/{username}/pkakeys")
    public ResponseEntity<Response<List<PkakeyDto>>> getPkakeysUser(@PathVariable("username") String username) throws NotFoundException, AppUserNotLoggedInException {
    }

    @PostMapping("/current")
    public ResponseEntity<Response<AppUserDto>> updateCurrent(@Principal @RequestParam String loginId, @RequestBody AppUserAppUserDto appUserDto) throws AppUserNotLoggedInException, NotFoundException {
    }
}

```

Fonte: Autoria própria.

Figura 20 - Authentication Controller

```

@RestController
@RequestMapping("/api/v1/auth")
@CrossOrigin("")
public class AuthController {

    private final AuthenticationService authenticationService;
    private final AppUserService appUserService;

    @Autowired
    public AuthController(AuthenticationService authenticationService, AppUserService appUserService) {
        this.authenticationService = authenticationService;
        this.appUserService = appUserService;
    }

    @PostMapping("/login")
    public ResponseEntity<Response<String>> login(@RequestBody LoginDto request) throws NotFoundException {
        Response<String> response = new Response();
        JwtTokenDto authentication = authenticationService.authenticate(request);
        response.setData(authentication.token());
        return ResponseEntity.ok(response);
    }

    @PostMapping("/register")
    public ResponseEntity<Response<String>> register(@Valid @RequestBody RegisterDto request) throws FieldInUseException, NotFoundException {
        Response<String> response = new Response<>();
        response.setData(authenticationService.register(request).token());
        return ResponseEntity.ok(response);
    }
}

```

Fonte: Autoria própria.

Figura 21 - Exception Controller

```

@RestControllerAdvice
public class ExceptionControllerAdvice {
    @ExceptionHandler
    public ResponseEntity<Response<Object>> handleExceptions(final Exception exception) {
        final Response<Object> response = new Response<>();
        response.getErrors().add(exception.getMessage());
        return new ResponseEntity<>(response, HttpStatus.FORBIDDEN);
    }
    @ExceptionHandler({NotFoundException.class})
    public ResponseEntity<Response<Object>> handleNotFoundExceptions(final Exception exception) {
        final Response<Object> response = new Response<>();
        response.getErrors().add(exception.getMessage());
        return new ResponseEntity<>(response, HttpStatus.NOT_FOUND);
    }
    @ExceptionHandler({MethodArgumentNotValidException.class})
    public ResponseEntity<Response<Object>> handleMethodArgumentNotValid(final MethodArgumentNotValidException exception) {
        final Response<Object> response = new Response<>();
        exception.getFieldErrors().forEach(i -> response.getErrors().add(i.getDefaultMessage()));
        return new ResponseEntity<>(response, HttpStatus.BAD_REQUEST);
    }
}

```

Fonte: Autoria própria.

Figura 22 – Image Controller

```

@RestController
@RequestMapping("/api/v3")
@CrossOrigin("")
public class ImageController {
    private final StorageService storageService;
    private final AppUserService appUserService;

    @Autowired
    public ImageController(StorageService storageService, AppUserService appUserService) {
        this.storageService = storageService;
        this.appUserService = appUserService;
    }

    @GetMapping("/pets/{id}/image")
    public byte[] downloadPetImage( @PathVariable("id") UUID id ) throws IOException {
        return storageService.download(id, "pets");
    }

    @PostMapping(
        path = "/pets/{id}/image",
        consumes = MediaType.MULTIPART_FORM_DATA_VALUE,
        produces = MediaType.APPLICATION_JSON_VALUE
    )
    public ResponseEntity<Response<String>> uploadImage(
        @RequestParam("file") MultipartFile file,
        @PathVariable UUID id
    ) throws IOException, ImageOverSizedException {
        var response = new Response<String>();
        storageService.save(id, "pets", file);
        response.setBody("file saved.");
        return ResponseEntity.ok().body(response);
    }

    @GetMapping("/users/{name}/image")
    public byte[] downloadAppUserProfilePicture( @PathVariable("name") String name ) throws IOException, NotFoundException {
        AppUser appUser = appUserService.findByIdByName(name);
        return storageService.download("users/profile-picture/"+name+"/"+appUser.getProfilePicture());
    }

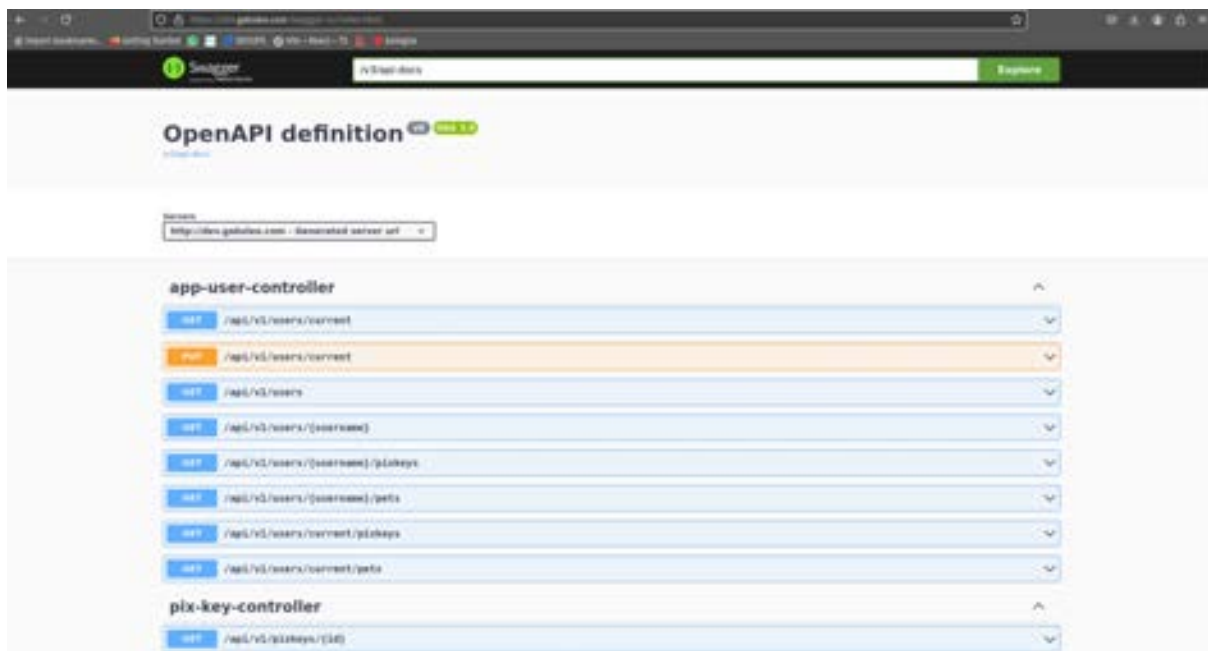
    @PostMapping(
        path = "/users/{name}/image",
        consumes = MediaType.MULTIPART_FORM_DATA_VALUE,
        produces = MediaType.APPLICATION_JSON_VALUE
    )
    public ResponseEntity<Response<String>> uploadProfileImage(
        @RequestParam("file") MultipartFile file,
        @PathVariable("name") String name
    ) throws IOException, NotFoundException, ImageOverSizedException {
        var response = new Response<String>();
        storageService.saveAppUserProfilePicture(name, "users/profile-picture/"+name, file);
        response.setBody("file saved.");
        return ResponseEntity.ok().body(response);
    }
}

```

Fonte: Autoria própria

Para testes de rota, a aplicação conta com a implementação do Swagger, biblioteca de documentação e teste das rotas.

Figura 23 - Swagger



Fonte: Autoria própria.

4.1.2 Front-end

Para o frontend foram criadas as páginas por componentes, garantindo que possam ser reutilizadas quando necessário. Para a entrega e rota de páginas, foi utilizado o react router, que fará o redirecionamento.

Figura 24 - FrontEnd

```
1 import { Home } from './pages/home'
2 import './main.css'
3 import { BrowserRouter, Route, Routes } from 'react-router-dom';
4 import { Pets } from './pages/pets';
5 import { AppUserProvider } from './providers/AppUserProvider';
6 import { Profile } from './pages/profile';
7 import { Login } from './pages/login';
8 import { Register } from './pages/register';
9
10 export const App = () => {
11   return (
12     <div className="App">
13       <AppUserProvider>
14         <BrowserRouter>
15           <Routes>
16             <Route path="/" element={Home} /> />
17             <Route path="/donations" element={Pets} /> />
18             <Route path="/notifications" element={Pets} /> />
19             <Route path="/pets" element={Pets} /> />
20             <Route path="/login" element={Login} /> />
21             <Route path="/register" element={Register} /> />
22             <Route path="/profile" element={Profile} /> />
23             <Route path="/profile/username" element={Profile} /> />
24           </Routes>
25         </BrowserRouter>
26       </AppUserProvider>
27     </div>
28   )
29 }
```

Fonte: Autoria própria.

4.1.3 Infraestrutura

A página principal do terraform foi feita através de módulos, para que seja fácil e rápido para a implementação de novos ambientes. O terraform irá criar dentro da GCP um cloud run para a entrega do backend, uma VPC para garantir segurança de rede e comunicação entre o banco e o backend, um banco de dados para armazenar informações da API, um bucket para guardar imagens e um refistry para guardar imagens do docker.

Figura 25 - Infraestrutura

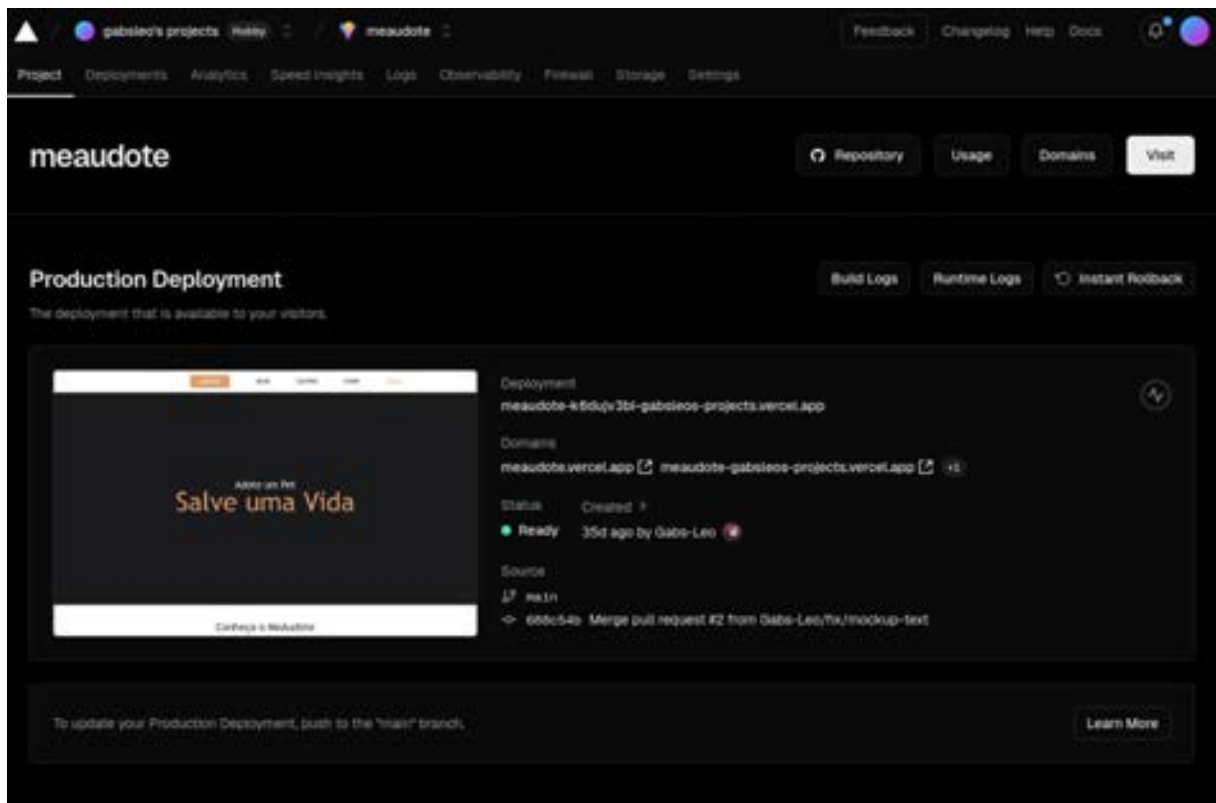
```
1  module "vpc" {
2    source = "../modules/vpc"
3    subnet_public_cidrs = var.subnet_public_cidrs
4    subnet_private_cidrs = var.subnet_private_cidrs
5    project = var.project
6    environment = terraform.workspace
7    region = var.region
8  }
9
10 module "cloud_run" {
11   source = "../modules/cloud_run"
12   depends_on = [ module.vpc ]
13   region = var.region
14   project = var.project
15   environment = terraform.workspace
16   subnet_name = module.vpc.private_subnet_1_name
17   registry_name = module.artifact_registry.registry_id
18   env_variables = var.env_variables
19   database_host = module.sql_database.db_host
20 }
21
22 module "sql_database" {
23   source = "../modules/sql_database"
24   depends_on = [ module.vpc ]
25
26   project = var.project
27   environment = terraform.workspace
28   region = var.region
29   vpc_id = module.vpc.vpc_id
30   db_user = var.db_user
31   db_password = var.db_password
32 }
33
34 module "artifact_registry" {
35   source = "../modules/artifact_registry"
36   region = var.region
37   project = var.project
38   environment = terraform.workspace
39 }
40
```

Fonte: Autoria própria

4.1.4 Vercel

O deploy do front end é feito através da plataforma da vercel, que realiza automaticamente o build da aplicação e prepara a entrega contínua para facilitar entregas em release e snapshot.

Figura 26 - Vercel



Fonte: Autoria própria.

4.1.5 Github Actions

O deploy do back end é feito através da plataforma do github actions, que realiza automaticamente o build da imagem docker da aplicação, publica no registry da Google Cloud Platform e prepara a entrega contínua para facilitar entregas em release e snapshot.

Figura 27 - Github Actions

```
name: build and deploy to Cloud Run

on:
  push:
    branches: [ "main" ]

jobs:
  build:
    name: Build and Deploy to Cloud Run
    runs-on: ubuntu-latest
    permissions:
      contents: read
      id-token: write
    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Setup Node
        uses: actions/setup-node@v3
        with:
          node-version: '16.x'

      - name: Install dependencies
        run: npm install

      - name: Build
        run: npm run build

      - name: Deploy to Cloud Run
        uses: google-github-actions/deploy-cloudrun@v1
        with:
          service: my-service
          image: gcr.io/my-project/my-image
          registry: gcr.io
          region: us-central1
          env_vars: |
            PROJECT_ID=my-project-id
            SERVICE_NAME=my-service-name
            REGION=us-central1
            IMAGE_NAME=gcr.io/my-project/my-image

  test:
    name: Test
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v3

      - name: Setup Node
        uses: actions/setup-node@v3
        with:
          node-version: '16.x'

      - name: Install dependencies
        run: npm install

      - name: Run tests
        run: npm run test
```

Fonte: Autoria própria.

5 RESULTADOS E DISCUSSÃO

5.1 Análise de Portabilidade

Infraestrutura: Computação em Nuvem – Armazenamento; Conta no Github; Container Registry; Computação em Nuvem – Hospedagem; Conta na Vercel; Instância do Cloud Run; Instância Banco de Dados – PostgreSQL 1CPU, 10GB SSD, 500GB Transferência; Licenças; Vercel; Sistema Operacional da imagem base do Cloud Run.

OBS: A quantidade de usuários pode variar, em caso de crescimento do app é recomendada a implementação de um serviço de escalonamento automático.

Recursos Humanos: Por volta de 80 usuários simultâneos; Usuários de diferentes locais do Brasil terão acesso ao app.

Requisitos de Usuários: Hardware – Computadores Pessoais (quantidade e descrição); um aparelho com acesso à internet (computador, celular, tablet) para cada usuário simultâneo; Hardware – Periféricos (quantidade e descrição); Mouse e Teclado para cada computador.

5.2 Cálculos das Métricas

A figura 27 e 28 representam as métricas do sistema, feitas por uma aplicação desenvolvida por um aluno da Fatec.

Figura 28 - Métricas do sistema

Métricas

Tabelas

Tabela A =

Tabela B =

Tabela C =

Tabela D =

Tabela E =

Tabela F =

38

	Simplex Médio Complexo			Peso		
	Simplex	Médio	Complexo	Simplex	Médio	Complexo
Entrada	2	0	3	6	0	18
Saída	2	4	0	8	20	0
Consulta	2	1	3	6	4	18
Arquivo	5	0	0	35	0	0
Interface	6	0	0	30	0	0

Tipo de Linguagem

Cobol

Pascal

Java / Delphi / Visual Basic

Linguagem Orientada a Obj. (C++)

Geradores de Código (SQL + HTML)

Valor da linguagem = 20

Valor do kLoc = 3920.00

Fonte: Autoria própria.

Figura 29 - Continuação das métricas do sistema

Tipo de Sistema

Sistema Web
 Sistema Comercial
 Sistema Eletrônico

Enviar

Valor do sistema = 3300

Gerar Indicador

Valor do Indicador = 1.18

Valor da hora: Enviar

Valor da hora = 20

Valor do FPB = 145

Valor do FPR = 196

Dias Horas Minutos

25 5 45

Calcular Prazo

Calcular Total

3115.20

Fonte: Autoria própria.

5.3 Proposta Comercial

5.3.1 Introdução

O sistema adoção de animais visa auxiliar no processo de adoção de animais resgatados, bem como prestar auxílio com o acompanhamento de tratamento de injúrias.

Com a implantação do sistema, será possível auxiliar no processo de adoção de animais, no processo de arrecadação de fundos para financiar o tratamento de animais injuriados.

Através da interface é possível criar contas como um usuário, que poderão registrar animais para adoção ou solicitar páginas de arrecadação de fundos, que serão aprovadas por um administrador antes de irem ao ar. O layout e coloração das páginas foi pensado para que o usuário visualize com mais clareza seus objetivos.

Essa solução de tecnologia digital deve facilitar os processos que se sujeita a prestar apoio e reduzir o número de abandonos.

5.3.2 Solução Proposta

A plataforma “Meaudote” se compromete a proporcionar um meio facilitador para trabalhar com adoção e arrecadação de fundos para tratamentos de animais necessitados. A plataforma visa trazer uma forma de usuários poderem registrar livremente pets resgatados que não possuem um dono, compartilhar o link para que possam espalhar por suas redes sociais.

Além de pessoas que resgataram animais, a plataforma visa auxiliar ONGs, podendo registrar suas chaves pix para receberem doações ou mesmo seus animais para facilitar o processo de doação.

Com o uso de uma entrevista realizada com ONGs da região de Franca (São Paulo), o problema com abandonos recorrentes de pets foi exposto, e a partir dele o tema foi definido. Visando prestar apoio à ONGs e pessoas que resgataram animais, mas que não podem cuidar, foi decidido desenvolver uma solução sistêmica utilizando-se de tecnologias digitais de informação e comunicação.

É de conhecimento geral que as ferramentas tecnológicas disponíveis atualmente ajudam na otimização e organização de processos; sendo assim, implantá-las como recurso de auxílio evidencia suma importância. Essa solução de tecnologia digital deve simplificar os processos que se sujeita a prestar apoio, gerar ganho de tempo, explicitar problemas, sucessos e facilitar a identificação de oportunidades.

5.3.3 Visão geral da solução

Tendo como base as informações coletadas durante o processo de reconhecimento de campo, foi decidido trabalhar em cima do problema relatado que se relaciona com o número e frequência de abandonos, foi arquitetado uma solução sistêmica web que se propõe a proporcionar um ambiente interativo capaz de armazenar e gerir as demandas de forma simplificada em todos os dispositivos com acesso a um navegador.

O sistema possibilitará não só o armazenamento dos dados, mas também a gestão do processo proposto a ser solucionado. Informações como chaves pix para doação, animais para adoção, local de busca por animais, dentre outras.

Por fins de conveniência, é possível aplicar filtros para uma pessoa procurando por um animalzinho, poder escolher mais facilmente o tipo de animal que deseja, a idade e sua localização.

5.3.4 Escopo da solução

Por meio da Estrutura Analítica de Projeto (EAP), ferramenta para gerência de projetos, determinou-se as 5 etapas do desenvolvimento da documentação da solução sistêmica proposta, as quais são: iniciação, planejamento, execução, controle e encerramento. A iniciação consiste em selecionar o cliente e identificar as fraquezas em seu negócio, assim como refiná-las e determinar aquelas cabíveis de serem solucionadas a partir do uso das tecnologias de informação e comunicação.

As fases denominadas como planejamento, execução e controle guiam o desenvolvimento robusto do projeto e auxiliam na avaliação do que pode ou não se tornar sistema, encaminhando-se assim para a etapa final: o encerramento. O encerramento é o arremate final de tudo o que foi documentado e planejado; sendo assim, possível delimitar os custos totais do projeto.

5.3.5 Prazos

De acordo com as métricas do projeto, fica determinado que é possível desenvolver o sistema proposto em cerca de 180 horas.

5.3.6 Investimento

Foi instituído, a partir da realização das métricas, o custo total da licença do sistema proposto. De forma a não incluir o acesso ao código fonte, o suporte e/ou possíveis futuras manutenções, modificações, é possível desenvolver a solução sistêmica por 4 parcelas de R\$900,00 BRL (Real brasileiro), totalizando R\$3600,00 BRL.

O custo fixo mensal para utilização do sistema se dá em R\$100,00 BRL para manutenção de infraestrutura e domínio. Chamados extras denotam em valor adicional de R\$40,00 BRL.

6 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento da plataforma, foi notado que a questão do abandono animal carece de visibilidade e que os softwares nem sempre precisam ser

voltados para o uso comercial, mas também servir a causas, com pequenas mudanças é possível ter grandes impactos. Nas entrevistas com duas pessoas responsáveis por ONGs, foi definido o que seria feito e como seria a experiência dos usuários ao visualizar a plataforma. A questão principal foi: como o software poderia realmente ajudar a causa animal? Análises e pesquisas foram feitas não apenas em relação à estrutura da plataforma, mas também alternativas para a implementação, levando à criação de um sistema totalmente gratuito para oferecer apoio.

Considerando o aumento do abandono animal em Franca-SP, e a insuficiência de ONGs para suprir essa demanda, o objetivo foi reduzir os problemas de abandono e maus-tratos, promovendo o cuidado animal na cidade. Foi desenvolvido um sistema focado na adoção e doação de animais que funciona como uma rede social de doações.

Inicialmente, foi identificada uma falta de integração entre as iniciativas de adoção de animais, um dos maiores desafios para tornar o sistema intuitivo tanto para quem deseja doar quanto para quem busca adotar. Com a análise de diversos sites, foi decidido que o layout simples que lembra o de uma rede social como o “Twitter” se destaca pela facilidade de leitura e interação. Esse estilo foi adaptado para a aplicação, onde são registrados novos pets e um usuário pode fazer customizações em seu perfil.

Para a adoção de animais, foram incluídos filtros de pesquisa, permitindo que os usuários busquem pets não só por peso, mas também por raça e outras características, além de informações detalhadas sobre o histórico do animal, idade, entre outros critérios. Essa funcionalidade visa facilitar a busca e a conexão entre adotantes e animais.

Como futuras implementações, o plano é criar parcerias com casas de ração, possibilitando a compra de ração diretamente na plataforma, facilitando para aqueles que adotam um animal. Incluir perfis específicos para ONGs e seus responsáveis também é um objetivo do desenvolvimento, ampliando a visibilidade e facilitando a conexão. Foi pensada também uma ferramenta para arrecadação de fundos para animais necessitados, permitindo que além de adotar, a aplicação seja capaz também de ajudar aqueles que já possuem um lar, mas necessitam de cuidados especiais. A adição dessa função permitiria a inclusão de veterinários parceiros no site, que

poderiam ajudar no diagnóstico, recomendação de remédios e na validação da necessidade, auxiliando no processo e evitando fraudes.

REFERÊNCIAS

1. ARTIA. Como fazer EAP na gestão de projetos. Artia Blog. Disponível em: <https://artia.com/blog/como-fazer-eap-na-gestao-de-projetos/>. Acesso em: 7 out. 2024.
2. ATÉ O MOMENTO. O que é caso de uso? s.d. Disponível em: <https://www.ateomomento.com.br/o-que-e-caso-de-uso/>. Acesso em: 15 set. 2024.
3. CAMPOS, André Luiz N. *Modelagem de processos com BPMN*. 2. ed. Rio de Janeiro: Brasport, 2014.
4. DEVMEDIA. Contagem de Pontos de Função. DevMedia. Disponível em: <https://www.devmedia.com.br/contagem-de-pontos-de-funcao/34390>. Acesso em: 22 set. 2024.
5. GUIA DEV. Technical Documentation. s.d. Disponível em: <https://guia.dev/pt/pillars/software-architecture/technical-documentation.html>. Acesso em: 25 ago. 2024.
6. MIRO. Miro: O que é BPMN? s.d. Disponível em: <https://miro.com/pt/diagrama/o-que-e-bpmn/>. Acesso em: 10 out. 2024.
7. OSTERWALDER, Alexander; PIGNEUR, Yves. *Business model generation: a handbook for visionaries, game changers, and challengers*. Hoboken: Wiley, 2011.
8. PRESSMAN, Roger S. *Engenharia de software: uma abordagem profissional*. 7. ed. São Paulo: McGraw-Hill, 2010.
9. QUERO BOLSA. Requisitos Funcionais e Não Funcionais: Entenda a Diferença. s.d. Disponível em: <https://querobolsa.com.br/revista/requisitos-funcionais-e-nao-funcionais>. Acesso em: 22 out. 2024.
10. RUPERT, Steven. *Introduction to use cases: understanding and modeling system requirements*. Boston: Addison-Wesley, 2016.
11. SITEWARE. Modelo Canvas: O que é, como fazer e exemplos. s.d. Disponível em: <https://www.siteware.com.br/blog/metodologias/modelo-canvas/>. Acesso em: 29 set. 2024.
12. TOTVS. BPMN: O que é e como funciona a notação mais usada na indústria. s.d. Disponível em: <https://www.totvs.com/blog/gestao-industrial/bpmn/>. Acesso em: 3 nov. 2024.

13. Universidade Estadual do Ceará. *Fundamentos de Engenharia de Software*. Disponível em: <https://www.uece.br/cct/wp-content/uploads/sites/28/2021/07/Fundamentos-de-Engenharia-de-Software.pdf>. Acesso em: 2 nov. 2024.

14. VISURE SOLUTIONS. Levantamento de Requisitos: o que é e por que é importante? s.d. Disponível em: <https://visuresolutions.com/pt/blog/requirements-gathering/>. Acesso em: 8 set. 2024.