

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**ARTHUR DALLA TORRE BOUÇAS
IRIS OLIVEIRA SANTOS
MATEUS DOBBINS SANCHES**

**PetMatch:
Aplicação mobile para adoção de animais**

**Santos, SP
2023**

**ARTHUR DALLA TORRE BOUÇAS
IRIS OLIVEIRA SANTOS
MATEUS DOBBINS SANCHES**

**PetMatch:
Aplicação mobile para adoção de animais**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Davi Silvestre Moreira dos Reis

**Santos, SP
2023**

RESUMO

Cresce o número de animais abandonados no Brasil, estudos apontam que a taxa de abandono obteve um aumento de 70% no ano de 2020. Diante destes fatos, esse trabalho possui como objetivo projetar um aplicativo para adoção de animais, a fim de auxiliar na busca por interesse no amparo desses bichos e conectar abrigos e protetores independentes com indivíduos interessados. O aplicativo foi criado baseando-se em artigos acerca do aumento do abandono animal durante a pandemia, aliado a pesquisas de trabalho de *ONG's* desse nicho. Para entender melhor as necessidades do projeto, foi feita uma análise de aplicativos de adoção já existentes no mercado e de suas possíveis falhas. Com base nessas informações, foram levantadas as principais funcionalidades da aplicação e, em seguida, ela foi desenvolvida. Os resultados obtidos validam que o *app* atenderia os principais requisitos para o acolhimento de *pets*, ajudando tanto instituições quanto indivíduos comuns a atingirem êxito nessa causa.

Palavras-chaves: Animais. Acolhimento. Amparo.

ABSTRACT

The number of abandoned animals in Brazil is growing, and studies show that the abandonment rate increased by 70% in 2020. Given these facts, the aim of this work is to design an application for animal adoption, in order to help find people interested in caring for these animals and connect shelters and independent protectors with interested individuals. The app was created based on articles about the increase in animal abandonment during the pandemic, together with research into the work of NGOs in this niche. To better understand the needs of the project, an analysis was made of adoption apps already on the market and their possible flaws. Based on this information, the app's main functionalities were identified and it was then developed. The results obtained validate that the app would meet the main requirements for fostering pets, helping both institutions and ordinary individuals to succeed in this cause.

Keywords: Animals. Shelter. Support.

LISTA DE ABREVIATURAS E SIGLAS

ONGS – ORGANIZAÇÃO DA SOCIEDADE CIVIL.....	8
IOS – IPHONE OPERATING SYSTEM	11
CCZS – CENTRO DE CONTROLE DE ZOOSES.....	12
CNPJ – CADASTRO NACIONAL DE PESSOAS JURÍDICAS.....	13
JIT– JUST IN TIME	15
AOT – AHEAD OF TIME	15
HTTP – HYPERTEXT TRANSFER PROTOCOL	16
UML – UNIFIED MODELING LANGUAGE	17
RF– REQUISITOS FUNCIONAIS	17
RNF – REQUISITOS NÃO FUNCIONAIS.....	19
WAMMI – WEBAPLICATIVO ANALYSIS AND MEASUREMENT INVENTORY	34

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 01 – NA.MOSCA (LOGO)	11
ILUSTRAÇÃO 02 – MEAUDOTE (LOGO)	12
ILUSTRAÇÃO 03 – HYPPET (LOGO)	12
ILUSTRAÇÃO 04 – LOGO DA LINGUAGEM DART	14
ILUSTRAÇÃO 05 – LOGO DA FRAMEWORK FLUTTER	15
ILUSTRAÇÃO 06 – LOGO DA LINGUAGEM NODE.JS	15
ILUSTRAÇÃO 07 – REQUISITOS FUNCIONAIS DO SISTEMA	17
ILUSTRAÇÃO 08 – REQUISITOS NÃO FUNCIONAIS DO SISTEMA	19
ILUSTRAÇÃO 09 – CASO DE USO 01 - CADASTRO DO USUARIO	20
ILUSTRAÇÃO 10 – CASO DE USO 02 - LOGIN DO USUARIO	21
ILUSTRAÇÃO 11 – CASO DE USO 03 - CADASTRAR PET	21
ILUSTRAÇÃO 12 – CASO DE USO 04 - VISUALIZAR PET	22
ILUSTRAÇÃO 13 – CASO DE USO 05 - ADOTAR PET.....	23
ILUSTRAÇÃO 14 – CASO DE USO 06 - GERENCIAR PET	24
ILUSTRAÇÃO 15 – CASO DE USO 07 - AJUDAR UMA INSTITUIÇÃO	25
ILUSTRAÇÃO 16 – DIAGRAMA BANCO DE DADOS.....	26
ILUSTRAÇÃO 17 – CÓDIGO IMPLEMENTAÇÃO APPLYFILTERS	27
ILUSTRAÇÃO 18 – CÓDIGO DO HOMEPAGESCREEN	27
ILUSTRAÇÃO 19 – CÓDIGO DO ADOPTIONCONTROLLER	28
ILUSTRAÇÃO 20 – CÓDIGO DE AÇÃO DO MECANISMO PIX.....	28
ILUSTRAÇÃO 21 – CÓDIGO DO MÉTODO REGISTERPET	28
ILUSTRAÇÃO 22 – TELA DE LOGIN/CADASTRO	29
ILUSTRAÇÃO 23 – TELA INICIAL DO APLICATIVO	30
ILUSTRAÇÃO 24 – TELA DE FILTRO DE ADOÇÃO	30
ILUSTRAÇÃO 25 – TELA DE ADOÇÃO	31
ILUSTRAÇÃO 26 – TELA DE CADASTRO DO PET	31
ILUSTRAÇÃO 27 – MODAL QUERO ADOTAR	32
ILUSTRAÇÃO 28 – GRÁFICO DE RESPOSTAS DA PERGUNTA 1	34
ILUSTRAÇÃO 29 – GRÁFICO DE RESPOSTAS DA PERGUNTA 2.....	34
ILUSTRAÇÃO 30 – GRÁFICO DE RESPOSTAS DA PERGUNTA 3.....	35
ILUSTRAÇÃO 31 – GRÁFICO DE RESPOSTAS DA PERGUNTA 4.....	35
ILUSTRAÇÃO 32 – GRÁFICO DE RESPOSTAS DA PERGUNTA 5.....	36

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVO	9
1.1.1 OBJETIVO GERAL	10
1.1.2 OBJETIVOS ESPECÍFICOS.....	10
1.2 ESTADO DA ARTE.....	10
2 DESENVOLVIMENTO.....	14
2.1 ANÁLISE DO SISTEMA.....	14
2.1.1 ANÁLISE DE REQUISITOS.....	17
2.1.2 DIAGRAMA DE CASO DE USO	19
2.1.3 FLUXO DE EVENTOS.....	25
2.2 BANCO DE DADOS	26
2.3 CAMADA DE NEGÓCIO.....	26
2.4 CAMADA DE APRESENTAÇÃO	29
3 RESULTADO.....	33
3.1 RESULTADOS DOS TESTES	33
3.2 RESULTADOS DOS QUESTIONÁRIOS	33
3.3 CONCLUSÃO.....	36
REFERÊNCIAS BIBLIOGRÁFICAS	38
APENDICE A.....	41
APENDICE B.....	42

1 INTRODUÇÃO

Atualmente, o índice de animais abandonados vem crescendo significativamente. Em um levantamento feito pela AMPARA (2021), contando com 530 instituições e protetores independentes de animais revelou que, durante a pandemia, a taxa de abandono de animais obteve um aumento de 70% no ano de 2020.

Segundo o último estudo realizado em 2010, os principais motivos de abandono de cachorros e gatos foram: ninhadas inesperadas (14%), mudança de casa (13,7%), fatores econômicos (13,2%), perda de interesse pelo animal (11,2%) e comportamento problemático do animal de estimação (11%). Entre os motivos menos frequentes, temos: fim da temporada de caça (10,2%), alergia de algum membro da família (7,7%), nascimento de um filho (6,4%), internamento ou morte do proprietário (3,5%), férias (2,6%) ou o medo de pegar toxoplasmose durante a gravidez (2,4%) (MY AFFINITY 2010).

De acordo com uma pesquisa realizada pelo Instituto Pet Brasil (2019), o número de animais em condições de vulnerabilidade atualmente no Brasil ultrapassa 4 milhões, um número alarmante quando comparado ao de adotados.

Diante disso, os números anteriormente citados significam uma grande preocupação tanto para a saúde e bem-estar dos animais quanto para a saúde pública.

Em contrapartida, temos o importante trabalho realizado por ONGS e protetores independentes que se dedicam no trabalho do resgate, tratamento e encaminhamento responsável de animais para adoção tendo como objetivo reverter o cenário atual em que vivemos.

As organizações e protetores independentes, a fim de obterem êxito em conseguirem lares para esses animais, utilizam de estratégias como realização de feiras de adoção, divulgação em sites, anúncios em redes e mídias sociais. Porém, eles deparam-se com diversas dificuldades durante as etapas do processo de adoção dos pets, como sites de adoção com design antigo e pouco intuitivos, além da grande maioria possuir apenas sua versão web.

Ao refletir sobre essas dificuldades, revelou-se a necessidade da criação de um aplicativo que fornecesse auxílio durante as fases do processo de acolhimento e conectasse os adotantes com os protetores e instituições.

Com isso, surge o projeto *PetMatch*, idealizado para apoiar a adoção de animais de estimação, prática cada vez mais comum e importante em nossa sociedade, baseado em uma aplicação móvel com a finalidade de conectar usuários e seus futuros *pets*, de forma eficaz, organizada e intuitiva, facilitando o processo de pessoas que estão à procura de um novo animal de estimação.

O projeto se destaca como uma resposta para encarar os problemas e detalhes relacionados ao acolhimento de animais de estimação. O aplicativo não só simplifica a busca por um novo amigo, mas também desempenha uma função importante na diminuição da taxa de abandono de animais. Sua missão vai além da facilitação do processo adotivo, ela influencia a responsabilidade de uma adoção saudável, desestimulando a compra de animais e garantindo um *match* perfeito entre o animal e seu futuro dono.

Ao conectar de maneira eficaz adotantes, protetores e instituições, o *PetMatch* acelera as etapas do processo de adoção como também realiza uma análise apurada das entidades cadastradas, filtrando questões comportamentais que visam alcançar um *match* perfeito. Esta abordagem visa não somente encontrar lares amorosos para os animais, mas também criar um vínculo sólido entre os usuários e seus *pets*, evidenciando o compromisso que os amantes de animais têm por seus companheiros de quatro patas.

Dessa forma, o *PetMatch* não é apenas um aplicativo, é uma iniciativa que representa o compromisso com o bem-estar dos animais e a criação de lares repletos de amor e cuidado.

1.1 OBJETIVO

O aplicativo *PetMatch* foi pensado e desenvolvido com a missão principal de facilitar e fortalecer o processo de adoção de animais de estimação, criando uma ponte eficiente e segura entre pessoas interessadas em adotar com animais de abrigos/protetores independentes que precisam de um novo lar. Este projeto visa transcender as barreiras tradicionais da adoção, introduzindo uma plataforma digital que não apenas simplifica, mas também aprimora significativamente a experiência de adoção.

1.1.1 OBJETIVO GERAL

O objetivo do aplicativo *PetMatch* é conectar de forma eficiente e segura pessoas interessadas em adotar animais de estimação com abrigos e protetores independentes.

1.1.2 OBJETIVOS ESPECÍFICOS

O projeto tem alguns objetivos específicos, sendo eles:

- Avaliar a necessidade de uma plataforma com o intuito de adoção de animais;
- Levantar requisitos necessários para desenvolver a mesma, partindo da modelagem junto aos requisitos obtidos através de pesquisas;
- Projetar um aplicativo que auxilie na busca por interesse em adoção de animais;

1.2 ESTADO DA ARTE

Para o desenvolvimento do projeto, nos baseamos em alguns aplicativos móveis de temas similares, de forma a provar auxílio nos objetivos, funcionalidades e comparações de performances a fim de que a aplicação seja a opção ideal de ajuda no mercado de adoção animal.

Realizamos nossas buscas pela plataforma do *Google*, *Play Store* e *Apple Store*, buscando pelos aplicativos mais avaliados e de acordo com o *ranking* de *downloads* pelas próprias plataformas.

Visando isso, os aplicativos que tivemos como base foram: (1)*Na.Mosca*; (2) *MeAuDote*; (3)*Hyppet*.

- Na.Mosca

Ilustração 1 – Na.Mosca (logo)



Fonte: NaMosca(2021)

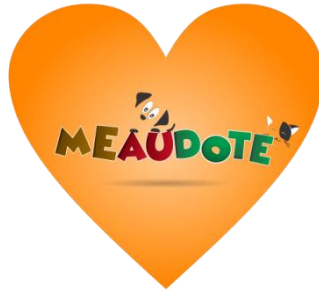
Como mostrado na ilustração 1, NaMosca é um aplicativo desenvolvido em 2021, reconhecido no mercado digital de adoção de pets, tendo como intuito ajudar na adoção de animais por meio de “match”, sendo disponibilizado para *download* de forma gratuita em dispositivos *IOS* e *Android*.

O modo de utilização consta com cadastro e *login*, informando sua cidade e instituição desejada para realizar a adoção. Após a etapa de registro no aplicativo, o usuário terá de responder um questionário sobre sua rotina e disponibilidade de tempo para se dedicar ao cuidado do animal. Ao finalizar o questionário, o usuário será direcionado para alguns pets que correspondem com suas respostas anteriores, podendo então causar o “match” entre ambos.

Após essas etapas, ocorre o contato entre o usuário e a instituição responsável pelo *pet*, dando assim, continuidade no processo de adoção, sendo realizada junto uma entrevista com o mesmo. Ao final do processo, o usuário receberá informações confidenciais no aplicativo, recebendo vouchers de desconto e acesso a conteúdos exclusivos, os quais serão disponibilizados pelas empresas parceiras da plataforma.

- MeAuDote

Ilustração 2 – MeAuDote (logo)



Fonte: MeAuDote

O MeAuDote apresentado na ilustração 2, foi desenvolvido em 2017 e hoje já conta com mais de 50 mil downloads. Seu intuito é voltado para doação e adoção de animais, podendo ser cadastradas *ONGS*, *CCZs*(Centros de Controle de Zoonoses), protetores, pessoas que estão atualmente com animais em lares temporários e pessoas com o intuito de doar e de adotar um animal.

Ao baixar o aplicativo, o usuário irá se cadastrar, completando seus dados pessoais e, logo em seguida, será encaminhado para a página inicial, onde aparecerão os pets que estão para adoção.

O sistema foi pensado e criado através de tecnologias como *Cloud computing* e *Geolocation*, sendo disponibilizados de forma gratuita em dispositivos *Android* e *IOS*.

- *Hyppet*

Ilustração 3 – *Hyppet* (logo)



Fonte: Hyppet

O *Hyppet* mostrado na ilustração 3, é um aplicativo desenvolvido em 2019 pelos criadores Vitor Pereira e Vitor Marques contando com mais de 1000 adoções. Seu desenvolvimento teve como objetivo facilitar o encontro do pet com o seu futuro dono, podendo ser utilizada por *ONGS* e pessoas jurídicas, disponível para *Android* e *IOS*.

Durante o cadastro comum, o usuário terá opções como entrevistas durante o processo de adoção e contato direto com o doador. Já no caso de *ONGS*, de maneira diferente, durante o cadastramento, deverá ser realizado o envio do *CNPJ* recebendo, assim, um selo de verificação pela plataforma.

O aplicativo conta, também, com a possibilidade de visualização de “Meu Perfil”, “Meus Pets” e “Pets Curtidos”.

O sistema também disponibiliza filtros para uma pesquisa específica do adotante para com seu futuro pet como raça, cor e tamanho, além de dispor uma seção apenas de pets de seus institutos parceiros.

2 DESENVOLVIMENTO

Nesse capítulo será abordado as fases do desenvolvimento do sistema proposto. Para tanto, esse capítulo será subdividido nos seguintes itens; análise do sistema, a estrutura do Banco de Dados, as ferramentas utilizadas para o desenvolvimento do *front-end* e as principais rotinas implementadas no *back-end*.

2.1 ANÁLISE DO SISTEMA

As linguagens utilizadas no desenvolvimento do aplicativo, contendo *frameworks*, banco de dados e as ferramentas necessárias.

Dart

Ilustração 04 – logo da linguagem *Dart*



Fonte: Wikipedia

Criada pela *Google*, o *Dart*, ilustração 4, é uma linguagem de programação fortemente tipada foi iniciada com a missão de ser a principal linguagem introduzida aos navegadores, sendo uma substituição a linguagem de programação *JavaScript*.

De acordo com BARRO (2021), projetos em *Dart*, tem duas formas de serem executados, ou por uma máquina virtual ou também compilados para a linguagem *JavaScript*. Ao iniciar no mercado, a linguagem teve um aumento considerado em sua utilização, sendo considerada uma linguagem multi-paradigma (linguagens que podem suportar mais de um paradigma de programação com função de solucionarem futuros problemas).

A linguagem foi escolhida para compor o projeto pois além de ser uma linguagem mais segura em termo de possíveis erros durante o desenvolvimento, sendo uma linguagem compilada, ela permite a criação de aplicações de forma

muito rápida devido aos seus recursos *JIT* e *AOT* (*just-in-time* e *ahead-of-time*) (BARRO, 2021).

Flutter

Ilustração 05 – logo do *framework Flutter*



Fonte: Flutter

Não sendo considerada uma linguagem de programação, *Flutter*, ilustração 5, é um *framework* (conjunto de bibliotecas que oferecem códigos pré-prontos com a finalidade de facilitar o desenvolvimento de aplicações) criada pela *Google* com foco voltado em multiplataforma em dispositivos moveis (ROCHA, 2023).

Atualmente, o *framework* sofreu um grande avanço no mercado, tornando-se mais utilizado, o que permitiu seu uso para criação também de aplicações *desktop*.

A escolha do *framework* se dá ao fato, de acordo com ROCHA (2023), da grande versatilidade em sua utilização, o que possibilita um aprendizado fácil e ágil, além da viabilidade de desenvolver aplicações em diferentes plataformas (*Android* e *IOS*) sendo usado apenas um código para ambos os sistemas operacionais.

NodeJS

Ilustração 06 – logo da linguagem *NodeJS*



Fonte: Node.js

A linguagem *NodeJS* representada na ilustração 06, teve seu início em 2009 pelo engenheiro de *software* Ryan Dahl, como propósito de sua criação ser uma segunda opção para o *Apache HTTP Server*, porém, atualmente é mantido pela *OpenJS Foundation*.

Segundo ESTRELLA (2023), tendo o objetivo de ser um ambiente de suporte de execução para a linguagem *JavaScript*, com sua utilização tem a possibilidade de criação de qualquer tipo de aplicação web, oferecendo serviços de servidores para sites estáticos e dinâmicos, indo até projetos tem base em microserviços.

A escolha dessa linguagem como *backend* deu-se, pois, além de ser utilizado em aplicações que futuramente possam ser escaláveis, ele tem a vantagem de ser executada em *single-thread*, ou seja, facilitam na alocação de seus recursos computacionais no sistema, sendo alocados apenas uma vez durante o tempo em que a aplicação for executada, aumentando, assim, sua performance (ESTRELLA, 2023).

Strap.io

Desenvolvido em *Node.js*, o *Strap.io* é um sistema *open-source* com o objetivo de favorecer a criação de serviços *REST* através de um painel administrativo totalmente funcional, sendo um caminho para a criação do *backend* do projeto (PINHEIRO, 2018).

SQLite

O *SQLite*, segundo VIEIRA (2021), diferentemente de outros bancos de dados, não possui a necessidade de armazenar informações dentro de um servidor, sendo um banco de dados relacional, ou seja, ele armazena e fornece informações e acesso para diferentes pontos de dados que estão relacionados a si, com código aberto e gratuito e de fácil entendimento, sendo muito utilizado em aplicações *mobile*, com um foco maior em dispositivos *Android*.

Junto a vantagem de não necessitar um servidor, o *SQLite* não precisa de nenhuma configuração, se tornando uma base de dados de leve carregamento, com uma dinâmica rápida e leve, suportando uma grande quantidade de linguagens durante seu uso, sendo bem versátil na utilização.

Unified Modeling Language (UML)

Entre vários mecanismos de análise disponíveis para o desenvolvimento de um sistema, optou-se pela escolha de *Unified Modeling Language (UML)* por ser uma linguagem de modelagem, isto é, uma forma de especificar um sistema de maneira visual (ANDRADE, 2019).

UML é reconhecida como a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com qualquer processo de desenvolvimento e com diferentes tecnologias de implementação [Furlan 1998]. A *UML* estabelece um padrão de modelagem de projetos de sistemas, incluindo seus aspectos conceituais, além das classes escritas em determinada linguagem de programação, processos de banco de dados e componentes de *software* reutilizáveis. (XAVIER e col, 2019, p.3).

2.1.1 ANÁLISE DE REQUISITOS

De acordo com Sommerville (2018, p. 85), “os requisitos de um sistema são descrições dos serviços que o sistema deve prestar e as restrições a sua operação. Esses requisitos refletem as necessidades dos clientes de um sistema que atende a um determinado propósito [...]”.

Requisitos Funcionais

Requisitos funcionais representados na ilustração 7, têm grande importância na realização de projetos, pois detém a função de apresentar os problemas e necessidades que devem ser atendidos e resolvidos pelo *software* por meio de funções ou serviços.

Abaixo, serão apresentados os requisitos funcionais presentes na aplicação.

Ilustração 07 – Requisitos funcionais do sistema

Requisito	Descrição
RF001	O sistema deve permitir cadastro de usuário
RF002	O sistema deve permitir que o usuário realiza acesso com suas informações.

RF003	O sistema deve permitir que o usuário edite suas informações pessoais.
RF004	O sistema deve permitir que o usuário já logado na sua conta, cadastre um <i>PET</i>
RF005	O sistema deve permitir que o usuário gerencie as informações do <i>PET</i> cadastrado
RF006	O sistema deve permitir que o usuário logado mostre os <i>PET'S</i> que estão para adoção.
RF007	O sistema permite que o usuário realize a adoção do <i>PET</i> desejado.
RF008	O sistema permite que o usuário ajude financeiramente uma instituição.

Fonte: Elaborado pelos autores (2023).

Requisitos não funcionais

Requisitos não funcionais representado na ilustração 8, são as especificações impostas a um sistema que definem seus atributos de qualidade, sendo geralmente indicados por adjetivos como segurança, desempenho e escalabilidade, e Sommerville (2011) complementa “requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários”. (SOMMERVILLE, 2011).

Abaixo, serão apresentados os requisitos não funcionais presentes na aplicação.

Ilustração 08 – Requisitos não funcionais do sistema

Requisito	Descrição
RNF001	A interface do sistema deve ser de fácil navegação.
RNF002	O desenvolvimento do sistema deve ser realizado utilizando o framework Flutter.
RNF003	O sistema deve salvar informações do usuário através do login.
RNF004	O sistema deve utilizar da senha para proteger informações pessoais do usuário.

Fonte: Elaborado pelos autores (2023).

2.1.2 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso tem por finalidade documentar o que o sistema faz do ponto de vista do usuário, descrevendo as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema.

De acordo com BOOCH, RUMBAUCH e JACOBSON (2005, p. 227):

“Um caso de uso especifica o comportamento de um sistema ou de parte de um sistema e é uma descrição de um conjunto de sequência de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.” (BOOCH, RUMBAUCH e JACOBSON, p. 227, 2005).

Os casos de uso são criados de uma forma fácil de entender, para que os desenvolvedores e os usuários finais cheguem a uma compreensão comum, sem ter algo realmente implementado.

O Diagrama de Caso de Uso pode ser consultado no Apêndice A. Abaixo, serão detalhados os casos de uso apresentados acima.

- Ilustração 09, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Cadastro do usuário”.

Ilustração 09 – Caso de uso 01 – Cadastro do usuário

UC001 – Cadastro do usuário
Descrição: Cadastrar a conta do usuário no aplicativo
Pré-condições: --
Pós-condições: O usuário terá a disposição de utilizar o aplicativo, com o seu e-mail e senha.
Requisitos funcionais: RF001
Fluxo Principal
Passo 1: O usuário irá abrir o aplicativo no seu dispositivo
Passo 2: Irá aparecer na primeira tela um botão de “cadastre-se”
Passo 3: Ao clicar no botão, aparecerá um formulário para ser preenchido com informações pessoais do usuário
Passo 4: Ao terminar o preenchimento, o usuário irá enviar suas informações e será direcionado para a tela inicial do aplicativo.
Fluxo alternativo A
Passo 1: Ao realizar seu preenchimento (passo 4), o usuário esquece de preencher algum campo obrigatório
Passo 2: O usuário recebe uma mensagem notificando a obrigação do preenchimento do campo.
Passo 3: Ao terminar o preenchimento, o usuário irá enviar suas informações e será direcionado para a tela inicial do aplicativo.

Fonte: Elaborado pelos autores (2023).

- Ilustração 10, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Login do usuário”.

Ilustração 10 – Caso de uso 02 – Login do usuário

UC002 – Login do usuário
Descrição: Entrar na conta já existente com e-mail e senha
Pré-condições: Já ter realizado o cadastramento anteriormente
Pós-condições: O usuário irá entrar no aplicativo na tela inicial
Requisitos funcionais: RF002
Fluxo Principal
Passo 1: O usuário irá abrir o aplicativo no seu dispositivo
Passo 2: Irá aparecer na primeira tela um botão de “Entrar”
Passo 3: Ao clicar no botão, aparecerá os campos de e-mail e senha
Passo 4: Ao terminar o preenchimento, o usuário será direcionado para tela inicial do aplicativo, já estando na sua conta
Fluxo alternativo A
Passo 1: Ao clicar no botão de “Entrar”, o usuário preenche com alguma informação incorreta
Passo 2: O usuário recebe uma mensagem notificando que alguma informação está incorreta e retorna para o Passo 2 do fluxo principal

Fonte: Elaborado pelos autores (2023).

- Ilustração 11, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Cadastrar PET”.

Ilustração 11 – Caso de uso 03 – Cadastrar PET

UC003 – Cadastrar PET
Descrição: Cadastrar um novo PET para adoção
Pré-condições: Já ter realizado o login anteriormente
Pós-condições: O usuário terá um PET cadastrado no aplicativo
Requisitos funcionais: RF004
Fluxo Principal
Passo 1: O usuário já logado na sua conta, irá clicar no botão “Cadastrar um focinho”

Passo 2: O usuário será direcionado para a página de cadastro
Passo 3: O usuário preencherá as informações necessárias
Passo 4: Chegará uma mensagem de cadastrado feito com sucesso
Passo 5: O usuário será direcionado para a página com o seu PET ja cadastrado
Fluxo alternativo A
Passo 1: Ao realizar o passo 4, o usuário deixa de preencher algum campo obrigatório
Passo 2: O usuário recebe uma mensagem notificando a obrigação do preenchimento do campo.
Passo 3: O usuário é direcionado de volta ao formulário para preencher o campo vazio
Passo 4: Ao finalizar o cadastro, o usuário clica para salvar e ele retorna para o passo 5 do Fluxo Principal

Fonte: Elaborado pelos autores (2023).

- Ilustração 12, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Visualizar PET”.

Ilustração 12 – Caso de uso 04 – Visualizar PET

UC004 – Visualizar PET
Descrição: Visualizar PET’S que estão para adoção
Pré-condições: Já ter realizado o login anteriormente
Pós-condições: O usuário irá visualizar a listagem de PET’S
Requisitos funcionais: RF006
Fluxo Principal
Passo 1: O usuário irá abrir o aplicativo no seu dispositivo
Passo 2: O usuário realiza o login
Passo 3: Ao terminar o preenchimento, o usuário será direcionado para tela

inicial do aplicativo
Passo 4: Na tela inicial, irá ter uma opção da listagem dos animais
Passo 5: Ao clicar no botão, o usuário será direcionado para a tela de informações

Fonte: Elaborado pelos autores (2023).

- Ilustração 13, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Adotar PET”.

Ilustração 13 –Caso de uso 05 – Adotar PET

UC005 – Adotar PET
Descrição: o usuário poderá adotar um PET
Pré-condições: Já ter realizado o login anteriormente e o pet já estar cadastrado no aplicativo
Pós-condições: O usuário irá conseguir adotar um PET
Requisitos funcionais: RF007
Fluxo Principal
Passo 1: O usuário já logado, visualizará na página principal e lista de PET’S para adoção
Passo 2: O usuário se identifica com algum PET e entra na página do mesmo
Passo 3: Ao entrar, aparecerá as informações do PET, junto com um botão de “Quero adotar”
Passo 4: Abre o modal com o botão de E-mail ou WhatsApp do doador
Passo 5: O usuário escolhe a forma de contato e é direcionado para a mesma
Fluxo Alternativo
Passo 1: O usuário clica em “Adoção” pelo menu
Passo 2: O usuário é redicionado a página com filtros de busca
Passo 3: O usuário preenche de acordo com o PET desejado e clica em “Buscar”
Passo 4: Ao clicar, será direcionado a uma página com PETS de acordo com as informações definidas anteriormente

Passo 5: Ao clicar no PET desejado, o usuário é redirecionado para informações do animal com o botão de “Quero adotar”
Passo 6: Clica no botão e abre o modal com as opções de E-mail ou WhatsApp
Passo 7: O usuário escolhe a forma de contato e é direcionado para a mesma

Fonte: Elaborado pelos autores (2023).

- Ilustração 14, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Gerenciar PET”.

Ilustração 14 – Caso de uso 06 – Gerenciar PET

UC006 – Gerenciar PET
Descrição: Editar PET’S que estão para adoção
Pré-condições: Já ter realizado o login anteriormente
Pós-condições: O usuário irá visualizar a listagem de PET’S
Requisitos funcionais: RF005
Fluxo Principal
Passo 1: O usuário irá abrir o aplicativo no seu dispositivo
Passo 2: O usuário realiza o login
Passo 3: Ao terminar o preenchimento, o usuário será direcionado para tela inicial do aplicativo
Passo 4: Ao clicar, a tela irá abrir e ao clicar no PET, o usuário será redirecionado para a tela de informação do PET
Passo 5: O usuário irá clicar no três pontinho no canto superior direito da tela e irá clicar na opção de editar informações
Passo 6: O usuário irá editar as informações e clicará em concluir
Passo 7: Chegará uma notificação de informações editadas com sucesso
Passo 8: O usuário será direcionado para a tela do PET com as informações salvas

Fonte: Elaborado pelos autores (2023).

- Ilustração 15, apresentada o fluxo principal e o fluxo alternativo do caso de uso “Adotar PET”.

Ilustração 15 – Caso de uso 07 – Ajudar uma instituição

UC007 – Ajudar uma instituição
Descrição: o usuário poderá ajudar uma instituição
Pré-condições: Já ter realizado o login anteriormente e o pet já estar cadastrado no aplicativo
Pós-condições: O usuário irá conseguir adotar um PET
Requisitos funcionais: RF008
Fluxo Principal
Passo 1: O usuário já logado, visualizará na página principal e lista de PET'S para adoção
Passo 2: O usuário se identifica com algum PET e entra na página do mesmo
Passo 3: Ao entrar, aparecerá as informações do PET, junto com um botão de quero ajudar
Passo 4: O usuário clica no botão e abre o modal com texto da ONG e botão de copiar o código pix
Passo 5: O usuário clica em copiar o código e o modal fecha, dando a possibilidade de realizar a transferência.

Fonte: Elaborado pelos autores (2023).

2.1.3 FLUXO DE EVENTOS

Segundo MACORATTI (2023), o fluxo de eventos tem por intenção caracterizar o início e o término do caso de uso, estabelecer a interação dos atores no processo em descrição, gerando assim um fluxo principal e um fluxo alternativo do comportamento de cada caso de uso do sistema.

A descrição dos fluxos de eventos desse sistema pode ser consultada no Apêndice B.

Ilustração 17 – Código implementação *applyFilters*

```

54 void applyFilters() {
55   var filters = <String, List<String>> ();
56   var filterKind = false;
57   for (final kindOption in state.kind) {
58     if (!kindOption) {
59       filterKind = true;
60       break;
61     }
62   }
63   if (filterKind) {
64     filters['species'] = <String> [];
65     for (var i = 0 ; i < state.kind.length ; ++i) {
66       if (state.kind[i]) {
67         filters['species']!.add(PetKind.values[i].toString());
68       }
69     }
70   }
71   var filterGender = false;
72   for (final genderOption in state.gender) {
73     if (!genderOption) {
74       filterGender = true;
75       break;
76     }
77   }
78   if (filterGender) {
79     filters['sex'] = <String> [];
80     for (var i = 0 ; i < state.gender.length ; ++i) {
81       if (state.gender[i]) {
82         filters['sex']!.add(PetGender.values[i].toString());
83       }
84     }
85   }
86 }

```

Fonte: Elaborado pelos autores (2023).

O código representado na ilustração 17, demonstra como é feita a filtragem das categorias selecionadas:

Para cada categoria, ocorre uma verificação para saber se alguma opção não está selecionada. Após isso, para cada categoria que haja filtragem, as opções que devem ser filtradas são adicionadas no objeto “*filters*”, um *Map<String, dynamic>* onde a chave representa a categoria e “*dynamic*” os filtros.

Ilustração 18 – Código do *HomePageScreen*

```

29 child: Column(//
30   crossAxisAlignment: CrossAxisAlignment.start,
31   children: [
32     Padding(
33       padding: const EdgeInsets.only(top: 8, left: 8),
34       child: Text('Pets de instituição >', style: Theme.of(context).textTheme.titleSmall),
35     ),
36     Consumer<HomeController>{
37       builder: (context, controller, _) {
38         final pets = controller.institutionPetsList;
39         return Expanded(child: ListView.builder(
40           scrollDirection: Axis.horizontal,
41           itemCount: pets.length,
42           itemBuilder: (context, index) {
43             return Padding(padding: const EdgeInsets.only(top: 30.0, left: 10.0),
44               child: GestureDetector(
45                 onTap: () {
46                   Navigator.of(context).pushNamed('/adoption/details', arguments: pets[index]);
47                 },
48               child: Column(
49                 children: [
50                   pets[index].localPicture.isNotEmpty ?
51                     CircleAvatar(
52                       backgroundColor: PetMatchColors.pinky,
53                       radius: 42,
54                       child: CircleAvatar(backgroundImage: FileImage(File(pets[index].localPicture)), radius: 40),
55                     ) :
56                     const CircleAvatar(backgroundColor: PetMatchColors.grayish, radius: 40),
57                   const Padding(padding: EdgeInsets.only(bottom: 16)),
58                   Text(pets[index].name, style: Theme.of(context).textTheme.bodySmall!.copyWith(color: Colors.black))

```

Fonte: Elaborado pelos autores (2023).

A ilustração 18 demonstra como a lista do *controller* é consumida para criar a lista de *pets*. A classe *Consumer* é utilizada para providenciar acesso do *HomepageController* à *widget*, e um *ListView* é utilizado para criar a lista dinâmica com scroll a partir da lista de *pets* do *controller*.

Ilustração 19 – Código de *AdoptionController*

```

14 class AdoptionController extends ChangeNotifier {
15   PetOwnerData contacts = PetOwnerData.kEmpty;
16
17   void fetchOwnerContact(int ownerId) async {
18     if (contacts != PetOwnerData.kEmpty) {
19       return;
20     }
21
22     final token = await Authentication.getToken();
23     HttpService().getPetOwnerContact(token, ownerId)
24       .then((result) {
25         result.fold(
26           (ownerContacts) {
27             contacts = ownerContacts;
28             notifyListeners();
29           },
30           (error) => throw Exception('Program in invalid state because of: $error'),
31         );
32       });
33   }
34 }

```

Fonte: Elaborado pelos autores (2023).

O código apresentado na ilustração 19 demonstra a lógica do *controller* da página de adoção, efetuando o *download* dos dados do dono do animal selecionado anteriormente.

Ilustração 20 – Código de ação do mecanismo *Pix*

```

193 actions: {
194   FilledButton(
195     style: PetMatchThemeHelper.pinkyButtonStyleSmall,
196     onPressed: () {
197       Clipboard.setData(ClipboardData(text: controller.contacts.pix)).then(() {
198         ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('PIX copiado com sucesso!')));
199       });
200       Navigator.of(context).pop();
201     },
202     child: const Text('Copiar código PIX')
203   ),
204 }

```

Fonte: Elaborado pelos autores (2023).

O código mostrado na ilustração 20 mostra a ação do diálogo de *pix* que realizará uma cópia do *PIX* da instituição (a partir dos dados baixados pelo *controller* anteriormente) para o *Clipboard* do sistema operacional - e então dispara um *Toast* (*SnackBar*) notificando o usuário de que a cópia foi efetuada.

Ilustração 21 – Código do método *registerPet*

```

114 Future<RegisterPetResult> registerPet() async {
115   if (name.isEmpty) {
116     return Future.value(RegisterPetResult.emptyName);
117   }
118   if (city.isEmpty) {
119     return Future.value(RegisterPetResult.emptyCity);
120   }
121   if (state.isEmpty) {
122     return Future.value(RegisterPetResult.emptyState);
123   }
124   final token = await Authentication.getToken();
125   final user = await Authentication.getUser();
126   final result = await HttpService().registerPet(token, Pet(
127     name: name,
128     state: state,
129     city: city,
130     kind: selectedKind,
131     gender: selectedGender,
132     size: selectedSize,
133     age: selectedAge,
134     owner: user!.id,
135     ownerCnpj: user.cnpj,
136     cares: medicalCares
137       .whereIndexed((index, element) => medicalCareSelections[index])
138       .toList(),
139     sociableTraits: sociableTraits
140       .whereIndexed((index, element) => sociableTraitSelections[index])
141       .toList(),
142     sociableWiths: sociableRelations
143       .whereIndexed((index, element) => sociableRelationSelections[index])
144       .toList(),
145     localPicture: _image != null ? _image!.path : ''
146   ));
147   if (result.isErr) {
148     debugPrint('Sresult');
149     return RegisterPetResult.error;
150   }
151   return RegisterPetResult.ok;
152 }
153 }

```

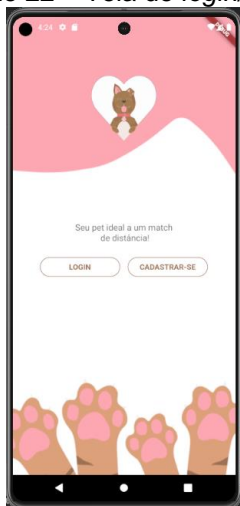
Fonte: Elaborado pelos autores (2023).

O código na ilustração 21 acima apresenta a função de cadastro do *pet* na classe *RegisterPetController*, onde é feito o upload das informações através da chamada *registerPet* providenciada pelo serviço *HttpService*, após todas as informações obrigatórias terem sido preenchidas na tela pelo usuário.

2.4 CAMADA DE APRESENTAÇÃO

Nesta seção, são apresentadas as telas do aplicativo, criadas para atender os requisitos propostos, e estabelecer um *layout* estrutural para a aplicação.

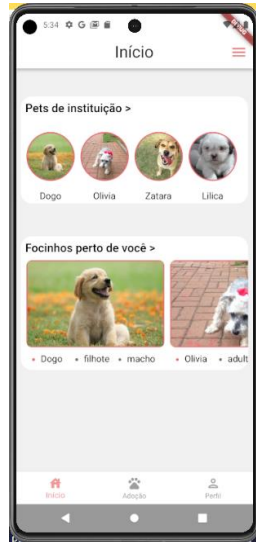
Ilustração 22 – Tela de *login*/cadastro



Fonte: Elaborado pelos autores (2023).

A ilustração 22 demonstra o *layout* inicial da aplicação, uma tela *clean* com campo de opções de entrar no aplicativo, como realizando o *login* caso já tenha conta ou iniciando uma nova conta clicando em cadastrar-se.

Ilustração 23 – Tela inicial do aplicativo



Fonte: Elaborado pelos autores (2023).

Como mostrado na ilustração 23, após realizar o primeiro acesso no aplicativo, precisará acessar a tela de login, tendo como campos obrigatórios o *email* e senha, e assim sendo direcionado para a tela inicial ilustrada na imagem acima. Chegando na tela de *home*, a mesma irá conter carrosséis de *pets* vindos de Instituições, *pets* perto do usuário e da categoria “idoso”.

Ilustração 24 – Tela de filtro de adoção

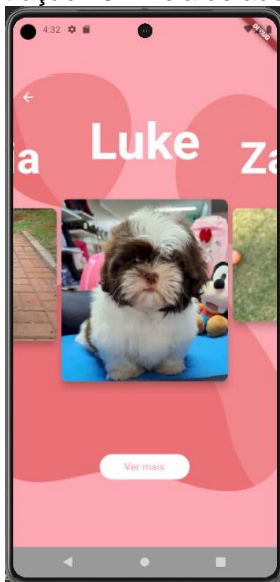


Fonte: Elaborado pelos autores (2023).

Na ilustração 24 apresenta a tela de buscar um pet, contendo campos onde o usuário poderá escolher as características que deseja que seu pet tenha, como por

exemplo, o porte do animal, sua idade e sexo. Após utilizar os filtros e clicar em buscar, ocorre o redirecionamento para a tela de adoção, representada pela Ilustração 25.

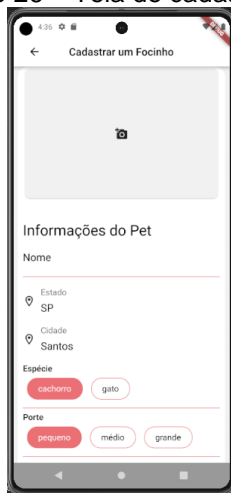
Ilustração 25 – Tela de adoção



Fonte: Elaborado pelos autores (2023).

Na ilustração 25 acima, apresenta a página de adoção dos *pets*, contendo um carrossel de animais disponíveis para adoção, a depender da escolha de filtros na página anterior, mostrando o nome e sua respectiva imagem, contendo também um botão para saber mais informações do *pet*.

Ilustração 26 – Tela de cadastro do pet



Fonte: Elaborado pelos autores (2023).

Na ilustração 26 acima, a tela mostra ao usuário campos para cadastrar um animal no aplicativo, com informações como nome, raça, estado e cidade. Após

completar todos os campos obrigatórios, o usuário irá clicar em “Cadastrar Focinho”, botão que se encontra no final da página, e assim o *pet* já estará disponível para ser adotado.

Ilustração 27 – Modal Quero Adotar



Fonte: Elaborado pelos autores (2023).

Como mostra acima na ilustração 27, a página apresenta a tela de informações do *pet*, contendo o nome do bichinho, um pequeno texto de apresentação e algumas informações definidas pelo usuário que está responsável pela adoção do *pet*. Ao final da página, um botão de “Quero Adotar” estará disponível, o que levará o usuário ao modal de informações do responsável por esse animal, para que o usuário consiga realizar a adoção.

3. RESULTADO

Foi realizado uma pesquisa com 16 usuários sendo passadas tarefas para serem realizadas dentro da aplicação, sem limitação de tempo para realizá-las. Após todos terminarem, foram computadas suas informações e em seguida responderam um questionário no *Google Forms* com suas opiniões sobre a aplicação.

Tarefas passadas para os usuários:

- Criar conta;
- Entrar com *login* e senha;
- Cadastrar um *pet*;
- Editar informação do *pet*;
- Entrar no perfil de algum *pet* que está para adoção;
- Realizar uma adoção de *pet*;
- Sair da conta.

3.1 RESULTADOS DOS TESTES

De acordo com os resultados realizados por usuários nos meses de Outubro e Novembro, com idades entre 18 e 50 anos, sendo moradores de Santos, Praia Grande e Itanhaém, pode-se observar que a maioria obtiveram êxito na realização dos testes, não tendo dificuldade na hora de realizar as tarefas, sendo uma média de 8 minutos de realização completa das tarefas.

Após o término das atividades, os usuários enviaram feedbacks sobre o aplicativo, vindo acompanhadas de algumas melhorias, como por exemplo, o aumento da fonte. O aplicativo foi muito bem avaliado e recomendado, sendo elogiado por conta do design clean e bem intuitivo, prático de achar e realizar as funções exigidas, por conter funcionalidades de grande importância, recebeu elogios também pela funcionalidade do carrossel “De uma chance para eles” e pela opção de poder ajudar uma instituição.

3.2 RESULTADOS DOS QUESTIONÁRIOS

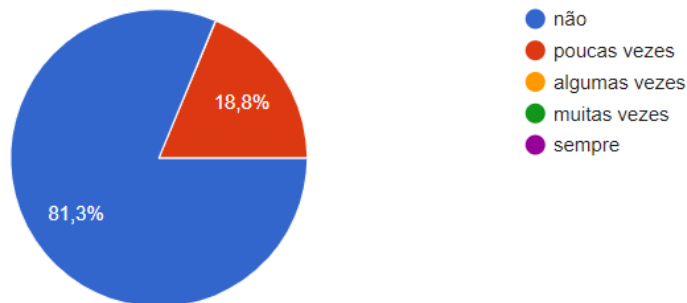
Nesse tópico serão apresentados os resultados dos questionários enviados para os usuários que realizaram os testes. O questionário foi realizado *pele Google Forms*, contendo 5 perguntas baseadas no *WAMMI (Webaplicativo Analysis and MeasureMent Inventory)*.

Ilustração 28 – Gráfico das respostas da pergunta 1

É difícil se mover neste aplicativo.

 Copiar

16 respostas



Fonte: Elaborado pelos autores (2023).

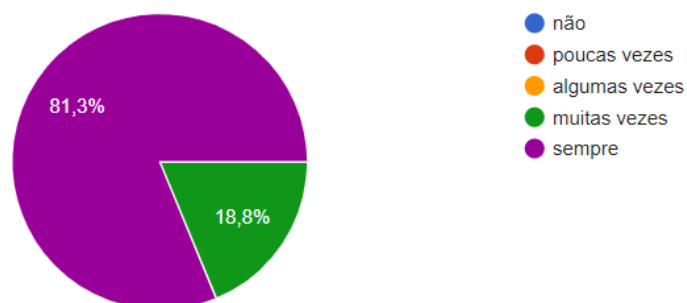
Na ilustração 28 é apresentado o gráfico com o resultado da pergunta 1 respondida pelos participantes. Na análise, a maioria dos usuários respondeu que foi de fácil navegação no aplicativo, sendo compreendido que está bem intuitivo e prático nos campos.

Ilustração 29 – Gráfico das respostas da pergunta 2

Eu pude encontrar facilmente o que eu queria neste aplicativo.

 Copiar

16 respostas



Fonte: Elaborado pelos autores (2023).

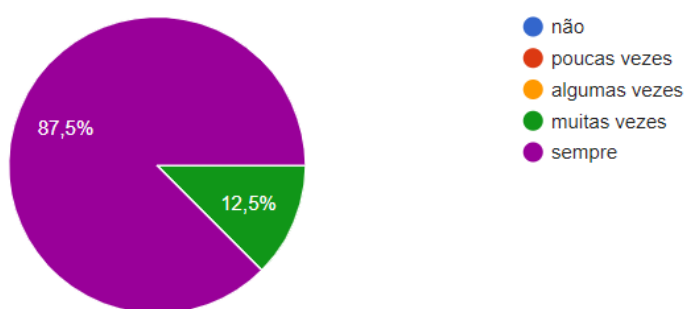
Na ilustração 29 é apresentado o gráfico com o resultado da pergunta 2. Diante dos resultados, 81,3% participantes votaram que encontraram com facilidade informações que precisavam para concluir as tarefas, enquanto 18,8% votaram que, na maioria das vezes, obteve facilidade no encontro das informações. No geral, tornando assim as páginas objetivas e claras como esperado.

Ilustração 30 – Gráfico das respostas da pergunta 3

As páginas deste aplicativo são muito atrativas

 Copiar

16 respostas



Fonte: Elaborado pelos autores (2023).

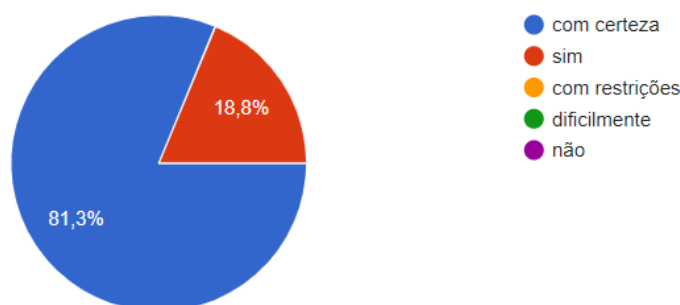
Na ilustração 30 é apresentado o gráfico com o resultado da pergunta 3. Dos 16 participantes, 87,5% deles acharam as páginas atrativas, podendo levar ao fato a dificuldade de visualizar informações por conta do tamanho da fonte, precisando melhorar esse ponto e deixar com fontes melhores para leitura.

Ilustração 31 – Gráfico das respostas da pergunta 4

Eu me sinto no controle quando utilizo este aplicativo

 Copiar

16 respostas



Fonte: Elaborado pelos autores (2023).

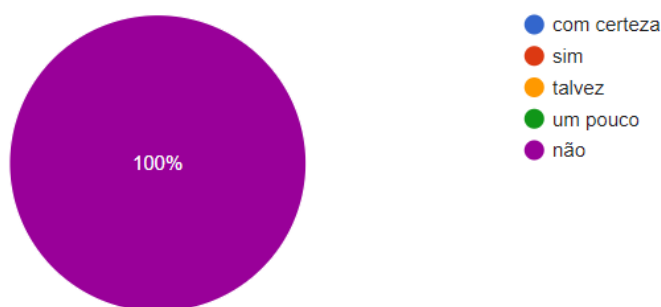
Na ilustração 31 é apresentado o gráfico com o resultado da pergunta 4. Dos 16 respondentes, 81,3% se sentiram no controle enquanto utilizava a aplicação, podendo passar uma perspectiva que não houve interrupções durante a realização das tarefas.

Ilustração 32 – Gráfico das respostas da pergunta 5

Aprender a encontrar meu caminho de volta neste aplicativo é um problema.

 Copiar

16 respostas



Fonte: Elaborado pelos autores (2023).

Na ilustração 32 é apresentado o gráfico com o resultado da pergunta 5, onde os 16 participantes votaram que durante a navegação não houve dificuldades em transitar pelas telas, tendo fácil acesso e de forma clara e objetiva onde e em quais botões clicar para ir e voltar no aplicativo.

3.3 CONCLUSÃO

Diante a realização da documentação e projeto, foi possível identificar que o índice de abandono de animais nos últimos anos vem crescendo a cada dia (AMPARA, 2021). Porém, de acordo com LEMOS (2021), existe um crescimento de pessoas que estão à procura de ter um animal de estimação para adoção, sendo uma das soluções para o índice de abandono diminuir.

Com isso, surgiu o propósito do aplicativo *PetMatch*, uma aplicação que possibilita facilitar o contato de pessoas que estão à procura de adotar um bichinho, tendo como funcionalidades cadastrar o *pet*, adotar, gerenciar e visualizar animais disponíveis para adoção.

Foram realizados testes com 16 usuários com o objetivo de testar a usabilidade e navegação, sendo passadas tarefas a eles para realizarem no

aplicativo, e após a análise feita dos resultados, pôde-se observar um alto nível de satisfação dos usuários que testaram o aplicativo e responderam às perguntas, com *feedbacks* positivos em relação a *design*, praticidade na navegação, propósito da aplicação, junto recebendo sugestões de melhoria, como por exemplo, o aumento do tamanho da fonte para melhor visibilidade.

Desde o começo do desenvolvimento até sua entrega final surgiram funcionalidades para melhor utilização do aplicativo, no entanto algumas delas não foram implementadas, porém, através de trabalhos futuros essas funções poderão ter uma continuidade no projeto.

As funcionalidades para melhorar a utilização e experiência do aplicativo são as de criação de um feed interativo (para compartilhar experiências pessoais ou animais perdidos com sua devida localização), criação de uma aba exclusiva para ajudar ONG's (com a possibilidade de doação de alimento e não somente financeira), parcerias com *PetShop's* e veterinários (através de anúncios, cupons no próprio *app* e descontos em vacinação e castração de animais) e brindes para o usuário que completar 1 ano de adoção de um de nossos *pets*. Essas melhorias e novas *features* terão papel importante na intenção de manter o usuário dentro do aplicativo por um maior período, e que o mesmo possa ter outras utilidades além de somente a adoção do animal, evitando a desinstalação do aplicativo pós uso.

Os benefícios que a aplicação *PetMatch* trará irá muito além do ato de adotar um animal, mas sim do cuidado com a saúde e bem-estar pós adoção, da interação entre outros usuários "pais de *pet*" para obtenção de dicas, na ajuda financeira de instituições e em ajudar a busca e encontro animais perdidos, possibilitando a devolução de tantos animais amados para suas famílias.

Após análise do projeto e dos fatos citados anteriormente, em suma, obteve-se êxito nos objetivos propostos para o projeto, bem como sua aprovação vinda dos usuários que o utilizaram.

REFERÊNCIAS

ANDRADE, A. (ED.). **O que é UML?** Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-uml>>. Acesso em: 11 dez. 2023.

ANIMAL, AMPARA. Abandono de animais domésticos cresce 70% durante a pandemia. *In: Cobasi. Cobasi Blog. [S.l.].* 25 fev. 2021. Disponível em: <<https://blog.cobasi.com.br/abandono-de-animais-domesticos-cresce-70-durante-a-pandemia/>>. Acesso em: 5 nov. 2023.

BARRO, B. (ED.). **Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem.** 01 mar. 2021 Disponível em: <<https://www.hostinger.com.br/tutoriais/linguagem-dart>>. Acesso em: 11 dez. 2023.

BOOCH, G.; RUMBAUGH, J.; JACOBSON. **The Unified Modeling Language User Guide.** 2nd. Ed. Addison-Wesley: Professional, 2005. 227 p.

BRASIL, I. P. Censo **Pet: 139,3 milhões de animais de estimação no Brasil.** 2019. Disponível em: <<http://institutopetbrasil.com/imprensa/censo-pet-1393-milhoes-de-animais-de-estimacao-no-brasil/>>. Acesso em: 25 Mai.2023.

CAMPOS, Diego P. G.; PEREIRA, Kaique E.; CARVALHO, Samuel A.; et al. APLICAÇÃO DE ARMAZENAMENTO EM NUVEM UTILIZANDO A PLATAFORMA NODE.JS. **RE3C - Revista Eletrônica Científica de Ciência da Computação**, v. 11, n. 1, 2016. Disponível em: <<http://revistas.unifenas.br/index.php/RE3C/article/view/160>>. Acesso em: 05 Set. 2023.

DATE, C.J. (2004) **Introdução a sistemas de banco de dados.** Tradução de: Daniel Vieira. 8. Ed. Corpus, Rio de Janeiro. Acesso em: 21 Nov. 2023.

ESTRELLA, C. (ED.). **O Que é Node.Js: Aplicações Práticas e Como Instalá-lo.** Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-node-js>>.

Flutter – Medium. Disponível em: <<https://medium.com/flutter>>. Acesso em: 11 dez. 2023.

GRAD, B.; BERGIN, T. J. **History of Database Management Systems.** IEEE Ann. Hist. Comput., vol. 31, no. 4, p. 3 - 5, 2009.

Hyppet. Disponível em: <<https://hyppet.com/>>. Acesso em: 11 dez. 2023.

LEMOS, S. (ED.). **Cresce o número de adoções e de abandono de animais na pandemia**. Disponível em: <[MACORATTI, J. **UML - Casos de Uso - Conceitos \(revisão\)**. Disponível em: <\[https://www.macoratti.net/11/10/uml_rev1.htm\]\(https://www.macoratti.net/11/10/uml_rev1.htm\)>. Acesso em: 11 dez. 2023.](https://jornal.usp.br/atualidades/cresce-o-numero-de-adocoes-e-de-abandono-de-animais-na-pandemia/#:~:text=Cresce%20o%20n%C3%BAmero%20de%20ado%C3%A7%C3%B5es%20e%20de%20abandono%20de%20animais%20na%20pandemia,-Os%20animais%20n%C3%A3o&text=Dados%20da%20Uipa%20%E2%80%93%20Uni%C3%A3o%20Internacional,mais%20reclusas%20em%20suas%20resid%C3%A4ncias.>. Acesso em: 11 dez. 2023.</p></div><div data-bbox=)

MeAuDote – Um ato de amor que – Apps no Google Play. Disponível em: <https://play.google.com/store/apps/details?id=br.com.meuadote&hl=pt_BR&pli=1>. Acesso em: 11 dez. 2023.

MY AFFINITY. **Os motivos por trás do abandono de um animal de estimação**. 2010. Disponível em: <<https://www.affinity-petcare.com/br/os-motivos-por-tras-do-abandono-de-um-animal-de-estimacao>> Acesso em: 23 mai.2023.

Na mosca app – Seu pet, num match. Disponível em: <<https://namoscaapp.com.br/>>. Acesso em: 11 dez. 2023.

NODE.JS. **Node.js**. Disponível em: <<https://nodejs.org/en>>.

PINHEIRO, F. (ED.). **Criando APIs com o Strapi.io**. Disponível em: <<https://www.treinaweb.com.br/blog/criando-apis-com-o-strapi-io>>. Acesso em: 11 dez. 2023.

ROCHA, W. (ED.). **Desenvolvimento Ágil: Descubra o Poder do Flutter em Apps Mobile**. 2023. Disponível em: <<https://www.dio.me/articles/desenvolvimento-agil-descubra-o-poder-do-flutter-em-apps-mobile>>. Acesso em: 11 dez. 2023.

SILBERSCHATZ, Abranham; KORTH, Henry F. e SUDARSHAN, S. **Sistema de Banco de Dados**. 3TM ed. São Paulo: MAKRON Books, 1999.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

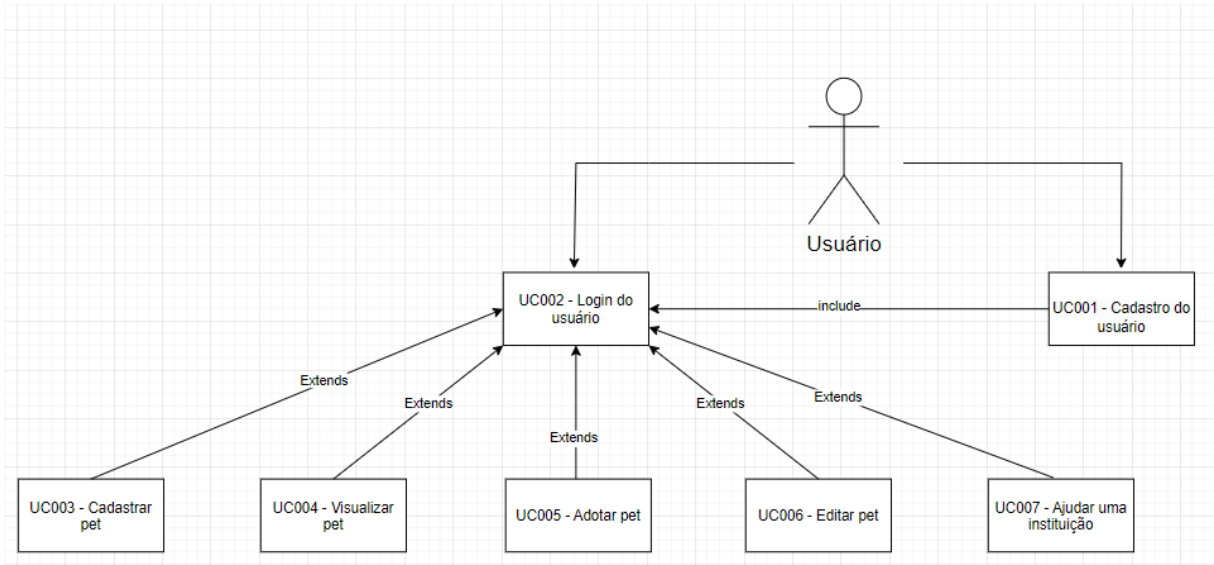
SOMMERVILLE, Ian. **Engenharia de software**, 10^a ed. Editora Pearson, 2018

VIEIRA, D. (ED.). **SQLite: o que é, como funciona e qual é a diferença entre o MySQL**. Disponível em: <<https://www.hostgator.com.br/blog/sqlite-o-que-e-como-funciona-e-qual-e-a-diferenca-entre-o-mysql/>>. Acesso em: 11 dez. 2023.

WIKIPEDIA. **Ficheiro:Dart programming language logo.svg – Wikipédia, a enciclopédia livre**. Disponível em: <https://pt.m.wikipedia.org/wiki/Ficheiro:Dart_programming_language_logo.svg>. Acesso em: 11 dez. 2023.

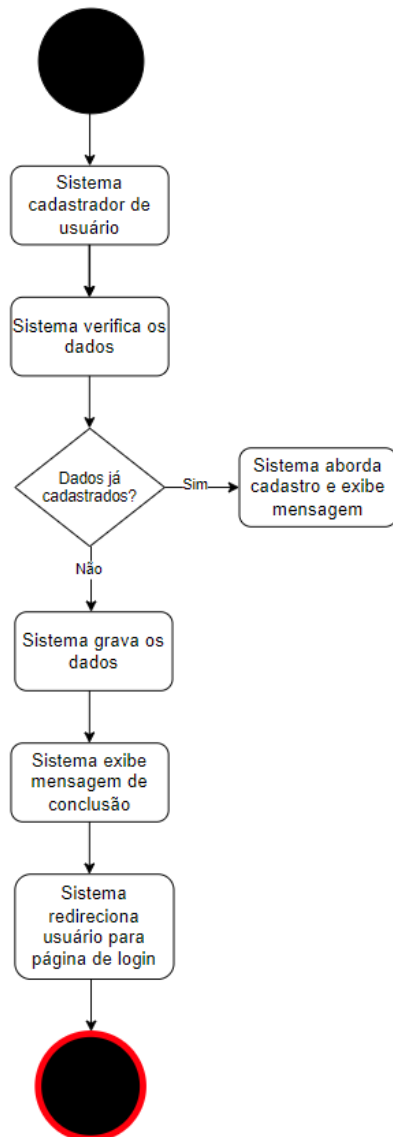
XAVIER e col. **Aplicação da UML no contexto das metodologias ágeis**. 2019. Disponível em: <<https://sol.sbc.org.br/index.php/encompif/article/view/6353/6250>>. Acesso em: 05 Set 2023.

APÊNDICE A



APÊNDICE B

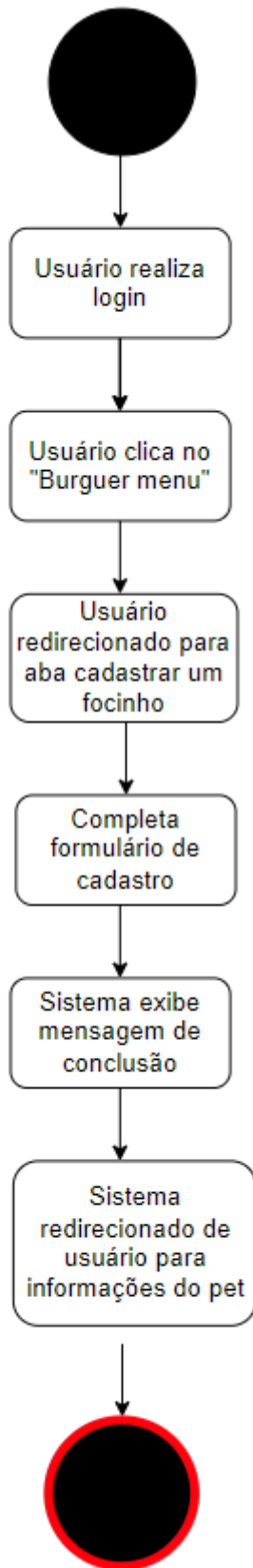
Cadastro do usuário



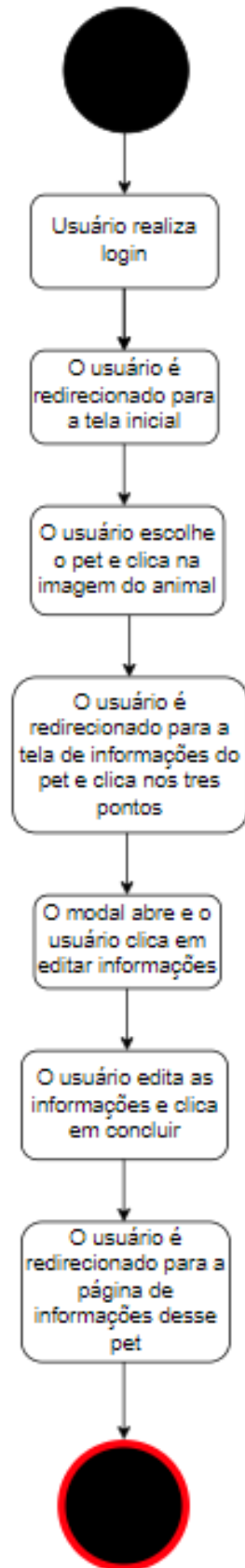
Login do Usuário



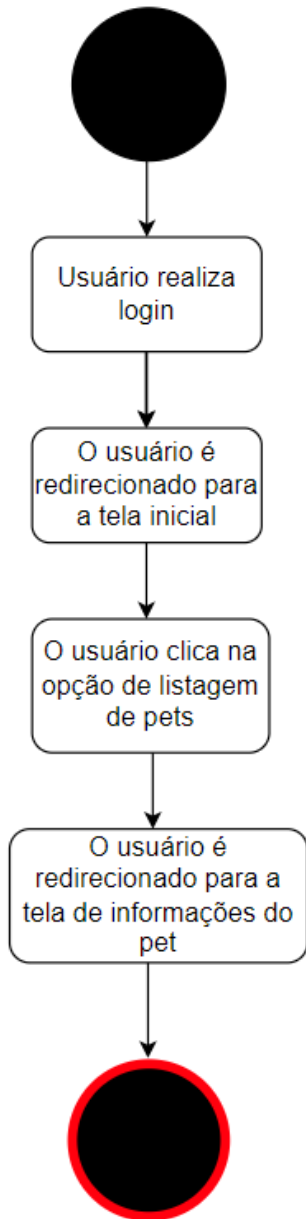
Cadastrar pet



Gerenciar pet



Visualizar pet



Adotar pet

