

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

KAREN FISCHBORN

**Sistema de Gerenciamento de Informações  
para a ONG Patas Dadas**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em Ciência  
da Computação

Orientador: Prof. Dr. Marcelo Pimenta

Porto Alegre  
2019

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof<sup>a</sup>. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof<sup>a</sup>. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“I am big enough to admit that I am often inspired by myself.”*

— LESLIE KNOPE

## **AGRADECIMENTOS**

Agradeço a minha família e amigos pelo apoio e carinho durante todos estes anos, especialmente ao Thales Alves Pereira pelo amor, incentivo, força e apoio incondicional. Agradeço ao meu orientador, Professor Dr. Marcelo Soares Pimenta, pela orientação e às professoras Renata de Matos Galante e Taisy Silva Weber agradeço as observações e sugestões feitas como membros da banca de avaliação.

## RESUMO

Este trabalho tem como objetivo descrever a concepção e o desenvolvimento de uma aplicação Web de gerenciamento e administração de tabelas - para uso dos voluntários da ONG Patas Dadas - referentes ao controle de materiais utilizados pela ONG, ao fluxo de caixa gerado pela venda de agendas e calendários, a organização de eventos arrecadatórios e ao controle da lista de empresas parceiras. Essa aplicação visa facilitar o gerenciamento de informações e possibilitar a consulta dos dados de forma prática, além de tornar-se uma ferramenta de controle e organização financeira. São apresentadas no texto as características principais das tecnologias adotadas e de como foram empregadas no desenvolvimento, assim como uma descrição da aplicação desenvolvida e suas funcionalidades.

**Palavras-chave:** ONG Patas Dadas. Sistema Web de gerenciamento. Voluntários.

## **Information Management System for NGO Patas Dadas**

### **RESUMO**

This paper aims to describe the design and development of a management Web application - for the use of volunteers of the non-governmental organization (NGO) Patas Dadas - using tables related to the control of materials used by the NGO, to the cash flow generated by the sale of agendas and calendars, the organization of events and to control the list of partner companies. This application aims to facilitate the information management and make it possible to consult the data in a practical way and can also become a tool to control and organize finances. In the text are presented the main characteristics of the adopted technologies and how they were used in the development, as well as a description of the application developed and its functionalities.

**Palavras-chave:** NGO Patas Dadas. Web management system. Volunteers.

## LISTA DE FIGURAS

Figura 2.1	Interface do sistema Meu Dinheiro.....	22
Figura 2.2	Interface do sistema Evernote.....	23
Figura 2.3	Interface do sistema Jira Software.....	24
Figura 3.1	Diagrama Entidade-Relacionamento.....	39
Figura 4.1	Fluxo de informações a cada interação com o usuário.....	41
Figura 4.2	Arquivo de configuração de dependência do projeto.....	42
Figura 4.3	Estrutura do projeto.....	43
Figura 4.4	Funções serializeUser() e deserializeUser().....	44
Figura 4.5	Estratégia de autenticação.....	44
Figura 4.6	Sessão da aplicação.....	45
Figura 4.7	Rota de redirecionamento para login e cadastro.....	45
Figura 4.8	Criação da rota.....	46
Figura 4.9	Chamada de cada rota utilizada no projeto.....	46
Figura 4.10	Exemplo de edição dos dados na tabela dos colaboradores.....	47
Figura 4.11	Teste e2e de adição na tabela dos colaboradores.....	48
Figura 5.1	Tela de login.....	50
Figura 5.2	Tela de cadastro.....	51
Figura 5.3	Tela após o cadastramento de um novo voluntário.....	51
Figura 5.4	Cabeçalho de navegação.....	51
Figura 5.5	Tela do perfil.....	52
Figura 5.6	Tela da tabela de colaboradores sem a inserção de dados.....	53
Figura 5.7	Tela da adição de novos colaboradores.....	53
Figura 5.8	Tela da tabela de colaboradores após a inserção dos dados.....	54
Figura 5.9	Tela da edição da lista de colaboradores.....	54
Figura 5.10	Tela da tabela de colaboradores após a edição dos dados.....	55
Figura 5.11	Tela da tabela de colaboradores após a deleção dos dados.....	55
Figura 5.12	Tela da tabela de coleta sem a inserção de dados.....	56
Figura 5.13	Tela da adição de novos itens a serem coletados.....	56
Figura 5.14	Tela da tabela de coleta após a inserção dos dados.....	57
Figura 5.15	Tela da edição da tabela de pontos de coletas.....	57
Figura 5.16	Tela da tabela de coleta após a edição dos dados.....	58
Figura 5.17	Tela da tabela de coleta após a deleção dos dados.....	58
Figura 5.18	Tela da tabela de estoque sem a inserção de dados.....	59
Figura 5.19	Tela da adição de novos itens em estoque.....	59
Figura 5.20	Tela da tabela de estoque após a inserção dos dados.....	60
Figura 5.21	Tela da edição dos itens em estoque.....	60
Figura 5.22	Tela da tabela de estoque após a edição dos dados.....	61
Figura 5.23	Tela da tabela de estoque após a deleção dos dados.....	61
Figura 5.24	Tela da tabela de eventos sem a inserção de dados.....	62
Figura 5.25	Tela da adição de novos eventos.....	62
Figura 5.26	Tela da tabela de eventos após a inserção dos dados.....	63
Figura 5.27	Tela da edição dos eventos.....	63
Figura 5.28	Tela da tabela de eventos após a edição dos dados.....	64
Figura 5.29	Tela da tabela de eventos após o cálculo das vendas dos produtos.....	64
Figura 5.30	Tela da tabela de eventos após a deleção dos dados.....	65

Figura 5.31	Tela da tabela dos pontos de venda sem a inserção de dados.....	65
Figura 5.32	Tela de adição dos preços das agendas e dos calendários. ....	66
Figura 5.33	Tela da adição de novos pontos de venda. ....	66
Figura 5.34	Tela das tabelas após a inserção dos dados.....	67
Figura 5.35	Tela da edição dos pontos de venda.....	67
Figura 5.36	Tela da tabela de pontos de venda após a edição dos dados.....	68
Figura 5.37	Tela da tabela de pontos de venda após o cálculo das vendas dos produtos.	68
Figura 5.38	Tela da tabela de pontos de venda após a deleção dos dados. ....	69
Figura 5.39	Tela da edição do valor dos preços. ....	69
Figura 5.40	Tela da tabela de pontos de venda após a alteração do valor dos preços.....	70
Figura 5.41	Tela da deleção do valor dos preços. ....	70

## LISTA DE TABELAS

Tabela 2.1	Comparativo de funcionalidades de ambas aplicações.....	25
Tabela 6.1	Tabela de análise das heurísticas de Nielsen. ....	71
Tabela 6.2	Tabela do plano de coleta de dados. ....	72
Tabela 7.1	Tabela de análise comparativa completa. ....	74

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DOM	Document Object Model
E2E	End-to-End
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model-View-Controller
NPM	Node Package Manager
ONG	Organização Não Governamental
RNF	Requisito Não-Funcional
SQL	Structured Query Language
SPA	Single-Page Application
UFRGS	Universidade Federal do Rio Grande do Sul
URL	Uniform Resource Locator

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
<b>1.1 Objetivo</b> .....	<b>14</b>
<b>1.2 Estrutura do Trabalho</b> .....	<b>14</b>
<b>2 FUNDAMENTOS, APLICAÇÕES E TRABALHOS RELACIONADOS</b> .....	<b>15</b>
<b>2.1 Conceitos e Tecnologias Utilizadas</b> .....	<b>15</b>
2.1.1 Metodologia Ágil .....	15
2.1.2 Protótipo.....	16
2.1.3 Front-End / Back-End.....	16
2.1.4 HTML, CSS e DOM.....	16
2.1.5 JavaScript.....	17
2.1.6 JQuery .....	17
2.1.7 Bootstrap.....	17
2.1.8 Node.js .....	18
2.1.9 NPM.....	18
2.1.10 Express.....	18
2.1.11 Bcrypt.....	19
2.1.12 Passport.....	19
2.1.13 Nodemon.....	19
2.1.14 CommonJS.....	19
2.1.15 Nightwatch.....	19
2.1.16 MySQL .....	19
2.1.17 MVC .....	20
2.1.18 Heurísticas de Nielsen.....	20
<b>2.2 Aplicações e Trabalhos Relacionados</b> .....	<b>21</b>
2.2.1 Meu Dinheiro .....	22
2.2.2 Evernote .....	22
2.2.3 Jira Software .....	23
2.2.4 Comparativo de Funcionalidades.....	24
<b>3 PROJETO DA APLICAÇÃO</b> .....	<b>26</b>
<b>3.1 Dificuldades</b> .....	<b>26</b>
<b>3.2 Requisitos Funcionais</b> .....	<b>26</b>
<b>3.3 Requisitos Não Funcionais</b> .....	<b>27</b>
3.3.1 Requisitos de Produtos.....	27
3.3.1.1 Requisitos de Usabilidade.....	27
3.3.1.2 Requisitos de Confiabilidade .....	28
3.3.1.3 Requisitos de Portabilidade.....	28
3.3.1.4 Requisitos de Segurança .....	28
3.3.2 Requisitos Organizacionais.....	28
3.3.2.1 Requisitos de Implementação .....	28
3.3.3 Requisitos Externos .....	29
3.3.3.1 Requisitos Éticos.....	29
<b>3.4 Histórias de Usuário</b> .....	<b>29</b>
3.4.1 Histórias do Voluntário .....	29
<b>3.5 Entidades</b> .....	<b>32</b>
3.5.1 Users .....	32
3.5.2 Colaboradores .....	33
3.5.3 Coleta .....	34
3.5.4 Estoque.....	34

3.5.5	Eventos.....	35
3.5.6	Vendas.....	36
3.5.7	Preços.....	37
3.5.8	Soma_eventos.....	37
3.5.9	Soma_vendas.....	38
<b>4</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO.....</b>	<b>40</b>
<b>4.1</b>	<b>Arquitetura da Aplicação.....</b>	<b>40</b>
<b>4.2</b>	<b>Configuração e Criação do Projeto.....</b>	<b>41</b>
<b>4.3</b>	<b>Estrutura do Projeto.....</b>	<b>42</b>
<b>4.4</b>	<b>Autenticação.....</b>	<b>43</b>
<b>4.5</b>	<b>Rotas.....</b>	<b>46</b>
4.5.1	Funções da Rota.....	46
4.5.2	Chamada das Rotas.....	46
4.5.3	CRUD.....	47
<b>4.6</b>	<b>Banco de Dados.....</b>	<b>47</b>
<b>4.7</b>	<b>Testes.....</b>	<b>48</b>
<b>4.8</b>	<b>Versão Final.....</b>	<b>49</b>
<b>5</b>	<b>GUIA DE USO DO SISTEMA.....</b>	<b>50</b>
<b>5.1</b>	<b>Autenticação.....</b>	<b>50</b>
<b>5.2</b>	<b>Cadastro.....</b>	<b>50</b>
<b>5.3</b>	<b>Navbar.....</b>	<b>51</b>
<b>5.4</b>	<b>Perfil.....</b>	<b>52</b>
<b>5.5</b>	<b>Tabelas.....</b>	<b>52</b>
5.5.1	Colaboradores.....	53
5.5.2	Pontos de Coleta.....	55
5.5.3	Estoque.....	58
5.5.4	Eventos.....	61
5.5.5	Pontos de Venda.....	65
<b>6</b>	<b>ANÁLISE DA APLICAÇÃO.....</b>	<b>71</b>
<b>6.1</b>	<b>Avaliação Heurística.....</b>	<b>71</b>
<b>6.2</b>	<b>Pesquisa Qualitativa.....</b>	<b>72</b>
<b>7</b>	<b>CONCLUSÃO.....</b>	<b>74</b>
<b>7.1</b>	<b>Aspectos Técnicos.....</b>	<b>74</b>
<b>7.2</b>	<b>Limitações do Trabalho.....</b>	<b>75</b>
<b>7.3</b>	<b>Trabalhos Futuros.....</b>	<b>75</b>
	<b>REFERÊNCIAS.....</b>	<b>76</b>
	<b>APÊNDICE A — ROTEIRO DA ENTREVISTA.....</b>	<b>79</b>

## 1 INTRODUÇÃO

No Brasil, segundo a OMS (Organização Mundial da Saúde), existem mais de 30 milhões de animais que vivem nas ruas. Em algumas cidades o poder público tem atuado para proteger esses animais, mas mesmo esses esforços não são suficientes e as ONGs acabam desempenhando um papel importante de auxílio nesse trabalho, porém quando essa política não é praticada pelo governo elas passam a ser a única esperança para os animais abandonados.

Os animais acolhidos por essas instituições exigem cuidados especiais, como a castração, aplicação de vacinas, vermífugos e controle da alimentação que são realizados por voluntários. Entre as ONGs que ajudam todos os dias a salvar animais está o Patas Dadas, que surgiu em resposta a uma série de maus tratos aos animais abandonados no Campus do Vale da UFRGS. O Patas Dadas tem como missão resgatar animais em situação de abandono, proporcionando o atendimento veterinário necessário até estarem prontos para a adoção, buscando conscientizar a população da importância da adoção responsável.

O Patas Dadas é composto por 54 voluntários que realizam diversas tarefas, como, por exemplo, banho, tosa, vacinação e tratamento de doenças e machucados nos animais acolhidos pela ONG. Afora isso, administrativamente o Patas Dadas trabalha com a coleta de doações, venda de calendários e agendas, organização de eventos e estoque de materiais utilizados pela ONG. Além disso, também conta com o auxílio de colaboradores, empresas que são responsáveis por servir como ponto de referência para a coleta de doações e venda das agendas e dos calendários, além de participar dos eventos promovidos pela ONG.

Devido a grande quantidade de voluntários na ONG e a ausência de um sistema regulatório confiável surgiram alguns complicadores na gestão dos recursos relativos a coleta e estoque de materiais doados e dos recursos financeiros relativos a venda dos produtos (calendários e agendas) do Patas Dadas. O Excel é a principal ferramenta utilizada para o armazenamento e organização da informações, resultando na dificuldade de centralizar os dados em um mesmo lugar ou em uma mesma planilha devido a diversidade de informações existentes, além de não haver registros de alterações armazenadas no sistema.

Afim de oferecer uma ferramenta que auxilie o Patas Dadas, a autora deste trabalho se propõe a desenvolver uma ferramenta de organização e gerência de informações relativas aos pontos de venda de agendas e calendários, de coleta e estoque de materiais utilizados pela ONG, dos eventos organizados para arrecadar fundos e uma lista de colaboradores

que auxiliam o Patas Dadas.

## **1.1 Objetivo**

O objetivo deste trabalho é o projeto e o desenvolvimento de uma aplicação Web para a gestão e controle de dados para a ONG Patas Dadas. Com esse projeto os voluntários poderão ter um controle dos colaboradores que auxiliam a ONG, além disso, os voluntários poderão administrar os materiais presentes nos pontos de coleta (empresas colaboradoras) e estoque e gerenciar o dinheiro arrecadado pela venda das agendas e dos calendários nos eventos da ONG e nos pontos de venda.

## **1.2 Estrutura do Trabalho**

Este trabalho está dividido em 7 capítulos. Após esta introdução, onde foi abordada a contextualização e a motivação do trabalho, apresentando os objetivos e estrutura do texto, o capítulo 2 resume as tecnologias utilizadas neste trabalho, aplicações e trabalhos relacionados que se assemelham com este, comparando-os e descrevendo os diferenciais entre as aplicações. O capítulo 3 explica o projeto da aplicação, descrevendo suas funcionalidades. O capítulo 4 fala sobre o desenvolvimento da aplicação, detalhando as abordagens adotadas e os detalhes da implementação. O capítulo 5 apresenta o uso da aplicação e suas funcionalidades. O capítulo 6 apresenta as formas de avaliação da aplicação e a análise dos dados obtidos. Finalmente, o capítulo 7 apresenta as conclusões, retrospectivas, limitações e resultados obtidos, além de relacionar outras funcionalidades que podem ser adicionadas em versões futuras do projeto.

## 2 FUNDAMENTOS, APLICAÇÕES E TRABALHOS RELACIONADOS

Neste capítulo são abordados dois temas: o primeiro é uma base teórica e conjunto de fundamentos e tecnologias utilizados para o desenvolvimento deste trabalho. O segundo consiste numa descrição das aplicações existentes que oferecem soluções (similares) para a organização de dados e informações necessárias para um bom gerenciamento financeiro.

### 2.1 Conceitos e Tecnologias Utilizadas

Dado que o foco deste trabalho é uma aplicação Web, é fundamental que seja feita uma descrição detalhada das tecnologias utilizadas. Para tanto, serão demonstrados aqui o básico de alguns conceitos e tecnologias empregados nesta aplicação, como a separação entre Front-End e Back-End, Javascript, Bootstrap e MySQL. Para a realização desse trabalho somente software livres foram utilizados.

#### 2.1.1 Metodologia Ágil

Para o desenvolvimento do software deste projeto foram utilizados conceitos das metodologias ágeis, que são um conjunto de técnicas de desenvolvimento de software que têm o objetivo de acelerar os processos, visando uma melhoria contínua e incremental, de modo a entregar parcelas funcionais do projeto aos usuários, possibilitando uma rápida resposta quando da necessidade de alterações (FOWLER, 2015).

Com essa metodologia ocorre a subdivisão dos objetivos em metas menores com janelas de tempo para sua conclusão. Essas janelas de tempo recebem o nome de *sprints* e tem o objetivo de, além de facilitar o acompanhamento das metas, propiciar a entrega parcial de etapas do projeto ao usuário. Cada *sprint* é uma iteração do projeto e antes de ser iniciada é realizado um planejamento prevendo todas as ações a serem realizadas durante sua execução. Da mesma maneira ao final de cada iteração é feita uma retrospectiva de avaliação.

Desta maneira, uma iteração sempre começa com suas dependências satisfeitas pela iteração anterior, caso contrário, as tarefas desta devem ser adiadas para uma iteração futura, e novas tarefas para a iteração atual devem ser definidas. Com isso, garante-se que cada funcionalidade a ser implementada passe por etapas de planejamento, análise de

requisitos, projeto, codificação e testes dentro das iterações (HELABS, 2013/2014).

### **2.1.2 Protótipo**

A ferramenta utilizada para fazer a prototipação do sistema foi a *Proto.io* (PROTO. IO, 2019). Ela é uma ferramenta que permite a criação de protótipos de aplicativos para diversas plataformas. Com a Proto.io já é possível simular praticamente todo o funcionamento das ferramentas além de criar o design da aplicação pretendida.

### **2.1.3 Front-End / Back-End**

O Front-End de uma aplicação é o conjunto de módulos que fazem a captura de dados inseridos por um usuário para serem processados e módulos que transmitem uma informação processada ou armazenada de volta para o usuário. Enquanto que o Back-End é o conjunto de módulos que recebem dados enviados por um usuário, os processa, armazena ou não, e envia para o usuário feedback de suas ações ou respostas aos seus pedidos.

Esta separação também é conhecida como Client-Side / Server-Side (TABLELESS, 2012), em que se define que uma aplicação é Client-Side, se sua execução e processamentos ocorrem no navegador do usuário, ou Server-Side, caso haja um servidor responsável por executar os pedidos de um usuário para posteriormente lhes enviar a resposta desejada.

### **2.1.4 HTML, CSS e DOM**

HTML e CSS são as ferramentas básicas para gerar páginas Web estáticas. A linguagem de marcação HTML é reconhecida por todos os navegadores modernos, enquanto que o CSS entra no conjunto das ferramentas suportadas pelos navegadores modernos agindo como um conjunto de regras de estilo para certos componentes. Com tais regras é possível alterar totalmente a aparência de um documento, tornando a aplicação mais agradável para o usuário final.

O código HTML, agregado aos estilos do CSS, é compilado e transformado no DOM, que fica armazenado na memória do navegador de internet (FRANKLIN, 2011; COYIER, 2013). O armazenamento do DOM se dá pela estrutura de árvore e este é acessado por meio do DOM API, que oferece métodos para acessar um nodo específico

desta árvore para manipular seu conteúdo, estilo ou eventos relacionados. É comum que a manipulação do DOM seja feita com o auxílio de JavaScript e bibliotecas como, por exemplo, JQuery.

Neste trabalho serão utilizados HTML e CSS para a geração das páginas da aplicação. Adicionalmente, o Bootstrap (seção 2.1.7) foi utilizado para uma melhor experiência do usuário, garantindo uma aplicação responsiva e adaptativa sem duplicação de código ou manipulação direta no DOM.

### **2.1.5 JavaScript**

JavaScript (MOZILLA, 2019) é uma linguagem de programação muito utilizada na Web, inicialmente criada para ser utilizada como uma parte dos navegadores, de tal modo a executar scripts dentro do próprio navegador, sem a necessidade desses scripts serem processados do lado do servidor. A partir da utilização do Node.js tornou-se possível escrever aplicações totalmente em JavaScript, bastando apenas submeter os scripts para o Node.js, para que ele interprete os comandos e os converta em linguagem de máquina, permitindo, portanto, o controle de recursos do sistema operacional, ou seja, com apenas uma linguagem de programação, no caso o JavaScript, é possível criar tanto scripts para o Front-End quanto scripts para o Back-End.

### **2.1.6 JQuery**

JQuery é uma biblioteca que simplifica a interação do JavaScript com o HTML. É rápida, leve e possui muitas funcionalidades que tornam a manipulação de dados, tratamento de eventos, animações e chamadas AJAX mais simples, utilizando uma API compatível com a maioria dos navegadores modernos (FOUNDATION, 2019). Hoje em dia está altamente difundida e grande parte dos programadores criam os códigos JavaScript através da JQuery ao invés do JavaScript puro.

### **2.1.7 Bootstrap**

Bootstrap é um framework de desenvolvimento Web Front-End criado no Twitter em 2010 que oferece uma série de componentes e comportamentos desenvolvidos com

jQuery (BOOTSTRAP, 2019). Este framework contribuiu para o desenvolvimento da aplicação por oferecer uma série de novas diretivas HTML para diversos componentes como barras de navegação ou seletores de datas e também oferecer aparência pré-fabricada para basicamente todas as diretivas padrão do HTML.

### **2.1.8 Node.js**

Node.js é um interpretador de códigos JavaScript, sua construção foi baseada no V8 Javascript engine do Google Chrome (NODE.JS, 2019). Do ponto de vista do desenvolvimento Web, o Node.js implementa diversos recursos que o torna capaz de responder a diversos tipos de protocolos diferentes, dentre esses protocolos o HTTP, isso significa que apesar do Node.js não ser um servidor HTTP propriamente dito, é possível implementar os recursos necessários para que a aplicação seja capaz de responder a requisições feitas a partir desse protocolo. Com o Node.js é possível trazer a linguagem de programação JavaScript para o Back-End.

### **2.1.9 NPM**

NPM (NPM, 2019) é um gerenciador de pacotes JavaScript, com ele é possível acrescentar módulos ao sistema, aplicação, de maneira muito simples.

### **2.1.10 Express**

Express (EXPRESS, 2019) é um framework Node.js para aplicações Web, ou seja, ele implementa uma API e diversos métodos para manipular de forma mais fácil funções do Node.js para, por exemplo, controle de rotas ou controle dos middlewares dos módulos que estão entre a requisição e a resposta para o browser. O Express fornece diversos recursos para otimizar e agilizar o desenvolvimento.

### **2.1.11 Bcrypt**

Bcrypt (HALE, 2011) foi a biblioteca escolhida para a criação do hash de senha de acesso do voluntário.

### **2.1.12 Passport**

Passport (HANSON, 2015) é um middleware de autenticação utilizado no Node.js.

### **2.1.13 Nodemon**

Nodemon (NODEMON, 2019) é um utilitário que reinicia automaticamente o servidor Node.js quando houver qualquer alteração nos scripts.

### **2.1.14 CommonJS**

CommonJS (COMMONJS, 2019) é um formato que define a construção dos módulos, ele é basicamente uma forma de escrita que permite que o módulo seja incorporado ao Node.js de forma mais "natural".

### **2.1.15 Nightwatch**

Nightwatch (NIGHTWATCH, 2019) é uma solução completa e integrada para testes *end-to-end* de aplicações Web.

### **2.1.16 MySQL**

MySQL (MYSQL, 2019) é um sistema de gerenciamento de banco de dados baseado em SQL que funciona em praticamente todas as plataformas, incluindo Linux, UNIX e Windows.

### 2.1.17 MVC

MVC é um padrão de arquitetura de software que separa a representação e a interação (por parte do usuário) dos dados. Esse tipo de padrão geralmente é usado para a criação de interfaces de usuário.

O *Modelo* é o componente central. Nele estão expressas as regras de negócios e o comportamento da aplicação em termos de domínio de problema, independente da interação com o usuário, tratando diretamente os dados e lógicas do sistema.

A *Visão* é toda a forma de interação com o usuário e visualização dos dados. As visões podem ser alteradas independentemente das regras de negócio e o mesmo conjunto de dados pode ser mostrado de diversas maneiras.

O *Controlador* é o intermediário da interação entre *Modelo* e *Visão*. Nele as entradas são tratadas e convertidas em elementos e comandos para a visualização (DEACON, 1995).

### 2.1.18 Heurísticas de Nielsen

O processo de avaliação das heurísticas de Nielsen and Molich (1990) pode ser descrito como: uma lista de diretrizes de projeto que é usada como base para avaliação de um sistema ou aplicação. Em 1993 Nielsen adaptou a lista para:

- *Visibilidade do estado do sistema:* o sistema deve sempre manter o usuário informado sobre o que está acontecendo no momento da interação.
- *Equivalência entre o sistema e o mundo real:* cada palavra, frase e conceito presentes no sistema deve ser familiar ao usuário, ou seja, não deve ser orientado a termos técnicos, deve seguir as convenções do mundo real, fazendo a informação aparecer em uma ordem natural e lógica.
- *Liberdade e controle do usuário:* nesta heurística a preocupação é de passar para o usuário a liberdade de ele fazer o que quiser dentro do sistema, com exceção das regras que vão contra o negócio ou interferem em outra funcionalidade. É necessário dar suporte para o usuário desfazer e refazer suas ações, permitindo retornar a estados anteriores.
- *Consistência e padrões:* manter a consistência visual e de linguagem. Manter padrões de interação em diferentes contextos. O usuário não deve se perguntar se

palavras, situações ou ações diferentes significam a mesma coisa.

- *Prevenção de erro*: não deixar o usuário errar sem explicar previamente o motivo do erro.
- *Reconhecer ao invés de lembrar*: torne objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte da iteração para outra. As instruções de uso do sistema devem ser visíveis ou facilmente recuperáveis sempre que apropriado.
- *Flexibilidade e eficiência de uso*: deixar uma experiência boa com o sistema desde o usuário mais leigo até o mais avançado.
- *Estética e design minimalista*: toda informação extra para o usuário pode na verdade adicionar mais uma dúvida, ou seja, deixar o layout e o conteúdo o mais simples e direto possível.
- *Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas*: as mensagens de erros tem que ser claras e próximas do conteúdo ou ação que causou o erro.
- *Ajuda e documentação*: mesmo que o sistema possa ser utilizado sem documentação, ela pode ser necessária para fornecer ajuda ao usuário. Qualquer informação desse tipo deve ser de fácil acesso e pesquisa, focada na tarefa do usuário, listar as etapas concretas a serem executadas e não ser muito extensa.

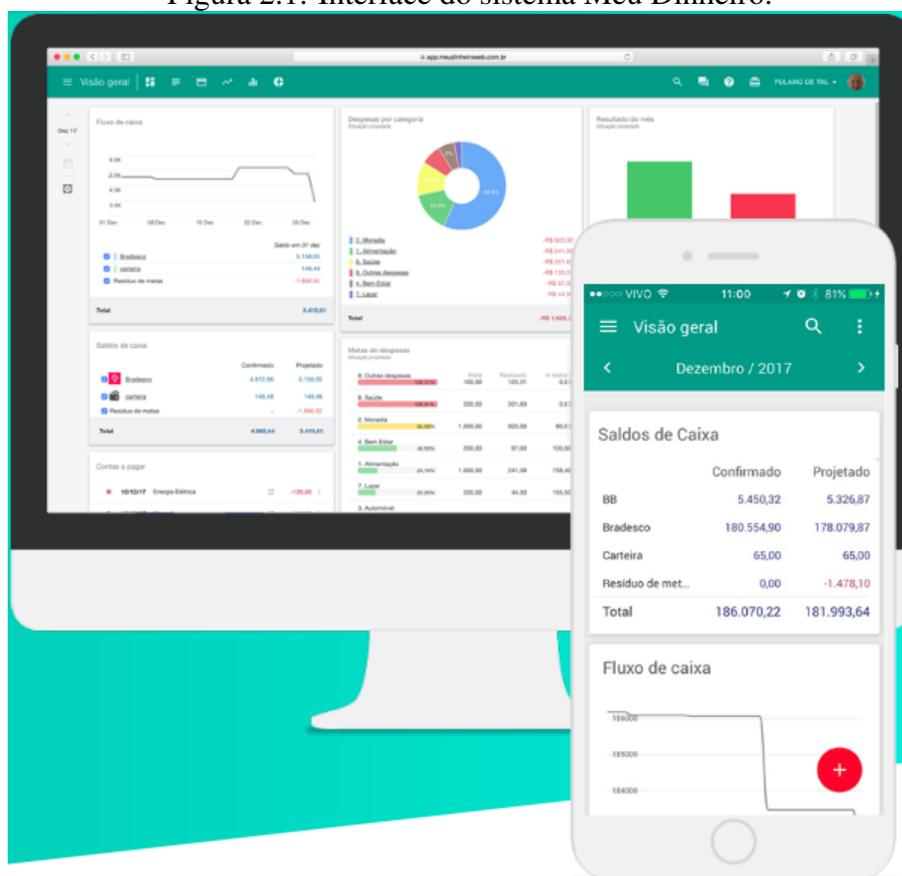
## 2.2 Aplicações e Trabalhos Relacionados

Nesta seção serão apresentados alguns programas e aplicativos encontrados no mercado que possuem objetivos relacionados ao desenvolvido deste trabalho. Sendo eles o Meu Dinheiro, o Evernote e o Jira Software. Os parâmetros utilizados para a escolha das aplicações foram o seu design e as suas funcionalidades, com ambos os parâmetros é possível analisar como os dados são dispostos em modo de exibição de listas e tabelas, mesmo tendo diferentes conteúdos. No final desse capítulo também será realizada uma análise comparativa dos sistemas apresentados.

## 2.2.1 Meu Dinheiro

O Meu Dinheiro <sup>1</sup> é um gerenciador financeiro e de investimentos online. Alguns dos recursos de controle financeiro da aplicação são os saldos de contas; fluxo de caixa; contas a pagar e a receber; despesas e receitas por categorias; cadastro de contatos de clientes e fornecedores; planilha de contas; entre outros. O gerenciador está disponível em plataforma online ou aplicativo e possui versão totalmente gratuita ou planos pessoais e empresariais.

Figura 2.1: Interface do sistema Meu Dinheiro.



## 2.2.2 Evernote

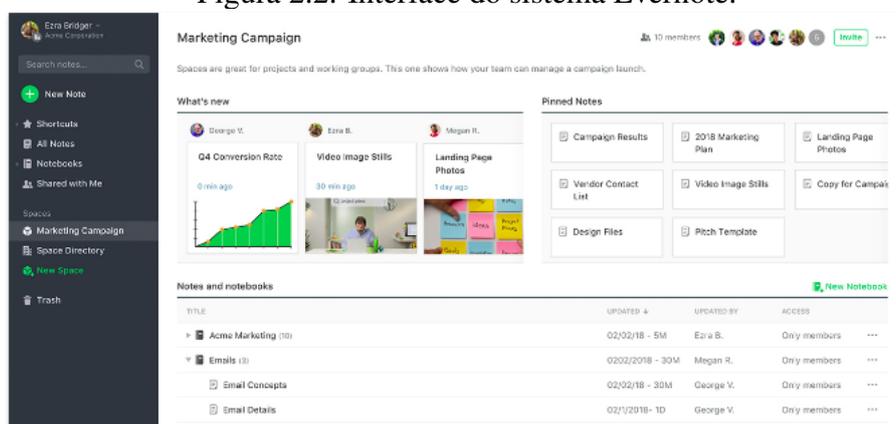
O Evernote <sup>2</sup> é uma espécie de banco de anotações, ele oferece flexibilidade para criar anotações separadas por cadernos e incluir textos formatados com cores e em lista, com anexos e desenhos feitos à mão, ainda permite adicionar datas de lembrete aos itens

<sup>1</sup><https://www.meudinheiroweb.com.br/>

<sup>2</sup><https://evernote.com/intl/pt-br/>

anotados, além de incluir etiquetas para ajudar a pesquisar mais tarde pelo conteúdo desejado. Afora isso, o Evernote funciona em diversas plataformas e realiza sincronização automática, fazendo com que as anotações fiquem acessíveis em todos os seus dispositivos. O sistema possui versão totalmente gratuita ou paga - conta *premium*, que aumenta o limite mensal de 1 GB de uploads (contra 60 MB da versão gratuita), tamanho máximo de 100 MB por nota, remoção de publicidade do programa, modo offline, suporte prioritário e pesquisa em PDFs.

Figura 2.2: Interface do sistema Evernote.

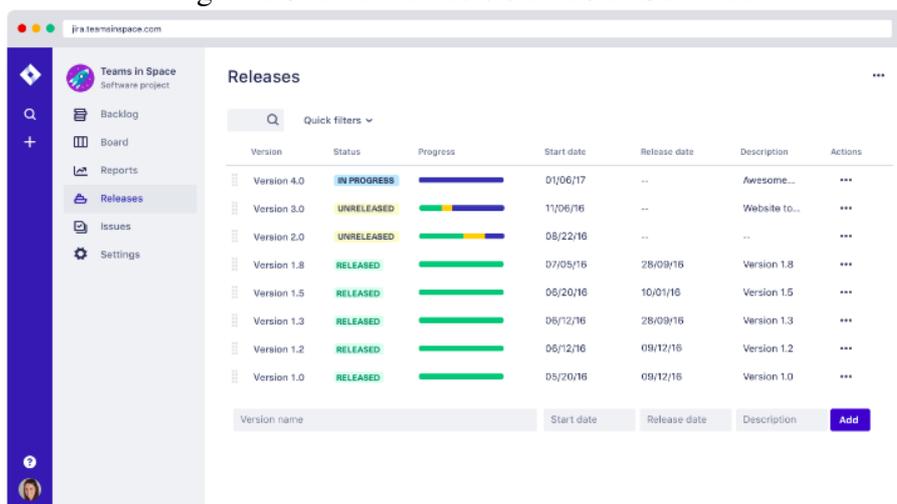


### 2.2.3 Jira Software

O Jira Software <sup>3</sup> é uma plataforma que combina recursos de coleta, de pendências e de gerenciamento de projetos ágeis em uma única aplicação, ajudando a planejar e organizar tarefas, fluxos de trabalho e relatórios. A plataforma oferece suporte a qualquer metodologia ágil, como o Scrum, Kanban, entre outras. A partir de quadros ágeis para relatórios, há a possibilidade de planejar, rastrear e gerenciar todos os projetos de desenvolvimento ágil de software a partir de uma única ferramenta. Jira possui apenas versões pagas, cujo preço varia de acordo com o número de usuários que utilizarão a plataforma.

<sup>3</sup><https://www.atlassian.com/software/jira>

Figura 2.3: Interface do sistema Jira Software.



## 2.2.4 Comparativo de Funcionalidades

Apesar de não serem aplicações equivalentes, é possível gerar um quadro comparativo de funcionalidades presentes em cada aplicação. Na tabela 2.1 é mostrado um comparativo por aplicação de cada funcionalidade. As funcionalidades foram categorizadas em três grupos:

- *Distribuição*: agrupa funcionalidades referentes ao modo como a aplicação é distribuída na rede, seja possuindo versões gratuitas ou pagas ou possuindo versões Web ou mobile.
- *Gerência*: agrupa funcionalidades de gerência dos dados, como, por exemplo, controlar o pagamento de contas e cadastro de contatos e os tipos de dados que a aplicação dá suporte.
- *Visualização*: agrupa funcionalidades de visualização entre os usuários, como ter um perfil e poder acessar a plataforma em modo offline.

Tabela 2.1: Comparativo de funcionalidades de ambas aplicações.

<b>Funcionalidades/Aplicação</b>		<i>Meu Dinheiro</i>	<i>Evernote</i>	<i>Jira Software</i>
Distribuição	Versão gratuita	✓	✓	
	Versão paga	✓	✓	✓
	Versão web	✓	✓	✓
	Versão móvel	✓	✓	
Gerência	Fluxo de caixa	✓		
	Contas a pagar	✓		
	Controle de receitas	✓		
	Anotações		✓	✓
	Gerenciamento de projetos		✓	✓
Visualização	Criação de perfil	✓	✓	✓
	Modo offline		✓	
	Grupo de usuários	✓	✓	✓

### **3 PROJETO DA APLICAÇÃO**

Neste capítulo será feita uma apresentação do projeto deste trabalho, que tem como objetivo ser uma ferramenta de organização e gerência de informações relativas aos eventos - organizados pelos voluntários do Patas Dadas - e aos pontos de venda de agendas e calendários, auxiliando no controle do fluxo de caixa. Afora isso, a aplicação também auxilia no controle de materiais disponíveis para coleta nas empresas colaboradoras e dos materiais, já em posse da ONG, disponíveis em estoque. Além disso, há a gerência de uma lista de colaboradores do Patas Dadas, em que os voluntários classificam a participação de cada empresa colaboradora.

#### **3.1 Dificuldades**

A ideia para o desenvolvimento da aplicação surgiu devido a ausência de um sistema regulatório confiável, gerando alguns complicadores na gestão dos recursos relativos a coleta e estoque de materiais doados e dos recursos financeiros relativos a venda dos produtos (calendários e agendas) do Patas Dadas.

Tendo dados relativos a quantidade de itens disponíveis em estoque, em pontos de coleta e venda de produtos e em eventos, além dos dados relativos ao dinheiro arrecadado com a venda desses produtos, as datas dos eventos, os voluntários envolvidos nas atividades, entre outros, e a falta de uma aplicação para auxiliar no gerenciamento desses dados tornou a gestão dessas informações árdua.

Um outro problema enfrentado pela ONG na administração das informações é a ausência do registro de alterações, inclusões e deleções dos dados, os voluntários não conseguem comprovar a autenticidade das informações, quando a informação foi incluída, alterada ou deletada do sistema e qual voluntário realizou a ação.

#### **3.2 Requisitos Funcionais**

Os requisitos funcionais para um sistema descrevem a funcionalidade ou os serviços que se espera que o sistema forneça. Eles dependem do tipo de software que está sendo desenvolvido, dos usuários de software que se espera verificar e do tipo de sistema que está sendo desenvolvido (SOMMERVILLE, 2004).

- **RF01:** para acessar o sistema, o usuário deverá efetuar o processo de login, utilizando um usuário e uma senha.
- **RF02:** para sair o sistema, o usuário deverá efetuar o processo de logout.
- **RF03:** o sistema deve permitir o cadastramento de novos usuários.
- **RF04:** o sistema deve permitir a alteração da senha do usuário já cadastrado.
- **RF05:** o sistema deve permitir que um voluntário veja a lista de voluntários cadastrados no sistema.
- **RF06:** o sistema deve permitir a inserção de novos dados.
- **RF07:** o sistema deve permitir a edição de um dado já existente.
- **RF08:** o sistema deve permitir a exclusão de um dado já existente.
- **RF09:** o sistema deve permitir que um usuário veja a lista de dados cadastrados na tabela.

### **3.3 Requisitos Não Funcionais**

Requisitos não funcionais são aqueles que não estão diretamente relacionados à funcionalidade de um sistema, eles abordam aspectos de qualidade importantes em sistemas de software (CHUNG, NIXON, YU, MYLOPOULOS, 2000). Para uma melhor organização da especificação e semântica do projeto do software, RNFs são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das categorias utilizadas neste projeto.

#### **3.3.1 Requisitos de Produtos**

##### *3.3.1.1 Requisitos de Usabilidade*

- O sistema deverá ser operável por usuários sem a necessidade de treinamento prévio.
- O sistema deverá fazer uso de design responsivo na implementação de suas interfaces gráficas, sendo assim, deverá se comportar adequadamente em navegadores acessados via computador, smartphone e tablet.

- O sistema deve estar disponível em português.
- Enquanto dentro de uma *text box*, as setas do teclado devem possibilitar a navegação entre dados previamente inseridos para facilitar o preenchimento da mesma.
- Todas as páginas disponibilizadas pelo site devem apresentar a logo do Patas Dadas no canto superior esquerdo da tela com tamanho 76px por 76px.
- As operações devem ser efetivamente utilizadas e consideradas úteis pelo usuário, facéis de aprender e reaprender, de fácil recordação.

#### *3.3.1.2 Requisitos de Confiabilidade*

- Disponibilidade: o software deve estar disponível 24 horas por dia, 7 dias por semana, com a possibilidade de paradas pré-programadas para manutenção.

#### *3.3.1.3 Requisitos de Portabilidade*

- O sistema deverá ser suportado pelos navegadores Google Chrome, Internet Explorer, Mozilla Firefox e Safari.

#### *3.3.1.4 Requisitos de Segurança*

- Apenas os voluntários cadastrados devem possuir acesso ao sistema.
- O sistema deve fornecer mecanismos de segurança e autenticação.
- Todas as senhas armazenadas devem ser criptografadas.

### **3.3.2 Requisitos Organizacionais**

#### *3.3.2.1 Requisitos de Implementação*

- O sistema será desenvolvido utilizando a linguagem JavaScript sob o framework Bootstrap 4.
- O sistema deve ser implementado utilizando um sistema de gerenciamento de dados relacional com licença *open source*.

### 3.3.3 Requisitos Externos

#### 3.3.3.1 Requisitos Éticos

- O sistema não apresentará aos usuários quaisquer dados de caráter privativo, tais como informações pessoais de outros usuários.

### 3.4 Histórias de Usuário

Esta seção lista as funcionalidades que foram levantadas para o sistema de gerenciamento de tabelas para a ONG Patas Dadas. Essas funcionalidades estão apresentadas em *user stories*. Uma *user story* é uma descrição curta e simples de uma funcionalidade, da perspectiva de uma pessoa que deseja tal funcionalidade, geralmente um usuário ou cliente (COHN, 2004). Geralmente, elas são escritas no seguinte padrão:

*Como um <tipo de usuário> quero <algum objetivo> para que <alguma razão>.*

O Patas Dadas conta com dois tipos de modos de atuação: os voluntários - que realizam diversas tarefas, como, por exemplo, banho, tosa, vacinação e tratamento de doenças e machucados nos animais acolhidos pela ONG; e os colaboradores - empresas que são responsáveis por servir como ponto de referência para a coleta de doações e venda das agendas e dos calendários, além de participar dos eventos promovidos pela ONG.

Neste projeto tem-se apenas um tipo de usuário, o voluntário, visto que apenas os voluntários terão acesso ao sistema. Os colaboradores não terão acesso a nenhum recurso do sistema. Também não há administradores do sistema, conforme solicitado pelos voluntários, cada usuário cadastrado no sistema terá acesso aos mesmos recursos que os outros usuários cadastrados.

#### 3.4.1 Histórias do Voluntário

O Patas Dadas, além de resgatar animais abandonados, trabalha com a coleta de doações, venda de calendários e agendas, organização de eventos e estoque de materiais utilizados pela ONG. Devido a grande quantidade de voluntários na ONG e a ausência de um sistema regulatório confiável surgiram alguns complicadores na gestão dos recursos

relativos a coleta e estoque de materiais doados e dos recursos financeiros relativos a venda dos produtos (calendários e agendas) do Patas Dadas.

Necessitando de uma ferramenta que auxilie o Patas Dadas a resolver esses problemas, os voluntários solicitaram o desenvolvimento de uma ferramenta de organização e gestão de informações relativas aos pontos de venda de agendas e calendários, de coleta e estoque de materiais utilizados pela ONG, dos eventos organizados para arrecadar fundos e uma lista de colaboradores que auxiliam o Patas Dadas.

Para resolver a ausência do registro de alterações, inclusões e deleções dos dados, em que os voluntários não conseguem comprovar a autenticidade das informações, os voluntários solicitaram a opção de visualização do relatório de movimentação dos dados realizado no prazo de 30 dias a contar da data da alteração.

Em suma, visto que não há outros usuários além do voluntário, será o voluntário o encarregado por manter as informações atualizadas, adicionando, atualizando e excluindo dados quando necessário, assim como cadastrando novos voluntários no sistema.

- *Como voluntário quero acessar o sistema com segurança para que apenas pessoas autorizadas tenham permissão de acesso ao sistema.*
- *Como voluntário quero alterar minha senha para quando eu a esquecer.*
- *Como um voluntário quero cadastrar novos voluntários para que estes possam utilizar as funcionalidades do sistema.*
- *Como um voluntário quero ter acesso a lista de usuários cadastrados no sistema para saber se esta lista está atualizada de acordo com os voluntários que estão em atividade na ONG.*
- *Como um voluntário quero poder inserir novos dados na tabela de colaboradores para manter as informações atualizadas.*
- *Como um voluntário quero poder inserir novos dados na tabela de pontos de coleta para manter as informações atualizadas.*
- *Como um voluntário quero poder inserir novos dados na tabela de estoque para manter as informações atualizadas.*
- *Como um voluntário quero poder inserir novos dados na tabela de eventos para manter as informações atualizadas.*

- *Como um voluntário quero poder inserir novos dados na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder editar os dados na tabela de colaboradores para manter as informações atualizadas.*
- *Como um voluntário quero poder editar os dados na tabela de pontos de coleta para manter as informações atualizadas.*
- *Como um voluntário quero poder editar os dados na tabela de estoque para manter as informações atualizadas.*
- *Como um voluntário quero poder editar os dados na tabela de eventos para manter as informações atualizadas.*
- *Como um voluntário quero poder editar os dados na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder deletar os dados na tabela de colaboradores para manter as informações atualizadas.*
- *Como um voluntário quero poder deletar os dados na tabela de pontos de coleta para manter as informações atualizadas.*
- *Como um voluntário quero poder deletar os dados na tabela de estoque para manter as informações atualizadas.*
- *Como um voluntário quero poder deletar os dados na tabela de eventos para manter as informações atualizadas.*
- *Como um voluntário quero poder deletar os dados na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder verificar os registros de alterações feitos na tabela de colaboradores para verificar o usuário que realizou a modificação.*
- *Como um voluntário quero poder verificar os registros de alterações feitos na tabela de coletas para verificar o usuário que realizou a modificação.*
- *Como um voluntário quero poder verificar os registros de alterações feitos na tabela de estoque para verificar o usuário que realizou a modificação.*

- *Como um voluntário quero poder verificar os registros de alterações feitos na tabela de eventos para verificar o usuário que realizou a modificação.*
- *Como um voluntário quero poder verificar os registros de alterações feitos na tabela de pontos de venda para verificar o usuário que realizou a modificação.*
- *Como um voluntário quero poder inserir o preço das agendas e calendários na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder editar o preço das agendas e calendários na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder excluir o preço das agendas e calendários na tabela de pontos de venda para manter as informações atualizadas.*
- *Como um voluntário quero poder inserir o preço das agendas e calendários na tabela de pontos de venda e receber a alteração dos valores na tabela eventos para manter os cálculos das vendas dos produtos nos eventos atualizados.*
- *Como um voluntário quero poder alterar o preço das agendas e calendários na tabela de pontos de venda e receber a alteração dos valores na tabela eventos para manter os cálculos das vendas dos produtos nos eventos atualizados.*

### **3.5 Entidades**

A aplicação foi projetada contendo alguns tipos de entidades definidas previamente em suas estórias. Podemos, a partir disso, gerar uma definição mais clara de o que cada entidade precisa ter e qual o seu relacionamento no sistema com os demais componentes. A figura 3.1 explicita o modelo de entidades e relacionamentos projetados para este sistema.

Nesta seção serão apresentadas as entidades que fazem parte do sistema. Em cada seção será detalhada cada uma das entidades do banco de dados, apresentando seus atributos e funcionalidades.

#### **3.5.1 Users**

A entidade users define o usuário do sistema e suas informações obtidas durante o cadastro de um usuário no sistema. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **username:** nome de usuário informado pelo usuário, que deve ser único em todo o sistema.
- **senha:** senha informada pelo usuário durante o cadastro.
- **nome\_completo:** nome completo do usuário.
- **criacao:** data da criação do cadastro do voluntário.
- **data\_registro:** data, em português, da criação do cadastro do voluntário.

### 3.5.2 Colaboradores

A entidade colaboradores define a tabela com a lista de colaboradores da ONG Patas Dadas e suas avaliações. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **recusaram:** lista de empresas que se recusaram a ser parceiras do Patas Dadas.
- **problemas:** lista de colaboradores em que houve algum problema na parceria.
- **bons:** lista com os bons colaboradores.
- **neutros:** lista com os colaboradores considerados neutros pelos voluntários do Patas Dadas.
- **registro:** armazena se os dados inseridos no banco de dados são uma inclusão, alteração ou deleção.
- **criacao:** data da inserção dos dados na tabela.
- **data\_registro:** data, em português, da inserção dos dados na tabela.
- **user\_id:** chave estrangeira que referencia o identificador único do usuário na entidade users.

### 3.5.3 Coleta

A entidade coleta define a tabela com a lista de locais em que os voluntários da ONG Patas Dadas devem coletar doações. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **local:** nome dos locais em que há doações disponíveis para coleta.
- **dia:** dia da coleta, ou seja, o dia em que a doação chegou ao local parceiro.
- **mes:** mês da coleta, ou seja, o mês em que a doação chegou ao local parceiro.
- **ano:** ano da coleta, ou seja, o ano em que a doação chegou ao local parceiro.
- **item:** item a ser coletado.
- **quantidade:** quantidade de itens a serem coletados.
- **registro:** armazena se os dados inseridos no banco de dados são uma inclusão, alteração ou deleção.
- **criacao:** data da inserção dos dados na tabela.
- **data\_registro:** data, em português, da inserção dos dados na tabela.
- **user\_id:** chave estrangeira que referencia o identificador único do usuário na entidade users.

### 3.5.4 Estoque

A entidade estoque define a tabela com a lista de produtos em estoque com os voluntários da ONG Patas Dadas. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **nome:** nome do voluntário que está armazenando consigo o item.
- **item:** produto armazenado pelo voluntário.
- **q\_recebida:** quantidade do produto que está sendo armazenado pelo voluntário.

- **q\_devolvida:** quantidade do produto que o voluntário deixou de armazenar.
- **observacao:** campo destinado para possíveis observações feitas pelos voluntários.
- **registro:** armazena se os dados inseridos no banco de dados são uma inclusão, alteração ou deleção.
- **criacao:** data da inserção dos dados na tabela.
- **data\_registro:** data, em português, da inserção dos dados na tabela.
- **user\_id:** chave estrangeira que referencia o identificador único do usuário na entidade users.

### 3.5.5 Eventos

A entidade eventos define o cronograma dos eventos sociais em que a ONG Patas Dadas participa. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **dia:** dia do evento.
- **mes:** mês do evento.
- **ano:** ano do evento.
- **evento:** nome do evento em que ocorre a participação do Patas Dadas.
- **responsavel:** voluntário responsável pelo evento.
- **agenda:** quantidade de agendas que devem ser levadas para o evento para serem vendidas.
- **calendario:** quantidade de calendários que devem ser levados para o evento para serem vendidos.
- **a\_vendida:** quantidade de agendas vendidas no evento.
- **c\_vendido:** quantidade de calendários vendidos no evento.
- **total:** valor total arrecadado com a venda das agendas e dos calendários.

- **observacao:** campo destinado para possíveis observações feitas pelos voluntários.
- **registro:** armazena se os dados inseridos no banco de dados são uma inclusão, alteração ou deleção.
- **criacao:** data da inserção dos dados na tabela.
- **data\_registro:** data, em português, da inserção dos dados na tabela.
- **user\_id:** chave estrangeira que referencia o identificador único do usuário na entidade users.
- **preco\_id:** chave estrangeira que referencia o identificador único do preço na entidade preco.
- **soma\_eventos\_id:** chave estrangeira que referencia o identificador único da soma dos valores na entidade soma\_eventos.

### 3.5.6 Vendas

A entidade vendas define os dados relativos aos pontos de venda das agendas e dos calendários. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **local:** nome do local do ponto de venda.
- **endereco:** endereço do ponto de venda.
- **contato:** contato do local do ponto de venda.
- **leva:** pessoa encarregada por levar as agendas e calendários para os pontos de venda.
- **v\_responsavel:** voluntário responsável pelo monitoramento dos itens nos pontos de venda.
- **agenda:** quantidade de agendas que estão nos pontos de venda.
- **calendario:** quantidade de calendários que estão nos pontos de venda.
- **total:** valor total dos preços das agendas e calendários, vendidos e não vendidos.

- **pago:** valor total recebido pela venda das agendas e calendários.
- **pagar:** valor a ser recebido após o restante das vendas das agendas e calendários.
- **observacao:** campo destinado para possíveis observações feitas pelos voluntários.
- **registro:** armazena se os dados inseridos no banco de dados são uma inclusão, alteração ou deleção.
- **criacao:** data da inserção dos dados na tabela.
- **data\_registro:** data, em português, da inserção dos dados na tabela.
- **user\_id:** chave estrangeira que referencia o identificador único do usuário na entidade users.
- **preco\_id:** chave estrangeira que referencia o identificador único do preço na entidade preco.
- **soma\_vendas\_id:** chave estrangeira que referencia o identificador único da soma dos valores na entidade soma\_vendas.

### 3.5.7 Preços

A entidade precos define os preços das agendas e calendários vendidos pela ONG Patas Dadas. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **agenda:** preço de venda da agenda.
- **calendario:** preço de venda do calendário.

### 3.5.8 Soma\_eventos

A entidade soma\_eventos define o resultado das somas do número de agendas e calendários vendidos nos eventos e da movimentação financeira gerada por essas vendas. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.

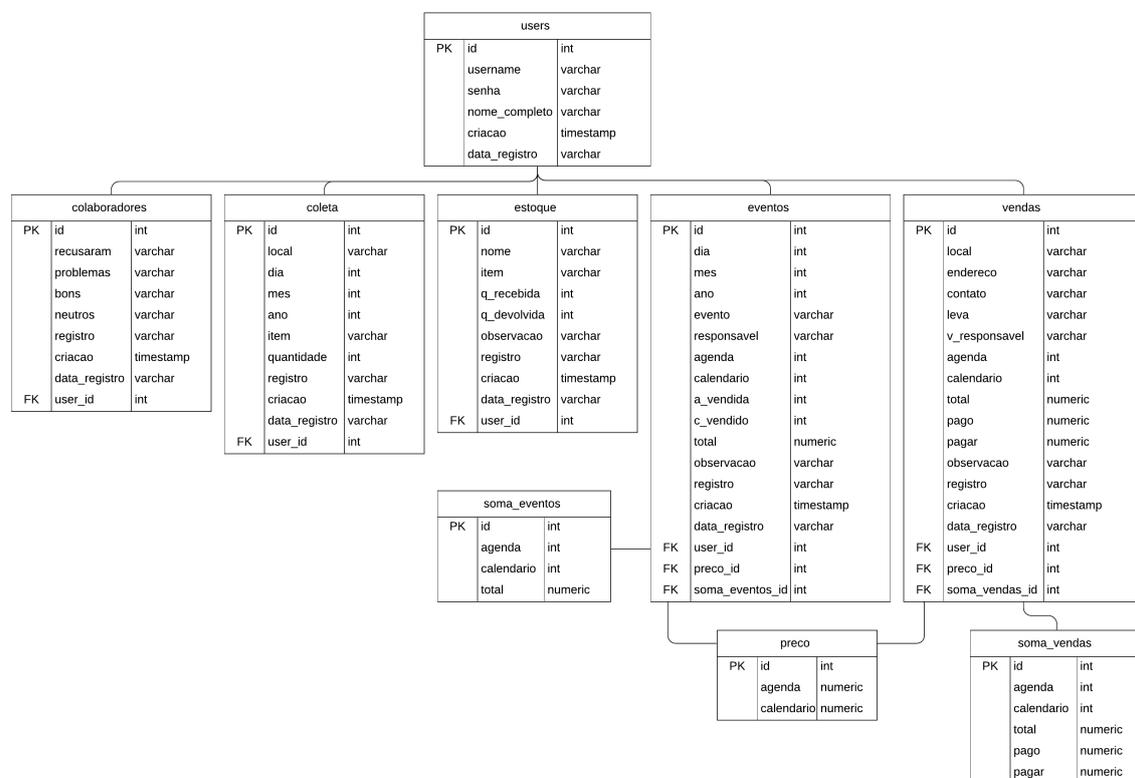
- **agenda:** soma total do número de agendas vendidas.
- **calendario:** soma total do número de calendários vendidos.
- **total:** soma do valor total dos itens vendidos.

### 3.5.9 Soma\_vendas

A entidade soma\_vendas define o resultado das somas do número de agendas e calendários vendidos nos pontos de venda e da movimentação financeira gerada por essas vendas. Essa entidade possui os seguintes atributos:

- **id:** identificador único gerado automaticamente pelo MySQL.
- **agenda:** soma do número de agendas.
- **calendario:** soma do número de calendários.
- **total:** soma total do valor dos itens, vendidos ou não.
- **pago:** soma do valor total recebido pelo venda dos produtos.
- **pagar:** soma do valor total dos itens que ainda não foram vendidos.

Figura 3.1: Diagrama Entidade-Relacionamento.



## 4 DESENVOLVIMENTO DA APLICAÇÃO

Nesta seção serão comentadas algumas das implementações principais utilizadas na camada de Back-End da aplicação, explicando os fluxos de tratamento de dados e redirecionamentos e algumas das funcionalidades centrais.

### 4.1 Arquitetura da Aplicação

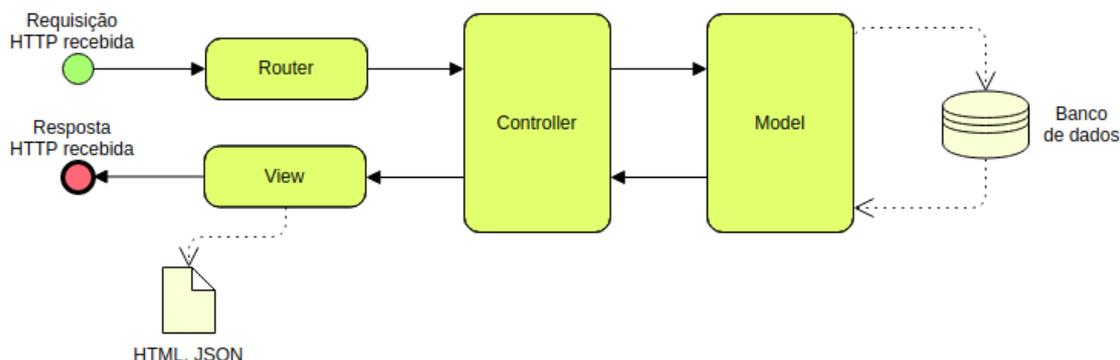
O modelo de arquitetura utilizada no sistema desenvolvido é do tipo cliente - servidor. O conceito básico por trás desse modelo é a existência de pares de processos que se comunicam por mensagem transportadas através da rede. Cada par de processos em comunicação possui um processo rotulado como Cliente e outro rotulado como Servidor. Se o usuário acessa uma aplicação pela Web, por exemplo, o navegador do qual ele acessa o sistema seria o processo Cliente e o servidor no qual a aplicação está rodando seria o processo Servidor (KUROSE; ROSS, 2017).

Quando uma página é carregada no lado do Cliente, o navegador solicita a mesma ao servidor, acionando o controlador da aplicação, no processo Servidor, que devolve o código requisitado, juntamente com os dados, para ser exibido pelo navegador do Cliente. Os dados trazidos pelo controlador são gerados através de chamadas ao banco de dados, feitas por intermédio dos modelo.

No caso de requisições assíncronas utilizadas para a fazer a busca dos dados das tabelas, por exemplo - a chamada vinda da visão no navegador do Cliente é recebida pelo controlador (no lado Servidor), que busca os dados através do modelo, e devolve para a visão, no formato JSON, e os dados são tratados através do JavaScript pelo navegador do Cliente, sem a necessidade de recarregar a visão por inteiro.

O fluxo completo da requisição HTTP passando pela lógica de roteamento, pelas camadas de controlador e modelo do MVC, chegando ao banco de dados e retornando até a saída em HTML ou JSON na resposta HTTP pode ser observado na figura 4.1.

Figura 4.1: Fluxo de informações a cada interação com o usuário.



## 4.2 Configuração e Criação do Projeto

O dinamismo da aplicação e a resposta aos comandos dos usuários é garantido através de documentos JavaScript auxiliares, que são responsáveis pela apresentação, modificação, inclusão e exclusão de elementos gráficos e controles da página, através da manipulação de trechos de código da página quando necessário. O framework Bootstrap 4 foi utilizado para a padronização da aparência e comportamento dos elementos da página.

É comum em muitos projetos a utilização de ferramentas de controle de pacotes ou *Package Managers*. Um pacote em Node.js contém todos os arquivos necessários para um módulo - módulos são bibliotecas JavaScript que podem ser inclusas no projeto. Em especial, dois grandes softwares fazem o controle de pacotes para projetos executados com Node.js, sendo eles o *NPM* e o *Yarn*. Esses softwares fazem uso de um arquivo JSON que descreva as dependências de bibliotecas e módulos que a aplicação possui para ser executada corretamente em qualquer servidor.

Os módulos utilizados no projeto foram facilmente instalados, bastou inserir o comando `npm install <package>` no terminal do sistema utilizado. O download dos pacotes é realizado no diretório local denominado `node_modules`, que contém todos os pacotes e suas dependências. Os pacotes baixados são registrados como dependências do projeto em um arquivo chamado `package.json`, que também contém informações sobre o projeto e os módulos (figura 4.2).

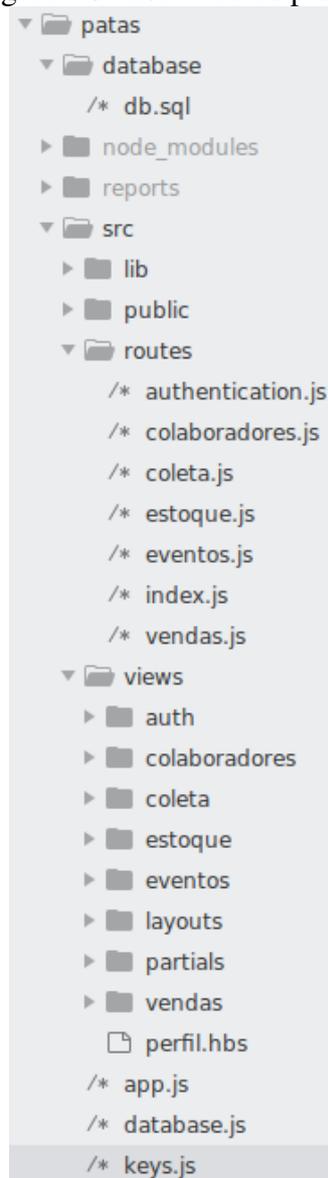
Figura 4.2: Arquivo de configuração de dependência do projeto.

```
{
  "author": "",
  "dependencies": {
    "bcryptjs": "2.4.3",
    "body-parser": "1.19.0",
    "connect-flash": "0.1.1",
    "express": "4.17.1",
    "express-handlebars": "3.1.0",
    "express-mysql-session": "2.1.0",
    "express-session": "1.16.2",
    "express-validator": "6.1.1",
    "json-stringify": "1.0.0",
    "morgan": "1.9.1",
    "mysql": "2.17.1",
    "nightwatch": "1.1.13",
    "node-sass": "4.12.0",
    "passport": "0.4.0",
    "passport-local": "1.0.0",
    "timeago.js": "4.0.0-beta.2",
    "upgrade-node-modules": "1.2.0"
  },
  "description": "",
  "devDependencies": {
    "chromedriver": "75.0.1",
    "nodemon": "1.19.1"
  },
  "directories": {
    "doc": "docs"
  },
  "keywords": [
  ],
  "license": "ISC",
  "main": "app.js",
  "name": "patas",
  "scripts": {
    "start": "node src/app.js",
    "dev": "nodemon src/app.js",
    "e2e": "nightwatch ./test/e2e/*.js"
  },
  "version": "1.0.0"
}
```

### 4.3 Estrutura do Projeto

Na estrutura do projeto há a pasta *routes* que corresponde a parte dos *controllers*, já a pasta *database* contém a *user model*, enquanto que a pasta *views* possui as *views* com a extensão *.handlebars* (figura 4.3).

Figura 4.3: Estrutura do projeto.



#### 4.4 Autenticação

O sistema de autenticação fornecido pelo Node.js utilizado neste projeto é o *Passport*, um middleware para o Node.js que faz a implementação de autenticação em um aplicativo de maneira rápida e fácil. Para esta aplicação foi desenvolvido o modo de se autenticar utilizando username e senha, sem a possibilidade de fazer uma autenticação utilizando redes sociais. Quando o usuário se logar, o registro do usuário é armazenado na sessão a fim de manter o estado de logado enquanto ele navega em seu site. Funções *serializeUser()* e *deserializeUser()* são fornecidas para controlar esse processo, conforme mostra a figura 4.4.

Figura 4.4: Funções `serializeUser()` e `deserializeUser()`.

```
passport.serializeUser((user, done) => {
  done(null, user.id);
});

passport.deserializeUser(async (id, done) => {
  const rows = await pool.query('SELECT * FROM users WHERE id = ?', [id]);
  done(null, rows[0]);
});
```

A função `done` é nativa do Passport e serve para sinalizar erro e passar informações do usuário para o Passport. Na função `serializeUser` é passada a chave primária para ser serializado no cookie do usuário. Já no `deserializeUser` o id é recebido, buscado no banco de dados e retornado como o usuário inteiro. A estratégia de autenticação, chamada de `LocalStrategy`, está representada na figura 4.5:

Figura 4.5: Estratégia de autenticação.

```
passport.use('local.signin', new LocalStrategy({
  usernameField: 'username',
  passwordField: 'password',
  passReqToCallback: true
}), async (req, username, password, done) => {
  const rows = await pool.query('SELECT * FROM users WHERE username = ?', [username]);
  if (rows.length > 0) {
    const user = rows[0];
    const validPassword = await helpers.matchPassword(password, user.password)
    if (validPassword) {
      done(null, user, req.flash('success', 'Olá ' + user.username));
    } else {
      done(null, false, req.flash('message', 'Senha incorreta'));
    }
  } else {
    return done(null, false, req.flash('message', '0 usuário não existe'));
  }
});

passport.use('local.signup', new LocalStrategy({
  usernameField: 'username',
  passwordField: 'password',
  passReqToCallback: true
}), async (req, username, password, done) => {
  const { fullname } = req.body;
  let newUser = {
    fullname,
    username,
    password
  };
  newUser.password = await helpers.encryptPassword(password);
  const result = await pool.query('INSERT INTO users SET ? ', newUser);
  newUser.id = result.insertId;
  return done(null, false, req.flash('success', 'Usuário ' + newUser.username + ' cadastrado com sucesso'));
});
```

- `usernameField` e `passwordField`: são os nomes dos inputs no form HTML que será postado durante a autenticação;
- `rows`: acessa o banco de dados para buscar usuário por username.
- `helpers.matchPassword`: se o usuário existir, será usada a `helpers.matchPassword` entre a senha digitada no form de login (senha) e a senha salva (hash) no banco de dados, o retorno dessa comparação também é sinalizado ao Passport via a função `done`.

Para manipular a sessão de aplicações Web feitas usando Express foi utilizado o módulo `express-session`, que permite armazenar os dados da sua sessão em diferentes

mecanismos de persistência, desde que usando os conectores adequados. A configuração encontra-se no arquivo `app.js` do projeto (figura 4.6):

Figura 4.6: Sessão da aplicação.

```
const session = require('express-session');
const validator = require('express-validator');
const passport = require('passport');
const MySQLStore = require('express-mysql-session')(session);
const app = express();
require('./lib/passport');
app.use(passport.initialize());
app.use(passport.session());
app.use(validator());
```

Após a manipulação da sessão, a criação das rotas para para redirecionamento caso sucesso ou falha foi realizada utilizando-se GET e POST. Com o GET o usuário é redirecionado para a página do login ou cadastro, enquanto que o POST captura as requisições, passando ao Passport a tarefa de autenticar usando a estratégia local. O objeto de configuração passado como segundo parâmetro da função `authenticate` define as URLs de sucesso e de falha, conforme o resultado que o Passport definir. O código dessa configuração encontra-se no arquivo `authentication.js` (figura 4.7):

Figura 4.7: Rota de redirecionamento para login e cadastro.

```
router.get('/cadastrar', (req, res) => {
  res.render('auth/cadastrar');
});

router.post('/cadastrar', passport.authenticate('local.cadastrar', {
  successRedirect: '/perfil',
  failureRedirect: '/cadastrar',
  failureFlash: true
}));

router.get('/login', (req, res) => {
  res.render('auth/login');
});

router.post('/login', (req, res, next) => {
  req.check('username', 'Usuário não informado').notEmpty();
  req.check('password', 'Senha não informada').notEmpty();
  const errors = req.validationErrors();
  if (errors.length > 0) {
    req.flash('message', errors[0].msg);
    res.redirect('/login');
  }
  passport.authenticate('local.login', {
    successRedirect: '/perfil',
    failureRedirect: '/login',
    failureFlash: true
  })(req, res, next);
});
```

## 4.5 Rotas

As rotas são uma seção do Express que associa um verbo do HTTP (GET, POST, PUT, DELETE, entre outros), uma URL e a função que é chamada para lidar com o caminho passado. A figura 4.8 mostra a criação da rota associada com a aplicação Express no projeto desenvolvido.

Figura 4.8: Criação da rota.

```
const express = require('express');
const router = express.Router();

const pool = require('../database');
const { isLoggedIn } = require('../lib/auth');
```

### 4.5.1 Funções da Rota

A *router* provê métodos para todos os verbos HTTP, como, por exemplo, GET e POST. Utilizando o exemplo do método *router.get*, que responde apenas a chamadas HTTP GET, o primeiro argumento para este método é o caminho da URL, o segundo é o *callback* da função que irá definir se a requisição do caminho foi recebida. O *callback* recebe geralmente três argumentos denominados *req*, *res*, *next* que irão conter a requisição HTTP, sua resposta e a próxima função do middleware.

### 4.5.2 Chamada das Rotas

Usando o *Router()* é possível dividir a aplicação em partes, mantendo a aplicação limpa e organizada, movendo cada definição de roteamento para seu respectivo arquivo e apenas o instanciando e passando para o *app.use()* conforme a figura 4.9 - no arquivo *app.js* foram definidas as variáveis que serão passadas na URL usando a sintaxe rota / página.

Figura 4.9: Chamada de cada rota utilizada no projeto.

```
app.use(require('./routes/index'));
app.use(require('./routes/authentication'));
app.use('/colaboradores', require('./routes/colaboradores'));
app.use('/coleta', require('./routes/coleta'));
app.use('/estoque', require('./routes/estoque'));
app.use('/eventos', require('./routes/eventos'));
app.use('/venda', require('./routes/vendas'));
```

### 4.5.3 CRUD

Para a criação das tabelas de colaboradores, coleta, estoque, eventos e pontos de venda, os controladores possuem métodos que implementam as funcionalidades de CRUD (Criar, Ler, Atualizar e Remover). O CRUD foi implementado nas rotas do projeto, utilizando-se as funções de GET e POST do HTTP para receber os dados passados pelos usuários e assim realizar a funcionalidade solicitada.

Utilizando-se do código mostrado na figura 4.10 para exemplificar o processo de edição da página de colaboradores, podemos perceber que *router.get()* recebe os dados presentes no banco de dados e renderiza a página com essas informações. Através do *router.post()* os dados editados são atualizados no banco de dados.

Figura 4.10: Exemplo de edição dos dados na tabela dos colaboradores.

```
router.get('/editar-colaboradores/:id', async (req, res) => {
  const { id } = req.params;
  const colaboradores = await pool.query('SELECT * FROM colaboradores WHERE id = ?', [id]);
  res.render('colaboradores/editar-colaboradores', {colaborador: colaboradores[0]});
});

router.post('/editar-colaboradores/:id', async (req, res) => {
  const { id } = req.params;
  const { recusaram, problemas, bons, neutros } = req.body;
  const registro = 'editado';
  const newLink = {
    recusaram,
    problemas,
    bons,
    neutros,
    registro
  };
  await pool.query('UPDATE colaboradores SET ? WHERE id = ?', [newLink, id]);
  req.flash('success', 'Atualizado com sucesso');
  res.redirect('/colaboradores');
});
```

### 4.6 Banco de Dados

O banco de dados utilizado no projeto foi o MySQL, visto que atualmente ele é um dos sistemas de gerenciamento de banco de dados mais utilizados no mundo (Software Testing Help, 2019), apesar de poder ser usado em uma vasta variedade de aplicações, tem seu uso mais comum nas aplicações baseadas em Web e possui ampla documentação. Além disso, o MySQL é eficiente para resolver conflitos de acessos simultâneos, realizando o bloqueio dentro do próprio servidor MySQL para gerenciar a contenção do conteúdo da tabela por várias sessões. O MySQL usa o bloqueio de nível de linha para tabelas InnoDB para suportar o acesso de gravação simultânea por várias sessões, tornando-as adequadas em situação de diversas alterações simultâneas.

## 4.7 Testes

Durante o desenvolvimento do projeto foram realizados testes e2e (end-to-end), que são uma metodologia utilizada para testar se o fluxo da aplicação está sendo executado conforme o projetado. A ferramenta utilizada para a criação dos testes foi a Nightwatch. Na figura 4.11 está o exemplo de um dos testes realizados - a simulação do acesso a aplicação através do login, o acesso ao perfil do usuário e em seguida o redirecionamento para a página da tabela dos colaboradores, em que há o teste da adição de novos dados.

Figura 4.11: Teste e2e de adição na tabela dos colaboradores.

```

module.exports = {
  'Step one: Acessar o site e fazer login' : function (client) {
    client
      .url('http://localhost:4000/login')
      .setValue('input[name="username"]', 'KF')
      .setValue('input[name="password"]', 'teste')
      .pause(2000)
      .click('#btnLogin')
      .pause(1000)
  },

  'Step two: Acessar o perfil' : function (client) {
    client
      .url('http://localhost:4000/perfil')
      .pause(1000)
  },

  'Step three: Clicar no navbar' : function (client){
    client
      .click('#navbarDropdown')
      .pause(1000)
  },

  'Step four: Clicar na tabela de colaboradores' : function(client){
    client
      .click('#navbarColaboradores')
      .pause(4000)
  },

  'Step five: Adicionar colaboradores' : function(client){
    client
      .click('#btnAddColab')
      .pause(1000)
  },

  'Step six: Salvar colaboradores adicionados' : function(client){
    client
      .pause(2000)
      .setValue('input[name="recusaram"]', 'Govinda')
      .setValue('input[name="problemas"]', 'Ve')
      .setValue('input[name="bons"]', 'PetShop')
      .setValue('input[name="neutros"]', 'Veterinária')
      .pause(2000)
      .click('#btnSalvaAddColab')
      .pause(4000)
  },
}

```

A avaliação dos testes foi baseada de acordo com o tempo de resposta, plataformas

compatíveis e conflito de acesso simultâneo. Os testes foram executados simultaneamente nos navegadores Google Chrome, Internet Explorer, Mozilla Firefox e Safari com um tempo limite de 10 segundos para encontrar os elementos da página, caso não haja o elemento na tela depois desse tempo, o teste irá quebrar.

Os testes apresentaram sucesso em todos os casos testados, sendo possível determinar que o sistema é compatível com os navegadores, tendo um tempo máximo de resposta do sistema de 10 segundos e os conflitos de acesso simultâneo no banco de dados foram resolvidos com sucesso.

#### **4.8 Versão Final**

A versão final do sistema apresentado possui todas as funcionalidades solicitadas pelos voluntários do Patas Dadas, porém por ser uma etapa inicial do sistema, existem outras funcionalidades que podem ser implementadas em versões futuras, como, por exemplo, a criação de um ambiente para os colaboradores da ONG e a criação de mais tabelas para, por exemplo, auxiliar na organização referente aos animais adotados pelo Patas Dadas.

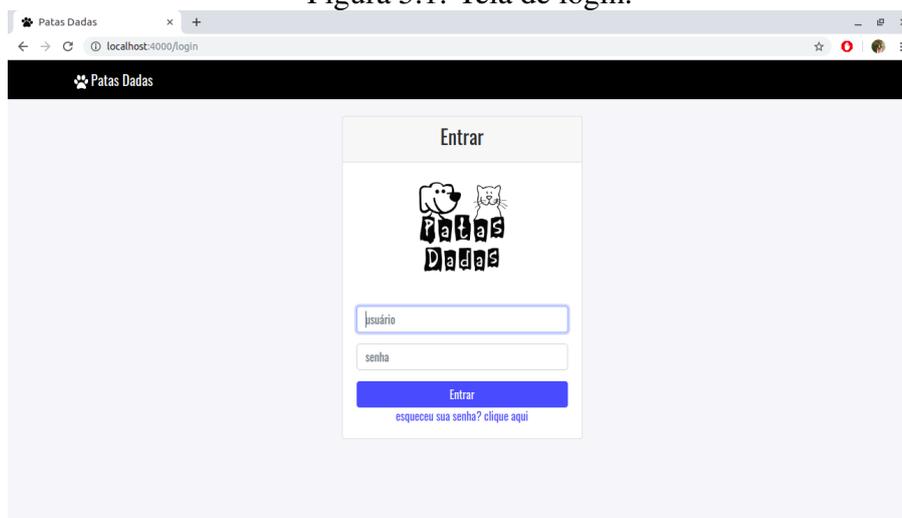
## 5 GUIA DE USO DO SISTEMA

Este capítulo mostra o comportamento visual da aplicação de página única (SPA) de acordo com a interação do usuário.

### 5.1 Autenticação

Um dos requisitos para o usuário ter acesso a aplicação é estar logado, então assim que entrar na tela inicial da aplicação o usuário deve fazer login com uma conta já existente. Apenas duas informações são necessárias para realizar o login: usuário e senha.

Figura 5.1: Tela de login.



### 5.2 Cadastro

Como o sistema é restrito para voluntários do Patas Dadas, o cadastro é liberado apenas para quem já possui acesso realizar o cadastro de novos usuários.

Para cadastrar uma nova conta o usuário tem que estar logado no sistema, clicar em "Cadastrar" e escolher um nome de usuário (único), uma senha e inserir o nome completo do novo usuário. Ao preencher todas as informações corretamente e clicar no botão "Cadastrar" da tela de registro o usuário receberá um aviso de que o cadastro foi realizado com sucesso.

Figura 5.2: Tela de cadastro.

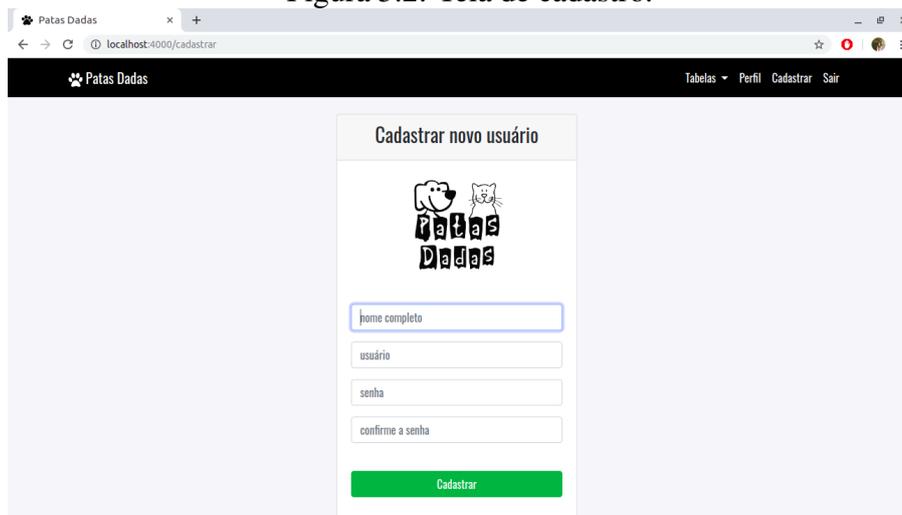
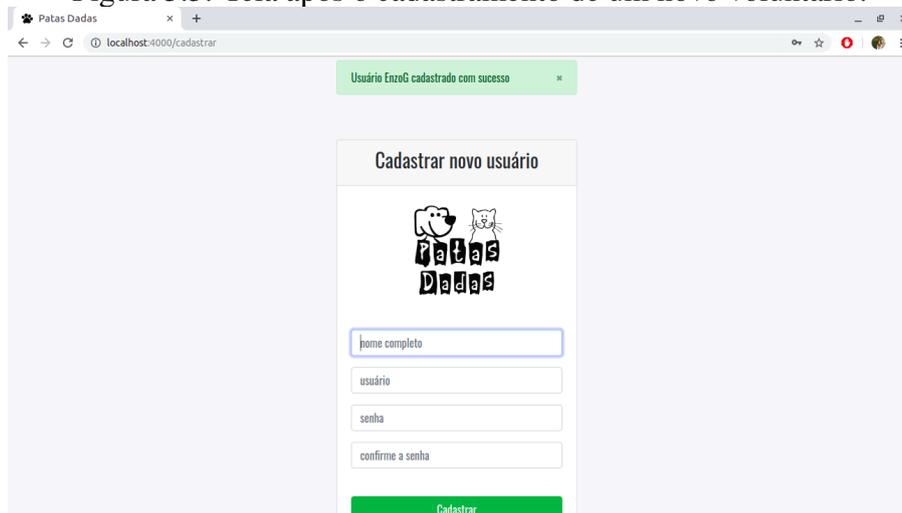


Figura 5.3: Tela após o cadastramento de um novo voluntário.



### 5.3 Navbar

O cabeçalho de navegação (navbar) utilizado no projeto é responsivo e inclui suporte para logo, navegação e plugin collapse. Através do navbar o usuário pode acessar as tabelas, a opção de perfil, o cadastramento, sair da aplicação e ir para a página inicial do Patas Dadas.

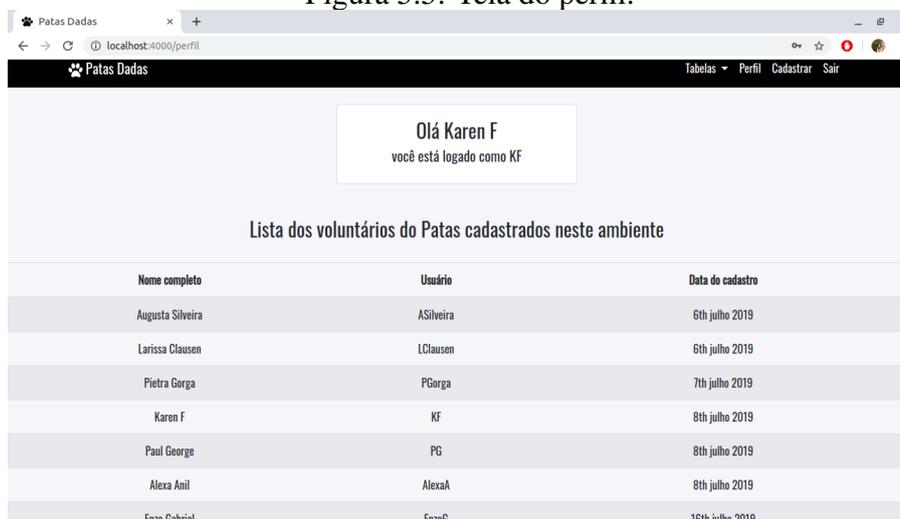
Figura 5.4: Cabeçalho de navegação.



## 5.4 Perfil

Após acessar o sistema o voluntário é redirecionado para a página de perfil, em que consta os seus dados de cadastro (nome completo e username), além disso há a lista com todos os usuários com cadastro no sistema, para que os voluntários tenham o controle do número de pessoas cadastradas e se a lista de usuários está atualizada.

Figura 5.5: Tela do perfil.



Nome completo	Usuário	Data do cadastro
Augusta Silveira	ASilveira	6th julho 2019
Larissa Clausen	LClausen	6th julho 2019
Pietra Gorga	PGorga	7th julho 2019
Karen F	KF	8th julho 2019
Paul George	PG	8th julho 2019
Alexa Anil	AlexaA	8th julho 2019
Enzo Gabriel	EnzoG	16th julho 2019

## 5.5 Tabelas

Os voluntários do Patas Dadas solicitaram 5 tipos de tabelas, uma responsável por gerenciar a classificação dos colaboradores da ONG, duas tabelas para controlar os produtos disponíveis para coleta e em estoque, uma outra para armazenar os dados dos eventos em que eles participam e uma quinta tabela responsável pelo gerenciamento dos pontos de venda das agendas e calendários do Patas Dadas.

Em todas as tabelas os voluntários solicitaram a adição de uma lista que contenha as modificações feitas durante os últimos 30 dias a contar da data da modificação, assim como o nome do usuário que realizou essa modificação, o tipo de modificação (adição, alteração, deleção) e a data da ação. Na parte inferior de cada tabela será possível ver os registros contendo essas modificações.

### 5.5.1 Colaboradores

Na tabela dos colaboradores o voluntário pode adicionar, editar ou deletar a lista de colaboradores da ONG, bem como avaliá-los pelo nível de ajuda e participação. Além disso, há o registro das ações realizadas pelos voluntários na tabela.

A figura 5.6 mostra a tabela sem nenhum dado inserido, se o voluntário desejar inserir algum dado ele deve clicar no botão "Adicionar dados", para assim ser redirecionado para a página de adição dos dados (figura 5.7). Após a inserção dos dados aparecerá uma notificação de sucesso na inclusão da informação e o usuário receberá duas informações na tela: os dados que ele inseriu na tabela de colaboradores e os dados presentes no registro de alteração da tabela, conforme a imagem 5.8 apresenta.

Figura 5.6: Tela da tabela de colaboradores sem a inserção de dados.



Figura 5.7: Tela da adição de novos colaboradores.

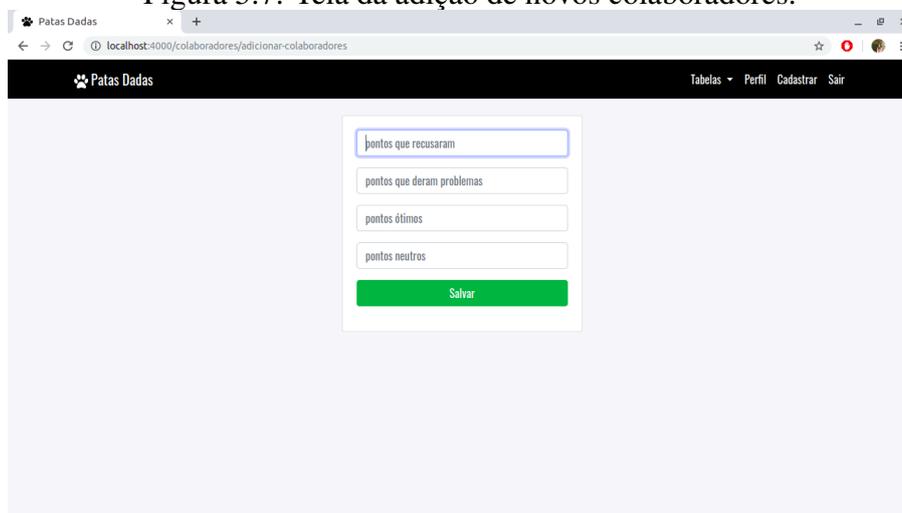
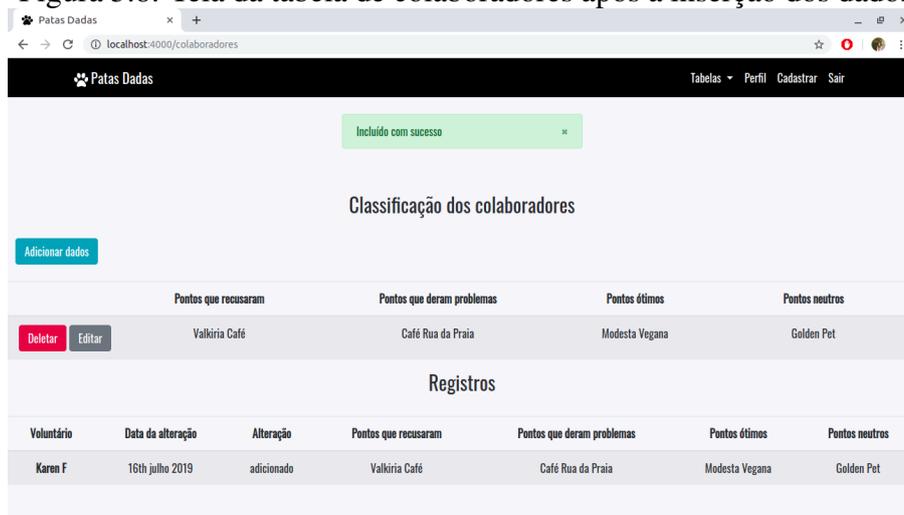


Figura 5.8: Tela da tabela de colaboradores após a inserção dos dados.



Para a edição dos dados o processo é semelhante, o voluntário deve clicar no botão "Editar" para ser redirecionado para a tela de edição (figura 5.9). Após a alteração dos dados o usuário será novamente encaminhado para a tabela de colaboradores, que agora contém os dados que foram editados - figura 5.10.

Figura 5.9: Tela da edição da lista de colaboradores.

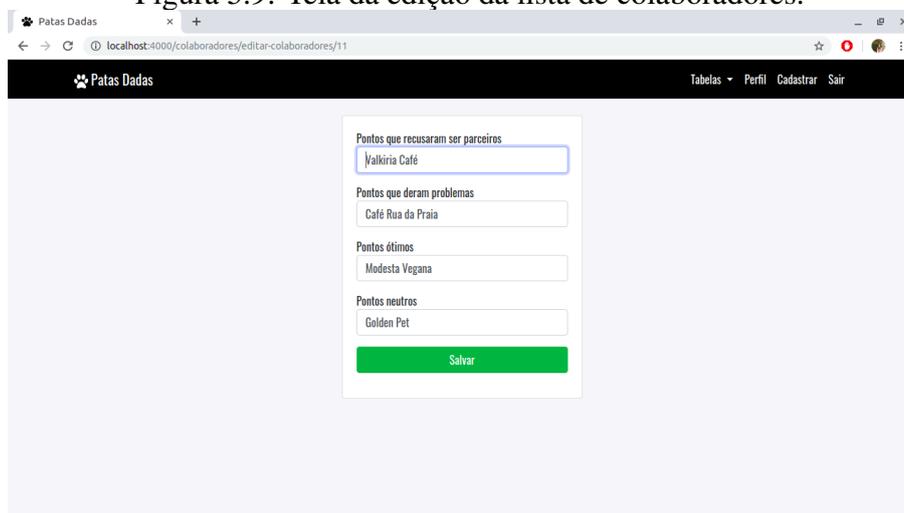


Figura 5.10: Tela da tabela de colaboradores após a edição dos dados.

Atualizado com sucesso

Classificação dos colaboradores

Adicionar dados

Pontos que recusaram	Pontos que deram problemas	Pontos ótimos	Pontos neutros
Govinda	Café Rua da Praia	Modesta Vegana	Golden Pet

Registros

Voluntário	Data da alteração	Alteração	Pontos que recusaram	Pontos que deram problemas	Pontos ótimos	Pontos neutros
Karen F	16th julho 2019	editado	Govinda	Café Rua da Praia	Modesta Vegana	Golden Pet

Para deletar alguma informação basta o voluntário clicar no botão "Deletar" que automaticamente serão atualizadas as tabelas de colaboradores e registros - figura 5.11.

Figura 5.11: Tela da tabela de colaboradores após a deleção dos dados.

Removido com sucesso

Classificação dos colaboradores

Adicionar dados

Pontos que recusaram	Pontos que deram problemas	Pontos ótimos	Pontos neutros
Chévere	Boutique do Bicho	Luther	Mania de Cachorro

Registros

Voluntário	Data da alteração	Alteração	Pontos que recusaram	Pontos que deram problemas	Pontos ótimos	Pontos neutros
Karen F	16th julho 2019	deletado	Govinda	Café Rua da Praia	Modesta Vegana	Golden Pet
Karen F	16th julho 2019	adicionado	Chévere	Boutique do Bicho	Luther	Mania de Cachorro

## 5.5.2 Pontos de Coleta

Na tabela das coletas o voluntário pode adicionar, editar ou deletar dados referentes as doações existentes para coleta nos pontos parceiros. Além disso, há o registro das ações realizadas pelos voluntários na tabela.

A figura 5.12 mostra a tabela sem nenhum dado inserido, se o voluntário desejar inserir algum dado ele deve clicar no botão "Adicionar dados", para assim ser redirecionado para a página de adição dos dados (figura 5.13). Após a inserção dos dados aparecerá uma notificação de sucesso na inclusão da informação e o usuário receberá duas informações

na tela: os dados que ele inseriu na tabela de coleta e os dados presentes no registro de alteração da tabela, conforme a imagem 5.14 apresenta.

Figura 5.12: Tela da tabela de coleta sem a inserção de dados.

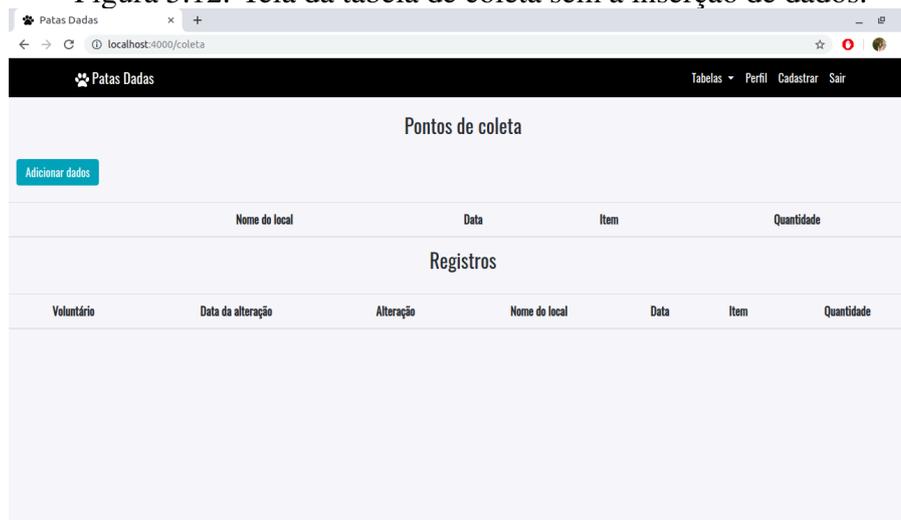


Figura 5.13: Tela da adição de novos itens a serem coletados.



Figura 5.14: Tela da tabela de coleta após a inserção dos dados.

The screenshot shows a web application interface for 'Patás Dadas'. At the top, there is a navigation bar with 'Tabelas', 'Perfil', 'Cadastrar', and 'Sair'. A green notification box at the top center says 'Incluído com sucesso'. Below this is the title 'Pontos de coleta' and a blue button 'Adicionar dados'. The main content is a table with the following data:

	Nome do local	Data	Item	Quantidade
<a href="#">Deletar</a> <a href="#">Editar</a>	Águia	22/6/2019	cama	2

Below the table is a section titled 'Registros' with a table showing the history of changes:

Voluntário	Data da alteração	Alteração	Nome do local	Data	Item	Quantidade
Karen F	16th julho 2019	adicionado	Águia	22/6/2019	cama	2

Para a edição dos dados o processo é semelhante, o voluntário deve clicar no botão "Editar" para ser redirecionado para a tela de edição (figura 5.15). Após a alteração dos dados o usuário será novamente encaminhado para a tabela de pontos de coleta, que agora contém os dados que foram editados - figura 5.16.

Figura 5.15: Tela da edição da tabela de pontos de coletas.

The screenshot shows the 'editar-coleta/18' page in the 'Patás Dadas' application. The page contains a form for editing a collection point record. The form fields are:

- Nome do local: Águia
- Data: 22/6/2019
- Item: cama
- Quantidade: 4

A green 'Salvar' button is located at the bottom of the form.

Figura 5.16: Tela da tabela de coleta após a edição dos dados.

Atualizado com sucesso

Pontos de coleta

Adicionar dados

Nome do local	Data	Item	Quantidade
Águia	22/6/2019	cama	4

Registros

Voluntário	Data da alteração	Alteração	Nome do local	Data	Item	Quantidade
Karen F	16th julho 2019	editado	Águia	22/6/2019	cama	4

Para deletar alguma informação basta o voluntário clicar no botão "Deletar" que automaticamente serão atualizadas as tabelas de coleta e registros - figura 5.17.

Figura 5.17: Tela da tabela de coleta após a deleção dos dados.

Removido com sucesso

Pontos de coleta

Adicionar dados

Nome do local	Data	Item	Quantidade
Molecão	30/6/2019	coleira	8

Registros

Voluntário	Data da alteração	Alteração	Nome do local	Data	Item	Quantidade
Karen F	16th julho 2019	deletado	Águia	22/6/2019	cama	4
Karen F	16th julho 2019	adicionado	Molecão	30/6/2019	coleira	8

### 5.5.3 Estoque

Na tabela dos estoques o voluntário pode adicionar, editar ou deletar a relação entre os itens disponíveis em estoque que estão sob os cuidados dos voluntários do Patas Dadas. Além disso, há o registro das ações realizadas pelos voluntários na tabela.

A figura 5.18 mostra a tabela sem nenhum dado inserido, se o voluntário desejar inserir algum dado ele deve clicar no botão "Adicionar dados", para assim ser redirecionado para a página de adição dos dados (figura 5.19). Após a inserção dos dados aparecerá uma notificação de sucesso na inclusão da informação e o usuário receberá duas informações

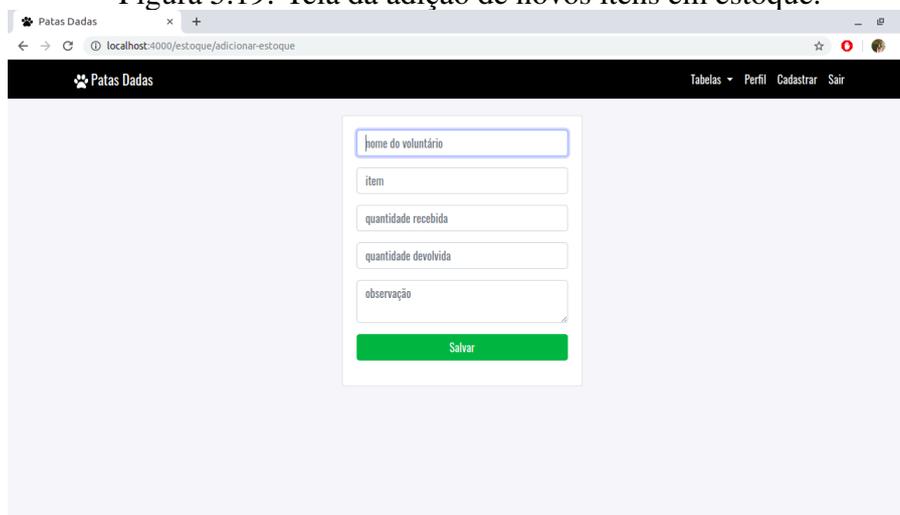
na tela: os dados que ele inseriu na tabela de estoque e os dados presentes no registro de alteração da tabela, conforme a imagem 5.20 apresenta.

Figura 5.18: Tela da tabela de estoque sem a inserção de dados.



The screenshot shows a web browser window with the URL `localhost:4000/estoque`. The page title is "Produtos em estoque". There is a navigation bar with "Tabelas", "Perfil", "Cadastrar", and "Sair". A blue button labeled "Adicionar dados" is on the left. The main content area contains a table with the following headers: "Nome do voluntário", "Item", "Quantidade recebida", "Quantidade devolvida", and "Observação". Below this is a section titled "Registros" with a table that has headers: "Voluntário", "Data da alteração", "Alteração", "Nome do voluntário", "Item", "Quantidade recebida", "Quantidade devolvida", and "Observação". The table body is empty.

Figura 5.19: Tela da adição de novos itens em estoque.



The screenshot shows a web browser window with the URL `localhost:4000/estoque/adicionar-estoque`. The page title is "adicionar-estoque". There is a navigation bar with "Tabelas", "Perfil", "Cadastrar", and "Sair". The main content area contains a form with the following fields: "nome do voluntário", "item", "quantidade recebida", "quantidade devolvida", and "observação". A green button labeled "Salvar" is at the bottom of the form.

Figura 5.20: Tela da tabela de estoque após a inserção dos dados.

Patras Dadas

Tabelas Perfil Cadastrar Sair

Incluído com sucesso

### Produtos em estoque

Adicionar dados

	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Deletar Editar	Vanessa	agenda	20	5	5 agendas estão no evento da feira vegana

### Registros

Voluntário	Data da alteração	Alteração	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Karen F	16th julho 2019	adicionado	Vanessa	agenda	20	5	5 agendas estão no evento da feira vegana

Para a edição dos dados o processo é semelhante, o voluntário deve clicar no botão "Editar" para ser redirecionado para a tela de edição (figura 5.21). Após a alteração dos dados o usuário será novamente encaminhado para a tabela de estoque, que agora contém os dados que foram editados - figura 5.22.

Figura 5.21: Tela da edição dos itens em estoque.

Patras Dadas

Tabelas Perfil Cadastrar Sair

Nome do voluntário  
Vanessa

Item  
agenda

Quantidade recebida  
25

Quantidade devolvida  
5

Observação  
5 agendas estão no evento da feira vegana

Salvar

Figura 5.22: Tela da tabela de estoque após a edição dos dados.

Atualizado com sucesso

Produtos em estoque

Adicionar dados

	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Deletar Editar	Vanessa	agenda	25	5	5 agendas estão no evento da feira vegana

Registros

Voluntário	Data da alteração	Alteração	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Karen F	16th julho 2019	editado	Vanessa	agenda	25	5	5 agendas estão no evento da feira vegana

Para deletar alguma informação basta o voluntário clicar no botão "Deletar" que automaticamente serão atualizadas as tabelas de estoque e registros - figura 5.23.

Figura 5.23: Tela da tabela de estoque após a deleção dos dados.

Removido com sucesso

Produtos em estoque

Adicionar dados

	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Deletar Editar	João	calendário	10	0	

Registros

Voluntário	Data da alteração	Alteração	Nome do voluntário	Item	Quantidade recebida	Quantidade devolvida	Observação
Karen F	16th julho 2019	deletado	Vanessa	agenda	25	5	5 agendas estão no evento da feira vegana
Karen F	16th julho 2019	adicionado	João	calendário	10	0	

#### 5.5.4 Eventos

Na tabela dos eventos o voluntário pode adicionar, editar ou deletar o cronograma dos eventos em que o Patas Dadas participa. Além disso, há o registro das ações realizadas pelos voluntários na tabela.

A figura 5.24 mostra a tabela sem nenhum dado inserido, se o voluntário desejar inserir algum dado ele deve clicar no botão "Adicionar dados", para assim ser redirecionado para a página de adição dos dados (figura 5.25). Após a inserção dos dados aparecerá uma notificação de sucesso na inclusão da informação e o usuário receberá duas informações

na tela: os dados que ele inseriu na tabela de eventos e os dados presentes no registro de alteração da tabela, conforme a imagem 5.26 apresenta.

Figura 5.24: Tela da tabela de eventos sem a inserção de dados.

The screenshot shows a web browser window with the URL 'localhost:4000/eventos'. The page title is 'Cronograma dos eventos'. There is a navigation bar with 'Patras Dadas' and links for 'Tabelas', 'Perfil', 'Cadastrar', and 'Sair'. A green button labeled 'Adicionar dados' is visible. Below the main heading, there is a table with the following columns: Data, Evento, Voluntário responsável, Agendas para levar, Calendários para levar, Agendas vendidas, Calendários vendidos, Total arrecadado, and Observação. Below this table, there is a section titled 'Registros' with a table that has columns: Voluntário, Data da alteração, Alteração, Data, Evento, Voluntário responsável, Agendas para levar, Calendários para levar, Agendas vendidas, Calendários vendidos, Total arrecadado, and Observação. The table content is currently empty.

Figura 5.25: Tela da adição de novos eventos.

The screenshot shows a web browser window with the URL 'localhost:4000/eventos/adicionar-evento'. The page title is 'adicionar-evento'. There is a navigation bar with 'Patras Dadas' and links for 'Tabelas', 'Perfil', 'Cadastrar', and 'Sair'. The main content area contains a form with the following fields: a date selector with 'dia', 'mês', and 'ano' dropdowns; a text input for 'evento'; a text input for 'voluntário responsável'; a text input for 'agendas para levar'; a text input for 'calendários para levar'; a text input for 'agendas vendidas'; a text input for 'calendários vendidos'; and a text area for 'observação'. A green 'Salvar' button is at the bottom of the form.

Figura 5.26: Tela da tabela de eventos após a inserção dos dados.

Incluído com sucesso

### Cronograma dos eventos

Adicionar dados

	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
<input type="button" value="Deletar"/> <input type="button" value="Editar"/>	9/2/2019	Feira de lançamento		300	200	180	10	RS1900	mais 15 ecobags = RS555
<b>TOTAL</b>						180	10	RS1900	

### Registros

Voluntário	Data da alteração	Alteração	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
Karen F	16th julho 2019	adicionado	9/2/2019	Feira de lançamento		300	200	180	10	RS1900	mais 15 ecobags = RS555

Para a edição dos dados o processo é semelhante, o voluntário deve clicar no botão "Editar" para ser redirecionado para a tela de edição (figura 5.27). Após a alteração dos dados o usuário será novamente encaminhado para a tabela de eventos, que agora contém os dados que foram editados - figura 5.28.

Figura 5.27: Tela da edição dos eventos.

Data do evento

9 / 2 / 2019

Evento

Feira de lançamento

Voluntário responsável

voluntário responsável

Agendas para levar

300

Calendários para levar

200

Agendas vendidas

180

Calendários vendidos

100

Observação

mais 15 ecobags = RS555

Salvar

Figura 5.28: Tela da tabela de eventos após a edição dos dados.

Atualizado com sucesso

### Cronograma dos eventos

Adicionar dados

	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
<a href="#">Deletar</a> <a href="#">Editar</a>	9/2/2019	Feira de lançamento		300	200	180	100	RS1990	mais 15 ecobags = RS555
<b>TOTAL</b>						180	100	RS1990	

### Registros

Voluntário	Data da alteração	Alteração	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
Karen F	16th julho 2019	editado	9/2/2019	Feira de lançamento		300	200	180	100	RS1990	mais 15 ecobags = RS555

Como há produtos do Patas Dadas sendo vendidos nos eventos, os voluntários precisam, além de ter os dados relativos ao evento em si, ter um controle material e financeiro sobre os itens a serem vendidos e o dinheiro arrecadado. Na figura 5.29 é possível observar a soma dos itens vendidos e o montante arrecadado - esses valores também são salvos no banco de dados. Nota: o valor calculado pelas vendas dos produtos se dá de acordo com o preço das agendas e dos calendários que são inseridos na tabela dos pontos de venda.

Figura 5.29: Tela da tabela de eventos após o cálculo das vendas dos produtos.

Incluído com sucesso

### Cronograma dos eventos

Adicionar dados

	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
<a href="#">Deletar</a> <a href="#">Editar</a>	9/2/2019	Feira de lançamento		300	200	180	100	RS1990	mais 15 ecobags = RS555
<a href="#">Deletar</a> <a href="#">Editar</a>	10/2/2019	Feira vegana	Eva	100	50	80	40	RS880	1 ecobag
<b>TOTAL</b>						260	140	RS2870	

### Registros

Voluntário	Data da alteração	Alteração	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
Karen F	16th julho 2019	editado	9/2/2019	Feira de lançamento		300	200	180	100	RS1990	mais 15 ecobags = RS555
Karen F	16th julho 2019	adicionado	10/2/2019	Feira vegana	Eva	100	50	80	40	RS880	1 ecobag

Para deletar alguma informação basta o voluntário clicar no botão "Deletar" que automaticamente serão atualizadas as tabelas de eventos e registros - figura 5.30.

Figura 5.30: Tela da tabela de eventos após a deleção dos dados.

Removido com sucesso

Cronograma dos eventos

Adicionar dados

Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
10/2/2019	Feira vegana	Eva	100	50	80	40	R\$880	1 ecobag
<b>TOTAL</b>					80	40	R\$880	

Registros

Voluntário	Data da alteração	Alteração	Data	Evento	Voluntário responsável	Agendas para levar	Calendários para levar	Agendas vendidas	Calendários vendidos	Total arrecadado	Observação
Karen F	16th julho 2019	deletado	9/2/2019	Feira de lançamento		300	200	180	100	R\$1990	mais 15 ecobags = R\$555
Karen F	16th julho 2019	adicionado	10/2/2019	Feira vegana	Eva	100	50	80	40	R\$880	1 ecobag

### 5.5.5 Pontos de Venda

Na tabela dos pontos de venda o voluntário pode adicionar, editar ou deletar os detalhes dos pontos de venda das agendas e calendários da ONG. Assim como adicionar, editar ou deletar o valor do preço das agendas e dos calendários a serem vendidos. Além disso, há o registro das ações realizadas pelos voluntários na tabela.

A figura 5.31 mostra as tabelas de preço e pontos de venda sem nenhum dado inserido, para adicionar o valor do preço unitário das agendas e dos calendários a serem vendidos basta clicar no botão "Adicionar dados" na tabela dos preços (figura 5.32), caso o preço já tenha sido adicionado não aparecerá novamente o botão para realizar a adição.

Figura 5.31: Tela da tabela dos pontos de venda sem a inserção de dados.

Preço das agendas e dos calendários

Adicionar valores

Agenda Calendário

Pontos de venda das agendas e dos calendários

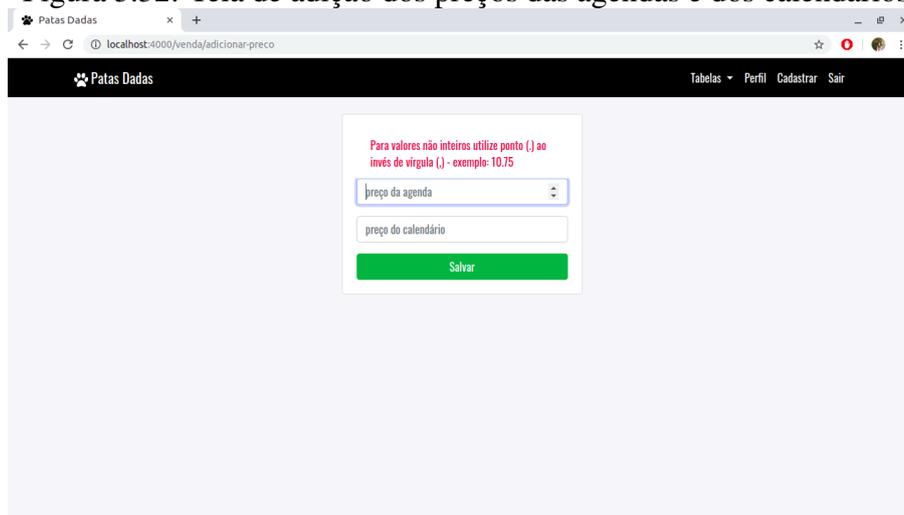
Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
-------	----------	---------	-----------	------------------------	--------	------------	-------	---------	---------	------------

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
------------	-------------------	-----------	-------	----------	---------	-----------	------------------------	--------	------------	-------	---------	---------	------------

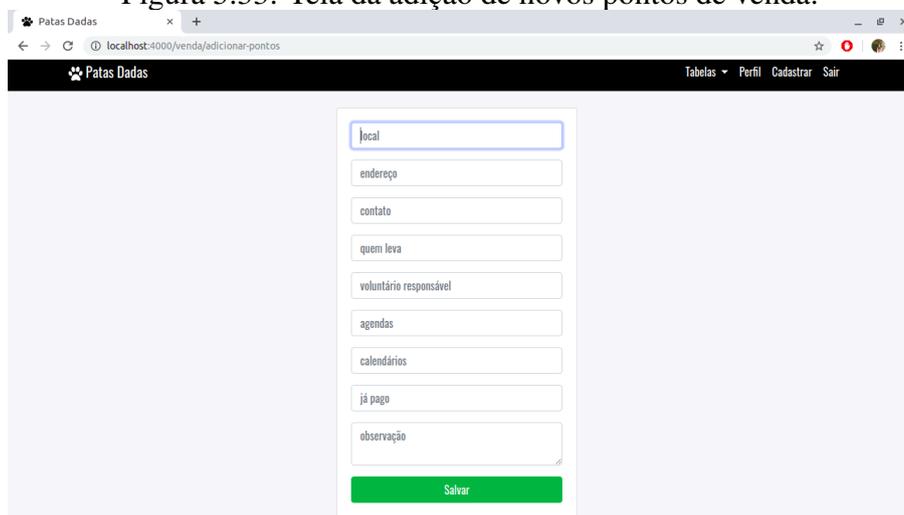
Figura 5.32: Tela de adição dos preços das agendas e dos calendários.



The screenshot shows a web browser window with the URL 'localhost:4000/venda/adicionar-preco'. The page has a dark header with the logo 'Patás Dadas' and navigation links: 'Tabelas', 'Perfil', 'Cadastrar', and 'Sair'. The main content area features a form with a red error message: 'Para valores não inteiros utilize ponto (.) ao invés de vírgula (,) - exemplo: 10.75'. Below the message are two input fields: 'preço da agenda' and 'preço do calendário', followed by a green 'Salvar' button.

Se o voluntário desejar inserir algum dado na tabela dos pontos de venda ele deve clicar no botão "Adicionar dados", para assim ser redirecionado para a página de adição dos dados (figura 5.33). Após a inserção dos dados aparecerá uma notificação de sucesso na inclusão da informação e o usuário receberá duas informações na tela: os dados que ele inseriu na tabela de pontos de venda e os dados presentes no registro de alteração da tabela, conforme a imagem 5.34 apresenta.

Figura 5.33: Tela da adição de novos pontos de venda.



The screenshot shows a web browser window with the URL 'localhost:4000/venda/adicionar-pontos'. The page has a dark header with the logo 'Patás Dadas' and navigation links: 'Tabelas', 'Perfil', 'Cadastrar', and 'Sair'. The main content area features a form with several input fields: 'local', 'endereço', 'contato', 'quem leva', 'voluntário responsável', 'agendas', 'calendários', 'já pago', and 'observação'. A green 'Salvar' button is located at the bottom of the form.

Figura 5.34: Tela das tabelas após a inserção dos dados.

Incluído com sucesso

Preço das agendas e dos calendários

Agenda	Calendário
RS10.5	RS1

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Modesta Vegana	R. Miguel Testes, 88	Laureen	Ester	50	40	R\$565	RS10	R\$555		
<b>TOTAL</b>					50	40	R\$565	RS10	R\$555	

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	18th julho 2019	adicionado	Modesta Vegana	R. Miguel Testes, 88	Laureen	Ester	50	40	R\$565	RS10	R\$555		

Para a edição dos dados o processo é semelhante, o voluntário deve clicar no botão "Editar" para ser redirecionado para a tela de edição (figura 5.35). Após a alteração dos dados o usuário será novamente encaminhado para a tabela de pontos de venda, que agora contém os dados que foram editados - figura 5.36.

Figura 5.35: Tela da edição dos pontos de venda.

Local: Modesta Vegana

Endereço: R. Miguel Testes, 215

Contato: contato

Quem leva: Laureen

Voluntário responsável: Ester

Agendas: 100

Calendários: 40

Já pago: 100

Observação: observação

Salvar

Figura 5.36: Tela da tabela de pontos de venda após a edição dos dados.

Atualizado com sucesso

Preço das agendas e dos calendários

Agenda	Calendário
RS10.5	RS1

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação	
Modesta Vegana	R. Miguel Testes, 88	Laureen	Esther	100	40	RS1090	RS10	RS1080			
<b>TOTAL</b>							100	40	RS1090	RS10	RS1080

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	16th julho 2019	editado	Modesta Vegana	R. Miguel Testes, 88	Laureen	Esther	100	40	RS1090	RS10	RS1080		

Como há produtos do Patas Dadas sendo vendidos, os voluntários precisam, além de ter os dados relativos aos pontos em si, ter um controle material e financeiro sobre os itens a serem vendidos e o dinheiro arrecadado. Na figura 5.37 é possível observar a soma dos itens vendidos e o montante arrecadado - esses valores também são salvos no banco de dados. Nota: o valor calculado pelas vendas dos produtos se dá de acordo com o valor do preço das agendas e dos calendários que são inseridos em suas tabelas.

Figura 5.37: Tela da tabela de pontos de venda após o cálculo das vendas dos produtos.

Preço das agendas e dos calendários

Agenda	Calendário
RS10.5	RS1

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação	
Modesta Vegana	R. Miguel Testes, 88	Laureen	Esther	100	40	RS1090	RS10	RS1080			
Café Riza da Praia	R. João Guimarães, 88 - Santa Cecília	Mônica B		10	5	RS110	RS0	RS110			
<b>TOTAL</b>							110	45	RS1200	RS10	RS1190

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	16th julho 2019	editado	Modesta Vegana	R. Miguel Testes, 88	Laureen	Esther	100	40	RS1090	RS10	RS1080		
Karen F	16th julho 2019	adicionado	Café Riza da Praia	R. João Guimarães, 88 - Santa Cecília	Mônica B		10	5	RS110	RS0	RS110		

Para deletar alguma informação basta o voluntário clicar no botão "Deletar" que automaticamente serão atualizadas as tabelas de pontos de venda e registros - figura 5.38.

Figura 5.38: Tela da tabela de pontos de venda após a deleção dos dados.

Removido com sucesso

Preço das agendas e dos calendários

Agenda	Calendário
RS10.5	RS1

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Café Rua da Praia	R. João Guimarães, 88 - Santa Cecília	Mônica B			10	5	RS110	RS0	RS110	
<b>TOTAL</b>					10	5	RS110	RS0	RS110	

Registros

Voluntário	Data de alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	16th julho 2019	deletado	Modesta Vegana	R. Miguel Testes, 68		Lauren	Ester	100	40	RS1090	RS10	RS1080	
Karen F	16th julho 2019	adicionado	Café Rua da Praia	R. João Guimarães, 88 - Santa Cecília	Mônica B			10	5	RS110	RS0	RS110	

Para editar o valor do preço das agendas e dos calendários basta clicar no botão "Editar"(figura 5.39). Comparando as figuras 5.40 e 5.37 é possível perceber a modificação no valor do dinheiro arrecadado pelo Patás Dadas com a venda dos materiais. E, finalmente, para deletar os valores basta clicar no botão "Deletar"(figura 5.41).

Figura 5.39: Tela da edição do valor dos preços.

Removido com sucesso

Patás Dadas

Tabelas Perfil Cadastros Sair

Para valores não inteiros utilize ponto (.) ao invés de vírgula (,) - exemplo: 10.75

Preço da agenda

10.5

Preço do calendário

7.25

Salvar

Figura 5.40: Tela da tabela de pontos de venda após a alteração do valor dos preços.

Preço das agendas e dos calendários

Agenda	Calendário
RS10.5	RS7.25

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação	
Café Rua da Praia	R. João Guimarães, 88 - Santa Cecilia	Mônica B			10	5	RS141.25	RS0	RS110		
<b>TOTAL</b>							10	5	RS141.25	RS0	RS110

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	16th julho 2019	deletado	Modesta Vegana	R. Miguel Testes, 88		Laureen	Ester	100	40	RS1340	RS10	RS1330	
Karen F	16th julho 2019	adicionado	Café Rua da Praia	R. João Guimarães, 88 - Santa Cecilia	Mônica B			10	5	RS141.25	RS0	RS110	

Figura 5.41: Tela da deleção do valor dos preços.

Preço das agendas e dos calendários

Adicionar valores

Pontos de venda das agendas e dos calendários

Adicionar dados

Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação	
Café Rua da Praia	R. João Guimarães, 88 - Santa Cecilia	Mônica B			10	5	RS0	RS0	RS110		
<b>TOTAL</b>							10	5	RS0	RS0	RS110

Registros

Voluntário	Data da alteração	Alteração	Local	Endereço	Contato	Quem leva	Voluntário responsável	Agenda	Calendário	Total	Já pago	A pagar	Observação
Karen F	16th julho 2019	deletado	Modesta Vegana	R. Miguel Testes, 88		Laureen	Ester	100	40	RS0	RS10	RS-10	
Karen F	16th julho 2019	adicionado	Café Rua da Praia	R. João Guimarães, 88 - Santa Cecilia	Mônica B			10	5	RS0	RS0	RS110	

## 6 ANÁLISE DA APLICAÇÃO

Neste capítulo serão apresentadas as duas formas utilizadas para avaliar a aplicação e a análise dos resultados obtidos. Para validar o estado atual em que o sistema se encontra foram utilizadas como formas de avaliação as heurísticas de Nielsen e uma pesquisa qualitativa realizada através de entrevista.

### 6.1 Avaliação Heurística

A primeira forma de avaliação da aplicação foi uma avaliação heurística, usando como base as heurísticas apresentadas por Nielsen (1993) e descritas na seção 2.1.18. As heurísticas são utilizadas para analisar os pontos positivos e negativos da interface de interação com o usuário. A análise heurística apresentada nesta seção foi realizada pela própria autora deste trabalho em conjunto com uma voluntária do Patas Dadas.

A partir dos dados obtidos (Tabela 6.1) é possível identificar que um dos pontos que obteve o melhor resultado foi o da heurística de equivalência do sistema com o mundo real. Além disso, outra heurística que obteve a maior pontuação foi a de consistência e padrões.

Tabela 6.1: Tabela de análise das heurísticas de Nielsen.

<i>Heurística</i>	<i>Ruim</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>Bom</i>
Visibilidade do estado do sistema		✓							
Equivalência entre o sistema e o mundo real								✓	
Liberdade e controle do usuário							✓		
Consistência e padrões								✓	
Prevenção de erro							✓		
Reconhecer ao invés de relembrar						✓			
Flexibilidade e eficiência de uso							✓		
Estética e design minimalista							✓		
Auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas								✓	
Ajuda e documentação		✓							

As demais heurísticas como liberdade e controle do usuário, prevenção de erro, flexibilidade e eficiência de uso, estética e design minimalista e auxiliar usuários a reconhecer, diagnosticar e recuperar ações erradas apresentaram uma boa avaliação (nota 6). Também é possível identificar que o sistema apresenta um baixo desempenho nas heurísticas de visibilidade do estado do sistema e de ajuda e documentação. Na aplicação

não há a visibilidade do sistema, porém é a falta de uma documentação de auxílio para o usuário que é o ponto mais crítico do sistema, visto que se o usuário necessitar de ajuda para alguma ação, o sistema não fornecerá ajuda.

## 6.2 Pesquisa Qualitativa

A segunda forma de avaliação da aplicação foi uma pesquisa qualitativa realizada através de entrevista com apenas uma voluntária da ONG Patas Dadas após a apresentação da aplicação. A entrevista realizada em 22 de junho de 2019 foi semiestruturada e com questões abertas para possibilitar o melhor entendimento da perspectiva da entrevistada (ROESCH, 2005). Como instrumento de coleta de dados foi utilizado um questionário construído com base nos objetivos deste estudo. O roteiro de entrevista está no Apêndice A.

Em virtude de cada voluntário ter a sua função dentro da ONG, apenas um voluntário estava disponível para participar ativamente deste projeto. A voluntária em questão é estudante de medicina veterinária na UFRGS. O nosso contato iniciou devido a essa voluntária ter a tarefa de responder as mensagens enviadas via redes sociais ou e-mail. Foi ela a responsável por repassar os problemas enfrentados pela ONG, explicar o modelo de sistema desejado pelos voluntários e testar e avaliar este projeto.

Os dados coletados para avaliar a aplicação tem como objetivos identificar os pontos apresentados na tabela 6.2. A elaboração desses pontos foi baseada em: identificar a existência de um problema enfrentado pelo Patas Dadas, descobrir qual a maior dificuldade existente para resolver esse problema e se o sistema desenvolvido conseguiu resolver o problema.

Tabela 6.2: Tabela do plano de coleta de dados.

<b>Objetivos da entrevista</b>
Identificar a principal dificuldade enfrentada sem o uso da aplicação
Identificar o principal ponto positivo da aplicação
Identificar qual é o impacto da aplicação na ONG Patas Dadas

Abaixo serão apresentados e analisados os dados obtidos através da entrevista.

- *Identificação da principal dificuldade enfrentada sem o uso da aplicação:* dificuldade para fazer a gestão de tantas informações presentes no gerenciamento dos dados relativos aos pontos de venda, colaboradores, coletas, estoques e eventos. O

Excel é a principal ferramenta utilizada para o armazenamento e organização da informações, porém existe a dificuldade de centralizar os dados em um mesmo lugar ou na mesma planilha devido a diversidade de informações existentes.

- *Identificação do principal ponto positivo da aplicação:* o registro de movimentação dos dados, cada dado alterado por um usuário é armazenado e exibido na tela por 30 dias (número definido pelos voluntários), com isso os voluntários têm o controle exato da autenticidade das informações, visto que no modelo antigo os dados não possuem fonte de origem e portanto não são tidos como confiáveis pelos voluntários.
- *Identificação do impacto da aplicação na ONG Patas Dadas:* o principal impacto da aplicação foi identificado como a facilitação da organização de informações, visto que elimina a necessidade do uso de diversas planilhas de Excel para organizar os dados, tornando a gestão financeira e organizacional da ONG mais eficiente.

## 7 CONCLUSÃO

Este trabalho teve como principal objetivo o desenvolvimento de uma aplicação Web de gerenciamento de dados relativos aos materiais utilizados - pelo Patas Dadas - a serem coletados e aos materiais presentes em estoque, na administração dos eventos em que a ONG participa, dos pontos de venda existentes e na listagem dos colaboradores. Outros objetivos apresentados foram: facilitar a organização de informações e a consulta de dados de forma prática.

Em relação ao objetivo principal de desenvolvimento da aplicação Web, o mesmo foi atingido, visto que a aplicação foi desenvolvida conforme apresentado no capítulo 5. A tabela 7.1 agrupa novamente as aplicações que possuem funcionalidades e design que influenciaram o desenvolvimento deste trabalho, porém agora com o incremento do conjunto de funcionalidades que o sistema desenvolvido aqui cobre.

Tabela 7.1: Tabela de análise comparativa completa.

<b>Funcionalidades/Aplicação</b>		<i>Meu Dinheiro</i>	<i>Evernote</i>	<i>Jira Software</i>	<i>Patas</i>
Distribuição	Versão gratuita	✓	✓		✓
	Versão paga	✓	✓	✓	
	Versão web	✓	✓	✓	✓
	Versão móvel	✓	✓		✓
Gerência	Fluxo de caixa	✓			✓
	Contas a pagar	✓			
	Controle de receitas	✓			✓
	Anotações		✓	✓	✓
	Gerenciamento de projetos		✓	✓	
Visualização	Criação de perfil	✓	✓	✓	✓
	Modo offline		✓		
	Grupo de usuários	✓	✓	✓	✓

### 7.1 Aspectos Técnicos

As tecnologias adotadas para a construção deste trabalho apresentaram um resultado satisfatório, muitas dessas tecnologias já eram familiares à autora deste trabalho e assim foram selecionadas afim de minimizar as dificuldades na execução do projeto, como, por exemplo, a opção por um ambiente Web utilizando HTML e Javascript.

## **7.2 Limitações do Trabalho**

Uma limitação presente neste trabalho foi o desenvolvimento feito com base apenas nas necessidades dos voluntários do Patas Dadas. Além disso, durante a realização deste trabalho os testes foram realizados com apenas uma voluntária da ONG para avaliar a qualidade da aplicação.

A principal limitação encontrada no desenvolvimento deste projeto foi a conexão com o banco de dados utilizado pela aplicação Web atual da ONG, a unificação dos bancos de dados não foi possível de ser realizada visto que o banco utilizado pelo Patas Dadas está em reformulação e não ficou pronto até a conclusão deste trabalho.

## **7.3 Trabalhos Futuros**

Mesmo considerando que a construção desta aplicação foi concluída com êxito, existem outras funcionalidades que podem ser implementadas em versões futuras - visto que o projeto apresentado neste trabalho representa uma etapa inicial do sistema - como, por exemplo, a unificação do banco de dados, a criação de um ambiente para os colaboradores da ONG Patas Dadas e a criação de mais tabelas para, por exemplo, auxiliar na organização referente aos animais adotados pela ONG.

## REFERÊNCIAS

COHN, M. **User Stories Applied: For Agile Software Development**. [S.l.]: Addison Wesley, 2004.

FOWLER, M. **Agile Software Development**. 2015. Disponível em: <<https://martinfowler.com/agile.html>>. [Online: acessado 10 de Julho de 2019].

HELABS. **O que é Scrum?** 2013/2014. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. [Online: acessado 5 de Abril de 2019].

NIELSEN, J. **Usability Engineering**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 9780080520292.

NIELSEN, J.; MOLICH, R. **Heuristic evaluation of user interfaces**. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 1990. (CHI '90), p. 249–256.

ROESCH, S. M. A. **Projeto de estágio e de pesquisa em administração: guia para estágios, trabalhos de conclusão dissertações e estudos de caso**. São Paulo: Atlas, 2005.

DEACON, J. Model-view-controller (mvc) architecture. **JOHN DEACON Computer Systems Development, Consulting Training**. Aug 1995.

PROTO.IO. **Proto.io**. 2019. <<https://proto.io/>>. [Online: acessado 5 de Abril de 2019].

MOZILLA. **HTML: HyperText Markup Language**. 2019. <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. [Online: acessado 5 de Maio de 2019].

FOUNDATION jQuery. **What is jQuery?** 2019. <<https://jquery.com>>. [Online: acessado 5 de Maio de 2019].

MOZILLA. **CSS**. 2019. <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. [Online: acessado 5 de Maio de 2019].

BOOTSTRAP. **About Bootstrap**. 2019. <<https://getbootstrap.com/docs/4.3/getting-started/introduction/>>. [Online: acessado 5 de Maio de 2019].

- MOZILLA. **JavaScript**. 2019. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. [Online: acessado 5 de Maio de 2019].
- COYIER, C. **What is the DOM?** 2013. <<https://css-tricks.com/dom/>>. [Online: acessado 5 de Maio de 2019].
- FRANKLIN, A. **Tableless - Tenha o DOM**. 2011. <<http://tableless.com.br/tenha-o-dom/>>. [Online: acessado 5 de Maio de 2019].
- NODEJS. **NodeJS**. 2019. <<https://nodejs.org/api/>>. [Online: acessado 5 de Maio de 2019].
- NPM. **NPM - Node Package Manager**. 2019. <<https://docs.npmjs.com/>>. [Online: acessado 5 de Maio de 2019].
- EXPRESS. **Express**. 2019. <<https://www.npmjs.com/package/express>>. [Online: acessado 5 de Maio de 2019].
- HALE, C. **How To Safely Store A Password**. 2011. <<https://codahale.com/how-to-safely-store-a-password/>>. [Online: acessado 5 de Maio de 2019].
- HANSON, J. **Passport**. 2015. <<http://www.passportjs.org/docs/>>. [Online: acessado 5 de Maio de 2019].
- NODEMON. **Nodemon**. 2019. <<https://www.npmjs.com/package/nodemon>>. [Online: acessado 5 de Maio de 2019].
- COMMONJS. **CommonJS**. 2019. <<https://www.npmjs.com/package/common-js>>. [Online: acessado 5 de Maio de 2019].
- NIGHTWATCH. **Nightwatch**. 2019. <<https://www.npmjs.com/package/nightwatch>>. [Online: acessado 5 de Maio de 2019].
- SOMMERVILLE, I. **Software Engineering: Seventh Edition**. Pearson Education Publisher, 2004.
- CHUNG, L.; NIXON, B.A.; YU, E.; MYLOPOULOS, J. **Non-Functional Requirements in Software Engineering**. Springer US Publisher, 2000.
- MySQL. **MySQL Documentation**. 2019. <<https://dev.mysql.com/doc/>>. [Online: acessado 5 de Maio de 2019].

SHINDE, V.; DESYATNIKOV, R.; SEELA, S. **Software Testing Help**. 2019.  
<<https://www.softwaretestinghelp.com/database-management-software/>>. [Online:  
acessado 5 de Maio de 2019].

TABLELESS. **O que é client-side e server-side?** 2012.  
<<http://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html>>.  
[Online: acessado 5 de Maio de 2019].

## **APÊNDICE A — ROTEIRO DA ENTREVISTA**

1. Qual a principal dificuldade enfrentada sem o uso da aplicação?
2. Qual o principal ponto positivo identificado através do uso da aplicação?
3. Qual a principal melhoria que o desenvolvimento dessa aplicação traz para o Patas Dadas?