



SOS Bichinhos

**SOS
Bichinhos**



PUC Minas



PUC Minas

Trabalho Interdisciplinar: Back-End

Membros da equipe:

Davi Cândido de Almeida , Letícia da Silva Rocha, Matheus
Eduardo Campos Soares, Rayssa Mell de Souza Silva

Professores:

Max, Sandro, Wladimir

15 PROTEGER A
VIDA TERRESTRE



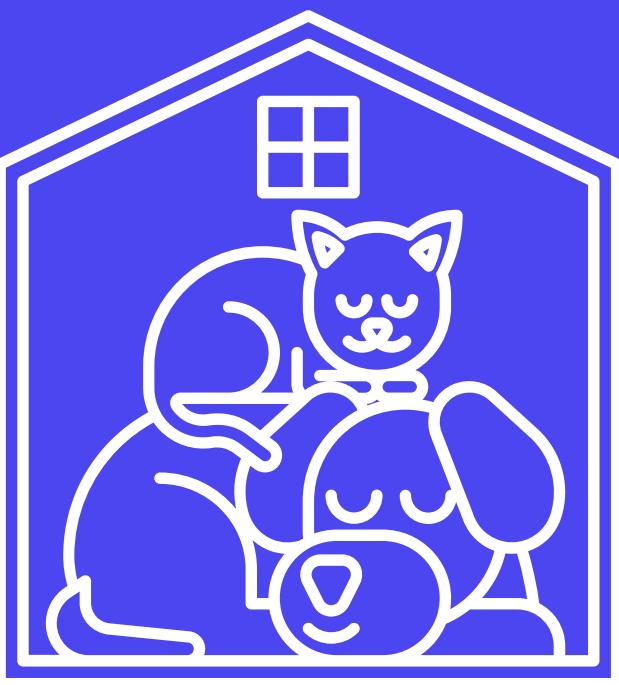


PUC Minas

Tópicos

Sprint-4

- Contexto
- Problema
- Público Alvo
- Objetivo
- Impacto
- Como o projeto se sustentará
- Requisitos
- MiniMundo
- Diagramas
- Arquitetura de Solução
- Organização das pastas
- Fluxo das Telas
- Vídeo sobre as telas
- Sobre a IA
- Intelligent System Canvas
- Vídeo sobre a IA



Conteúdo

- Crescente número de animais abandonados no Brasil.

The screenshot shows the header of the O TEMPO Cidades website. The top navigation bar includes links for ÚLTIMAS, CIDADES, ESPORTE, POLÍTICA, ECONOMIA, ENTRETENIMENTO, MINAS S/A, CANAL O TEMPO, PODCASTS, and PROMOÇÕES. Below the navigation is a dark green bar with icons for search, news, and user login. The main title 'O TEMPO | CIDADES' is displayed in large green letters. To the right are buttons for 'ASSINE' and 'ENTRAR'. The page content below the header discusses the 'Dia Nacional da Adoção' and the increasing number of abandoned animals in MG.

O TEMPO > Cidades > Artigo

DIA NACIONAL DA ADOÇÃO

Adoção muda vida de animais abandonados em MG, que soma denúncias a cada 2 horas

Número de maus-tratos aos animais subiu 3% com relação ao ano passado; voluntários tentam driblar alta de demanda

Por Isabela Abalen

Publicado em 04 de outubro de 2023 | 06:00



15 PROTEGER A VIDA TERRESTRE



Grande
parcela de
animais em
condições de
ruas

Programas
para a adoção
de animais já
existem, contudo
não são
suficientemente
divulgados

Aumento
da procura
por
animais de
estimação

Objetivo e Motivação

Aproximar possíveis adotantes de ONGs e abrigos de animais abandonados

Promover lares dignos para animais em situação de rua, longe de condições precárias

Fornecer maneiras de auxiliar a causa, além da adoção



Público Alvo

Pessoas que
buscam
adotar animais
para uma vida
mais feliz.

ONGs e
Abrigos
que apoiam
a causa

Pessoas que
resgatam e
encontram animais,
encaminhando
para instituições de
confiança.



Impacto

A computação pode ajudar a resolver esse problema ao:

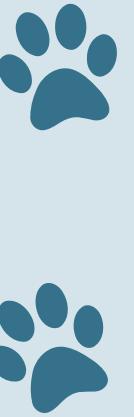
- Facilitar a conexão entre indivíduos
- Apoiar ONGs
- Superar dificuldades na adoção





Como o projeto se sustentará?





Minimundo

Uma **pessoa** encontrou um **animal** abandonado na rua e buscando uma forma pra **disponibilizá-lo para adoção** viu que uma ONG exigia o **envio de algumas informações** como a **situação do animal, sua localização, seu tipo e uma foto do animal para coleta de dados** que será feito através **de um sistema inteligente** capaz de analisar o animal e retornar informações necessárias a ONG.

Após o recebimento do animal informado, a ONG **cadastra** sua **especie, raça, sexo, nome, idade, porte, historia, vacinas e suas tags (características)** tornando-o **disponível para a adoção**. Para que uma **pessoa** possa **adotar**, é necessário possuir um **cadastro** que exige **nome, email, senha, moradia, endereço e uma imagem**. Além disso, também deve **preencher um formulário** contendo: **nome, idade, sexo, cidade, email, telefone para contato, seu tipo de moradia**, e caso for em apartamento **se é liberado ter animais, se a família estará ciente da adoção do animal, se a pessoa já teve algum animal, se o animal ficará sozinho, e caso precisar onde ou com quem ele poderia ficar**, e permitir uma visita da ONG para verificar se as informações concedidas são verdadeiras.

As pessoas também poderão **comentar** sobre os **animais disponibilizados** e caso deseje poderão contribuir com seu apoio a ONG.

Legenda

- Entidade
- Atributos
- Funcionalidades

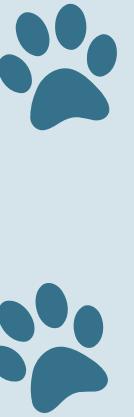
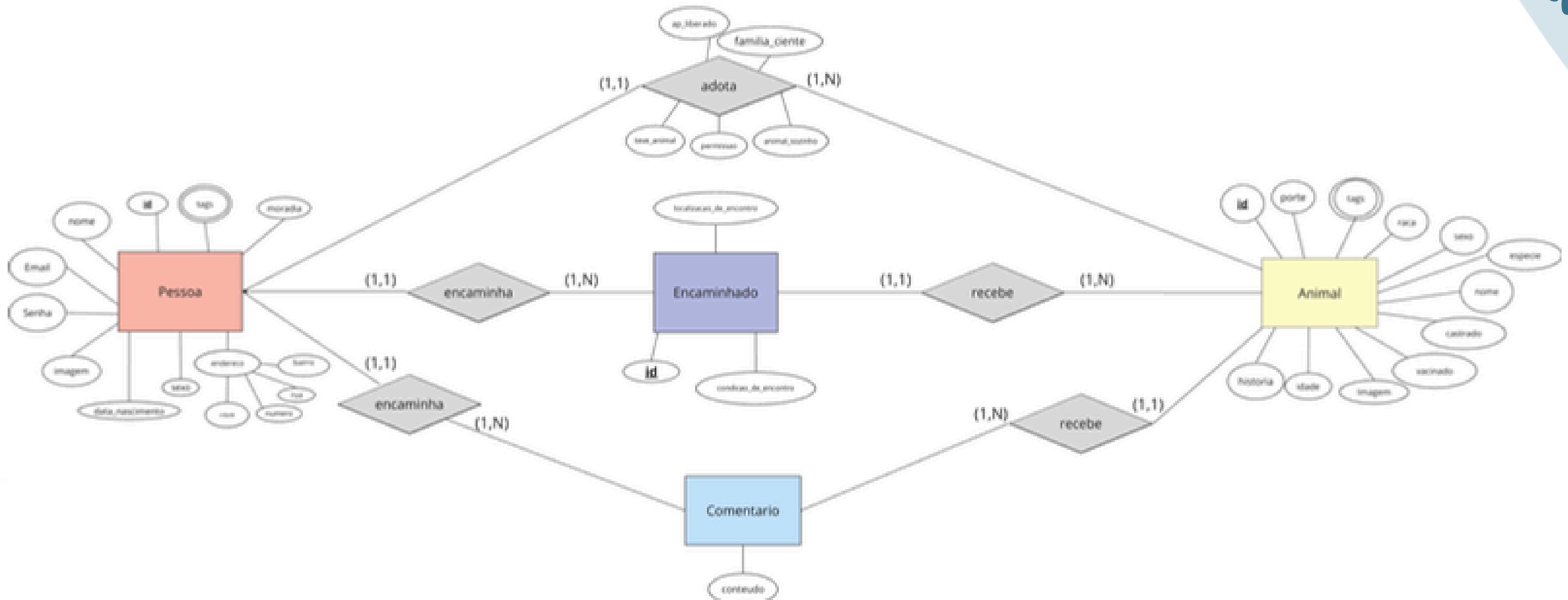
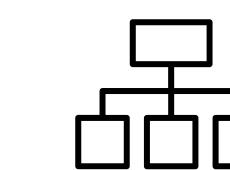


Diagrama DER

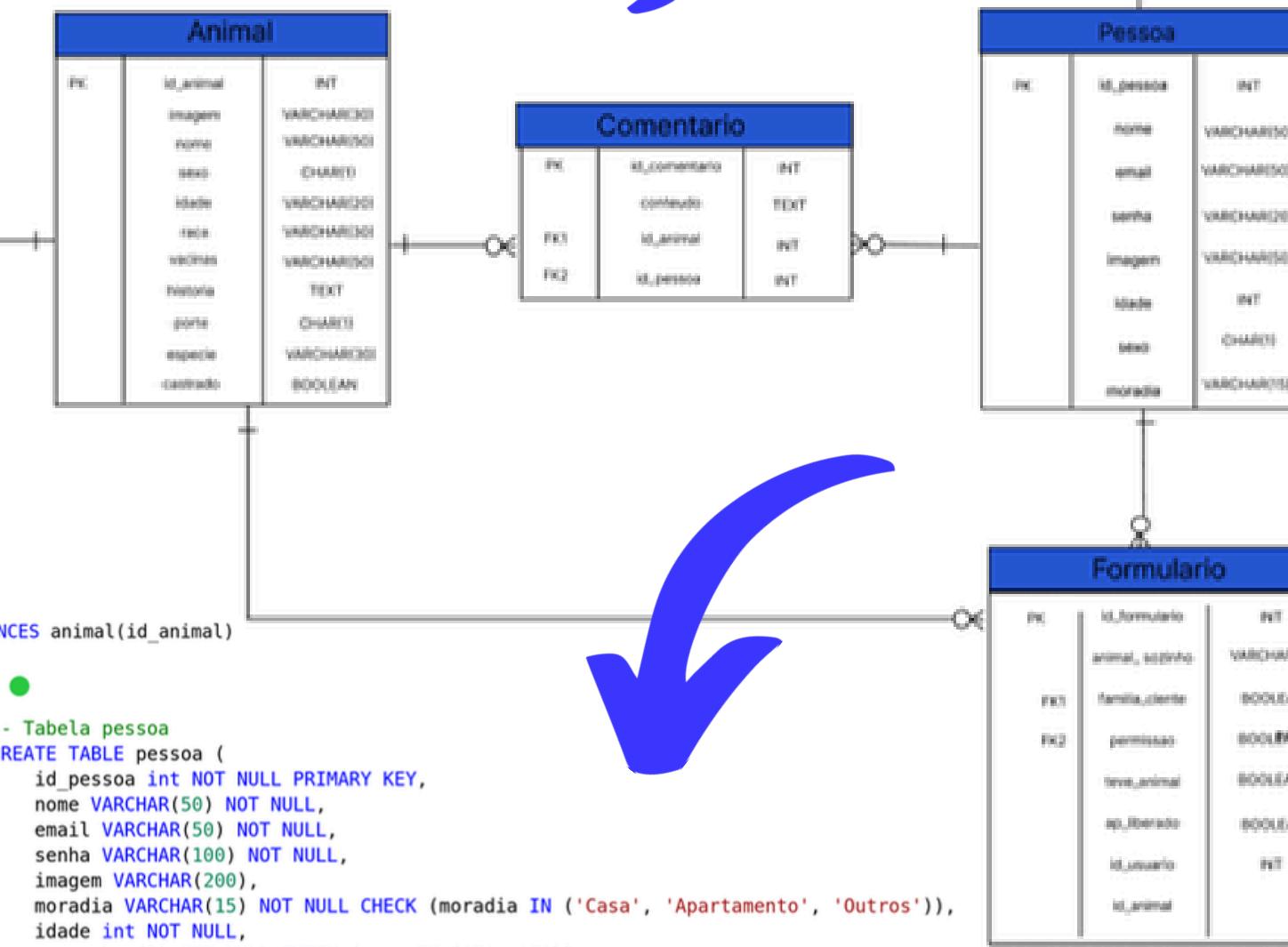


DER Estendido em Notação Pé de Galinha e scripts SQL



```

    ● ● ●
1 -- Tabela Animal
2 CREATE TABLE animal (
3   id_animal int NOT NULL PRIMARY KEY ,
4   nome VARCHAR(50) NOT NULL,
5   sexo char (1) NOT NULL CHECK (sexo IN ('M', 'F')),
6   idade VARCHAR(20) NOT NULL,
7   castrado BOOLEAN NOT NULL,
8   raca VARCHAR(30) NOT NULL,
9   vacinas VARCHAR(50) NOT NULL,
10  historia text,
11  porte CHAR(1) NOT NULL CHECK (porte IN ('P', 'M', 'G')),
12  especie VARCHAR(30) NOT NULL
13 );
14 );
    
```



```

    ● ● ●
1 -- Tabela tagsAnimal
2 CREATE TABLE tagsAnimal (
3   id_tagAnimal int NOT NULL PRIMARY KEY ,
4   conteudo_tag VARCHAR(20),
5   id_animal int NOT NULL,
6   CONSTRAINT FK_tagsAnimal_animal FOREIGN KEY (id_animal) REFERENCES animal(id_animal)
7 );
11 );
    
```

```

    ● ● ●
1 -- Tabela pessoa
2 CREATE TABLE pessoa (
3   id_pessoa int NOT NULL PRIMARY KEY ,
4   nome VARCHAR(50) NOT NULL,
5   email VARCHAR(50) NOT NULL,
6   senha VARCHAR(100) NOT NULL,
7   imagem VARCHAR(200),
8   moradia VARCHAR(15) NOT NULL CHECK (moradia IN ('Casa', 'Apartamento', 'Outros')),
9   idade int NOT NULL,
10  sexo char(1) NOT NULL CHECK (sexo IN ('M', 'F'))
11 );
    
```

Enviado		
Frame 1	id_enviado	INT
FK	id_pessoa	INT
	imagem	VARCHAR(200)
	sexo	CHAR(1)
	especie	VARCHAR(300)
	raça	VARCHAR(300)
	idade	VARCHAR(300)
	porte	VARCHAR(300)
	temperamento	VARCHAR(300)
	necessidades_especiais	VARCHAR(300)
	estado_de_saude	VARCHAR(300)
	caracteristicas_gerais	VARCHAR(300)

```

    ● ● ●
1 -- Tabela enviado
2 CREATE TABLE enviado (
3   id_enviado int NOT NULL PRIMARY KEY ,
4   imagem VARCHAR(200) NOT NULL,
5   sexo char (1) NOT NULL CHECK (sexo IN ('M', 'F')),
6   especie VARCHAR(300) NOT NULL,
7   raça VARCHAR(300) NOT NULL,
8   idade VARCHAR(300) NOT NULL,
9   porte VARCHAR(300) NOT NULL,
10  temperamento VARCHAR(300) NOT NULL,
11  necessidades_especiais VARCHAR(300) NOT NULL,
12  estado_de_saude VARCHAR(300) NOT NULL,
13  carateristicas_gerais VARCHAR(300) NOT NULL,
14  id_pessoa int NOT NULL ,
15  CONSTRAINT FK_enviado_pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
16 );
17 );
    
```

```

    ● ● ●
1 CREATE TABLE endereco (
2   id_endereco int NOT NULL PRIMARY KEY ,
3   bairro VARCHAR(50) NOT NULL,
4   rua VARCHAR(50) NOT NULL,
5   numero VARCHAR(20) NOT NULL,
6   cidade VARCHAR(40) NOT NULL,
7   estado VARCHAR(20) NOT NULL,
8   id_pessoa int NOT NULL ,
9   CONSTRAINT FK_endereco_pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
10 );
11 );
12 );
    
```

Endereco		
FK	id_endereco	INT
	bairro	VARCHAR(50)
	rua	VARCHAR(50)
	numero	VARCHAR(20)
	cidade	VARCHAR(40)
	estado	VARCHAR(20)
	id_pessoa	INT

```

    ● ● ●
1 -- Tabela formulario
2 CREATE TABLE formulario (
3   id_formulario int NOT NULL PRIMARY KEY ,
4   animal_sozinho VARCHAR(30),
5   familia_cliente BOOLEAN,
6   permissao BOOLEAN NOT NULL,
7   teve_animal BOOLEAN NOT NULL,
8   id_animal int NOT NULL,
9   CONSTRAINT FK_formulario_Animal FOREIGN KEY (id_animal) REFERENCES animal(id_animal),
10  id_pessoa int NOT NULL ,
11  CONSTRAINT FK_formulario_Pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
12 );
13 );
14 );
    
```

Modelo Lógico do Banco de Dados: Diagrama de Esquema (DSC)

PESSOA

<u>Id</u>	Nome	Senha	Email	Moradia	Imagen	Idade	Sexo
-----------	------	-------	-------	---------	--------	-------	------

ANIMAL

<u>Id</u>	Nome	Raca	Vacinas	Imagen	Idade	Sexo	Historia	Castrado	Porte	Especie
-----------	------	------	---------	--------	-------	------	----------	----------	-------	---------

TAGS ANIMAL

Animal	<u>Id</u>	Conteudo
--------	-----------	----------

ENCAMINHAMENTO

Pessoa	Condicao_de_encontro	Localizacao_de_encontro
--------	----------------------	-------------------------

FORMULARIO

Pessoa	Animal	<u>Id</u>	Ap_liberado	Familia_ciente	Teve_animal	Permissao	Animal_sozinho
--------	--------	-----------	-------------	----------------	-------------	-----------	----------------

COMENTARIO

<u>Pessoa</u>	<u>Animal</u>	<u>Id</u>	Conteudo
---------------	---------------	-----------	----------

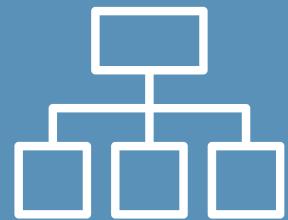
ENDERECO

Pessoa	<u>Id</u>	Bairro	Rua	Numero	Cidade
--------	-----------	--------	-----	--------	--------

ENVIADO

<u>Id_enviado</u>	Pessoa	Imagen	Sexo	Especie	Raca	Idade	Porte	Temperamento	Necessidades_Especiais	Estado_de_Saude	Caracteristicas_Gerais
-------------------	--------	--------	------	---------	------	-------	-------	--------------	------------------------	-----------------	------------------------

Diagrama



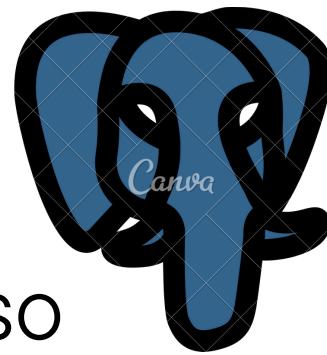
Arquitetura da Solução

< Front-End >

Para a estruturação do Front-End:

- **JavaScript** tornou o site interativo
- **CSS** garantiu um design atraente
- **HTML** estruturou o conteúdo.

Além disso contamos com o uso da plataforma replit para conectarmos nosso site.



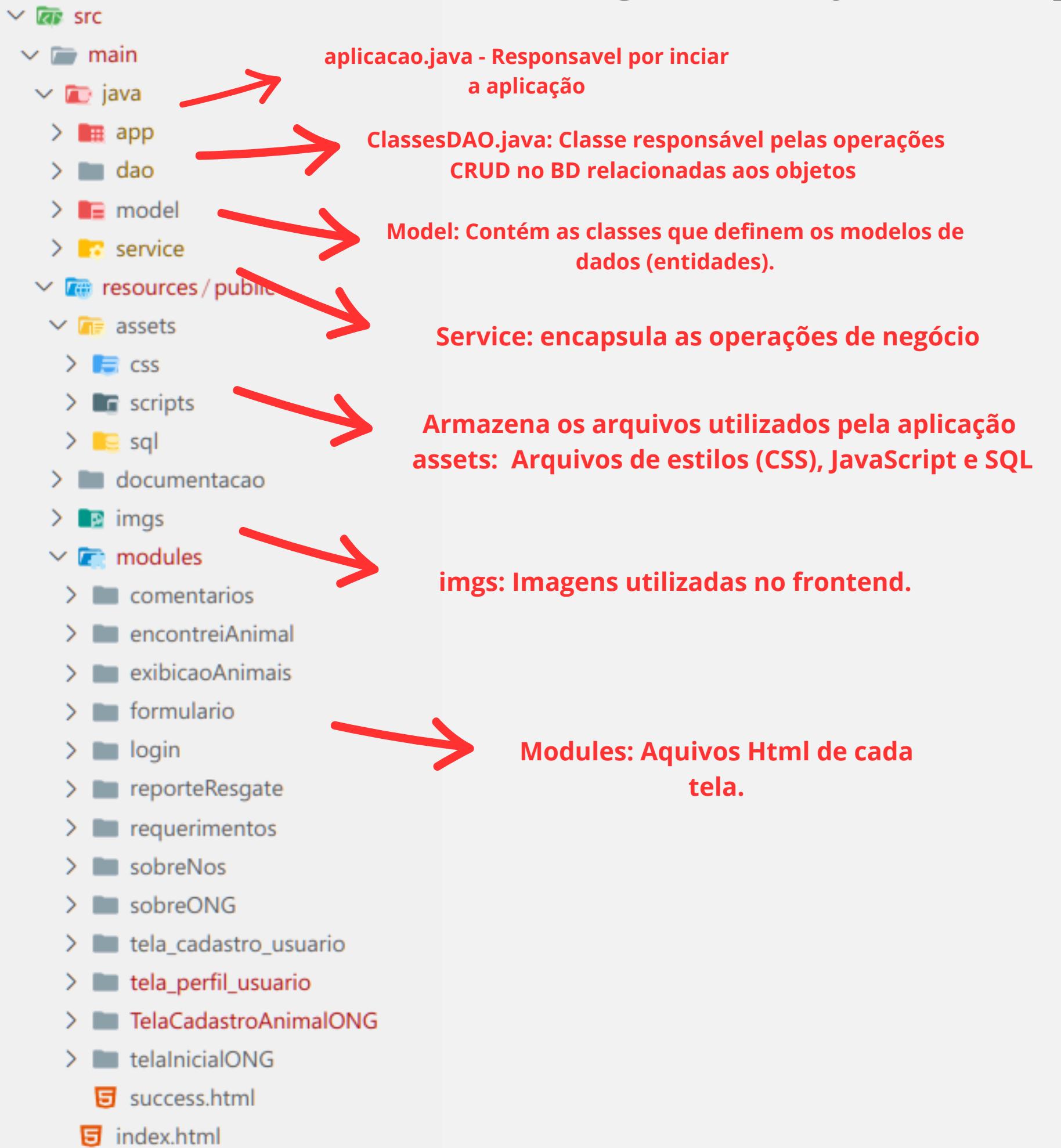
</ Back-End >

E para a estruturação do Back-End:

- **Java** (Comunicação com o banco de dados)
- **PostgresSQL** (Armazenamento do Banco de dados de forma [online AZURE](#))
- **Spark - Framework** (Criar API RESTful para consultas ao banco de dados)
- **Maven** (Gerenciar as dependências do projeto)
- **Analise de Imagem - [IA Gemine \(S.I.\)](#)**
 - Analise inteligente de imagens para agilizar o cadastro de animais



Organização de pastas



aplicacao.java - Responsavel por iniciar a aplicação

ClassesDAO.java: Classe responsável pelas operações CRUD no BD relacionadas aos objetos

Model: Contém as classes que definem os modelos de dados (entidades).

Service: encapsula as operações de negócio

Armazena os arquivos utilizados pela aplicação
assets: Arquivos de estilos (CSS), JavaScript e SQL

imgs: Imagens utilizadas no frontend.

Modules: Arquivos Html de cada tela.

Requisitos

• Requisitos Funcionais

O sistema deve permitir que o usuário faça o cadastro do animal.

O sistema deve permitir que o usuário faça seu cadastro pessoal.

O sistema deve exibir quais animais estão disponíveis para adoção.

O sistema deve permitir que o usuário envie uma imagem de um animal encontrado.

O sistema deve permitir que o adotante preencha a documentação para a adoção.

O sistema deve permitir que a ONG visualize os requisitos de adoção.

O sistema deve permitir que o usuário trafegue entre as telas.

O sistema deve permitir que a ONG visualize os animais encaminhados.

O sistema deve permitir que os usuários comentem sobre os animais .

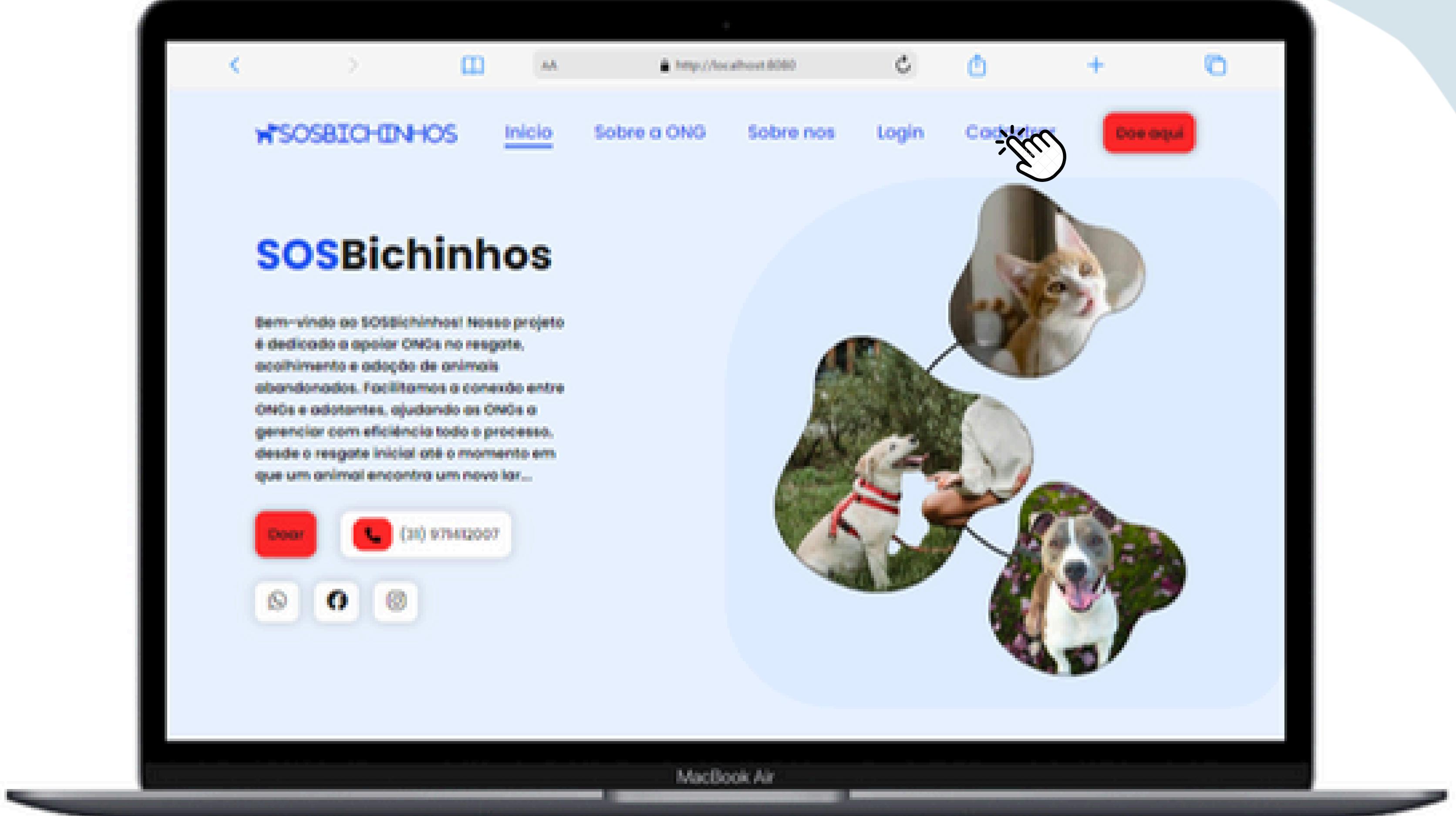
• Requisitos Não Funcionais

O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, Java e JavaScript e o sistema gerenciador de banco de dados PostgreSQL.

O sistema deve ter alta disponibilidade e ser capaz de lidar com um grande volume de acessos simultâneos.

O sistema deve ser intuitivo e fácil de usar, mesmo para usuários com pouca experiência em tecnologia, ou deficientes. Além de ter uma boa performance, mesmo em conexões de internet lentas.

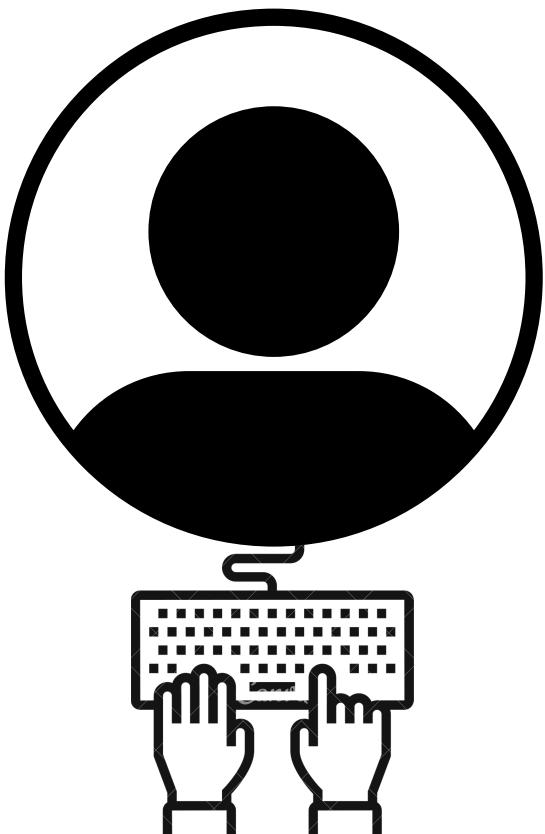
O sistema deve fornecer suporte técnico eficiente para os usuários em caso de problemas ou dúvidas.



MacBook Air

Levando dados para o Back-End

Cadastro de pessoa



A screenshot of a web browser displaying a 'Cadastro' (Registration) form. The form contains the following fields:

- Nome: Usuarionovo
- Moradia: Casa
- Bairro: Coração Eucaristico
- Rua: Rua Coração Eucaristico de J
- Número: 123
- Cidade: Belo Horizonte
- Estado: MG
- Idade: 20
- Sexo: Masculino
- E-mail: UsuarioNovo@gmail.com
- Senha:
- Confirme:

At the bottom are three buttons: 'Cadastrar' (with a hand cursor), 'Limpar', and 'Já tem uma conta?'

● ● ●

```
1 async function HG_createUserAndEndereco(user, endereco) {
2   try {
3     // Criação do usuário
4     console.log(user);
5     const userResponse = await fetch('/usuario', {
6       method: 'POST',
7       headers: {
8         'Content-Type': 'application/json',
9       },
10      body: JSON.stringify(user)
11    });
12
13    if (!userResponse.ok) {
14      throw new Error('Erro ao criar o usuário');
15    }
16  }
```

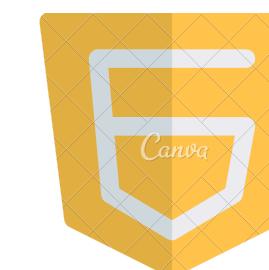
HTML

```
<form id="form-contato">
  <!-- Campos existentes -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Moradia -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Campos de endereço -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Idade -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Sexo -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Campo E-mail -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Senha, confirmar senha, e botões -->
  ><div class="form-group row align-items-center mb-3">...</div> flex
  ><div class="form-group row align-items-center mb-3">...</div> flex
  ><div class="form-group row align-items-center">...</div> flex
</form>
```



JAVASCRIPT

```
1 const createdUser = await userResponse.json();
2 console.log(createdUser);
3 LCdisplayMessage("Sucesso ao criar usuário");
4
5 // Atribuir o ID do usuário recém-criado ao endereço
6 endereco.id_pessoa = createdUser.id;
7
8 // Criação do endereço
9 console.log(endereco);
10 const enderecoResponse = await fetch('/endereco', {
11   method: 'POST',
12   headers: {
13     'Content-Type': 'application/json',
14   },
15   body: JSON.stringify(endereco)
16});
```





Recebendo dados no Back-End



Aplicação

```
● ● ●  
1 // -----Aplicações CRUD Pessoa ----- //  
2  
3     post("/usuario", (request, response) -> aplicacaousuario.add(request, response));  
4  
5     get("/usuario/:id", (request, response) -> aplicacaousuario.get(request, response));  
6  
7     post("/usuario/update/:id", (request, response) -> aplicacaousuario.update(request, response));  
8  
9     post("/usuario/delete/:id", (request, response) -> aplicacaousuario.remove(request, response));  
10  
11    get("/usuario", (request, response) -> aplicacaousuario.getAll(request, response));  
12  
13  
14 // -----Aplicações CRUD Endereço ----- //  
15  
16  
17     post("/endereco", (request, response) -> aplicacaoEndereco.add(request, response));  
18  
19     get("/endereco/:id", (request, response) -> aplicacaoEndereco.get(request, response));  
20  
21     post("/endereco/update/:id", (request, response) -> aplicacaoEndereco.update(request, response));  
22  
23     post("/endereco/delete/:id", (request, response) -> aplicacaoEndereco.remove(request, response));  
24  
25     get("/endereco", (request, response) -> aplicacaoEndereco.getAll(request, response));  
26
```

PessoaService

```
● ● ●  
1 public Object add(Request request, Response response) {  
2     Gson gson = new Gson();  
3     Pessoa registro = gson.fromJson(request.body(), Pessoa.class);  
4  
5     int id = this.pessoDAO.getMaxId_pessoa() + 1; // Corrigido o getMaxId  
6     registro.setId(id);  
7     System.out.println(registro);  
8  
9     pessoaDAO.inserirPessoa(registro);  
10  
11    response.status(201); // 201 Created  
12    return id;  
13 }
```

EnderecoService

```
● ● ●  
1 public Object add(Request request, Response response) {  
2     Gson gson = new Gson();  
3     Endereco registro = gson.fromJson(request.body(), Endereco.class); // Altere para Endereco  
4  
5     // Gera o próximo id para o Endereco  
6     int id = this.enderecoDAO.getMaxId_endereco() + 1; // Altere para getMaxIdEndereco()  
7     int id_pessoa = this.pessoDAO.getMaxId_pessoa();  
8  
9     registro.setId_pessoa(id_pessoa);  
10    registro.setId_endereco(id); // Altere para setId_endereco  
11  
12    System.out.println(registro);  
13  
14    enderecoDAO.inserirEndereco(registro); // Altere para inserirEndereco  
15  
16    response.status(201); // 201 Created  
17    return id;  
18 }
```

Back-End -> PessoaDao



Perfil

Nome: UsuarioNovo

E-mail: Informe o e-mail

Escolher arquivo

Idade: 20

Sexo: Masculino

Moradia: Selecionar opção

Endereço

Bairro: Coração Eucarístico

Rua: Rua Coração Eucarístico de Jesus

Número: 123

Cidade: Belo Horizonte

Estado: MG

Senha: Informe sua senha

Confirme:

Alterar Sair

Deletar

PreparedStatement Proteção a SQL Injection

```
1 public boolean inserirpessoa(Pessoa pessoa) {  
2     boolean status = false;  
3     String sql = "INSERT INTO pessoa (id_pessoa, nome, email, senha, moradia, imagem, idade, sexo) VALUES (?, ?, ?, ?, ?, ?, ?, ?);"  
4  
5     try (PreparedStatement ps = conexao.prepareStatement(sql)) {  
6         ps.setInt(1, pessoa.getId());  
7         ps.setString(2, pessoa.getNome());  
8         ps.setString(3, pessoa.getEmail());  
9         ps.setString(4, CriptografiaAES.criptografar(pessoa.getSenha()));  
10        ps.setString(5, pessoa.getMoradia());  
11        ps.setString(6, pessoa.getImagen());  
12        ps.setInt(7, pessoa.getIdade());  
13        ps.setString(8, pessoa.getSexo());  
14  
15        ps.executeUpdate();  
16        status = true;  
17    } catch (Exception e) {  
18        e.printStackTrace();  
19    }  
20    return status;  
21 }
```



CriptografiaAES

	id_pessoa	nome	email	senha	imagem	moradia	idade	sexo
	integer	character varying	character varying	character varying	character varying	character varying	integer	character
1	1	ONG	SOSBichinhosdiw@gmail.com	RJ110kp362%UrZ5pxfFb+w==	null	Casa	19	F
2	2	teste	teste@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Casa	19	F
3	4	maria	mari@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Casa	19	F
4	5	evre	tata@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Apartamento	18	F
5	7	Matheus Eduardo Campos Soares	matheus@gmail.com	DGYYLscQhABl66pHxebOkA==	https://ibb.co/DMcS2Fn/Omn-Forjador-De-Galaxias.jpg	Casa	20	M
6	6	mel	mel@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Casa	19	F
7	8	Rayssa	rmei@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Apartamento	19	F
8	9	Davi	davi@gmail.com	NjhGA9sIBMb3SL2QatE9Hg==	null	Apartamento	27	M
9	3	Leticia	leticia@gmail.com	DGYYLscQhABl66pHxebOkA==	null	Casa	19	F
10	10	UsuarioNovo	UsuarioNovo@gmail.com	PwV17PYPDavdLSqb3FQBA==	null	Casa	20	M

Vídeo do CRUD

ENDEREÇO: AV. TRINTA E UM DE MARÇO, 1020 -
DOM CABRAL, BELO HORIZONTE - MG, 30535-000

SOS Bichinhos

MAIS INFORMAÇÕES

HISTÓRIA

A ONG SOS Bichinhos foi fundada em 2024, com o objetivo de ajudar os animais abandonados e desaparecidos.

LOCAL

O mapa mostra a localização da ONG SOS Bichinhos no cruzamento da Rua Guatambu com a Rua Manoel Pinto, próximo ao Museu de Ciências Naturais PUC Minas. Outros pontos marcados incluem o PUC Minas Virtual e o 299 Speed Shop.

PUC Minas Virtual
Ver mapa ampliado

Museu de Ciências Naturais PUC Minas

R. Guatambu

PUC Minas Virtual

299 Speed Shop

R. Manoel Pinto

Gameleira

Atalhos do teclado Dados cartográficos ©2024 Termos Informar erro no mapa

Conexão entre ONGs e Adotantes!

@ SOS Bichinhos

Instagram Facebook Twitter

Sobre a Inteligência do Sistema



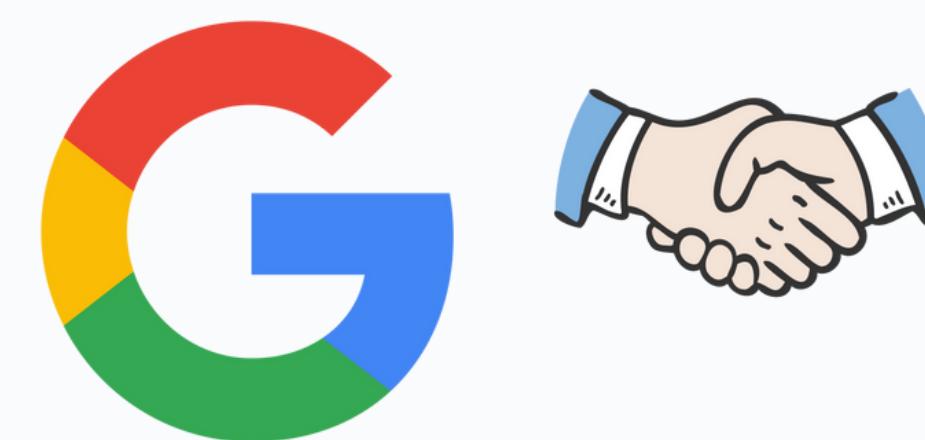
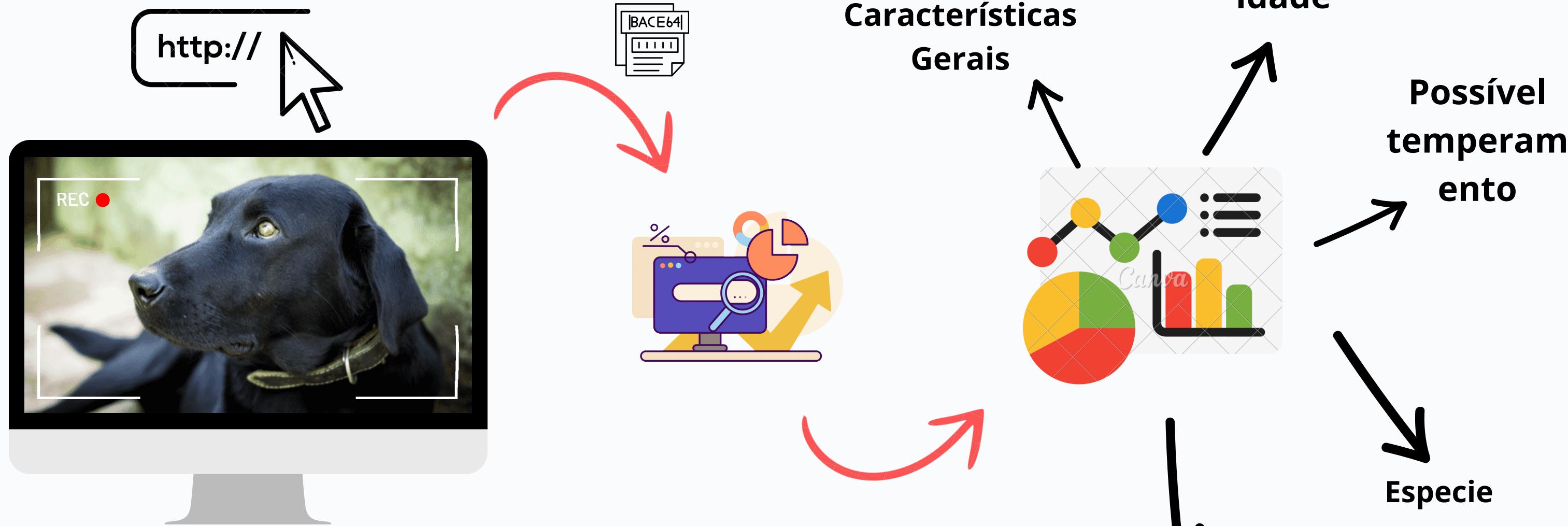
Gemini vision

[Acesse aqui](#)

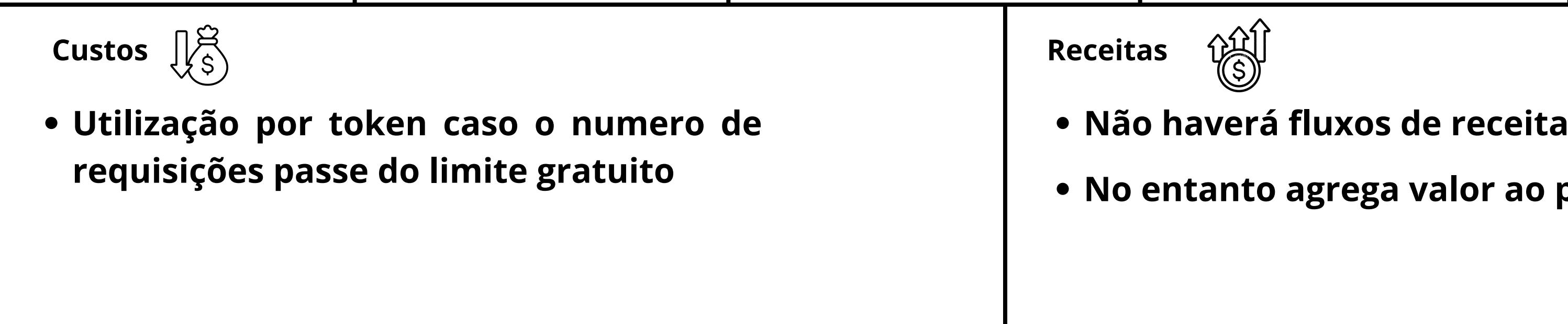
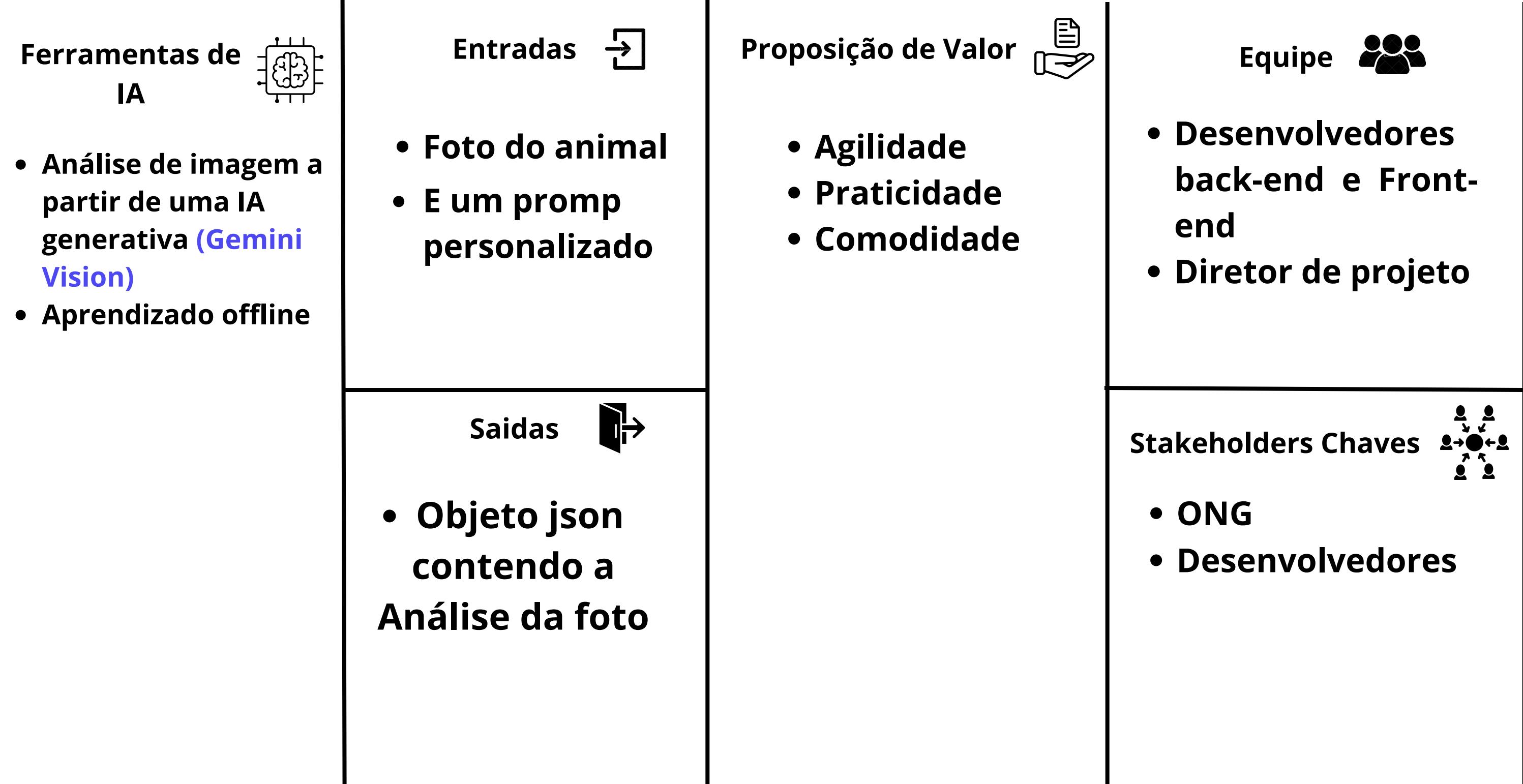


```
private static final String endpoint = "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-pro-latest:generateContent?key="; // Endpoint da API da Gemini
```

Recursos, técnicas e fornecedor

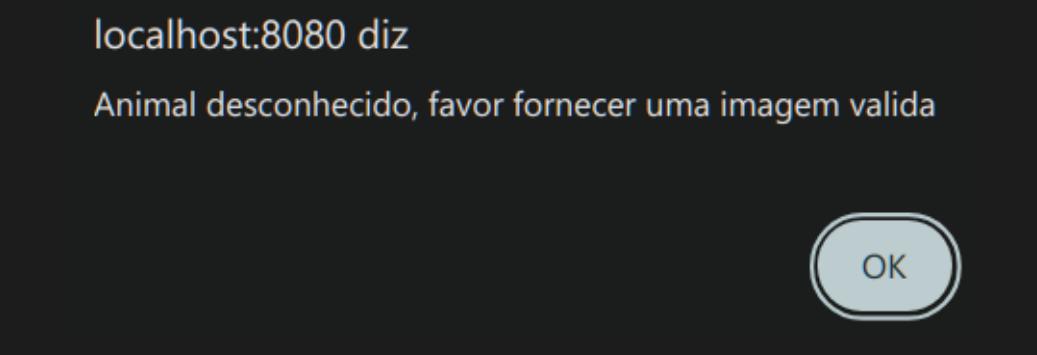
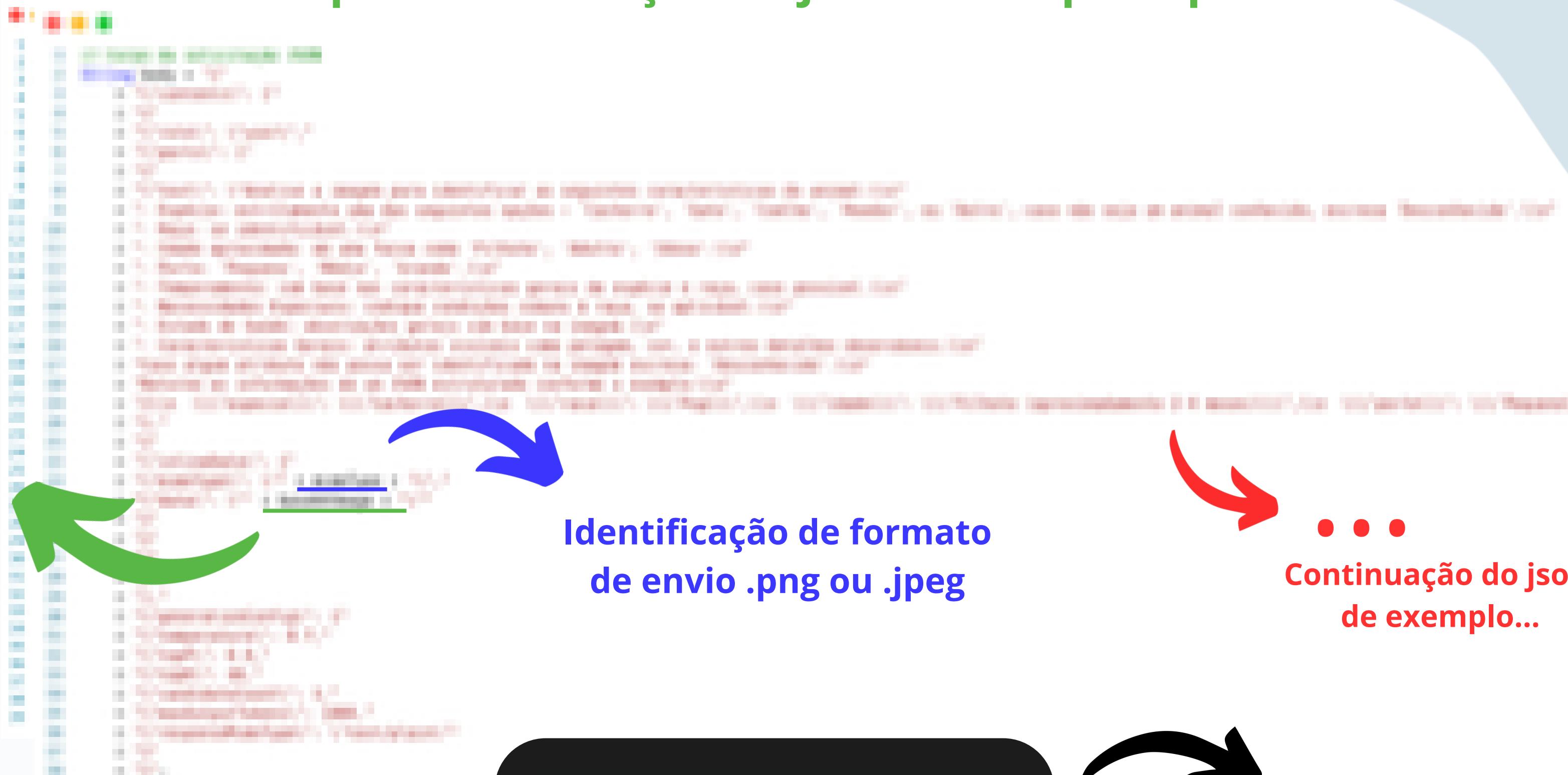


Intelligent System Canvas



Corpo de solicitação de json com o prompt

Imagem
em
Base64



Tratamento de erro,
ou inserção de
imagem não valida

Vídeo da demonstração da IA

The screenshot shows a web application interface for animal adoption. At the top, there is a blue header bar with the SOS Bichinhos logo on the left, a navigation menu with "Catalogo de adoção" and "Indicar Animal para ONG" buttons, and a user profile icon on the right. Below the header is a large central form titled "Encontrei um Animal". On the left side of the form is a circular image of a small, fluffy orange and white kitten. The form contains several input fields and dropdown menus:

- Espécie:** Especie do Animal (dropdown menu)
- Raça:** Raça do Animal (dropdown menu)
- Idade:** Idade aproximada do Animal (dropdown menu)
- Sexo:** Selecionar Opção (dropdown menu)
- Porte:** Selecionar Opção (dropdown menu)
- Localização:** Localização do Animal (dropdown menu)
- Estado de saúde do Animal:** Qual o estado de saúde do animal? (text input field)
- Possível temperamento do Animal:** Qual o temperamento do animal? (text input field)
- Necessidades Especiais:** O animal possui alguma necessidade especial? (text input field)
- Características Gerais:** Mais alguma característica? (text input field)

At the bottom of the form is a blue "Submit" button. The background of the page features a repeating pattern of paw prints.

© SOS Bichinhos

Conexão entre ONGs e Adotantes!

Social media icons: WhatsApp, Facebook, Instagram



Ajude uma ONG,
ajude uma vida,
ajude um Animal!



<https://github.com/ICEI-PUC-Minas-CC-TI/plmg-cc-ti2-2024-2-g02-sosbichinhos.git>