

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, Waldimir

15 PROTEGER A
VIDA TERRESTRE





PUC Minas



SOS Bichinhos

SOS Bichinhos



PUC Minas

Tópicos

- Funções do Sistema
- Fluxo de Usuário - User Flow
- Modelo Conceitual - DER
- Modelo Lógico - Diagrama de Esquema (DSC).
- .DER Estendido em notação Pé de Galinha
- Comandos SQL



Requisitos

• Requisitos Funcionais

Sistema de cadastros

Exibição dos Animais

Permitir que os usuários enviem fotos e comentários sobre os animais

A funcionalidade "Match de animais e donos"

Envio de imagem de um animal encontrado.

• Requisitos Não Funcionais

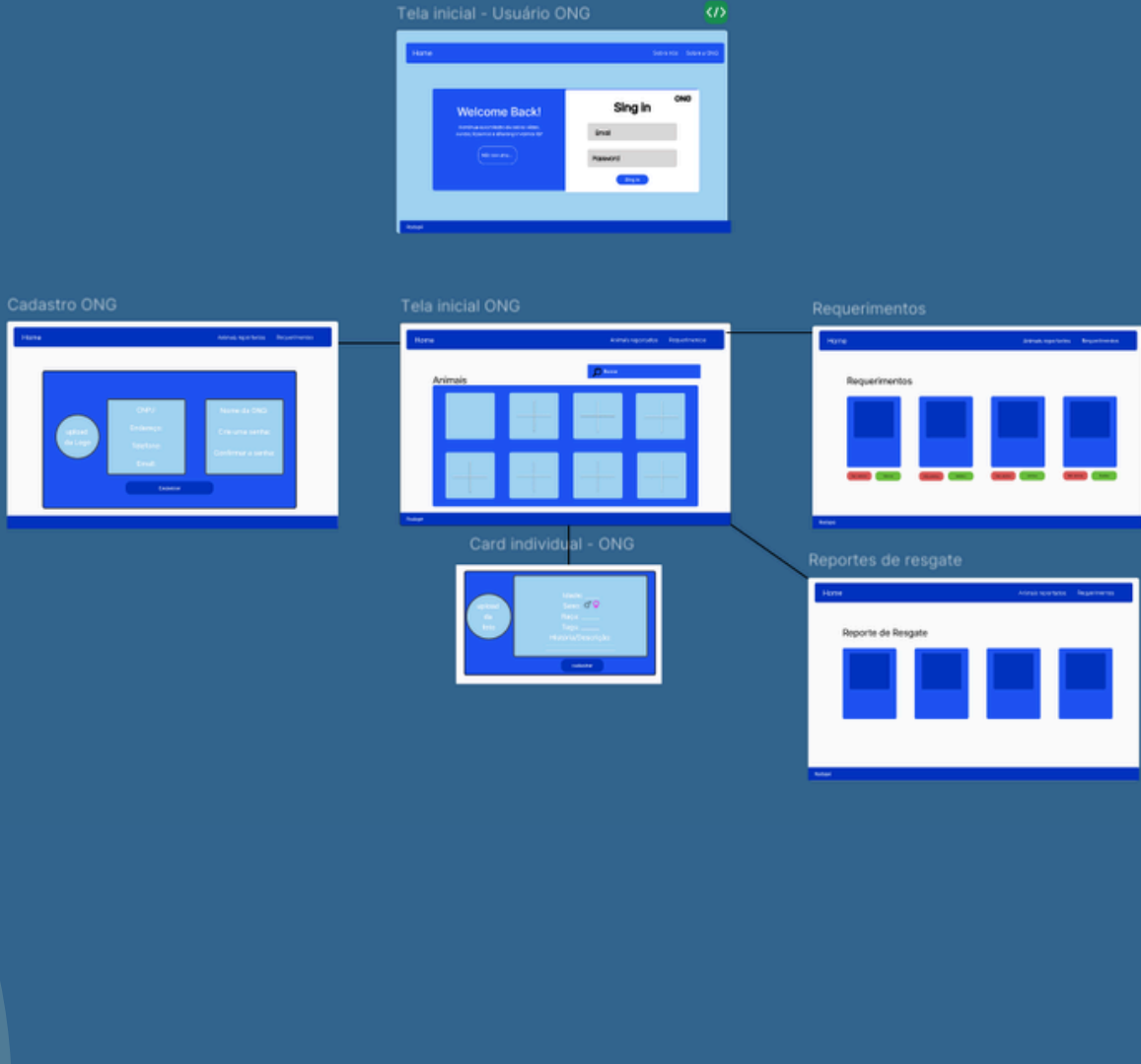
Lidar volume de acessos simultâneos:

O sistema deve ter um backup de dados confiável

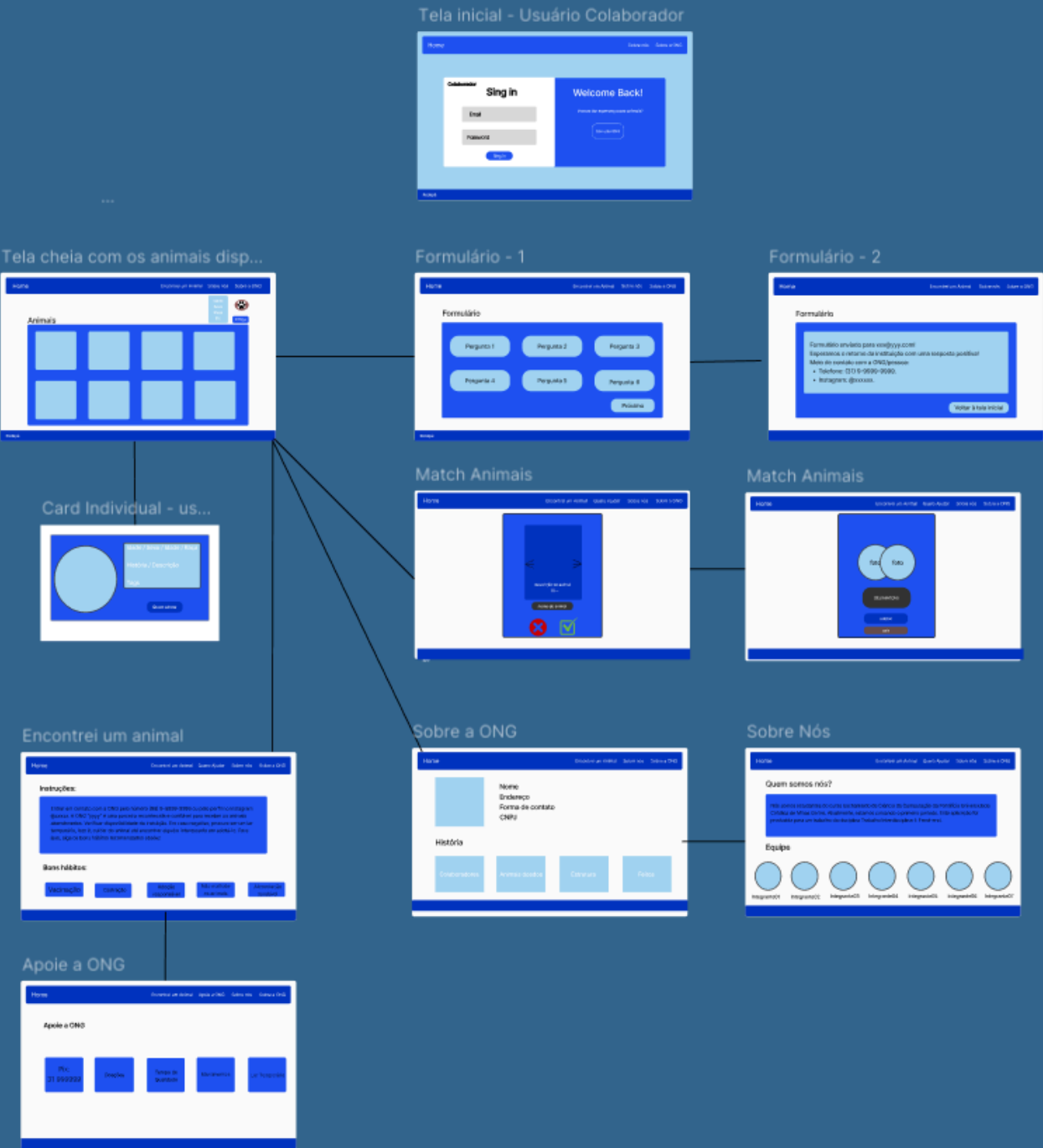
Linguagens HTML, CSS, JavaScript, java, postgresQL;

Fluxo de usuários (User Flow)

TELAS ONG



TELAS USUARIO



[Acessar Link](#)

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** 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, tags 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 também experimentar uma funcionalidade divertida e interativa de **Match** para encontrar animais com características compatíveis a sua.

Legenda




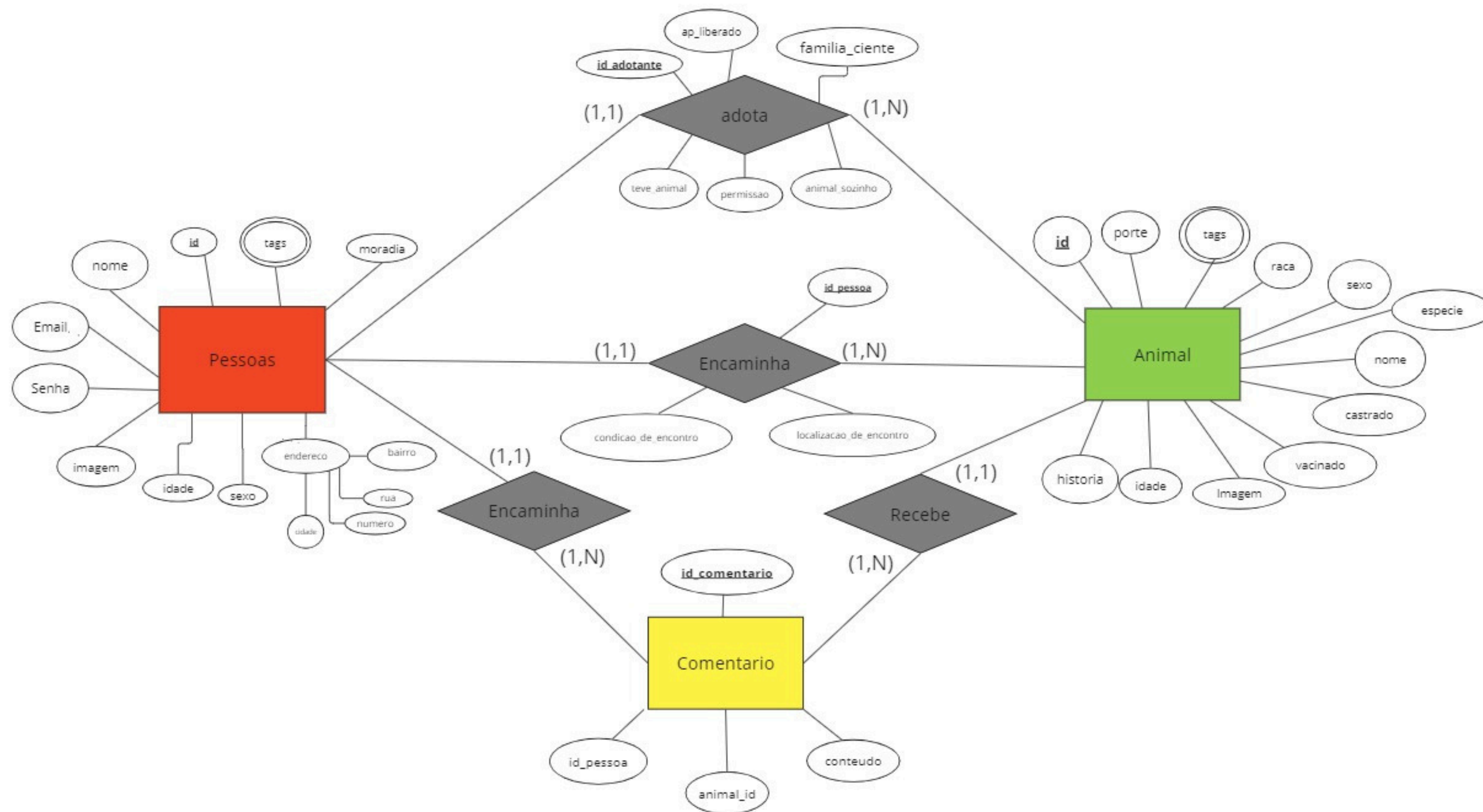
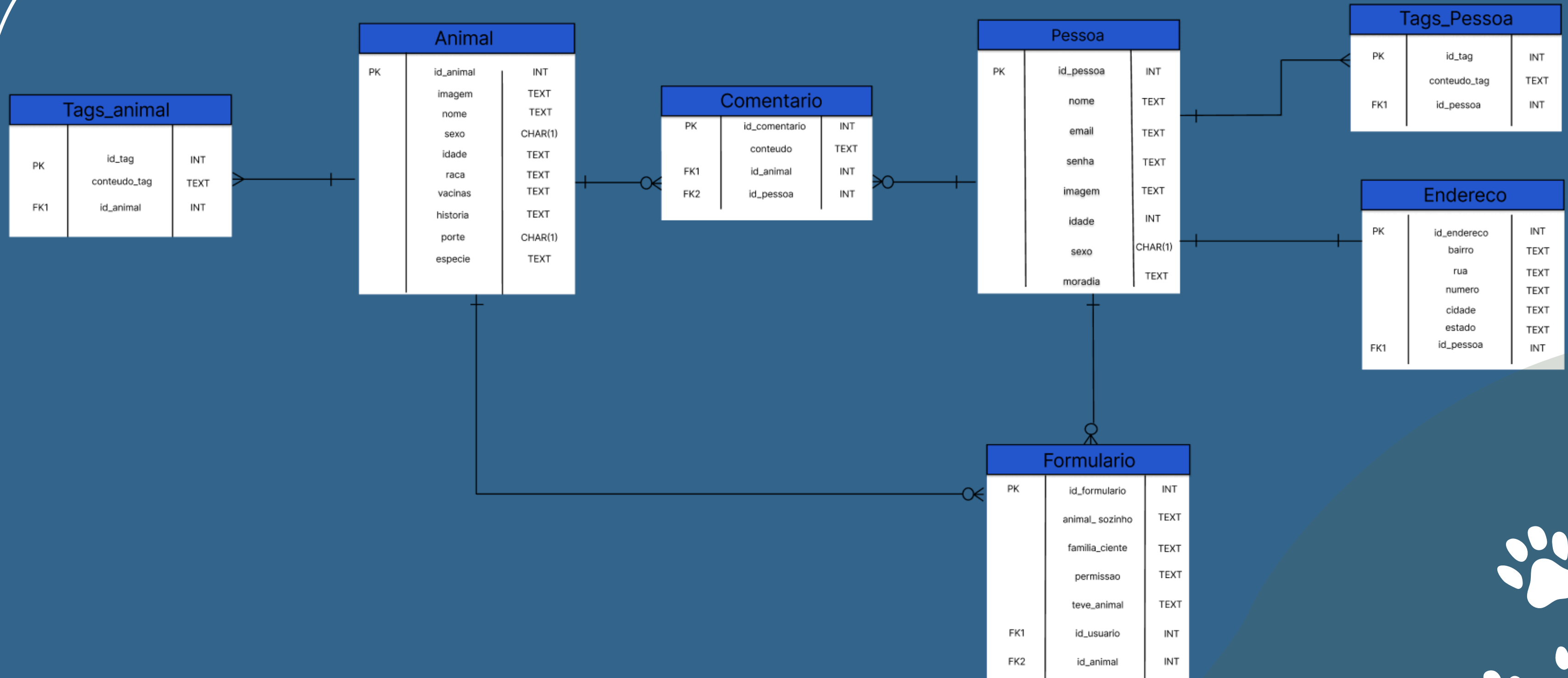
-  Entidade
-  Atributos
-  Funcionalidades



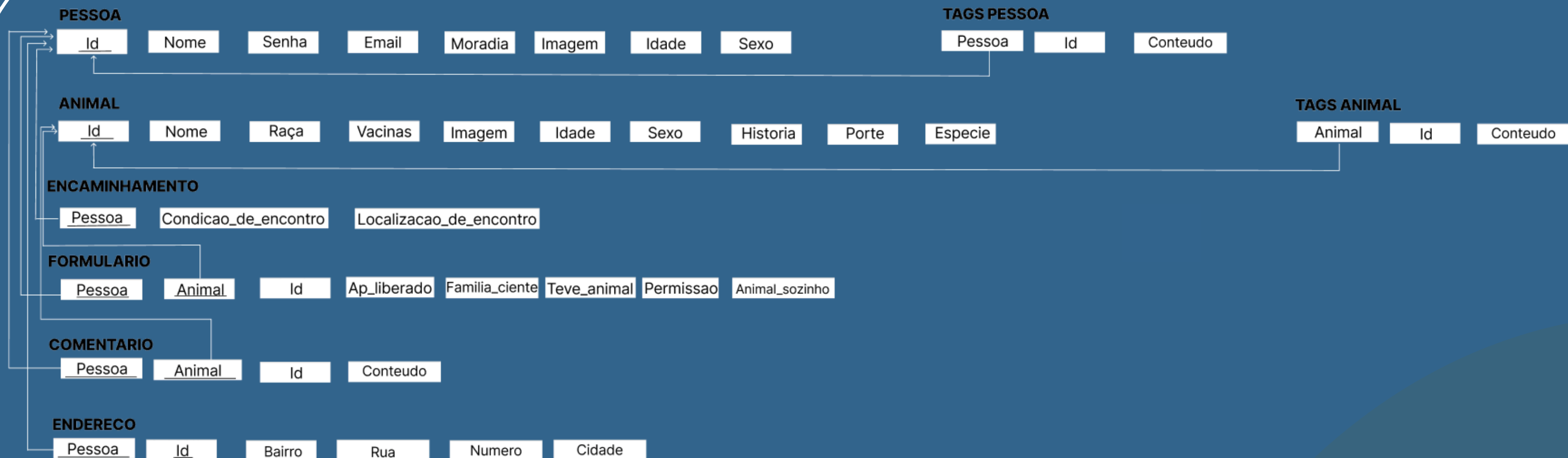
Diagrama DER



DER Estendido em Notação Pé de Galinha



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



Comandos SQL para a criação das tabelas



```
1  -- Tabela Animal
2  CREATE TABLE animal (
3      id_animal int PRIMARY KEY,
4      imagem text,
5      nome text,
6      sexo char (1),
7      idade text,
8      raca text,
9      vacinas text,
10     historia text,
11     porte text,
12     especie text
13 );
```

Animal		
PK	id_animal	INT
	imagem	TEXT
	nome	TEXT
	sexo	CHAR(1)
	idade	TEXT
	raça	TEXT
	vacinas	TEXT
	historia	TEXT
	porte	CHAR(1)
	especie	TEXT

Comandos SQL para a criação das tabelas



```
1  -- Tabela pessoa
2  CREATE TABLE pessoa (
3      id_pessoa int PRIMARY KEY,
4      nome text,
5      email text,
6      senha text,
7      imagem text,
8      idade int,
9      sexo char(1)
10 );
```

Pessoa		
PK	id_pessoa	INT
	nome	TEXT
	email	TEXT
	senha	TEXT
	imagem	TEXT
	idade	INT
	sexo	CHAR(1)

Comandos SQL para a criação das tabelas

```
1  -- Tabela tagsAnimal
2  CREATE TABLE tagsAnimal (
3      id_tagAnimal int PRIMARY KEY,
4      conteudo_tag text,
5
6      id_animal int,
7      CONSTRAINT FK_tagsAnimal_animal FOREIGN KEY (id_animal) REFERENCES animal(id_animal)
8
9  );
10
11 -- Tabela tagPessoas
12 CREATE TABLE tagsPessoa (
13     id_tagPessoa int PRIMARY KEY,
14     conteudo_tag text,
15
16     id_pessoa int,
17     CONSTRAINT FK_tagsPessoa_pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
18 );
```

Tags_animal		
PK	id_tag	INT
	conteudo_tag	TEXT
FK1	id_animal	INT

Tags_Pessoa		
PK	id_tag	INT
	conteudo_tag	TEXT
FK1	id_pessoa	INT

Comandos SQL para a criação das tabelas



```
1  -- Tabela comentario
2  CREATE TABLE comentario(
3      id_formulario int PRIMARY KEY,
4      conteudo text,
5
6      id_animal int,
7      CONSTRAINT FK_Comentario_Animal FOREIGN KEY (id_animal) REFERENCES animal(id_animal),
8      id_pessoa int,
9      CONSTRAINT FK_Comentario_Pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
10 );
```

Comentario		
PK	id_comentario	INT
	conteudo	TEXT
FK1	id_animal	INT
FK2	id_pessoa	INT

Comandos SQL para a criação das tabelas

```
1
2 CREATE TABLE endereco (
3     id_endereco int PRIMARY KEY,
4     bairro text,
5     rua text,
6     numero text,
7     cidade text,
8     estado text,
9
10    id_pessoa int,
11    CONSTRAINT FK_tagsPessoa_pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
12 );
```

Endereco		
PK	id_endereco	INT
	bairro	TEXT
	rua	TEXT
	numero	TEXT
	cidade	TEXT
	estado	TEXT
FK1	id_pessoa	INT

Comandos SQL para a criação das tabelas

```
1  -- Tabela formulario
2  CREATE TABLE formulario (
3      id_formulario int PRIMARY KEY,
4      animal_sozinho text,
5      familia_ciente text,
6      permissao text,
7      teve_animal text,
8
9      id_animal int,
10     CONSTRAINT FK_formulario_Animal FOREIGN KEY (id_animal) REFERENCES animal(id_animal),
11     id_pessoa int,
12     CONSTRAINT FK_formulario_Pessoa FOREIGN KEY (id_pessoa) REFERENCES pessoa(id_pessoa)
13 );
```

Formulario		
PK	id_formulario	INT
	animal_sozinho	TEXT
	familia_ciente	TEXT
	permissao	TEXT
	teve_animal	TEXT
FK1	id_usuario	INT
FK2	id_animal	INT



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

