**Uma imagem contendo Logotipo

Descrição gerada automaticamenteCENTRO UNIVERSITÁRIO DO DISTRITO FEDERAL – UDF**

**ANALISE E DESENVOLVIMENTO DE SISTEMAS**

**DESIGN GRÁFICO**

**Letícia da Silva Bezerra**

**Luis Eduardo Braga Santos**

**Michelle Chahini Escudero**

**Pedro Henrique Costa**

**Documentação do Sistema para Controle de Adoção de Animais" (documentação de sistemas)**

**BRASÍLIA**

**2023**

**Letícia da Silva Bezerra**

**Luis Eduardo Braga Santos**

**Michelle Chahini Escudero**

**Pedro Henrique Costa**

Documentação do Sistema para Controle de Adoção de Animais" (documentação de sistemas)

Trabalho da disciplina Projeto Interdisciplinar apresentado à Coordenação do curso de Análise e Desenvolvimento de Sistemas, do Centro Universitário do Distrito Federal - UDF, como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimeto de Sistemas.

Orientadora: Profª. Dra. Leticia Zoby

SUMÁRIO

[1. INTRODUÇÃO 13](#_Toc129265530)

[2. OBJETIVOS 1](#_Toc129265531)4

[a. Geral 1](#_Toc129265532)4

[b. Especifico 1](#_Toc129265533)4

[3. Fundamentação teórica 1](#_Toc129265534)5

[3.1 Engenharia de Software 1](#_Toc129265535)5

[3.2 Ciclo de Desenvolvimento de Software 16](#_Toc129265536)

[4. referencial Tecnológico 1](#_Toc129265537)7

[4.1 Linguagem de programação 1](#_Toc129265538)7

[4.2 Banco de dados 1](#_Toc129265539)7

[5 Modelagem do sistema 1](#_Toc129265540)8

[5.1 Diagrama de classes](#_Toc129265541) 21

[5.2 Diagrama de Casos de Uso 22](#_Toc129265542)

[5.3 Protótipos 23](#_Toc129265543)

[conclusão 2](#_Toc129265544)7

[Referências 2](#_Toc129265545)8

[Glossário Erro! Indicador não definido.](#_Toc129265546)

[Apêndice Erro! Indicador não definido.](#_Toc129265547)

[Anexos Erro! Indicador não definido.](#_Toc129265548)

# INTRODUÇÃO

A Declaração Universal dos Direitos dos Animais foi aprovada pela Organização das Nações Unidas para a Educação, a Ciência e a Cultura (UNESCO) e, posteriormente, pela Organização das Nações Unidas (ONU) em 15 de outubro de 1978. A proposta foi levada por ativistas da causa pela defesa dos direitos animais, que visa criar parâmetros jurídicos para os países membros da Organização das Nações Unidas, sobre os direitos animais. (ATAÍDE; FURLAN, 2021)

A pesar que existem direitos para os animais, atualmente as cifras de abandono continuam aumentando, de acordo com GEBARA, gerente de projetos da Ampara Animal, ONG parceira da Cobasi, o índice de abandono e de recolhimento de animais aumentou, em média, 61% entre julho de 2020 até o terceiro trimestre de 2021.

“Alguns protetores declararam aumento de abandono de 300%, de 150%, outros de 30%. Este dado se torna ainda mais agravante quando vemos que o número de doações também diminuiu por causa da pandemia, em que quase não houve eventos de adoção. A crise econômica e social exacerbou um problema antigo que é a falta de responsabilidade das pessoas com os animais. Então, quando a pessoa está passando por um momento difícil, a primeira coisa que ela faz é abandonar o mais vulnerável”, diz a gerente.

De acordo com a gerente, o Brasil não tem muitas pesquisas na área e, normalmente, o abandono é realizado de maneira escondida, largando cães e gatos em espaços públicos. Porém, estudos americanos mostram que as principais causas para o abandono dos pets são problemas comportamentais (47%) e mudanças no espaço ou na rotina (30%). O fato de deixar os pets, antes domesticados, nas ruas, gera sofrimento extremo, que afeta a saúde e o bem-estar do animal, além de deixá-los sujeitos a maus-tratos, atropelamentos e doenças, diminuindo sua expectativa de vida, sendo que a vida nas ruas gera um total desamparo para o pet sendo exposto a passar fome, sede, estresse, medo e angústia e faz com que a imunidade caia e aumente o risco de contraírem zoonoses. Portanto, o excesso de animais abandonados nas ruas também é um problema de saúde pública.

Tendo em conta estas circunstâncias e tentando dar uma resposta a essa problemática o propósito deste projeto é implementar um sistema para gerenciar as adoções destes animais que tem sido resgatado das ruas ou que foram expostos a maus tratos, recuperando-os até se tornarem novamente desejáveis pela sociedade. O sistema facilitará o processo de adoção destes animais o qual permitirá que as organizações que trabalham a favor deste propósito aumentem sua produtividade e possam atender um número maior de animais, conseguindo cobrir a demanda e consequentemente reduzir o número desses animais na rua, graças ao acolhimento dentro de uma família que os deseje adotar.

# OBJETIVOS

# Geral

Desta forma, este trabalho se justifica por auxiliar na gestão das organizações de proteção e acolhimento de animais abandonados, visando simplificar o processo de adoção. No entanto, o sistema a ser apresentado durante esse trabalho promete uma inovação que busca atender as necessidades dos interessados na adoção já que poderão fazer uma consulta no catálogo online por espécie, por porte, encontrando o animal de sua preferência entre os cadastrados. Existem muitas informações relevantes a serem contempladas como vacinas, castração, se convive bem com outros animais, entre outras informações importantes ao adotante. Assim, o sistema pretende oferecer uma melhor visualização que capte a atenção do adotante e aumente as probabilidades da adoção ser feita, além disso, permitirá ajudar em atividades específicas da instituição.

# Especifico

O objetivo específico é aumentar o número de adoções de animais de abrigos e de resgates, bem como de outros tipos de entidades de acolhimento de animais, a fim de reduzir o número de animais abandonados e/ou sacrificados.

# Fundamentação teórica

# 3.1 Engenharia de Software

O CatchMe é um sistema para a adoção de animais o qual permitirá que os interessados ​​em adotar animais façam a pesquisa de um animal adequado, aprovem a adoção e, finalmente, façam a adoção. Além disso, o sistema fornecerá informações importantes sobre o animal, como o tamanho, o comportamento, a saúde, etc. O sistema permitirá aos interessados ​​em adotar uma variedade de opções de filtragem para ajudá-los a encontrar o animal certo.

O sistema também permitirá que os abrigos de animais carreguem os animais à venda, incluindo informações sobre o animal, como a raça, a idade e a descrição. Além disso, o sistema terá um sistema de pagamento para que os interessados ​​possam pagar facilmente a taxa de adoção.

O sistema também terá uma funcionalidade para doação de insumos variados. Esta funcionalidade pode incluir pedidos de coleta.

Por fim, o sistema fornecerá recursos de informação para aqueles interessados ​​em adotar animais, incluindo guias sobre cuidado e comportamento apropriado para animais. Estes recursos ajudarão a garantir que os animais sejam cuidados adequadamente em suas novas casas.

**REQUISITOS FUNCIONAIS PARA O CADASTRO:**

* O Sistema deve pedir para qualquer usuário autenticar;
* O administrador manter e controlar as seguintes entidades dentro do sistema: usuário, estoque e fornecedor;
* O usuário pode realizar cadastrar de animais;
* O usuário pode visualizar perfil;
* O usuário pode alterar dados;
* O usuário pode fazer login;
* O usuário pode validar senha;
* O usuário pode visualizar catálogo;
* O usuário pode solicitar adoção;
* O usuário pode gerenciar catálogo.

**REQUISITOS NÃO FUNCIONAIS PARA O CADASTRO:**

* Os usuários devem possuir níveis de acesso a partir da categoria em que foi cadastrado;
* O sistema deve garantir a segurança dos dados utilizados no cadastro dos usuários, evitando exposição de dados sensíveis;
* Só deve ser possível cadastrar pessoas, animais e insumos depois de todos os campos preenchidos;
* Os animais cadastrados deverão possuir no mínimo uma foto.
* Os dados coletados dos animais devem ser flexíveis e podem ser utilizados para cães e gatos.

# 3.2 Ciclo de Desenvolvimento de Software

O software foi projetado para ajudar os abrigos de animais a se conectar com potenciais adotantes. O ciclo de desenvolvimento envolve projetar, desenvolver, testar e implementar um aplicativo que fornece aos usuários um meio eficaz de encontrar animais para adoção. O processo inclui a integração de recursos, como um sistema de busca para encontrar animais adequados, bem como a capacidade de gerenciar informações sobre os animais e seus potenciais adotantes. O objetivo do processo é proporcionar aos usuários uma experiência simples e intuitiva de procurar e adotar animais de abrigos.

1. Planejamento: Esta etapa envolve o planejamento do projeto, documentando os requisitos, definindo o alcance, definindo os recursos necessários, estabelecendo os prazos e orçamento e desenvolvendo um cronograma detalhado.

2. Análise: A etapa de análise envolve o entendimento das necessidades do usuário, a análise dos requisitos e a identificação de possíveis soluções.

3. Projeto: Esta etapa envolve a criação de um design detalhado do sistema, incluindo a especificação de interfaces, funcionalidades e regras de negócio.

4. Implementação: Esta etapa envolve a codificação do sistema, testes e depuração.

5. Testes: Esta etapa envolve a verificação do sistema para garantir que este cumpra com os requisitos estabelecidos.

6. Liberação: Esta etapa envolve a preparação do sistema para a implementação, incluindo a preparação dos arquivos de instalação, a criação de documentação e a realização de testes finais.

7. Manutenção: Esta etapa envolve a correção de erros, a atualização do sistema e a realização de melhorias.

# referencial Tecnológico

# 4.1 Linguagem de programação

As Stacks utilizadas foram **JavaScript, React, ReactNative** e **SQL** estas foram utilizadas por serem linguagens e tecnologias voltadas para programação Mobile. A vantagem de se utilizar **JavaScript** é deixar o aplicativo mais dinâmico. A principal vantagem em utilizar o **React Native** no desenvolvimento de aplicativos é que os projetos se tornam muito mais rápidos, já que o código é totalmente aproveitado entre as plataformas Android e iOS. E a principal vantagem de utilizar o **React** é por ele ser flexível e facilita a interface com outras bibliotecas e frameworks. Além disso, mesmo sendo uma biblioteca **SPA**, ele torna possível escrever aplicações que renderizam no servidor através do **Next.js**. Utilizar Banco de Dados é necessário pois é um sistema de logins e cadastros. Escolhemos o **PostgreSQL** pois suporta nativamente um grande número de tipos de dados padrão, como **JSON**, **XML**, etc. O PostgreSQL obtém vantagem disso, pois é um dos poucos bancos de dados relacionais que oferece forte suporte para a funcionalidade NoSQL. Além disso, permite que os usuários definam seus próprios tipos de dados.

# 4.2 Banco de dados

A linguagem utilizada para criar o banco de dados foi a SQL e a plataforma utilizada foi o Postgre.

O aplicativo foi desenvolvido para ajudar organizações de adoção de animais a gerenciar informações sobre os animais disponíveis para adoção. Ele armazena informações sobre o animal, incluindo sua raça, tamanho, cor, idade, bem como informações do adotante, como suas preferências e detalhes de contato. O banco de dados também pode ser usado para rastrear pagamentos relacionados à adoção e doação de insumos.

1. Modelo Conceitual:

O modelo conceitual representa as entidades e seus relacionamentos, sem se preocupar com detalhes de implementação. O modelo conceitual para o projeto de adoção de animais seria:

* Entidades:

Usuário: representa um usuário do sistema, que pode adotar animais.

Animal: representa um animal que pode ser adotado.

Instituição de Adoção: representa uma instituição que oferece animais para adoção.

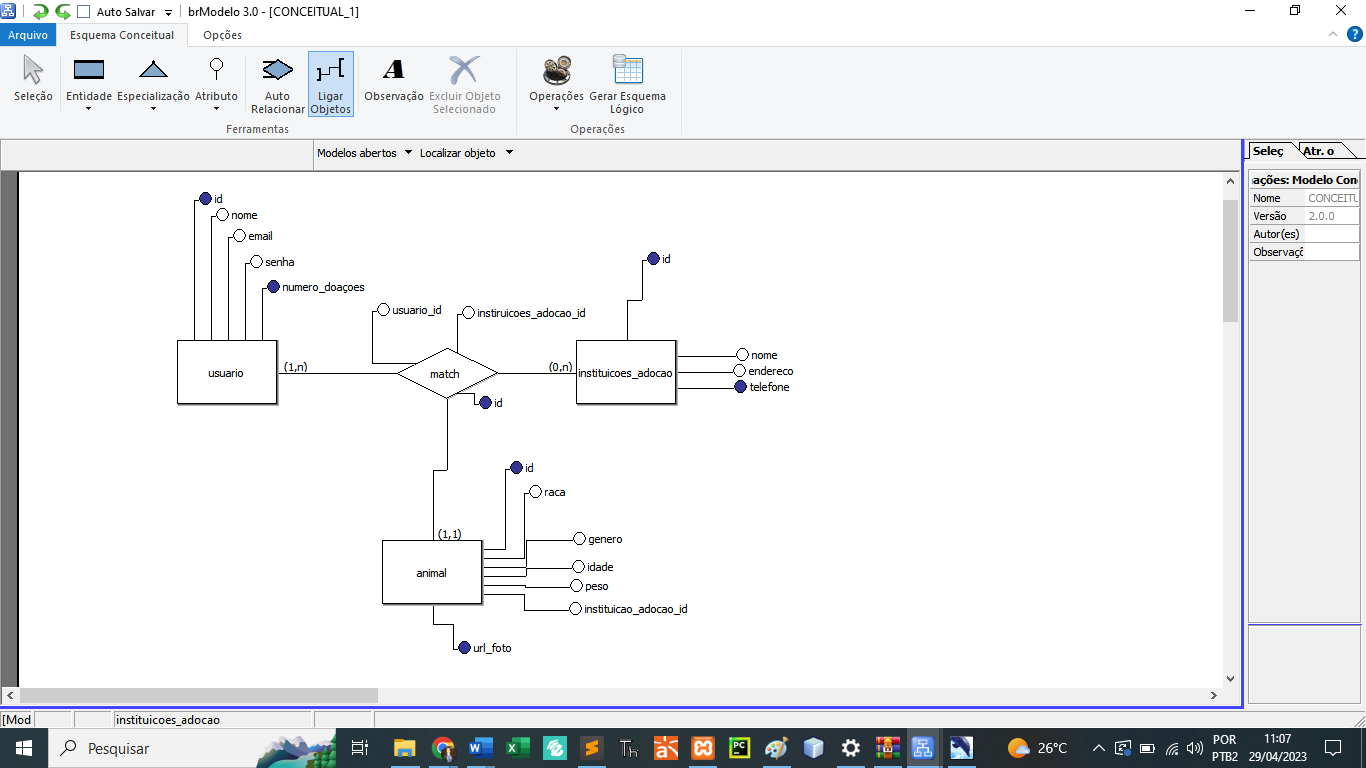
Match: representa uma adoção de um animal por um usuário.

* Relacionamentos:

Um usuário pode fazer muitos matches, mas um match pertence apenas a um usuário.

Um animal pode participar de muitos matches, mas um match pertence apenas a um animal.

Uma instituição de adoção pode ter muitos animais, mas um animal pertence apenas a uma instituição de adoção.



**Figura 1**: Modelo Conceitual

1. Modelo Lógico:

Este modelo é usado para definir a estrutura lógica dos dados. Ele usa diagramas de classes para representar as entidades e seus atributos, bem como as regras de negócio que as regem. O principal objetivo deste modelo é garantir que os dados sejam representados de forma consistente em todas as partes do sistema.

Tabelas:

* usuarios: representa a entidade Usuário do modelo conceitual.

colunas: id (chave primária), nome, email (único), senha, numero\_adocoes.

* animais: representa a entidade Animal do modelo conceitual.

colunas: id (chave primária), nome, idade, raca, genero, peso, url\_foto, instituicao\_adocao\_id (chave estrangeira referenciando a tabela instituicoes\_adocao).

* instituicoes\_adocao: representa a entidade Instituição de Adoção do modelo conceitual.

colunas: id (chave primária), nome, endereco, telefone.

* matches: representa a entidade Match do modelo conceitual.

colunas: id (chave primária), usuario\_id (chave estrangeira referenciando a tabela usuarios), animal\_id (chave estrangeira referenciando a tabela animais).

1. Modelo Físico:

Este modelo é usado para representar a estrutura física dos dados. Ele usa diagramas de banco de dados para mostrar como os dados serão armazenados na memória, bem como como os dados serão recuperados e manipulados. Este modelo é particularmente importante para os desenvolvedores que precisam entender como os dados serão lidos e gravados.

* Criação da tabela de usuários

CREATE TABLE usuarios (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL UNIQUE,

senha VARCHAR(255) NOT NULL,

numero\_adocoes INT NOT NULL DEFAULT 0

);

* Criação da tabela de instituições de adoção

CREATE TABLE instituicoes\_adocao (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

endereco VARCHAR(255) NOT NULL,

telefone VARCHAR(20) NOT NULL

);

* Criação da tabela de animais cadastrados

CREATE TABLE animais (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

idade INT NOT NULL,

raca VARCHAR(255) NOT NULL,

genero ENUM('M', 'F') NOT NULL,

peso FLOAT NOT NULL,

url\_foto VARCHAR(255) NOT NULL,

instituicao\_adocao\_id INT NOT NULL,

FOREIGN KEY (instituicao\_adocao\_id) REFERENCES instituicoes\_adocao(id)

);

* Criação da tabela de matches

CREATE TABLE matches (

id INT PRIMARY KEY AUTO\_INCREMENT,

usuario\_id INT NOT NULL,

animal\_id INT NOT NULL,

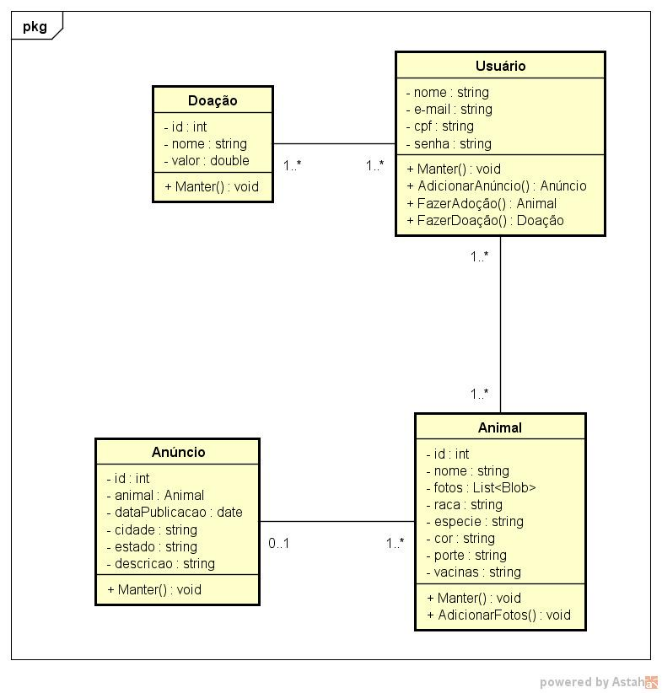
FOREIGN KEY (usuario\_id) REFERENCES usuarios(id),

FOREIGN KEY (animal\_id) REFERENCES animais(id)

);

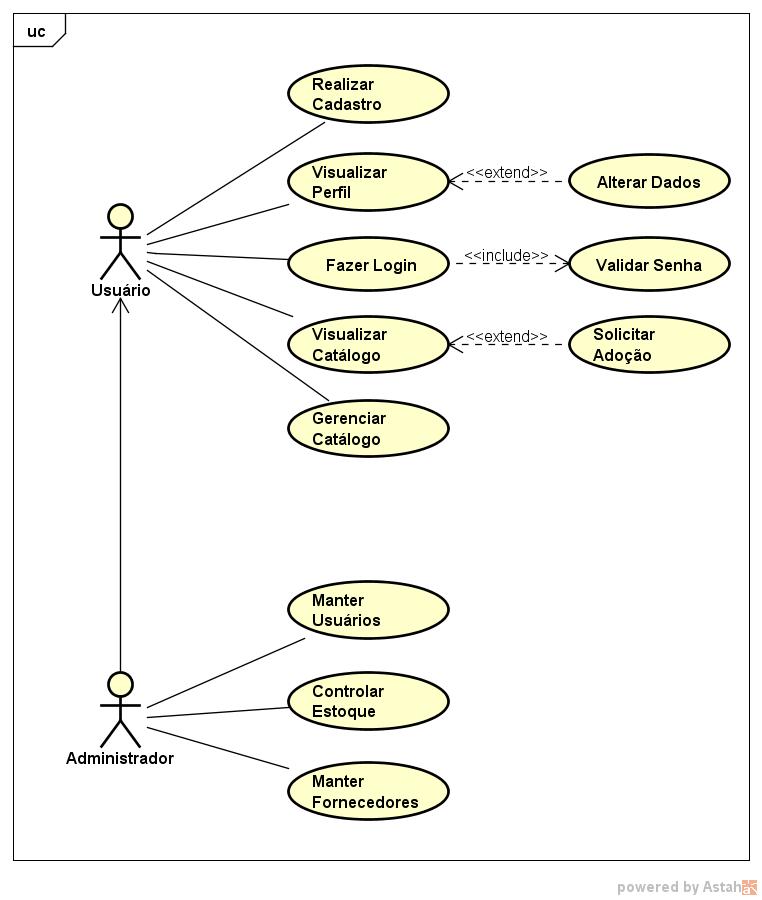
# 5 Modelagem do sistema

# 5.1 Diagrama de classes



**Figura 2**: Diagrama de Classe

# 5.2 Diagrama de Casos de Uso



**Figura 3**: Caso de Uso

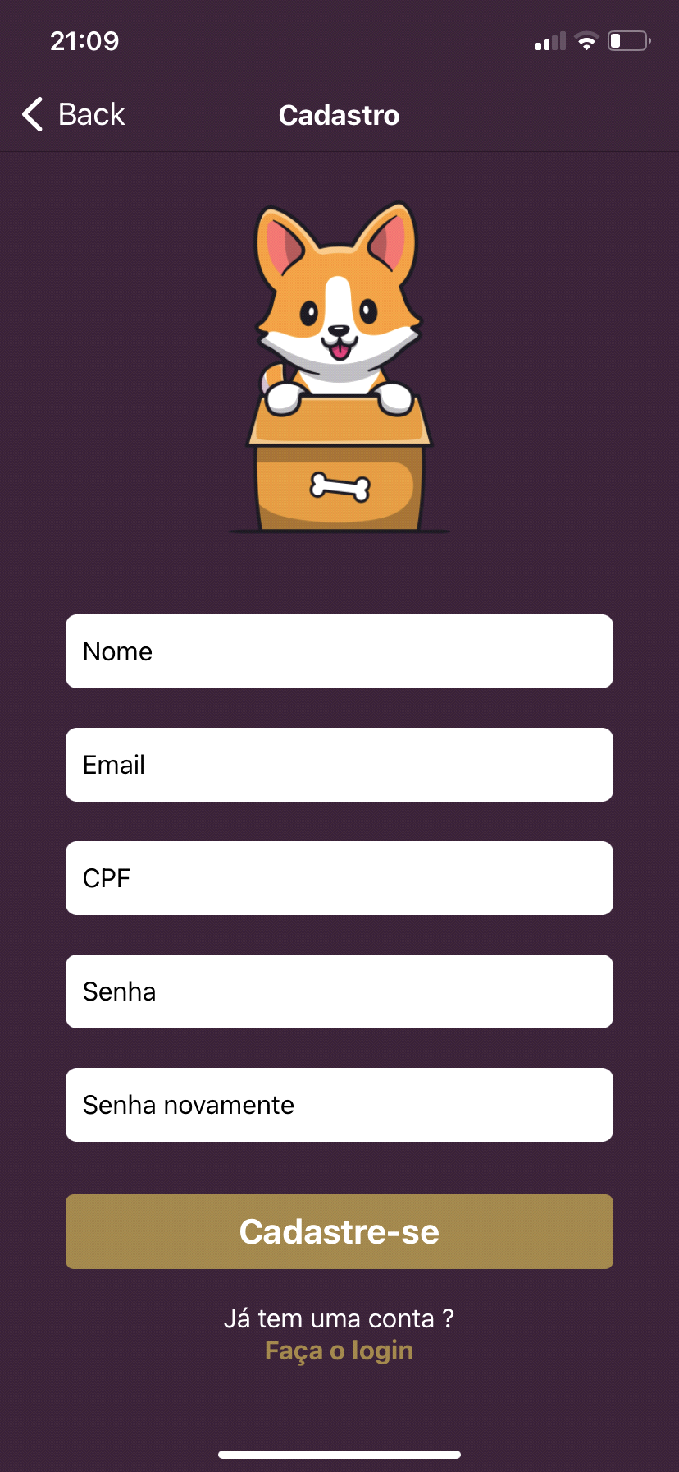
# 5.3 Protótipos

7.4.1) Tela Inicial



**Figura 4**: protótipo tela inicial

7.4.2) Cadastrar Cliente



**Figura 5**: protótipo cadastrar cliente

7.4.3) Login Cliente



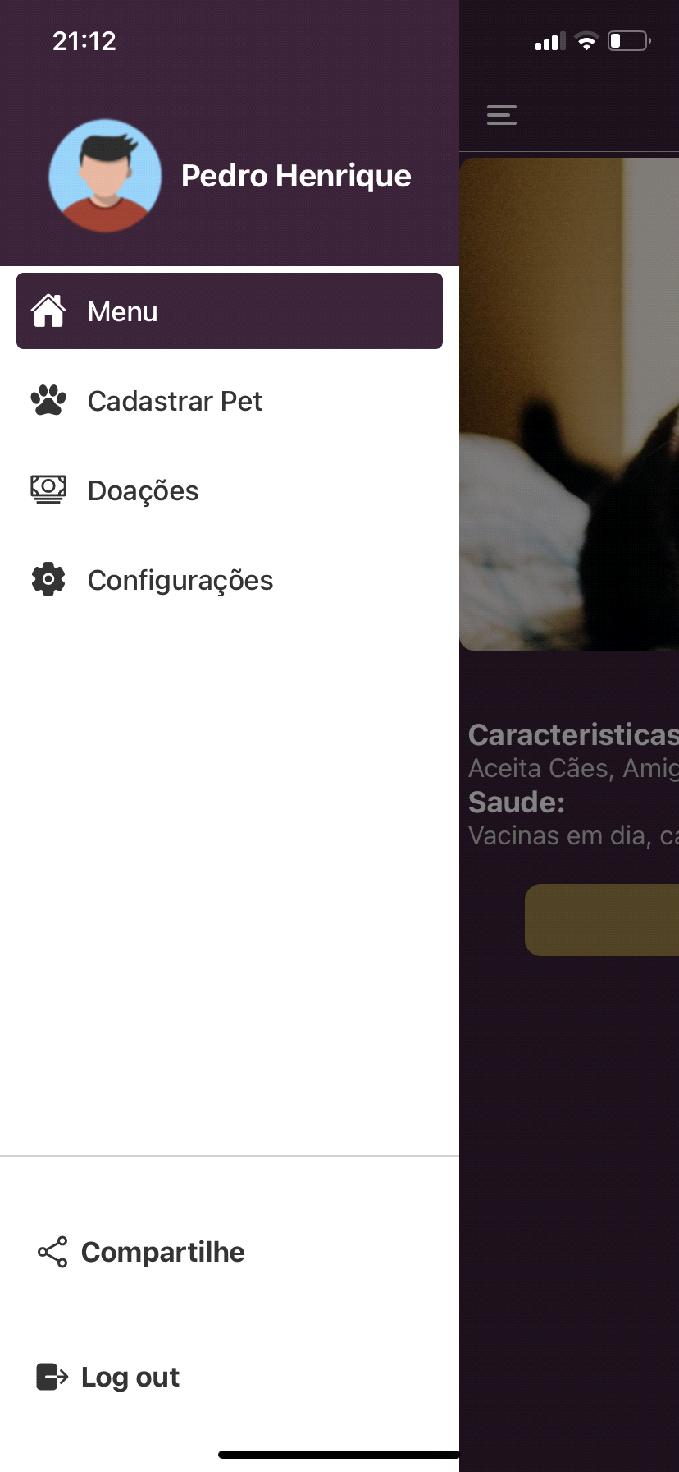
**Figura 6**: protótipo login cliente

7.4.4) Informações Pet



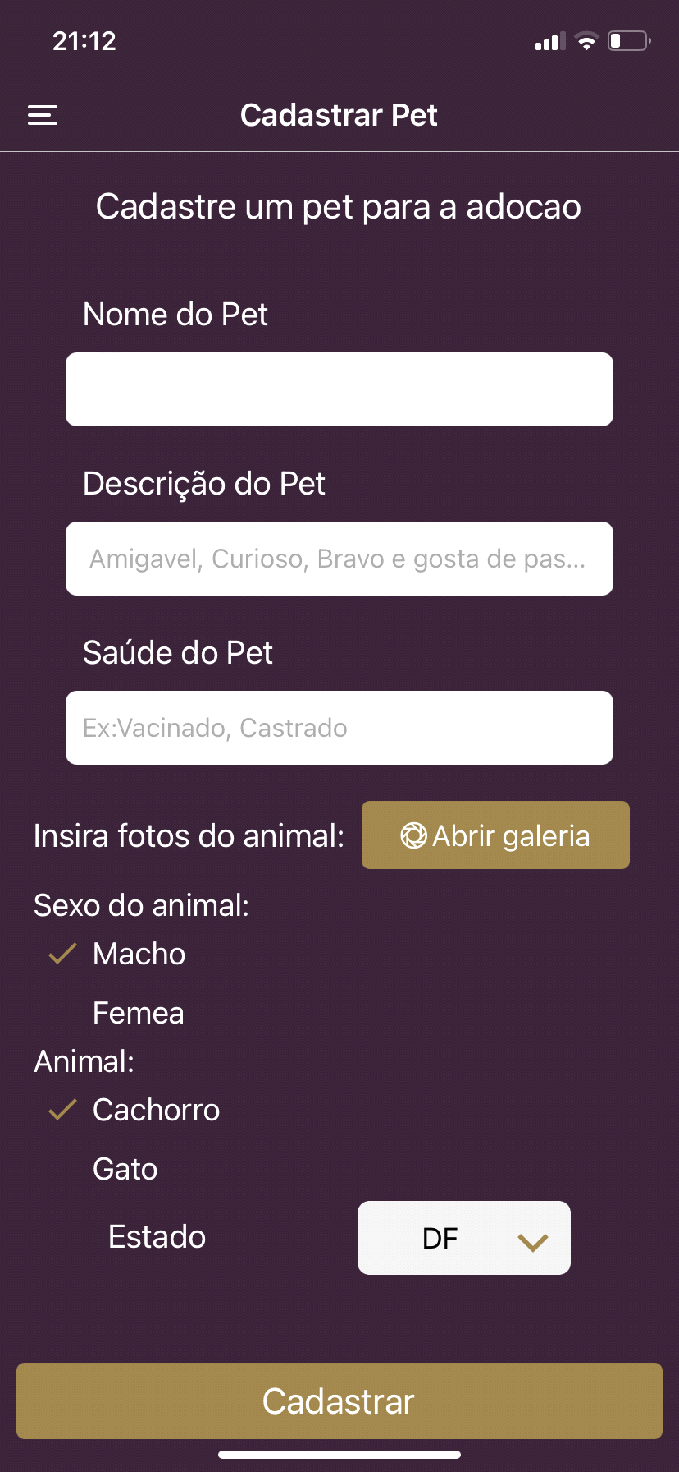
**Figura 7**: protótipo informações pet

7.4.5) Barra de navegação



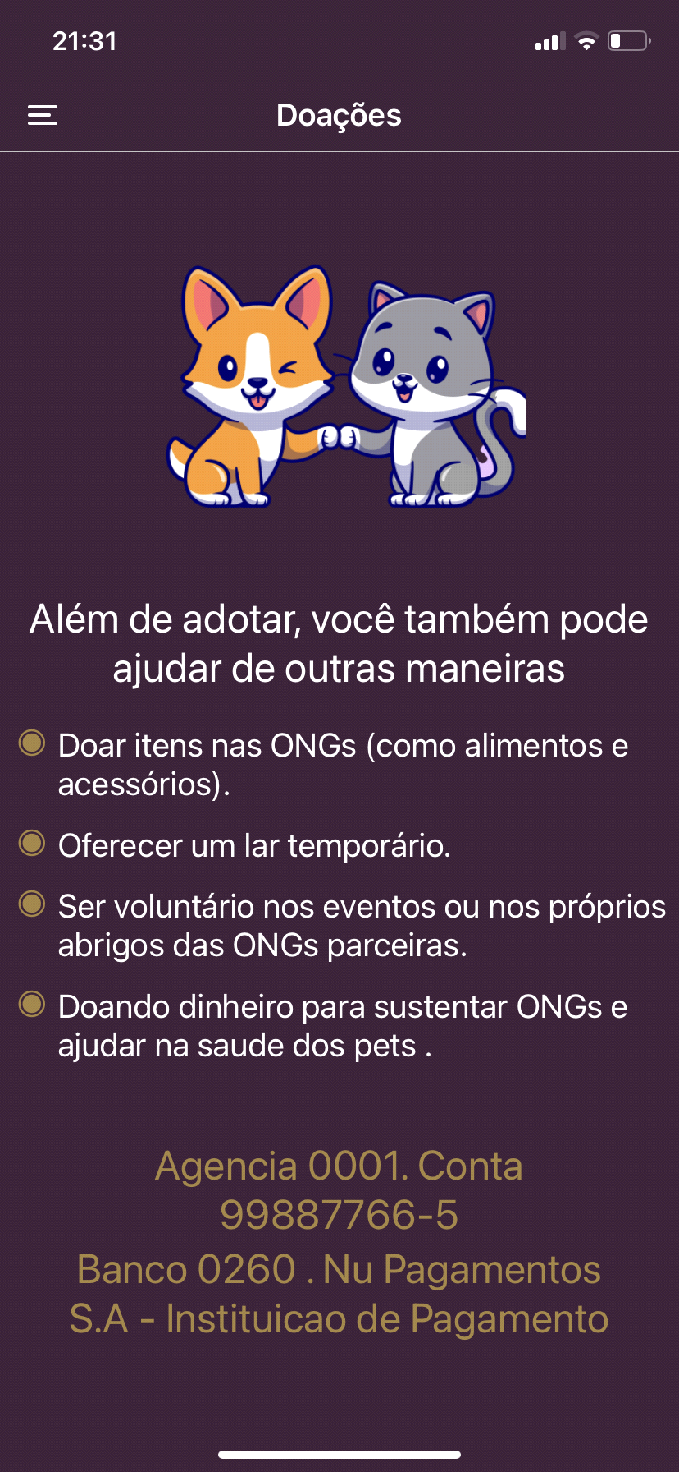
**Figura 8**: protótipo barra de navegação

7.4.6) Cadastro de Pets



**Figura 9**: protótipo cadastrar pet

7.4.7) Doações



**Figura 10**: protótipo cadastrar cliente

# conclusão

Este trabalho teve como propósito desenvolver o “CatMe”, o qual se trata de aplicativo cuja finalidade foi facilitar o desempenho do gerenciamento de adoção de animais dentro de organizações que implementam atividades que buscam atender este desígnio. O aplicativo traz diversas vantagens como o aumento da produtividade nas organizações, de tal maneira que possibilitam atender um número maior de animais, conseguindo cobrir a demanda e consequentemente reduzir o número desses animais na rua. Além disso, o aplicativo apresenta o catálogo de animais disponíveis para a adoção, o que proporciona uma melhor visualização dos perfis dos animais e captura com maior facilidade a atenção daquelas pessoas cujo interesse é adotar um animal de estimação.

Qualquer pessoa pode se cadastrar no aplicativo e ter acesso ao catálogo de animais, podendo realizar uma adoção ou postar um novo anúncio, com características específicas tanto dos usuários como dos animais que logo serviram para o controle e análise de dados, por um lado para as pesquisas de animais dentro do aplicativo e por outro para a tomada de decisões das solicitações, podendo ser aprovadas ou rejeitadas.

Outra vantagem da implementação deste aplicativo é o aumento da interação entre as organizações e as pessoas que têm interesses pelos animais, de tal forma que não só se trata de adotar animais, senão também abrange fazer divulgação com o propósito de incentivar as pessoas para fazer doações, seja de produtos ou monetárias, assim como aderir-se às atividades das organizações.

Do ponto de vista acadêmico o desenvolvimento deste projeto traz como benefícios o aprimoramento técnico que outorga o amadurecimento de cada um dos membros da equipe, graças aos conhecimentos que foram adquiridos na elaboração deste projeto.

# Referências

ATAÍDE, Vicente; FURLAN, Anderson. Dia internacional dos direitos dos animais. Seção judiciária do Paraná, 30, dezembro, 2021. Disponível em: <https://www.jfpr.jus.br/wp-content/uploads/2021/12/10-de-dezembro-Dia-Internacional-dos-Direitos-dos-Animais.pdf>

GEBARA, Rosangela. Por que não devemos comprar cães e gatos?. AMPARA, Brasil, outubro, 2021. Disponível em: <https://www.amparanimal.org.br/blog/por-que-nao-devemos-comprar-caes-e-gatos>

POSSEBON, Cláudia; SOUZA, Wagner. Sistemas de informação gerencial. Brasil: Editora Senac São Paulo, 2020. Disponível em: <https://www.google.com.br/books/edition/Sistemas_de_informa%C3%A7%C3%A3o_gerencial/soUEEAAAQBAJ?hl=pt-BR&gbpv=0>

LEAL, Tiago; NETO, Anderson. Sistemas de informação gerenciais e sua importância para tomada de decisões. 2008. Disponível em: <https://administradores.com.br/artigos/sig-e-sua-importancia-para-tomada-de-decisoes>

BIVAR, Wasmália et al. Pesquisa Domiciliar sobre cãe e gatos e Humanização e Padrões de Consumo. 2007. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv39560.pdf>

# GLOSSARIO

**Unesco** - Organização das Nações Unidas para a Educação, a Ciência e a Cultura **Onu** - Organização das Nações Unidas

***CatchMe*** -Nome do aplicativo

***Software*** - programa de computador que permite ao usuário interagir com o computador para realizar diversas tarefas ou processos. Ele pode ser usado para realizar tarefas simples como navegar na internet ou tarefas mais complexas como a programação de um sistema.

***Stacks*** - estrutura de dados baseada em pilha, caracterizada por duas operações básicas, a inserção (push) e a remoção (pop) de elementos, sendo que a remoção é realizada sempre do elemento que foi inserido por último. É uma estrutura de dados importante e útil para a realização de diversas tarefas, como por exemplo, a resolução de problemas de programação.

***JavaScript*** - linguagem de programação interpretada de alto nível, com tipagem dinâmica fraca e multi-paradigma. É utilizada principalmente para criar e controlar conteúdos dinâmicos na World Wide Web, como formulários interativos, animações, games e aplicativos em geral.

***React*** - uma biblioteca JavaScript de código aberto para criar interfaces de usuário. Ela foi criada pela equipe do Facebook para facilitar a criação de interfaces de usuário complexas em aplicações web e mobile. React usa códigos reutilizáveis ​​para criar componentes de interface de usuário que podem ser compartilhados entre aplicativos. Os componentes podem ser facilmente manipulados para atualizar e renderizar partes dinâmicas das interfaces de usuário.

***ReactNative*** - uma biblioteca JavaScript criada pela equipe de engenharia do Facebook. Ele permite a criação de aplicativos nativos para iOS e Android usando o mesmo código JavaScript. React Native usa o mesmo design que o React, simplificando o desenvolvimento de aplicativos para dispositivos móveis.

***SQL*** - Structured Query Language é uma linguagem de programação padronizada usada para acessar, manipular e gerenciar bancos de dados relacionais, como Oracle, Sybase, Informix, Microsoft SQL Server, Access, entre outros. Ela permite aos usuários criar e executar consultas, inserir, alterar e excluir dados.

***Android*** - sistema operacional móvel desenvolvido pelo Google e baseado no núcleo Linux. Foi lançado em dezembro de 2008, com o objetivo de proporcionar uma plataforma móvel para dispositivos como smartphones e tablets.

***iOS*** - sistema operacional móvel da Apple Inc. para dispositivos iPhone, iPad e iPod touch.

***frameworks*** - conjunto de ferramentas que ajuda os desenvolvedores a criar aplicativos móveis, web e desktop. Ele fornece uma estrutura base de código, bibliotecas e interfaces, bem como guias de melhores práticas para ajudar os desenvolvedores a criar aplicativos de maneira mais eficiente.

***SPA*** - Single Page Application é uma biblioteca de software que permite aos desenvolvedores criar aplicativos web de uma única página. Esta biblioteca permite aos desenvolvedores criarem aplicativos web que reagem rapidamente ao usuário, com carregamento único, atualização dinâmica da interface do usuário e carregamento de conteúdo em segundo plano. Os aplicativos SPA também são muito flexíveis, pois permitem que os desenvolvedores criem aplicativos web que sejam executados a partir de qualquer dispositivo, incluindo PCs, laptops, tablets e smartphones.

***Renderizam*** - processo de transformar dados gráficos em imagens visíveis por meio de computador. É um processo computacional que converte dados de uma forma que possa ser exibida em um dispositivo de saída. É usado em muitos campos, incluindo computação gráfica, computação científica, computação medicina e programação de jogos.

***Next.js*** - biblioteca de código aberto para criar aplicativos da web reactivos. Ele oferece suporte ao servidor de renderização para criar páginas web que se carregam mais rapidamente, melhorando o desempenho do SEO. Ele também permite aos desenvolvedores criar aplicativos reutilizáveis e escaláveis com sintaxe e APIs consistentes.

***PostgreSQL*** -sistema de gerenciamento de banco de dados relacional de objetos (RDBMS) desenvolvido como software de código aberto. É um dos principais bancos de dados relacionais mais populares, com suporte para um grande número de linguagens de programação, como SQL, Python e JavaScript. Além disso, PostgreSQL fornece ferramentas poderosas para gerenciar grandes volumes de dados.

***JSON*** - JavaScript Object Notation é um formato de texto leve para armazenar e trocar dados entre computadores. É facilmente lido e escrito por humanos e máquinas, tornando-o ideal para transportar dados na web. É uma das principais alternativas para XML. É baseado em objetos JavaScript, mas é independente da linguagem de programação.

***XML*** - Extensible Markup Language é uma linguagem de marcação de dados desenvolvida para armazenar e transportar informações. É amplamente utilizada para intercâmbio de informações entre aplicações, pois possui uma estrutura hierárquica que facilita a análise e interpretação dos dados.

**APÊNDICE A –** modelo físico do banco de dados

 Criação da tabela de usuários

CREATE TABLE usuarios (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL UNIQUE,

senha VARCHAR(255) NOT NULL,

numero\_adocoes INT NOT NULL DEFAULT 0

);

 Criação da tabela de instituições de adoção

CREATE TABLE instituicoes\_adocao (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

endereco VARCHAR(255) NOT NULL,

telefone VARCHAR(20) NOT NULL

);

 Criação da tabela de animais cadastrados

CREATE TABLE animais (

id INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

idade INT NOT NULL,

raca VARCHAR(255) NOT NULL,

genero ENUM('M', 'F') NOT NULL,

peso FLOAT NOT NULL,

url\_foto VARCHAR(255) NOT NULL,

instituicao\_adocao\_id INT NOT NULL,

FOREIGN KEY (instituicao\_adocao\_id) REFERENCES instituicoes\_adocao(id)

);

 Criação da tabela de matches

CREATE TABLE matches (

id INT PRIMARY KEY AUTO\_INCREMENT,

usuario\_id INT NOT NULL,

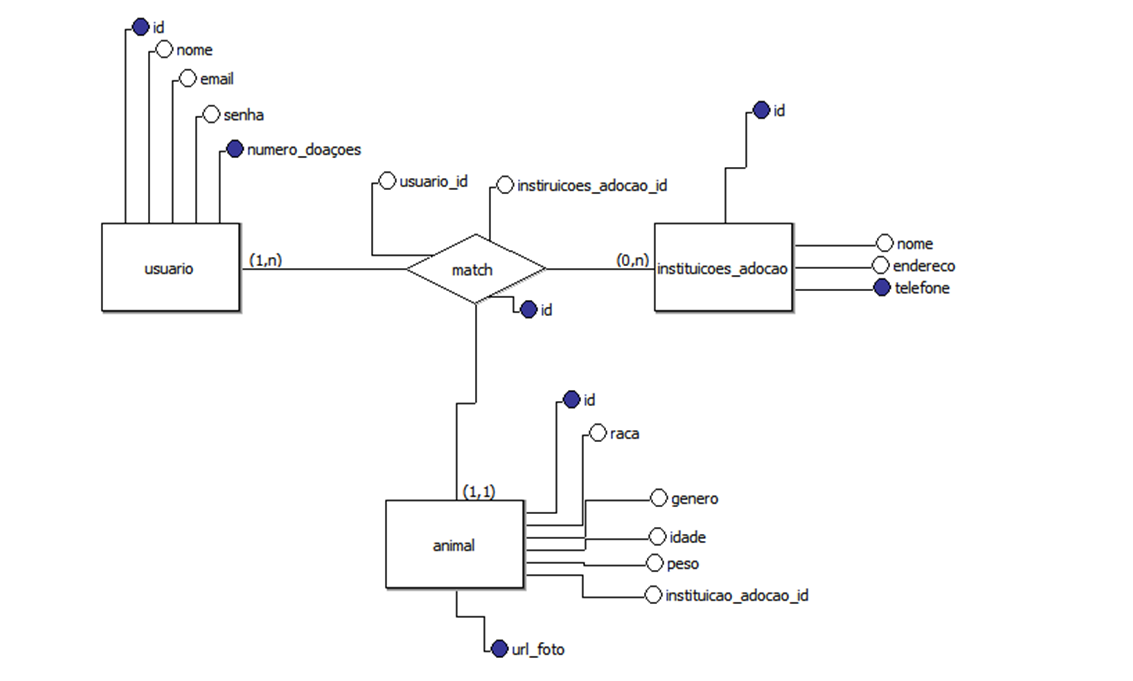
animal\_id INT NOT NULL,

FOREIGN KEY (usuario\_id) REFERENCES usuarios(id),

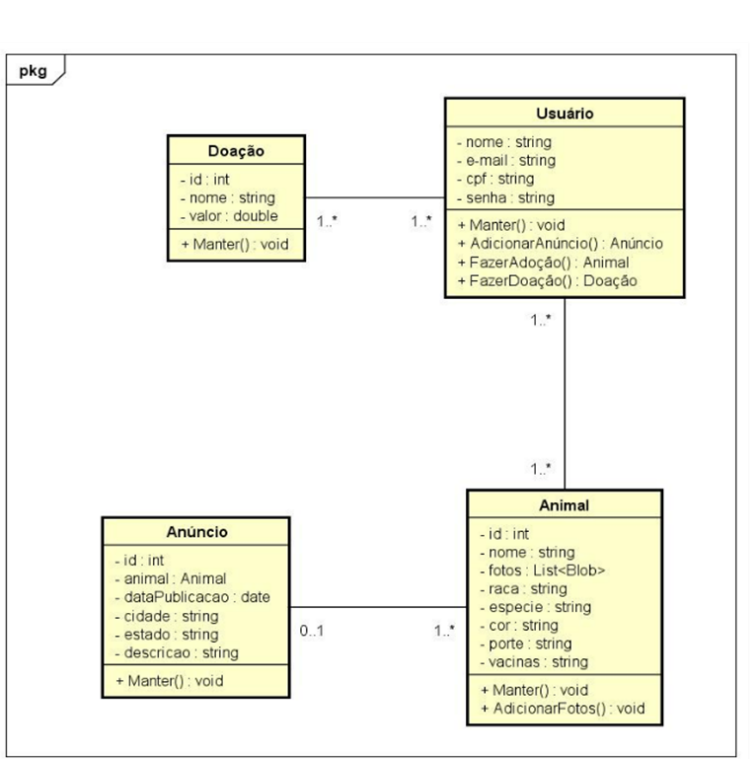
FOREIGN KEY (animal\_id) REFERENCES animais(id)

);

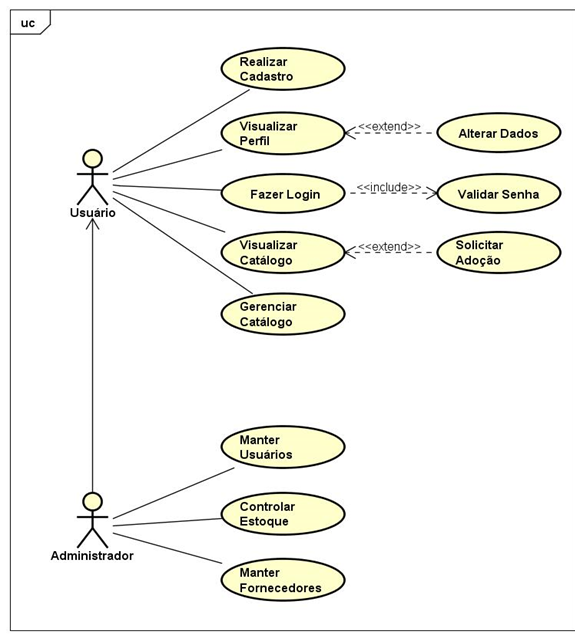
**ANEXOS**



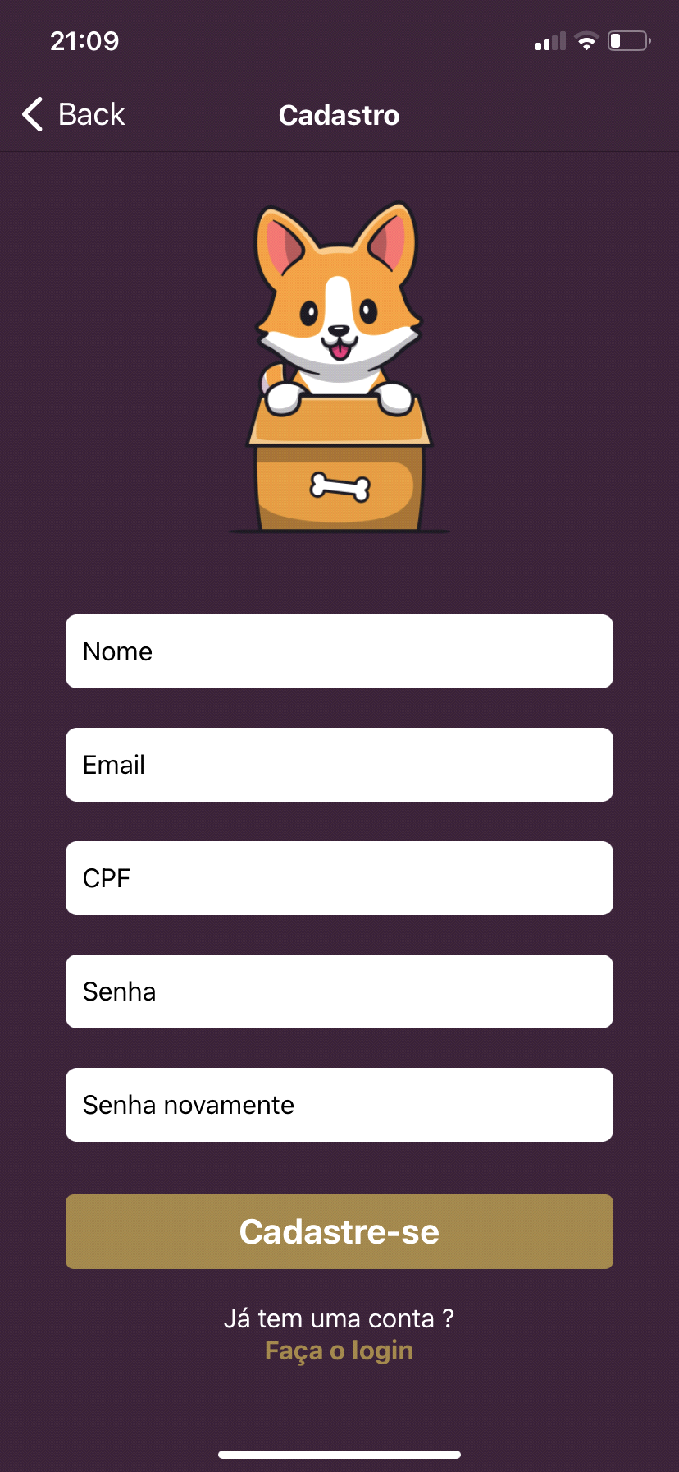




**Figura 2**: Diagrama de Classe



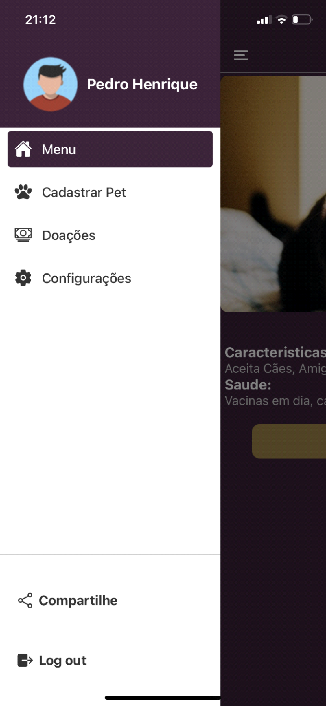
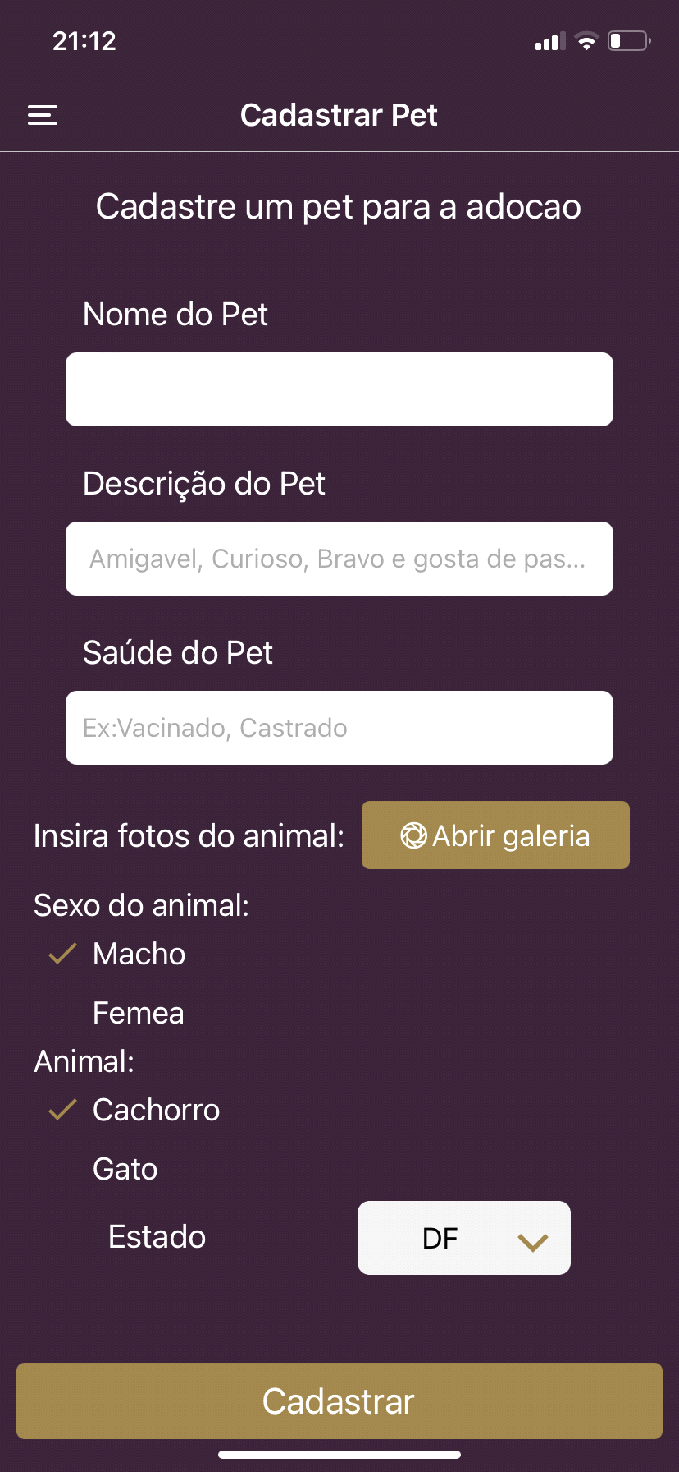
**Figura 3**: Caso de Uso

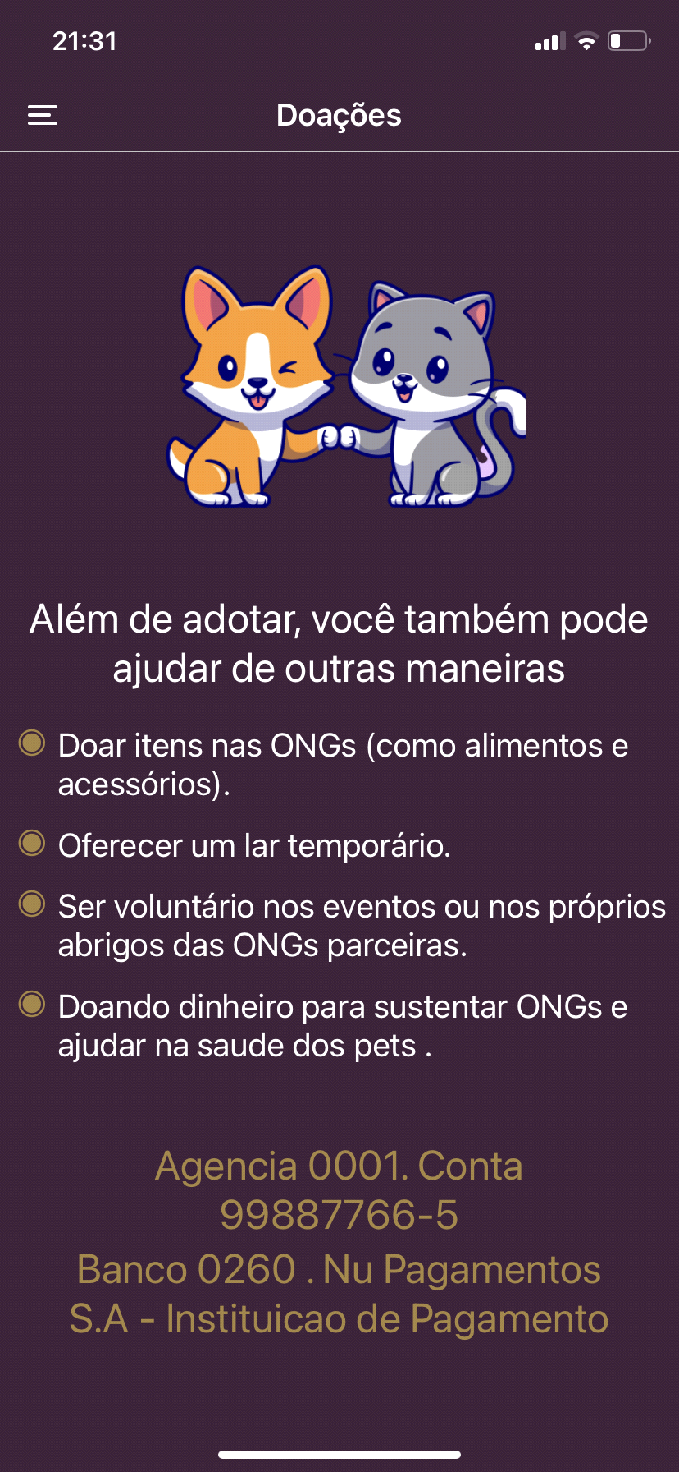
**Figura 4**: protótipo tela inicial **Figura 5**: protótipo cadastrar cliente

**Figura 6**: protótipo login cliente **Figura 7**: protótipo informações pet

**Figura 8**: protótipo barra de navegação **Figura 9**: protótipo cadastrar pet



**Figura 10**: protótipo cadastrar cliente