

UNIVERSIDADE PAULISTA

Adalberto Paulino da Costa Junior – RA G3DJB8

César Augusto Matos Moreira Marques – RA G373863

Kaique Muñoz de Souza – RA T098EG4

Nicolas da Silva Dias – RA T094151

Wilton da Luz Silva – RA G234FD8

**Desenvolvimento de uma plataforma colaborativa para a
socialização e ajuda para animais em situações de vulnerabilidade**

SANTANA DE PARNAÍBA - SP

2024

Adalberto Paulino da Costa Junior – RA G3DJB8
César Augusto Matos Moreira Marques – RA G373863
Kaique Muñoz de Souza – RA T098EG4
Nicolas da Silva Dias – RA T094151
Wilton da Luz Silva – RA G234FD8

**Desenvolvimento de uma plataforma colaborativa para a
socialização e ajuda para animais em situações de vulnerabilidade**

Trabalho de conclusão de curso para a
obtenção do título de graduação em Ciência
da Computação apresentado à Universidade
Paulista – UNIP.

Orientador Prof. Edy Hayashida

SANTANA DE PARNAÍBA – SP

2024

Ficha Catalográfica

CIP - Catalogação na Publicação

Desenvolvimento de uma plataforma colaborativa para a socialização e ajuda para animais em situações de vulnerabilidade / Kaique Munoz de Souza, Adalberto Paulino da Costa Junior, César Augusto Matos Moreira Marques, Nicolas da Silva Dias, Wilton da Luz Silva[et al.]. - 2024.
139 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto de Ciência Exatas e Tecnologia da Universidade Paulista, Santana de Parnaíba, 2024.

Área de Concentração: Tecnologia.

Orientadora: Prof.^a Me. Edy Carlos Hidemi Hayashida.

1. Plataforma colaborativa. 2. Tecnologia. 3. Animais em situação de vulnerabilidade. 4. ONG's . I. Souza, Kaique Munoz de. II. Junior, Adalberto Paulino da Costa. III. Marques, César Augusto Matos Moreira. IV. Dias, Nicolas da Silva. V. Silva, Wilton da Luz. VI. Hayashida, Edy Carlos Hidemi (orientador).

Adalberto Paulino da Costa Junior – RA G3DJB8

César Augusto Matos Moreira Marques – RA G373863

Kaique Muñoz de Souza – RA T098EG4

Nicolas da Silva Dias – RA T094151

Wilton da Luz Silva – RA G234FD8

Desenvolvimento de uma plataforma colaborativa para a socialização e ajuda para animais em situações de vulnerabilidade

Trabalho de conclusão de curso para a obtenção do título de graduação em Ciência da Computação apresentado à Universidade Paulista – UNIP.

Aprovado em:

BANCA EXAMINADORA

_____/____/____

Prof. Edy Hayashida

Universidade Paulista – UNIP

_____/____/____

Prof. Adriano Domingues

Universidade Paulista – UNIP

_____/____/____

Prof. Marcelo Castro

Universidade Paulista – UNIP

Dedicatória

Dedicamos este trabalho a todos que estiveram conosco ao longo dessa jornada acadêmica.

Aos nossos colegas de sala, que compartilharam conosco os desafios, as alegrias e o crescimento pessoal e profissional. Cada momento vivido juntos contribuiu para que chegássemos mais longe.

Aos professores da nossa universidade, que com dedicação e paciência nos guiaram no caminho do conhecimento. Eles nos inspiraram a buscar mais, a pensar criticamente e a nunca desistir.

E ao nosso orientador, Professor Edy, que, com seu apoio, orientações e incentivo constante, foi fundamental para a realização deste trabalho.

A todos, o nosso mais sincero agradecimento.

RESUMO

O aplicativo de apoio a animais em situação de vulnerabilidade social tem o intuito de abraçar esta causa social e conectar a sociedade junto com as ONGs (Organizações não Governamentais) e instituições de apoio para melhorar esta questão fazendo com que fique de mais fácil acesso localizar os animais que estão nesta situação e reportar para as instituições que irão conseguir prestar o apoio para estes animais. Inicialmente será necessário o cadastro das ONGs contendo as informações de onde estará localizada, nome e o certificado de entidade beneficente para manter um controle de registro e se mantém para os usuários que será preciso realizar um cadastro inicial e após o cadastro será criado a conta de usuário contendo e-mail senha .Para localizar estes animais e o aplicativo fará o papel de intermediar o contato entre o usuário e as ONGs que irão fazer parte do aplicativo, pois desta forma irá facilitar o trabalho social das organizações para encontrarem os animais em situação de abandono. Para incentivar a adoção de animais o aplicativo contará com uma aba específica para os usuários que queiram adotar e nesta aba terá as informações do animal é de onde ele se encontra. Para as instituições irá facilitar para a divulgação de campanhas de adoção e facilitando o sistema de doações para que as instituições consigam adquirir recursos para promover mudanças nesta questão social. O aplicativo contara com a geolocalização então a população poderá fazer parte desta ação social e conseguirão ver bons resultados nas suas ações, transformando esta corrente de caridade cada vez mais para que esta ação cresça cada vez mais. Será utilizada a framework Flutter e o Firebase para o desenvolvimento do aplicativo. O sistema está visando reduzir o número de animais em situação de rua promovendo a facilidade de acesso para a população que busca ajudar nesta questão e facilitando para as instituições para que consigam mais recursos e que possam através do aplicativo visualizar as áreas que precisam de um foco maior para resgate dos animais, por esses motivos o aplicativo se torna uma alternativa viável para quem quer contribuir para o bem-estar dos animais.

Palavras-chave: Geolocalização, Organizações Não Governamentais (ONGs), Aplicativo, Vulnerabilidade.

ABSTRACT

The application to support animals in socially vulnerable situations aims to embrace this social cause and connect society together with NGOs (Non-Governmental Organizations) and support institutions to improve this issue by making it easier to find animals. animals that are in this situation and report to the institutions that will be able to provide support for these animals. Initially it will be necessary to register the NGOs containing information on where they will be located, name and certificate of the charitable entity to maintain registration control and users will need to carry out an initial registration and after the registration the account will be created. user containing email password. To locate these animals and the application will play the role of intermediating contact between the user and the NGOs that will be part of the application, as this will facilitate the social work of organizations to find animals in situation of abandonment. To encourage the adoption of animals, the application will have a specific tab for users who want to adopt and this tab will contain information about the animal and where it is located. For institutions, it will make it easier to publicize adoption campaigns and facilitate the donation system so that institutions can acquire resources to promote changes in this social issue. The application will have geolocation so the population can be part of of this social action and will be able to see good results in their actions, transforming this chain of charity more and more so that this action grows more and more. The Flutter framework and Firebase will be used to develop the application. The system aims to reduce the number of homeless animals by promoting ease of access for the population seeking to help with this issue and making it easier for institutions to obtain more resources and, through the application, visualize the areas that need focus. for animal rescue, for these reasons the application becomes a viable alternative for anyone who wants to contribute to animal welfare

Keywords: Collaborative platforms, Non-Governmental Organizations (NGOs), Application, Vulnerability

LISTA DE ILUSTRAÇÕES

Figura 1 – Firebase Authentication

Figura 2 – Firestore Database

Figura 3 – Firebase Storage

Figura 4 - Tela de Apresentação

Figura 5 - Tela de Login

Figura 6 - 1ª Tela de Registro

Figura 7 – 2ª Tela de Registro

Figura 8 – 3ª Tela de Registro

Figura 9 - Tela Inicial

Figura 10 - Tela de Resgate

Figura 11 - Tela de Registro de Animal

Figura 12 - 2ª Parte da Tela de Registro de Animal

Figura 13 - Tela de Status do Resgate (Detalhada)

Figura 14 - Tela de Status do Resgate (Geral)

Figura 15 - Instrução do Mapa

Figura 16 - Mapa para Localização de Animais e ONG's

Figura 17 - Tela Inicial com Notificações Recebidas

Figura 18 - Tela de Notificações

Figura 19 - Tela de Animais Favoritos

Figura 20 - Tela de Pets para Adoção

Figura 21 - Opções de Filtro de Pets

Figura 22 - Tela de Adoção

Figura 23 - Opção para Agendar Visita

Figura 24 - Tela de Agendamento

Figura 25 - Notificação de Agendamento Realizado

Figura 26 – Tela de Status de Adoção

LISTA DE ABREVIATURAS e SIGLAS

ONGs – Organizações não governamentais

API - Interface de Programação de Aplicação

ADI - Ação Direta de Inconstitucionalidade

PNSB - Pesquisa Nacional de Saneamento Básico

EJA - Educação de Jovens e Adultos

MST - Movimento dos Trabalhadores Rurais Sem Terra

OSCs - Organizações da Sociedade Civil

IDEs - Ambientes de Desenvolvimento Integrado

STF - Supremo Tribunal Federal

ADI - Ação Direta de Inconstitucionalidade

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Problematização / Motivação	13
1.2 Objetivo Geral	14
1.3 Objetivo Específico.....	15
1.4 Metodologia.....	16
1.5 Organização do Texto	17
2 REVISÃO DA LITERATURA	18
2.1 Abandono de Animais no Brasil.....	18
2.2 Projetos Sociais no Brasil.....	22
2.3 ONGS no Brasil: Situação Atual e Desafios	25
2.4 Plataformas Colaborativas.....	29
2.5 Impacto Social das Plataformas Colaborativas	31
2.6 Tecnologias para desenvolvimento de aplicativos Android	33
2.7 APIs de Localização	35
2.8 Sistemas Distribuídos	37
2.8.1 Modelo Cliente Servidor.....	37
2.8.2 CRUD	37
3 MATERIAL E MÉTODOS	38
3.1 Material	38
3.1.1 Hardware	38
3.1.2 Framework.....	38
3.1.3 Ambiente.....	39
3.1.4 Back-End	39
3.1.5 Kotlin.....	39
3.2 Métodos	39
3.2.1 Cliente-Servidor	39
3.2.2 CRUD	44
3.2.2.1 Read (Leitura).....	45
3.2.2.2 Update (Atualização)	45
3.2.2.3 Remoção de Animais Favoritos	45

3.2.2.4 Exclusão de Registros de Resgates	46
3.2.3 Flutter.....	47
3.2.4 Dart.....	48
3.2.5 Códigos.....	Erro! Indicador não definido. 9
4 RESULTADOS	127
5 CONCLUSÃO.....	1500
REFERÊNCIAS BIBLIOGRÁFICAS	1511

1 INTRODUÇÃO

O presente projeto tem como objetivo desenvolver um aplicativo de apoio social destinado a animais em situação de vulnerabilidade, visando resgatá-los dessa condição e proporcionando tratamento adequado e uma moradia digna. O foco principal será o acolhimento de animais em situação de rua, funcionando também como uma ferramenta facilitadora para pessoas que encontram animais abandonados e não sabem a quem recorrer.

O aplicativo será direcionado para Organizações Não Governamentais (ONGs) voltadas à proteção animal. Essas ONGs poderão se cadastrar na plataforma, preenchendo seus dados institucionais e, a partir desse cadastro, terão acesso às localizações de animais em situação de vulnerabilidade. Isso permitirá que prestem o apoio necessário, seja por meio de resgates, abrigo ou cuidados veterinários.

O aplicativo contará com a integração de uma API do Google Maps, a qual permitirá o mapeamento das áreas com maior concentração de animais vulneráveis. Assim, as ONGs cadastradas terão acesso a essas informações e os usuários que transitarem por áreas mapeadas poderão, por meio do aplicativo, solicitar o resgate de animais em situação de risco.

Além disso, os usuários do aplicativo terão a possibilidade de verificar a autenticidade das ONGs cadastradas no sistema. Informações como dados cadastrais, localização e o responsável legal de cada organização estarão disponíveis, garantindo maior transparência e confiabilidade na plataforma.

1.1 Problematização / Motivação

O índice de abandono de animais no Brasil é alarmante. De acordo com um estudo realizado pela Organização Mundial de Saúde, estima-se que existam 30 milhões de cães e gatos abandonados no país. As causas desse abandono são diversas, incluindo as dificuldades financeiras enfrentadas pelos tutores e a perda de seus animais de estimação (REGINA, 2022).

Além disso, desastres naturais são um fator significativo que agrava essa situação. Um exemplo recente foram as enchentes ocorridas no estado do Rio Grande do Sul, que resultaram no abandono de aproximadamente 19 mil cães e gatos. A destruição causada pelas fortes chuvas fez com que muitas famílias perdessem suas residências, levando, conseqüentemente, ao abandono de seus animais de companhia. (DUTRA, 2024)

Atualmente, a internet proporciona amplo acesso à informação, e com o avanço da globalização, torna-se imprescindível a criação de um aplicativo e um site dedicados à problemática do abandono de animais. Essas ferramentas desempenham um papel crucial na disseminação de informações para pessoas interessadas em adotar, cuidar ou resgatar animais em situação de vulnerabilidade.

Diante dessa problemática social, o desenvolvimento de um aplicativo contribuirá de forma significativa para a redução do número de animais abandonados. Ele promoverá a interação entre instituições de caridade e a população, oferecendo uma plataforma que facilite a comunicação e a adoção. Além disso, espera-se que os usuários do aplicativo se sintam realizados ao realizar ações que promovam o bem-estar e o cuidado com esses animais.

1.2 Objetivo Geral

O presente projeto tem como objetivo desenvolver um aplicativo que possibilite a interação entre usuários e Organizações Não Governamentais (ONGs) ou institutos de caridade, cuja finalidade seja oferecer auxílio a animais em situação de vulnerabilidade social. O intuito é promover a conscientização acerca da proteção animal, incentivando a participação de pessoas interessadas em contribuir para esse segmento específico.

O aplicativo contará com uma ferramenta de geolocalização que beneficiará tanto os usuários quanto as instituições envolvidas. Com essa funcionalidade, será possível identificar regiões com maior concentração de animais em situação de rua, facilitando o trabalho das ONGs e instituições de caridade, que poderão localizar esses animais de forma mais ágil e prestar o auxílio necessário.

Adicionalmente, o projeto incluirá a criação de um site, no qual os usuários poderão acompanhar os resultados obtidos por meio do aplicativo, ler depoimentos de outros usuários e conhecer o trabalho realizado com os animais encontrados através da plataforma. Essa interface proporcionará maior segurança para aqueles que desejam contribuir socialmente, além de fortalecer a rede de solidariedade, beneficiando diretamente os animais e conferindo maior credibilidade às instituições participantes dessa iniciativa social. O presente projeto tem como objetivo desenvolver um aplicativo que possibilite a interação entre usuários e Organizações Não Governamentais (ONGs) ou institutos de caridade, cuja finalidade seja oferecer auxílio a animais em situação de vulnerabilidade social. O intuito é promover a

conscientização acerca da proteção animal, incentivando a participação de pessoas interessadas em contribuir para esse segmento específico.

O aplicativo contará com uma ferramenta de geolocalização que beneficiará tanto os usuários quanto as instituições envolvidas. Com essa funcionalidade, será possível identificar regiões com maior concentração de animais em situação de rua, facilitando o trabalho das ONGs e instituições de caridade, que poderão localizar esses animais de forma mais ágil e prestar o auxílio necessário.

Adicionalmente, o projeto incluirá a criação de um site, no qual os usuários poderão acompanhar os resultados obtidos por meio do aplicativo, ler depoimentos de outros usuários e conhecer o trabalho realizado com os animais encontrados através da plataforma. Essa interface proporcionará maior segurança para aqueles que desejam contribuir socialmente, além de fortalecer a rede de solidariedade, beneficiando diretamente os animais e conferindo maior credibilidade às instituições participantes dessa iniciativa social.

1.3 Objetivo Específico

O presente projeto tem como objetivo principal apoiar animais em situação de vulnerabilidade social. Para tanto, será desenvolvida uma ferramenta com a funcionalidade de geolocalização, utilizando a API do Google Maps, que permitirá ao usuário localizar e reportar animais em situação de rua em sua região. Essa funcionalidade visa facilitar o resgate dos animais pelas instituições de apoio.

Além disso, o aplicativo oferecerá informações sobre as instituições parceiras para aqueles que desejam contribuir financeiramente com a causa. Serão fornecidos dados cadastrais, localização das instituições e instruções sobre como realizar doações, seja de suprimentos ou em forma de contribuição financeira.

Dada a relevância social da causa animal, o aplicativo também disponibilizará conteúdos educativos relacionados aos cuidados com os animais, à importância da adoção, e a outros temas pertinentes. O intuito é sensibilizar os usuários e reforçar a proposta central da plataforma. O aplicativo também facilitará o voluntariado, oferecendo uma rede de apoio para usuários que desejem atuar como voluntários em instituições necessitadas. Essa rede permitirá que os interessados participem de eventos ou atividades que demandem mão de obra, ampliando o engajamento social.

Por fim, será implementado um sistema de avaliação, onde usuários que adotaram algum animal poderão compartilhar suas experiências e opiniões. Esse

recurso auxiliará outros usuários, além de contribuir para a melhoria contínua dos serviços prestados pelas ONGs e abrigos, agregando maior credibilidade às instituições.

1.4 Metodologia

A presente pesquisa tem como objetivo esclarecer as dúvidas dos leitores acerca do tema social a ser abordado, enfatizando a utilidade do aplicativo proposto para pessoas interessadas em auxiliar na questão social. Em um projeto, existem diferentes formas de conduzir uma pesquisa, sendo expostas as principais metodologias e indicando quais serão empregadas neste trabalho.

Adotar-se-á um procedimento metodológico de abordagem qualitativa, denominado pesquisa exploratória, cuja finalidade é a elaboração de um instrumento de pesquisa adequado à realidade. A pesquisa exploratória visa detalhar completamente o assunto, buscando identificar o significado e as definições pertinentes ao tema tratado, de modo a familiarizar o leitor com os termos utilizados no desenvolvimento da pesquisa. (PIOVESAN, 1995)

A pesquisa descritiva tem por objetivo descrever um fenômeno ou situação com precisão, abrangendo de maneira clara as características de um indivíduo, grupo ou situação. Sua função é explicar qualquer circunstância presente no texto ou no projeto, garantindo ao leitor uma compreensão clara do que está ocorrendo. (SOUZA, 2016)

Além disso, outra metodologia complementar que será utilizada, segundo Tolfo (2009): “uma pesquisa explicativa pode ser a continuação de outra pesquisa descritiva, visto que a identificação de fatores que determinam um fenômeno exige que este esteja suficientemente descrito e detalhado”. Conforme mencionado, a pesquisa explicativa complementa a descritiva, garantindo coerência e completude à investigação. (TOLFO, 2009)

Neste projeto, será utilizada a pesquisa descritiva para delinear como o projeto será executado e quais ferramentas serão empregadas. Também será aplicada a pesquisa explicativa, com o intuito de esclarecer como serão realizados os processos de desenvolvimento da plataforma e de que forma esta contribuirá para a causa social, que é o principal objetivo do aplicativo.

Por fim, a pesquisa exploratória também desempenhará um papel importante no projeto, pois será necessária para o detalhamento dos softwares a serem utilizados no desenvolvimento. Essa metodologia permitirá que o leitor compreenda como as ferramentas escolhidas contribuirão para o resultado do projeto.

1.5 Organização do Texto

A presente dissertação será desenvolvida de maneira explicativa e objetiva, abordando o funcionamento do projeto proposto. Na introdução, descreve-se o problema relacionado aos animais em situação de vulnerabilidade social, ressaltando a importância desse tema, o qual constitui o foco central do desenvolvimento do aplicativo.

No capítulo 1 são apresentados os objetivos do trabalho, destacando o foco social no abandono de animais em situação de vulnerabilidade. O aplicativo, que será criado como solução para esse problema, é apresentado de forma sucinta, explicando como funcionará e o que será utilizado para seu desenvolvimento.

No capítulo 2 cada tema será detalhado. O uso das tecnologias no desenvolvimento de aplicativos será explorado para esclarecer ao leitor que o projeto contará com softwares específicos, evidenciando a função apropriada de cada um. Também será explicado o papel social das organizações não governamentais (ONGs), destacando sua importância no contexto do aplicativo. Por fim, será discutido o motivo do abandono de animais e como isso influenciou a criação do projeto.

No capítulo 3 "desenvolvimento e métodos" abordará o planejamento do projeto e os requisitos para sua produção. Um ponto importante é o público-alvo do aplicativo: as ONGs e os usuários que serão cadastrados para sua utilização. Outro aspecto fundamental é o funcionamento do aplicativo para o cadastro de animais, utilizando a API integrada ao Google Maps para a localização. A comunicação entre os usuários e as ONGs também será discutida, explicando detalhadamente como irá funcionar e como o desenvolvimento do código foi feito.

No capítulo 4 irá ser abordado sobre a produção visual do aplicativo sendo colocado e explicado parte a parte sobre as telas do aplicativo e o que cada uma delas representa para que o leitor entenda de como funciona o aplicativo na prática.

No capítulo 5 após o projeto ser organizado por um todo mantendo uma lógica para que fique mais claro para o leitor iremos apresentar a conclusão, finalizando e explicando a serventia do projeto para a causa social que vem sendo abordada e colocando em pauta a importância da tecnologia para o auxílio de causas que necessitam de apoio.

No capítulo 6 será apresentado as fontes de pesquisas que utilizamos para que comprove o embasamento das teses que colocamos em pauta no desenvolvimento do nosso trabalho, assim comprovando o que está sendo escrito no projeto.

A organização textual do trabalho é essencial para que as informações sejam apresentadas de maneira clara e objetiva. Um projeto bem estruturado, com contextualização adequada e revisão criteriosa, garante que as informações sejam compreendidas de forma eficiente pelo leitor. Assim, a clareza e a coesão textual são fatores determinantes para o sucesso do projeto.

2 REVISÃO DA LITERATURA

A revisão da literatura tem como objetivo explorar temas cruciais para o desenvolvimento deste projeto, que visa criar uma plataforma colaborativa para auxiliar animais em situação de vulnerabilidade no Brasil. Primeiramente, serão abordados os contextos de abandono de animais domésticos no país, analisando suas causas e consequências, além de projetos sociais e o papel das ONGs no combate a esse problema. Em seguida, será discutido o uso de plataformas colaborativas como uma solução inovadora, examinando seu impacto social e sua aplicação em diversas áreas. A tecnologia Android, amplamente utilizada no desenvolvimento de aplicativos móveis, será apresentada como a base para a criação da plataforma, com destaque para a integração de APIs de localização, como a do Google Maps, oferecem funcionalidades essenciais para o mapeamento e geolocalização de animais e instituições.

2.1 Abandono de Animais no Brasil

No Brasil, possuir um animal de estimação, seja ele um cão ou um gato, é algo habitual e apreciado por muitas famílias. Numerosos desses pets são adotados desde filhotes, convivendo lado a lado com seus cuidadores e se tornando um componente indispensável da residência. Eles recebem afeto, atenção especial, como vestimentas para o frio, uma alimentação balanceada e consultas frequentes ao veterinário. Esta conexão afetuosa e íntima é bastante comum, porém, infelizmente, contrasta com um problema que se torna cada vez mais evidente: o abandono de animais de estimação (HUERB, 2018).

Existem várias razões para o abandono de animais de estimação, sendo o descuido e a falta de organização de alguns cuidadores uma das mais frequentes. Muitas pessoas optam por adotar um animal sem ponderar profundamente sobre os

cuidados diários necessários e o espaço apropriado para que ele possa viver confortavelmente. Com o passar do tempo, ao se depararem com a responsabilidade e a dedicação requeridos, algumas se desanimam e, infelizmente, escolhem o desamparo. Este gesto evidencia a relevância de uma sensibilização acerca da adoção responsável e a demanda por suporte para esses animais, que frequentemente acabam nas ruas sem um lar. (FRIZZO, 2016).

O nascimento de filhotes também é um fator relevante no abandono de animais. Quando uma fêmea dá à luz, o ideal seria encontrar famílias que cuidem de cada filhote, porém, frequentemente, as famílias encontram obstáculos para encontrar quem os adote. Infelizmente, devido à falta de tempo e condição financeira para cuidar de todos, alguns tutores acabam escolhendo o abandono, o que não apenas impacta a vida dos animais, mas também contribui para o crescimento do número de animais em situação de rua. Esta situação enfatiza a importância da sensibilização e do suporte para incentivar a adoção e o cuidado responsável. (KAROLINA, 2019)

Além disso, segundo Karolina (2019): "a compra de animais por impulso, seja para si mesmo ou como presente, é uma causa importante de abandono". Oferecer um animal de estimação a alguém requer muita sensibilidade e ponderação, pois implica uma grande responsabilidade. É crucial verificar se o indivíduo possui as condições financeiras, o tempo e o comprometimento necessários para cuidar adequadamente do animal. Sem essas condições, o animal pode ser vítima de negligência ou até mesmo de abandono. Com uma análise minuciosa antes da decisão, é possível prevenir situações como essas, assegurando que o animal seja acolhido com carinho em um ambiente familiar preparado e acolhedor. (KAROLINA, 2019)

Uma outra razão frequente para o abandono está relacionada ao comportamento dos animais. Alguns tutores interpretam o comportamento do animal como "problemático" e, ao invés de procurar auxílio ou soluções para o problema, acabam abandonando-o. Segundo o Conselho Regional de Medicina Veterinária de São Paulo, "um animal que se comporta de uma maneira específica pode ser visto como problemático para uma família, porém, esse mesmo animal pode ser facilmente aceito em outra comunidade." Conforme o animal se desenvolve e requer mais cuidado, atenção e espaço, alguns proprietários podem perder o interesse e, infelizmente, escolher o abandono. Esta circunstância nos recorda a relevância de

compreender as necessidades de cada animal e procurar soluções, ao invés de se render aos obstáculos. (Maria Emília Yamamoto; 2024)

A crescente quantidade de animais abandonados nas vias públicas provoca outro problema: a multiplicação descontrolada. Isso intensifica o problema, uma vez que se observa um crescimento contínuo de animais em situações de vulnerabilidade. Segundo David (2022), "a alta taxa de natalidade de cães e gatos abandonados é uma demanda social e ambiental urgente, visto que a acumulação desses animais nas ruas traz problemas para a gestão pública." (DAVID, 2022)

Tanto os animais quanto a sociedade são afetados pelo abandono de animais. Os animais abandonados podem constituir um perigo para a saúde pública, devido à disseminação de zoonoses (doenças que são transmitidas de animais para seres humanos), além de provocar desconforto social e ecológico. Conforme relatado por Silva: "o abandono de animais é considerado uma ameaça potencial nas áreas de saúde pública, social, ecológica e econômica."(SILVA, 2024)

Esses animais, que não possuem um habitat apropriado, frequentemente habitam áreas comerciais ou residenciais, gerando insegurança entre aqueles que não estão habituados a coexistir com eles. Isso pode até afetar adversamente o comércio em regiões com grande quantidade de animais sem lar.

Além disso, os animais de rua poluem o ambiente, segundo Karolina” ficam expostos a doenças, seus dejetos são deixados em qualquer lugar, contaminam solo, água, trazendo muitas zoonoses para população”. A existência de animais abandonados nas vias públicas tem graves impactos sobre o meio ambiente e a segurança pública. Os resíduos e a falta de higiene contribuem para questões ambientais, enquanto a sua presença nas estradas eleva o perigo de colisões. Esses acidentes, além de colocar a vida dos animais em risco, podem provocar problemas no trânsito e, infelizmente, deixar cadáveres à mostra, tornando o ambiente insalubre e perigoso para todos. (KAROLINA, 2019)

A questão humanitária também é relevante, pois muitos desses animais morrem desnecessariamente, seja por atropelamento ou doenças que poderiam ser tratadas. Segundo Bortoloti (2007): falta na referência bibliográfica "muitos animais vão a óbito desnecessariamente devido a atropelamentos sem socorro ou doenças tratáveis que, sem os devidos cuidados veterinários, tornam-se perigosas." (BORTOLOTI, 2007)

Do ponto de vista da saúde pública, o abandono de animais representa perigos consideráveis. Sem o devido cuidado e atenção, os animais abandonados podem se transformar em propagadores de enfermidades como cólera, toxoplasmose, giardíase, sarna, entre outras zoonoses que também impactam os humanos. Esta circunstância destaca a importância de medidas preventivas e sensibilização acerca do efeito do abandono, que ultrapassa as dores dos animais e ameaça a saúde de toda a comunidade (MYLENA, 2023).

Ao refletirmos sobre a saúde pública, notamos que a ausência de supervisão e atenção aos animais abandonados nas ruas pode resultar em graves impactos para a população. Esses animais têm a capacidade de propagar enfermidades que, se não forem detectadas e tratadas rapidamente, podem acarretar sérias consequências para a saúde humana. Sem o diagnóstico apropriado e a utilização de medicamentos apropriados, vírus e bactérias disseminados por animais podem se disseminar e constituir um risco concreto. Isso enfatiza a necessidade de tratarmos o abandono com seriedade e de colaborarmos na prevenção desses perigos.

A raiva é uma doença que afeta principalmente caninos. Apesar da diminuição dos casos entre animais domésticos devido às campanhas de vacinação, a situação é diferente para os animais abandonados, onde não há o mesmo controle. Isso representa um risco para a população, pois animais de rua com raiva podem se tornar agressivos e atacar pessoas, transmitindo a doença. (GONÇALVES, 2017)

Se um animal infectado morde um indivíduo, a raiva pode ser transmitida, resultando em graves problemas para o sistema imunológico e outros graves impactos na saúde de quem foi exposto. Este perigo destaca a necessidade de um controle mais eficiente da população de animais abandonados, crucial para evitar a disseminação desta e de outras enfermidades.

A questão econômica também é importante. O abandono de animais gera custos com a captura, tratamento e cuidado desses animais. Segundo a citação já utilizada na contextualização e problematização temos um número alto de animais abandonados no Brasil. Esse número elevado demanda recursos financeiros para campanhas de vacinação, controle populacional e tratamento de animais feridos ou doentes. (CEZAR, 2020)

É crucial ponderar sobre os recursos destinados ao cuidado desses animais, já que assegurar uma vida digna implica em campanhas de imunização que evitam enfermidades e salvaguardam tanto os animais quanto a comunidade. Ademais, os

frequentes incidentes de trânsito que envolvem animais abandonados requerem assistência veterinária especializada. Se essa questão não for tratada com a seriedade necessária, as despesas para o país podem ser elevadas, tanto no tratamento desses animais quanto na gestão dos problemas resultantes, destacando a necessidade de uma atenção meticulosa a este assunto.

Em 2018 havia no Brasil 139,3 milhões desses animais. Destes 54,2 milhões de cães, 39,8 milhões de aves, 23,9 milhões de gatos, 19,1 milhões de peixes e 2,3 milhões de outras espécies (PAIVA, 2021).

Portanto, é crucial debatermos a relevância de políticas governamentais que se referem ao bem-estar desses animais e que combatam a crueldade. É necessário compreender melhor quais medidas legais devem ser implementadas em situações de abuso ou negligência, e verificar se esse assunto é devidamente abordado na legislação brasileira. Essas ações são essenciais para formar uma sociedade mais consciente e engajada na proteção dos animais.

A legislação brasileira tem avançado nesse sentido. O Supremo Tribunal Federal (STF), ao julgar a Ação Direta de Inconstitucionalidade (ADI) nº 4.983, afirmou que a vedação de crueldade contra os animais, segundo Santos é "um valor eminentemente moral que o constituinte conferiu ao propô-la em benefício dos animais sencientes". Esses avanços legais são essenciais para garantir que a violência contra os animais seja levada a sério e que os responsáveis sejam punidos adequadamente. (PAIVA, 2021)

É necessário um esforço conjunto de toda a sociedade para combater o abandono de animais. É crucial estabelecer mais delegacias especializadas em proteção animal e aplicar leis mais severas para assegurar o respeito aos direitos dos animais e a conscientização das pessoas sobre a responsabilidade que temos com esses seres. É crucial implementar políticas públicas efetivas para criar uma sociedade mais empática e dedicada ao bem-estar animal, fomentando um futuro mais equitativo para todos. (BARCELOS, 2023.)

2.2 Projetos Sociais no Brasil

Os projetos sociais sempre tiveram uma função crucial, com finalidades e formatos variados, frequentemente incentivados por empresas, pelo governo ou pela própria sociedade civil. Eles proporcionam vantagens para todos, particularmente para os mais vulneráveis, gerando um efeito positivo tanto para quem é auxiliado quanto para os voluntários que participam. O ato de se oferecer como voluntário é

extremamente recompensador, já que estender a mão a quem necessita é uma autêntica demonstração de empatia, solidariedade e humanidade. (COELHO, 2011)

Ao ponderarmos sobre a situação social no Brasil, é impossível ignorar a desigualdade que se evidencia em diversos aspectos do dia a dia. A diferença entre as regiões de baixa e alta renda é evidente. Embora as famílias das classes médias e altas geralmente residam em casas ou apartamentos em condomínios fechados, com completa infraestrutura de lazer e segurança, as famílias de menor renda lidam com circunstâncias bastante distintas. Elas residem em regiões com grande número de habitações apertadas, frequentemente erguidas em terrenos acidentados, sem a infraestrutura e segurança adequadas, tornando essas comunidades ainda mais suscetíveis a catástrofes e desafios diários.

Outro aspecto importante da desigualdade social é a questão alimentar. Não se trata apenas de racionamento de alimentos, mas de uma parcela significativa da população não ter o que comer. Em 2023, cerca de 27,6% (ou 21,6 milhões) dos domicílios brasileiros estavam em situação de insegurança alimentar. Este dado destaca o desafio que muitos brasileiros enfrentam, destacando a relevância de medidas que busquem auxiliar o próximo. (FERREIRA, 2023)

O saneamento básico no Brasil também é um problema estrutural. Desde o início da urbanização desordenada, com a favelização, o saneamento nunca foi tratado como prioridade para as populações mais vulneráveis. De acordo com a Pesquisa Nacional de Saneamento Básico (PNSB) de 2020, cerca de 39,7% dos municípios brasileiros ainda não possuem serviços de esgotamento sanitário. Esse dado revela a necessidade urgente de projetos específicos para mitigar esse problema. (CABRAL, 2020)

No âmbito educacional, as discrepâncias entre escolas públicas e privadas são particularmente notáveis. Conforme informações do Inep, as inscrições na Educação de Jovens e Adultos (EJA) têm apresentado uma queda desde 2018. No ano de 2023, foram registrados 2,5 milhões de alunos, sendo 2,3 milhões na rede pública e aproximadamente 200 mil na privada. Estes dados indicam que a maioria dos estudantes que procuram uma chance de educação por meio da EJA estuda em escolas públicas, que muitas vezes lidam com obstáculos como orçamentos limitados, materiais impróprios e infraestrutura precária. Esta situação prejudica a qualidade da educação, prejudicando o crescimento desses jovens e adultos e restringindo suas

chances de se prepararem adequadamente para o mercado de trabalho ou para testes como o vestibular. (INEP, 2024)

Em face dos vários desafios estruturais que o país enfrenta, é crucial recorrer à força de projetos sociais que fomentem o crescimento pessoal e comunitário. Estes projetos precisam se desenvolver e se espalhar, atingindo a maior quantidade possível de comunidades, sempre levando em conta as necessidades únicas e específicas de cada uma, a fim de que possam causar um efeito concreto e positivo na vida das pessoas.

Em face dos desafios socioeconômicos que o Brasil vivencia, diversos projetos sociais têm sido desenvolvidos com a finalidade de fomentar o desenvolvimento em regiões que necessitam de investimento e infraestrutura adequada. Estes projetos têm como objetivo fornecer soluções para os setores mais frágeis, proporcionando suporte e oportunidades para comunidades carentes. Em seguida, serão mostrados alguns exemplos desses projetos que impactam positivamente a sociedade.

Em face dos desafios socioeconômicos que o Brasil enfrenta, vários projetos sociais têm sido desenvolvidos para auxiliar áreas que necessitam de investimento e infraestrutura. Por exemplo, muitas ONGs se empenham em promover a educação escolar para a população, particularmente para as comunidades mais desfavorecidas. O Instituto Ayrton Senna, por exemplo, se empenha em fornecer soluções que assegurem um ensino de alto padrão nas escolas públicas, contribuindo para a mudança da realidade de inúmeras crianças e adolescentes. (MARINHO, 2018)

Também existem iniciativas significativas na saúde, como campanhas de vacinação e consultas sem custo, que são cruciais para o progresso da saúde pública no país. Ações como campanhas de imunização e consultas sem custo são essenciais para promover a saúde no país, proporcionando aos mais necessitados a chance de cuidar da sua saúde e prevenir enfermidades. Essas medidas são fundamentais para aprimorar a qualidade de vida de muitos indivíduos e reforçar o sistema de saúde em sua totalidade.

Por outro lado, a questão da habitação é uma questão urgente, particularmente devido ao aumento das favelas. Um caso notável de iniciativa focada na habitação é o Movimento dos Trabalhadores Rurais Sem Terra (MST). Apesar de ter provocado polêmicas no começo, o MST se estabeleceu como um dos principais apoiadores da batalha pela terra no Brasil. Atualmente, atua em 24 estados e auxiliou

aproximadamente 450 mil famílias a adquirir terras através da organização e mobilização dos trabalhadores do campo. (GOHN, 2000).

Esses projetos sociais, independentemente de suas áreas de atuação, têm em comum o objetivo de ajudar as comunidades mais vulneráveis, promovendo a transformação social e o bem-estar da população.

2.3 ONGS no Brasil: Situação Atual e Desafios

As Organizações Não Governamentais (ONGs) têm um papel crucial no progresso social do Brasil, dando destaque a comunidades, indivíduos e iniciativas que anteriormente eram menos valorizados. Um caso inspirador é o efeito das organizações não governamentais no turismo em territórios indígenas. Essas entidades contribuíram para a geração de empregos para os habitantes locais, formação de guias turísticos, diminuição da migração para as áreas urbanas e, simultaneamente, a conservação do patrimônio cultural e ambiental das comunidades. O suporte dessas organizações não governamentais foi crucial para que este processo ocorresse de maneira sustentável e respeitosa. (CORBARI, 2017).

Essas organizações não governamentais são essenciais para o desenvolvimento cultural e econômico do Brasil, uma vez que exercem uma função vital no suporte à população em situação de vulnerabilidade social. Elas auxiliam na transformação de desafios em chances de superação, proporcionando recursos e suporte para que essas comunidades possam evoluir e alcançar um futuro mais promissor. A atuação dessas entidades reforça a inclusão e abre possibilidades para uma vida mais digna e repleta de oportunidades.

A formação das ONGs foi influenciada por várias circunstâncias históricas, particularmente após a Segunda Guerra Mundial, período em que muitas pessoas necessitavam de suporte e recursos. De acordo com Alberto (2001): “devemos admitir que os anos de 1970 marcam a emergência de um novo sentido que se agrega ao conceito de ONG, enfatizando valores e conceitos como desenvolvimento social, cidadania e sociedade civil.” A partir desse período, as ONGs passaram a ter um papel mais significativo no desenvolvimento social das áreas necessitadas, conquistando o reconhecimento da população. (ALBERTO, 2001)

Como se pode notar, a partir dos anos 70, as entidades começaram a adotar um enfoque distinto, direcionando-se para o progresso social das regiões que mais necessitavam de auxílio. Esta nova orientação fez com que a população

reconhecesse a relevância e o valor dessas ONGs, que passaram a ser consideradas essenciais para impulsionar transformações significativas nas comunidades e assegurar melhores condições de vida para os mais vulneráveis.

No Brasil, as ONGs têm um papel fundamental no panorama social, particularmente em períodos de mudança política, como durante a ditadura e na instauração da democracia. Essas entidades foram cruciais para promover a proteção dos direitos humanos, auxiliar os mais vulneráveis e auxiliar na formação de uma nação mais equitativa e democrática, exercendo um papel crucial em períodos de grandes transformações políticas e sociais.

Segundo Rocha (2005) “A história da democracia brasileira é recente, marcada por um regime militar entre 1964 e 1985, além de outros períodos autoritários, como o Estado Novo de Getúlio Vargas”. Essas transições estimularam o crescimento das ONGs, em parte devido à deterioração da qualidade de vida e da infraestrutura nas cidades brasileiras, o que gerou uma demanda por apoio e investimentos sociais. (ROCHA, 2005)

Passado por um desenvolvimento nas políticas públicas do país o que auxiliou para a movimentação das organizações do país, de acordo com Rocha (2005): “Outro fator que estimulou o crescimento de ONGs no Brasil foi a rápida e crescente deterioração da qualidade de vida e da infraestrutura das cidades brasileiras.” Com a evolução política que o Brasil teve acabou acontecendo o fato de muitas regiões ficarem carenciadas de investimentos para a infraestrutura e bem-estar da população que habita nos locais. (ROCHA, 2005)

Assim, as entidades não governamentais se expandiram consideravelmente para fornecer suporte e assistência às comunidades que mais precisam. Este aumento ocorre devido à escassez de investimentos públicos em regiões desfavorecidas e à falta de fundos governamentais, o que impulsiona as ONGs a agir para preencher essas lacunas. Elas têm um papel crucial em aprimorar a qualidade de vida da população, proporcionando recursos e serviços que são cruciais para o bem-estar de muitos indivíduos.

No Brasil, a questão ambiental é crucial, já que o país possui várias reservas naturais e extensas florestas espalhadas por seu território. Portanto, é essencial uma supervisão eficiente do uso dessas áreas, além de um suporte social para a sua conservação. Nesse cenário, as entidades não governamentais tiveram um papel fundamental, particularmente durante a ECO-92 no Rio de Janeiro, um acontecimento

que conectou ONGs e a administração ambiental do país. Este instante foi significativo, pois intensificou a sensibilização ambiental e solidificou a função das ONGs na salvaguarda dos nossos recursos naturais. (WALDMAN, 2007)

Inicialmente, as entidades não governamentais encontraram muitos obstáculos para obter reconhecimento em nosso país. Para vencer esses obstáculos, elas entenderam a importância de se unirem em busca de maior exposição e reconhecimento. Assim, em 1991, nasceu a ABONG, uma entidade relevante que tem desempenhado um papel crucial ao dar voz e reforçar a atuação das ONGs no Brasil que de acordo com (PAES, 1998) teve o objetivo de “expressar a identidade de um conjunto de ONGs envolvidas no processo de democratização e tradicionalmente comprometidas com os setores populares.” (PAES, 1998)

A sustentabilidade financeira é um dos maiores desafios enfrentados pelas ONGs, que muitas vezes dependem de doações e trabalho voluntário. De acordo com Jorge Gonzaga de Oliveira, da ONG Aheda:

Um dos maiores problemas enfrentados pelas ONGs é a sustentabilidade. Em encontro realizado em Campinas para debater essa questão, Jorge Gonzaga de Oliveira, da ONG Aheda, disse que "as dificuldades para captar recursos estão maiores: é preciso aprender a elaborar bons projetos e parcerias, conhecer os incentivos fiscais e empresas interessadas em financiar projetos sociais". (GOUVEIA, 2007)

Como é evidente, para manter as organizações sociais em funcionamento e eficientes, é crucial ter investimentos ou patrocínios, uma vez que se referem a causas sociais e não a ações privadas. Uma das maiores barreiras que essas entidades enfrentam é a falta de recursos, o que compromete a continuidade de suas atividades e a ampliação de suas iniciativas em prol da comunidade.

Frequentemente, os projetos sociais conduzidos por organizações não governamentais não obtêm a atenção necessária para atingir efetivamente o público pretendido. Como já mencionamos, a ausência de visibilidade limita a capacidade das ONGs de atrair investidores interessados em apoiar suas causas sociais, o que pode comprometer a expansão do efeito positivo que essas entidades poderiam causar na sociedade.

A maioria das ONGs no Brasil não conta com recursos governamentais, dependendo de outras formas de financiamento. Segundo Moura (2006) “apenas 21%

recebem recurso público, ao passo que 46% trabalham com a geração de recurso próprio e 33% contam com investimento privado.” (MOURA, 2006)

A maior parte das entidades não recebe financiamento do governo, tornando-se vulneráveis a outras formas de financiamento, como doações e ações de angariação de fundos. Esses sistemas de suporte são essenciais para assegurar que as ONGs possam prosseguir com suas atividades, pois frequentemente lidam com problemas financeiros que colocam em risco a continuidade de seus projetos e a realização de suas missões sociais.

Gradualmente, a pandemia de COVID-19 evoluiu para um acontecimento que afetou intensamente todas as áreas do Brasil, com um efeito particularmente severo na economia. Numerosas pessoas perderam seus postos de trabalho devido à rápida propagação do vírus, forçando o governo a adotar ações como o confinamento e outras limitações para assegurar a permanência da população em suas residências. Essas medidas levaram à suspensão de diversas atividades presenciais de empresas, intensificando a crise econômica no país. Isso também impactou a operação das ONGs, que passaram a lidar com desafios financeiros e operacionais maiores.

O setor das ONGs foi particularmente afetado. De acordo com Peixoto (2021) “67% das Organizações da Sociedade Civil (OSCs) tiveram sua arrecadação reduzida em 50% durante a pandemia.” Isso reforçou a necessidade de uma gestão eficiente para que as ONGs possam se manter ativas e continuar cumprindo seu papel social. (PEIXOTO, 2021)

O que se pode analisar é que as organizações não governamentais passam por diversas dificuldades e para se manter em equilíbrio e ativas é necessária uma boa gestão para o crescimento e visibilidade da organização pois segundo Katsurayama (2008): “é importante que o líder de uma organização não-governamental sem fins lucrativos tenha conhecimento e noções de Administração, podendo aproveitar adequadamente os recursos disponíveis e agir pensando no futuro”. (KATSURAYAMA, 2008)

Em suma, as ONGs têm um papel crucial no Brasil, contribuindo para superar desafios sociais e fomentando a inclusão dos mais necessitados. Para garantir a continuidade do impacto positivo dessas organizações nas comunidades desfavorecidas, é crucial o suporte de todos e a garantia de sua sustentabilidade, a fim de continuar transformando vidas e gerando oportunidades para os mais vulneráveis.

2.4 Plataformas Colaborativas

As plataformas colaborativas são recursos digitais desenvolvidos para promover a interação entre organizações e indivíduos, possibilitando a união para atingir as metas de um projeto ou atividade. Estes sistemas são criados para permitir a partilha eficaz de conhecimentos e recursos, ultrapassando obstáculos geográficos e assegurando que todos os participantes possam colaborar em tempo real.

Vários sistemas colaborativos, tais como redes sociais e aplicativos de gerenciamento de projetos, proporcionam ferramentas que contribuem para estabelecer um ambiente seguro para decisões e resolução conjunta de problemas. Essas ferramentas estão se tornando cada vez mais vitais para as empresas no dia a dia, uma vez que aumentam a produtividade e a eficácia. As vantagens englobam o aprimoramento do trabalho coletivo e a estruturação dos projetos. As organizações não governamentais (ONGs) também se aproveitam dessas plataformas, empregando-as para unificar os esforços de todos os participantes em suas causas sociais. As plataformas de colaboração promovem inovação, acelerando a procura por respostas para questões complexas da sociedade, além de simplificar o acesso à informação e a comunicação entre as entidades e população. (DIAS, 2004)

Há várias ferramentas de colaboração, cada uma com um propósito específico, tais como comunicação, organização e gerenciamento de projetos, todas voltadas para aprimorar a eficiência do trabalho. Por exemplo, o Google Drive simplifica a troca de arquivos de qualquer local com conexão à internet; o WhatsApp possibilita uma comunicação instantânea e imediata; e o Trello e o Monday auxiliam na organização das fases e tarefas atribuídas durante os projetos, tornando a administração mais rápida e eficiente. Esses instrumentos são essenciais para aprimorar a colaboração e assegurar que todos estejam em sintonia e sejam produtivos. (GOMES, 2021)

Com o aumento da demanda por processos mais eficazes, as plataformas de colaboração surgiram para conectar indivíduos e grupos em busca de um propósito compartilhado. Devido ao progresso tecnológico e ao crescimento da utilização da internet, essas plataformas se desenvolveram ao longo dos anos. Elas surgiram como instrumentos básicos para a partilha de conteúdo online, como as "wikis", que possibilitavam a qualquer usuário a partilha do seu saber sobre um assunto específico, fomentando um intercâmbio livre e colaborativo de informações. (JUNIOR, 2024).

O progresso tecnológico impulsionou fortemente as plataformas de colaboração, tornando a internet um ambiente mais interativo, onde indivíduos podem acessar e trocar informações de qualquer parte do globo. Com a disseminação dos smartphones, essas plataformas se tornaram ainda mais acessíveis, possibilitando o monitoramento em tempo real das atividades e colaborações, o que tornou a comunicação e o trabalho em equipe mais dinâmicos e ágeis. (OLIVEIRA, 2023)

Algumas plataformas começaram a incorporar APIs (Interfaces de Programação de Aplicações), o que aumentou sua eficiência e adaptabilidade. O Google Maps, por exemplo, oferece serviços de geolocalização através de suas APIs, simplificando a elaboração de soluções para problemas sociais. Este recurso tem se mostrado útil em ações como a assistência a animais em situação de vulnerabilidade, foco principal deste estudo, ao possibilitar a disseminação rápida e fácil de informações cruciais. (ALLAN, 2013).

Atualmente, as plataformas colaborativas estão recebendo novos recursos de inteligência artificial (IA), que examinam o comportamento dos usuários para propor aprimoramentos e otimizar as funções de forma assertiva. Esta tecnologia tem desempenhado um papel crucial no progresso dessas plataformas, possibilitando a criação de soluções mais sofisticadas e revolucionando a maneira como as tarefas colaborativas são realizadas. Isso possibilita que as entidades obtenham mais eficácia e ajustem suas táticas às demandas autênticas dos usuários, trazendo vantagens para todos os participantes e para os resultados finais. (FILHO, 2023).

As plataformas colaborativas empregam diversas tecnologias para assegurar uma comunicação eficaz e simplificar o trabalho em grupo, focando em elementos como infraestrutura, integração e conectividade. A infraestrutura tecnológica engloba serviços na nuvem, que possibilitam o armazenamento e o tratamento de grandes quantidades de dados de maneira segura. Exemplos disso são o Google Cloud e o Microsoft Azure, que proporcionam os recursos fundamentais para o funcionamento contínuo das aplicações. Neste projeto, esses instrumentos são fundamentais para que as ONGs tenham acesso a informações a qualquer instante, simplificando suas atividades. Ademais, as APIs são essenciais para a integração de serviços externos. Um exemplo é a API do Google Maps, que proporciona aos usuários a capacidade de navegar pelos mapas. O mapa permite à plataforma identificar os animais em situação de vulnerabilidade, aproximando a comunidade do resgate e da assistência que esses animais tanto necessitam. (NETO, 2009)

Para que diversos sistemas e programas funcionem de forma harmoniosa, as plataformas de colaboração necessitam de uma boa interoperabilidade, que garanta o fluxo contínuo de dados. As APIs desempenham um papel fundamental nesse processo, permitindo uma conexão eficiente e unificada com serviços externos. Ademais, a utilização de microsserviços, que segmenta as aplicações em módulos autônomos, torna essas plataformas mais versáteis e eficazes, simplificando alterações e aprimoramentos quando necessário. Tudo isso contribui para uma experiência de uso mais suave e ajustada às demandas dos utilizadores. (GOODWIN, 2024)

As tecnologias usadas nessas plataformas colaborativas se complementam para garantir um funcionamento eficiente e trazer benefícios tanto para o ambiente de trabalho quanto para a vida social. Com ferramentas de comunicação como chats e videoconferências, essas plataformas quebram barreiras físicas e permitem interações em tempo real, aproximando as pessoas independentemente de onde estejam. Além disso, facilitam a gestão de tarefas, permitindo acompanhar o andamento de cada atividade e, com isso, impulsionando a produtividade. A redução de custos é outro grande benefício, pois diminui a necessidade de deslocamentos e possibilita o trabalho colaborativo remoto, otimizando tempo e recursos. E, por fim, essas plataformas promovem inclusão, facilitando o acesso ao conhecimento e incentivando a participação em projetos sociais e educacionais, superando as barreiras geográficas e econômicas que muitas vezes limitam o alcance desses projetos.

2.5 Impacto Social das Plataformas Colaborativas

As plataformas de colaboração estão impulsionando uma mudança concreta na maneira como indivíduos e entidades lidam com desafios sociais, ambientais e humanitários. Com a ajuda da tecnologia, elas estão desenvolvendo soluções inovadoras que impactam diretamente o envolvimento em causas que requerem a participação de todos. Estes sistemas promovem a interação e colaboração entre pessoas e entidades, frequentemente de forma descentralizada, potencializando o impacto social e conectando as pessoas a ações que impactam o mundo. (MAIA, 2018)

Plataformas de colaboração são essenciais para estabelecer redes de suporte e potencializar ações em campos como meio ambiente, direitos humanos, saúde

pública e luta contra a desigualdade. Essas plataformas unem ONGs, voluntários, empresas e indivíduos comuns, possibilitando a colaboração para lidar com desafios complexos e desenvolver soluções inovadoras de maneira mais eficaz. Um exemplo disso é o crowdfunding, que simplifica a obtenção de fundos para causas sociais e humanitárias, possibilitando o suporte financeiro a projetos que anteriormente encontravam obstáculos para obter patrocínios convencionais. (ABES, 2023)

Um caso emblemático é a plataforma GoFundMe, que possibilita que indivíduos comuns contribuam para diversas causas sociais, desde tratamentos médicos até projetos de justiça social. Este método de financiamento coletivo democratiza o processo, concentrando a autoridade na comunidade em si, sem se basear unicamente em grandes empresas ou no governo. (GOFUNDME, 2024)

Um exemplo similar é o Change.org, uma plataforma que congrega milhões de indivíduos engajados na elaboração de petições com o objetivo de fomentar transformações políticas e sociais. Por meio dela, os cidadãos têm a oportunidade de se envolver ativamente em campanhas que frequentemente impactam decisões governamentais e empresariais, promovendo a responsabilidade social e dando voz à comunidade em assuntos relevantes. (CHANGE.ORG, 2024)

Ademais, plataformas como a EcoCrowd se concentram em iniciativas ambientais, auxiliando entidades e ativistas na captação de recursos para iniciativas ecológicas. Portanto, as plataformas colaborativas oferecem uma nova perspectiva para a filantropia, ligando diretamente as soluções a quem tem capacidade financeira para financiá-las. Esta dinâmica intensifica o sentimento de comunidade e estabelece um objetivo comum, no qual todos participam para causar um impacto positivo. (ECOCROWD, 2024)

As plataformas colaborativas não se limitam a simplificar o suporte a causas sociais; são poderosas ferramentas de empoderamento para comunidades que estão à margem da sociedade. Essas plataformas ligam grupos que, de outra forma, teriam problemas para obter recursos e informações, estabelecendo espaços onde podem trocar experiências e táticas para lidar com desafios comuns.

Um caso inspirador é a Kiva, uma plataforma de microcrédito que possibilita que indivíduos ao redor do globo financiem pequenos negócios em áreas em expansão. Com financiamentos acessíveis, empreendedores locais, frequentemente mulheres ou integrantes de comunidades marginalizadas, conseguem começar ou ampliar seus empreendimentos, ultrapassando obstáculos financeiros. Este efeito

social possui dois efeitos: aprimora a vida do empresário e estimula a economia local, fomentando a inclusão e a criação de novas oportunidades na comunidade. (KIVA, 2024)

Estas plataformas ultrapassam a simples distribuição de recursos; elas estabelecem autênticas redes de solidariedade, concedendo voz e influência a grupos que, historicamente, tiveram escassa visibilidade. Ao ligar essas comunidades a um público mundial e ao saber coletivo, as plataformas colaborativas reduzem a distância entre quem detém o poder e quem mais necessita de suporte, fomentando a inclusão e a igualdade de oportunidades.

As plataformas de colaboração possuem a capacidade de revolucionar a maneira como lidamos com os grandes desafios mundiais, proporcionando chances para que pessoas e organizações inovem na procura de respostas para problemas sociais e ambientais. Ao fomentar a autonomia das comunidades e simplificar o suporte a causas relevantes, essas plataformas estão contribuindo para aproximar problemas complexos da solução através da colaboração. Com o progresso contínuo da tecnologia, espera-se que esses recursos assumam uma função ainda mais crucial na batalha contra as desigualdades e na promoção de um crescimento sustentável e inclusivo

2.6 Tecnologias para desenvolvimento de aplicativos Android

A plataforma Android é extremamente popular no universo tecnológico e está presente no cotidiano de milhões de indivíduos que utilizam smartphones. A maioria das grandes empresas de telefonia móvel opta por este sistema operacional para proporcionar uma experiência positiva aos seus clientes. Devido à sua popularidade e alcance, diversos aplicativos são desenvolvidos exclusivamente para Android, tirando proveito dessa vasta base de usuários. Como dito por Rodrigues: “Buscamos aprender e nos dedicamos ao desenvolvimento de um aplicativo mobile, previsto para a plataforma Android, a qual possui a maior quantidade de usuários.” (RODRIGUES, 2015)

A história do Android começa nos Estados Unidos, com o objetivo inicial de criar um sistema inovador que proporcionasse acessibilidade tanto para desenvolvedores quanto para usuários. Conforme Santos: “Desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White, todos com experiência na área de

tecnologia, decidiram criar um sistema inovador. Assim, fundaram a Android Inc.” (SANTOS, 2017)

Em 2003, a Android Inc. foi criada em Palo Alto, Califórnia (EUA), sendo posteriormente comprada pela Google em julho de 2005. O Google, uma empresa de referência global em termos de crescimento, adquiriu a Android com o intuito de expandir suas operações e adaptar-se às futuras atualizações tecnológicas, oferecendo aos usuários serviços de qualidade e acessíveis. (SOMBRA, 2014)

No ano de 2005, a Android tinha dificuldades financeiras, o que fez o Google, interessado no mercado de smartphones, comprar a empresa e sua equipe de programadores. Assim surgiu o projeto Google Mobile Division, voltado para a criação de aparelhos celulares com o objetivo de rivalizar com gigantes do ramo, como a Apple e a Nokia. Esta compra assinalou o começo da jornada do Google no mundo móvel, fomentando a ascensão do Android como uma plataforma dominante. (SANTOS, 2017)

Com o progresso tecnológico, a evolução dos smartphones gerou uma chance ideal para o Google investir neste novo segmento de mercado. O Android foi concebido para ser mais adaptável, oferecendo aos usuários uma experiência positiva, mesmo diante da intensa concorrência de outros sistemas, como o iOS da Apple e o Windows da Nokia.

O Android surgiu a partir de outro sistema operacional, utilizando sua ideia principal para simplificar e tornar mais acessível o seu uso. Segundo Gaio, “O sistema operacional móvel Android possui código aberto e roda sobre o núcleo Linux.” Isso implica que qualquer indivíduo tem acesso ao código-fonte, possibilitando que os produtores modifiquem e customizem o sistema conforme suas próprias demandas. (GAIO, 2021)

Para desenvolver aplicativos e atualizações do sistema, é necessária uma linguagem de programação. No caso do Android, os desenvolvedores utilizam a linguagem Java, como explica Gaio (2021): “Os desenvolvedores usam a linguagem de programação Java para programação no Android.” (GAIO, 2021)

O processo de desenvolvimento do Android envolve a compilação de código em Java, que é convertida em bytecode específico para o Android. Sombra (2014) detalha: “O SDK compila o código Java e traduz o bytecode para um arquivo .dex, específico da máquina virtual do Android.” Esse processo é amplamente conhecido e utilizado pelos desenvolvedores. (SOMBRA, 2014)

Além disso, os desenvolvedores contam com IDE's (Ambientes de Desenvolvimento Integrado), que permitem a edição e modificação dos programas. Uma das ferramentas mais conhecidas para desenvolvimento de aplicativos Android é o Eclipse, que, segundo Gaio (2021), "foi primeiramente projetado para o desenvolvimento em Java, mas permite programar em diversas linguagens."

Com a inovação trazida pelo Android e sua integração ao Google, foi apenas uma questão de tempo até que grandes empresas de tecnologia, como Samsung, Motorola e LG, adotassem o sistema em seus dispositivos. Mesmo com a concorrência de outras organizações, a plataforma Android se destacou no mercado. Conforme Rosatto (2018), "No segundo trimestre de 2013, os aparelhos com sistema operacional Android respondiam por 79,3% do mercado global, enquanto os aparelhos da Apple, com sistema iOS, representavam apenas 13,2%." (ROSATTO, 2018)

Estes números indicam que, em um dado instante, o Android conseguiu superar o iOS, que até então dominava o mercado de smartphones. Este crescimento pode ser creditado às alianças estratégicas estabelecidas pelo Android e à facilidade de uso que o sistema proporciona aos usuários, fazendo dele uma alternativa mais atrativa para um público progressivamente maior.

2.7 APIs de Localização

As APIs de geolocalização tornaram-se instrumentos indispensáveis para desenvolver aplicativos que necessitam interagir com mapas, proporcionar navegação em tempo real e identificar usuários. A API do Google Maps é uma das mais conhecidas e frequentemente empregadas, pois proporciona diversas funcionalidades, tais como a visualização de mapas, a determinação de rotas, a identificação de locais de interesse e serviços de geolocalização. A sua aplicação é relativamente simples, e a documentação minuciosa simplifica a utilização por programadores de todas as habilidades. Portanto, o Google Maps é a opção preferida para diversos aplicativos móveis que necessitam de uma localização exata e constante, como os de transporte, turismo e plataformas de colaboração. (MUNDO API, 2018).

A Google Maps API oferece três serviços principais: Mapas, Direções e Locais. O primeiro permite adicionar mapas interativos aos aplicativos, proporcionando uma visualização clara e fácil de entender; o segundo facilita a criação de rotas e fornece orientações detalhadas entre diferentes pontos; e o terceiro oferece informações

sobre locais próximos ao usuário, como restaurantes, lojas e parques. Além disso, essa API se conecta de forma simples a outras plataformas, o que a torna muito flexível e aplicável a diferentes tipos de aplicativos. Ela se destaca pela precisão e confiabilidade, já que a base de dados do Google é constantemente atualizada, refletindo as mudanças no ambiente urbano, como a abertura de novos negócios, alterações nas ruas e estradas, e até o fechamento de empresas. (ALLAN, 2013).

Contudo, um dos maiores obstáculos ao utilizar a Google Maps API é o custo associado ao uso de suas funcionalidades em larga escala. Apesar de existir um plano gratuito, a quantidade de pedidos pode facilmente exceder o limite, especialmente em aplicativos que se baseiam fortemente na geolocalização e possuem muitos usuários. Isso pode representar um desafio considerável para programadores e entidades com recursos financeiros restritos. Adicionalmente, inquietações acerca da privacidade e a dependência de uma única plataforma global têm impulsionado muitos programadores a procurar opções que proporcionem mais adaptabilidade e maior domínio sobre as informações. (ALLAN, 2013).

As APIs mais utilizadas incluem o OpenStreetMap (OSM) e o Mapbox. O OpenStreetMap é um projeto colaborativo que disponibiliza informações geográficas gratuitas e abertas, possibilitando que os programadores modifiquem os mapas de acordo com suas demandas. Numerosas entidades optam pelo OSM para escapar dos custos das APIs proprietárias e obter maior controle sobre a gestão dos dados de localização. Contudo, a exatidão e a qualidade dos dados podem diferir conforme a região, pois as atualizações são realizadas por voluntários, o que pode afetar a uniformidade dos dados em diversas regiões. (OSM, 2024)

Por outro lado, a API Mapbox proporciona uma experiência de desenvolvimento muito semelhante à do Google Maps, apresentando uma interface de fácil utilização e uma vasta variedade de funcionalidades, como visualização de mapas, geolocalização e orientações. Uma característica distinta da Mapbox é a capacidade de customização visual, possibilitando que os mapas sejam ajustados de acordo com os gostos estéticos do usuário. Apesar de também possuir uma estrutura de preços, a Mapbox costuma ser considerada uma opção mais barata em comparação à Google Maps API, além de ser mais adaptável para customizações complexas. No final das contas, a seleção da API de localização ideal é influenciada por diversos elementos, tais como o custo, a quantidade de usuários e a exigência de personalização. (MAPBOX, 2024)

Apesar da API do Google Maps ser muito apreciada por sua exatidão e confiabilidade, opções como o OpenStreetMap e o Mapbox apresentam alternativas interessantes, particularmente para projetos que necessitam de maior adaptabilidade e controle sobre informações geográficas. Em um contexto de avanço tecnológico acelerado, as APIs de geolocalização persistem em seu desenvolvimento, gerando novas possibilidades para os programadores e afetando diretamente a maneira como nos relacionamos com o mundo que nos rodeia.

2.8 Sistemas Distribuídos

Os sistemas distribuídos integram a classe de computação paralela e são classificados de acordo com o nível de paralelismo suportado pelo hardware, além da distância entre os processos que compõem o landscape da aplicação. Sistemas distribuídos são caracterizados por uma rede de computadores interligados que se comunicam através de uma rede com o objetivo de executar uma atividade de maneira distribuída entre eles a fim de diminuir o tempo de execução assim aumentando seu desempenho

2.8.1 Modelo Cliente Servidor

O Modelo cliente-servidor é amplamente utilizado na computação, onde os dispositivos são divididos em dois grupos um cliente e outro servidor. O cliente é responsável por realizar requisições de serviços ou recursos, enquanto o servidor é quem atende elas essas solicitações. As requisições ocorrem por meio de um canal de comunicação baseado em um protocolo de rede, como o HTTP ou o TCP/IP, que definem regras de comunicação entre os dispositivos e garantem uma correta interpretação das requisições e respostas. Esse modelo possui um ambiente personalizável capaz de atender as necessidades de diferentes empresas além de ser altamente escalável, flexível, seguro e confiável assim consolidando sua popularidade em implementações de rede em larga escala (NAPOLEON, 2024).

2.8.2 CRUD

CRUD é um acrônimo com as iniciais das quatro operações fundamentais de qualquer banco de dados: Create (criar), Read (ler), Update (atualizar) e Delete (apagar). Essas operações formam o núcleo da maioria dos softwares de banco de dados e dos sistemas de memória computacional, como as memórias persistentes e voláteis. Memórias persistentes mantêm a informação mesmo após serem desligadas,

enquanto memórias voláteis perdem o conteúdo armazenado quando são desligadas da fonte de energia (SUMOLOGIC, 2024).

Um dos tipos de bancos de dados onde o CRUD é amplamente aplicado são os bancos de dados relacionais, que são compostos por tabelas de informações conectadas a outras tabelas complementares. O conceito de CRUD, no entanto, não se restringe apenas a essas quatro operações básicas; podendo ser estendido a funções de alto nível, como exclusões reversíveis, nas quais a informação não é completamente deletada, mas marcada com o status de "deletada".

O CRUD também pode ser implementado no lado do usuário em aplicações, permitindo que eles criem registros, recuperem dados, atualizem informações e excluam registros. Esse modelo é um componente essencial no desenvolvimento de sistemas que manipulam dados, pois oferece uma abordagem simples e eficiente para o gerenciamento de informações (MIRANDA, 2024).

3 MATERIAIS E MÉTODOS

No capítulo de Materiais e Métodos, serão apresentados os recursos tecnológicos empregados no desenvolvimento da plataforma colaborativa. Serão detalhados o hardware necessário para a implementação do aplicativo, a estrutura do backend, que viabiliza a integração entre os serviços da aplicação e a API do Google Maps, visando garantir uma experiência eficiente e intuitiva para o usuário.

3.1 Material e Ferramentas

3.1.1 Hardware

Computador Básico: Um computador simples para a programação em Flutter e FireBase do aplicativo. Componentes: I5 10210U, 8GB Ram DDR4, Nvidia GeForce MX110 2GB de Vram GDDR4.

3.1.2 Framework

Flutter: Para a criação do nosso aplicativo decidimos utilizar o Framework Flutter ele é de código aberto desenvolvido pelo Google para a criação de aplicativos nativos multiplataforma (iOS, Android, web e desktop) a partir de um único código-fonte. Baseado na linguagem Dart, o Flutter se destaca pela capacidade de renderizar componentes de interface diretamente, sem a necessidade de interações com APIs nativas, proporcionando um desempenho elevado. Além disso, o Flutter oferece uma vasta biblioteca de widgets personalizáveis, permitindo uma grande flexibilidade no

design das interfaces. O uso de Flutter em projetos de desenvolvimento tem crescido devido à sua eficiência e à facilidade de manutenção de código, já que permite o desenvolvimento multiplataforma com um único código fonte (ALBERTO, 2024).

3.1.3 Ambiente

Firestore: Para o funcionamento do front-end e a integração com o back-end, utilizamos o Firestore, que se caracteriza como um Backend as a Service (BaaS). O Firestore oferece um conjunto de serviços de back-end, como autenticação, análise de dados, armazenamento de dados, notificações e banco de dados, entre outros. Dessa forma, possibilita a implementação de diversas funcionalidades sem a necessidade de desenvolver um back-end personalizado, otimizando o tempo e os recursos utilizados no desenvolvimento (RIBEIRO, 2024).

3.1.4 Back-End

Dart: É uma linguagem de programação que tem como um dos objetivos facilitar o desenvolvimento de aplicativos multiplataforma, como aplicações web, desktop e móveis. Ela foi desenvolvida pelo Google, ela possui semelhanças com linguagens como Java, C++ e JavaScript. Dart é considerada "levemente tipada" e pode ser tanto interpretada quanto compilada. Foi criada com o objetivo de unificar o front-end, as aplicações móveis e o servidor, possibilitando também a criação de aplicações nativas, móveis e web (BARRO, 2024).

3.1.5 Kotlin

Kotlin é uma linguagem desenvolvida pela JetBrains e é atualmente uma das principais linguagens para o desenvolvimento de softwares e aplicações Android, sendo a principal opção para programação nativa. Totalmente compatível com Java, Kotlin aplica o conceito de interoperabilidade de forma eficaz, permitindo chamadas entre as duas linguagens, ou seja, é possível chamar funções Java usando Kotlin e vice-versa. Essa característica torna o Kotlin um substituto atraente para o Java, oferecendo maior praticidade aos desenvolvedores (ALICE; FERNANDES, 2024).

3.2 Métodos

3.2.1 Cliente-Servidor

O aplicativo é desenvolvido com uma arquitetura cliente-servidor, utilizando Firestore como backend para gerenciamento de autenticação, armazenamento de dados e hospedagem de arquivos. A arquitetura cliente-servidor utilizada permite uma separação clara das responsabilidades entre a interface do usuário (cliente) e a gestão de dados e autenticação (servidor).

A arquitetura cliente-servidor do aplicativo foi desenvolvida com base em dois componentes principais: o cliente, que fornece a interface de interação com o usuário, e o servidor, responsável pelo gerenciamento de dados e autenticação. Essa abordagem possibilita uma separação eficaz entre a lógica de apresentação, que é feita pelo cliente, e a lógica de negócios, que é gerida pelo servidor.

O cliente é desenvolvido utilizando o framework Flutter, que permite a criação de interfaces de usuário modernas e interativas. A principal função do cliente é gerenciar as interações do usuário, capturar entradas e enviar requisições ao servidor para operações como autenticação, consulta e gravação de dados. A comunicação entre cliente e servidor é feita utilizando APIs providas pelo Firebase, possibilitando que as operações sejam realizadas de forma eficiente e segura.

O cliente gerencia toda a navegação entre as telas do aplicativo, como login, cadastro, adoção e resgate de animais. A coleta de informações do usuário, como credenciais de login e dados de perfil, é feita pelo cliente, que em seguida envia esses dados para o servidor para validação e armazenamento.

O servidor é sustentado pelo Firebase, que oferece serviços de backend como autenticação, banco de dados e armazenamento de arquivos. A utilização do Firebase como servidor proporciona uma solução escalável e segura para gerenciar a comunicação com o cliente, autenticar usuários e armazenar informações de forma eficiente.

Os principais serviços do Firebase utilizados são: O Firebase Authentication é responsável pela autenticação dos usuários, permitindo que estes façam login no aplicativo utilizando e-mail e senha ou através de uma conta do Google. O cliente coleta as credenciais fornecidas pelo usuário e as envia ao servidor Firebase, que realiza a validação e, caso as credenciais sejam válidas, permite o acesso aos recursos do sistema. Essa autenticação garante que apenas usuários autorizados tenham acesso às funcionalidades do aplicativo.

Essa abordagem traz diversos benefícios para o nosso projeto. Em primeiro lugar, garante que apenas usuários autenticados e autorizados tenham acesso às funcionalidades do aplicativo, protegendo informações sensíveis e evitando acessos não autorizados. Isso é fundamental para manter a integridade e a confiabilidade da plataforma, especialmente em um contexto que envolve dados pessoais e interações entre usuários.

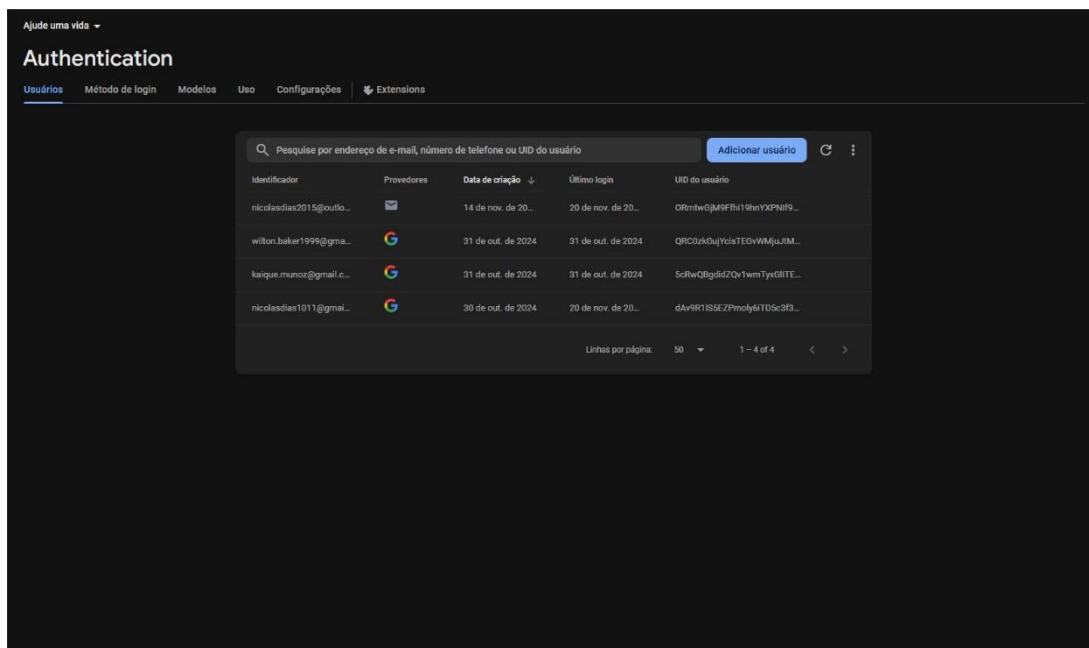


Figura 1 - Firebase Authentication

O Firestore Database desempenha um papel crucial no armazenamento e gerenciamento das informações referentes ao aplicativo desenvolvido. Como um banco de dados NoSQL em tempo real, o Firestore permite a persistência de dados de forma escalável e eficiente, atendendo às necessidades de aplicativos modernos que demandam atualizações instantâneas e acesso simultâneo por múltiplos usuários.

No contexto deste projeto, o Firestore é utilizado para armazenar perfis de usuários, registros de animais disponíveis para adoção e detalhes relacionados a resgates. Quando um usuário interage com o aplicativo, seja criando um perfil, registrando um animal ou atualizando informações de resgate, o cliente envia esses dados para o servidor Firebase. O Firestore, então, armazena essas informações em coleções e documentos estruturados de forma hierárquica, facilitando a organização e recuperação dos dados.

Essa integração com o Firebase proporciona diversos benefícios, dentre os quais se destacam a sincronização em tempo real e a consistência dos dados. A sincronização em tempo real permite que quaisquer alterações feitas no banco de dados sejam imediatamente refletidas em todos os dispositivos conectados, garantindo que todos os usuários tenham acesso às informações mais atualizadas. Isso é particularmente importante em um aplicativo que depende da colaboração e interação entre diferentes usuários, como no caso de adoções e resgates de animais.

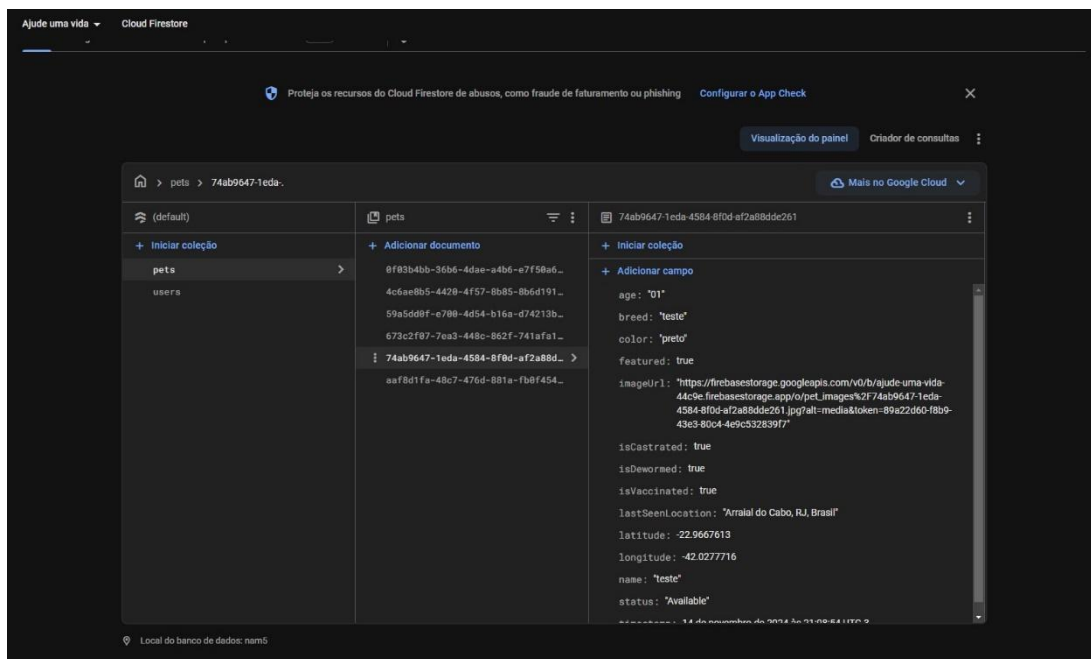


Figura 2 - Firestore Database

O Firebase Storage é utilizado para o armazenamento de arquivos, como imagens dos animais disponíveis para adoção. Quando o usuário faz o upload de uma imagem, o cliente envia o arquivo para o servidor Firebase, que o armazena no Firebase Storage. Após o armazenamento, o servidor retorna a URL da imagem, que é utilizada pelo cliente para exibir o conteúdo de forma adequada. Essa abordagem facilita o gerenciamento de arquivos de mídia, garantindo o acesso seguro e organizado.

Essa solução simplifica o gerenciamento de arquivos de mídia. Em vez de lidar com servidores complexos ou sistemas de armazenamento próprios, o Firebase Storage oferece uma plataforma pronta para uso, escalável e integrada com os demais serviços do Firebase. Isso garante que as imagens estejam sempre acessíveis e protegidas, graças aos recursos de segurança e controle de acesso oferecidos pelo serviço.

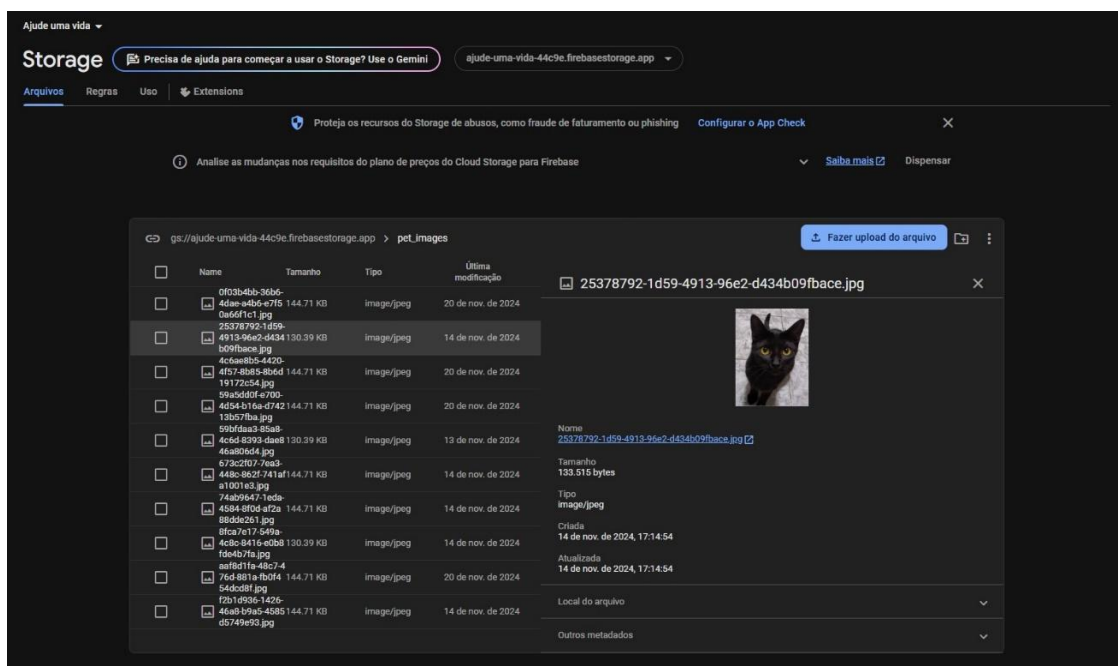


Figura 3 - Firebase Storage

A comunicação entre o cliente e o servidor ocorre de maneira contínua e é fundamental para o funcionamento do aplicativo. O cliente envia requisições ao servidor sempre que precisa realizar operações como autenticação de usuário, gravação de informações no banco de dados ou upload de imagens. Essas requisições são realizadas através das APIs do Firebase.

O servidor recebe as requisições, processa os dados e retorna as respostas necessárias ao cliente. Por exemplo, ao realizar o login, o cliente envia as credenciais ao servidor, que verifica a validade das informações e retorna uma resposta, autorizando ou negando o acesso. Da mesma forma, ao registrar um novo pet para adoção, o cliente coleta os dados fornecidos pelo usuário, envia para o servidor, que os armazena no Firestore e retorna uma confirmação da operação.

Essa comunicação bidirecional é realizada utilizando protocolos seguros, garantindo a integridade e a confidencialidade dos dados trafegados. A integração com o Firebase também permite que os dados sejam sincronizados em tempo real entre diferentes dispositivos, garantindo uma experiência de usuário mais dinâmica e eficiente.

A arquitetura cliente-servidor implementada no desenvolvimento deste aplicativo permite uma separação clara entre a interface do usuário e as funcionalidades de backend. O cliente, desenvolvido com Flutter, é responsável por

gerenciar a interação direta com os usuários, enquanto o servidor, utilizando os serviços do Firebase, realiza operações de autenticação, armazenamento de dados e hospedagem de arquivos. Essa separação de responsabilidades proporciona maior escalabilidade e segurança ao sistema, além de facilitar a manutenção e a adição de novas funcionalidades. A utilização de Firebase Authentication, Firestore e Firebase Storage mostra-se eficiente para gerenciar os principais aspectos do backend, garantindo que o aplicativo funcione de maneira consistente e segura.

3.2.2 CRUD

CRUD é um acrônimo que representa as quatro operações fundamentais de qualquer sistema de gerenciamento de dados: Create (Criar), Read (Ler), Update (Atualizar) e Delete (Excluir). Essas operações são utilizadas para manipular dados de forma eficiente e são aplicadas amplamente em sistemas que necessitam de armazenamento e manutenção de informações, como aplicativos web e móveis.

O CRUD é a base do gerenciamento de dados, sendo essencial para garantir a interação entre o usuário e o banco de dados de maneira eficiente e compreensível. As operações CRUD oferecem as funcionalidades básicas para qualquer aplicação que envolva entrada, visualização, modificação e exclusão de informações, tornando possível gerenciar um ciclo completo de dados.

No aplicativo desenvolvido, que visa facilitar o resgate e a adoção de animais, a aplicação do CRUD está integrada em diversas funcionalidades, abrangendo desde o registro de novos usuários até a atualização de informações de resgate. A seguir, são descritas as aplicações de cada uma das operações CRUD no contexto do aplicativo, bem como a sua integração com o Firebase.

A operação de criação está presente em diferentes aspectos do aplicativo, desde o cadastro de novos usuários até o registro de animais resgatados e a adição de informações sobre ONGs parceiras.

Cadastro de Usuários: Durante o registro de um novo usuário, todas as informações fornecidas, como nome, e-mail, telefone e localização, são enviadas do cliente (Flutter) para o servidor (Firebase). Essa operação resulta na criação de um novo documento no banco de dados Firestore, onde as informações são armazenadas.

Registro de Animais: Ao resgatar um animal, o usuário preenche informações como nome, idade, raça e localização do pet. Esses dados, juntamente com uma

imagem do animal, são enviados ao servidor para criação de um novo registro no Firestore e Firebase Storage. Assim, o aplicativo mantém um banco de dados de animais resgatados disponível para consulta futura.

3.2.2.1 Read (Leitura)

A operação de leitura é amplamente utilizada em diferentes telas do aplicativo, permitindo a visualização das informações já registradas e garantindo uma experiência consistente para o usuário.

Visualização de Pets para Adoção: A tela de adoção utiliza a operação de leitura para obter as informações dos pets registrados no Firestore. Esses dados são então exibidos para o usuário, possibilitando que ele veja as características dos animais, como raça, idade e outras informações.

Listagem de ONGs e Animais Favoritos: O Firestore também é utilizado para armazenar informações sobre ONGs parceiras e animais marcados como favoritos. A operação de leitura permite ao usuário acessar, de forma rápida e eficiente, a lista de ONGs, bem como os pets que ele adicionou aos favoritos.

3.2.2.2 Update (Atualização)

A atualização de dados é essencial para garantir que as informações armazenadas no sistema estejam sempre corretas e reflitam o estado atual dos registros.

Atualização de Perfil: O usuário pode atualizar suas informações de perfil, como telefone ou localização. Nesse caso, a operação de atualização é realizada quando o cliente envia os novos dados para o servidor, que então atualiza o documento correspondente no Firestore.

Mudança no Status de Pets: A atualização de informações dos animais ocorre, por exemplo, quando um pet é adotado. Nesse cenário, o status do animal é atualizado no Firestore para indicar que ele não está mais disponível para adoção. Essa funcionalidade é fundamental para manter a consistência dos dados e evitar que animais já adotados sejam mostrados como disponíveis.

A operação de exclusão é utilizada para remover informações que não são mais necessárias ou que foram registradas incorretamente.

3.2.2.3 Remoção de Animais Favoritos

Quando o usuário decide remover um pet da sua lista de favoritos, uma operação de exclusão é executada. O cliente envia uma solicitação ao servidor para

que o registro correspondente seja removido do Firestore, garantindo que a lista de favoritos reflita apenas os animais nos quais o usuário ainda está interessado.

3.2.2.4 Exclusão de Registros de Resgates

Caso um animal resgatado seja adotado ou outra situação similar ocorra, as informações do pet podem ser excluídas do banco de dados. Isso é feito para manter o sistema limpo e evitar o armazenamento de dados desnecessários ou desatualizados.

Todas as operações CRUD descritas são implementadas por meio da integração do Flutter (cliente) com o Firebase (servidor). O Firebase oferece uma plataforma robusta e integrada que permite que as operações CRUD sejam realizadas de maneira eficiente e segura.

Utilizado como o banco de dados principal, o Firestore é a peça central para a execução das operações CRUD. Ele permite o armazenamento de documentos que representam os registros de usuários, animais e ONGs. As operações Create, Read, Update e Delete são realizadas diretamente através das APIs do Firebase, garantindo que os dados estejam sempre sincronizados e consistentes.

O Firebase Storage é utilizado para armazenar arquivos, como as imagens dos animais. A operação Create é usada para fazer o upload dessas imagens, e a operação Read permite que a URL da imagem seja recuperada e exibida na interface do aplicativo, proporcionando uma melhor apresentação dos pets.

A aplicação das operações CRUD no desenvolvimento deste aplicativo mostra-se essencial para garantir a funcionalidade completa do sistema. Cada uma das operações desempenha um papel importante, desde a inserção e consulta de dados até a atualização e exclusão, garantindo que os dados do aplicativo estejam sempre atualizados e corretos. A integração com o Firebase proporciona uma maneira eficaz de gerenciar todas essas operações, assegurando que o aplicativo seja seguro, escalável e fácil de manter.

Portanto, o CRUD é uma parte crucial do desenvolvimento do aplicativo de adoção e resgate de animais, garantindo que os usuários possam interagir com o sistema de forma intuitiva e eficiente. Essa estrutura de gerenciamento de dados permite não apenas a inclusão de novas informações, mas também a modificação, consulta e exclusão de registros, cobrindo assim todas as necessidades operacionais do aplicativo e permitindo que ele cumpra seu propósito de facilitar o resgate e a adoção de animais.

3.2.3 Flutter

A escolha do framework Flutter para o desenvolvimento da aplicação móvel neste trabalho deve-se à busca pela homogeneidade da stack tecnológica utilizada, mantendo o foco no ambiente Google. O Flutter é um framework de código aberto desenvolvido pela Google que permite a criação de aplicativos nativos de alta performance para iOS e Android a partir de uma única base de código escrita em Dart, também uma linguagem criada pela Google.

Ao optar pelo Flutter, garantiu-se a consistência na utilização de tecnologias pertencentes ao ecossistema Google, facilitando a integração entre serviços e ferramentas, além de proporcionar uma curva de aprendizado mais linear. Essa homogeneidade é fundamental para a eficiência no desenvolvimento, manutenção e escalabilidade do projeto.

O Flutter oferece diversos benefícios, tais como:

- **Desenvolvimento Rápido:** Com o recurso de Hot Reload, é possível visualizar instantaneamente as alterações feitas no código, agilizando o processo de desenvolvimento e testes.
- **Interface de Usuário Expressiva e Flexível:** O framework fornece uma ampla gama de widgets personalizáveis que facilitam a criação de interfaces modernas e responsivas.
- **Performance Nativa:** Diferentemente de outros frameworks que utilizam bridges para comunicação com os componentes nativos, o Flutter compila diretamente para código nativo, garantindo alta performance.
- **Comunidade Ativa e Suporte:** Sendo mantido pela Google, o Flutter possui uma comunidade crescente e um suporte robusto, contribuindo para a resolução de problemas e atualização constante da tecnologia.

A manutenção do foco no ambiente Google, utilizando o Flutter para o front-end e serviços como o Firebase para o back-end, permitiu uma integração mais fluida entre os componentes do sistema. Dessa forma, aproveitou-se ao máximo os recursos oferecidos pela plataforma, alinhando as necessidades técnicas do projeto com a estratégia de manter uma stack coesa e eficiente.

Portanto, a utilização do Flutter não apenas atende aos requisitos funcionais e não funcionais do projeto, como também potencializa os benefícios decorrentes de uma stack tecnológica homogênea e centrada no ambiente Google.

3.2.4 Dart

A escolha da linguagem de programação Dart foi essencial para o desenvolvimento da aplicação móvel, em conjunto com o framework Flutter. Dart é uma linguagem moderna, orientada a objetos, desenvolvida pela Google, que oferece uma sintaxe clara e concisa, facilitando a escrita e manutenção do código. Ao utilizar Dart, foi possível alcançar uma elevada produtividade, uma vez que sua sintaxe simplificada e tipagem estática auxiliam na detecção precoce de erros, otimizando o tempo de desenvolvimento.

Além disso, Dart proporciona performance otimizada, permitindo tanto a compilação Just-In-Time (JIT) quanto Ahead-Of-Time (AOT), o que garante que o aplicativo tenha desempenho próximo ao nativo em diferentes plataformas. A facilidade de aprendizado da linguagem, por possuir características semelhantes a outras linguagens amplamente utilizadas, como Java e JavaScript, contribuiu para uma curva de aprendizado suave para a equipe de desenvolvimento.

A integração com o Flutter é outro ponto de destaque, pois, sendo a linguagem nativa do framework, Dart permite uma integração perfeita, aproveitando ao máximo os recursos e funcionalidades disponíveis. A linguagem também oferece suporte robusto à programação assíncrona, com recursos como `async/await` e `Futures`, facilitando o gerenciamento de operações essenciais para aplicativos que dependem de chamadas a APIs e interações com serviços externos.

No contexto deste projeto, o uso do Dart alinhou-se ao objetivo de manter a homogeneidade da stack tecnológica, mantendo o foco no ambiente Google. Essa escolha garantiu consistência na utilização das tecnologias, facilitando a integração entre serviços e ferramentas e proporcionando uma base sólida para o desenvolvimento da aplicação. A linguagem contribuiu significativamente para a manutenção e escalabilidade do código, graças à sua clareza e ao suporte a princípios de programação orientada a objetos, fundamentais para projetos que buscam atender não apenas aos requisitos atuais, mas também permitir futuras expansões e adaptações com agilidade e segurança.

Portanto, a utilização do Dart não apenas atendeu aos requisitos funcionais e não funcionais do projeto, como também potencializou os benefícios decorrentes de uma stack tecnológica homogênea e centrada no ambiente Google. Essa escolha

reforçou os objetivos do projeto, garantindo eficiência, desempenho e qualidade na aplicação desenvolvida.

3.2.5 Códigos

Adoção:

```
import 'package:flutter/material.dart';
import 'pet-details.dart';

class AdoptionPageScreen extends StatelessWidget {
  final List<Map<String, String>> featuredPets = [
    {
      'name': 'Max',
      'breed': 'Golden',
      'age': '4 Meses',
      'gender': 'Macho',
      'imagePath': 'assets/images/adopt1.jpg',
    },
    {
      'name': 'Figo',
      'breed': 'Pixie',
      'age': '8 Meses',
      'gender': 'Macho',
      'imagePath': 'assets/images/adopt.jpg',
    },
    {
      'name': 'Karupi',
      'breed': 'Border collie',
      'age': '1 ano',
      'gender': 'Macho',
      'imagePath': 'assets/images/karupi.jpg',
    },
    {
```



```

    'name': 'Charlie',
    'breed': 'Golden retriever',
    'age': '5 Meses',
    'gender': 'Macho',
    'imagePath': 'assets/images/charlie.jpeg',
  },
  {
    'name': 'Bella',
    'breed': 'Poddle',
    'age': '1 ano',
    'gender': 'Fêmea',
    'imagePath': 'assets/images/bella.png',
  },
];

```

```
@override
```

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: const Color(0xFFFFFCB3C),
      elevation: 0,
      title: const Text(
        'Pets',
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
    leading: IconButton(
      icon: const Icon(Icons.arrow_back, color: Colors.white),
      onPressed: () {
        Navigator.pop(context);
      },
    ),
  ),

```

```

),
body: SafeArea(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      children: [
        // Barra de pesquisa
        Padding(
          padding: const EdgeInsets.only(bottom: 16.0),
          child: TextField(
            decoration: InputDecoration(
              prefixIcon: const Icon(Icons.search),
              hintText: 'Pesquisar',
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
                borderSide: BorderSide.none,
              ),
            ),
            filled: true,
            fillColor: Colors.grey[200],
          ),
        ),
      ],
    ),
  ),
  // Usar Flexible para garantir que o GridView não cause overflow
  Flexible(
    child: GridView.builder(
      gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
        crossAxisSpacing: 20, // Aumentar o espaçamento entre as
        mainAxisSpacing: 20, // Aumentar o espaçamento entre as linhas
        childAspectRatio: 3 / 4,
      ),
      itemCount: featuredPets.length,
      itemBuilder: (context, index) {

```

colunas

```

final pet = featuredPets[index];
return GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PetDetailScreen(
          name: pet['name']!,
          breed: pet['breed']!,
          age: pet['age']!,
          gender: pet['gender']!, // Adicionado parâmetro de gênero
          imagePath: pet['imagePath']!,
        ),
      ),
    );
  },
  child: Container(
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(15),
      boxShadow: [
        BoxShadow(
          color: Colors.grey.withOpacity(0.3),
          spreadRadius: 2,
          blurRadius: 5,
        ),
      ],
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Expanded(
          child: ClipRRect(
            borderRadius: const BorderRadius.only(

```

```

    topLeft: Radius.circular(15),
    topRight: Radius.circular(15),
  ),
  child: Image.asset(
    pet['imagePath']!,
    fit: BoxFit.cover,
    width: double.infinity,
  ),
),
),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        pet['name']!,
        style: const TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 16,
          color: Colors.black,
        ),
        overflow: TextOverflow.ellipsis, // Lidar com overflow de

```

texto

```

    ),
    const SizedBox(height: 4),
    Text(
      pet['breed']!,
      style: const TextStyle(
        fontSize: 14,
        color: Colors.black54,
      ),
      overflow: TextOverflow.ellipsis, // Lidar com overflow de

```

texto

```

        ),
        const SizedBox(height: 4),
        Text(
          pet['age']!,
          style: const TextStyle(
            fontSize: 14,
            color: Colors.black54,
          ),
          overflow: TextOverflow.ellipsis, // Lidar com overflow de
texto
        ),
      ],
    ),
  ],
),
),
);
},
),
),
),
),
),
),
),
);
}
}

```

```

import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:geolocator/geolocator.dart';

```

```

class ExplorePageScreen extends StatefulWidget {

```

```

final double? petLatitude;
final double? petLongitude;
final String? petName;

```

```

ExplorePageScreen({
  this.petLatitude,
  this.petLongitude,
  this.petName,
});

```

```

@override
_ExplorePageScreenState createState() => _ExplorePageScreenState();
}

```

```

class _ExplorePageScreenState extends State<ExplorePageScreen> {
  Position? _currentLocation;
  GoogleMapController? _mapController;
  bool _isLoading = true;
  String? _locationError;

```

```

@override
void initState() {
  super.initState();
  _getCurrentLocation();
}

```

```

Future<void> _getCurrentLocation() async {
  bool serviceEnabled;
  LocationPermission permission;

  try {
    // Verificar se os serviços de localização estão habilitados
    serviceEnabled = await Geolocator.isLocationServiceEnabled();
    if (!serviceEnabled) {

```

```
    setState() {
      _locationError = "Os serviços de localização estão desativados. Ative-os
nas configurações.";
      _isLoading = false;
    });
    await Geolocator.openLocationSettings();
    return;
  }

  // Verificar permissões de localização
  permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    permission = await Geolocator.requestPermission();
    if (permission == LocationPermission.denied) {
      setState() {
        _locationError = "As permissões de localização foram negadas.";
        _isLoading = false;
      });
      return;
    }
  }

  if (permission == LocationPermission.deniedForever) {
    setState() {
      _locationError = "As permissões de localização foram negadas
permanentemente. Permita o acesso à localização nas configurações.";
      _isLoading = false;
    });
    return;
  }

  // Obter a localização atual
  Position position = await Geolocator.getCurrentPosition();
  setState() {
```

```

        _currentLocation = position;
        _isLoading = false;
    });
} catch (e) {
    setState(() {
        _locationError = "Falha ao obter a localização atual: $e";
        _isLoading = false;
    });
}
}
}

```

```

@override
void dispose() {
    _mapController?.dispose();
    super.dispose();
}

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('Mapa', style: TextStyle(color: Colors.white)),
            backgroundColor: const Color(0xFF3161A3),
            leading: IconButton(
                icon: const Icon(Icons.arrow_back, color: Colors.white),
                onPressed: () {
                    Navigator.pop(context);
                },
            ),
        ),
        body: _isLoading
            ? const Center(child: CircularProgressIndicator())
            : (_locationError != null
                ? Center(child: Text(_locationError!, textAlign: TextAlign.center))
            )
    );
}

```



```

        : GoogleMap(
  initialCameraPosition: CameraPosition(
    target: LatLng(_currentLocation!.latitude, _currentLocation!.longitude),
    zoom: 14,
  ),
  myLocationEnabled: true,
  markers: {
    if (widget.petLatitude != null && widget.petLongitude != null)
      Marker(
        markerId: MarkerId('pet_marker'),
        position: LatLng(widget.petLatitude!, widget.petLongitude!),
        infoWindow: InfoWindow(
          title: widget.petName ?? "Localização do Pet",
        ),
      ),
  },
  onMapCreated: (GoogleMapController controller) {
    _mapController = controller;
  },
)),
);
}
}

```

Firestore Options:

firebase_options.dart

```

// Arquivo gerado pelo FlutterFire CLI.
// ignore_for_file: type=lint
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
  show defaultTargetPlatform, kIsWeb, TargetPlatform;

```

```

/// [FirebaseOptions] padrão para uso com seus aplicativos Firebase.
///
/// Exemplo:
/// dart
/// import 'firebase_options.dart';
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
///
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
        return ios;
      case TargetPlatform.macOS:
        return macos;
      case TargetPlatform.windows:
        return windows;
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions não foi configurado para Linux - '
          'você pode reconfigurá-lo executando o FlutterFire CLI novamente.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions não é suportado para esta plataforma.',
        );
    }
  }
}

```

```
}

```

```
static const FirebaseOptions web = FirebaseOptions(
  apiKey: 'AlzaSyCfE3dE_gpQcibDRa_o4oQicDyGsHgHj_Y',
  appId: '1:601242567291:web:5e4fea994eb6cd876658de',
  messagingSenderId: '601242567291',
  projectId: 'ajude-uma-vida-44c9e',
  authDomain: 'ajude-uma-vida-44c9e.firebaseio.com',
  storageBucket: 'ajude-uma-vida-44c9e.firebaseio.com',
  measurementId: 'G-SG711XKG74',
);

```

```
static const FirebaseOptions android = FirebaseOptions(
  apiKey: 'AlzaSyDVBHAgeUGBHR0udpimclyd4l_JPBzBbPE',
  appId: '1:601242567291:android:1d4bb602485d37276658de',
  messagingSenderId: '601242567291',
  projectId: 'ajude-uma-vida-44c9e',
  storageBucket: 'ajude-uma-vida-44c9e.firebaseio.com',
);

```

```
static const FirebaseOptions ios = FirebaseOptions(
  apiKey: 'AlzaSyADqirW_qWCc6lDgau1JsJVuO7e0rE-DB4',
  appId: '1:601242567291:ios:9f6c00d930f7a61a6658de',
  messagingSenderId: '601242567291',
  projectId: 'ajude-uma-vida-44c9e',
  storageBucket: 'ajude-uma-vida-44c9e.firebaseio.com',
  iosClientId: '601242567291-
gvf52miagj59b9fcdt5itr4hhp1ud9io.apps.googleusercontent.com',
  iosBundleId: 'com.example.untitled1',
);

```

```
static const FirebaseOptions macos = FirebaseOptions(
  apiKey: 'AlzaSyADqirW_qWCc6lDgau1JsJVuO7e0rE-DB4',
  appId: '1:601242567291:ios:9f6c00d930f7a61a6658de',

```

```

    messagingSenderId: '601242567291',
    projectId: 'ajude-uma-vida-44c9e',
    storageBucket: 'ajude-uma-vida-44c9e.firebaseio.com',
    iosClientId: '601242567291-
gvf52miagj59b9fcdt5itr4hhp1ud9io.apps.googleusercontent.com',
    iosBundleId: 'com.example.untitled1',
  );

```

```

static const FirebaseOptions windows = FirebaseOptions(
  apiKey: 'AlzaSyCfE3dE_gpQcibDRa_o4oQicDyGsHgHj_Y',
  appId: '1:601242567291:web:402d2ed3b86d71e56658de',
  messagingSenderId: '601242567291',
  projectId: 'ajude-uma-vida-44c9e',
  authDomain: 'ajude-uma-vida-44c9e.firebaseio.com',
  storageBucket: 'ajude-uma-vida-44c9e.firebaseio.com',
  measurementId: 'G-1LH2DVE412',
);
}

```

forget_password.dart

```

import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

class ForgotPasswordScreen extends StatefulWidget {
  const ForgotPasswordScreen({Key? key}) : super(key: key);

  @override
  State<ForgotPasswordScreen> createState() =>
    _ForgotPasswordScreenState();
}

class _ForgotPasswordScreenState extends State<ForgotPasswordScreen> {
  final TextEditingController _emailController = TextEditingController();

```

```

// Método para enviar o e-mail de redefinição de senha
Future<void> _sendPasswordResetEmail() async {
  try {
    await FirebaseAuth.instance.sendPasswordResetEmail(
      email: _emailController.text.trim(),
    );
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('O email foi enviado. Verifique a Inbox.')),
    );
    Navigator.pop(context); // Retorna para a tela anterior após o envio do e-
mail
  } on FirebaseAuthException catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(e.message ?? 'Falha. Tente novamente.')),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Esqueceu a senha?'),
      backgroundColor: const Color(0xFF3161A3),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          const Text(
            'Informe o e-mail cadastrado para redefinir a senha.',
            textAlign: TextAlign.center,

```

```

        style: TextStyle(fontSize: 16),
      ),
      const SizedBox(height: 20),
      TextField(
        controller: _emailController,
        decoration: InputDecoration(
          hintText: 'Email',
          prefixIcon: const Icon(Icons.email),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(15),
          ),
        ),
      ),
      const SizedBox(height: 20),
      ElevatedButton(
        onPressed: _sendPasswordResetEmail,
        style: ElevatedButton.styleFrom(
          backgroundColor: const Color(0xFF3161A3),
          padding: const EdgeInsets.symmetric(vertical: 16),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(15),
          ),
        ),
        child: const Text(
          'Confirmar',
          style: TextStyle(fontSize: 18),
        ),
      ),
    ],
  ),
);
}
}

```

login_input:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:flutter_facebook_auth/flutter_facebook_auth.dart';
import 'register1.dart'; // Importa RegisterStep1Screen
import 'home_screen.dart'; // Importa HomeScreen
import 'forget-password.dart'; // Importa a tela ForgotPasswordScreen

class LoginInputScreen extends StatefulWidget {
  const LoginInputScreen({Key? key}) : super(key: key);

  @override
  State<LoginInputScreen> createState() => _LoginInputScreenState();
}

class _LoginInputScreenState extends State<LoginInputScreen> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  bool _isPasswordVisible = false;

  // Método de login com Google
  Future<User?> _signInWithGoogle() async {
    final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();
    if (googleUser != null) {
      final GoogleSignInAuthentication googleAuth =
        await googleUser.authentication;
      final credential = GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );
    }
  }
}
```

```

        return (await FirebaseAuth.instance.signInWithCredential(credential)).user;
    }
    return null;
}

// Método de login com Facebook
Future<User?> _signInWithFacebook() async {
    final LoginResult result = await FacebookAuth.instance.login();

    if (result.status == LoginStatus.success) {
        final OAuthCredential facebookAuthCredential =
            FirebaseAuthProvider.credential(result.accessToken!.token);
        return (await FirebaseAuth.instance
            .signInWithCredential(facebookAuthCredential)
            .user;
    } else {
        ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text(result.message ?? 'Falha no login com
Facebook')),
        );
    }
    return null;
}

// Método de login com email e senha
Future<void> _signInWithEmailPassword() async {
    try {
        final UserCredential userCredential = await FirebaseAuth.instance
            .signInWithEmailAndPassword(
                email: _emailController.text.trim(),
                password: _passwordController.text,
            );
        final user = userCredential.user;
        if (user != null) {

```



```

Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => const HomePageScreen(),
  ),
);
}
} on FirebaseAuthException catch (e) {
  // Tratamento de erro (ex.: usuário não encontrado, senha incorreta)
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(e.message ?? 'Falha no login')),
  );
}
}

```

```

void _togglePasswordVisibility() {
  setState(() {
    _isPasswordVisible = !_isPasswordVisible;
  });
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Stack(
      children: [
        // Fundo azul
        Container(
          color: const Color(0xFF3161A3),
          height: 350,
          width: double.infinity,
        ),
        // Conteúdo rolável
        SingleChildScrollView(

```

```
padding: const EdgeInsets.all(16.0),
child: Column(
  children: [
    const SizedBox(height: 100),
    Stack(
      clipBehavior: Clip.none,
      alignment: Alignment.topCenter,
      children: [
        Container(
          margin: const EdgeInsets.only(top: 100),
          padding: const EdgeInsets.all(16.0),
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(25.0),
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              const SizedBox(height: 100),
              const Text(
                'Login',
                style: TextStyle(
                  fontSize: 32,
                  fontWeight: FontWeight.bold,
                  color: Color(0xFF3161A3),
                ),
                textAlign: TextAlign.center,
              ),
              const SizedBox(height: 30),
              // Entrada de email
              TextField(
                controller: _emailController,
                decoration: InputDecoration(
                  hintText: 'Email',
```

```

    prefixIcon: const Icon(Icons.email),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(15),
    ),
  ),
),
const SizedBox(height: 20),
// Entrada de senha
TextField(
  controller: _passwordController,
  obscureText: !_isPasswordVisible,
  decoration: InputDecoration(
    hintText: 'Senha',
    prefixIcon: const Icon(Icons.lock),
    suffixIcon: IconButton(
      icon: Icon(
        _isPasswordVisible
          ? Icons.visibility
          : Icons.visibility_off,
      ),
      onPressed: _togglePasswordVisibility,
    ),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(15),
    ),
  ),
),
const SizedBox(height: 10),
Align(
  alignment: Alignment.centerRight,
  child: GestureDetector(
    onTap: () {
      Navigator.push(
        context,

```

```

MaterialPageRoute(
  builder: (context) =>
    const ForgotPasswordScreen(),
),
);
},
child: const Text(
  'Esqueceu a senha?',
  style: TextStyle(
    color: Color(0xFF3161A3),
    fontWeight: FontWeight.bold,
  ),
),
),
),
const SizedBox(height: 30),
// Botão de login
ElevatedButton(
  onPressed: _signInWithEmailPassword,
  style: ElevatedButton.styleFrom(
    backgroundColor: const Color(0xFF3161A3),
    padding: const EdgeInsets.symmetric(vertical: 16),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(15),
    ),
  ),
  child: const Text(
    'Login',
    style: TextStyle(
      fontSize: 18,
      color: Colors.white, // Define a cor do texto para branco
    ),
  ),
),
),
),

```

```

const SizedBox(height: 20),
// Ou continue com
Row(
  children: const [
    Expanded(child: Divider()),
    Padding(
      padding: EdgeInsets.symmetric(horizontal: 8.0),
      child: Text('ou continue com'),
    ),
    Expanded(child: Divider()),
  ],
),
const SizedBox(height: 20),
// Botões de redes sociais
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    _buildSocialButton(
      icon: Icons.g_mobiledata,
      onPressed: () async {
        User? user = await _signInWithGoogle();
        if (user != null) {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) =>
                const HomePageScreen(),
            ),
          );
        }
      },
    ),
    _buildSocialButton(
      icon: Icons.facebook,

```

```

onPressed: () async {
  User? user = await _signInWithFacebook();
  if (user != null) {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          const HomePageScreen(),
      ),
    );
  }
},
),
],
),
const SizedBox(height: 20),
// Link para cadastro
const Text(
  "Não tem uma conta?",
  textAlign: TextAlign.center,
),
const SizedBox(height: 10),
GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          const RegisterStep1Screen(),
      ),
    );
  },
  child: const Text(
    'Cadastre-se',

```

```

        style: TextStyle(
          color: Color(0xFF3161A3),
          fontWeight: FontWeight.bold,
        ),
        textAlign: TextAlign.center,
      ),
    ),
  ],
),
),
Positioned(
  top: -140,
  child: Image.asset(
    'assets/images/dog.png',
    height: 300,
  ),
),
],
),
],
),
),
);
}

```

```

Widget _buildSocialButton(
  {required IconData icon, required VoidCallback onPressed}) {
  return ElevatedButton(
    onPressed: onPressed,
    style: ElevatedButton.styleFrom(
      shape: const CircleBorder(),
      padding: const EdgeInsets.all(15),
    ),
  );
}

```

```

        backgroundColor: Colors.white,
      ),
      child: Icon(icon, color: const Color(0xFF3161A3)),
    );
  }
}

```

home_screen.dart

```

import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'pet-details.dart';
import 'favorites.dart';
import 'adopt.dart';
import 'explore.dart';
import 'rescue_welcome.dart';
import 'NGOs.dart';
import 'rescue_status.dart';

class HomePageScreen extends StatefulWidget {
  const HomePageScreen({Key? key}) : super(key: key);

  @override
  _HomePageScreenState createState() => _HomePageScreenState();
}

class _HomePageScreenState extends State<HomePageScreen> {
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  final List<Map<String, dynamic>> favoritePets = [];
  final Set<String> favoritedPetNames = {};
  String _userName = 'Usuário';
  String _userInitials = '';
  String _userPhone = '';

```



```
List<Map<String, dynamic>> featuredPets = [  
  {  
    'name': 'Max',  
    'breed': 'Golden',  
    'age': '4 Meses',  
    'gender': 'Macho',  
    'imagePath': 'assets/images/adopt1.jpg',  
  },  
  {  
    'name': 'Figo',  
    'breed': 'Pixie',  
    'age': '8 Meses',  
    'gender': 'Macho',  
    'imagePath': 'assets/images/adopt.jpg',  
  },  
  {  
    'name': 'Karupi',  
    'breed': 'Border collie',  
    'age': '1 ano',  
    'gender': 'Macho',  
    'imagePath': 'assets/images/karupi.jpg',  
  },  
  {  
    'name': 'Charlie',  
    'breed': 'Golden retriever',  
    'age': '5 Meses',  
    'gender': 'Macho',  
    'imagePath': 'assets/images/charlie.jpeg',  
  },  
  {  
    'name': 'Bella',  
    'breed': 'Poodle',  
    'age': '1 ano',
```

```

    'gender': 'Fêmea',
    'imagePath': 'assets/images/bella.png',
  },
];

```

```
@override
```

```

void initState() {
  super.initState();
  _getUserInfo();
  _getFeaturedPetsFromFirebase();
}

```

```
// Método para obter as informações do usuário
```

```

Future<void> _getUserInfo() async {
  try {
    User? user = FirebaseAuth.instance.currentUser;
    if (user != null) {
      DocumentSnapshot userDoc = await
      FirebaseFirestore.instance.collection('users').doc(user.uid).get();
      if (userDoc.exists) {
        setState(() {
          _userName = userDoc['fullName'] ?? 'Usuário';
          _userPhone = userDoc['phone'] ?? '+55 11 12345-67890';
          _userInitials = _getInitials(_userName);
        });
      }
    }
  } catch (e) {
    print("Falha ao obter informações do usuário: $e");
  }
}

```

```
// Método para obter os pets em destaque do Firebase
```

```

Future<void> _getFeaturedPetsFromFirebase() async {

```

```

// Lógica para obter pets do Firebase (omitida por brevidade)
}

// Método para obter as iniciais do nome do usuário
String _getInitials(String name) {
  List<String> nameParts = name.split(' ');
  String initials = "";
  for (var part in nameParts) {
    if (part.isNotEmpty) {
      initials += part[0];
    }
  }
  return initials.toUpperCase();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _scaffoldKey,
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      leading: IconButton(
        icon: const Icon(Icons.arrow_back, color: Color(0xFF3161A3)),
        onPressed: () {
          Navigator.pop(context);
        },
      ),
    actions: [
      IconButton(
        icon: const Icon(Icons.notifications, color: Color(0xFF3161A3)),
        onPressed: () {},
      ),
      IconButton(

```

```

    icon: const Icon(Icons.menu, color: Color(0xFF3161A3)),
    onPressed: () {
      _scaffoldKey.currentState?.openEndDrawer();
    },
  ),
],
),
endDrawer: Drawer(
  child: ListView(
    padding: EdgeInsets.zero,
    children: [
      DrawerHeader(
        decoration: const BoxDecoration(
          color: Color(0xFF3161A3),
        ),
        child: Row(
          children: [
            CircleAvatar(
              radius: 32,
              backgroundColor: Colors.black,
              child: Text(
                _userInitials,
                style: const TextStyle(color: Colors.white, fontSize: 24),
              ),
            ),
            const SizedBox(width: 16),
            Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Olá, $_userName!',
                  style: const TextStyle(color: Colors.white, fontSize: 18),
                ),
              ],
            ),
          ],
        ),
      ),
    ],
  ),
),

```

```
    const SizedBox(height: 4),  
    Text(  
      _userPhone,  
      style: const TextStyle(color: Colors.white70),  
    ),  
  ],  
),  
],  
) ,  
) ,  
_buildDrawerItem(Icons.home, 'Início', () {}),  
const SizedBox(height: 8),  
_buildDrawerItem(Icons.notifications, 'Notificações', () {}),  
const SizedBox(height: 8),  
const SizedBox(height: 8),  
_buildDrawerItem(Icons.search, 'Pesquisar', () {}),  
const SizedBox(height: 8),  
_buildDrawerItem(Icons.pets, 'Pets', () {}),  
const SizedBox(height: 8),  
_buildDrawerItem(Icons.support, 'Ajuda', () {}),  
],  
) ,  
) ,  
body: SafeArea(  
  child: SingleChildScrollView(  
    padding: const EdgeInsets.all(16.0),  
    child: Column(  
      crossAxisAlignment: CrossAxisAlignment.start,  
      children: [  
        Container(  
          padding: const EdgeInsets.all(16),  
          decoration: BoxDecoration(  
            color: const Color(0xFFFFFCB3C),  
            borderRadius: BorderRadius.circular(20),
```

```

),
child: Row(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Expanded(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text(
            'Sua ajuda salva vidas!',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
          ),
          const SizedBox(height: 16),
          ElevatedButton(
            onPressed: () {},
            style: ButtonStyle(
              backgroundColor: MaterialStateProperty.all<Color>(const
Color(0xFF3161A3)),
              shape:
MaterialStateProperty.all<RoundedRectangleBorder>(
                RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(25),
                ),
              ),
            ),
            child: const Text(
              'Saiba mais',
              style: TextStyle(color: Colors.white),
            ),
          ),
        ],
      ),
    ),
  ],
),

```

```

    ],
  ),
),
Image.asset(
  'assets/images/cats.png',
  height: 120,
  width: 170,
  fit: BoxFit.cover,
),
],
),
),
const SizedBox(height: 24),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    const Text(
      'Pets em Destaque',
      style: TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 18,
        color: Colors.black,
      ),
    ),
  ),
GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => AdoptionPageScreen(),
      ),
    );
  },
  child: const Text(

```

```

    'Ver mais',
    style: TextStyle(
      color: Color(0xFF3161A3),
      fontWeight: FontWeight.bold,
    ),
  ),
),
],
),
const SizedBox(height: 16),
SizedBox(
  height: 250,
  child: ListView.builder(
    scrollDirection: Axis.horizontal,
    itemCount: featuredPets.length,
    itemBuilder: (context, index) {
      final pet = featuredPets[index];
      return Padding(
        padding: const EdgeInsets.only(right: 16.0),
        child: _buildPetCard(
          pet['name'],
          pet['breed'],
          pet['age'],
          pet['gender'],
          pet['imagePath'],
        ),
      );
    },
  ),
),
const SizedBox(height: 24),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: const [

```



```

Text(
  'ONGs',
  style: TextStyle(
    fontWeight: FontWeight.bold,
    fontSize: 18,
    color: Colors.black,
  ),
),
Text(
  'Ver mais',
  style: TextStyle(
    color: Color(0xFF3161A3),
    fontWeight: FontWeight.bold,
  ),
),
],
),
const SizedBox(height: 16),
SizedBox(
  height: 120,
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: [
      _buildShelterCard('Bem me quer', '1.5 Km',
Colors.amber.shade200),
      const SizedBox(width: 16),
      _buildShelterCard('Sete vidas', '4 Km', Colors.green.shade200),
      const SizedBox(width: 16),
      _buildShelterCard('Adote um amigo', '3.5 Km',
Colors.blue.shade200),
    ],
  ),
),
const SizedBox(height: 16),

```

```

    ],
  ),
),
bottomNavigationBar: ClipRRect(
  borderRadius: const BorderRadius.only(
    topLeft: Radius.circular(20),
    topRight: Radius.circular(20),
  ),
  child: BottomAppBar(
    color: const Color(0xFF3161A3),
    elevation: 0,
    child: Container(
      height: 80,
      color: const Color(0xFF3161A3),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          _buildBottomNavigationItem(Icons.home, 'Início', true),
          _buildBottomNavigationItem(Icons.pets, 'Adoção', false),
          _buildBottomNavigationItem(Icons.explore, 'Explorar', false,
isMiddle: true),
          _buildBottomNavigationItem(Icons.volunteer_activism, 'Resgate',
false),
          _buildBottomNavigationItem(Icons.favorite, 'Favoritos', false),
        ],
      ),
    ),
  ),
);
}

```

```
Widget _buildDrawerItem(IconData icon, String title, VoidCallback onTap) {
```

```

return ListTile(
  leading: Icon(icon, color: Colors.black),
  title: Text(title),
  onTap: onTap,
);
}

```

```

Widget _buildPetCard(String name, String breed, String age, String gender,
String imagePath) {
  ValueNotifier<bool> isFavorited =
ValueNotifier(favoritedPetNames.contains(name));

```

```

return GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => PetDetailScreen(
          name: name,
          breed: breed,
          age: age,
          gender: gender,
          imagePath: imagePath,
        ),
      ),
    );
  },
  child: Container(
    width: 180,
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(15),
      boxShadow: [
        BoxShadow(

```

```

        color: Colors.grey.withOpacity(0.3),
        spreadRadius: 2,
        blurRadius: 5,
      ),
    ],
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      Stack(
        children: [
          ClipRRect(
            borderRadius: const BorderRadius.only(
              topLeft: Radius.circular(15),
              topRight: Radius.circular(15),
            ),
            child: imagePath.startsWith('assets/')
              ? Image.asset(
                  imagePath,
                  height: 130,
                  width: double.infinity,
                  fit: BoxFit.cover,
                )
              : Image.network(
                  imagePath,
                  height: 130,
                  width: double.infinity,
                  fit: BoxFit.cover,
                ),
          ),
        ],
      ),
      Positioned(
        top: 8,
        left: 8,
        child: ValueListenableBuilder<bool>(

```

```

valueListenable: isFavorited,
builder: (context, value, child) {
  return IconButton(
    icon: Icon(
      value ? Icons.favorite : Icons.favorite_border,
      color: value ? Colors.red : Colors.blue.shade900,
    ),
    onPressed: () {
      if (value) {
        favoritedPetNames.remove(name);
        favoritePets.removeWhere((pet) => pet['name'] == name);
      } else {
        favoritedPetNames.add(name);
        favoritePets.add({'name': name, 'breed': breed, 'age': age,
'gender': gender, 'imagePath': imagePath});
      }
      isFavorited.value = !value;
    },
  );
},
),
),
],
),
const SizedBox(height: 8),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 8.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        name,
        style: const TextStyle(
          fontWeight: FontWeight.bold,

```

```

      fontSize: 16,
      color: Colors.black,
    ),
  ),
  Icon(
    gender.toLowerCase() == 'macho' ? Icons.male : Icons.female,
    color: gender.toLowerCase() == 'macho' ? Colors.blue.shade900 :
Colors.pink,
    size: 24,
  ),
],
),
),
const SizedBox(height: 4),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 8.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        breed,
        style: const TextStyle(fontSize: 14, color: Colors.black54),
      ),
      const SizedBox(height: 4),
      Text(
        age,
        style: const TextStyle(fontSize: 14, color: Colors.black54),
      ),
    ],
  ),
),
],
),
),

```

```

    );
  }

Widget _buildShelterCard(String name, String distance, Color
backgroundColor) {
  return GestureDetector(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => NGOsScreen(name: name),
        ),
      );
    },
    child: Container(
      width: 140,
      padding: const EdgeInsets.all(8),
      decoration: BoxDecoration(
        color: backgroundColor,
        borderRadius: BorderRadius.circular(15),
        boxShadow: [
          BoxShadow(
            color: Colors.grey.withOpacity(0.3),
            spreadRadius: 2,
            blurRadius: 5,
          ),
        ],
      ),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          const Icon(Icons.pets, color: Colors.white),
          const SizedBox(height: 8),
          Text(

```

```

        name,
        style: const TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 14,
          color: Colors.white,
        ),
        textAlign: TextAlign.center,
      ),
      const SizedBox(height: 8),
      Text(
        distance,
        style: const TextStyle(
          fontSize: 12,
          color: Colors.white,
        ),
        textAlign: TextAlign.center,
      ),
    ],
  ),
);
}

```

```

Widget _buildBottomNavigationItem(IconData icon, String label, bool
isSelected, {bool isMiddle = false}) {
  return GestureDetector(
    onTap: () {
      if (label == 'Adoção') {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => AdoptionPageScreen(),
          ),
        );
      }
    }
  );
}

```



```

} else if (label == 'Explorar') {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => ExplorePageScreen(),
    ),
  );
} else if (label == 'Resgate') {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => RescueWelcomeScreen(),
    ),
  );
} else if (label == 'Favoritos') {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => FavoritesScreen(favoritePets: favoritePets),
    ),
  );
}
},
child: Column(
  mainAxisAlignment: MainAxisAlignment.min,
  children: [
    isMiddle
    ? Container(
      decoration: BoxDecoration(
        color: Colors.white,
        shape: BoxShape.circle,
      ),
    ),
    child: Icon(
      icon,

```

```

        color: const Color(0xFF3161A3),
        size: 30,
      ),
    )
    : Icon(
      icon,
      color: Colors.white,
    ),
    if (!isMiddle)
      Text(
        label,
        style: const TextStyle(
          color: Colors.white,
          fontSize: 12,
        ),
      ),
    ],
  ),
);
}
}

```

```

import 'package:flutter/material.dart';
import 'rescue.dart'; // Importa RescuePageScreen
import 'rescue_status.dart'; // Importa ReportUpdateScreen

```

```

class RescueWelcomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(

```

```

crossAxisAlignment: CrossAxisAlignment.center,
children: [
  Align(
    alignment: Alignment.topLeft,
    child: IconButton(
      icon: Icon(Icons.arrow_back),
      onPressed: () {
        // Volta para a tela anterior
        Navigator.pop(context);
      },
    ),
  ),
  SizedBox(height: 30), // Espaço acima do texto
  Text(
    'Ajude um Pet',
    style: TextStyle(
      fontSize: 28, // Tamanho de fonte maior
      fontWeight: FontWeight.w400,
      color: Colors.black,
    ),
  ),
  Text(
    'Resgate já!',
    style: TextStyle(
      fontSize: 34, // Tamanho de fonte maior para ênfase
      fontWeight: FontWeight.bold,
      color: Color(0xFF3161A3),
    ),
  ),
  SizedBox(height: 30), // Ajusta o espaço abaixo do texto
  // Usa Flexible para garantir que a imagem se ajuste ao espaço
  disponível
  Flexible(
    child: Image.asset(

```

disponível

```

'assets/images/rescue.png', // Caminho do asset da imagem
fit: BoxFit.contain, // Ajusta o tamanho para não transbordar
),
),
 SizedBox(height: 30), // Espaço antes dos botões
SizedBox(
  width: double.infinity, // Largura do botão para preencher o espaço

  child: ElevatedButton(
    onPressed: () {
      // Navega para a tela RescuePageScreen
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => RescuePageScreen(),
        ),
      );
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Color(0xFF3161A3), // Cor azul do botão
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(20),
      ),
      padding: EdgeInsets.symmetric(vertical: 16),
    ),
    child: Text(
      'Resgatar',
      style: TextStyle(
        fontSize: 16,
        color: Colors.white,
        fontWeight: FontWeight.bold,
      ),
    ),
  ),
),
),

```

```

),
  SizedBox(height: 16),
  SizedBox(
    width: double.infinity, // Largura do botão para preencher o espaço
    disponível
    child: ElevatedButton(
      onPressed: () {
        // Navega para a tela ReportUpdateScreen
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ReportUpdateScreen(), // Atualizado aqui
          ),
        );
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Color(0xFFFFCB3C), // Cor amarela do botão
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(20),
        ),
        padding: EdgeInsets.symmetric(vertical: 16),
      ),
      child: Text(
        'Status de Resgate',
        style: TextStyle(
          fontSize: 16,
          color: Colors.black,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),
  SizedBox(height: 20), // Espaço na parte inferior
],

```

```

    ),
  ),
),
);
}
}

```

```
import 'package:flutter/material.dart';
```

```

class ReportUpdateScreen extends StatelessWidget {
  const ReportUpdateScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: const Color(0xFFFFFCB3C), // Cor de fundo do AppBar
        elevation: 0, // Remove a sombra do AppBar
        leading: IconButton(
          icon: const Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            // Volta para a tela anterior
            Navigator.pop(context);
          },
        ),
        title: const Text(
          'Status de Resgate', // Título no AppBar
          style: TextStyle(color: Colors.black),
        ),
        centerTitle: true, // Centraliza o título
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(

```

```

crossAxisAlignment: CrossAxisAlignment.start,
children: [
  Center(
    child: CircleAvatar(
      radius: 60, // Tamanho do avatar
      backgroundColor: Colors.grey[300], // Cor de fundo do avatar
      child: ClipOval(
        child: Image.asset(
          'assets/images/charlie.jpeg', // Substitua pelo caminho da sua
imagem
          fit: BoxFit.cover,
          width: 150, // Largura da imagem
          height: 150, // Altura da imagem
        ),
      ),
    ),
  ),
  const SizedBox(height: 24), // Espaço entre o avatar e o próximo
conteúdo
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: const [
      Text(
        'Status', // Texto do lado esquerdo
        style: TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
      Text(
        'Data de Registro', // Texto do lado direito
        style: TextStyle(
          fontSize: 16,

```

```

        fontWeight: FontWeight.bold,
        color: Colors.black,
    ),
),
],
),
const SizedBox(height: 8), // Espaço entre as linhas
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: const [
    Text(
      'Acompanhamento', // Texto do lado esquerdo
      style: TextStyle(
        fontSize: 14,
        color: Colors.black54,
      ),
    ),
    Text(
      '09/16/2024', // Data no lado direito
      style: TextStyle(
        fontSize: 14,
        color: Colors.black54,
      ),
    ),
  ],
),
const SizedBox(height: 24), // Espaço antes da lista de status
Expanded(
  child: ListView(
    children: [
      _buildStatusStep('Chamado criado', true), // Status da primeira
etapa
      _buildStatusStep('ONG entrou em contato', true), // Status da
segunda etapa

```



```

        _buildStatusStep('ONG à caminho', true), // Status da terceira etapa
        _buildStatusStep('Pet resgatado', true), // Status da quarta etapa
        _buildStatusStep('Pronto para adoção', true), // Status da quinta
etapa
    ],
    ),
    ),
    ],
    ),
    ),
bottomNavigationBar: ClipRRect(
  borderRadius: const BorderRadius.only(
    topLeft: Radius.circular(20), // Borda arredondada no topo esquerdo
    topRight: Radius.circular(20), // Borda arredondada no topo direito
  ),
  child: BottomAppBar(
    color: const Color(0xFF3161A3), // Cor de fundo do BottomAppBar
    elevation: 0, // Remove a sombra
    child: Container(
      height: 80, // Altura do BottomAppBar
      color: const Color(0xFF3161A3),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          _buildBottomNavigationItem(Icons.home, 'Início', true), // Item de
navegação "Início"
          _buildBottomNavigationItem(Icons.pets, 'Adoção', false), // Item de
navegação "Adoção"
          _buildBottomNavigationItem(Icons.explore, 'Explorar', false,
isMiddle: true), // Item de navegação "Explorar" com destaque
          _buildBottomNavigationItem(Icons.volunteer_activism, 'Resgatar',
false), // Item de navegação "Resgatar"
          _buildBottomNavigationItem(Icons.favorite, 'Favoritos', false), // Item
de navegação "Favoritos"

```

```

    ],
  ),
),
),
),
);
}

```

// Método para construir um item de etapa de status

```
Widget _buildStatusStep(String label, bool isCompleted) {
```

```
  return Row(
```

```
    children: [
```

```
      Column(
```

```
        children: [
```

```
          Icon(
```

```
            Icons.check_circle, // Ícone de círculo com marca de verificação
```

```
            color: isCompleted ? Color(0xFFFFCB3C) : Colors.grey, // Cor
```

dependendo se a etapa está concluída

```
          ),
```

```
          Container(
```

```
            width: 2, // Largura da linha vertical
```

```
            height: 30, // Altura da linha vertical
```

```
            color: isCompleted ? Color(0xFFFFCB3C) : Colors.grey, // Cor da linha
```

vertical

```
          ),
```

```
        ],
```

```
      ),
```

```
      const SizedBox(width: 16), // Espaço entre o ícone e o texto
```

```
      Text(
```

```
        label, // Texto da etapa
```

```
        style: TextStyle(
```

```
          fontSize: 16,
```

```
          fontWeight: FontWeight.bold,
```

color: isCompleted ? Colors.black : Colors.black54, // Cor dependendo se a etapa está concluída

```

    ),
  ),
],
);
}

```

// Método para construir um item de navegação no BottomAppBar

```

Widget _buildBottomNavigationItem(IconData icon, String label, bool
isSelected, {bool isMiddle = false}) {

```

```

  return GestureDetector(
    onTap: () {}, // Ação ao tocar no item
    child: Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        isMiddle
          ? Container(
              decoration: BoxDecoration(
                color: Colors.white, // Cor de fundo para o ícone destacado
                shape: BoxShape.circle, // Forma circular
              ),
              child: Icon(
                icon,
                color: const Color(0xFF3161A3), // Cor do ícone
                size: 30,
              ),
            )
          : Icon(
              icon,
              color: Colors.white, // Cor do ícone para itens não destacados
            ),
        if (!isMiddle)
          Text(

```

```

        label, // Texto do item de navegação
        style: const TextStyle(
          color: Colors.white,
          fontSize: 12,
        ),
      ),
    ],
  ),
);
}
}

```

```

import 'package:flutter/material.dart';
import 'register.dart'; // Importa a tela final de registro

```

```

class RegisterStep2Screen extends StatefulWidget {
  final String userType; // Tipo de usuário recebido como parâmetro

```

```

  const RegisterStep2Screen({Key? key, required this.userType}) : super(key:
key);

```

```

  @override
  State<RegisterStep2Screen> createState() => _RegisterStep2ScreenState();
}

```

```

class _RegisterStep2ScreenState extends State<RegisterStep2Screen> {
  // Controladores de texto para capturar o nome completo, celular e localização
  final TextEditingController _fullNameController = TextEditingController();
  final TextEditingController _phoneController = TextEditingController();
  final TextEditingController _locationController = TextEditingController();
  bool isLoading = false; // Variável para controlar o estado de carregamento

```

```

  @override
  void dispose() {

```

```

// Limpa os controladores ao sair da tela
_fullNameController.dispose();
_phoneController.dispose();
_locationController.dispose();
super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: const Color(0xFF3161A3), // Cor de fundo do AppBar
      leading: IconButton(
        icon: const Icon(Icons.arrow_back, color: Colors.white), // Ícone de voltar
        onPressed: () {
          Navigator.pop(context); // Volta para a tela anterior
        },
      ),
      title: const Text('Crie sua conta, é rápido!', style: TextStyle(color:
Colors.white)), // Título do AppBar
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0), // Espaçamento ao redor do conteúdo
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start, // Alinha o conteúdo no
início da coluna
        children: [
          const SizedBox(height: 20), // Espaço entre o topo e o primeiro campo
          TextField(
            controller: _fullNameController, // Controlador para o campo de nome
            decoration: const InputDecoration(
              labelText: 'Nome Completo', // Rótulo do campo
              hintText: 'Nome e Sobrenome', // Texto de dica
            ),
          ),
        ],
      ),
    ),
  );
}

```

```

        filled: true,
        fillColor: Color(0xFFE0E0E0), // Cor de fundo do campo
        border: OutlineInputBorder(
          borderSide: BorderSide.none,
          borderRadius: BorderRadius.all(Radius.circular(10)), // Borda
arredondada
        ),
      ),
    ),
    const SizedBox(height: 20), // Espaço entre os campos
    TextField(
      controller: _phoneController, // Controlador para o campo de celular
      decoration: const InputDecoration(
        labelText: 'Celular', // Rótulo do campo
        hintText: '(99) 9999 9999', // Texto de dica
        filled: true,
        fillColor: Color(0xFFE0E0E0), // Cor de fundo do campo
        border: OutlineInputBorder(
          borderSide: BorderSide.none,
          borderRadius: BorderRadius.all(Radius.circular(10)), // Borda
arredondada
        ),
      ),
    ),
    const SizedBox(height: 20), // Espaço entre os campos
    TextField(
      controller: _locationController, // Controlador para o campo de
localização
      decoration: const InputDecoration(
        labelText: 'Localização', // Rótulo do campo
        hintText: 'Bairro, Cidade - Estado', // Texto de dica
        prefixIcon: Icon(Icons.location_on), // Ícone de localização
        filled: true,
        fillColor: Color(0xFFE0E0E0), // Cor de fundo do campo

```

```

border: OutlineInputBorder(
  borderSide: BorderSide.none,
  borderRadius: BorderRadius.all(Radius.circular(10)), // Borda
arredondada
),
),
),
const SizedBox(height: 30), // Espaço antes do botão
ElevatedButton(
  onPressed: isLoading ? null : () {
    // Navega para a próxima tela de registro
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => RegisterScreen(
          userType: widget.userType, // Passa o tipo de usuário
          fullName: _fullNameController.text.trim(), // Passa o nome
completo
          phone: _phoneController.text.trim(), // Passa o celular
          location: _locationController.text.trim(), // Passa a localização
        ),
      ),
    );
  },
  style: ElevatedButton.styleFrom(
    backgroundColor: const Color(0xFF3161A3), // Cor de fundo do
botão
    padding: const EdgeInsets.symmetric(vertical: 16.0, horizontal: 32.0),
    // Espaçamento interno do botão
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10), // Borda arredondada do
botão
    ),
  ),
),

```

```

        child: const Text(
          'Continuar', // Texto do botão
          style: TextStyle(fontSize: 18, color: Colors.white), // Estilo do texto do
botão
        ),
      ),
      const Spacer(), // Espaço flexível para empurrar os ícones para o fundo
      Row(
        mainAxisAlignment: MainAxisAlignment.center, // Centraliza os ícones
        children: const [
          Icon(Icons.circle, size: 10, color: Colors.blue), // Primeiro ícone
          SizedBox(width: 8), // Espaço entre os ícones
          Icon(Icons.circle, size: 10, color: Colors.grey), // Segundo ícone
        ],
      ),
      const SizedBox(height: 16), // Espaço no final
    ],
  ),
);
}
}

```

```
import 'package:flutter/material.dart';
```

```
import 'register2.dart'; // Importa a próxima tela de registro
```

```
class RegisterStep1Screen extends StatelessWidget {
```

```
  const RegisterStep1Screen({Key? key}) : super(key: key);
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Scaffold(
```

```
      appBar: AppBar(
```

```
        backgroundColor: const Color(0xFF3161A3), // Cor de fundo do AppBar

```



```

leading: IconButton(
  icon: const Icon(Icons.arrow_back, color: Colors.white), // Ícone de voltar
  onPressed: () {
    Navigator.pop(context); // Volta para a tela anterior
  },
),
title: const Text('Qual perfil você se encaixa?', style: TextStyle(color:
Colors.white)), // Título do AppBar
),
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center, // Alinha o conteúdo no
centro verticalmente
    children: [
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0), //
Espaçamento ao redor do botão
        child: SizedBox(
          width: 250, // Largura do botão
          child: ElevatedButton(
            style: ElevatedButton.styleFrom(
              backgroundColor: const Color(0xFFE0E0E0), // Cor de fundo do
botão
              foregroundColor: Colors.black, // Cor do texto do botão
              padding: const EdgeInsets.symmetric(vertical: 16.0), //
Espaçamento interno do botão
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10), // Borda arredondada
do botão
              ),
            ),
            onPressed: () {
              // Passa o tipo de usuário para a próxima tela
              Navigator.push(

```

```

        context,
        MaterialPageRoute(
            builder: (context) => RegisterStep2Screen(userType: "Pessoa
Física"), // Define o tipo como "Pessoa Física"
        ),
    );
},
    child: const Text('Sou uma Pessoa Física'), // Texto do botão
),
),
),
    Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0), //
Espaçamento ao redor do botão
        child: SizedBox(
            width: 250, // Largura do botão
            child: ElevatedButton(
                style: ElevatedButton.styleFrom(
                    backgroundColor: const Color(0xFFE0E0E0), // Cor de fundo do
botão
                    foregroundColor: Colors.black, // Cor do texto do botão
                    padding: const EdgeInsets.symmetric(vertical: 16.0), //
Espaçamento interno do botão
                    shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(10), // Borda arredondada
do botão
                    ),
                ),
            ),
            onPressed: () {
                // Passa o tipo de usuário para a próxima tela
                Navigator.push(
                    context,
                    MaterialPageRoute(

```

```

        builder: (context) => RegisterStep2Screen(userType: "ONG"), //
Define o tipo como "ONG"
        ),
    );
  },
  child: const Text('Sou uma ONG'), // Texto do botão
),
),
),
],
),
),
);
}
}

```

```
import 'package:flutter/material.dart';
```

```

class PetDetailScreen extends StatelessWidget {
  // Parâmetros para os detalhes do pet
  final String name;
  final String breed;
  final String age;
  final String gender; // Adiciona o parâmetro de gênero
  final String imagePath;

  const PetDetailScreen({
    Key? key,
    required this.name,
    required this.breed,
    required this.age,
    required this.gender, // Linha adicionada para o gênero
    required this.imagePath,
  }) : super(key: key);

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.transparent, // Fundo transparente para o
AppBar
      elevation: 0, // Remove a sombra do AppBar
      leading: IconButton(
        icon: const Icon(Icons.arrow_back, color: Colors.black), // Ícone de voltar
        onPressed: () {
          Navigator.pop(context); // Volta para a tela anterior
        },
      ),
      actions: [
        IconButton(
          icon: const Icon(Icons.share, color: Colors.black), // Ícone de
compartilhar
          onPressed: () {},
        ),
        IconButton(
          icon: const Icon(Icons.favorite_border, color: Colors.black), // Ícone de
favorito
          onPressed: () {},
        ),
      ],
    body: Padding(
      padding: const EdgeInsets.symmetric(horizontal: 16.0), // Espaçamento
horizontal
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start, // Alinha o conteúdo à
esquerda

```

```

children: [
  Center(
    child: AspectRatio(
      aspectRatio: 4 / 3, // Ajusta a proporção da imagem
      child: Container(
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20), // Bordas arredondadas
          color: Colors.grey[300], // Cor de fundo da imagem
        ),
        child: ClipRRect(
          borderRadius: BorderRadius.circular(20), // Bordas arredondadas
          child: imagePath.startsWith('http') // Verifica se o caminho da
da imagem
imagem é uma URL
            ? Image.network(
              imagePath,
              fit: BoxFit.cover, // Ajusta a imagem para preencher o espaço
              width: double.infinity,
            )
            : Image.asset(
              imagePath,
              fit: BoxFit.cover, // Ajusta a imagem para preencher o espaço
              width: double.infinity,
            ),
          ),
        ),
      ),
    ),
  const SizedBox(height: 16), // Espaço entre a imagem e o texto
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween, // Espaça o
nome e o ícone de gênero
    children: [
      Text(

```

```

    name, // Nome do pet
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 24,
    ),
  ),
  Icon(
    // Ícone de gênero, mostra masculino ou feminino
    gender.toLowerCase() == 'macho' ? Icons.male : Icons.female,
    color: Colors.black,
    size: 24,
  ),
],
),
const SizedBox(height: 4), // Espaço entre o nome e o rótulo
Container(
  padding: const EdgeInsets.symmetric(horizontal: 8, vertical: 4), //
Espaçamento interno do rótulo
  decoration: BoxDecoration(
    color: Colors.grey[200], // Cor de fundo do rótulo
    borderRadius: BorderRadius.circular(12), // Bordas arredondadas
do rótulo
  ),
  child: const Text(
    'Deficiência Visual', // Texto do rótulo
    style: TextStyle(
      fontSize: 14,
      color: Colors.black,
    ),
  ),
),
const SizedBox(height: 16), // Espaço entre o rótulo e os detalhes
Row(

```

```

        mainAxisAlignment: MainAxisAlignment.spaceBetween, // Espaça os
detalhes do pet
        children: [
            _buildDetailChip('Idade', age), // Chip de detalhe com a idade
            _buildDetailChip('Raça', breed), // Chip de detalhe com a raça
            _buildDetailChip('Cor', 'Preto'), // Chip de detalhe com a cor
        ],
    ),
    const SizedBox(height: 16), // Espaço entre os detalhes e o texto
    "Sobre mim"
    const Text(
        'Sobre mim', // Seção "Sobre mim"
        style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 18,
        ),
    ),
    const SizedBox(height: 8), // Espaço antes da descrição
    const Text(
        // Descrição do pet
        'Olá pessoal, tudo bem? Sou novato por essas bandas e fui resgatado
de um condomínio onde eu não tinha proteção nenhuma. Os voluntários lá me
disseram que as pessoas vêm aqui pra procurar o amorzinho da vida delas. Olha, quer
dizer que me sinto pronto pra encher a sua casa de alegria, ronrons e chamegos. Só
falta você vir me buscar!',
        style: TextStyle(fontSize: 16),
    ),
    const SizedBox(height: 24), // Espaço antes das informações médicas
    const Text(
        'Informações Médicas', // Seção "Informações Médicas"
        style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 18,
        ),
    ),

```

```

    ),
    const SizedBox(height: 8), // Espaço antes dos chips médicos
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly, // Espaça os
chips médicos uniformemente
      children: [
        _buildMedicalInfoChip('Castrado'), // Chip médico "Castrado"
        _buildMedicalInfoChip('Vacinado'), // Chip médico "Vacinado"
        _buildMedicalInfoChip('Vermifugado'), // Chip médico
"Vermifugado"
      ],
    ),
  ],
),
),
),
);
}

```

// Método para construir um chip de detalhe

```

Widget _buildDetailChip(String label, String value) {
  return Container(
    padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 8), //
Espaçamento interno do chip
    decoration: BoxDecoration(
      color: Colors.grey[200], // Cor de fundo do chip
      borderRadius: BorderRadius.circular(20), // Bordas arredondadas do chip
    ),
    child: Column(
      children: [
        Text(
          label, // Rótulo do chip
          style: const TextStyle(
            fontSize: 14,

```



```

        color: Colors.black54,
      ),
    ),
    const SizedBox(height: 4), // Espaço entre o rótulo e o valor
    Text(
      value, // Valor do chip
      style: const TextStyle(
        fontSize: 16,
        fontWeight: FontWeight.bold,
      ),
    ),
  ],
),
);
}

```

```

// Método para construir um chip de informação médica
Widget _buildMedicalInfoChip(String label) {
  return Container(
    padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 16), //
Espaçamento interno do chip médico
    decoration: BoxDecoration(
      color: Colors.grey[200], // Cor de fundo do chip médico
      borderRadius: BorderRadius.circular(20), // Bordas arredondadas do chip
médico
      border: Border.all(color: Colors.grey.shade400), // Borda do chip médico
    ),
    child: Column(
      children: [
        Icon(
          Icons.check_circle, // Ícone de check para indicar a informação médica
          color: Colors.black,
          size: 24,
        ),

```

```

const SizedBox(height: 8), // Espaço entre o ícone e o texto
Text(
  label, // Texto do chip médico
  style: const TextStyle(
    fontSize: 16,
    fontWeight: FontWeight.bold,
  ),
),
],
),
);
}
}

```

```

import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

```

```

// Tela que exibe as informações de uma ONG específica

```

```

class NGOsScreen extends StatelessWidget {

```

```

  final String name;

```

```

  // Construtor da tela que recebe o nome da ONG

```

```

  const NGOsScreen({Key? key, required this.name}) : super(key: key);

```

```

  @override

```

```

  Widget build(BuildContext context) {

```

```

    return Scaffold(

```

```

      appBar: AppBar(

```

```

        backgroundColor: Colors.transparent, // Fundo transparente para o

```

```
AppBar

```

```

        elevation: 0,

```

```

        leading: IconButton(

```

```

          icon: const Icon(Icons.arrow_back, color: Colors.black),

```

```

onPressed: () {
  Navigator.pop(context); // Retorna para a tela anterior
},
),
actions: [
  IconButton(
    icon: const Icon(Icons.share, color: Colors.black),
    onPressed: () {
      // Lógica para compartilhar as informações da ONG
    },
  ),
],
),
body: SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        // Seção que exibe a logo da ONG e o nome
        Center(
          child: Column(
            children: [
              Container(
                height: 100,
                width: 100,
                decoration: BoxDecoration(
                  color: Colors.grey.shade300,
                  shape: BoxShape.circle,
                  image: const DecorationImage(
                    image: AssetImage('assets/images/logo.png'),
                    fit: BoxFit.cover,
                  ),
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  ),
)

```

```

    ),
    const SizedBox(height: 16),
    Text(
      name,
      style: const TextStyle(
        fontSize: 28,
        fontWeight: FontWeight.bold,
        color: Colors.black,
      ),
    ),
    const SizedBox(height: 8),
    Container(
      padding: const EdgeInsets.symmetric(horizontal: 12, vertical: 4),
      decoration: BoxDecoration(
        color: Colors.teal,
        borderRadius: BorderRadius.circular(12),
      ),
      child: const Text(
        'Cachorros e Gatos', // Categoria da ONG
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ],
),
const SizedBox(height: 24),
// Seção "Sobre mim"
const Text(
  'Sobre mim',
  style: TextStyle(
    fontSize: 20,

```

```

        fontWeight: FontWeight.bold,
        color: Colors.black,
    ),
),
const SizedBox(height: 8),
const Text(
    'A Bem me Quer nasceu em 2016, é uma entidade não
    Governamental, sem fins lucrativos, que visa tirar o maior número possível de animais
    das ruas, tratar de possíveis traumas, doenças e integrá-lo a famílias que estejam
    dispostas a dar muito amor e carinho.',
    style: TextStyle(
        fontSize: 16,
        color: Colors.black54,
    ),
),
const SizedBox(height: 24),
// Seção "Meus Contatos"
const Text(
    'Meus Contatos',
    style: TextStyle(
        fontSize: 20,
        fontWeight: FontWeight.bold,
        color: Colors.black,
    ),
),
const SizedBox(height: 16),
Row(
    children: const [
        Icon(Icons.location_on, color: Colors.black54),
        SizedBox(width: 8),
        Expanded(
            child: Text(
                'Centro, Barueri - SP',
                style: TextStyle(

```

```

        fontSize: 16,
        color: Colors.black54,
      ),
    ),
  ),
],
),
const SizedBox(height: 12),
Row(
  children: const [
    Icon(Icons.email, color: Colors.black54),
    SizedBox(width: 8),
    Expanded(
      child: Text(
        'bemmequer@ajudeumavida.com.br',
        style: TextStyle(
          fontSize: 16,
          color: Colors.black54,
        ),
      ),
    ),
  ],
),
const SizedBox(height: 12),
Row(
  children: const [
    Icon(Icons.language, color: Colors.black54),
    SizedBox(width: 8),
    Expanded(
      child: Text(
        'www.ajudeumavida.com.br',
        style: TextStyle(
          fontSize: 16,
          color: Colors.black54,

```

```
    ),
  ),
),
],
),
const SizedBox(height: 24),
// Seção "Responsável"
const Text(
  'Responsável',
  style: TextStyle(
    fontSize: 20,
    fontWeight: FontWeight.bold,
    color: Colors.black,
  ),
),
const SizedBox(height: 16),
Row(
  children: [
    CircleAvatar(
      radius: 24,
      backgroundImage: AssetImage('assets/images/person.png'),
    ),
    const SizedBox(width: 16),
    Expanded(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: const [
          Text(
            'Kaique Muñoz', // Nome do responsável
            style: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: Colors.black,
            ),
          ),
        ],
      ),
    ),
  ],
),
```

```

    ),
    SizedBox(height: 4),
    Text(
      '9 km de distância', // Distância do responsável
      style: TextStyle(
        fontSize: 14,
        color: Colors.black54,
      ),
    ),
  ],
),
),
// Botões para ligar ou enviar mensagem via WhatsApp
IconButton(
  icon: const Icon(Icons.phone, color: Colors.black54),
  onPressed: () async {
    final Uri telUri = Uri(scheme: 'tel', path: '5511999999999'); //
Substituir pelo número real
    if (await canLaunchUrl(telUri)) {
      await launchUrl(telUri, mode: LaunchMode.externalApplication);
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Não foi possível abrir os
contatos')),
      );
    }
  },
),
IconButton(
  icon: const Icon(Icons.whatsapp, color:
Colors.black54),
  onPressed: () async {
    final Uri whatsappUri =
Uri.parse('https://wa.me/5511999999999'); // Substituir pelo número real

```



```

        if (await canLaunchUrl(whatsappUri)) {
          await launchUrl(whatsappUri);
        } else {
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Não foi possível abrir o
WhatsApp')),
          );
        }
      },
    ),
  ],
),
],
),
),
),
);
}
}

```

```

import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart'; // Importa o FirebaseOptions para a configuração
específica da plataforma
import 'login-input.dart'; // Importa a tela de login principal
import 'splash.dart'; // Importa a tela de splash

void main() async {
  WidgetsFlutterBinding.ensureInitialized(); // Garante que os widgets sejam
inicializados corretamente

  // Inicializa o Firebase com as opções específicas da plataforma
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,

```

```

);

runApp(const MyApp()); // Executa o aplicativo principal
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false, // Remove o banner de depuração
      initialRoute: '/splash-screen', // Define a rota inicial como a tela de splash
      routes: {
        '/splash-screen': (context) => const SplashScreen(), // Rota para a tela de
splash
        '/login-input': (context) => const LoginInputScreen(), // Rota para a tela
principal de login
      },
    );
  }
}

```

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.ajudaumavida.app"
  android:versionCode="1" <!-- Código da versão do aplicativo -->
  android:versionName="1.0"> <!-- Nome da versão do aplicativo -->

  <!-- Permissão para acessar a internet -->
  <uses-permission android:name="android.permission.INTERNET"/>

  <!-- Permissões para acessar a localização -->

```

```

    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/> <!-- Permissão
para localização precisa -->
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/> <!--
Permissão para localização aproximada -->

    <!-- Permissões para armazenamento e câmera -->
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/> <!-- Permissão
para ler armazenamento externo -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/> <!--
Permissão para escrever no armazenamento externo -->
    <uses-permission    android:name="android.permission.CAMERA"/> <!--
Permissão para usar a câmera -->

    <!-- Declara que o uso da câmera é opcional -->
    <uses-feature                android:name="android.hardware.camera"
android:required="false"/>

    <application
    android:label="ajudeumavida" <!-- Nome do aplicativo -->
    android:icon="@mipmap/ic_launcher" <!-- Ícone do aplicativo -->
    android:theme="@style/LaunchTheme"> <!-- Tema inicial do aplicativo -->

    <!-- Chave da API do Google Maps -->
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="SUA_CHAVE_API_AQUI"/> <!-- Substitua pela sua
chave API -->

    <activity
        android:name=".MainActivity" <!-- Atividade principal do aplicativo -->

```

```

        android:exported="true"
        android:launchMode="singleTop" <!-- Define o modo de lançamento
como singleTop -->

android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestS
creenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode" <!--
Configurações de mudanças de configuração -->
        android:hardwareAccelerated="true" <!-- Ativa a aceleração de
hardware -->
        android:windowSoftInputMode="adjustResize"> <!-- Ajusta o layout ao
exibir o teclado -->

<!-- Tema normal usado pelo Flutter -->
<meta-data
    android:name="io.flutter.embedding.android.NormalTheme"
    android:resource="@style/NormalTheme" />

<!-- Intent filter para definir a atividade principal que inicia o aplicativo --
>

<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>

<!-- Configuração para a nova API de incorporação do Flutter -->
<meta-data
    android:name="flutterEmbedding"
    android:value="2" />
</application>

<!-- Consultas para intents específicos -->
<queries>
    <intent>


```

```
        <action android:name="android.intent.action.PROCESS_TEXT"/> <!--  
Permite processar texto em outras apps -->  
        <data android:mimeType="text/plain"/> <!-- Tipo de dados de texto  
simples -->  
    </intent>  
</queries>  
</manifest>
```


4 RESULTADOS



Figura 4 - Tela de Apresentação





Login



[Esqueceu a senha?](#)

Login

ou continue com

Não tem uma conta?
[Cadastre-se](#)

Figura 5 - Tela de Login

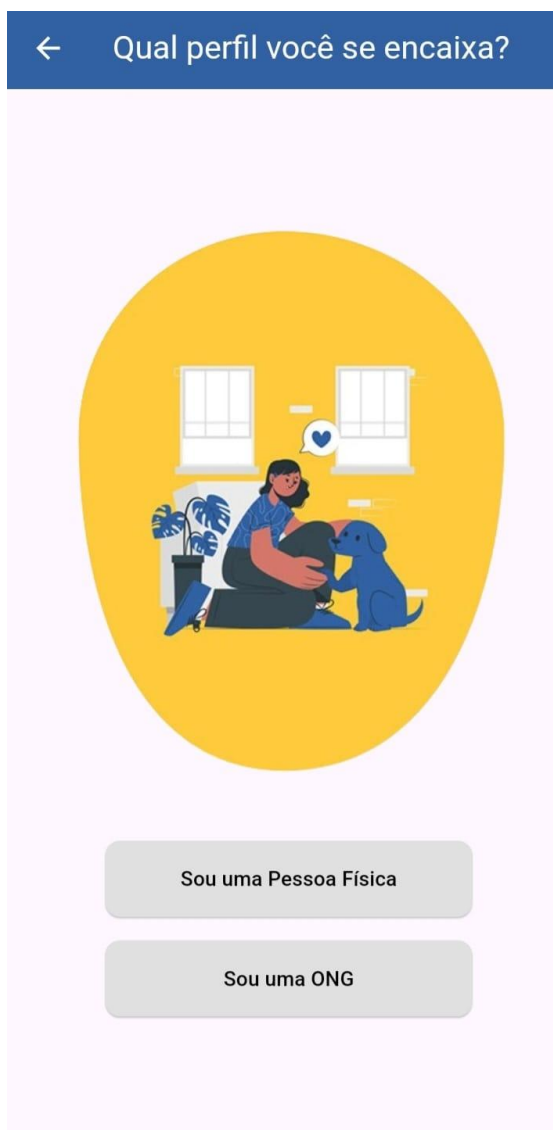
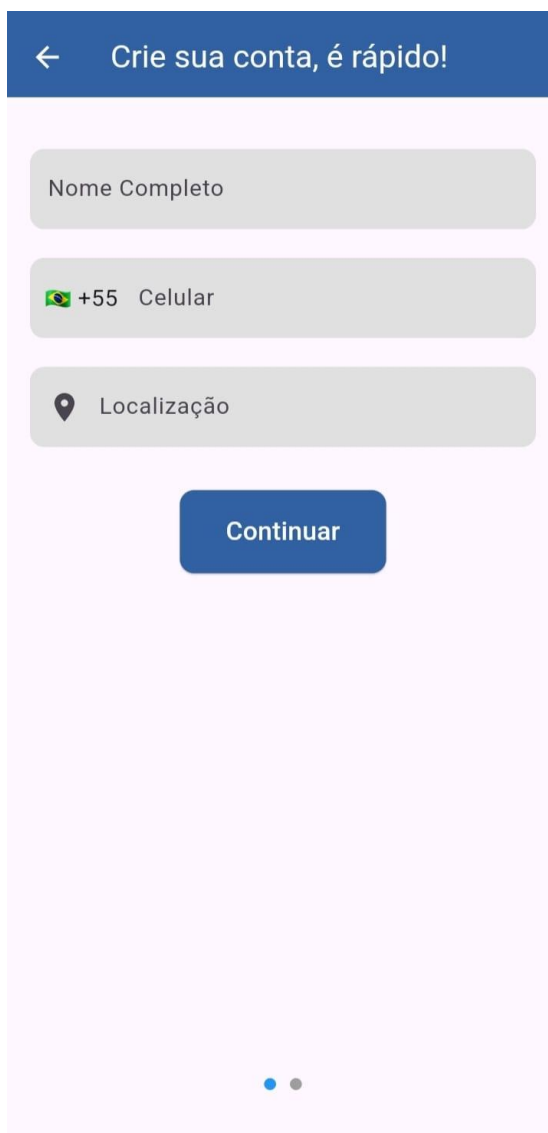


Figura 6 - 1ª Tela de Registro



The image shows a mobile application registration screen. At the top, there is a dark blue header with a white back arrow on the left and the text "Crie sua conta, é rápido!". Below the header, there are three light gray rounded rectangular input fields stacked vertically. The first field is labeled "Nome Completo". The second field is labeled with a Brazilian flag icon, "+55", and "Celular". The third field is labeled with a location pin icon and "Localização". Below these fields is a dark blue rounded rectangular button with the white text "Continuar". At the bottom center of the screen, there are two small dots, the first one is blue and the second is gray, indicating the current step in a multi-step process.

Figura 7 – 2ª Tela de Registro



Crie sua conta

Eu concordo com os Termos & Condições e Políticas de Privacidade

Confirmar

Já tem uma conta? [Login](#)

Figura 8 – 3ª Tela de Registro

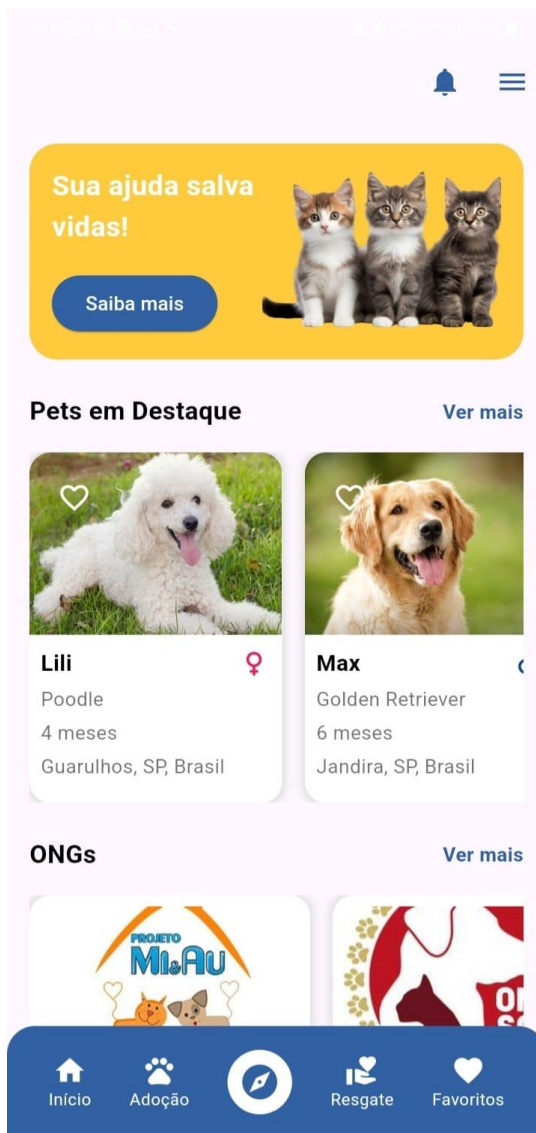


Figura 9 - Tela Inicial

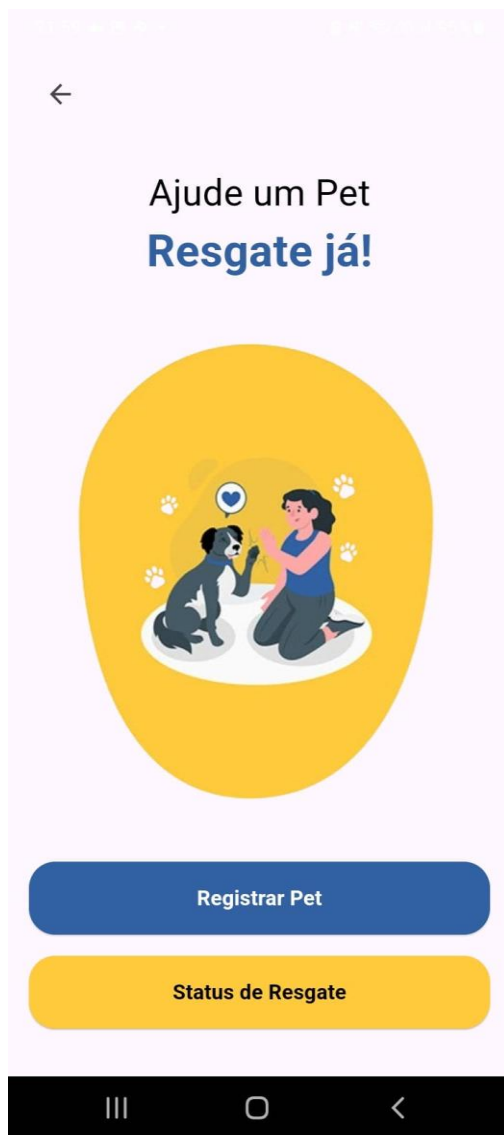


Figura 10 - Tela de Resgate

← Registrar Pet



Olá, Projeto Miau

Aqui você pode registrar animais para o resgate de uma ONG e serem adotados

Nome do Pet

Raça

Idade

Cor

Figura 11 - Tela de Registro de Animal

The image shows a mobile application screen titled "Registrar Pet". At the top, there is a blue header with a back arrow and the text "Registrar Pet". Below the header, the form consists of several input fields: a partially visible field at the top, followed by "Idade", "Cor", and "Gênero". The "Gênero" section has two radio button options: "Macho" and "Fêmea". Below these is a location field labeled "Localização onde o pet foi visto pel..." with a location pin icon. There are three checkboxes for "Castrado", "Vacinado", and "Vermifugado". Below the checkboxes, the text "Nenhuma imagem selecionada." is displayed, followed by a button labeled "Escolher Imagem". At the bottom of the form is a large blue button labeled "Registrar Pet".

Figura 12 - 2ª Parte da Tela de Registro de Animal

← Detalhes do Resgate



Max

Status	Data de Registro
Chamado criado	2024-11-26

Localização

Barueri, SP, Brasil

Criador: Nicolas

📞 11940296767

✉️ nicolasdias2015@outlook.com

- ✓ Chamado criado
- ✓ ONG entrou em contato
- ✓ ONG à caminho
- ✓ Pet resgatado
- ✓ Pronto para adoção

Figura 13 - Tela de Status do Resgate (Detalhada)

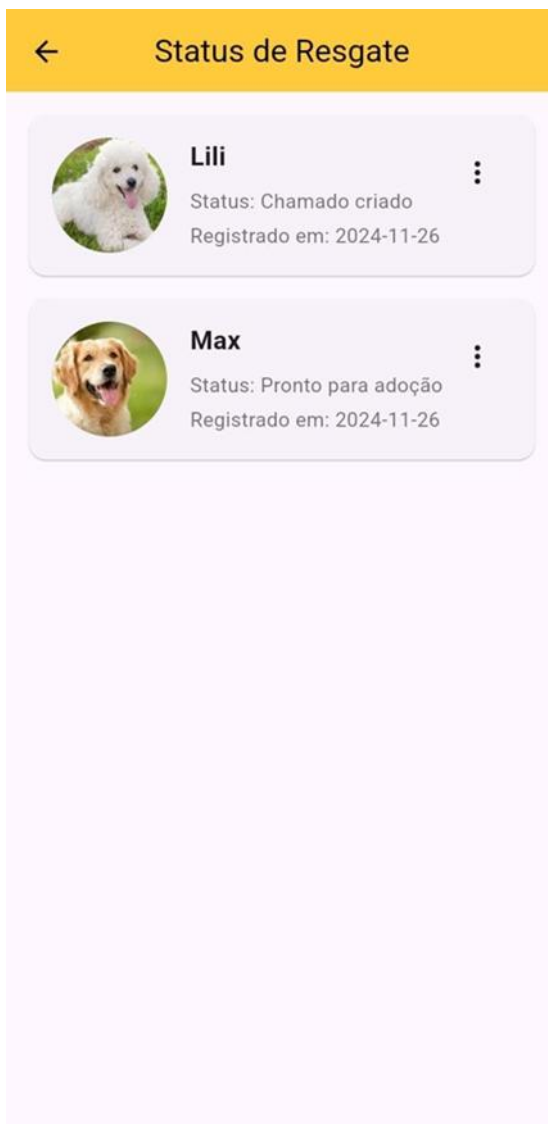


Figura 14 - Tela de Status do Resgate (Geral)

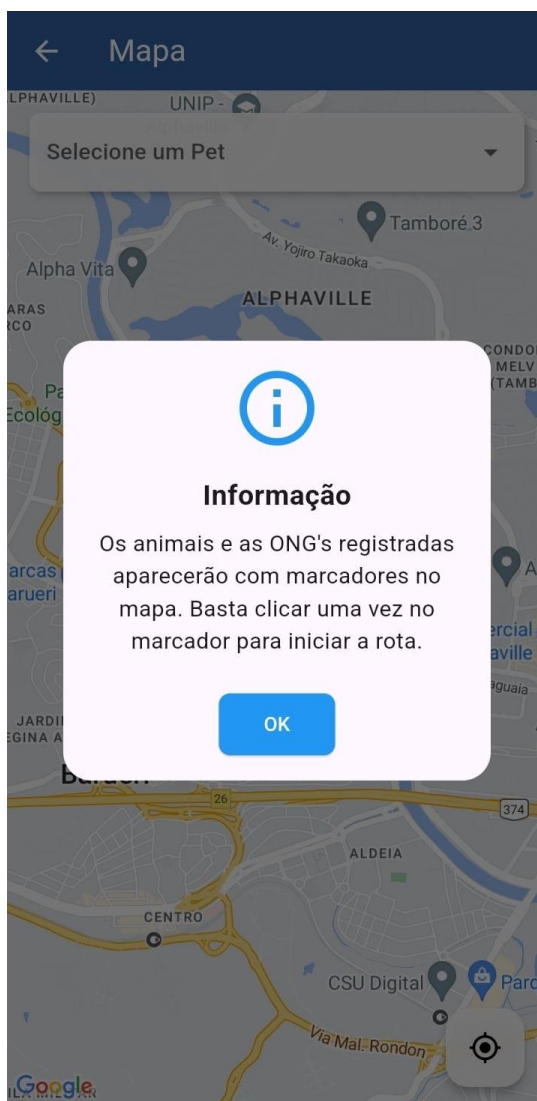


Figura 15 - Instrução do Mapa

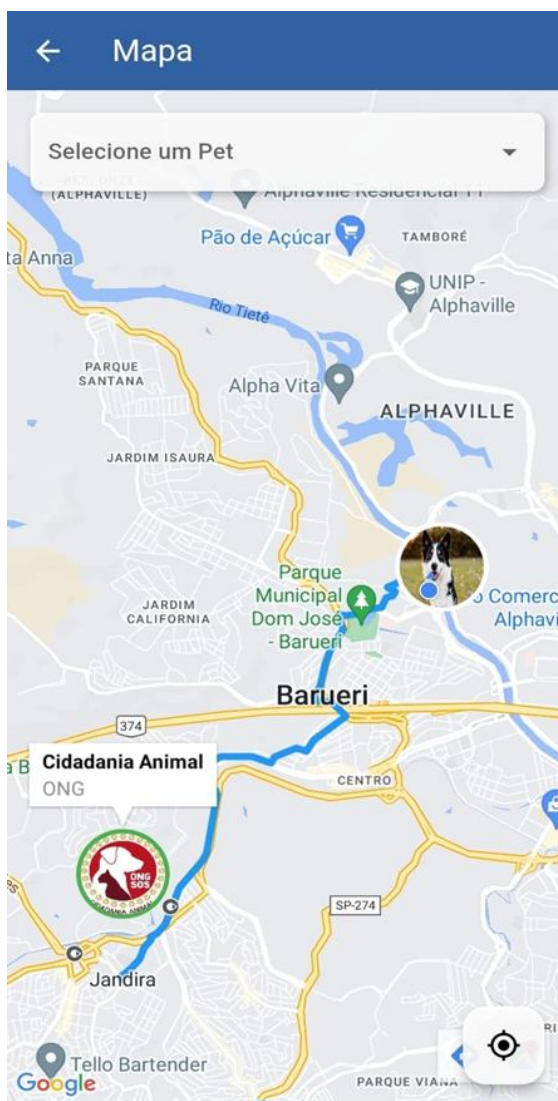


Figura 16 - Mapa para Localização de Animais e ONG's



Figura 17 - Tela Inicial com Notificações Recebidas



Figura 18 - Tela de Notificações

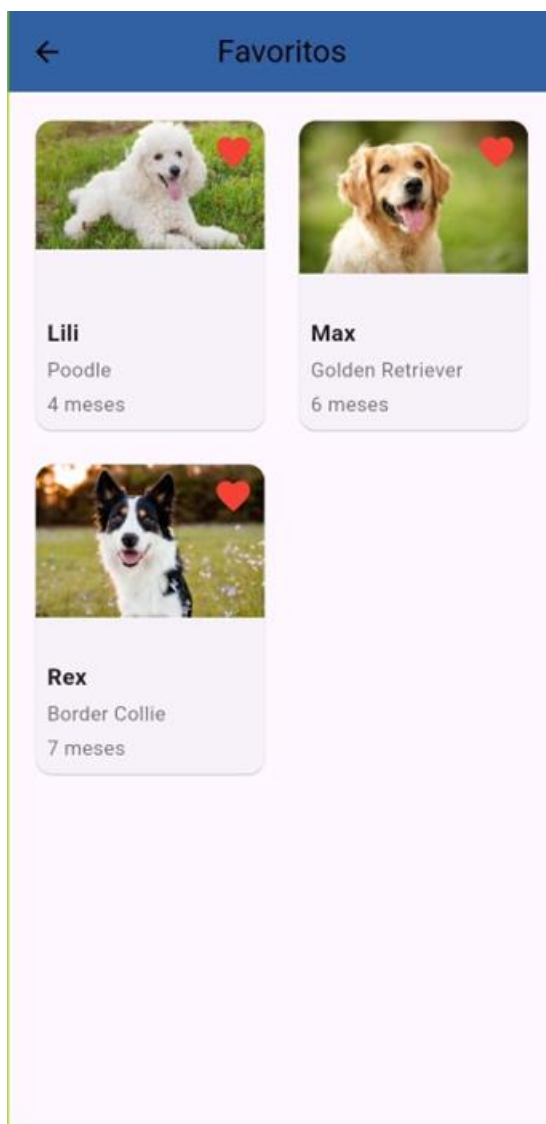


Figura 19 - Tela de Animais Favoritos

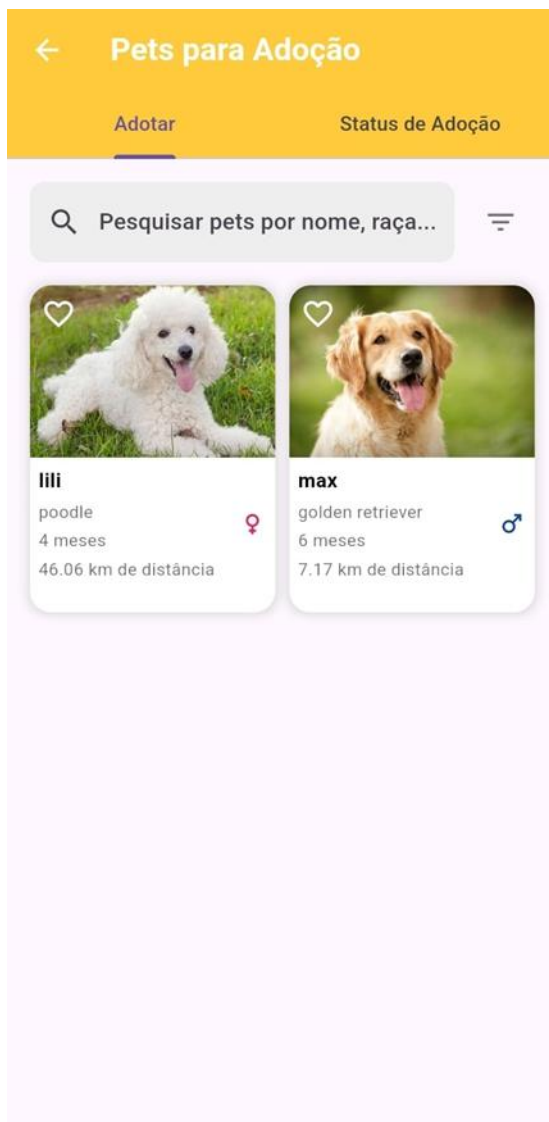


Figura 20 - Tela de Pets para Adoção

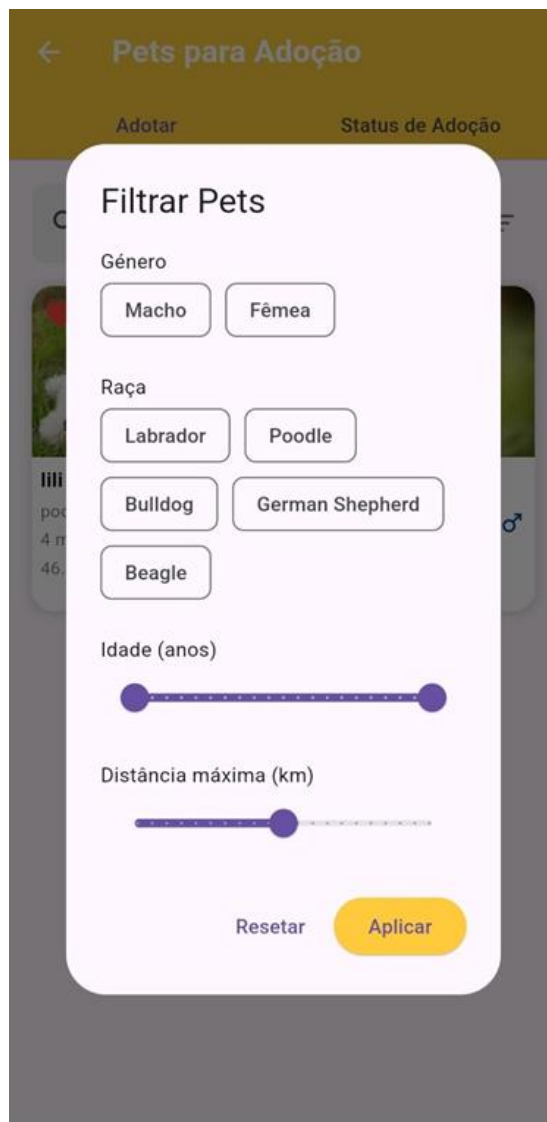


Figura 21 - Opções de Filtro de Pets



The screenshot displays a mobile application interface for dog adoption. At the top, there are navigation icons: a back arrow, a share icon, and a heart icon. Below these is a large photograph of a white, fluffy Poodle dog sitting on green grass with its tongue out. Under the photo, the name "Lili" is displayed in bold black text, followed by a female gender symbol. A blue location banner indicates "Projeto Miau" in "Guarulhos, SP, Brasil" at a distance of "46.06 km". Below this, three light gray buttons show the dog's attributes: "Idade" (4 meses), "Raça" (Poodle), and "Cor" (Branca). A section titled "Sobre mim" contains a text description of the dog's background. At the bottom of the text is an orange button labeled "Me adote". The entire screen is framed by a light pink border, and a black Android navigation bar is visible at the very bottom.

←

→

♡



Lili ♀

Projeto Miau
Guarulhos, SP, Brasil - 46.06 km

Idade
4 meses

Raça
Poodle

Cor
Branca

Sobre mim

Olá pessoal, tudo bem? Sou novato por essas bandas e fui resgatado de um condomínio onde eu não tinha proteção nenhuma. Os voluntários lá me disseram que as pessoas vêm aqui pra procurar o amorzinho da vida delas. Olha, quer dizer que me sinto muito feliz em encher a sua casa de alegria e amor. Só falta você vir me buscar.

Me adote

||| ○ <

Figura 22 - Tela de Adoção

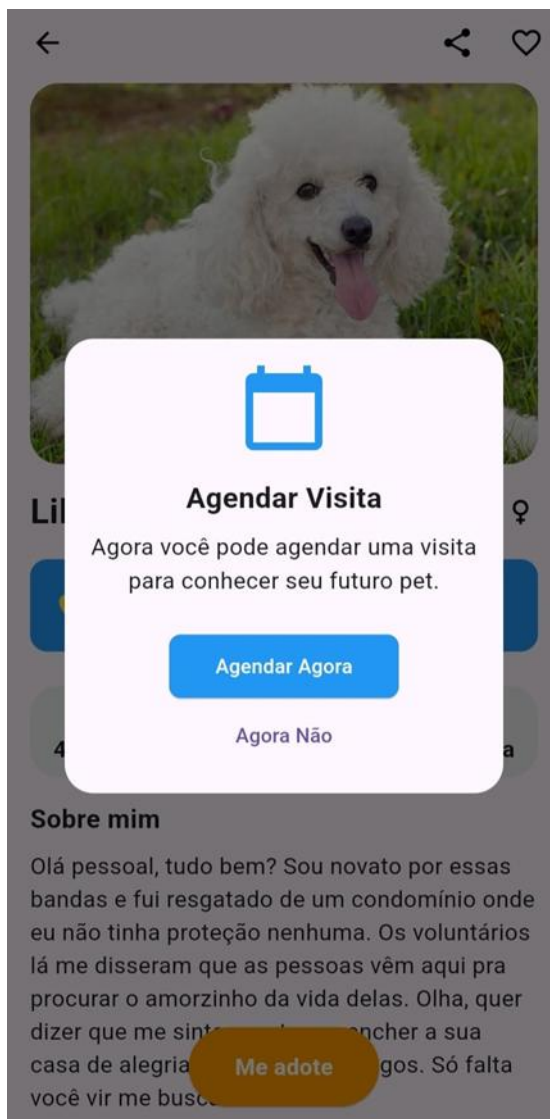


Figura 23 - Opção para Agendar Visita



← Agendar Visita

← Novembro 2024 →

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Horários Disponíveis

08:00	09:00	10:00	11:00
12:00	13:00	14:00	15:00
16:00	17:00	18:00	

Confirmar Horário

Figura 24 - Tela de Agendamento



Figura 25 - Notificação de Agendamento Realizado

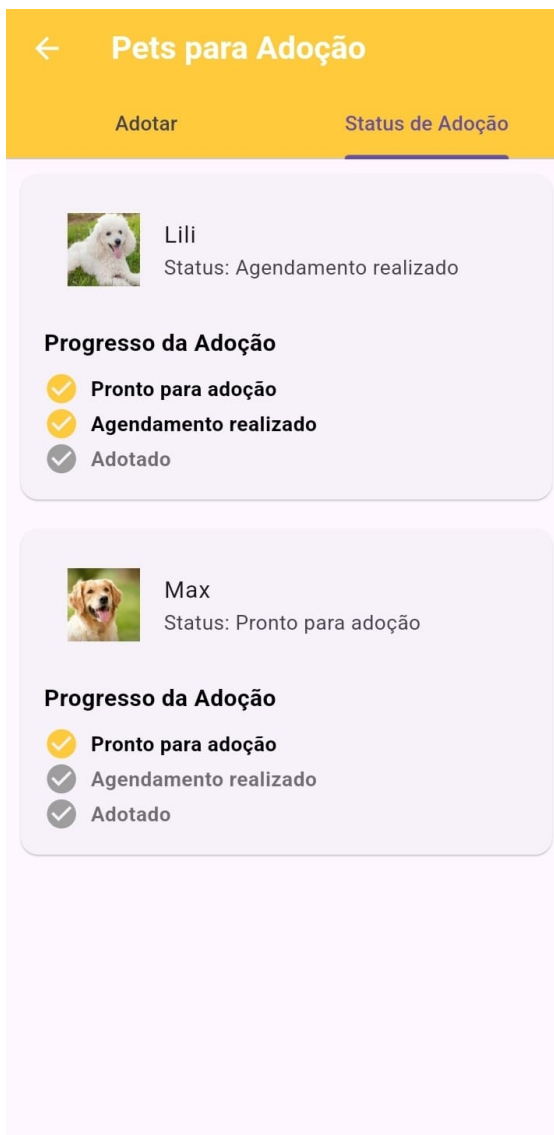


Figura 26 – Tela de Status de Adoção

5 CONCLUSÃO

O objetivo do aplicativo de auxílio a animais em situação de vulnerabilidade social é ligar a sociedade, incluindo ONGs e outras entidades de apoio, formando uma rede de solidariedade para salvar, guardar e fomentar o bem-estar dos animais sem lar. Por meio de uma plataforma colaborativa, o aplicativo tem como objetivo simplificar o acesso à localização e ao resgate desses animais, possibilitando que a população reconheça e comunique casos de animais abandonados diretamente às entidades que podem oferecer auxílio. Um dos maiores destaques do aplicativo é a funcionalidade de geolocalização, que permite um mapeamento exato dos animais em situação de risco. Assim, o público interessado pode reconhecer áreas com maior necessidade de auxílio, incentivando uma ação mais ágil e estruturada. A Plataformas colaborativas também proporciona outras opções. oferece às ONGs um recurso para promover campanhas de adoção, aumentar a visibilidade de eventos e angariar recursos de maneira integrada e acessível. Este conjunto de medidas é crucial para que as entidades obtenham os fundos necessários para ampliar suas atividades e impulsionar ações mais efetivas em prol dos animais. Ademais, o aplicativo possibilita que as entidades identifiquem as regiões mais vulneráveis e orientem esforços específicos para resgatar e auxiliar os animais de forma estratégica, potencializando o efeito de suas medidas. A meta final é diminuir a quantidade de animais abandonados nas ruas, incentivando a participação social e a responsabilidade compartilhada na defesa dos animais. Portanto, o aplicativo não apenas simplifica a participação do público nesta causa, mas também proporciona às pessoas a oportunidade de se engajarem. Uma forma eficaz de estruturar e intensificar suas atividades. Ao unir sociedade e, Organizações Não Governamentais (ONGs) numa ação conjunta, a plataforma surge como uma opção viável e acessível para aqueles que desejam contribuir para o bem-estar animal, promovendo uma corrente de benevolência e possibilitando uma mudança concreta neste problema social.

REFERÊNCIAS BIBLIOGRÁFICAS

ABES. Plataformas Digitais e seu impacto no desenvolvimento econômico e social do Brasil. 2023. Disponível em: <<https://abes.com.br/plataformas-digitais-e-seu-impacto-no-desenvolvimento-economico-e-social-do-brasil/>> . Acesso em: 10 set. 2024

ALLAN. Introdução à Google Maps API. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-a-google-maps-api/26967>> . Acesso em: 12 set. 2024

ALVES, A. J. S. e et al. Abandono de cães na América Latina: revisão de literatura. 2012. Disponível em: <<https://www.revistamvez-crmvsp.com.br/index.php/recmvz/article/view/16221/17087>> . Acesso em: 13 set. 2024

ALVES, F. F. ANÁLISE E DESENVOLVIMENTO DE UMA APLICAÇÃO ANDROID DE AUTOATENDIMENTO PARA RESTAURANTE. Disponível em: <<https://repositorio.ueg.br/jspui/bitstream/riueg/1853/2/MG%20024%200027-2015.pdf>> . Acesso em: 14 set. 2024

ALVES, I. Lista: conheça 7 projetos educacionais do Brasil. 2018. Disponível em: <<https://observatorio3setor.org.br/lista-conheca-7-projetos-educacionais-do-brasil/>> . Acesso em: 15 set. 2024

ANGELINA NETO, M. M.; MONTE, J. P. do; OLIVEIRA JÚNIOR, G. M. de. CONSEQUÊNCIA DO ABANDONO DE ANIMAIS PARA A SAÚDE PÚBLICA NA ZONA URBANA DA CIDADE DE SERRITA-PE. 2023. Disponível em: <https://drive.google.com/file/d/1rztTJuKrfyclWgsg9eo3r5VR0_18ittl/view> . Acesso em: 16 set. 2024

BARBOSA, R. de M. Uma análise sobre os discursos que tratam do terceiro seto no Brasil: A ação das ONGs. Disponível em: <<https://core.ac.uk/download/pdf/161369634.pdf>> . Acesso em: 17 set. 2024

BONIN, J. C.; MAKIOLKI, S. J.; HÜLSE, L. O problema do abandono de animais domésticos e a importância da educação cidadã em uma escola de Educação Básica de Timbó Grande, Santa Catarina. 2020. Disponível em: <<https://devireducacao.ded.ufla.br/index.php/DEVIR/article/view/318/177>> . Acesso em: 17 set. 2024

BRITO, D. C. de. Plataformas Colaborativas e Democracia Digital: As Novas Formas de Participação. 2017. Disponível em: <<https://portalintercom.org.br/anais/sul2017/resumos/R55-1117-1.pdf>> . Acesso em: 18 set. 2024

CABRAL, U. Quatro em cada dez municípios não têm serviço de esgoto no país. 2020. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/28326-quatro-em-cada-dez-municipios-nao-tem-servico-de-esgoto-no-pais>> . Acesso em: 20 set. 2024

CEZAR, J. O problema do abandono de animais domésticos e a importância da educação cidadã em uma escola de Educação Básica de Timbó Grande, Santa Catarina. Disponível em: <<https://devireducao.ded.ufla.br/index.php/DEVIR/article/view/318/177>> . Acesso em: 22 set. 2024

CHANGE.ORG. Sobre a Change.org. Disponível em: <<https://www.change.org/about>> . Acesso em: 22 set. 2024

COELHO, M. Q.; GONÇALVES, C. A. AVALIAÇÃO DE PROJETOS SOCIAIS: A PERSPECTIVA DA COMUNIDADE.2011. Disponível em: <<https://periodicos.univali.br/index.php/ra/article/view/3391>> . Acesso em: 23 set. 2024

DALILA, S. Legislação Indigenista e Perspectivas para o Turismo em Terras Indígenas no Brasil. Disponível em: <<https://www.revistas.usp.br/rta/article/view/110460/128207>> . Acesso em: 23 set. 2024

DIAS, P. Desenvolvimento de objectos de aprendizagem para plataformas colaborativas. Disponível em: <<https://repositorioaberto.uab.pt/bitstream/10400.2/16296/1/Desenvolvimento%20de%20objectos%20de%20aprendizagem%20para%20plataformas%20colaborativas.pdf>> Acesso em: 25 set. 2024

DUTRA, F. Vítimas das chuvas no RS, mais de 19 mil cães e gatos buscam adoção. Disponível em: <<https://www.metropoles.com/distrito-federal/vitimas-das-chuvas-no-rs-mais-de-19-mil-caes-e-gatos-buscam-adoacao>> . Acesso em 31/10/2024. Acesso em: 26 set. 2024

GABRIEL FILHO, O. Inteligência artificial e aprendizagem de máquina: aspectos teóricos e aplicações. 1. ed. [S.l.]: Blucher, 2023. 31 páginas p. ISBN 9786555066166. Acesso em: 27 set. 2024

GAIO JÚNIOR, A. Uso do App Inventor (AI) no desenvolvimento de aplicativos para dispositivos Android. Disponível em: <<https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/download/28889/2833>> . Acesso em: 27 set. 2024

GALVÃO, P.; ARAUJO, C. M. M. ONGs NO BRASIL: CONTEXTUALIZAÇÃO HISTÓRICA DO CENÁRIO PARA ATUAÇÃO EM PSICOLOGIA ESCOLAR. 2018. Disponível em: <<https://periodicos.pucminas.br/index.php/psicologiaemrevista/article/view/8629/14056>> . Acesso em: 27 set. 2024

GERHARDT, T. E.; SILVEIRA, D. T. Metodos de Pesquisa. 2009. Disponível em: <<https://lume.ufrgs.br/bitstream/handle/10183/213838/000728731.pdf>> . Acesso em: 28 set. 2024

GOFOUNDME. Sobre o GoFundMe. Disponível em: <<https://www.gofundme.com/pt-pt>> . Acesso em: 29 set. 2024

GOHN, M. da G. 500 ANOS DE LUTAS SOCIAIS NO BRASIL: movimentos sociais, ONGs e terceiro setor. 2000. Disponível em: <<https://ojs.uel.br/revistas/uel/index.php/mediacoes/article/view/9194/7788>> . Acesso em: 30 set. 2024

GOMES, W. T. Ferramentas colaborativas: o emprego de novas tecnologias na perspectiva educação. 2021. Disponível em: <<https://ayaeditora.com.br/wp-content/uploads/2022/01/L92C26.pdf>> . Acesso em: 30 set. 2024

GONÇALVES, N. S.; SOARES, P. S.; SANTOS, D. C. Panorama epidemiológico da raiva humana no Brasil com foco na região sul do país. 2017. Disponível em: <<https://www.redalyc.org/journal/5704/570463738011/html/>> . Acesso em: 01 out. 2024

GOODWIN, M. O que é uma API? 2024. Disponível em:
<<https://www.ibm.com/br-pt/topics/>> api . Acesso em: 01 out. 2024

GOUVEIA, F. ONGs enfrentam desafios e ocupam espaço da ação pública. 2007. Disponível em: <http://cienciaecultura.bvs.br/scielo.php?pid=S0009-67252007000200003&script=sci_arttext> . Acesso em: 02 out. 2024

INEP. MEC e Inep divulgam resultados do Censo Escolar 2023. 2024. Disponível em: <<https://www.gov.br/inep/pt-br/assuntos/noticias/censo-escolar/mec-e-inep-divulgam-resultados-do-censo-escolar-2023>> . Acesso em: 02 out. 2024

INSTITUTO C. ONG – O que é e como funciona. Acesso em:
<<https://institutoc.org.br/ong/#>> . Acesso em: 03 out. 2024.

JUNIOR. Wikis: Entenda O Que São E Como Funcionam. 2024. Disponível em: <<https://digitalizapro.com.br/glossario/wikis-entenda-o-que-sao-e-como-funcionam/>> . Acesso em: 03 out. 2024.

KAROLINA, F. ABANDONO DE ANIMAIS NO BRASIL: CONSEQUÊNCIAS GERADAS À SOCIEDADE. Disponível em:
<<https://periodicos.ufam.edu.br/index.php/resbam/article/view/6615/6303>> . Acesso em: 04 out. 2024.

KIVA. Kiva expande acesso financeiro para ajudar comunidades carentes a prosperar. Disponível em: <<https://www.kiva.org/about>> . Acesso em: 04 out. 2024.

LIMA, D. V. D. A PROBLEMÁTICA DO ABANDONO DE ANIMAIS DOMÉSTICOS SOB O PONTO DE VISTA DAS ONGs E PROJETOS VOLUNTÁRIOS NO MUNICÍPIO DE SÃO LUÍS - MA. 2022. Disponível em:
<<https://rosario.ufma.br/jspui/bitstream/123456789/5467/1/DAVIDVIEIRADELIMA.pdf>>. Acesso em: 05 out. 2024.

MAIA, R. R. Os impactos das novas plataformas sociais na Sociedade Digital. 2018. Disponível em: <<https://orcid.org/0000-0003-1101-6086>> . Acesso em: 05 out. 2024.

MAPBOX. APIs de serviços da Web. Disponível em:
<<https://docs.mapbox.com/api/overview/>> . Acesso em: 06 out. 2024.

MUNDO API. 8 Alternativas poderosas para a API do Google Maps. 2018. Disponível em: <<https://mundoapi.com.br/apis-publicadas/8-alternativas-poderosas-para-a-api-do-google-maps/>> . Acesso em: 06 out. 2024.

MYLENA, M. CONSEQUÊNCIA DO ABANDONO DE ANIMAIS PARA A SAÚDE PÚBLICA NA ZONA URBANA DA CIDADE DE SERRITA-PE. Disponível em: <https://drive.google.com/file/d/1rztTJuKrfyCLwgsq9eo3r5VR0_18ittl/view> . Acesso em: 08 out. 2024.

OLIVEIRA, F. A. F. de. REVOLUÇÃO TECNOLÓGICA E SMARTPHONE: CONSIDERAÇÕES SOBRE A CONSTITUIÇÃO DO SUJEITO CONTEMPORÂNEO. 2023. Disponível em: <<https://www.scielo.br/j/pe/a/mp6sqT7Ff7kyCzcrwvQR55m/?lang=pt#>> . Acesso em: 08 out. 2024.

OMS. O OpenStreetMap fornece dados de mapa para milhares de sites, aplicativos móveis e dispositivos de hardware. Disponível em: <<https://www.openstreetmap.org/about>> . Acesso em: 09 out. 2024.

ORACLE. O que é um banco de dados? 2020. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>> . Acesso em: 09 out. 2024.

PAULA, A. P. P. D. REINVENTANDO A DEMOCRACIA: ONGs E MOVIMENTOS SOCIAIS NA CONSTRUÇÃO DE UMA NOVA GESTÃO PÚBLICA. 1998. Disponível em: <<https://repositorio.fgv.br/server/api/core/bitstreams/9acfd25b-8183-4c99-80b8-b9ab298eb4fe/content>> . Acesso em: 10 out. 2024.

PEDROSO, J. de S.; SILVA, K. S. da; SANTOS, L. P. dos. PESQUISA DESCRITIVA E PESQUISA PRESCRITIVA. 2016. Disponível em: <<https://unisantacruz.edu.br/revistas-old/index.php/JICEX/article/view/2604>> . Acesso em: 10 out. 2024.

PIOVESAN, A. PESQUISA EXPLORATORIA: PROCEDIMENTO METODOLOGICO PARA O ESTUDO DE FATORES HUMANOS NO CAMPO DA SAUDE PUBLICA. Disponível em: <<https://www.scielo.br/j/rsp/a/fF44L9rmXt8PVYLNvphJgTd/?format=html#>> . Acesso em: 12 out. 2024.

QUEIROZ, F. K. do N. et al. ABANDONO DE ANIMAIS NO BRASIL: CONSEQUÊNCIAS GERADAS À SOCIEDADE. 2019. Disponível em: <<https://periodicos.ufam.edu.br/index.php/resbam/article/view/6615/6303>> . Acesso em: 12 out. 2024.

REV. SAUDE PUBLICA. PESQUISA EXPLORATORIA: PROCEDIMENTO METODOLOGICO PARA O ESTUDO DE FATORES HUMANOS NO CAMPO DA SAUDE PUBLICA. 1995. Disponível em: <<https://www.scielo.br/j/rsp/a/fF44L9rmXt8PVYLNvphJgTd/?format=html>> . Acesso em: 14 out. 2024.

ROCHA, J. D.; BURSZTYN, M. A. A IMPORTANCIA DA PARTICIPAÇÃO SOCIAL NA SUSTENTABILIDADE DO DESENVOLVIMENTO LOCAL. 2005. Disponível em: <<https://interacoes.ucdb.br/interacoes/article/view/496/540>> . Acesso em: 14 out. 2024.

SALES, C. ELEMENTOS MARCANTES DA PRESTAÇÃO DE CONTAS VOLUNTARIAS DAS 100 MELHORES ONGS DO BRASIL. Disponível em: <<https://periodicos.ufv.br/apgs/article/view/9789/6460>> . Acesso em: 14 out. 2024.

SANTOS, N. PANORAMA EPIDEMIOLÓGICO DA RAIVA HUMANA NO BRASIL COM FOCO NA REGIÃO SUL DO PAÍS. Disponível em: <<https://www.redalyc.org/journal/5704/570463738011/html/>> . Acesso em: 15 out. 2024.

SANTOS, P. P. dos. A NECESSIDADE DE CONSOLIDAÇÃO DOS FUNDAMENTOS DOS DIREITOS DOS ANIMAIS DOMÉSTICOS NO BRASIL: BEM-ESTAR ANIMAL, COMBATE AOS MAUS-TRATOS E AO ABANDONO. Disponível em: <http://icts.unb.br/jspui/bitstream/10482/41935/1/2021_PauladePaivaSantos.pdf> . Acesso em: 15 out. 2024.

SILVA, A. de S. ÍNDICE DE ABANDONO NO BRASIL. 2024. Disponível em: <<https://institutomvc.org.br/site/index.php/2024/04/04/indice-de-abandono-no-brasil/>> . Acesso em: 20 out. 2024.

SOUSA NETO, M. V. de. Datacenter: componente central da infraestrutura de TI. 2009. ed. [S.l.]: Brasport. 374 páginas p. ISBN 9788574525082. Acesso em: 20 out. 2024.

SOUZA, J. de. PESQUISA DESCRITIVA E PESQUISA PRESCRITIVA. Disponível em: <<https://unisantacruz.edu.br/revistas-old/index.php/JICEX/article/view/2604>> . Acesso em: 20 out. 2024.

SYDLE. INTEROPERABILIDADE: O QUE É E COMO IMPLEMENTAR? 2023. Disponível em: <<https://www.sydle.com/br/blog/interoperabilidade-6400cb7457aff34f9ef24563>> . Acesso em: 22 out. 2024.

TOLFO, D. MÉTODOS DE PESQUISA. Disponível em: <<https://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>> . Acesso em: 22 out. 2024.

WALDMAN, M. O DESAFIO AMBIENTAL E AS ONGS. 2007. Disponível em:< https://mw.pro.br/mw/eco_desafio_ambiental_e_as_ongs.pdf >. Acesso em: 22 out. 2024.

FRANCISCO DUTRA. VITIMAS DAS CHUVAS NO RS, MAIS DE 19 MIL CÃES E GATOS BUSCAM ADOÇÃO. Disponível em: < <https://www.metropoles.com/distrito-federal/vitimas-das-chuvas-no-rs-mais-de-19-mil-caes-e-gatos-buscam-adocao> > Acesso em: 22 out. 2024.

SANDRA REGINA. ANIMAIS ABANDONADOS EM ALAGOA GRANDE - PB: UMA ESTIMATIVA SOCIOECONÔMICA PELO DO OLHAR DO PROTETOR INDEPENDENTE. Disponível em: < <https://repositorio.ufpb.br/jspui/bitstream/123456789/23598/1/SRC19072022MV370.pdf> > Acesso em: 24 out. 2024.

NILCINARA HUERB DE AZEVEDO. A APLICAÇÃO DO INSTITUTO DA GUARDA EM FACEAOS ANIMAIS DE ESTIMAÇÃO NO SISTEMA JURÍDICO BRASILEIRO. Disponível em:

< <https://cadernosdoceas.ucsal.br/index.php/rladna/article/view/786/683> > Acesso em: 24 out. 2024.

GABRIELE FRIZZO RIBEIRO. CCÔMPENDIO DE BOAS AÇÕES PARA O BEM - ESTAR ANIMAL. Disponível em: < <https://eventos.ifrs.edu.br/index.php/PEnsE/mostra2016/paper/view/1314> >

Acesso em: 24 out. 2024.

MARIA EMILIA YAMAMOTO & GILSON LUIZ VOLPATO.
COMPORTAMENTO ANIMAL. Disponível em: <
https://edisciplinas.usp.br/pluginfile.php/4115904/mod_resource/content/1/Livro%20COMPORTAMENTO_ANIMAL%20Yam%20%20Volpato.pdf > Acesso em: 24 out. 2024.

XAIENE SIMON. CONTROLE POPULACIONAL DE CÃES E GATOS:
DIFERENTES ABORDAGENS NO BRASIL. Disponível em: <
<https://lume.ufrgs.br/handle/10183/255791> > Acesso em: 26 out.2024

RENATO BORTOLOTTI, RENATA GROTTA D'AGOSTINO. AÇÕES PELO
CONTROLE REPRODUTIVO E POSSE RESPONSÁVEL DE
ANIMAIS DOMÉSTICOS INTERPRETADAS À LUZ DO CONCEITO DE
METACONTINGÊNCIA. Disponível em: <
<http://novoperiodicos.ufpa.br/periodicos/index.php/rebac/article/view/821/1159> >
Acesso em: 26 out.2024

IGOR FERREIRA. SEGURANÇA ALIMENTAR NOS DOMICÍLIOS
BRASILEIROS VOLTA A CRESCER EM 2023. Disponível em: <
<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/39838-seguranca-alimentar-nos-domicilios-brasileiros-volta-a-crescer-em-2023> > Acesso em: 30 out.2024

DANIELI OLIVEIRA BARCELOS. ABANDONO DE ANIMAIS: A FRÁGIL
FISCALIZAÇÃO ESTATAL E OS PROJETOS FUTUROS DO DISTRITO
FEDERAL. Disponível em: <
<https://revistas.icesp.br/index.php/Real/article/view/5183/2968> > Acesso em: 30 out.2024

MARCELO HENRIQUE DOS SANTOS. O MINISTÉRIO PÚBLICO E SUA
ATUAÇÃO CONCORRENTE PARA O FORTALECIMENTO DO TERCEIRO
SETOR: A EXPERIÊNCIA DA CURADORIA DE FUNDAÇÕES EM ANÁPOLIS.
Disponível em: <
<https://www.rincon061.org/bitstream/aee/464/1/Marcela%20Henrique.pdf> >
Acesso em: 02 nov.2024

JOICE PEIXOTO LOPES, LARISSA SAYURI TAKAHASHI MEMURA. IMPACTO DA PANDEMIA DO COVID-19 NAS ORGANIZAÇÕES DO TERCEIRO SETOR: Estudo de caso de uma Organização Não Governamental da cidade de Limeira. Disponível em: < <https://repositorio.unicamp.br/acervo/detalhe/1258235> > Acesso em: 02 nov.2024

NARA KATSURAYAMA CAZZOLATO. AS DIFICULDADES DE GESTÃO DAS ORGANIZAÇÕES NÃO-GOVERNAMENTAIS. Disponível em: < <https://core.ac.uk/download/pdf/229080571.pdf> > Acesso em: 04 nov.2024

SANDRA DALILA CORBARI. LEGISLAÇÃO INDIGENISTA E PERSPECTIVAS PARA O TURISMO EM TERRAS INDIGENAS NO BRASIL. Disponível em: < <https://www.revistas.usp.br/rta/article/view/110460/128207> > Acesso em: 04 nov.2024

SANTOS. Android e a influência do Sistema Operacional Linux. Disponível em: < https://web.archive.org/web/20180422233251id_/http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/viewFile/829/728 > Acesso em: 05 nov.2024

ANTÔNIO LUÍS SOMBRA. ANÁLISE DE SEGURANÇA NA PLATAFORMA ANDROID. Disponível em: < http://www.defesacibernetica.ime.eb.br/pub/repositorio/2014-Sombra_Helder.pdf > Acesso em: 05 nov.2024

ROSATTO. iPhone, Android, e a consolidação da cultura do smartphone: o papel do iPhone e do Sistema Operacional Android como catalisadores da consolidação no mercado de smartphones em escala global. Disponível em: < <https://www.redalyc.org/journal/4966/496659054004/496659054004.pdf> > Acesso em: 05 nov.2024

RODRIGUES. ANÁLISE E DESENVOLVIMENTO DE UMA APLICAÇÃO ANDROID DE AUTOATENDIMENTO PARA RESTAURANTE. Disponível em: < <https://repositorio.ueg.br/jspui/bitstream/riueg/1853/2/MG%20024%200027-2015.pdf> > Acessado em 07 nov. 2024

Bruna B. Barro. Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem. Disponível em: < <https://www.hostinger.com.br/tutoriais/linguagem-dart> > Acesso em 06 nov. 2024

LUIZ MIRANDA. Crud: o que é e como funciona na programação. Disponível em: < <https://querobolsa.com.br/revista/crud-o-que-e> > Acessado em 26 out. 2024

NAPOLEON. O que é: Client-Server Model. Disponível em: <
<https://napoleon.com.br/glossario/o-que-e-client-server-model-2/> > Acesso em 07 out.
2024

LOPES, José. Introdução ao Flutter: Um Framework para Desenvolvimento de
Aplicativos Multiplataforma Disponível em:<
[https://www.dio.me/articles/introducao-ao-flutter-um-framework-para-desenvolvimento-de-
aplicativos-multiplataforma](https://www.dio.me/articles/introducao-ao-flutter-um-framework-para-desenvolvimento-de-aplicativos-multiplataforma) > Acesso em: 05 nov. 2024.