



DOCUMENTAÇÃO DO SISTEMA ADOTE UM AMIGO

*Projeto Integrador III
Tecnologia em Sistemas para Internet*

ano/semestre

Sumário

1. Equipe e gestão do projeto.....	3
2. Descrição, características e usuários do sistema (Sprint 1).....	3
3. Requisitos (Sprint 1).....	3
3.1. Requisitos funcionais.....	3
3.2. Requisitos NÃO funcionais.....	4
3.3. Suporte a estatística (Sprint 2).....	4
4. Diagrama de casos de uso (Sprint 1).....	4
5. Banco de dados (Sprint 1 e 2).....	4
5.1. Modelo relacional diagramado (Sprint 1).....	4
5.2. Modelo físico do banco de dados (Sprint 2).....	4
6. Front-end (Sprint 1 e 2).....	4
6.1. Protótipos (Sprint 1).....	5
6.2. Implementação (Sprint 2).....	5
7. Back-end (Sprint 2 e 3).....	5
7.1. Diagrama de classes (Sprint 2).....	5
7.2. Implementação do back-end (Sprint 3).....	6
8. Integração front-end e back-end (Sprint 4).....	6
9. Referências.....	7

1. Equipe e gestão do projeto

Davi Campos Parente: Gerente de Projeto / Desenvolvedor Backend

Ivanilson Paixao Cirqueira: Analista de Sistemas

João Henrique: Desenvolvedor Front End

Divisão das Tarefas: As tarefas são divididas com base nas especializações de cada membro da equipe. O Gerente de Projeto (Davi Campos Parente) cria um cronograma inicial e aloca tarefas conforme a expertise de cada membro.

Comunicação do Andamento: O andamento de cada tarefa é comunicado via a ferramenta de gestão de projetos Trello, onde todos os membros atualizam o status de suas respectivas tarefas. Além disso, utilizamos o Whatsapp para discussões rápidas e atualizações de progresso em tempo real.

Registro das Tarefas: As tarefas a serem realizadas são registradas no Trello. Cada tarefa é detalhada com descrições, prazos e responsáveis, garantindo que todos saibam exatamente o que precisa ser feito e quando.

Reuniões da Equipe: Realizamos reuniões semanais toda quinta-feira antes da aula para planejar a semana e revisar o progresso da semana anterior.

2. Descrição, características e usuários do sistema (Sprint 1)

A aplicação "Projeto Acalanto-DF" é uma plataforma online criada por uma ONG dedicada à proteção e bem-estar de gatos. O objetivo da aplicação é facilitar a adoção de gatos, centralizando informações sobre gatos disponíveis para adoção e promovendo uma adoção responsável e informada.

Através do site, os usuários interessados em adotar um gato podem acessar perfis detalhados dos gatos disponíveis, incluindo fotos, idade e observações gerais. Além disso, a aplicação oferece um processo de adoção simplificado, com formulários de solicitação online, informações sobre os requisitos para adoção e etapas claras do processo. A aplicação também proporciona mecanismos de comunicação direta entre os adotantes em potencial e equipe da ONG Acalanto, facilitando dúvidas, agendamento de visitas e acompanhamento do processo de adoção.

Para promover a educação e a conscientização, o site oferece recursos educativos sobre a importância da adoção responsável, cuidados com gatos e bem-estar animal.

A aplicação "Projeto Acalanto-DF" é direcionada a diversos grupos, todos interessados em melhorar a vida dos gatos:

1. **Adotantes em Potencial:** Pessoas interessadas em adotar um novo animal de estimação.
2. **Amantes de Animais:** Indivíduos que amam gatos e desejam ajudar na causa de encontrar lares para animais abandonados.
3. **Comunidade em Geral:** Qualquer pessoa interessada em aprender mais sobre a adoção de cães e gatos.

3. Requisitos (Sprint 1)

1. Facilidade de Navegação:

- o A aplicação deve ter uma interface amigável e intuitiva, permitindo que os usuários naveguem por todas as seções.

2. Gestão de Gatos:

- o Por meio da Gestão de Gatos, por parte do Administrador, será possível realizar as operações de Criar, Selecionar, Alterar e Deletar gatos.

3. Formulário de Solicitação de Adoção:

- o O usuário poderá manifestar interesse em adotar determinado gato.

4. Gestão de Solicitações de Adoções:

- o Por meio da Gestão de Solicitações de Adoções, o Administrador poderá ver quem são os interessados em adotar os animais.

5. Informar o público-alvo:

- o Potenciais dúvidas sobre o processo de adoção e projeto poderão ser sanadas na sessão “Sobre”.

3.1. Requisitos funcionais

1. Login:

Para o perfil de Administrador, será possível realizar login. Com ele, será possível ter acesso às guias de Gestão de Gatos e Gestão de Solicitações de Adoções.

2. Cadastrar Gatos:

O administrador poderá cadastrar gatos na aplicação.

3. Selecionar Gatos:

Será possível filtrar os gatos de acordo com os seus atributos.

4. Alterar Gatos:

Em caso de erros nos atributos, será possível realizar a alteração das informações.

5. Deletar Gatos:

Em caso de necessidade de apagar algum gato que, por motivo indefinido, tenha sido inserido inadequadamente, poderá ser realizado o procedimento de deletamento.

6. Adotar Gatos

O Usuário poderá manifestar interesse em adotar algum gato cadastrado na aplicação.

7. Gestão de Solicitações de Adoções

O administrador poderá visualizar os interessados em adotar algum gato.

3.2. Requisitos NÃO funcionais

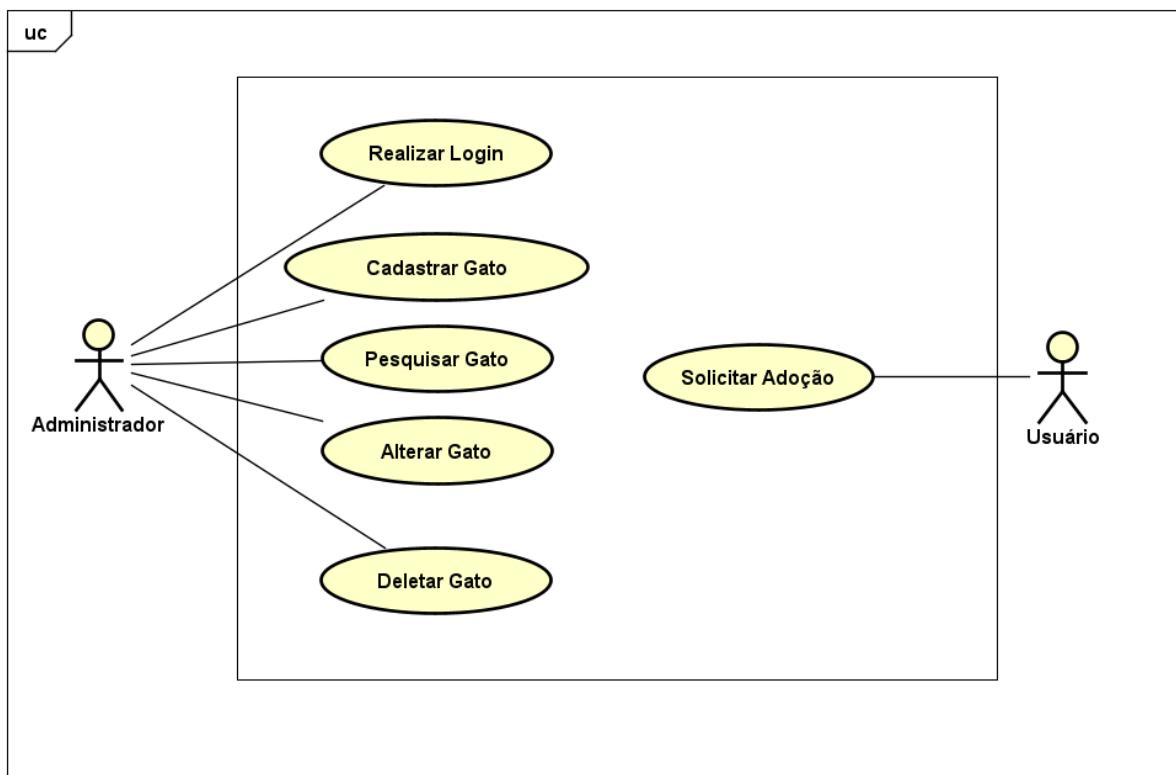
- 1. Segurança:** Garantir a segurança dos dados pessoais dos usuários, utilizando criptografia.
- 2. Desempenho:** Capacidade de lidar com múltiplos acessos simultâneos sem comprometer a qualidade.
- 3. Usabilidade:** Interface intuitiva e fácil de usar para uma experiência positiva.
- 4. Escalabilidade:** A aplicação deve ser escalável para adicionar novas funcionalidades conforme a demanda.
- 5. Manutenibilidade:** Desenvolvimento modular para facilitar futuras manutenções e atualizações.
- 6. Confiabilidade:** A aplicação deve ser confiável, minimizando o tempo de inatividade.
- 7. Privacidade:** Respeitar a privacidade dos usuários, cumprindo regulamentações de proteção de dados.

3.3. Suporte a estatística (Sprint 2)

Nessa parte, podem ser implementados recursos para apoiar a realização dos testes do produto com possíveis usuários, coletar dados como número de acessos, partes do produto mais acessadas, tempo de acesso, entre outras. Montar um questionário pós teste do aplicativo, com perguntas para melhoria do produto de acordo com o perfil dos usuários. Utilizar o Python para cálculo das medidas estatísticas, principalmente média e desvio padrão. Realizar testes de comparação estatísticos através dos testes estatísticos estudados pela turma. Isso também poderá ser realizado com o Python.

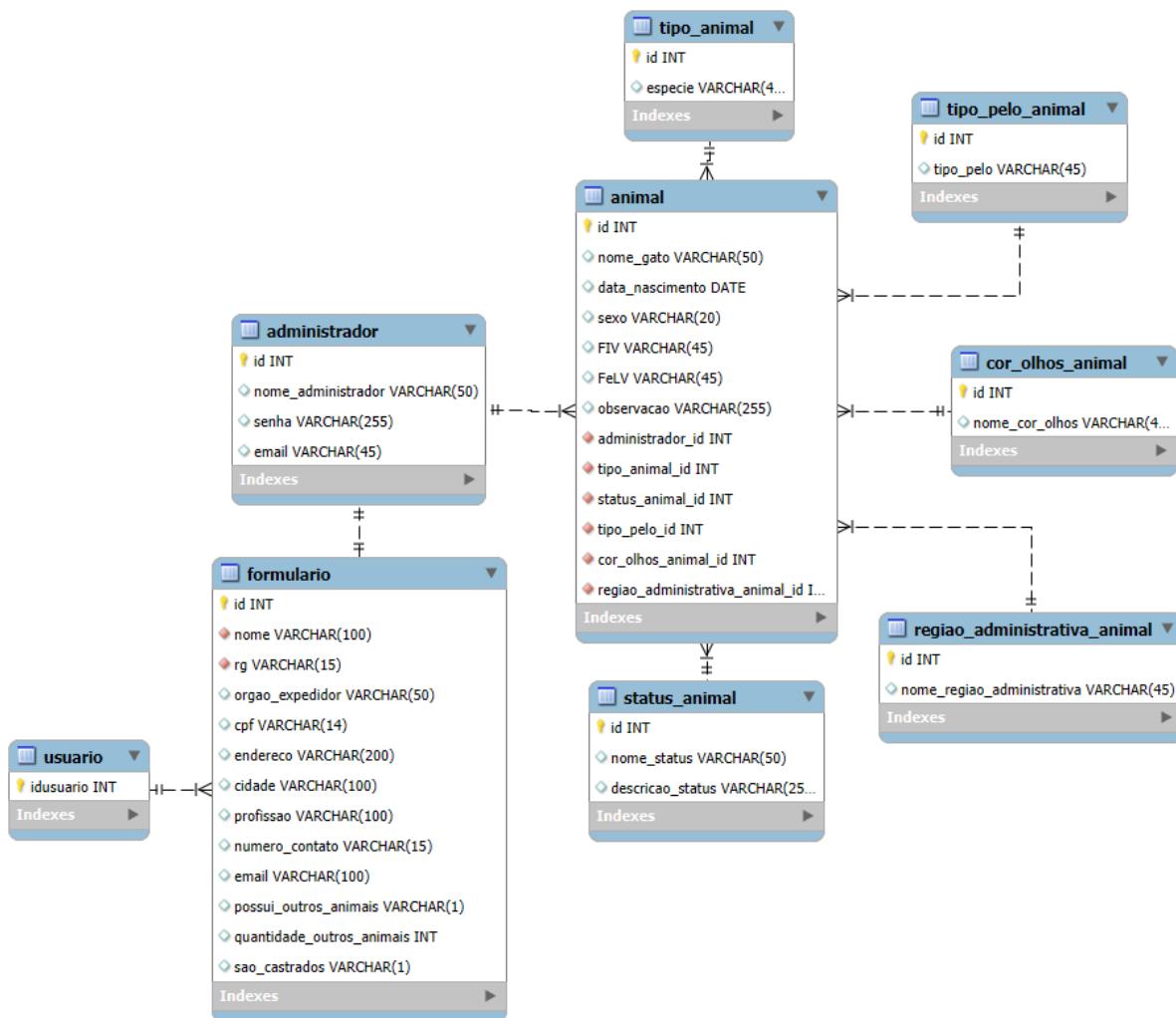
<https://trello.com/invite/b/6717028992588b8e8442c2f5/ATT1eb1fb134eff53c7b2dd37a27eeb6ad6178526BC4/pi-3>

4. Diagrama de casos de uso (Sprint 1)



5. Banco de dados (Sprint 1 e 2)

5.1. Modelo relacional diagramado (Sprint 1)



Segue exemplos de inserções nas tabelas:

```

INSERT INTO tipo_pelo_animal (id, tipo_pelo) VALUES
(1, 'Longo'),
(2, 'Curto'),
(3, 'Liso'),
(4, 'Cacheado'),
(5, 'Sedoso'),
(6, 'Denso'),
(7, 'Aveludado'),
(8, 'Ralo');

```

```

INSERT INTO cor_olhos_animal (id, nome_cor_olhos) VALUES
(1, 'Azul'),
(2, 'Verde'),
(3, 'Amarelo'),
(4, 'Castanho-claro'),

```

(5, 'Castanho-escuro'),
(6, 'Âmbar'),
(7, 'Avelã'),
(8, 'Heterocromia (duas cores diferentes)'),
(9, 'Cinza'),
(10, 'Preto');

INSERT INTO regiao_administrativa_animal (id, nome_regiao_administrativa) VALUES
(1, 'Plano Piloto'),
(2, 'Gama'),
(3, 'Taguatinga'),
(4, 'Brazlândia'),
(5, 'Sobradinho'),
(6, 'Planaltina'),
(7, 'Paranoá'),
(8, 'Núcleo Bandeirante'),
(9, 'Ceilândia'),
(10, 'Guará'),
(11, 'Cruzeiro'),
(12, 'Samambaia'),
(13, 'Santa Maria'),
(14, 'São Sebastião'),
(15, 'Recanto das Emas'),
(16, 'Lago Sul'),
(17, 'Riacho Fundo'),
(18, 'Lago Norte'),
(19, 'Candangolândia'),
(20, 'Águas Claras'),
(21, 'Riacho Fundo II'),
(22, 'Sudoeste/Octogonal'),
(23, 'Varjão'),
(24, 'Park Way'),
(25, 'SCIA (Setor Complementar de Indústria e Abastecimento)/Estrutural'),
(26, 'Sobradinho II'),
(27, 'Jardim Botânico'),
(28, 'Itapoã'),
(29, 'SIA (Setor de Indústria e Abastecimento)'),
(30, 'Vicente Pires'),
(31, 'Fercal');

```
INSERT INTO tipo_animal (id, especie) VALUES (1, 'Cachorro'),(2,'Gato');
```

```
INSERT INTO status_animal (id, nome_status, descricao_status) VALUES (1, 'Disponível', 'Animal Disponível'), (2, 'Indisponível', 'Animal Indisponível');
```

5.2. Modelo físico do banco de dados (Sprint 2)

```
-- Criação do banco de dados
```

```
CREATE DATABASE db_acalanto;
```

```
-- Seleciona o banco de dados para uso
```

```
USE db_acalanto;
```

```
-- Criação da tabela administrador
```

```
CREATE TABLE administrador (
```

```
    id INT,
```

```
    nome_administrador VARCHAR(50),
```

```
    PRIMARY KEY (id)
```

```
);
```

```
-- Criação da tabela usuario
```

```
CREATE TABLE usuario (
```

```
    id INT,
```

```
    PRIMARY KEY (id)
```

```
);
```

```
CREATE TABLE formulario (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    nome VARCHAR(100) NOT NULL,
```

```
    rg VARCHAR(15),
```

```
    orgao_expedidor VARCHAR(50),
```

```
    cpf VARCHAR(14) UNIQUE,
```

```
    endereco VARCHAR(200),
```

```
    cidade VARCHAR(100),
```

```
    profissao VARCHAR(100),
```

```
    whatsapp VARCHAR(15),
```

```
    email VARCHAR(100),      -- E-mail
```

```
    possui_outros_animais ENUM('Sim', 'Não'),
```

```
    quantidade_outros_animais INT,
```

```
    sao_castrados ENUM('Sim', 'Não')
```

```
);

-- Criação da tabela tipo_animal
CREATE TABLE tipo_animal (
    id INT,
    especie VARCHAR(45)
);

-- Criação da tabela status_gato
CREATE TABLE status_animal (
    id INT,
    nome_status VARCHAR(50),
    descricao_status VARCHAR(255),
    PRIMARY KEY (id)
);

-- Criação da tabela tipo_pelo_animal
CREATE TABLE tipo_pelo_animal (
    id INT,
    tipo_pelo VARCHAR(45),
    PRIMARY KEY (id)
);

-- Criação da tabela cor_olhos_animal
CREATE TABLE cor_olhos_animal (
    id INT,
    nome_cor_olhos VARCHAR(45),
    PRIMARY KEY (id)
);

-- Criação da tabela regiao_administrativa_animal
CREATE TABLE regiao_administrativa_animal (
    id INT,
    nome_regiao_administrativa VARCHAR(45),
    PRIMARY KEY (id)
);

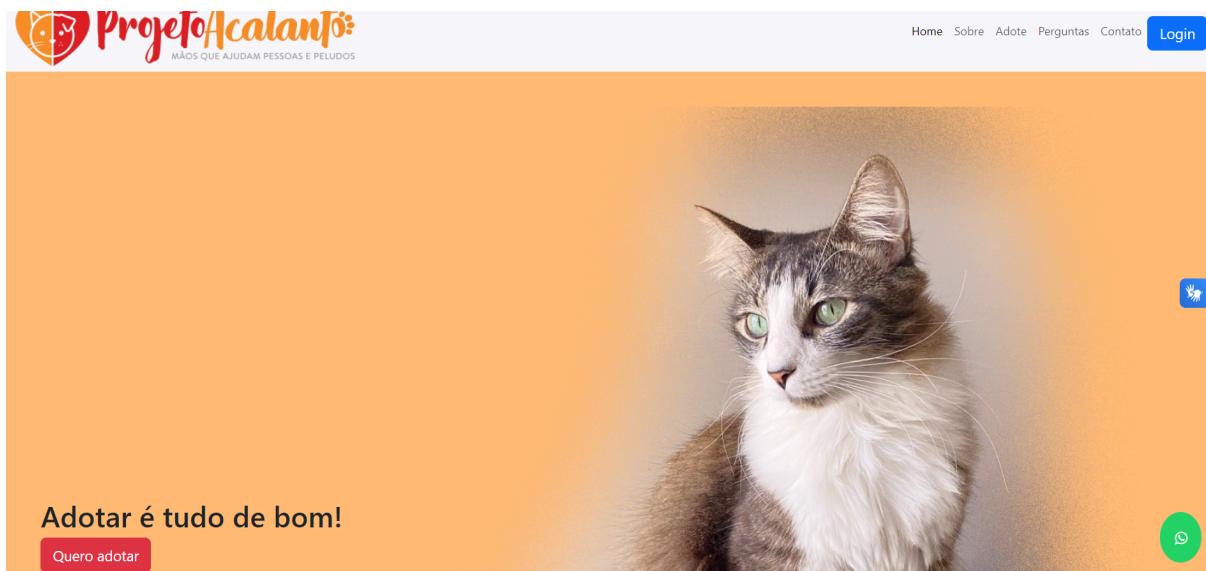
-- Criação da tabela animal
CREATE TABLE animal (
    id INT,
    nome VARCHAR(50),
```

```

dt_nascimento DATE,
genero VARCHAR(20),
observacao VARCHAR(255),
id_administrador INT,
id_status INT,
id_tipo_pelo INT,
id_regiao_administrativa_animal INT,
PRIMARY KEY (id),
FOREIGN KEY (id_administrador) REFERENCES administrador(id),
FOREIGN KEY (id_status) REFERENCES status_animal(id),
FOREIGN KEY (id_tipo_pelo_animal) REFERENCES tipo_pelo(id),
    FOREIGN KEY (id_regiao_administrativa_animal) REFERENCES
regiao_administrativa_animal(id)
);

```

6. Front-end (Sprint 1 e 2)



Tela 1 - Seção 1 “Home”

Conheça a ONG Acalanto

Existem muitos **gatos a espera de um lar** e com muito amor para oferecer.
Nós fazemos essa ponte para que você adote um amigo e faça a
diferença na vida de um animal carente.



[Saiba mais](#)



Tela 2 - Seção “Sobre”

Gatos para adoção

Cidade: Gênero: [Buscar](#)



Milk
Macho / Filhote
Samambaia



Tini
Macho / Filhote
Recanto das Emas



Zion
Macho / Filhote
Plano Piloto



Bird
Macho / Filhote
Sobradinho I

[Ver Todos](#)



Tela 3 - Seção “Adote”

Perguntas Frequentes

Como posso apoiar a ONG?	▼
As adoções são gratuitas?	▼
Os gatos para adoção são castrados?	▼
O que preciso saber antes de adotar um gato?	▼
Como fazer a adaptação em casa de um gato adotado?▼	

Tela 4 - Seção “Perguntas”

Fale conosco agora mesmo

Name:

E-mail:

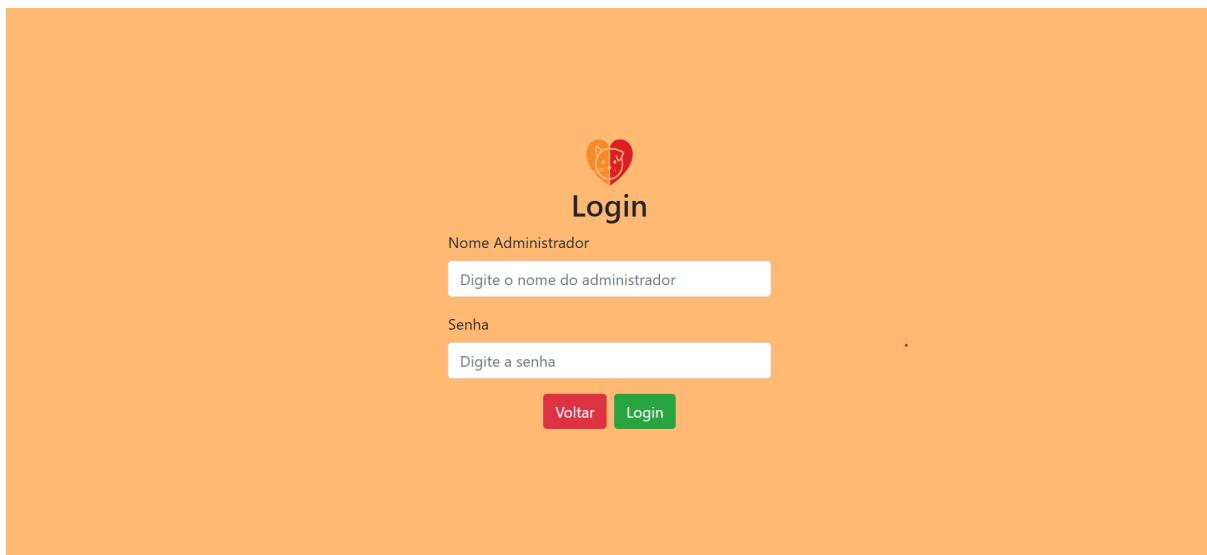
Celular:

Mensagem:



Tela 5 - Seção “Contato”



Tela 6 - Tela de Login

A screenshot of the "Gestão Gatos" (Cat Management) section of the Projeto Acalantos administrator panel. The top navigation bar includes the Projeto Acalantos logo, links for "Site" and "Sair" (Logout), and the title "Gestão Gatos". Below the title is a search bar with the placeholder "Digite para filtrar" (Type to filter), a blue "Filtrar" (Filter) button, and a green "Cadastrar Gato" (Register Cat) button. The main content area is titled "Acompanhamento de Solicitações" (Follow-up of Requests) and features a thumbnail image of a cat's face. To the right of the thumbnail, the cat's name "Milk" is listed, along with the names of the requester ("Nome Solicitante: Davi Parente") and their contact information ("Contato Solicitante: daviparente@rbe.edu.br").

Tela 7 - Painel de Administrador (Gestão Gatos)

6.1. Protótipos (Sprint 1)

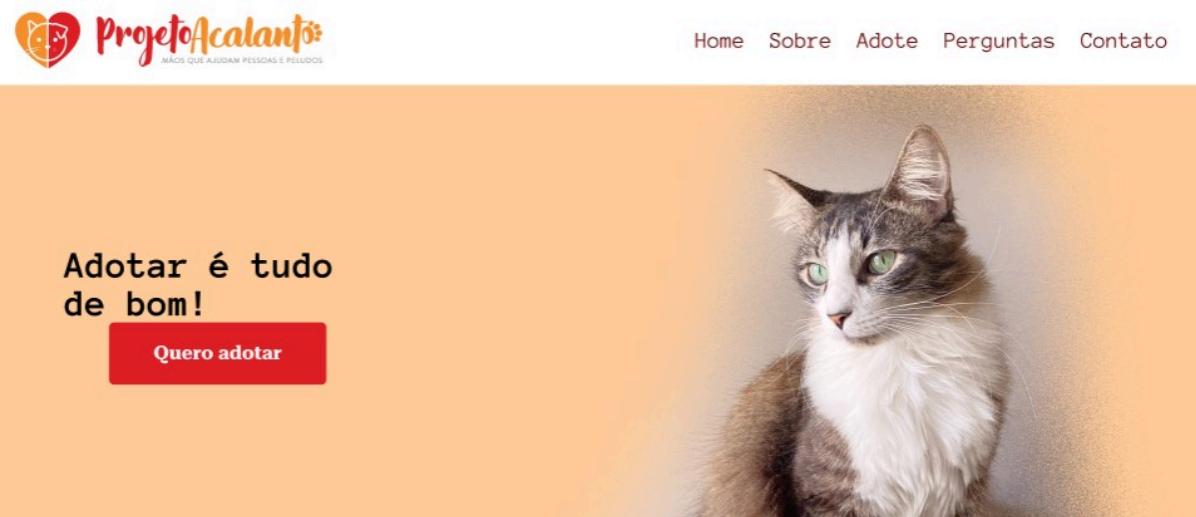


Figura 1 - A Figura 1 apresenta a seção “Home” no aplicativo de prototipagem Figma.



Figura 2 - A Figura 2 apresenta a seção “Sobre” no aplicativo de prototipagem Figma.

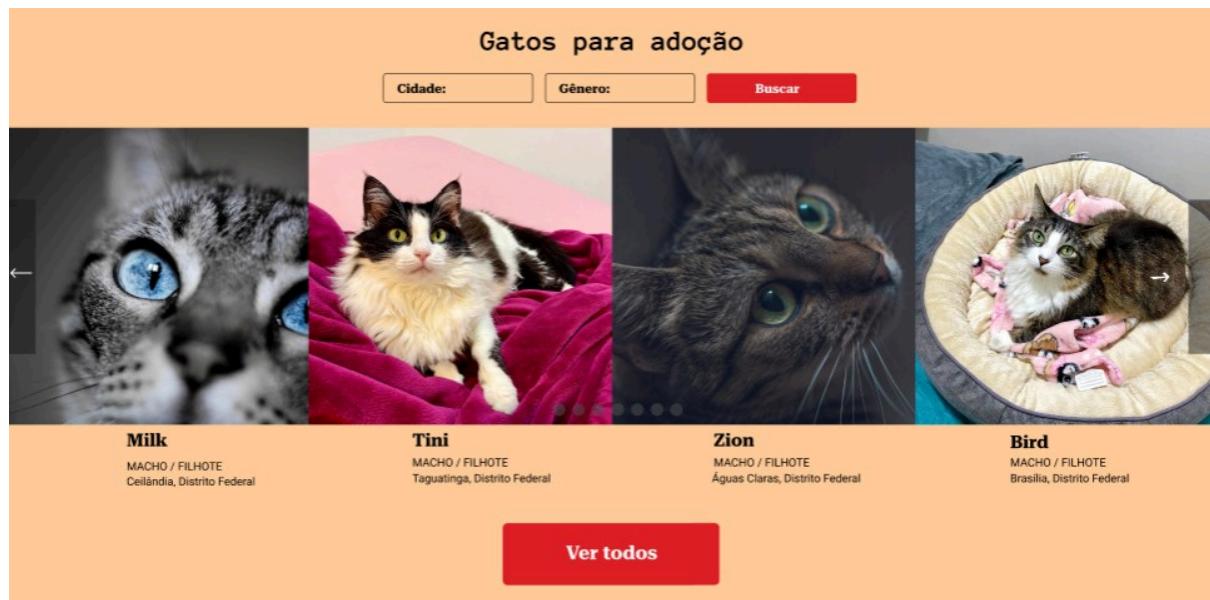


Figura 3 - A Figura 3 apresenta a seção “Adote” no aplicativo de prototipagem Figma.



Figura 4 - A Figura 4 apresenta a seção “Perguntas” no aplicativo de prototipagem Figma.

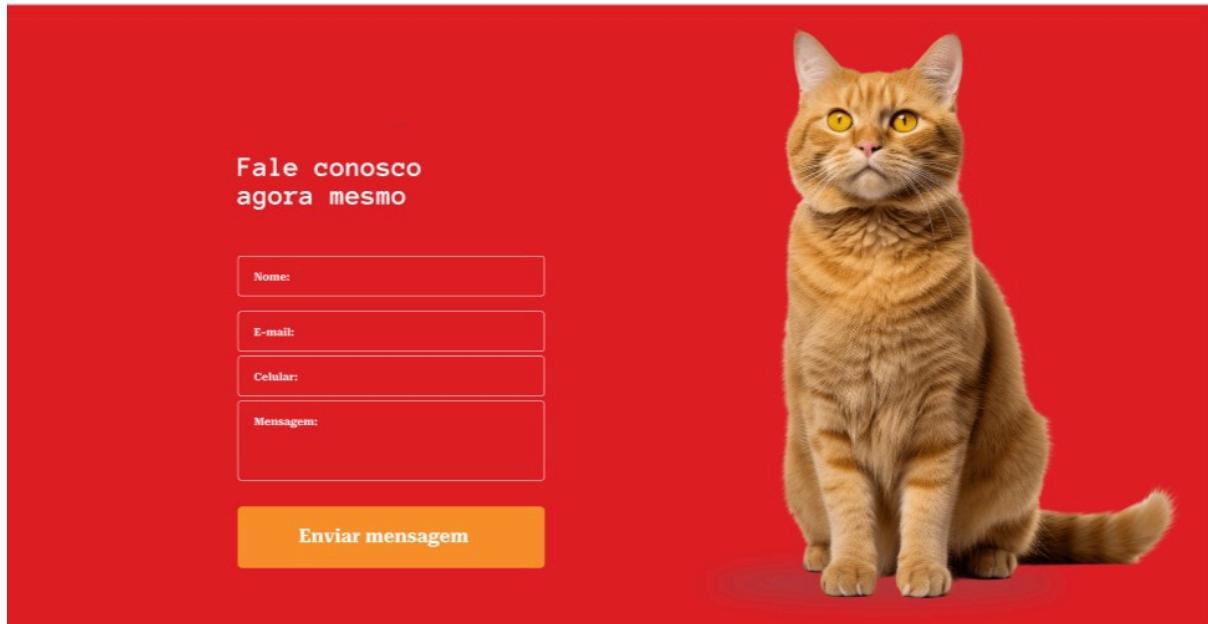


Figura 5 - A Figura 5 apresenta a seção “Contato” no aplicativo de prototipagem Figma.

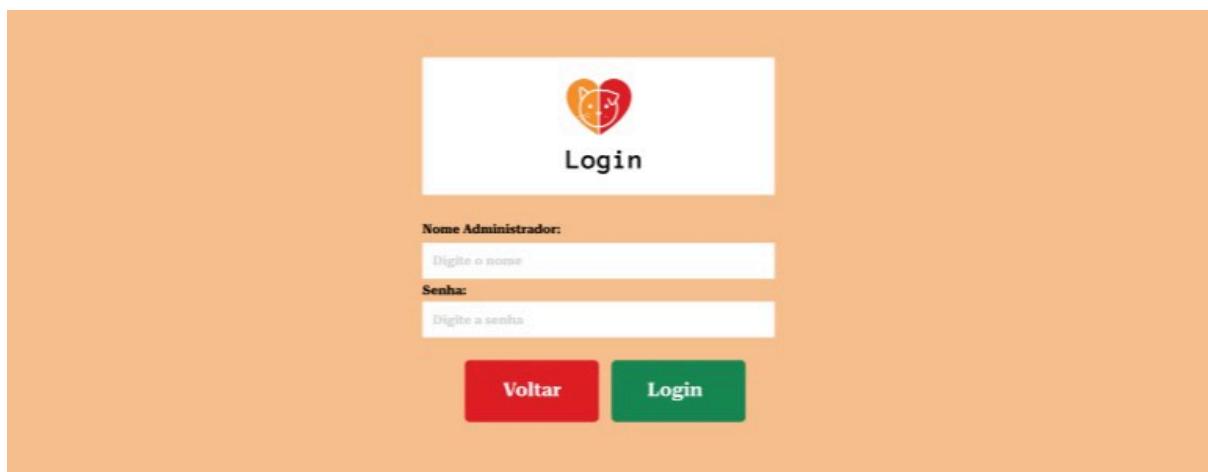


Figura 6 - A Figura 6 apresenta a tela de login do Administrador, no aplicativo de prototipagem Figma.



Figura 7 - A Figura 7 apresenta a tela de Gestão de Gatos, do perfil Administrador, no aplicativo de prototipagem Figma.

6.2. Implementação (Sprint 2)

Na implementação da aplicação “Acalanto”, foram utilizados as linguagens de marcação HTML e CSS (Cascading Style Sheets), e de programação JavaScript no FrontEnd. Na seguinte ordem, na estrutura de desenvolvimento do FrontEnd e BackEnd, foram utilizados os frameworks Bootstrap e Flask, esse último usado por meio da linguagem Python. Por fim, foi utilizado o Sistema Gerenciador de Banco de Dados MySQL, nele utilizado a linguagem de programação SQL.

7. Back-end (Sprint 2 e 3)

7.1. Diagrama de classes (Sprint 2)

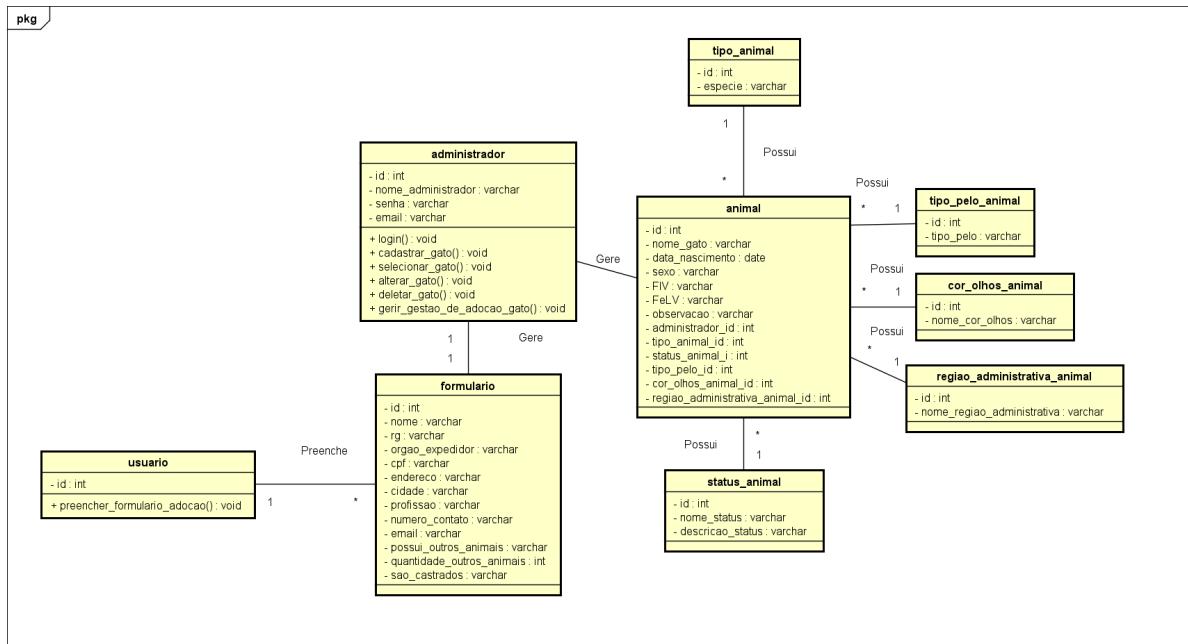


Diagrama de Classes - Projeto Acalanto

7.2. Implementação do back-end (Sprint 3)

Na implementação do Back-End foi utilizado o framework Flask da linguagem de programação Python. Segue imagens da implementação:

```

from flask import Blueprint, jsonify, request
from .models import db, Gato, SolicitudAdocao

main = Blueprint('main', __name__)

@main.route('/gatos', methods=['GET'])
def listar_gatos():
    gatos = Gato.query.all()
    return jsonify([{'id': g.id, 'nome': g.nome, 'idade': g.idade, 'genero': g.genero} for g in gatos])

@main.route('/gatos', methods=['POST'])
def criar_gato():
    data = request.json
    novo_gato = Gato(nome=data['nome'], idade=data['idade'], genero=data['genero'],
                     observacao_saude=data.get('observacao_saude'))
    db.session.add(novo_gato)
    db.session.commit()
    return jsonify({'id': novo_gato.id, 'message': 'Gato criado com sucesso!'}), 201

```

Imagen 1 - Métodos Listar Gatos (GET) e Criar Gato (POST)

```

@main.route('/gatos/<int:id>', methods=['PUT'])
def atualizar_gato(id):
    data = request.json
    gato = Gato.query.get_or_404(id)
    gato.nome = data['nome']
    gato.idade = data['idade']
    gato.genero = data['genero']
    gato.observacao_saude = data.get('observacao_saude')
    db.session.commit()
    return jsonify({'message': 'Gato atualizado com sucesso!'})

@main.route('/gatos/<int:id>', methods=['DELETE'])
def deletar_gato(id):
    gato = Gato.query.get_or_404(id)
    db.session.delete(gato)
    db.session.commit()
    return jsonify({'message': 'Gato deletado com sucesso!'})

```

Imagen 2 - Métodos Atualizar_gato (PUT) e deletar_gato(DELETE)

```

@main.route('/solicitacoes', methods=['POST'])
def criar_solicitacao():
    data = request.json
    solicitacao = SolicitacaoAdocao(nome_adotante=data['nome_adotante'], id_gato=data['id_gato'], mensagem=data.get('mensagem'))
    db.session.add(solicitacao)
    db.session.commit()
    return jsonify({'id': solicitacao.id, 'message': 'Solicitacao criada com sucesso!'}), 201

@main.route('/solicitacoes', methods=['GET'])
def listar_solicitacoes():
    solicitacoes = SolicitacaoAdocao.query.all()
    return jsonify([{'id': s.id, 'nome_adotante': s.nome_adotante, 'id_gato': s.id_gato, 'mensagem': s.mensagem} for s in solicitacoes])

```

Imagen 3 - Métodos criar_solicitacao(POST) e listar_solicitacoes(GET)

8. Integração front-end e back-end (Sprint 4)

Via Application Programming Interface (API), incluindo hospedagem do front-end e back-end.

9. Referências

- [1] Pfleeger, S. L. **Engenharia de software: teoria e prática**. Editora Pearson, 2003. E-book. ISBN: 9788587918314. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/476 epub/0>. Acesso em: 22 out. 2022.
- [2] DENNIS, Alan; WIXOM, Barbara H.; ROTH, Roberta M. **Análise e Projeto de Sistemas**. Grupo GEN, 2014. E-book. ISBN 978-85-216-2634-3. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-2634-3/>. Acesso em: 22 out. 2022.
- [3] BAZZI, Cláudio L. **Introdução a banco de dados**. Curitiba: Ed. UTFPR, 2013. e-ISBN: 978-85-7014-114-9. Disponível em: <http://proedu.rnp.br/handle/123456789/1550>. Acesso em: 03 mai. 2023.
- [4] KALBACH, James. Design de navegação web. Porto Alegre: Grupo A, 2009. E-book. ISBN 9788577805310. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577805310/>. Acesso em: 07 mar. 2024.
- [5] PORTAL INSIGHTS. Como fazer uma pesquisa estatística passo a passo? Portal Insights, 2024. Disponível em: <<https://www.portalinsights.com.br/perguntas-frequentes/como-fazer-uma-pesquisa-estatistica-passo-a-passo>>. Acesso em: 17 out. 2024.
- [6] Python para estatísticos. Disponível em: <https://tmfilho.github.io/pyestbook/intro.html>. Acesso em: abril de 2024
- [7] ROSA, Paulo Henrique C.. Desenvolvimento de Software tipo aplicativo de dispositivo móvel para auxílio em abordagem estatística na área de saúde. Dissertação de Mestrado. São Paulo: Universidade Brasil, 2020. Disponível em: https://universidadebrasil.edu.br/portal/_biblioteca/uploads/20210416143305.pdf