

Objetivo: Um site com o propósito de facilitar/apoiar o resgate de animais abandonados.

Cor: azul claro #A0D3F2, azul escuro #0433BF e branco FFFFFFFF.

Tela Inicial: Divisão entre login corporativo ONG, e diretamente do usuário para a parte de página de adoção.

Renda: Contrato em que o preço varia dependendo da condição da ONG.

Logo:



Páginas a adicionar/mudar:

Usuário:

Apoie a causa - ONG (Doações, apoio financeiro, tempo de qualidade, mantimentos).

Sobre nós - Informações sobre nós que fizemos o projeto.

Sobre a ONG - Descrição da ONG.

Diferencial - Encontrei um Animal: Envia uma imagem para a ONG e através dela é coletado as características do animal. (Aqui será aplicado o sistema Inteligente, com a análise da imagem, a localização da foto, e o email da pessoa).

Cadastro (Login de usuário).

Formulário - local onde o usuário preenche suas informações quando for adotar um animal.

ONG:

Reporte de resgate - Local onde a ONG irá receber as informações do animal que ela irá resgatar (do diferencial).

Cadastro do Animal

Requerimentos

Alteração da descrição

TABELA BANCO DE DADOS:

--Exemplo de tabela:

```
create table Produto (  
    CodigoProduto char(3) PRIMARY KEY,  
    NomeProduto char(30),  
    PrecoProduto numeric(6,2),  
    Categoria char(3),  
    Fornecedor char(3),  
    CONSTRAINT FK_Produto_Categoria FOREIGN KEY (Categoria) REFERENCES  
categoria(Codigo)
```

);

-----END EXEEMPLO-----

-- Criando as tabelas do banco de dados

-- Tabela pessoa

```
CREATE TABLE pessoa (  
    id_pessoa int NOT NULL PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    senha VARCHAR(20) NOT NULL,  
    imagem VARCHAR(30) NOT NULL,  
    moradia VARCHAR(15) NOT NULL CHECK (moradia IN ('Casa', 'Apartamento', 'Outros')),  
    idade int NOT NULL,  
    sexo char(1) NOT NULL CHECK (sexo IN ('M', 'F'))  
);
```

-- Tabela Animal

```
CREATE TABLE animal (  
  id_animal int NOT NULL PRIMARY KEY ,  
  imagem VARCHAR(30) NOT NULL,  
  nome VARCHAR(50) NOT NULL,  
  sexo char (1) NOT NULL CHECK (sexo IN ('M', 'F')),  
  idade VARCHAR(20) NOT NULL,  
  castrado BOOLEAN NOT NULL,  
  raca VARCHAR(30) NOT NULL,  
  vacinas VARCHAR(50) NOT NULL,  
  historia text,  
  porte CHAR(1) NOT NULL CHECK (sexo IN ('P', 'M', 'G')),  
  especie VARCHAR(30) NOT NULL  
);
```

-- Tabela comentario

```
CREATE TABLE comentario(  
  id_comentario int NOT NULL PRIMARY KEY,  
  conteudo text,  
  
  id_animal int NOT NULL,  
  CONSTRAINT FK_Comentario_Animal FOREIGN KEY (id_animal) REFERENCES  
  animal(id_animal),  
  id_pessoa int NOT NULL,  
  CONSTRAINT FK_Comentario_Pessoa FOREIGN KEY (id_pessoa) REFERENCES  
  pessoa(id_pessoa)  
);
```

-- Tabela formulario

```
CREATE TABLE formulario (  
  id_formulario int NOT NULL PRIMARY KEY,  
  animal_sozinho VARCHAR(30),  
  familia_ciente BOOLEAN NOT NULL,  
  permissao BOOLEAN NOT NULL,  
  teve_animal BOOLEAN NOT NULL,  
  
  id_animal int NOT NULL,  
  CONSTRAINT FK_formulario_Animal FOREIGN KEY (id_animal) REFERENCES  
  animal(id_animal),  
  id_pessoa int NOT NULL,  
  CONSTRAINT FK_formulario_Pessoa FOREIGN KEY (id_pessoa) REFERENCES  
  pessoa(id_pessoa)  
);
```

-- Tabela tagsAnimal

```
CREATE TABLE tagsAnimal (  
  id_tagAnimal int NOT NULL PRIMARY KEY,  
  conteudo_tag VARCHAR(15),  
  
  id_animal int NOT NULL,  
  CONSTRAINT FK_tagsAnimal_animal FOREIGN KEY (id_animal) REFERENCES  
  animal(id_animal)  
  
);  
  
-- Tabela tagPessoas  
CREATE TABLE tagsPessoa (  
  id_tagPessoa int NOT NULL PRIMARY KEY,  
  conteudo_tag VARCHAR(15),  
  
  id_pessoa int NOT NULL,  
  CONSTRAINT FK_tagsPessoa_pessoa FOREIGN KEY (id_pessoa) REFERENCES  
  pessoa(id_pessoa)  
  
);  
  
CREATE TABLE endereco (  
  id_endereco int NOT NULL PRIMARY KEY,  
  bairro VARCHAR(30) NOT NULL,  
  rua VARCHAR(30) NOT NULL,  
  numero VARCHAR(10) NOT NULL,  
  cidade VARCHAR(30) NOT NULL,  
  estado VARCHAR(20) NOT NULL,  
  
  id_pessoa int NOT NULL ,  
  CONSTRAINT FK_tagsPessoa_pessoa FOREIGN KEY (id_pessoa) REFERENCES  
  pessoa(id_pessoa)  
  
);
```

----- Especificação Funcional -----

Requisitos

Requisitos Funcionais

ID	Descrição do Requisito	Prioridade
RF-001	O sistema deve permitir que o usuário faça seu cadastro pessoal.	ALTA

RF-002	O sistema deve permitir que o usuário faça o cadastro do animal.	ALTA
RF-003	O sistema deve exibir quais animais estão disponíveis para adoção.	ALTA
RF-004	O sistema deve permitir que o usuário envie uma imagem de um animal encontrado.	ALTA
RF-005	O sistema deve permitir que o adotante preencha a documentação para a adoção.	ALTA
RF-006	O sistema deve permitir que a ONG visualize os requisitos de adoção.	ALTA
RF-007	O sistema deve permitir que o usuário trafegue entre as telas.	ALTA
RF-008	O sistema deve permitir que a ONG visualize os animais encaminhados.	ALTA
RF-009	O sistema deve permitir que os usuários comentários sobre os animais .	MÉDIA
RF-010	O sistema deve permitir que usuários se ofereçam como lares temporários.	MÉDIA
RF-011	O sistema deve ter a funcionalidade "Match de animais e donos".	BAIXA

Requisitos Não-Funcionais

ID	Descrição do Requisito	Prioridade
RNF-001	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, Java e JavaScript e o sistema gerenciador de banco de dados PostgreSQL.	ALTA

RNF-002	O sistema deve ser responsivo para rodar em dispositivos móveis.	MÉDIA
RNF-003	O sistema deve ter alta disponibilidade e ser capaz de lidar com um grande volume de acessos simultâneos.	MÉDIA
RNF-004	O sistema deve ter um sistema de backup de dados confiável para evitar perda de informações importantes.	ALTA
RNF-005	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.	ALTA
RNF-006	O sistema deve fornecer suporte técnico eficiente para os usuários em caso de problemas ou dúvidas.	MÉDIA