

**INSTITUTO FEDERAL DO SUDESTE DE MINAS GERAIS
CAMPUS AVANÇADO BOM SUCESSO**

LUIZ GUSTAVO SANTOS

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA ADOÇÃO DE
ANIMAIS**

BOM SUCESSO-MG

2021

LUIZ GUSTAVO SANTOS

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA ADOÇÃO DE
ANIMAIS**

Monografia apresentada como requisito parcial de aprovação da disciplina de Trabalho de Conclusão de Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Sudeste de Minas Gerais - Campus Avançado Bom Sucesso-MG.

Orientador: Graziany Thiago Fonseca

BOM SUCESSO-MG

2021

Dados internacionais de catalogação na publicação (CIP)
Bibliotecária responsável Maria de Lourdes Cardoso CRB-6/3242

S237d Santos, Luiz Gustavo, 1987 -

Desenvolvimento de um sistema *web* para adoção de animais / Luiz Gustavo Santos. -- 2021.

37 f. : il. ; 30 cm.

Orientador: Graziany Thiago Fonseca

Monografia (Graduação) - Instituto Federal do Sudeste de Minas Gerais, Campus Avançado Bom Sucesso, Curso de Análise e Desenvolvimento de Sistemas, Bom Sucesso-MG, 2021.

1. Software – Desenvolvimento. 2. PostgreSQL (Linguagem de programação de computador). 3. JavaScript (Linguagem de programação de computador). 4. Animais domésticos. I. Fonseca, Graziany Thiago. II. Instituto Federal do Sudeste de Minas Gerais, Campus Avançado Bom Sucesso. III. Título.

CDD: 005.1



MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SUDESTE DE MINAS GERAIS

COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA Nº 14 / 2021 -
BSCCCADS (11.01.10.01.01.02.01)

Nº do Protocolo: NÃO PROTOCOLADO

Juiz de Fora-MG, 01 de Setembro de 2021

TERMO DE APROVAÇÃO

LUIZ GUSTAVO SANTOS

DESENVOLVIMENTO DE UM SISTEMA WEB PARA ADOÇÃO DE ANIMAIS

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - *Campus* Avançado Bom Sucesso.

(Assinado digitalmente em 01/09/2021 21:40)

GRAZIANY THIAGO FONSECA
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 1966904

(Assinado digitalmente em 10/09/2021 21:44)

PEDRO HENRIQUE DE OLIVEIRA E SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 1758559

(Assinado digitalmente em 10/09/2021 21:44)

VICTOR SCHMIDT COMITTI
PROFESSOR ENS BASICO TECN TECNOLOGICO
Matrícula: 3082930

Para verificar a autenticidade deste documento entre em <https://sig.ifsudestemg.edu.br/documentos/> informando seu número: 14, ano: 2021, tipo: COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA, data de emissão: 01/09/2021 e o código de verificação: 6339145fd6

AGRADECIMENTOS

Agradeço a Deus por chegar até aqui, agradeço também aos meus professores e colegas de curso que me ajudaram nesta caminhada.

Um agradecimento especial a minha família, que está sempre ao meu lado em todos os momentos da minha vida.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema web para adoção de animais, uma SPA (*Single Page Application*), capaz de atender tanto a demanda de um adotante quanto a parte administrativa, sendo o primeiro o público alvo principal. A aplicação criada se adapta a diversos dispositivos, podendo ser acessada via computador, tablet ou pelo celular. O Sistema possui um *backend* administrativo e uma interface *frontend* para usuários gerais. O Cão sem casa é um sistema web desenvolvido utilizando o *framework* ReactJS juntamente com Javascript e CSS, e para banco de dados o PostgreSQL.

Palavras-chave: JavaScript. ReactJS. PostgreSQL. SPA. Sistema Web.

ABSTRACT

This work presents the development of a web system for animal adoption, a SPA(Single Page Application), capable of meeting both the demand of an adopter and the administrative part, the first being the main target audience. The created application adapts to various devices and can be accessed via computer, tablet or cell phone. The System has an administrative backend and a frontend interface for general users. Cão sem casa is a web system developed using the ReactJS framework together with Javascript and CSS, and PostgreSQL for database.

Keywords: Javascript. ReactJS. PostgreSQL. SPA. Sistema Web.

LISTA DE FIGURAS

Figura 1 – Fases do Processo Unificado	20
Figura 2 – Diagrama de Caso de Uso do Sistema	22
Figura 3 – Código Javascript para conexão com banco de dados utilizando o Sequelize	23
Figura 4 – Modelo Relacional do Banco de dados	24
Figura 5 – Tela Home, página principal da aplicação	25
Figura 6 – Tela da página Sobre Nós	26
Figura 7 – Tela da página Adotar	26
Figura 8 – Tela da página Perfil do Animal	27
Figura 9 – Tela da página Perfil do Animal com formulário de adoção	27
Figura 10 – Tela da página Sucesso	28
Figura 11 – Tela Cadastro	28
Figura 12 – Tela Login	29
Figura 13 – Tela da página Minha Conta	29
Figura 14 – Tela Login Dashboard	30
Figura 15 – Tela Principal do Dashboard	31
Figura 16 – Tela de listagem dos animais	31
Figura 17 – Formulário de cadastro de animais	32
Figura 18 – Formulário de edição dos dados do animal	32
Figura 19 – Tela com as solicitações de adoção	33
Figura 20 – Tela para confirmação da adoção	33
Figura 21 – Tela final adoção	34

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CORS	Cross-origin Resource Sharing
CSS	Cascading Style Sheets
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
NPM	Node Package Manager
ONG	Organização Não Governamental
ORM	Object-Relational Mapping
PU	Processo Unificado
REST	Representational State Transfer
SGBD	Sistemas de Gestão de Base de Dados
SPA	Single Page Application
SQL	Structured Query Language

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivo Específico	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Adoção de Animais	16
2.2	Ferramentas e Tecnologias	16
2.2.1	Node.js e NPM (<i>Node Package Manager</i>)	16
2.2.2	Javascript	17
2.2.3	ReactJS	17
2.2.4	React Router	17
2.2.5	Express	18
2.2.6	Sequelize	18
2.2.7	Axios	18
2.2.8	Cors	18
2.2.9	Banco de Dados	19
2.2.10	API REST	19
3	MATERIAIS E MÉTODOS	20
3.1	Metodologia	20
3.2	Métodos utilizados	21
3.3	Interface gráfica	21
3.4	Navegação	21
3.5	Servidores	21
4	PROJETO E DESENVOLVIMENTO	22
4.1	Modelagem	22

4.1.1	Diagrama de Caso de Uso	22
4.1.2	Manipulação dos dados	23
4.1.3	Modelo Relacional de Banco de dados	23
5	RESULTADOS	25
5.1	GUI - Interface Gráfica de Usuário	25
5.2	Interface do Administrador	30
6	CONCLUSÃO	35
	REFERÊNCIAS	36

1 INTRODUÇÃO

Segundo o Instituto Pet Brasil (2019) existem cerca de 139,3 milhões de animais de estimação no Brasil, destes sendo aproximadamente cerca de 54,2 milhões de cães e 23,9 milhões de gatos. Desses animais, cerca de 3,9 milhões são animais em condição de vulnerabilidade, aqueles que vivem sob tutela de famílias carentes ou que vivem nas ruas sob algum cuidado de pessoas.

Ainda conforme os dados do Instituto Pet Brasil (2019), entres os animais em condição de vulnerabilidade, não estão incluídos os animais que vivem por determinado tempo sem um tutor. Boa parte desses animais vivem sob a tutela de ONGs (Organizações não Governamentais) que tomam responsabilidade por preservar esses animais e facilitar a adoção voluntária.

Diante do cenário atual do nosso país e com as medidas de isolamento social para conter o avanço da pandemia do coronavírus, uma grande parte das pessoas têm se sentindo solitárias. Uma alternativa encontrada por muitos foi a adoção de animais de estimação, para melhorar essa nova rotina e ocupar parte do tempo livre (HUCS, 2020).

Entretanto hoje em dia em diversas ONGs e abrigos para animais o sistema de adoção ainda é feito da forma tradicional através de visitaçao ao local onde se encontram os animais ou em feiras de adoção, para que o potencial adotante possa escolher o animal que mais lhe agrade. Conforme descrito acima, esse tipo de abordagem não está sendo possível na maioria das cidades devido a situação em que o Brasil e restante do mundo está vivendo.

Este trabalho tem como proposta desenvolver um sistema que facilite a busca por animais a serem adotados, possibilitando ao usuário uma maior comodidade ao buscar por um novo animal de estimação.

O sistema proposto irá permitir ao usuário fazer buscas por diferentes raças, tamanhos e porte dos animais, além solicitar a adoção de um determinado animal. Também possibilitará ao administrador do sistema fazer o cadastro de novos animais, gerenciar o processo de adoção, cadastrar feiras e gerar relatórios.

O presente trabalho está dividido em seis capítulos. No primeiro capítulo apresenta-se o projeto, expondo uma breve contextualização e apresentando a problemática vislumbrada, assim como o objetivo geral. No segundo capítulo, é realizada uma

revisão da literatura relacionada à área de desenvolvimento de sistemas, engenharia de software e banco de dados. No terceiro, quarto e quinto capítulos são apresentados todo o processo de desenvolvimento do sistema, bem como os resultados alcançados, onde são descritos detalhadamente os requisitos do sistema e os diagramas criados para o desenvolvimento do mesmo, alinhados a cada funcionalidade que a ferramenta oferecerá. Por fim, o sexto capítulo contém as considerações finais e possibilidades para trabalhos futuros.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Apresentar todo o processo de desenvolvimento de um sistema web para Adoção de animais.

1.1.2 Objetivo Específico

Visando atingir o objetivo principal, alguns objetivos específicos são requeridos, entre eles:

- Análise da literatura de temas relacionados;
- Levantamento de requisitos da ferramenta a ser criada;
- Modelagem do sistema;
- Criação da Base de Dados;
- Desenvolvimento do Sistema, provendo uma interface amigável para o usuário, facilitando o processo de busca por determinado animal;

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os assuntos que fundamentam este trabalho, apresentando tecnologias envolvidas e conceitos computacionais utilizados na construção do sistema.

2.1 ADOÇÃO DE ANIMAIS

No Brasil, ainda não existe uma lei específica que rege os processos para adoção de animais, tanto domésticos quanto silvestres. Entretanto, existem regulamentações no que diz respeito ao abuso, maus tratos desses animais, a fim de controlar operações de compra e venda, e restringir o comércio ilegal, implicando em multa e detenção (BATISTA, 2018).

Adotar um animal de estimação traz consigo uma série de responsabilidades. Por isso as instituições defensoras dos animais lutam pela adoção consciente, para que os novos adotantes garantam o bem estar do animal.

Como pré-requisitos para adoção, algumas organizações estabelecem condições a serem seguidas pelo adotante para que o processo seja realizado, visando garantir que o animal, não volte às ruas (ABRIGO FLORA E FAUNA, 20201).

Dentre as condições necessárias para adoção, pode-se destacar:

- Ter mais de 21 anos;
- Ter um espaço suficiente para o novo membro da casa;
- Providenciar para que o animal tenha uma vida boa e feliz;
- Responder a uma entrevista e preencher o Termo de Adoção Responsável;
- Estar ciente que um animal pode viver mais de 15 anos;

2.2 FERRAMENTAS E TECNOLOGIAS

2.2.1 Node.js e NPM (*Node Package Manager*)

O Node.js é um *runtime* de Javascript no lado do servidor, *open-source* e multiplataforma. Possibilitando o desenvolvimento de aplicações de forma rápida e estável (GOSS, 2017).

Foi desenvolvido por Ryan Dahl no de 2009 em torno do motor V8 do chrome (navegador de internet), como uma alternativa para rodar programas Javascript fora do ambiente do *browser* (PFUTZENREUTER, 2021).

Para utilizar o Node, é preciso tê-lo instalado na máquina. Basta entrar no site do Node.js e seguir o guia de instalação recomendado conforme o sistema operacional escolhido (PEREIRA, 2016).

O NPM, é o gerenciador de pacotes do Node.js, ele é instalado junto com o Node. é uma ferramenta que facilita a instalação de módulos de terceiros, para utilizá-lo basta apenas usar o comando `npm install <nome do pacote>` que automaticamente todas as dependências deste módulo serão baixadas (NPM, 2021).

2.2.2 Javascript

JavaScript é uma linguagem interpretada e orientada a objetos, conhecida por sua utilização em páginas web, mas que também é muito utilizada em ambientes fora dos navegadores como é no caso do Node.js (MDN, 2021).

É uma linguagem de *scripting* dinâmica que permite a construção de objetos baseados em protótipos em tempo de execução, podendo funcionar tanto como uma linguagem procedural como uma linguagem orientada a objetos. Com uma sintaxe parecida com a Java e C++, tem o intuito de diminuir os conceitos necessários para se aprender a linguagem (MDN, 2021).

2.2.3 ReactJS

ReactJS é uma biblioteca JavaScript baseada em componentes para para criação de interfaces de usuários (REACTJS, 2021) desenvolvida pelo Facebook, ele possibilita uma melhor organização do conteúdo que será renderizado na tela do usuário.

Utiliza o virtual DOM (*Document Object Model*), que é um objeto JavaScript. Isso melhora o desempenho dos aplicativos, pois o virtual DOM JavaScript é mais rápido que o DOM nativo do *browser*, toda vez que acontece uma mudança no DOM real, o virtual se modifica imediatamente (L, 2021).

2.2.4 React Router

Quando estamos falando de uma SPA (*Single Page Application*), tudo acontece em uma única página e o processo de roteamento é o que define os *endpoints* para as requisições dos seus clientes. O React por padrão não vem com um sistema de rotea-

mento, a solução para este problema foi resolvida por dois engenheiros que criaram o React Router, que foi adotado pela comunidade ReactJS (BANKS; PORCELLO, 2017).

O React Router é um conjunto de componentes para navegação, eles facilitam a navegação por links sem precisar recarregar a página.

2.2.5 Express

O Express é um *framework web* para Node.js, que foi inspirado pelo *framework Sinatra* do Ruby. Com ele, podemos desenvolver aplicações de pequeno e grande porte, sem nenhum problema. Esse *framework* permite também a construção de APIs e aplicações web (PEREIRA, 2016).

Pereira (2016) diz que seu foco é trabalhar com perfeição manipulando *views*, *routes* e *controllers*, ficando à escolha do usuário trabalhar com *models* e usar qualquer *framework* de persistência sem gerar nenhum conflito ou incompatibilidade com Express.

2.2.6 Sequelize

Sequelize é um ORM (Object-Relational Mapping) Node.js baseado em *promises* que tem suporte a diversos bancos de dados como, PostgreSQL, MariaDB, MySQL, SQLite e MSSQL, ele faz o mapeamento de dados relacionais (tabelas, colunas e linhas) para objetos Javascript. (SEQUELIZE, 2021).

Segundo Orlandi (2018), ele permite criar, buscar, alterar e remover dados do banco de dados utilizando métodos JavaScript, além de permitir a modificação da estrutura das tabelas, com isso temos muita facilidade tanto na criação, população e migração de banco de dados.

2.2.7 Axios

Axios é um cliente HTTP (*Hypertext Transfer Protocol*) para fazer requisições, ele é baseado em *promisses*. Pode ser utilizado tanto no navegador quanto no Node.js (AXIOS, 2021).

2.2.8 Cors

O CORS (*Cross-origin Resource Sharing*) é um pacote Node.js para fornecer um *middleware Connect / Express* que pode ser usado para habilitar CORS com várias opções (EXPRESSJS, 2021).

2.2.9 Banco de Dados

Banco de dados são coleções organizadas de informações, isto é, uma estrutura bem organizada de dados que permite a extração de informações, geralmente esses dados são armazenados eletronicamente em um sistema de computador (ORACLE, 2021).

De acordo com Date (2004) os usuários de um sistema podem solicitar que o mesmo realize diversas operações, como acrescentar novos arquivos, inserir dados em arquivos existentes, buscar dados de arquivos, alterar e remover.

Dentre os vários banco de dados existentes, podemos citar o PostgreSQL, um sistema de banco de dados relacional de objeto que usa e estende a linguagem SQL combinada com diversos recursos para armazenamento e escalabilidade com segurança (POSTGRESQL, 2021).

Segundo Gomes (2019), as principais vantagens em seu uso estão relacionadas à economia e ao alto desempenho oferecidos pelo SGBD. Suporta um intenso fluxo de dados com garantia de estabilidade e segurança.

2.2.10 API REST

Antes de apresentar o conceito de API REST, é preciso entender o que é uma API (*Application Programming Interface*). De acordo com Souza (2020), são um conjunto de rotinas e padrões de programação que permite a comunicação de dados entre aplicações. Utiliza requisições HTTP para realização das operações necessárias para a manipulação dos dados.

API REST, ou API RESTful como também é chamada, é uma interface de programação de aplicações que é construída seguindo os padrões da arquitetura REST. A sigla REST significa *Representational State Transfer* (Transferência Representacional de Estado, em português) (RED HAT, 2021).

Algumas das restrições determinadas pela arquitetura REST são que as aplicações tanto no cliente quanto no servidor devem ser separadas, as requisições são feitas de forma independente, a API deve fazer uso do *cache* para evitar chamadas recorrentes ao servidor e ter uma interface uniforme, com identificação de recursos e a manipulação dos mesmos deve ser feita através de representação com mensagens auto descritivas e *links* para navegação pelo aplicativo (SOUZA, 2020).

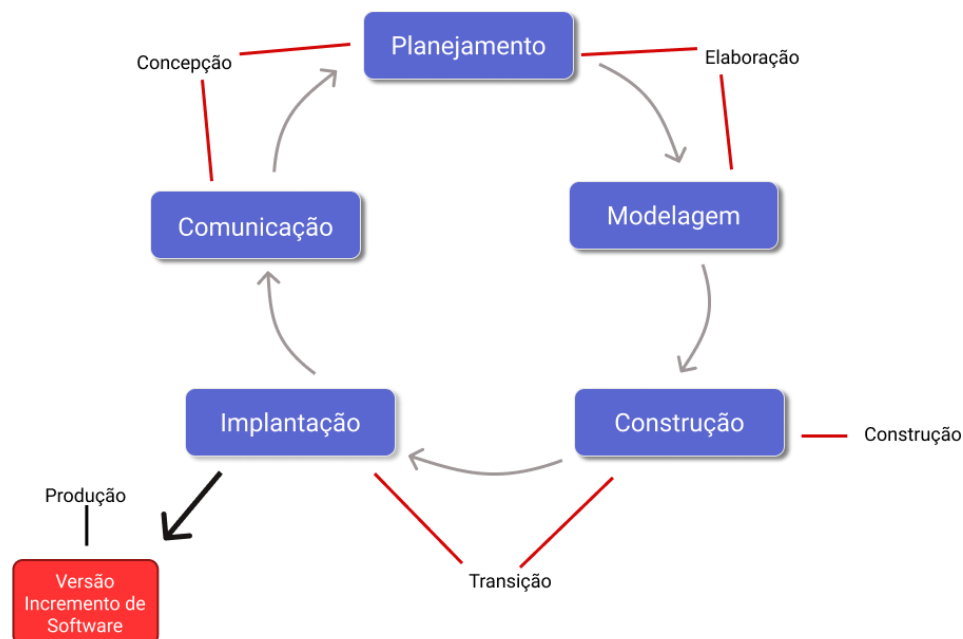
3 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados os materiais e os métodos utilizados neste trabalho.

3.1 METODOLOGIA

A metodologia utilizada neste trabalho foi inspirada no processo unificado (PU). Este processo é uma estrutura que pode ser customizada conforme as necessidades do projeto. Faz uso da Linguagem de Modelagem Unificada (*Unified Modeling Language*, UML) para organizar os artefatos do sistema. Este sistema possui um ciclo de vida que constitui de várias interações, produzido em quatro fases: Concepção, Elaboração, Construção e Transição. Cada uma das fases sendo subdividida em fluxos de requisitos, análise, projeto, implementação e teste (LARMAN, 2005).

Figura 1 – Fases do Processo Unificado



Fonte: Pressman (2009)

Durante a fase de concepção identificam-se as necessidades de negócio para o *software*, o planejamento é desenvolvido e se define um cronograma. Na fase de elaboração detém-se os requisitos funcionais do sistema que foram obtidos na fase anterior, faz-se a ampliação do esboço da arquitetura para uma arquitetura de

desenvolvimento do sistema. A fase de construção é onde o produto é desenvolvido. Já na fase de transição é onde o produto tem sua primeira versão entregue ao cliente para teste (LARMAN, 2005).

3.2 MÉTODOS UTILIZADOS

Conforme foram surgindo as necessidades durante o desenvolvimento da aplicação utilizando JavaScript e Node.js, foi preciso fazer a instalação de algumas dependências. Os módulos mencionados no item 2.2, foram utilizados de acordo com as necessidades que foram surgindo durante o desenvolvimento da aplicação.

3.3 INTERFACE GRÁFICA

Com ReactJS foi desenvolvida toda parte de componentes da interface, além de tratar das interações do usuário dentro do sistema, quanto a parte de estilos da aplicação foi utilizado o CSS (*Cascade Style Sheet*).

3.4 NAVEGAÇÃO

Para realizar a navegação entre as telas, foi utilizado o React Router, ele é uma biblioteca desenvolvida para ser utilizado em conjunto com o ReactJS, para gerenciar toda a parte de roteamento da aplicação, tornando nossa aplicação numa SPA.

Em uma aplicação SPA, a transição entre os templates é feita sem a necessidade de fazer reload da página, não sendo necessário o usuário ir de uma página para outra. Isso faz com que o sistema fique muito mais performático já que ele carrega o sistema completo na primeira requisição de forma assíncrona, sem que o usuário precise esperar que todo conteúdo seja carregado.

3.5 SERVIDORES

O servidor do sistema foi desenvolvido com o Express, além de outros módulos necessários para seu melhor desempenho, como o *Body Parser* que faz o parseamento dos dados que vem das requisições para o formato JSON (*JavaScript Object Notation*) e módulo Cors que cuida para que a comunicação com entre Servidor e o Cliente ReactJS seja estabelecida corretamente.

Para comunicação com o banco de dados PostgreSQL, foi utilizado o módulo Sequelize, que lida com a conexão com o banco de dados e trata da modelagem das tabelas no banco.

4 PROJETO E DESENVOLVIMENTO

Neste capítulo é descrito a modelagem do projeto - definição das funcionalidades da aplicação e manipulação dos dados. Depois de estabelecida a abordagem, foi feito o desenvolvimento da aplicação.

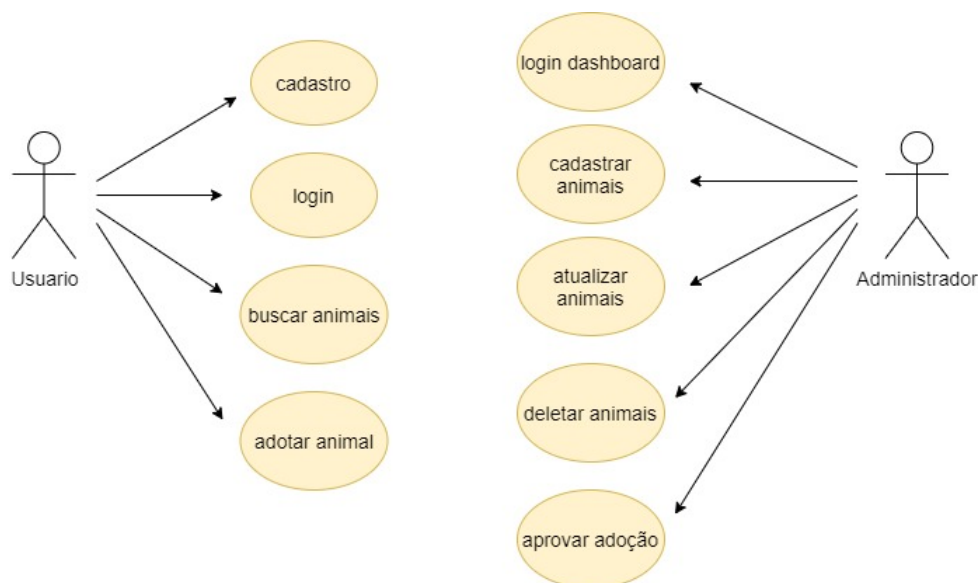
O nome escolhido para o sistema desenvolvido foi Cão Sem casa, por ser uma aplicação voltada para adoção de animais.

4.1 MODELAGEM

4.1.1 Diagrama de Caso de Uso

Antes de começar a programação da aplicação, é necessário planejar o que será preciso para o bom funcionamento do sistema, para isso foi feito então o diagrama de caso de uso do sistema, como pode ser visto na Figura 2.

Figura 2 – Diagrama de Caso de Uso do Sistema



Fonte: do Autor

Diagramas de caso de uso resumem os detalhes dos usuários do seu sistema (também conhecidos como atores) e suas interações com o sistema. Seu objetivo é demonstrar as diferentes maneiras que o usuário pode interagir com um sistema (LUCIDCHART, 2021).

4.1.2 Manipulação dos dados

Com a utilização do PostgreSQL como banco de dados do projeto, optou-se por fazer uso da biblioteca *Sequelize* para a comunicação entre o PostgreSQL e a aplicação, já que o banco de dados não se comunica diretamente com a aplicação. Na Figura 3 podemos ver o trecho de código de como foi feita a conexão com PostgreSQL.

Figura 3 – Código Javascript para conexão com banco de dados utilizando o Sequelize

```
1  const Sequelize = require('sequelize')
2
3  const env = process.env.NODE_ENV || 'development'
4  const config = require(`_${dirname}/../config/database.js`)[env]
5
6  const Adoption = require('../models/Adoption.js')
7  const Animal = require('../models/Animal.js')
8  const User = require('../models/User.js')
9
10 const models = [User, Animal, Adoption]
11
12 class Database {
13   constructor() {
14     this.init()
15   }
16
17   init() {
18     if (config.use_env_variable) {
19       this.connection = new Sequelize(
20         process.env[config.use_env_variable],
21         config
22       )
23     } else {
24       this.connection = new Sequelize(
25         config.database,
26         config.username,
27         config.password,
28         {
29           ...config
30         }
31       )
32     }
33     models
34       .map((model) => model.init(this.connection))
35       .map(
36         (model) => model.associate && model.associate(this.connection.models)
37       )
38   }
39 }
40
41 module.exports = new Database()
```

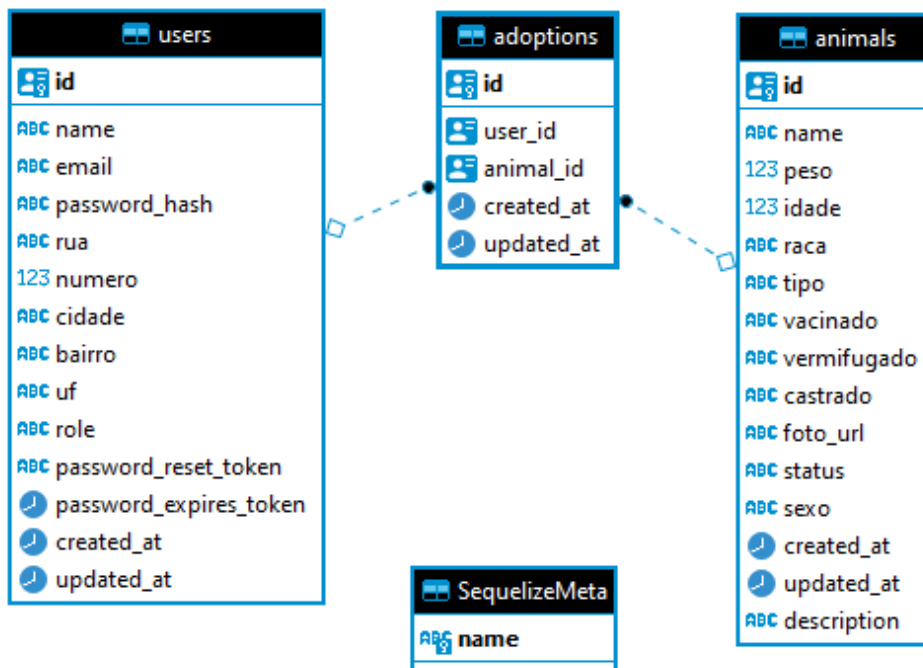
Fonte: do Autor

4.1.3 Modelo Relacional de Banco de dados

Através do *software* DBeaver, uma ferramenta gratuita de banco de dados multiplataforma para desenvolvedores, administradores de banco de dados e analistas, foi feito o gerenciamento do banco de dados.

A Figura 4 mostra a representação do modelo relacional do banco de dados.

Figura 4 – Modelo Relacional do Banco de dados



Fonte: do Autor

O modelo acima mostra uma representação das tabelas criadas no banco de dados, onde a tabela *users* guarda os dados de cada usuário cadastrado no sistema, a tabela *animals* as informações dos animais, já a tabela *adoption* guarda as informações do usuário que realizou a adoção e também do animal adotado e uma tabela chamada *sequelizeMeta*, esta é gerada automaticamente pelo Sequelize.

5 RESULTADOS

Neste capítulo será apresentada a aplicação que foi desenvolvida como resultado deste trabalho.

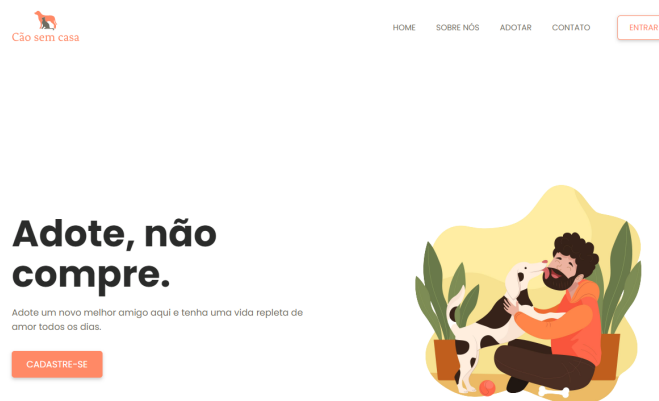
5.1 GUI - INTERFACE GRÁFICA DE USUÁRIO

A interface é por onde o usuário interage com o sistema, a aplicação Cão sem casa foi desenvolvida no formato de um site, onde o usuário pode navegar por partes do sistema sem precisar estar logado.

A tela *Home*, é a visão inicial do sistema, onde o usuário tem o primeiro contato com o sistema, ela contém um menu no topo para navegação entre as páginas e também botões para as páginas de login ou cadastro, caso o usuário ainda não seja cadastrado no sistema.

A Figura 5 mostra uma parte da página *Home* do sistema.

Figura 5 – Tela Home, página principal da aplicação



Fonte: do Autor

Na tela da página Sobre Nós é onde contém as informações sobre a entidade que gerencia o sistema, informações como missão, visão, valores e sobre o projeto Cão sem casa. No caso desta aplicação as informações contidas nesta página são fictícias, como pode ser visto na Figura 6.

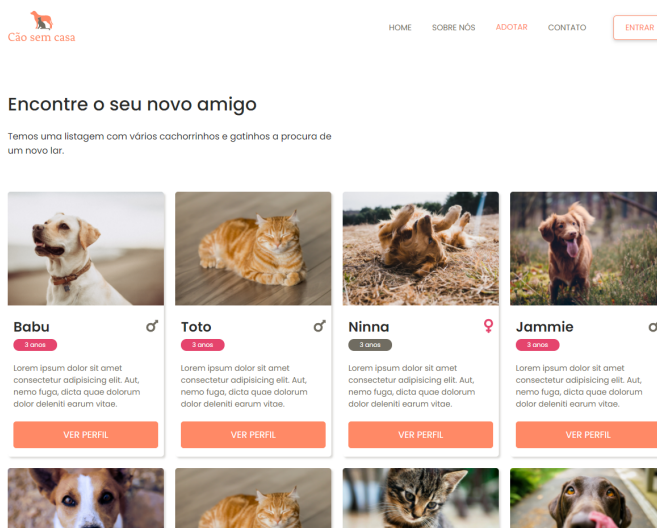
Figura 6 – Tela da página Sobre Nós



Fonte: do Autor

Na Figura 7 é mostrado a página com os animais cadastrados no sistema, o usuário pode navegar pela lista buscando por um animal para adotá-lo.

Figura 7 – Tela da página Adotar



Fonte: do Autor

Ao navegar pela lista o usuário pode clicar no botão ver perfil, para ver o perfil de um animal específico, ele será direcionado para uma nova página, conforme

mostrado na Figura 8 e nela poderá ver com mais detalhes as informações sobre o animal.

Figura 8 – Tela da página Perfil do Animal



Fonte: do Autor

Caso deseje adotar esse animal clicando no botão adotar, será exibido na mesma tela da imagem acima, um formulário para o usuário poder confirmar a adoção, mas isso só será possível se ele estiver logado, se não estiver ele será redirecionado para a tela de *login*.

A Figura 9 mostra um caso onde o usuário já esteja logado e deseja adotar um animal.

Figura 9 – Tela da página Perfil do Animal com formulário de adoção



Fonte: do Autor

O formulário já vem previamente preenchido com as informações do usuário logado, este tendo somente que conferir os dados e confirmar a adoção.

Depois de confirmar a adoção, uma nova tela com uma mensagem de sucesso será exibida ao usuário, com informações sobre o email que será enviado conforme é ilustrado na Figura 10.

Figura 10 – Tela da página Sucesso



Fonte: do Autor

O usuário que deseja se cadastrar no sistema, deverá clicar no botão cadastre-se onde será redirecionado para a página de cadastro, exibida na Figura 11.

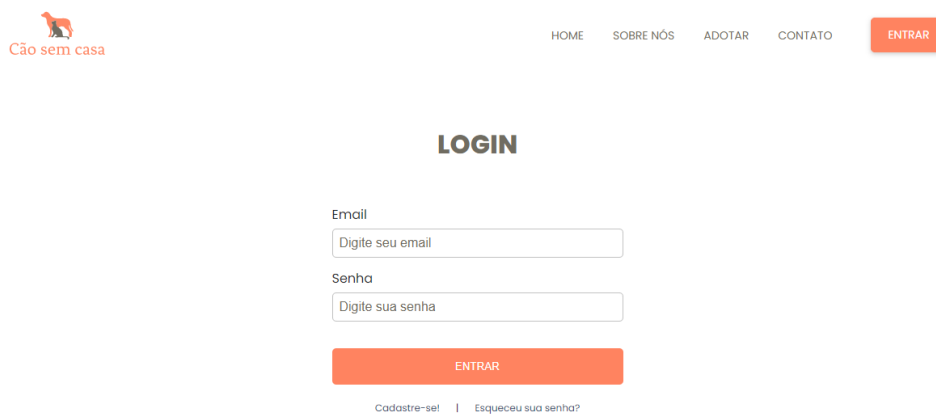
Figura 11 – Tela Cadastro

A imagem mostra a interface de usuário da tela de cadastro. No topo, há o logo "Cão sem casa" à esquerda e um menu de navegação com os itens "HOME", "SOBRE NÓS", "ADOTAR", "CONTATO" e um botão "ENTRAR" à direita. O conteúdo principal é centralizado e começa com o título "CRIAR UMA CONTA". Abaixo dele, há um formulário com os seguintes campos: "Nome" (com o placeholder "Digite seu nome"), "Email" (com o placeholder "Digite seu email"), "Rua" (com o placeholder "Digite o nome da rua") e "Número" (com o placeholder "Número"), "Bairro" (com o placeholder "Digite o nome do bairro"), "Cidade" (com o placeholder "Digite o nome da cidade") e "UF" (com o placeholder "ex: MG"), e "Senha" (com o placeholder "Digite sua senha"). Na base do formulário, há um botão "CADASTRAR" em laranja. Abaixo do botão, há o link "Já tem uma conta? Acesse!".

Fonte: do Autor

Realizado o cadastro o usuário será redirecionado para a página de *Login* onde poderá realizar a operação de *login* no sistema. A Figura 12 mostra a página de *login* do sistema.

Figura 12 – Tela Login

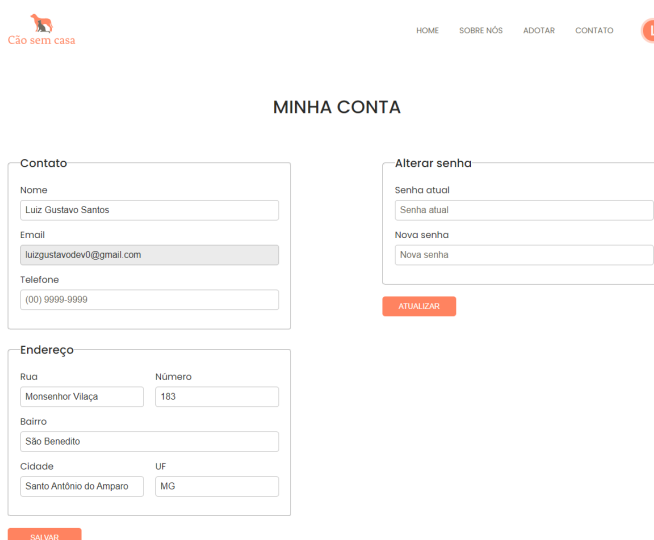


The screenshot shows the login page for 'Cão sem casa'. At the top left is the logo with a dog icon and the text 'Cão sem casa'. To the right is a navigation menu with links: HOME, SOBRE NÓS, ADOPTAR, CONTATO, and an orange 'ENTRAR' button. The main heading is 'LOGIN'. Below it are two input fields: 'Email' with the placeholder 'Digite seu email' and 'Senha' with the placeholder 'Digite sua senha'. A large orange 'ENTRAR' button is centered below the fields. At the bottom, there are two links: 'Cadastre-se!' and 'Esqueceu sua senha?'.

Fonte: do Autor

Caso faça *login* na aplicação o usuário terá acesso a página minha conta conforme apresentada na Figura 13, esta contendo os dados de cadastro do mesmo, podendo editar seus dados conforme necessário, não sendo possível editar o email.

Figura 13 – Tela da página Minha Conta



The screenshot shows the 'Minha Conta' page for 'Cão sem casa'. At the top left is the logo. To the right is a navigation menu with links: HOME, SOBRE NÓS, ADOPTAR, CONTATO, and a profile icon 'L'. The main heading is 'MINHA CONTA'. There are two main sections: 'Contato' and 'Endereço'. The 'Contato' section has fields for 'Nome' (Luiz Gustavo Santos), 'Email' (luizgustavodev0@gmail.com), and 'Telefone' ((00) 9999-9999). The 'Endereço' section has fields for 'Rua' (Monsenhor Vilaça), 'Número' (183), 'Bairro' (São Benedito), 'Cidade' (Santo Antônio do Amparo), and 'UF' (MG). There is an orange 'SALVAR' button below the address section. To the right of the 'Contato' section is the 'Alterar senha' section with fields for 'Senha atual', 'Nova senha', and 'Nova senha' (repeated), and an orange 'ATUALIZAR' button below it.

Fonte: do Autor

Ainda na página minha conta, o usuário também poderá excluir sua conta no sistema bastando clicar no botão remover conta localizado abaixo dos formulários de edição, fazendo isso todos seus dados serão removidos do sistema, não sendo possível recuperá-los.

5.2 INTERFACE DO ADMINISTRADOR

O sistema conta também com uma área administrativa para o gerenciamento dos animais, usuários e as adoções, sendo seu acesso restrito somente ao administrador do sistema.

A Figura 14 mostra a tela de login do *dashboard*, sendo esta a página inicial do sistema.

Figura 14 – Tela Login Dashboard

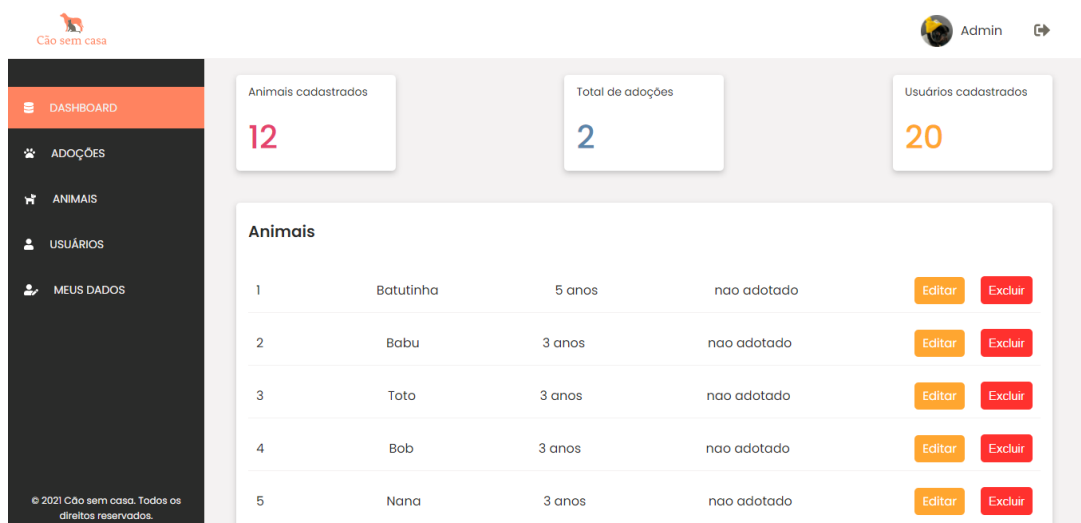


A imagem mostra a interface de login do sistema. No topo, há um ícone de um cão laranja e um gato preto, com o texto "Cão sem casa" em laranja. Abaixo, há dois campos de entrada: "Email" com o placeholder "Digite seu email" e "Senha" com o placeholder "Digite sua senha". Abaixo dos campos, há um botão laranja com o texto "ENTRAR".

Fonte: do Autor

Ao realizar o login no sistema o administrador tem acesso a página principal do *dashboard*, Figura 15, podendo navegar entre as demais telas através do menu lateral da página.

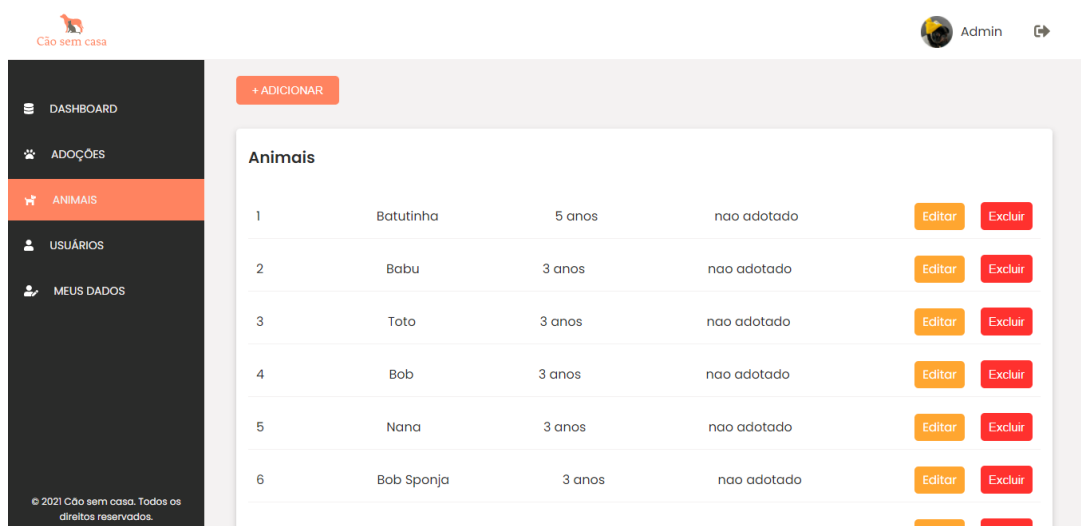
Figura 15 – Tela Principal do Dashboard



Fonte: do Autor

As Figuras 16, 17 e 18 mostram as telas relacionadas ao gerenciamento dos animais.

Figura 16 – Tela de listagem dos animais



Fonte: do Autor

Figura 17 – Formulário de cadastro de animais

Nome
Nome do animal

Peso
0

Idade
0

Raça
Raça do animal

Espécie
Ex: cachorro

Castrado
Digite sim ou não

Vermifugado
Digite sim ou não

Vacinado
Digite sim ou não

Sexo
Selecione

Foto
Escolher arquivo Nenhum arquivo selecionado

CADASTRAR

© 2021 Cão sem casa. Todos os direitos reservados.

Fonte: do Autor

Figura 18 – Formulário de edição dos dados do animal

Nome
Batutinha

Raça
vira-lata

Espécie
cachorro

Status
não adotado

Peso
3,00

Idade
5

Sexo
Macho

Castrado
sim

Vermifugado
sim

Vacinado
sim

Descrição

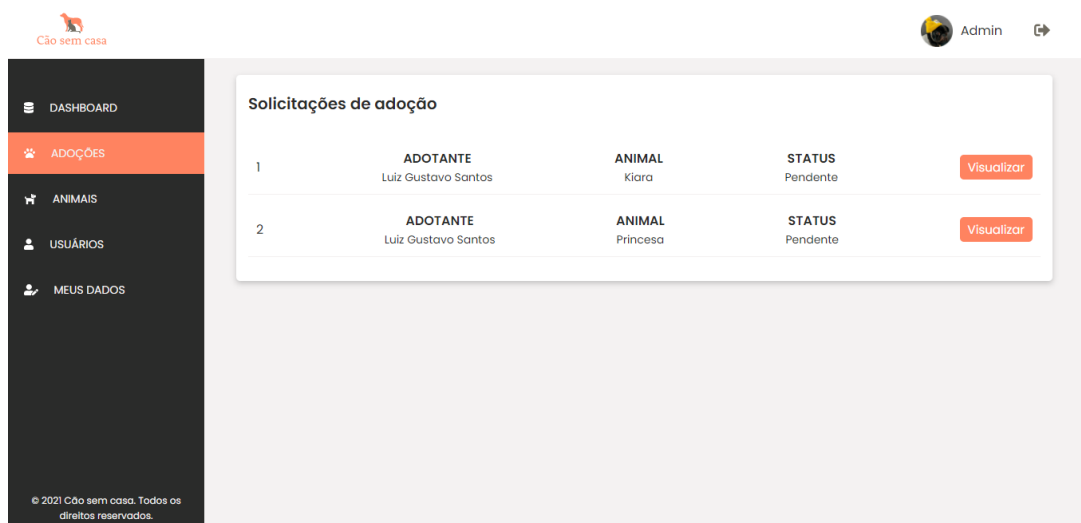
ATUALIZAR

© 2021 Cão sem casa. Todos os direitos reservados.

Fonte: do Autor

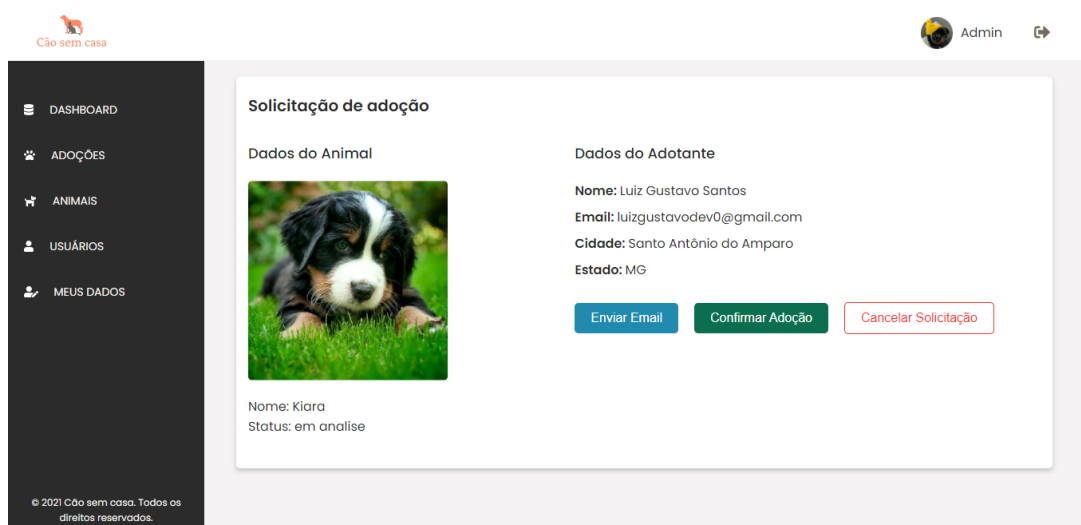
As Figuras 19 e 20 apresentam as telas de adoção e de confirmação de adoção.

Figura 19 – Tela com as solicitações de adoção



Fonte: do Autor

Figura 20 – Tela para confirmação da adoção

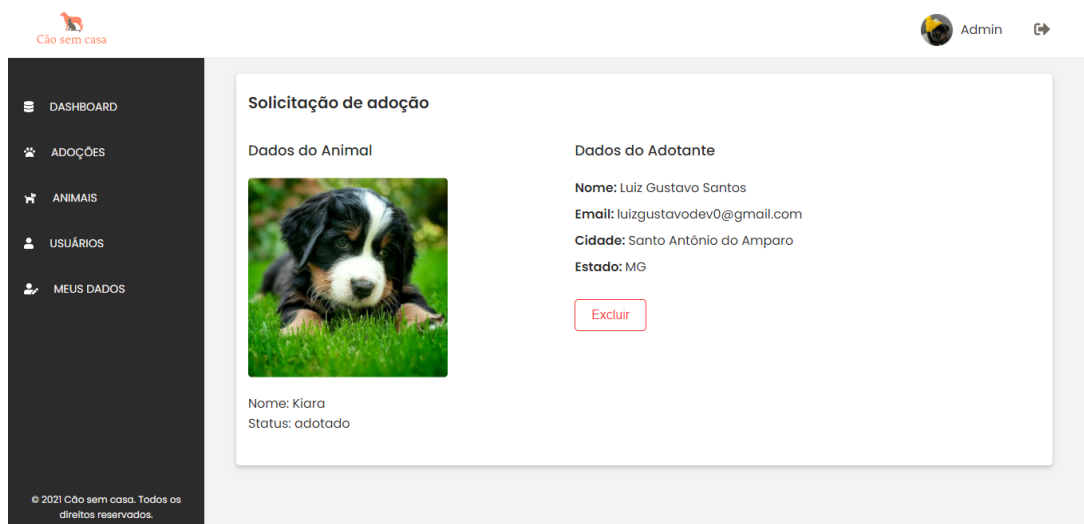


Fonte: do Autor

A Figura 20 mostra os dados relacionados à solicitação de adoção de um animal, quando é feita a solicitação o status do animal passa de não adotado para em análise como pode ser observado na Figura 20. O administrador enviará um email para o adotante conforme descrito na Figura 10, com as informações necessárias para o mesmo comparecer ao abrigo e realizar uma entrevista para validação dos dados do adotante e da adoção.

A Figura 21 mostra o resultado final quando uma adoção é concluída, alterando o status do animal de em análise para adotado.

Figura 21 – Tela final adoção



Fonte: do Autor

6 CONCLUSÃO

Neste trabalho, foi apresentado e desenvolvido um sistema para adoção de animais, juntamente com um *dashboard* para administração.

O projeto apresentado é capaz de cadastrar usuários e animais, além de realizar adoções e outras funcionalidades. Possui uma interface leve e intuitiva, com elementos dispostos de modo a facilitar a utilização do sistema tanto pelos adotantes quanto pelo administrador.

Como trabalhos futuros o sistema poderá fornecer um aplicativo para dispositivos móveis, visto que não seria necessário desenvolver outra API já que poderia utilizar a mesma API desenvolvida para este sistema. Isso facilitaria ainda mais o uso e também ajudaria na divulgação do sistema. Além disso a adição de novas funcionalidades ao *dashboard* para o melhor gerenciamento do sistema e também para a automação de alguns processos.

Apesar da complexidade do projeto desenvolvido dada a utilização dos *frameworks* utilizados, o projeto contribui na área de aplicações voltadas para Web. Além de que, este trabalho apresentou todo o poder que tecnologias JavaScript oferecem para o desenvolvimento de aplicações em geral, sejam elas para *desktops*, *web* ou até mesmo dispositivos móveis.

REFERÊNCIAS

- ABRIGO FLORA E FAUNA. **Como adotar?** 2020. Disponível em: <<https://www.abrigofloraefauna.org.br/como-adotar>>. Acesso em: 26 Mar 2021.
- AXIOS. **Promise based HTTP client for the browser and node.js.** 2021. Disponível em: <<https://github.com/axios/axios>>. Acesso em: 21 Mar 2021.
- BANKS, A.; PORCELLO, E. **Learning React: Functional Web Development with React and Redux.** [S.l.]: O'Reilly Media, 2017.
- BATISTA, R. **Tudo o que você precisa saber antes de adotar um animal de estimação.** 2018. Disponível em: <<https://www.gazetadopovo.com.br/viver-bem/animal/adocao-consciente-de-animais-de-estimacao>>. Acesso em: 26 Mar 2021.
- DATE, C. J. **Introdução a sistemas de banco de dados.** [S.l.]: Elsevier, 2004.
- EXPRESSJS. **Cors.** 2021. Disponível em: <<https://expressjs.com/en/resources/middleware/cors.html>>. Acesso em: 21 Mar 2021.
- GOMES, P. C. T. **Quais os principais bancos de dados e suas diferenças?** 2019. Disponível em: <<https://www.opservices.com.br/banco-de-dados/>>. Acesso em: 21 Mar 2021.
- GOSS, B. **Node.js: por que você deve conhecer essa tecnologia?** 2017. Disponível em: <<https://www.treinaweb.com.br/blog/node-js-por-que-voce-deve-conhecer-essa-tecnologia/>>. Acesso em: 21 Mar 2021.
- HUCS, C. **Adoção de animais na quarentena.** 2020. Disponível em: <<https://jornalismorio.espm.br/geral/adocao-de-animais-na-quarentena/>>. Acesso em: 24 Mar 2021.
- INSTITUTO PET BRASIL. **País tem 3,9 milhões de animais em condição de vulnerabilidade.** 2019. Disponível em: <<http://institutopetbrasil.com/imprensa/pais-tem-39-milhoes-de-animais-em-condicao-de-vulnerabilidade/>>. Acesso em: 24 Mar 2021.
- L, A. **O Que é React e Como Funciona?** 2021. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>>. Acesso em: 21 Mar 2021.
- LARMAN, C. **Utilizando UML e padrões: Uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo.** 3. ed. [S.l.]: Bookman, 2005.
- LUCIDCHART. **Diagrama de caso de uso UML: O que é, como fazer e exemplos.** 2021. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml/#section_0>. Acesso em: 18 Jul 2021.
- MDN. **Sobre JavaScript.** 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript>. Acesso em: 20 Mar 2021.

NPM. **About npm**. 2021. Disponível em: <<https://www.npmjs.com/about>>. Acesso em: 26 Mar 2021.

ORACLE. **O Que É um Banco de Dados?** 2021. Disponível em: <<https://www.oracle.com/br/database/what-is-database/#database-software>>. Acesso em: 21 Mar 2021.

ORLANDI, C. **Configurando o ORM Sequelize no NodeJS com ExpressJS**. 2018. Disponível em: <<https://blog.rocketseat.com.br/nodejs-express-sequelize/>>. Acesso em: 21 Mar 2021.

PEREIRA, C. R. **Construindo APIs REST com Node.js**. [S.l.]: Casa do Código, 2016.

PFUTZENREUTER, E. **Node.js: ferramenta e filosofia**. 2021. Disponível em: <<https://epxx.co/artigos/nodejs1.html>>. Acesso em: 21 Mar 2021.

POSTGRESQL. **About**. 2021. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 21 Mar 2021.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. [S.l.]: McGraw-Hill Education, 2009.

REACTJS. **React - A JavaScript library for building user interfaces**. 2021. Disponível em: <<https://reactjs.org/>>. Acesso em: 21 Mar 2021.

RED HAT. **O que é API REST?** 2021. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 21 Mar 2021.

SEQUELIZE. **Sequelize ORM**. 2021. Disponível em: <<https://sequelize.org>>. Acesso em: 21 Mar 2021.

SOUZA, I. de. **Entenda o que é Rest API e a importância dele para o site da sua empresa**. 2020. Disponível em: <<https://rockcontent.com/br/blog/rest-api/>>. Acesso em: 21 Mar 2021.