

**INSTITUTO FEDERAL GOIANO  
CAMPUS MORRINHOS**

**Brunno Henrique Campos Inácio**

**Calf Manager - Sistema para gerenciamento da produção de  
bezerros**

**Morrinhos  
2020**

BRUNNO HENRIQUE CAMPOS INÁCIO

Calf Manager - Sistema para gerenciamento da produção de  
bezerros

**Trabalho de conclusão de curso apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Goiano - Campus Morrinhos, como requisito parcial para a obtenção do título de Tecnólogo.**

Morrinhos, Março de 2020.

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**Sistema Integrado de Bibliotecas – SIBI/IF Goiano Campus Morrinhos**

I45c Inácio, Brunno Henrique Campos.  
Calf Manager - sistema para gerenciamento da produção de bezerros. /  
Brunno Henrique Campos Inácio. – Morrinhos, GO: IF Goiano, 2020.  
74 f. : il. color.

Orientador: Luciano Carlos Ribeiro da Silva.  
Coorientadora: Aline Sousa Camargos

Trabalho de conclusão de curso (tecnólogo) – Instituto Federal Goiano  
Campus Morrinhos, Curso de Tecnologia em Sistemas para Internet, 2020.

1. Tecnologia 2. Software. 3. Agronegócio e gestão. I. Silva, Luciano  
Carlos Ribeiro da. II. Camargos, Aline Sousa. III. Instituto Federal Goiano.  
IV. Título.

CDU 004.4

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

**Identificação da Produção Técnico-Científica**

- |  |   |
|--|---|
| <input type="checkbox"/> Tese                          | <input type="checkbox"/> Artigo Científico              |
| <input type="checkbox"/> Dissertação                   | <input type="checkbox"/> Capítulo de Livro              |
| <input type="checkbox"/> Monografia – Especialização   | <input type="checkbox"/> Livro                          |
| <input checked="" type="checkbox"/> TCC - Graduação    | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional | - Tipo:   |

Nome Completo do Autor: Brunno Henrique Campos Inácio  
Matrícula: 2015104211710098  
Título do Trabalho: Calf Manager - Sistema para gerenciamento da produção de bezerros.

**Restrições de Acesso ao Documento**

Documento confidencial:  Não  Sim, justifique: \_\_\_\_\_

Informe a data que poderá ser disponibilizado no RIIF Goiano: 30/06/2020

O documento está sujeito a registro de patente?  Sim  Não  
O documento pode vir a ser publicado como livro?  Sim  Não

**DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA**

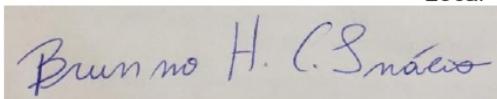
O/A referido/a autor/a declara que:

- o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Morrinhos - GO, 29/06/2020.

Local

Data



Assinatura do Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

LUCIANO CARLOS RIBEIRO  
DA SILVA:92141730104

Assinado digitalmente por LUCIANO CARLOS RIBEIRO DA SILVA:92141730104  
DN: cn=B.R., o=CP-Brasil, ou=Autentidade Certificadora R&B Brasília V2, ou=IC  
@LUTTI, o=IC@SO LUTTI, ou=IC, ou=20061647000195, o=Certificado PF A1,  
c=BR, ou=LUCIANO CARLOS RIBEIRO DA SILVA:92141730104  
Resolução: Diretor Geral - Port. 103 DO U 2006/2020  
Localizador: Morrinhos, GO  
Data: 2020-07-16 14:20:03

Assinatura do(a) orientador(a)



Serviço Público Federal  
Ministério da Educação  
Instituto Federal Goiano Campus Morrinhos  
Curso Superior de Tecnologia em Sistemas para Internet

### ATA DA APRESENTAÇÃO DO TRABALHO DE CURSO

Aos **16** dias do mês de **março** de **2020**, às **17** horas, foi realizada, nas dependências do Instituto Federal Goiano – campus Morrinhos, a apresentação pública do trabalho de curso do discente **Brunno Henrique Campos Inácio** intitulado **Calf Manager Sistema para Gerenciamento da Produção de Bezerros**, como requisito necessário para a conclusão do curso. A Banca Examinadora, constituída pelos professores: **Prof. Me Luciano Carlos Ribeiro da Silva** orientador, **Profa Dra. Aline Sousa Camargos** coorientadora, **Dr. Jesmmer da Silveira Alves**. Após a análise, emitiram o seguinte resultado:

- (X) Aprovado
- ( ) Aprovado com ressalva
- ( ) Reprovado com o seguinte parecer:

Por ser verdade firmamos a presente, Morrinhos 16 de março de 2020.

  
**Prof. Me Luciano Carlos Ribeiro da Silva** (Presidente da banca)

  
**Profa. Dra. Aline Sousa Camargos** (Coorientadora)

  
**Prof. Dr. Jesmmer da Silveira Alves** (Membro)



# Agradecimentos

Primeiramente à Deus, por ter me dado saúde e força para superar as dificuldades. A esta instituição de ensino, seu corpo docente, direção e administração que oportunizaram a janela que hoje, vislumbro um horizonte superior. Ao meu orientador Luciano Carlos Ribeiro da Silva e à coorientadora Aline Sousa Camargos, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos. À minha namorada Joslaine de Sá Guimarães Silva, aos meus pais e ao meu irmão pelo amor, incentivo e apoio incondicional. Aos meus amigos de faculdade que sempre estavam dispostos a me ajudar nos momentos em que mais precisei. E, por fim, a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.



# Resumo

No Brasil, o agronegócio, assim como outros setores, está em processo de evolução. Suas características familiares, estão se transformando para adequar-se a uma estrutura empresarial, adotando-se formas de gerenciamento mais controladoras, demandando tecnologias de informação com foco na gestão dos seus negócios. O objetivo deste trabalho foi o desenvolvimento de um *software* para o gerenciamento da produção de bezerros. Nele poderão ser gerenciados de forma individual os animais, seu histórico genealógico, pesagens, exames laboratoriais, controle de doenças, sanidade, controle de medicações ou vacinas, utilizando a plataforma *web* com total segurança. Por fim, o *software* atuará como uma ferramenta de grande auxílio do processo de tomada de decisões.

**Palavras-chave:** Tecnologia, Software, Agronegócio e Gestão.



# Abstract

In Brazil, agribusiness has been eluting, leaving its family resources to adapt to a business structure, adhering to a more controlling form of management, requiring information technologies focused on the management of its business. The objective of this project was the development of software for the management of calf production. It can be managed individually for animals, their genealogical history, weighing, laboratory tests, disease control, health, medication or vaccine control, using a web platform with total security. Finally, the software used as a tool of great help in the decision-making process.

**Keywords:**Technology, Software, Agribusiness and Management.



# Lista de ilustrações

Figura 1 – Diagrama de arquitetura REST. . . . .	32
Figura 2 – Diagrama de casos de uso. . . . .	37
Figura 3 – Significado dos objetos. . . . .	38
Figura 4 – Diagrama de classes. . . . .	39
Figura 5 – Diagrama de banco de dados. . . . .	40
Figura 6 – Tipos de relacionamentos. . . . .	40
Figura 7 – Diagrama de arquitetura. . . . .	43
Figura 8 – Apresentação das classes Cargo, CargoDao e CargoEntity. . . . .	44
Figura 9 – Tela de cadastro de fazenda. . . . .	48
Figura 10 – Tela de listagem de fazendas. . . . .	48
Figura 11 – Tela de cadastro de lotes. . . . .	49
Figura 12 – Tela de listagem de lotes. . . . .	49
Figura 13 – Tela de cadastro de doenças. . . . .	50
Figura 14 – Tela de listagem de doenças. . . . .	50
Figura 15 – Tela de cadastro de animais. . . . .	51
Figura 16 – Tela de listagem de animais. . . . .	52
Figura 17 – Tela do prontuário (parte 01). . . . .	53
Figura 18 – Tela do prontuário (parte 02). . . . .	53
Figura 19 – Tela do prontuário (parte 03). . . . .	53
Figura 20 – Janela de registro de adoecimento do animal. . . . .	54
Figura 21 – Janela de registro de medicação/vacinação no animal. . . . .	54
Figura 22 – Janela de registro de cura do animal. . . . .	55
Figura 23 – Janela de registro de morte do animal. . . . .	55
Figura 24 – Tela de cadastro de hemogramas. . . . .	56
Figura 25 – Gráficos de resultados de testes de hemogramas. . . . .	56
Figura 26 – Tela de listagem de hemogramas. . . . .	57
Figura 27 – Tela de cadastro de pesagens. . . . .	57
Figura 28 – Tela de relatório de pesagem. . . . .	58
Figura 29 – Tela de cadastro de medicamentos/vacinas. . . . .	59
Figura 30 – Tela de listagem de medicamentos/vacinas. . . . .	59
Figura 31 – Tela de entrada de medicamentos/vacinas. . . . .	60
Figura 32 – Tela de cadastro de aplicação de dosagem. . . . .	60
Figura 33 – Tela de listagem de aplicações. . . . .	61
Figura 34 – Tela de cadastro de funcionários. . . . .	62
Figura 35 – Tela de listagem de funcionários. . . . .	62
Figura 36 – Tela de cadastro de cargos. . . . .	63

Figura 37 – Tela de listagem de cargos. . . . .	63
Figura 38 – Tela de cadastro de grupos. . . . .	64
Figura 39 – Tela de listagem de grupos. . . . .	64
Figura 40 – Tela de cadastro de usuários. . . . .	65
Figura 41 – Tela de listagem de usuários. . . . .	65
Figura 42 – Tela de cadastro de permissões. . . . .	66
Figura 43 – Tela de listagem de permissões. . . . .	67
Figura 44 – Dashboard. . . . .	67
Figura 45 – Tela de login. . . . .	68

# Lista de tabelas

Tabela 1 – Objetivos estratégicos do cliente. . . . .	25
Tabela 2 – Premissas que podem afetar o projeto. . . . .	26
Tabela 3 – Requisitos do cliente. . . . .	28
Tabela 4 – Requisitos funcionais. . . . .	29
Tabela 5 – Descrição das tabelas do banco de dados. . . . .	42



# Lista de abreviaturas e siglas

PC – *Personal Computer.*

PPT – *Proteína Plasmática Total.*

MVC – *Model View Controller.*

REST – *Representational State Transfer.*

MYSQL – *Gerenciador de Banco de Dados.*

HTTP / HTTPS – *Hipertext Transfer Protocol / Hipertext Transfer Protocol Security.*

PHP – *Personal Home Page.*

ORM – *Object Relational Mapper.*

DOM - *Document Object Model.*

HTML - *Hipertext Markup Language.*

CSS – *Cascating style ship.*

DAO – *Data Acess Object.*

JSON – *Javascript Object Notation.*

API – *Application Programing Interface.*

ml – *Mililitros.*

Mg – *Miligramas.*

RG – *Registro Geral.*

TC – *Trabalho de Curso.*

CPF – *Cadastro de Pessoa Física.*

CRUD – *Create, Retreave, Update and Delete.*

RH – *Recursos Humanos.*

UML - *Unified Modeling Language.*

JWT – *Json Web Token.*

SGDB – Sistemas de Gestão de Base de Dados.

SQL - *Structured Query Language*.

DOM – *Document Object Model*.

HMAC - *Hash-based Message Authentication Code*.

PIB - Produto Interno Bruto.

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
1.1	Objetivos	21
1.2	Objetivos específicos	22
1.3	Estrutura do trabalho	22
<b>2</b>	<b>PROJETO</b>	<b>23</b>
2.1	Contexto	23
2.2	Levantamento de requisitos	25
2.3	Tecnologias utilizadas	30
2.3.1	PHP	30
2.3.2	MySQL	31
2.3.3	MVC	31
2.3.4	REST	32
2.3.5	<i>Frameworks</i>	33
2.3.6	Laravel	33
2.3.7	Slim	34
2.3.8	Valitron	34
2.3.9	Javascript	34
2.3.10	Vue.js	34
2.3.11	Vuetify	35
2.3.12	Axios	35
2.3.13	Echarts	35
2.3.14	JWT	36
<b>3</b>	<b>MODELAGEM DO SISTEMA</b>	<b>37</b>
3.1	Casos de uso	37
3.2	Descrição dos casos de uso	38
3.3	Diagrama de classes	39
3.4	Diagrama de banco de dados	39
3.5	Descrição das tabelas do banco de dados	41
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>43</b>
4.1	Arquitetura	43
4.2	Arquitetura <i>back-end</i>	44
4.3	Arquitetura <i>front-end</i>	45
4.4	Integração <i>back-end e front-end</i>	46

<b>5</b>	<b>APRESENTAÇÃO DA SOLUÇÃO</b>	<b>47</b>
<b>5.1</b>	<b>Módulos</b>	<b>47</b>
5.1.1	Módulo fazendas	47
5.1.2	Módulo lotes	48
5.1.3	Módulo doenças	50
5.1.4	Módulo animais	51
5.1.5	Módulo enfermaria	52
5.1.6	Módulo hemogramas	55
5.1.7	Módulo pesagens	57
5.1.8	Módulo medicamentos/vacinas	58
5.1.9	Módulo funcionários	61
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>69</b>
<b>7</b>	<b>TRABALHOS FUTUROS</b>	<b>71</b>
	<b>REFERÊNCIAS</b>	<b>73</b>

# 1 Introdução

O agronegócio vem ganhando destaque na economia brasileira, sendo um dos setores com participação significativa na formação do Produto Interno Bruto (PIB), na geração de empregos e exportação. De forma semelhante, a agroindústria vem se transformando, demandando das propriedades rurais e associações, um maior nível de profissionalismo no âmbito da gestão de seus negócios (YAMAGUCHI et al. 2016).

Segundo (MACHADO; NANTES; ROCHA, 2002a apud RODRIGUES, 1999), a informática tem papel fundamental na gestão de processos administrativos. No ambiente rural, tem grande importância na competitividade do setor, elevando a eficiência do processo produtivo como um todo.

Neste ambiente, as propriedades rurais, no intuito de obter maior controle, planejamento e expansão de sua competitividade no mercado, necessitam da incorporação de sistemas computacionais que permitam o maior controle da produção e ofereçam informações de qualidade para o auxílio na tomada de decisões (MACHADO; NANTES; ROCHA, 2002b apud SOUKI; XISTO; SALAZAR, 1999).

Assim, os formatos manuais de registro e controle dos dados dos rebanhos bovinos, geralmente armazenados em formulários de papel ou planilhas eletrônicas, dificultam ou até mesmo não permitem ao produtor um acompanhamento preciso das informações acerca dos seus animais. Ademais, consultar de forma eficaz essas informações pode vir a ser um processo trabalhoso, desgastante e, na pior das hipóteses, induzir a tomada de decisão errada.

## 1.1 Objetivos

Dado o problema, esse trabalho visa desenvolver uma solução *web* para o acompanhamento e gerenciamento de bezerras, seu histórico genealógico, pesagens, exames laboratoriais, controle de doenças, sanidade, medicações ou vacinas, desde o nascimento até o desmame ou morte, de acordo com o prazo preestabelecido.

Por meio dela, a saber, da solução *web*, todo o gerenciamento das informações dos animais e suas particularidades estarão disponíveis através de um único sistema, sendo acessado a partir de qualquer local que possua conexão com internet.

## 1.2 Objetivos específicos

- Prover por meio do *software* uma plataforma central de gerenciamento das informações.
- Desenvolver um sistema responsivo para atender aos usuários que acessem a partir de diversos tipos de dispositivos via *web* (*smartphone*, *tablet*, computador, etc).
- Controlar o acesso de informações por meio de autenticação de usuário.

## 1.3 Estrutura do trabalho

A estrutura deste trabalho é composta por sete capítulos. O Capítulo 2 apresenta a transcrição adaptada do artigo "Protótipo de *Software* para Gerenciamento de Produção de Bezerros", conforme as regras de Trabalho de Curso (TC) do Curso de Tecnologia de Sistemas para Internet do Instituto Federal Goiano Campus Morrinhos. Nele, é apresentada a análise do negócio, o projeto do sistema *web* e as tecnologias utilizadas, abordando uma visão generalizada sobre o ambiente e seus requisitos. O Capítulo 3 apresenta as modelagens do *software* utilizadas como plano estrutural para o desenvolvimento da aplicação e suas particularidades de regras de negócio, os requisitos elucidados e a apresentação da solução. O Capítulo 4 apresenta a metodologia e arquitetura utilizada no desenvolvimento. O Capítulo 5 apresenta a solução a partir de suas telas interativas. O Capítulo 6 apresenta as considerações finais desse trabalho. O Capítulo 7, por fim, apresenta a recomendação para trabalhos futuros.

## 2 Projeto

Este capítulo apresenta a transcrição adaptada do artigo "Protótipo de Software para Gerenciamento de Produção de Bezerros", publicado na Revista *Colloquium Agrariae*. Nele, são retratados o contexto e especificação de funcionalidades através do levantamento de requisitos, expondo as necessidades por meio de objetivos, premissas, requisitos, ferramentas e tecnologias utilizadas no desenvolvimento. A seguir, a Seção 2.1 apresenta o contexto do projeto; a Seção 2.2 apresenta o levantamento de requisitos; por fim, a Seção 2.3 apresenta as tecnologias utilizadas.

### 2.1 Contexto

O controle da produção de bezerros envolve a coleta e análise de diversos dados atinentes. Essas informações, que incluem detalhes de hereditariedade, anamnese, engorda, desmama e rastreamento, precisam estar consistentemente organizadas e altamente disponíveis para os técnicos responsáveis pelo controle dos animais.

Além disso, a consolidação das informações deve permitir análises individuais, temporais e dos lotes de bezerros pelos gestores da empresa. Esse trabalho culmina no desenvolvimento de um aplicativo de computador capaz de receber, gerenciar e recuperar essas informações.

O agronegócio brasileiro tem enfrentado diversas mudanças, tanto no âmbito nacional quanto no internacional. Essas alterações, como a abertura comercial, que insere as fazendas, agora denominadas empresas que atuam em mercados dos mais diversos, dentro de um vulto de competição acirrada, no qual a eficácia e eficiência da produção, aliados com a qualidade final do produto, são determinantes para a ampliação da competitividade e o conseqüente sucesso do negócio (MONTROYA e ROSSETTO 2002).

O sistema de produção de bezerros deita-se sobre três pilares: o manejo sanitário, o nutricional e o melhoramento genético. A coleta e a avaliação criteriosa dos dados concernentes ao desenvolvimento na fase de cria são definidoras do sucesso do rebanho (OLIVEIRA, ZANINE e SANTOS 2007).

Nesse sentido, o melhoramento bovino, assim como é peculiar das atividades rurais que agregam tecnologia, exige controles que possibilitem aos gestores visões multifacetadas sobre os dados coletados. No fim, os gestores técnicos desejam ser subsidiados com dados que permitam analisar, diagnosticar e traçar planos de ação de forma acertada (CARRIERI e Lima 1992).

Por essa razão, um sistema computacional que vise suportar a tomada de decisão na

produção de bezerros deverá possibilitar a coleta de dados que permita o cálculo de índices zootécnicos, tais como a fertilidade de novilhas, fertilidade de primíparas, fertilidade de multíparas, perda pré-parto, mortalidade de bezerros, taxa de desmame, hereditariedade, etc (NETO 1999).

Pela facilidade de uso, é comum encontrarmos os dados de gerenciamento de bezerros em planilhas eletrônicas. Contudo, o uso de arquivos para o cadastro de grande volume de dados, especialmente aqueles que apresentam estrutura e relacionamentos complexos, é desaconselhado (ELMASRI et al. 2005).

Os sistemas de banco de dados permitem a manutenção das informações garantindo sua integridade e consistência. Também facilitam a escalabilidade, acesso simultâneos e alta disponibilidade das informações (SUNDARSHAN, SILBERSCHATZ e KORTH 1999).

Ademais, uma vez assentada uma estrutura normalizada de armazenamento das informações, permite a criação de relatórios gerenciais e o cruzamento de dados não diretamente relacionados. Por fim, cabe salientar que os sistemas de banco de dados também possibilitam a programação sistemática de cópias de segurança, a minimização da redundância e o acesso seguro e controlado das informações (ELMASRI et al. 2005).

Já o uso de aplicativos de interface humano-computador, permite a criação de um ambiente intuitivo, de fácil utilização e que minimize erros no momento da inclusão dos dados. Sublinha-se ainda que os aplicativos também possibilitam a criação de relatórios e recursos gráficos dinâmicos e interativos, permitindo diversas modalidades de análise dos dados.

O objetivo deste trabalho foi a criação de um programa de computador que possibilitará, às empresas que necessitam gerenciar a produção de bezerros, o acesso a um aplicativo gratuito, de código aberto e capaz de oferecer o controle necessário para tomada de decisão.

O programa permite a inserção das informações relacionadas ao controle de bezerros e foi construído em observância da simplicidade de interação e asserção no cadastramento dos dados.

Além disso, utiliza uma linguagem de programação multiplataforma compatível com computadores do tipo PC (aplicativo *desktop*) e funciona em rede de computadores (modelo cliente-servidor). Por fim, o aplicativo também gera relatórios elementares que permitam a correta análise tanto dos dados consolidados quanto dos individualizados (lote/bezerros).

## 2.2 Levantamento de requisitos

O estudo da demanda foi conduzido em parceria com a empresa de biotecnologia da reprodução Samvet Embriões, do município de Morrinhos, GO, que forneceu o apoio técnico no processo de levantamento das regras do negócio e do levantamento de requisitos do sistema.

Desta forma, foram realizadas tanto uma revisão da literatura correlata como também entrevistas com os técnicos da Samvet para se definir o conjunto de dados, as necessidades e as regras do negócio necessário ao desenvolvimento do sistema.

Para idealizarmos este projeto, realizamos a coleta de requisitos que envolvem todas as necessidades de informatização do negócio. Requisitos estes que seguem por padrão atender os objetivos estratégicos do negócio. Na Tabela 1, visualizamos os objetivos que serão atendidos pelo projeto, de acordo com os papéis dos colaboradores.

Colaboradores	Objetivos estratégicos
Proprietário	<ol style="list-style-type: none"> <li>1. Acompanhará os resultados dos manejos de acordo com os relatórios.</li> <li>2. Cadastrará funcionários e seus cargos.</li> <li>3. Cadastrará animais e suas características relevantes ao sistema.</li> <li>4. Controlará estoque de medicamentos.</li> <li>5. Gerenciará os registros de morte dos animais.</li> <li>6. Gerenciará os registros de morte ao nascer (natimorto).</li> </ol>
Técnico de Laboratório	<ol style="list-style-type: none"> <li>1. Realizará os testes clínicos nos animais de acordo com o lote ao qual pertencem.</li> <li>2. Cadastrará os resultados dos testes realizados.</li> <li>3. Acompanhará o resultado e o histórico dos testes clínicos.</li> </ol>
Técnico de Campo	<ol style="list-style-type: none"> <li>1. Realizará a pesagem dos animais de acordo com lote ao qual pertencem.</li> <li>2. Cadastrará o peso.</li> <li>3. Cadastrará os animais por lote e fazenda.</li> <li>4. Cadastrará a ocorrência de morte de animais</li> </ol>

Tabela 1 – Objetivos estratégicos do cliente.

A partir dos objetivos, idealizamos a informatização de todo o controle do manejo realizado nos animais desde o cadastro do animal, incorporando os animais nascidos, sendo relacionados pelo lote. O controle se dará pelos procedimentos de manejo realizados, sendo inclusos: as funcionalidades que auxiliarão no controle de peso, procedimentos clínicos de teste de Proteína Plasmática Total (PPT) e hematócrito, controle de natalidade, mortalidade e natimorto.

Para o controle gerencial da fazenda, o sistema recebe informações sobre o manejo em geral como: mortalidade e natalidade dos animais, o constante cadastramento de peso, testes

de PPT e hematócrito nos animais. Estes bezerros são separados por lotes, apresentando essas informações gerenciadas através dos relatórios resultantes do processamento destes dados.

A informatização no meio rural sofre por dificuldades tecnológicas que afetam o desenvolvimento. Uma das premissas que afetam o desenvolvimento do projeto está ligada a um fator de grande importância, o acesso à internet no meio rural. Observando a possibilidade de frequentes interrupções, arquitetamos a solução a seguir.

De acordo com a estrutura de telecomunicação presente na propriedade, podem ser negociadas a possibilidade de implantação do sistema em servidores locais ou em nuvem, ficando a escolha dos gestores e colaboradores.

Nos casos de interrupção no fornecimento de energia elétrica no servidor local em que o sistema estiver implantado, geradores e baterias deverão sustentar o sistema *online* até o retorno do fornecimento. A Tabela 2 apresenta o estudo do caso do rompimento elétrico e de internet.

Premissas	Grau	Plano de Ação	Responsável	Situação atual
A falta de recursos de rede, principalmente a conexão com a internet, provindos de algum problema técnico no provedor ou fatores externos ocorridos na fazenda.	Alta	O <i>software</i> deverá ser estruturado para ser integrado por aplicações <i>mobile</i> , pois um aplicativo móvel alocado em um <i>smartphone</i> poderá prover acesso ao <i>stakeholder</i> para transacionar informações para o servidor, via internet, independente de seu local de hospedagem.	Provedor de serviços e departamento de assistência informática.	Concluído
A falta de recursos elétricos, em casos de falta de energia na fazenda.	Alta	A infraestrutura deverá conter equipamentos <i>nobreaks</i> para manter o sistema ainda ativo para ser acessado nos momentos de queda de energia.	Departamento de assistência informática e elétrica.	Pendente

Tabela 2 – Premissas que podem afetar o projeto.

O sistema em si visa atender as necessidades de controle da produção de bezerras. Essas necessidades pontuamos como requisitos a serem atendidos como uma só solução. Através dos objetivos dos colaboradores, podemos estabelecer quais serão as características do sistema, quais serão os requisitos que a solução abordará e suas características e particularidades. A Tabela 3 apresenta os requisitos do cliente de acordo com o que foi levantado conjuntamente com os colaboradores.

Para atender os objetivos, o sistema deverá ter funções específicas que são tipificadas de requisitos funcionais, ou seja, são funções que farão parte obrigatória do sistema para que ele atenda às necessidades a serem solucionadas. Afim de suprir as necessidades, ou seja requisitos funcionais que o sistema deverá realizar em sua execução. Na Tabela 4, os requisitos estão especificados de acordo com os requisitos do cliente.

ID	Requisitos do Cliente	Objetivo(s) de Colaboradores relacionado(s).
RC01	O proprietário deseja realizar o cadastro da fazenda onde é realizado o manejo dos animais.	1.1
RC02	O proprietário deseja realizar o cadastro da fazenda onde é realizado o manejo dos animais.	1.2
RC03	O proprietário e o técnico de campo desejam cadastrar o lote de animais.	1.3 e 3.1
RC04	O proprietário e o técnico de campo desejam cadastrar os animais.	1.4 e 3.2
RC05	O proprietário e o técnico de campo desejam cadastrar as vacinas que serão aplicadas nos animais quando necessário.	1.5 e 3.3
RC06	O proprietário e o técnico de campo desejam cadastrar os animais que morrem.	1.6, 1.8, 3.5 e 3.6
RC07	O proprietário e o técnico de campo desejam cadastrar os animais que nascem na fazenda.	1.7 e 3.5
RC08	O proprietário e o técnico de campo desejam cadastrar os animais que nascem mortos	1.8 e 3.6
RC09	O proprietário e o técnico de campo desejam cadastrar os dados do controle de ganho de peso dos animais.	1.9, 3.7 e 3.8
RC10	O proprietário e o técnico de campo desejam acompanhar o relatório dos animais residentes na fazenda.	1.10 e 3.9
RC11	O proprietário deseja acompanhar os relatórios de animais que nascem na fazenda.	1.11
RC12	O proprietário deseja acompanhar os relatórios de animais que morrem na fazenda.	1.12
RC13	O proprietário e o técnico de laboratório desejam acompanhar os relatórios de doenças que atacam a saúde dos animais.	1.13 e 2.4
RC14	O proprietário e o técnico de campo desejam acompanhar o ganho de peso dos animais.	1.14 e 3.10
RC15	O técnico de laboratório deseja inserir os resultados dos testes clínicos de PPT e Hematócrito.	2.1 e 2.2
RC16	O proprietário e o técnico de laboratório desejam acompanhar os relatórios de testes clínicos de PPT e Hematócrito.	1.15 e 2.3
RC17	O proprietário e o técnico de campo desejam acompanhar os relatórios de movimentação do estoque de vacinas.	1.16 e 3.11
RC18	O técnico de laboratório deseja encaminhar os animais doentes a enfermaria.	2.4
RC19	O proprietário e o técnico de laboratório desejam acompanhar o prontuário dos animais enfermos.	3.5
RC20	O proprietário deseja acompanhar o histórico de animais que vão ou estão na enfermaria.	1.17

Tabela 3 – Requisitos do cliente.

ID	Requisitos Funcionais do Sistema	Requisito(s) do Cliente relacionado(s).
RF01	O sistema deve permitir o cadastro de: fazendas, funcionários, animais, lote de animais e vacinas.	RC01, RC02, RC03, RC04 e RC05
RF02	O sistema deve permitir o registro de nascimentos, ocorrência de morte e mortes ao nascer (natimorto).	RC06, RC07 e RC08
RF03	O sistema deve permitir registrar o peso do animal e realizar o cálculo do ganho de peso.	RC09
RF04	O sistema deve permitir a geração de relatórios dos animais residentes na fazenda.	RC10
RF05	O sistema deve permitir a geração de relatórios de nascimentos dos animais na fazenda.	RC11
RF06	O sistema deve permitir a geração de relatórios de mortes de animais na fazenda.	RC12
RF07	O sistema deve permitir a geração de relatórios de doenças que atacam os animais.	RC13
RF08	O sistema deve permitir a geração de relatórios com as medidas de ganho de peso dos animais, sendo separados pelo lote a quem pertencem.	RC14
RF09	O sistema deve permitir o registro dos resultados dos testes clínicos de PPT e Hematócrito.	RC15
RF10	O sistema deve permitir a geração de relatórios dos testes clínicos realizados.	RC16
RF11	O sistema deve permitir a geração de relatórios da movimentação de vacinas no estoque.	RC17
RF12	O sistema deve permitir a criação do modulo de enfermaria onde serão registrados os animais que estão enfrentando problemas de saúde.	RC18
RF13	O sistema deve permitir o registro de animais doentes na enfermaria.	RC18
RF14	O sistema deve permitir a criação de um prontuário na enfermaria.	RC19
RF15	O sistema deve permitir o acompanhamento do prontuário dos animais que estão na enfermaria.	RC19
RF16	O sistema deve permitir a geração de relatórios dos animais que estão na enfermaria.	RC19 e RC20

Tabela 4 – Requisitos funcionais.

Definidos os objetivos e requisitos que o projeto deve atender, pudemos idealizar as dimensões e funcionalidades que o sistema deverá conter e também estipular as tecnologias que foram agregadas ao seu desenvolvimento.

## 2.3 Tecnologias utilizadas

As tecnologias são ferramentas que foram escolhidas para a realização do desenvolvimento deste projeto. Estas tecnologias foram estipuladas de acordo com as informações citadas nas Seções anteriores e são apresentadas a seguir. A Seção 2.3.1 apresenta a linguagem de programação *Personal Home Page* (PHP); a Seção 2.3.2 apresenta o sistema gerenciador de banco de dados MySQL; a Seção 2.3.3 apresenta a arquitetura *Model-View-Controller* (MVC); a Seção 2.3.4 apresenta a arquitetura *Representational State Transfer* (REST); a Seção 2.3.5 apresenta uma definição sobre *frameworks*; a Seção 2.3.6 apresenta o *framework* Laravel; a Seção 2.3.7 apresenta o framework Slim; a Seção 2.3.8 apresenta a biblioteca Valitron; a Seção 2.3.9 apresenta a linguagem de programação Javascript; a Seção 2.3.10 apresenta o *framework* Vue.js; a Seção 2.3.11 apresenta o *framework* Vuetify; a Seção 2.3.12 apresenta a biblioteca Axios; a Seção 2.3.13 apresenta a biblioteca ECharts; por fim, a Seção 2.3.14 apresenta o padrão de autenticação JSON *Web Token* (JWT).

### 2.3.1 PHP

O *Personal Home Page* (PHP) é uma linguagem de programação inicialmente criada como uma linguagem de *script* estruturada. Sua facilidade de aprendizado proporcionou um rápido aumento do número de programadores. Conseqüentemente, o surgimento de programas pouco eficientes e vulneráveis (MINETTO 2007).

Após um período de reformulações na estrutura da linguagem, a partir da versão 5 e com o surgimento de novas propriedades de orientação a objetos, a linguagem obteve a atenção de programadores familiarizados com paradigma, favorecendo o surgimento de aplicações melhores e robustas (MINETTO 2007).

Desenvolvido para ambiente *web*, o PHP possui licença livre, curva de aprendizado suave, simplicidade na integração com servidores *web* e facilidade de encontrar serviços de hospedagem (BENTO 2014).

Segundo (MELLO, 2017 apud REYES, 2009), o PHP possui diversas finalidades para o uso da linguagem, dentre elas estão: sistemas *e-commerce*, aplicações para o *Facebook*, para manuseio de conteúdo, etc. Outras vantagens são: facilidade de aprendizado, praticidade para a comunicação com MySQL e Oracle e produção de aplicações multiplataforma.

O PHP foi escolhido pelo fato de ser uma linguagem orientada a objetos, operar em diversas plataformas, pela sua praticidade na comunicação com o banco de dados, facilidade no aprendizado e de se encontrar hospedagens que suportam a linguagem.

### 2.3.2 MySQL

O *Structured Query Language* (SQL) é uma linguagem utilizada para gerar estruturas de bases de dados relacionais, realizar manutenções e gerenciar seu conteúdo (MELLO, 2017 apud SHELDON, 2009). Essa linguagem estabelece um padrão universal para todos os gerenciadores de banco de dados relacionais, que, mesmo podendo adicionar funcionalidades exógenas ao SQL, buscam sempre implementar os comandos definidos pela norma.

MySQL é um sistema gerenciador de banco de dados (SGBD) que utiliza SQL para manipulação de bancos de dados relacionais e é eficaz para atender aplicações de pequeno, médio e grande porte (Milani 2007). Além disso, o MySQL é um sistema multiplataforma, possui uma versão de licença de uso livre e é suportado por diferentes linguagens de programação, como Java, PHP, Python, etc. Por este motivo, temos a liberdade de utilizá-lo em diversos projetos e contribuir para o seu desenvolvimento (BENTO 2014).

Dentre suas vantagens o Mysql se destaca pela sua confiabilidade, velocidade, escalabilidade, suporte a grande carga de acessos simultâneos, alto desempenho na execução de comandos, controle de transações e capacidade para manipulação de tabelas com grande quantidade de registros (NIEDERAUER 2005).

Apesar de existir um *fork* de licença paga, fizemos a escolha do Mysql versão comunidade por ser um sistema gerenciador de banco de dados de licença livre, ser suportado por diversas linguagens e plataformas, manipular bancos de dados relacionais, confiabilidade e desempenho.

### 2.3.3 MVC

Segundo (Silva 2012), o *Model-View-Controller*(MVC) é um padrão que sugere o desenvolvimento de uma arquitetura de *software* separada por camadas, tendo por resultado um código organizado, garantindo manutenções seguras, reaproveitamento de código e eficiência no desenvolvimento. Sua arquitetura é dividida nas seguintes camadas:

A camada *view* ou camada de apresentação é responsável pela interação com o usuário, receptando suas ações e transmitindo para a camada *controller*.

A camada *controller* ou camada de negócios é responsável pela intermediação entre as camadas, recebendo as requisições da *view*, realizando as ações comportamentais e repassando as informações para a *model* que, por sua vez, pode gerar uma resposta para ser entregue para a *view*.

Camada *model* ou camada modelo é responsável pela manipulação e armazenamento das informações no banco de dados através da persistência. O padrão de arquitetura MVC foi adotado no projeto para obter uma clara organização estrutural, separação de responsabilidades por camadas, diminuindo acoplamento e aumentando o reaproveitamento.

### 2.3.4 REST

Representational State Transfer (REST) é uma arquitetura para projetos *web service*, que utiliza o protocolo *Hipertext Transfer Protocol / Hipertext Transfer Protocol Security* (HTTP/HTTPS) como meio de comunicação e o JSON como formato de dados (MONTANHEIRO; CARVALHO; RODRIGUES, 2017 apud FIELDING; TAYLOR; RIBEIRO; FRANCISCO, 2000, 2016).

*Web Service* é uma tecnologia para criar aplicações integradas e realizar a comunicação entre diferentes sistemas. Através dela é possível que novas aplicações possam se comunicar e trocar informações com sistemas já existentes independentemente da plataforma de desenvolvimento (ALEMIDA; BANDEIRA, apud RICARDO, 2015).

O REST permite que sistemas computacionais que operam em plataformas diferentes solicitem, acessem e manipulem representações textuais de recursos. Um recurso é qualquer informação que pode ser acessada, seja ela, registro do banco de dados, um objeto da aplicação, algoritmo, etc.

Para o REST, tudo é um recurso e cada um possui identificação própria, através do *Universal Resource Identifier* (URI) e também as operações nos métodos do protocolo HTTP: *GET*, *PUT*, *POST*, *DELETE*, *HEADER* e *OPTIONS*, retornando um recurso no formato *Javascript Object Notation* (JSON) ao chamador (CHEN et al. 2017).

Na Figura 1, temos a ilustração de um diagrama representando a tecnologia REST. Nela está representado o cliente que emite requisições ao servidor através dos métodos do protocolo de comunicação HTTP. Do outro lado, o servidor ouvindo as requisições e retornando arquivos no formato JSON.

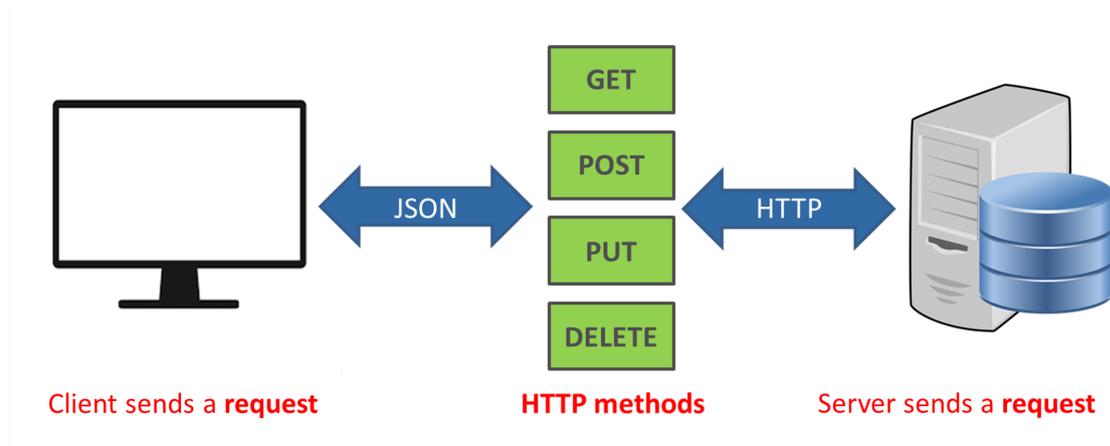


Figura 1 – Diagrama de arquitetura REST.

A arquitetura REST foi implementada visando desenvolver um projeto independente, tendo a capacidade de integração com diferentes tecnologias de interface de usuário para a *web* e *mobile*.

### 2.3.5 Frameworks

Um *framework* é uma coleção de componentes que possui uma arquitetura e estrutura interna básica para o desenvolvimento de sistemas. Em outras palavras, é uma aplicação semipronta, onde o programador deve entendê-la e personalizá-la para o correto funcionamento do sistema (SUPPI, 2012 apud JOBSTRAIBIZER, 2009).

Adotar o uso de um *framework* no desenvolvimento pode dar a ideia de perda de flexibilidade na programação, uma vez que é necessário que se utilize as mesmas regras, convenções, classes e bibliotecas que a ferramenta oferece. Por outro lado, obtemos uma padronização, reusabilidade e facilidade na manutenção, necessitando apenas de se inteirar sobre a forma que o *framework* trabalha e está implementado na arquitetura (MINETTO 2007).

Desta forma, a utilização de *frameworks* no processo de desenvolvimento de um *software* proporciona o aumento na produtividade, visto que os *frameworks* possuem uma série de recursos que, sem eles, os programadores ficariam a cargo de construí-los (BELÉM et al., 2014 apud MINETTO, 2007).

Foi adotada a utilização de *frameworks* neste projeto visando as vantagens de se obter uma padronização, reusabilidade, facilidade e agilidade no processo de desenvolvimento, devido a gama de propriedades que estes disponibilizam. A seguir, serão apresentados os *frameworks* utilizados no desenvolvimento do projeto.

### 2.3.6 Laravel

O Laravel é um *framework* PHP de desenvolvimento de *software* para a *web* e tem como principal objetivo construir aplicações seguras, eficientes com código simples e limpo, utilizando o padrão MVC (SILVA, 2017 apud ADRIEL, 2015).

O Laravel é gratuito, possui licença livre, é fácil de entender e possui alguns recursos como: autenticação, roteamento, gerenciamento de sessões, armazenamento em cache, ferramentas para migração de banco de dados e suporte a testes de unidade (CHEN et al. 2017). Para compreendermos o seu funcionamento, precisamos definir o conceito de ORM, que, segundo (CHEN et al. 2017):

“Object Relational Mapping (ORM) é uma técnica de programação para mapear dados entre o bancos de dados e objetos na memória ao usar linguagens de programação orientadas a objetos”.

Tendo posto, o Laravel realiza a comunicação com o banco de dados por meio da biblioteca Eloquent ORM. Esta ferramenta possui recursos para o gerenciamento da comunicação com o banco de dados, facilitando o desenvolvimento do *software* (MELLO, 2017 apud ADRIEL, 2015).

O Eloquent ORM é implementado da seguinte forma: no projeto, para cada tabela presente no banco de dados, o Eloquent possui uma classe modelo utilizada para interagir com o banco de dados, bem como realizar as operações de persistência (CHEN et al. 2017).

### 2.3.7 Slim

O Slim é um *microframework* para a linguagem PHP utilizado para desenvolver *web services* REST que possibilitam publicar, consumir ou redirecionar dados. É um *framework* rápido, simples e com pouco código. Sua principal responsabilidade se concentra em receber solicitações via protocolo HTTP, invocar uma rotina e retornar uma resposta (ALEMIDA e BANDEIRA).

### 2.3.8 Valitron

O Valitron é uma biblioteca de validação de dados de objetos. Diferente de outras bibliotecas que dependem de grandes componentes, o Valitron é simples, não possui dependências e utiliza métodos descomplicados de validação com foco em uma sintaxe resumida (LUCAS 2014).

A escolha de empregar os *frameworks* Laravel, Slim e Valitron foi pelo fato dessas tecnologias serem gratuitas, possuírem poderosos recursos para o desenvolvimento tanto para aplicações simples quanto robustas e são desenvolvidos para a linguagem PHP.

### 2.3.9 Javascript

O Javascript é uma linguagem de programação orientada a objetos que implementa elementos complexos em páginas *web*, permitindo manipulá-los como objetos distintos. A atualização dinâmica de dados só é possível em páginas HTML por meio do Javascript, sendo o responsável pela manipulação do HTML (PORTA, 2018 apud TABLELESS, 2018).

### 2.3.10 Vue.js

O Vue.js é um *framework* JavaScript de alto desempenho utilizado no desenvolvimento de interfaces para o usuário. Integrado ao projeto, seus componentes podem ser inseridos em qualquer parte no *front end*, simplesmente injetando uma *tag script*. (SHAH et al. 2019). Estruturado para ser adaptável, o Vue.js foi criado para a construção da camada visual, podendo ser facilmente adicionada junto a outras bibliotecas e projetos existentes (PORTA 2018).

O Vue.js é reativo. Sua biblioteca é capaz de observar e refletir qualquer mudança no estado do Document-Object-Model (DOM) do HTML. Em um processo de alteração do estado de um objeto, o Vue compara o DOM virtualizado (pois ele atua inicialmente nessa

versão temporária do DOM) e reflete as alterações no DOM real, mantendo a consistência do objeto (FERREIRA; ZUCHI; ALMEIDA, 2018, apud LACERDA; INCAU, 2017, 2017).

O Vue permite que qualquer tipo de interface seja moldado em uma estrutura de componentes, podendo ser reutilizados em diversas partes na construção do sistema, reduzindo o processo repetitivo no desenvolvimento (SONG; ZHANG; XIE, 2019 apud BAYLARI; MONTAZER; WEBER; LINGE N. PARSONS, 2009, 1969, 2006).

### 2.3.11 Vuetify

O Vuetify é um *framework* voltado para o desenvolvimento de interfaces visuais. É suportado por todos os navegadores modernos e oferece componentes e comportamentos reutilizáveis compatíveis com o Vue.js. Os componentes são adaptativos, se adequam a tela de qualquer dispositivo de navegação e também à manipulação de conteúdo (PORTA 2018, SHAH et al. 2019).

O Vue.js e o Vuetify foram aplicados no desenvolvimento do projeto devido a sua flexibilidade, desempenho, adaptabilidade, compatibilidade com navegadores modernos, reusabilidade e agilidade no processo de desenvolvimento.

### 2.3.12 Axios

Axios é uma biblioteca utilizada para realizar requisições a servidores REST por meio do protocolo HTTP. A biblioteca atua no lado do cliente, utilizando os métodos HTTP *GET* para obter uma informação, *POST* para inserir, *PUT* para atualizar e *DELETE* para remover. A requisição é destinada ao servidor através de um *link* utilizando os métodos HTTP correspondentes, obtendo de retorno respostas no formato JSON (SILVA 2019).

### 2.3.13 Echarts

O Echarts é uma ferramenta para construção de componentes de visualização interativa na camada visual. Esta ferramenta fornece componentes responsivos, modeláveis e configuráveis utilizando o formato JSON para declaração de componentes, estilos, dados de visualização e interações. Sua biblioteca possui licença de código aberto, permite integrações com outras tecnologias, suporte ao *framework* Vue.js e também a múltiplas plataformas (LI et al. 2018).

Dentre seus componentes visuais, o Echarts disponibiliza uma variedade de gráficos, assim como gráficos de barras, pizza e mapa de árvore. Sua estrutura principal implementa a biblioteca ZRender que é responsável pela exibição e gerenciamento de elementos para interface gráfica (LI et al. 2018). Os componentes visuais são compostos por 3 tipos

de funções: codificação visual, guia e interação. A codificação visual corresponde aos parâmetros de visualização orientados por dados assim como a altura em gráficos de barras e área em gráficos de pizza. O guia fornece configurações auxiliares para leitura e análise dos dados apresentados na visualização podendo utilizar etiquetas, grifos, linhas ou informações adicionais (LI et al. 2018).

A interação fornece funcionalidades para operações interativas. Por exemplo, em uma visualização de gráfico, permite ao usuário selecionar dados em legenda e apresentar informações detalhadas de indicadores (LI et al. 2018).

A biblioteca Echarts foi adicionada ao projeto para atendimento de requisito de apresentação de dados em relatórios, optando pela demonstração de determinados dados através de gráficos para que o usuário obtenha a máxima absorção e compreensão das informações.

### 2.3.14 JWT

O JSON *Web Token* (JWT) é um padrão de autenticação de usuário e transporte seguro de dados entre o cliente e o servidor por meio de um cabeçalho HTTP (MONTANHEIRO; CARVALHO; RODRIGUES, 2017 apud JONES M.; SCHNEIDER, 2015, 2016).

Diversas bibliotecas implementam o padrão JWT para a utilização em variadas linguagens de programação. A utilização do padrão para a autenticação de usuários é o cenário mais comum, mas também é utilizado como forma segura de transmissão de dados. Cada solicitação deve conter o *token* para a verificar a permissão de acesso e validade do conteúdo com base na assinatura criptografada (MONTANHEIRO; CARVALHO; RODRIGUES, 2017 apud JONES; JWT, 2011, 2017).

Em aplicações *web services*, o *token* é gerado pelo servidor através do JWT, nele estão contidas informações que podem ser criptografadas através do algoritmo *Hash-based Message Authentication Code* (HMAC) ou um par de chaves públicas e privadas. (MONTANHEIRO; CARVALHO; RODRIGUES, 2017 apud JONES M.; SCHNEIDER, 2015, 2016).

O JWT foi adotado no projeto para prover a segurança da aplicação na realização dos procedimentos de entrada e saída de usuário, registro, consulta, alteração e remoção de informações promovendo segurança as requisições, garantindo autenticidade e permissão para acesso.

Este capítulo apresentou o planejamento do projeto, os requisitos elucidados e as tecnologias que posteriormente foram utilizadas no desenvolvimento do *software*. A seguir, o Capítulo 05 apresenta a modelagem do *software* com base nas regras e planejamento.

## 3 Modelagem do Sistema

Este capítulo apresenta a modelagem do sistema utilizando os diagramas de UML, construídos com base nos requisitos obtidos em entrevistas com os *stakeholders* que contribuíram para a compreensão do contexto e no desenvolvimento do *software*. A seguir, a Seção 3.1 apresenta os diagramas de casos de uso; a Seção 3.2 apresenta a descrição de cada caso; a Seção 3.3 apresenta o diagrama de classes; a Seção 3.4 apresenta a modelagem de banco de dados; por fim, a Seção 3.5 apresenta a descrição das tabelas do banco de dados.

### 3.1 Casos de uso

O diagrama de casos de uso, Figura 2, apresenta o diagrama de casos de uso, uma visão geral das funcionalidades do *software*. A Figura 3 ilustra o significado de cada objeto da Figura 2.

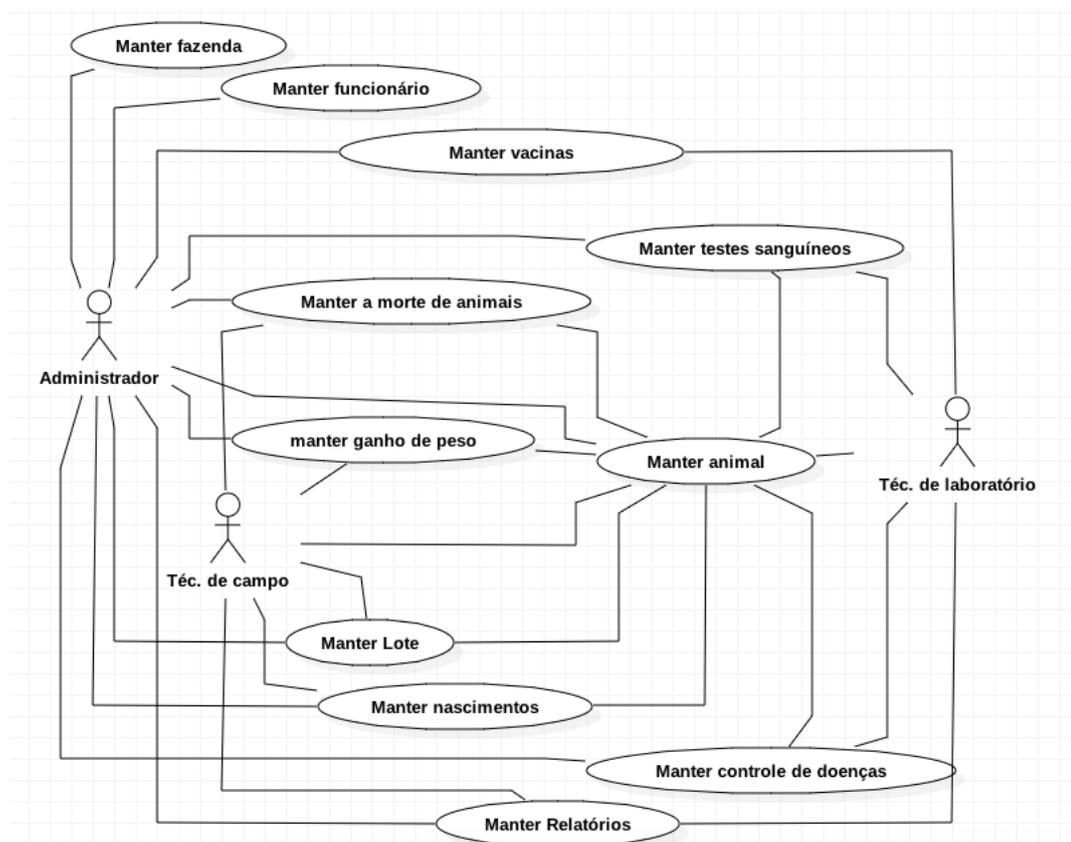


Figura 2 – Diagrama de casos de uso.



Figura 3 – Significado dos objetos.

## 3.2 Descrição dos casos de uso

Cada caso de uso representa uma função, que por sua vez, comunicam entre si e/ou com seus atores. Os atores representam uma interação humana ou de outro *software*. Os itens a seguir descrevem as atividades que compõem cada caso.

- Caso de uso Manter Fazenda: cadastrar, alterar, listar e excluir os dados da fazenda que se destinará o controle.
- Caso de uso Manter Lotes: cadastrar, alterar, listar e excluir os lotes de animais das fazendas.
- Caso de uso Manter Funcionários: Cadastrar, alterar, listar e excluir os dados dos funcionários colaboradores, permitindo-lhes o acesso a plataforma.
- Caso de uso Manter Medicamentos/Vacinas: Cadastrar, alterar, listar e excluir as vacinas e medicamentos que serão utilizados no controle, podendo controlar o estoque e as vacinações periódicas e/ou medicações para o combate de doenças.
- Caso de uso Manter Testes Sanguíneos: cadastrar, alterar, listar e excluir o resultado dos testes de hematócrito e PPT obtidos a partir do exame no sangue dos animais. Estas informações demonstrarão para o usuário, de acordo com parâmetros da área, qual o estado de saúde do animal.
- Caso de uso Manter Morte de Animais: registro de ocorrência de mortes de animais.
- Caso de uso Manter Controle de Doenças: Cadastrar, alterar, listar e excluir as doenças que podem aparecer no rebanho, realizando o controle de adoecimentos dos animais, referenciando as doenças cadastradas.
- Caso de uso Manter Animais: Cadastrar, alterar, listar e excluir as informações dos animais.
- Caso de uso Manter Nascimento: registro nascimentos ocorridos nas fazendas.
- Caso de uso Manter Ganhos de Peso: Cadastrar as pesagens realizadas em cada animal.

- Caso de uso Manter Controle de Acessos: Realizar autenticação, cadastrar, alterar, listar e excluir usuários, grupos e permissões de módulos.

### 3.3 Diagrama de classes

A Figura 4 representa o diagrama de classe baseado no diagrama de casos de uso, ilustra o mapeamento das classes que irão compor a estrutura do projeto. Nesse nível de visão, são apresentados o nome das entidades, os atributos e o tipo de relacionamento entre elas. A Figura 5 apresenta a simbologia de relacionamentos entre as classes adotadas.

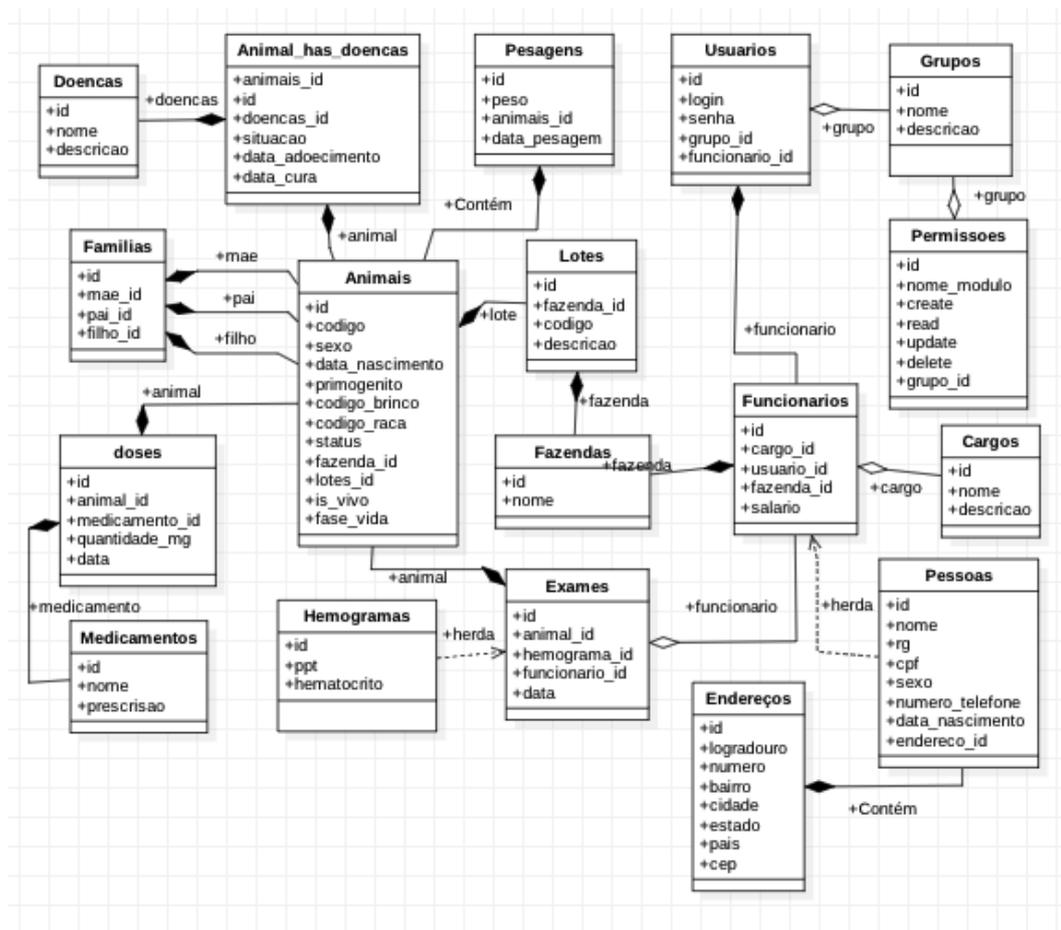


Figura 4 – Diagrama de classes.

### 3.4 Diagrama de banco de dados

A Figura 6 apresenta o diagrama de banco de dados e é a representação das entidades da camada de modelo dos dados. Nesse nível de visão, são apresentadas as tabelas, atributos e relacionamentos, e a cardinalidade entre eles.

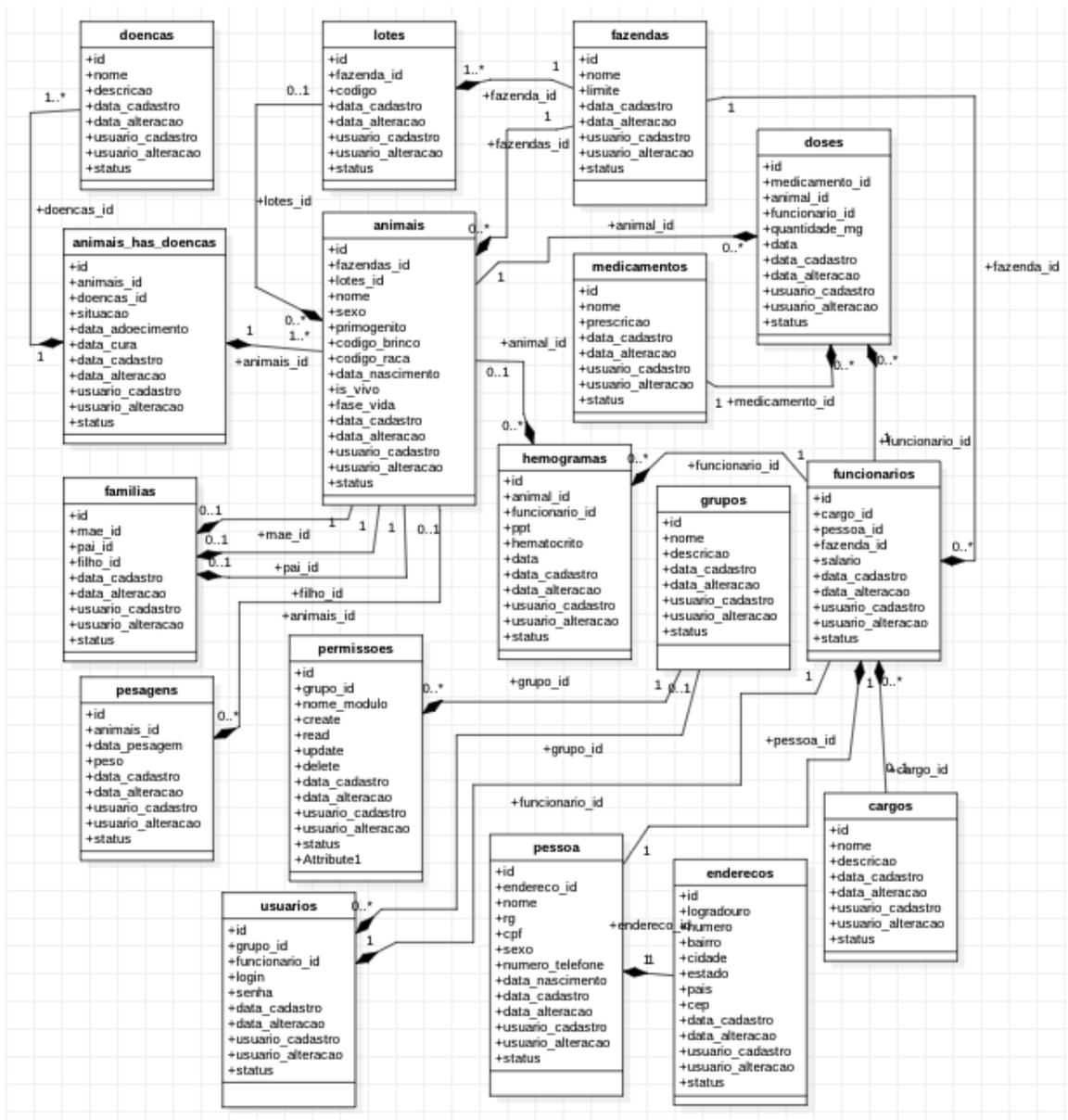


Figura 5 – Diagrama de banco de dados.

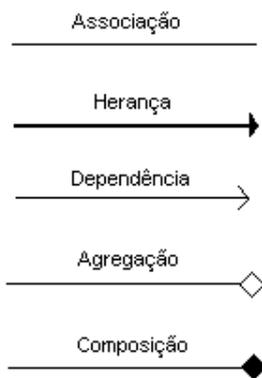


Figura 6 – Tipos de relacionamentos.

## 3.5 Descrição das tabelas do banco de dados

Adotamos o seguinte padrão para a criação das tabelas do banco de dados relativas às classes apresentadas na Figura 4.

1. O nome da tabela é igual ao nome da classe;
2. O nome dos atributos é igual ao nome das colunas;
3. O nome das tabelas e atributos são escritos com letra minúscula e separadas por *underline*, se for o caso;
4. Todas as chaves primárias são nomeadas 'id';
5. Todas as chaves estrangeiras são nomeadas com o nome da tabela referenciada seguido de '\_id'.

Essa padronização ajuda o *framework* ORM (Seção 3.6 do Capítulo 02) a mapear as classes (objetos) para persistência automatizada no banco de dados. A Tabela 05 apresenta uma breve descrição da finalidade das tabelas presentes no diagrama acima.

Este capítulo apresentou as modelagens desenvolvidas baseadas nos requisitos obtidos de acordo com a necessidade de informatização que este *software* atendeu, tendo sido elas o ponto de partida para o desenvolvimento do *software* que será apresentado no capítulo seguinte, aplicando os conceitos e tecnologias apresentadas no Capítulo 02.

Tabela	Descrição
animais	Tabela responsável por armazenar os dados dos animais.
doencas	Tabela responsável por armazenar as informações da doença que ataca a saúde dos animais.
animais_has_doencas	Tabela responsável por armazenar a referência de registros de animais (ID) e doenças para se registrar um adoecimento do animal, tornando-se uma tabela que corresponde a uma entidade fraca, agrupando duas entidades fortes.
fazenda	Tabela responsável por armazenar as informações de fazenda, podendo armazenar uma ou muitas.
medicamentos	Tabela responsável por armazenar os dados dos medicamentos.
dose	Tabela responsável por armazenar os dados de referência do animal (ID), do medicamento ou vacina e a quantidade que foi aplicada ao animal.
pessoa	Tabela responsável por armazenar as informações pessoais dos colaboradores.
funcionario	Tabela responsável por armazenar a referência de pessoa (ID), fazenda e cargo.
pessoa_endereco	Tabela responsável por armazenar os dados de endereços de pessoas.
usuario	Tabela responsável por armazenar as informações para a autenticação do usuário a plataforma.
cargo	Tabela responsável por armazenar os dados do cargo.
grupo	Tabela responsável por armazenar as informações do grupo, referenciando os registros de usuário ingressos no grupo.
permissao	Tabela responsável por armazenar os dados de permissões de acesso de usuários.

Tabela 5 – Descrição das tabelas do banco de dados.

## 4 Desenvolvimento

Este capítulo apresenta a metodologia utilizada no desenvolvimento do *software*. A seguir, a Seção 4.1 apresenta a arquitetura geral do *software*; a Seção 4.2 apresenta a arquitetura *back-end*; a Seção 4.3 apresenta a arquitetura *front-end*; por fim, a Seção 4.4 apresenta a integração entre *front-end* e *back-end*.

### 4.1 Arquitetura

A arquitetura deste projeto se divide em *front-end* e *back-end*. O *front-end* é responsável pela interação com o usuário e o *back-end* pelas ações comportamentais do sistema, processamento e armazenamento de dados.

A Figura 7 apresenta a distribuição das camadas do padrão MVC entre o *front-end* e o *back-end*. Desta forma, o *front-end* assume a camada *View* onde estão toda a estrutura necessária para a exibição de telas interativas e captura de ações do usuário. O *back-end* assume as camadas *Model* e *Controller* abrangendo toda a estrutura para o recebimento de requisições da camada *View*, processamento e manipulação dos dados do sistema.

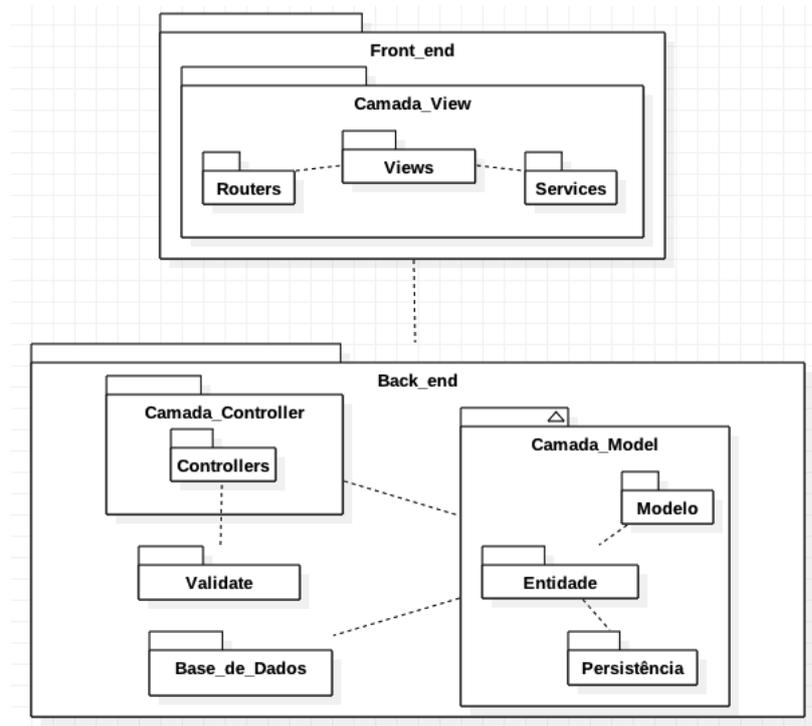


Figura 7 – Diagrama de arquitetura.

## 4.2 Arquitetura *back-end*

O *back-end* é a estrutura servidora do sistema, contendo em sua base algoritmos para funcionalidades, tratamento de dados, regras de negócio e atendimento de requisições.

Na construção da estrutura *back-end*, foi utilizada a linguagem PHP em conjunto com os *frameworks* Slim e Laravel, além das bibliotecas Eloquent e Valitron, que desempenharam um importante papel ao agilizar o desenvolvimento da validação e persistência dos dados.

Como dito anteriormente, a camada *Model* é composta pelos pacotes persistência, entidade e modelo. A Figura 8 exibe um excerto da classe Cargo, sendo representada em cada um dos pacotes da camada *Model*.

```

namespace CalfManager\Model;

use CalfManager\Model\Repository\CargoDAO;
use CalfManager\Utils\Config;
use Exception;

class Cargo extends Modelo
{
    private $nome;
    private $descricao;
}

class CargoDAO implements IDAO
{
    public function create($obj): ?int
    {...17 lines }

    public function update($obj): bool
    {...18 lines }

    public function retrieveAll(int $page): array
    {...15 lines }

    public function retrieveById(int $id): array
    {...9 lines }

    public function retrieveByNome(string $nome, int $page)
    {...15 lines }

    public function delete(int $id): bool
    {...9 lines }
}

class CargoEntity extends CalfEntity
{
    protected $stable = "cargos";
    protected $fillable = [
        'id',
        'nome',
        'descricao',
        'data_alteracao',
        'data_cadastro',
        'usuario_cadastro',
        'usuario_alteracao',
        'status'
    ];

    public function funcionarios()
    {
        return $this->hasMany("\CalfManager\Model\Repository\Entity\FuncionarioEntity");
    }
}

```

Figura 8 – Apresentação das classes Cargo, CargoDao e CargoEntity.

Observando a Figura 7, vemos que as classes do pacote que chamamos de modelo são responsáveis pela simples representação do objeto tratado, na medida do possível, seguindo a mesma estrutura das tabelas do banco de dados (nome da classe igual ao nome da tabela; nome e domínio dos atributos iguais as colunas da tabela, etc).

O pacote entidade é um requisito da biblioteca Eloquent ORM que, em sua estrutura

interna, utiliza as classes desse pacote para realizar o mapeamento entre as classes do pacote modelo e as classes do pacote de persistência, funcionando como uma interface, compatibilizando os objetos com a biblioteca Eloquent.

As classes do pacote de persistência, nomeadas com o sufixo *Data Access Object* (DAO), são responsáveis por realizarem as operações *create*, *retrieve*, *update*, *delete* (CRUD), utilizando os recursos do Eloquent na construção das funcionalidades para estas operações.

A base de dados foi desenvolvida utilizando o sistema gerenciador de banco de dados Mysql, em conjunto com o SQL, possibilitando estruturar a base conforme a modelagem de bando de dados apresentada na Seção 1.3 do Capítulo 3.

As classes do pacote *controller* foram implementadas com os recursos do *framework* Slim e são responsáveis por atender requisições e emitir respostas através do protocolo HTTP. Além disso, estas classes possuem tratamento de regras de negócio e funcionalidades comportamentais de acordo com o módulo correspondente.

Por questões de integridade e segurança dos dados, as classes do pacote *Validate* atuam na validação de dados enviados ao servidor por meio das funcionalidades da biblioteca *Valitron*.

### 4.3 Arquitetura *front-end*

O *front-end* é a estrutura cliente do sistema, responsável pela interação com o usuário. Nele, foram desenvolvidas a interface com o usuário através de telas com funcionalidades interativas.

Conforme a Figura 8, o *front-end* envolve a camada *view*. Nela, estão agregados os pacotes *routers*, *views* e *services*. No pacote de *Views* englobam-se os arquivos de telas. O pacote *routers* possui o arquivo onde são mapeadas as rotas de arquivos de telas para serem acessadas através de um menu. Por fim, o pacote *services* aglomera todos os arquivos contendo funcionalidades referente aos serviços de comunicação com o *back-end*.

O Vue.js foi a tecnologia utilizada no desenvolvimento da estrutura principal do *front-end*, sendo desenvolvidas as estruturas de telas, rotas de navegação e serviços. O Vuetify foi a tecnologia que forneceu todos os componentes necessários para a construção visual das telas.

Como visto no parágrafo anterior, as bibliotecas Vue.js e Vuetify foram responsáveis pela construção das telas, fornecendo para as telas de cadastros os componentes de formulários, de acordo com a sua proposta. As telas de relatórios, além destas tecnologias, em algumas funcionalidades, para se obter uma apresentação gráfica, foram adicionados recursos de renderização de gráficos de dados através da biblioteca Echarts.

## 4.4 Integração *back-end* e *front-end*

Os projetos *front-end* e *back-end* como visto na Figura 8 são independentes. A forma de integração entre ambos é por meio da implementação da arquitetura REST, permitindo que o *front-end* emita solicitações através da biblioteca axios ao *back-end*, sendo destinada esta solicitação às rotas predefinidas para os módulos, trafegando nestas solicitações, informações em arquivos JSON.

Como o projeto foi desenvolvido para a *web*, o servidor *back-end* poderá ser disponibilizado na rede para ser acessado através de algum endereço de acordo com a hospedagem ou então, ser disponibilizado em servidores em rede local. Pensando em garantir a segurança das informações, adotamos o JWT para garantir a segurança no acesso.

A segurança adotada pelo JWT atua da seguinte forma: ao realizar o *login*, o servidor emite como resposta um *token* para o *front-end*. Este *token* é armazenado para que toda requisição feita informe o *token* que será autenticado.

Se o *token* estiver válido e consistente, o servidor concede a permissão e realiza o procedimento solicitado. Se não, cancela a operação. Assim, garantimos que somente usuários e aplicações *front-ends* válidos acessem os módulos do nosso sistema.

Este capítulo apresentou os conceitos e práticas aplicados no desenvolvimento do *software*, explanando a estrutura dos projetos *back-end*, *front-end* e sua integração. A seguir, no Capítulo 05, será apresentada a solução.

# 5 Apresentação da solução

Neste Capítulo, apresentamos a solução desenvolvida, demonstrando as funcionalidades de cada módulo por meio de sua interface gráfica. A seguir, a Seção 5.1 apresenta um conceito sobre os módulos; a Seção 5.2 apresenta o módulo Fazendas; a Seção 5.3 apresenta o módulo Lotes; a seção 5.4 apresenta o módulo Doenças; a seção 5.5 apresenta o módulo Animais; a Seção 5.6 apresenta o módulo enfermarias; a Seção 5.7 apresenta o módulo hemogramas; a Seção 5.8 apresenta o módulo Pesagens; a Seção 5.9 apresenta o módulo Medicamentos/Vacinas; por fim, a Seção 5.10 apresenta o módulo Funcionários.

## 5.1 Módulos

Nesta solução, cada módulo corresponde a um caso de uso e possui as funcionalidades de CRUD e regras de negócio desenvolvidas de acordo com a necessidade de cada um deles, de acordo com o diagrama de casos de uso apresentado no Capítulo 03, Seção 3.1.

Para cada um dos módulos, o sistema possui telas de cadastros e listagem com as funcionalidades específicas para a realização das operações. A seguir, serão apresentados os módulos: Fazendas, Lotes, Doenças, Animais, Hemogramas, Enfermarias, Pesagens, Medicamentos/Vacinas e Funcionários.

### 5.1.1 Módulo fazendas

O módulo fazendas permite ao usuário registrar e gerenciar a(s) fazenda(s) que fazem parte do sistema de controle de bezerros, permitindo controlar uma ou várias fazendas, que posteriormente poderão ser referenciadas aos registros de lotes e animais.

Este módulo corresponde ao caso de uso manter fazenda e expõe suas funcionalidades através das telas de cadastro e listagem de fazendas. A tela de cadastro apresentada na Figura 9 é responsável por registrar o nome da fazenda.

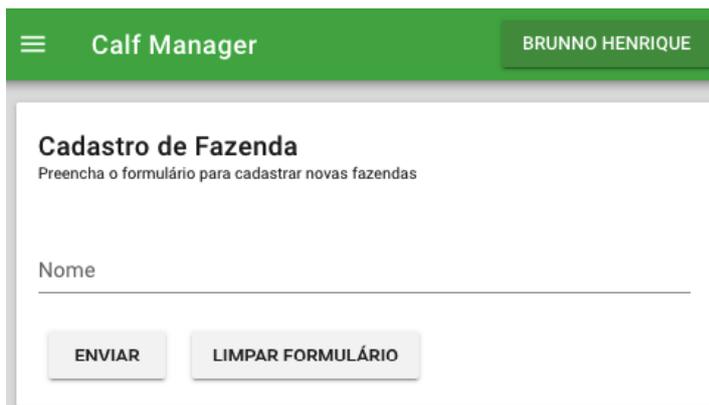
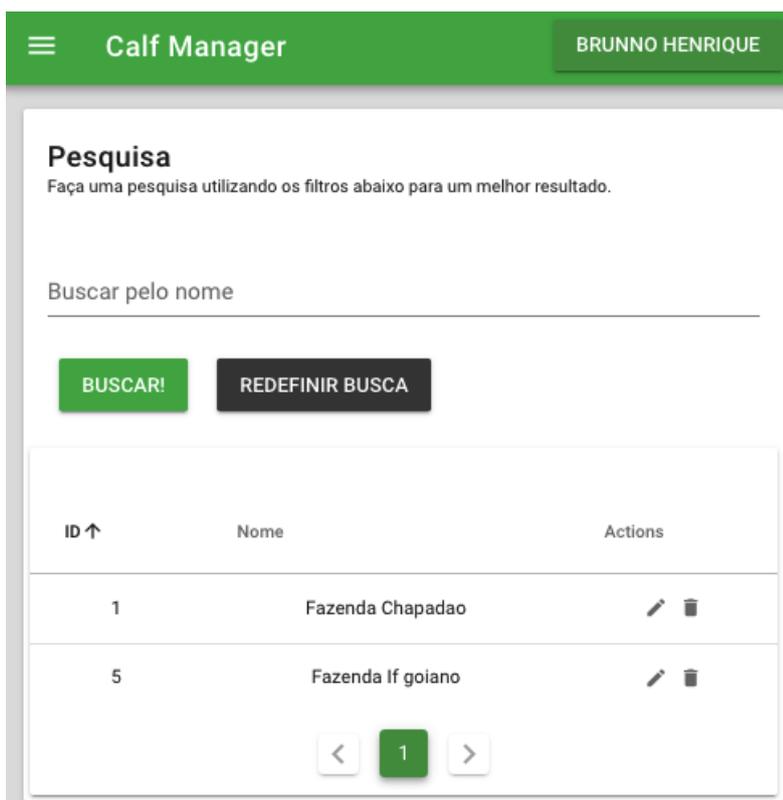


Figura 9 – Tela de cadastro de fazenda.

A tela de listagem apresentada na Figura 10 é responsável por apresentar o relatório de fazendas registradas por meio de lista. Nela estão incluídas as funcionalidades de filtragem, edição e exclusão dos registros.



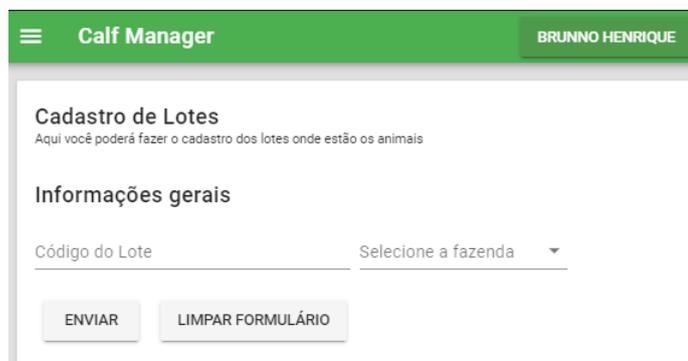
ID ↑	Nome	Actions
1	Fazenda Chapadao	 
5	Fazenda If goiano	 

Figura 10 – Tela de listagem de fazendas.

### 5.1.2 Módulo lotes

O módulo de lotes permite ao usuário registrar e gerenciar os lotes de animais da(s) fazenda(s), tendo seus registros de lote referenciados aos cadastros de fazenda e animais que compoem os lotes.

Este módulo corresponde ao caso de uso manter lote e expõe suas funcionalidades ao usuário através das telas de cadastro de listagem de lotes. A tela de cadastro apresentada na Figura 11 é responsável por registrar o código do lote e selecionar a fazenda que o lote pertence.



**Calf Manager** BRUNNO HENRIQUE

### Cadastro de Lotes

Aqui você poderá fazer o cadastro dos lotes onde estão os animais

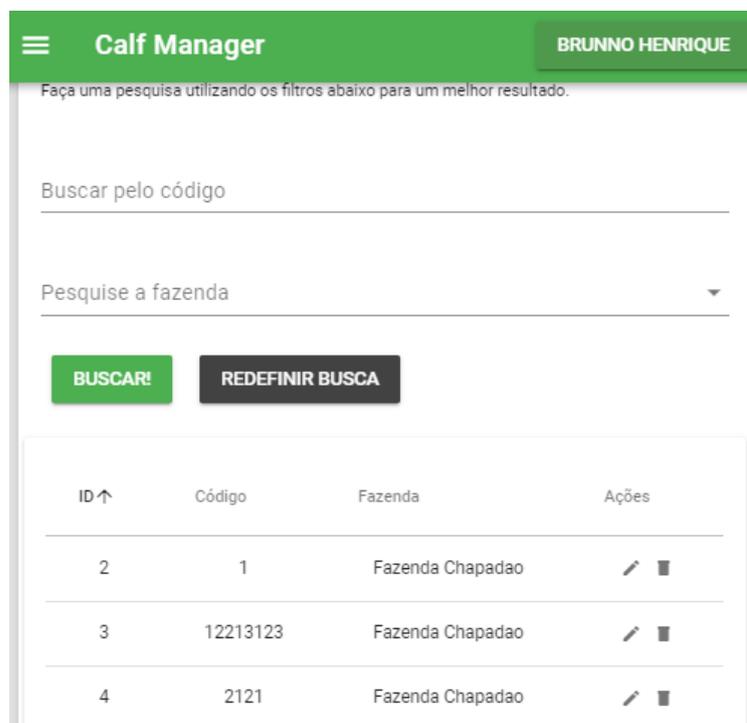
#### Informações gerais

Código do Lote  Selecione a fazenda

**ENVIAR** **LIMPAR FORMULÁRIO**

Figura 11 – Tela de cadastro de lotes.

A tela de listagem apresentada na Figura 12 é responsável por apresentar o relatório de lotes permitindo serem visualizados, por meio de lista, todos os lotes registrados ou filtrá-los por código ou pela fazenda referenciada. Nesta lista, estão incluídas as funcionalidades para edição e exclusão dos registros.



**Calf Manager** BRUNNO HENRIQUE

Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.

Buscar pelo código

Pesquise a fazenda

**BUSCAR!** **REDEFINIR BUSCA**

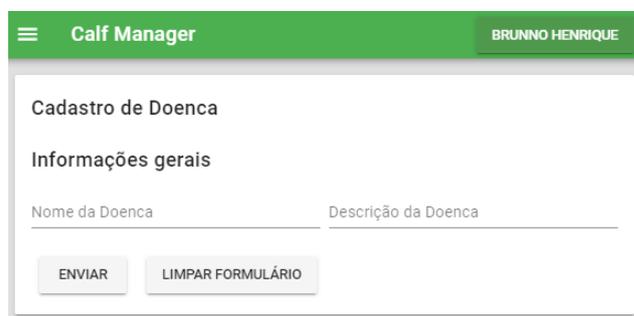
ID ↑	Código	Fazenda	Ações
2	1	Fazenda Chapadao	 
3	12213123	Fazenda Chapadao	 
4	2121	Fazenda Chapadao	 

Figura 12 – Tela de listagem de lotes.

### 5.1.3 Módulo doenças

O módulo doenças permite ao usuário registrar e gerenciar a(s) doença(s) que podem aparecer no rebanho durante o manejo. Suas informações são úteis para que sejam gerenciadas pelos técnicos de acordo com seu surgimento nos animais, para que possam ser controladas na fazenda.

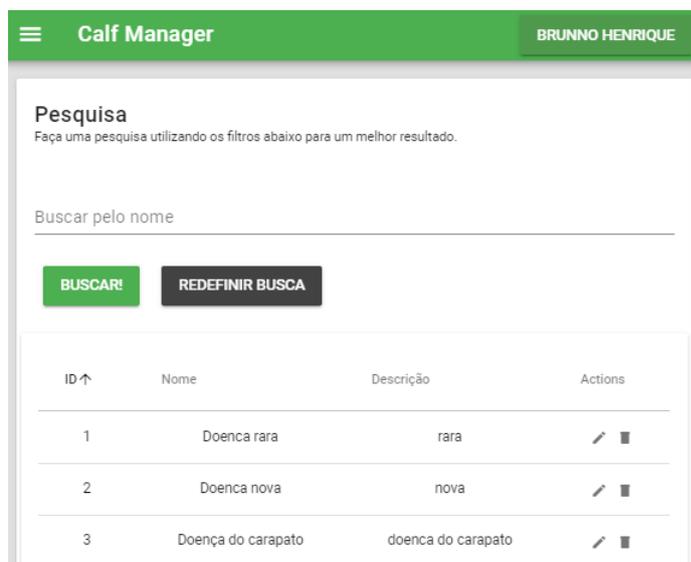
Este módulo corresponde ao caso de uso manter controle de doenças e expõe suas funcionalidades ao usuário através das telas de cadastro e listagem de doenças. A tela de cadastro apresentada na Figura 13 é responsável por registrar o nome e a descrição da doença.



A interface de cadastro de doenças no sistema Calf Manager. O formulário é dividido em seções: 'Cadastro de Doença' e 'Informações gerais'. Há dois campos de texto: 'Nome da Doença' e 'Descrição da Doença'. Abaixo dos campos, há dois botões: 'ENVIAR' e 'LIMPAR FORMULÁRIO'.

Figura 13 – Tela de cadastro de doenças.

A tela de listagem apresentada na Figura 14 é responsável por apresentar o relatório de doenças permitindo serem visualizados, por meio de lista, todas as doenças registradas ou filtrá-las pelo nome. Nesta lista, estão incluídas funcionalidades para edição e exclusão dos registros.



A interface de listagem de doenças no sistema Calf Manager. A tela mostra uma seção de pesquisa com o título 'Pesquisa' e o subtítulo 'Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.' Abaixo disso, há um campo de busca com o texto 'Buscar pelo nome' e dois botões: 'BUSCAR!' e 'REDEFINIR BUSCA'. Abaixo da seção de busca, há uma tabela com as seguintes colunas: ID ↑, Nome, Descrição e Actions. A tabela contém três registros:

ID ↑	Nome	Descrição	Actions
1	Doença rara	rara	 
2	Doença nova	nova	 
3	Doença do carapato	doença do carapato	 

Figura 14 – Tela de listagem de doenças.

### 5.1.4 Módulo animais

O módulo de animais é um recurso para registrar e gerenciar os dados dos animais da fazenda, permitindo controlar os animais recém nascidos e adultos, informando seus dados de acordo com suas particularidades.

Este módulo corresponde aos casos de uso de manter animais e nascimentos de animais, expondo suas funcionalidades ao usuário através das telas de cadastro de animais e listagem de animais.

A tela de cadastro de animais apresentada na Figura 15 é responsável por conter os campos para armazenar o nome ou código do animal, data de nascimento, informações familiares (pai e mãe) e o local de nascimento selecionando a fazenda e o lote que o animal pertence, informações da primeira pesagem, primeiro teste de hemograma, a ocorrência de doenças ou morte ao nascer.

The screenshot shows a mobile application interface for animal registration. The header is green with a back arrow and the text 'Caif Manager'. The user's name 'MARCIO LUCAS' is in the top right. The main title is 'Cadastro de Animal' with a subtitle 'Aqui você poderá fazer o cadastro dos animais.' Below this are three sections: 'Informações gerais' with fields for name, tag code, breed code, birth date, sex, and life stage; a 'É primogênito?' toggle; and search/dropdowns for mother and father. 'Fazenda de localização do animal' includes a farm dropdown, a lot dropdown, and an animal count field. 'Saúde' has an 'Está vivo?' toggle, a 'Primeira Pesagem' section with weight, date, PPT, hematocrit, and exam date fields, and a 'Doenças' section with a search field and a status dropdown. A green plus icon is at the bottom right.

Figura 15 – Tela de cadastro de animais.

A tela de listagem de animais apresentada na Figura 16 é responsável por apresentar o relatório de animais registrados permitindo serem visualizados, por meio de lista, todos os animais ou filtrá-los por nome ou código, por condições de vida (vivo ou morto), por qualidade de saúde (sadios ou doentes), por sexo (masculino ou feminino ) e por lote. Na lista, estão incluídas as funcionalidades edição e exclusão do cadastro dos animais.

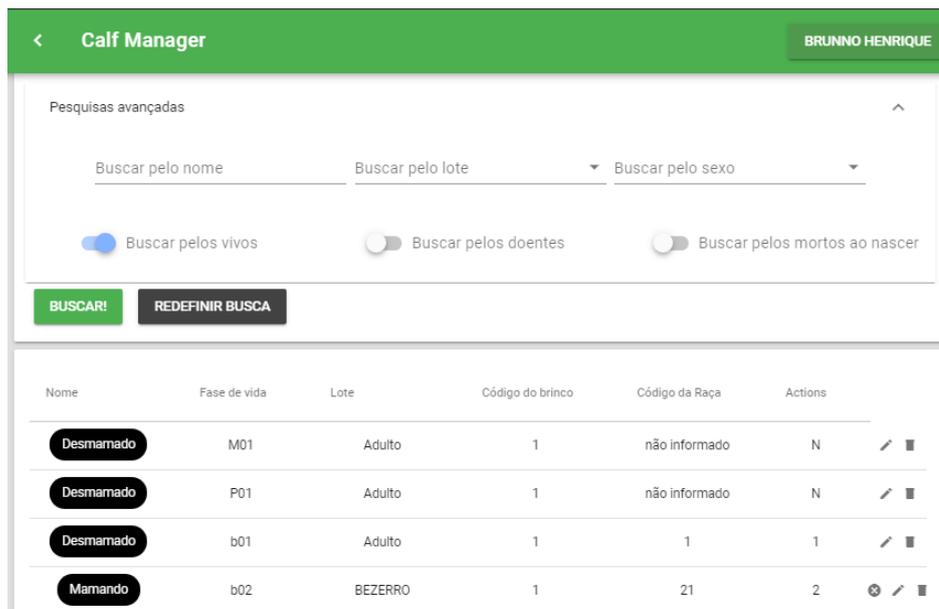


Figura 16 – Tela de listagem de animais.

### 5.1.5 Módulo enfermaria

O módulo enfermaria é um recurso para registrar e gerenciar os dados clínicos dos animais. Por meio dele, são obtidos as informações de animais preferencialmente doentes, obtendo seus dados cadastrais, dados de pesagem, resultados de exames de hemograma, dados de relações parentais e medicamentos/vacinas aplicadas no controle de doenças, apresentando seus históricos desde o nascimento ou primeiro registro.

Este módulo corresponde aos casos de uso de manter animais, manter nascimento e manter morte de animais e possui as funcionalidades para consultar os animais da fazenda utilizando os mesmos filtros apresentados na listagem de animais e acessar as informações de prontuário de cada animal cadastrado através da tela de prontuário.

A tela do prontuário apresentada nas Figuras 17, 18 e 19, é responsável por exibir todas as informações cadastrais do animal, a fazenda onde o animal está alocado, resultados de pesagens através de lista e gráfico, dados do histórico clínico de doenças e seu estado de saúde, dados de medicamentos/vacinas aplicados ao animal e a apresentação da árvore genealógica.

Calf Manager				BRUNNO HENRIQUE
<b>Prontuário animal</b>				
Aqui você poderá ver todas as informações pertencentes a este animal.				
<b>Informações gerais</b>				
Nome	Código do Brinco	Nascimento	Idade	
a11	11	13/03/2020	0 meses, 10 dias	
Código da raça	Sexo	Fase da vida	Vivo?	
11R	m	RECEM_NASCIDO	SIM	
<b>Histórico clínico</b>				
Nome ↑	Data de adoecimento	Data de cura	Situação	
Doença rara	2020-03-13		DOENTE	

Figura 17 – Tela do prontuário (parte 01).

Ultimas pesagens	
Peso ↑	Data da pesagem
3 @	13/03/2020
4 @	13/03/2020
4 @	15/03/2020
5 @	14/03/2020

[REGISTRAR DOENÇA](#)
[REGISTRAR VACINA](#)
[REGISTRAR CURA](#)
[REGISTRAR MORTE](#)

Vacinas Aplicadas			
Medicamento ↑	Prescrição	Data	Quantidade (Mg/ML)
remedio	remedio 12/12	2020-03-13 20:59:29	1.00

Figura 18 – Tela do prontuário (parte 02).

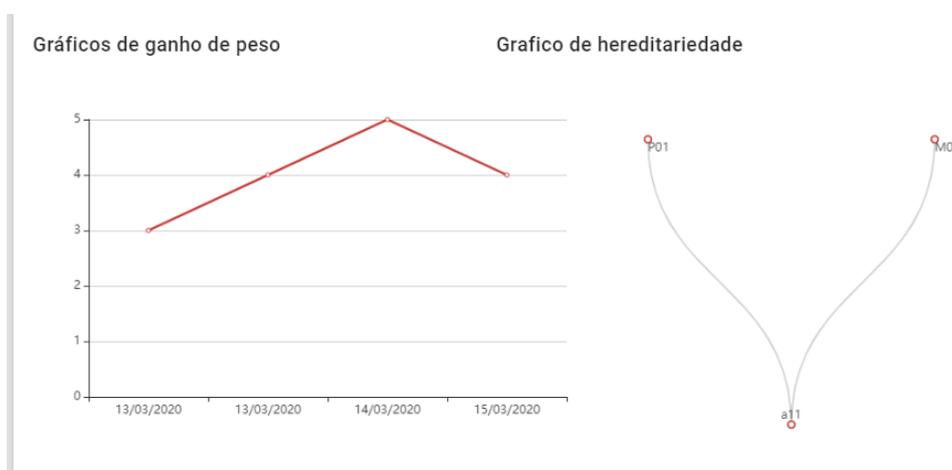
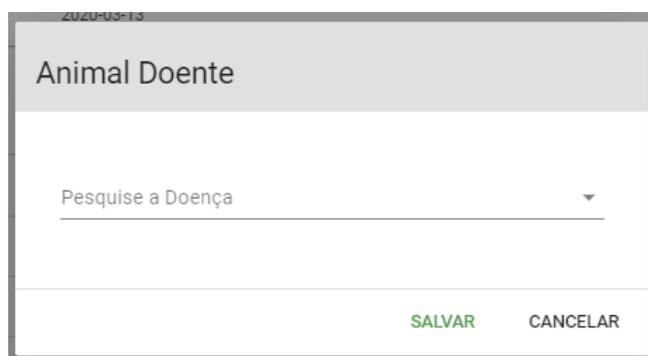


Figura 19 – Tela do prontuário (parte 03).

Na tela de apresentação do prontuário, além de apresentação dos dados correlacionados ao animal, também possui funcionalidades específicas de manipulação de dados referentes a sua saúde. Estas funcionalidades são: registrar doença, registrar medicação/vacinação, registrar cura e registrar morte. Na tela, elas são apresentadas por meio de botões de ação apresentados na Figura 17.

A funcionalidade de registrar doença fornece recursos, por meio de uma janela de diálogo, para que seja informado no sistema a doença contraída pelo animal, conforme apresentado na Figura 20. Os dados de adoecimento do animal, alteração do estado de saúde e a data de ocorrência são dados que são informados pela regra de negócio presente no servidor.



A imagem mostra uma janela de diálogo com o título "Animal Doente". No topo, há uma barra de busca com o texto "Pesquise a Doença" e uma seta para baixo. Abaixo, há dois botões: "SALVAR" em verde e "CANCELAR" em cinza.

Figura 20 – Janela de registro de adoecimento do animal.

A funcionalidade de registrar medicação/vacinação é um recurso para que seja informado para o sistema, por meio de uma janela de diálogo, qual foi a medicação/vacinação e a quantidade que foi aplicada ao animal como forma de tratamento ou prevenção contra doenças, conforme está apresentado na Figura 21.



A imagem mostra uma janela de diálogo com o título "Vacinar Animal". No topo, há uma barra de busca com o texto "Pesquisar pelo Medicamento" e uma seta para baixo. Abaixo, há um campo de entrada com o texto "Dose" e "Medida em miligramas". Abaixo, há dois botões: "SALVAR" em verde e "CANCELAR" em cinza.

Figura 21 – Janela de registro de medicação/vacinação no animal.

A funcionalidade de registrar cura é um recurso para que seja informado para o sistema, por meio de uma janela de diálogo, confirmando que o animal se encontra curado

de sua(s) doença(s), conforma apresentado na Figura 22.

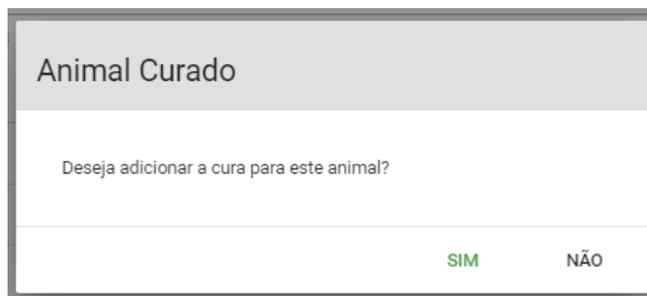


Figura 22 – Janela de registro de cura do animal.

A funcionalidade de registrar morte é um recurso para informar ao sistema confirmando que o animal não se encontra mais com vida, independentemente da situação, que pode ser analisada de acordo com seus históricos apresentados pelo prontuário ou ocorrência de alguma fatalidade. A Figura 23 apresenta a janela de diálogo confirmando a ocorrência da morte do animal.

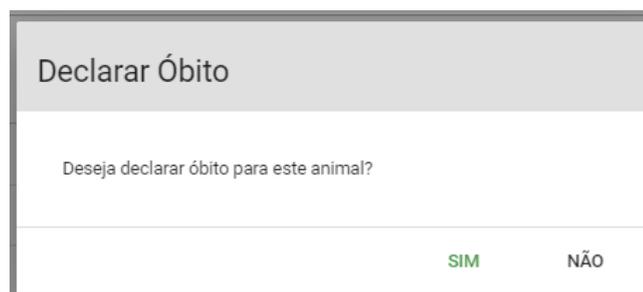


Figura 23 – Janela de registro de morte do animal.

### 5.1.6 Módulo hemogramas

O módulo hemogramas é um recurso que permite serem registrados os exames de hemogramas realizados em uma serie de animais. O registro de seus resultados de PPT e Hematócrito são registrados para controle sanitário e seus relatórios permitem o acompanhamento do bem-estar do animal.

Por meio dos relatórios, o produtor pode acompanhar a evolução de seus exames periódicos e atestar doenças caso seus resultados forem constantemente negativos em um período, antecipando-se para que seja aplicada uma decisão acertada visando o mínimo de prejuízo ao bem-estar do animal, consequentemente, minimizando prejuízos na produção.

Este módulo corresponde ao caso de uso manter testes sanguíneos e expõe suas funcionalidades ao usuário através das telas de cadastro e listagem de hemogramas.

A tela de cadastro de hemogramas, apresentada na Figura 24, tem a responsabilidade de cadastrar. Nela deve ser informado o animal que foi feito o teste, o resultado do teste de PPT, o resultado do teste de hematócrito e a data de realização.

Cadastrar Exame

Informações gerais

Pesquise o Ani... PPT Hematócrito Data do Ex...

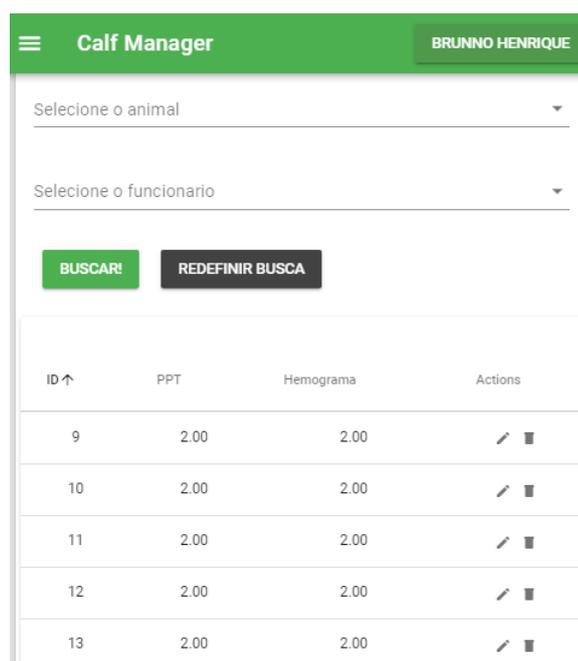
ENVIAR LIMPAR FORMULÁRIO

Figura 24 – Tela de cadastro de hemogramas.

A tela de listagem apresentada na Figura 25 é responsável por exibir o relatório de hemogramas por meio de gráficos, permitindo a visualização dos resultados dos testes de PPT e hematócrito, sendo necessário primeiramente selecionar o animal.



Figura 25 – Gráficos de resultados de testes de hemogramas.



ID ↑	PPT	Hemograma	Actions
9	2.00	2.00	 
10	2.00	2.00	 
11	2.00	2.00	 
12	2.00	2.00	 
13	2.00	2.00	 

Figura 26 – Tela de listagem de hemogramas.

### 5.1.7 Módulo pesagens

O módulo pesagem é um recurso para registrar e gerenciar as constantes pesagens realizadas nos animais para o acompanhamento da produção da fazenda. Seus relatórios lhe permitem o acompanhamento evolutivo do peso do animal, permitindo ao produtor tomar medidas hábeis de acordo com a progressão de sua pesagem, em casos de progressão positiva, arriscar melhorias, se não, tomar medidas corretivas.

Este módulo corresponde ao caso de uso manter ganho de peso e expõe suas funcionalidades ao usuário através das telas de cadastro e listagem de pesagens. A tela de cadastro apresentada na Figura 27 é responsável por cadastrar as pesagens informando o animal, o valor do peso e a data de realização.



Figura 27 – Tela de cadastro de pesagens.

A tela de relatório apresentada na Figura 26 é responsável por exibir o relatório de pesagens totais e a Figura 28, através de gráfico, permitindo a visualização dos dados das pesagens realizadas e os dados cadastrais básicos do animal, sendo necessário primeiramente selecionar o animal para se obter os dados e pesagens.

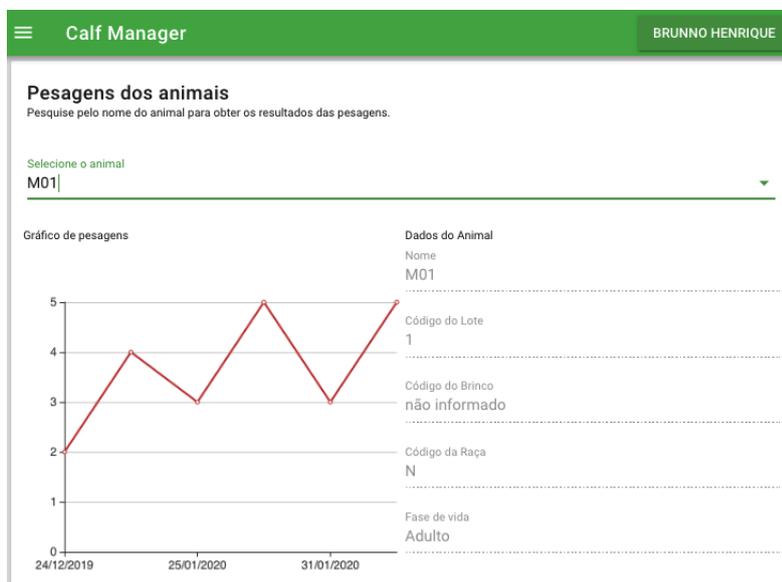


Figura 28 – Tela de relatório de pesagem.

### 5.1.8 Módulo medicamentos/vacinas

O módulo de medicamentos/vacinas permite ao usuário realizar o controle de medicamentos/vacinas que serão utilizadas no controle sanitário do rebanho. Por meio dele, podem ser registrados e gerenciados os medicamentos do estoque, registrar suas movimentações e utilização no combate ou prevenção a doenças.

Este módulo corresponde ao caso de uso manter medicamentos/vacinas e expõe ao usuário suas funcionalidades de controle de medicamentos/vacinas e seu estoque por meio das telas de cadastro e listagem de medicamentos/vacinas, entrada de medicamentos/vacinas e aplicação de dosagens.

A tela de cadastro de medicamentos/vacinas apresentada na Figura 29 é responsável por cadastrar os medicamentos utilizados nos tratamentos dos animais. Para cadastrar, devem ser informados o nome e a prescrição do medicamento.

The screenshot shows the 'Cadastro Medicamento/Vacina' form. At the top, there is a green header with a hamburger menu icon, the text 'Calf Manager', and the user name 'BRUNNO HENRIQUE'. Below the header, the title 'Cadastro Medicamento/Vacina' is displayed, followed by the instruction 'Preencha o formulário para cadastrar novos medicamentos'. Underneath, the section 'Informações gerais' contains two input fields: 'Nome do Medicamento' and 'Prescrição do Medicamento'. At the bottom of the form, there are two buttons: 'ENVIAR' and 'LIMPAR FORMULÁRIO'.

Figura 29 – Tela de cadastro de medicamentos/vacinas.

A tela de listagem apresentada na Figura 30 é responsável por apresentar o relatório de medicamentos/vacinas permitindo serem visualizados, por meio de lista, todos(as) os(as) medicamentos/vacinas registrados(as) e filtrá-los pelo nome. Nesta lista, estão incluídas as funcionalidades para edição e exclusão dos registros.

The screenshot shows the 'Pesquisa' (Search) screen. At the top, there is a green header with a hamburger menu icon, the text 'Calf Manager', and the user name 'BRUNNO HENRIQUE'. Below the header, the title 'Pesquisa' is displayed, followed by the instruction 'Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.'. Underneath, there is a search input field labeled 'Buscar pelo nome'. Below the input field, there are two buttons: 'BUSCAR!' and 'REDEFINIR BUSCA'. Below the buttons, there is a table with the following data:

ID ↑	Nome	Prescrição	Ações
13	Carrapaticida	aplicar de 12/12 hrs	 

Figura 30 – Tela de listagem de medicamentos/vacinas.

A Entrada de medicamentos/vacinas é uma funcionalidade responsável por registrar a entrada de produtos para o estoque, informando o produto, a quantidade de unidades ou pacotes. A Figura 31 apresenta a tela entrada de medicamentos/vacinas.

**Entrada de medicamentos/vacinas**

**Informações gerais**

Medicamentos/Vacinas Quantidade doses por unidade Quantidade unidades  
Pesquisar pelo ▾ Miligramas/mililitros UN / PCT

ENVIAR LIMPAR FORMULÁRIO

Figura 31 – Tela de entrada de medicamentos/vacinas.

A tela de cadastro de aplicações de dosagem apresentada na Figura 32 é responsável por cadastrar as aplicações de medicamentos/vacinas nos animais como forma de tratamentos ou prevenções. Para cadastrar devem ser informados o animal, o medicamento/vacina e a dosagem aplicada em miligramas (Mg) ou mililitros (ML).

**Sanidade Animal**

**Informações gerais**

Animais Medicamentos/Vacinas Dose QTD Doses  
an01 ▾ Carrapaticida ▾ 1 11

ENVIAR LIMPAR FORMULÁRIO

Figura 32 – Tela de cadastro de aplicação de dosagem.

A tela de listagem apresentada na Figura 33 é responsável por apresentar o relatório das aplicações de medicamentos/vacinas aos animais permitindo serem visualizados, por meio de lista, todos(as) os(as) registros de aplicações contendo o animal, o medicamento e a dosagem em miligramas (Mg) ou mililitros (ML). Nesta lista, estão incluídas as funcionalidades para edição e exclusão dos registros.



Animal	Medicamento	Quantidade (Mg/MI)	Actions
an01	Carrapaticida	1	 
a11	Carrapaticida	1	 
a122	Carrapaticida	1	 
a122	Carrapaticida	1	 

Figura 33 – Tela de listagem de aplicações.

### 5.1.9 Módulo funcionários

O módulo funcionários é um recurso para controle de acessos e recursos humanos, gerenciando os dados dos colaboradores da fazenda, o acesso a outros módulos por meio de permissões, garantindo segurança no acesso as informações.

Este módulo corresponde ao caso de uso manter funcionários e manter controle de acessos, nele estão dispostas as funcionalidades de CRUD, controle de acessos e autenticação de usuários do sistema por meio das telas de cadastro e listagem de funcionários, cargos, usuários, permissões, grupos de permissões, *dashboard* e página de *login*.

A tela de cadastro de funcionários apresentada na Figura 34 é responsável por registrar, informando seus dados pessoais, dados de endereço dos funcionários que terão acesso as funcionalidades do sistema.

**Calf Manager** BRUNNO HENRIQUE

### Pessoa

Nome \_\_\_\_\_ RG \_\_\_\_\_ CPF \_\_\_\_\_

Sexo  
Masculino ▼ Telefone \_\_\_\_\_

### Endereco

Logradouro \_\_\_\_\_ Número ou comple... \_\_\_\_\_ Bairro \_\_\_\_\_

Cidade \_\_\_\_\_ Estado ▼ País \_\_\_\_\_ CEP \_\_\_\_\_

### Alocação e Função

Pesquise a faz... ▼ Pesquise o ca... ▼ Salário \_\_\_\_\_

**ENVIAR** **LIMPAR FORMULÁRIO**

Figura 34 – Tela de cadastro de funcionários.

A tela de listagem apresentada na Figura 35 é responsável por apresentar o relatório de funcionários permitindo, por meio de lista, serem visualizados todos os registros e filtrá-los pelo nome. Nesta lista, também estão incluídas as funcionalidades para edição e exclusão dos registros.

**Calf Manager** BRUNNO HENRIQUE

### Pesquisa

Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.

Buscar pelo nome \_\_\_\_\_

**BUSCAR!** **REDEFINIR BUSCA**

ID ↑	Pessoa ↑	Fazenda	Cargo	Ações
1	Brunno Henrique	Fazenda Chapadao	Cargo	✎ 🗑
6	Pablo	Fazenda Chapadao	Cargo	✎ 🗑

Figura 35 – Tela de listagem de funcionários.

A tela de cadastro de cargos apresentada na Figura 36 é responsável por registrar os cargos definidos pelo setor de Recursos Humanos (RH), informando o nome e descrição do cargo.

The screenshot shows the 'Cadastrar Cargo' form in the Calf Manager application. The header is green with a hamburger menu icon, the text 'Calf Manager', and the user name 'BRUNNO HENRIQUE'. The main content area has a title 'Cadastrar Cargo' and a subtitle 'Preencha o formulário para cadastrar novas fazendas'. Below this is the 'Informações gerais' section with two input fields: 'Nome' and 'Descrição'. At the bottom, there are two buttons: 'ENVIAR' (green) and 'LIMPAR FORMULÁRIO' (grey).

Figura 36 – Tela de cadastro de cargos.

A tela de listagem apresentada na Figura 37 é responsável por apresentar o relatório de cargos permitindo, por meio de lista, serem visualizados todos os cargos registrados e filtrá-los pelo nome. Nesta lista, também estão incluídas as funcionalidades para edição e exclusão dos registros.

The screenshot shows the 'Pesquisa' list view in the Calf Manager application. The header is green with a hamburger menu icon, the text 'Calf Manager', and the user name 'BRUNNO HENRIQUE'. The main content area has a title 'Pesquisa' and a subtitle 'Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.'. Below this are two search input fields: 'Buscar pelo Id' and 'Buscar pelo nome'. At the bottom, there are two buttons: 'BUSCAR!' (green) and 'REDEFINIR BUSCA' (grey). Below the search fields is a table with three columns: 'ID ↑', 'Nome', and 'Descrição'. The table contains three rows of data, each with edit and delete icons.

ID ↑	Nome	Descrição		
10	Gestor	Gestor		
14	Tecnico de campo	Tec. de campo		
15	Tecnico de laboratório	Tecnico de laboratório		

Figura 37 – Tela de listagem de cargos.

A tela de cadastro de grupos de permissões apresentada na Figura 38 é responsável por registrar grupos que serão referenciados a um ou diversas permissões presentes no sistema, informando o nome e a descrição do grupo.



**Cadastrar Grupos**  
Preencha o formulário para cadastrar novos grupos.

**Informações gerais**

Nome  Descrição

**ENVIAR** **LIMPAR FORMULÁRIO**

Figura 38 – Tela de cadastro de grupos.

A tela de listagem apresentada na Figura 39 é responsável por apresentar o relatório de grupos de permissões permitindo, por meio de lista, serem visualizados todos os registros e filtrá-los pelo nome. Nesta lista, também estão incluídas as funcionalidades para edição e exclusão dos registros.



**Pesquisa**  
Faça uma pesquisa utilizando os filtros abaixo para um melhor resultado.

Buscar pelo nome

**BUSCAR!** **REDEFINIR BUSCA**

ID ↑	Nome	Ações
1	administrador	 
23	peões 2	 
24	Peões	 

Figura 39 – Tela de listagem de grupos.

A tela de cadastro de usuários apresentada na Figura 40 é responsável por registrar

os usuários do sistema, informando o funcionário, o grupo, o nome de usuário (*login*), senha e confirmação de senha.

**Cadastrar Usuário**  
Preencha o formulário para cadastrar novos funcionários

Pesquise o funcionário    Pesquise o grupo

Login    Senha    Confirma Senha

ENVIAR    LIMPAR FORMULÁRIO

Figura 40 – Tela de cadastro de usuários.

A tela de listagem apresentada na Figura 41 é responsável por apresentar o relatório de funcionários permitindo, por meio de lista, serem visualizados todos os registros e filtrá-los pelo nome. Nesta lista, também estão incluídas as funcionalidades para edição e exclusão dos registros.

Pesquise o Grupo

Pesquise o funcionário

BUSCAR!    REDEFINIR BUSCA

ID ↑	Login	Grupo	Ações
1	admin	administrador	 
6	pablo-vieira	Peões	 
7	pablo2	administrador	 

Figura 41 – Tela de listagem de usuários.

A tela de cadastro de permissões apresentada na Figura 42 é responsável por

registrar as permissões para os módulos do sistema, informando o nome do módulo, e as permissões para inserir, alterar, listar e excluir.



The screenshot shows a web application interface for 'Calf Manager'. At the top, there is a green header bar with a hamburger menu icon on the left, the text 'Calf Manager' in the center, and the user name 'BRUNNO HENRIQUE' on the right. Below the header, the main content area is titled 'Cadastro de Permissão' with a subtitle 'Preencha o formulário para cadastrar novas permissões'. Underneath, there is a section titled 'Informações gerais'. This section contains a text input field labeled 'Nome' and a dropdown menu labeled 'Pesquise o Grupo'. Below these fields, there are four checkboxes: 'Criar', 'Ler', 'Alterar', and 'Deletar'. At the bottom of the form, there are two buttons: 'ENVIAR' and 'LIMPAR FORMULÁRIO'.

Figura 42 – Tela de cadastro de permissões.

A tela de listagem apresentada na Figura 43 é responsável por apresentar o relatório de funcionários permitindo, por meio de lista, serem visualizados todos os registros e filtrá-los pelo nome. Nesta lista, também estão incluídas as funcionalidades para edição e exclusão dos registros.

A página inicial ou *dashboard* apresentada na Figura 44 é responsável por ser a primeira página de acesso após a autenticação do usuário e conter o menu de navegação na lateral esquerda da figura e no corpo central algumas informações gerenciais. Inclui também, o número total de lotes cadastrados, e quantidade de animais presentes e o relatório por meio de gráfico de pizza, informando suas taxas de acordo com seus quadros de saúde.

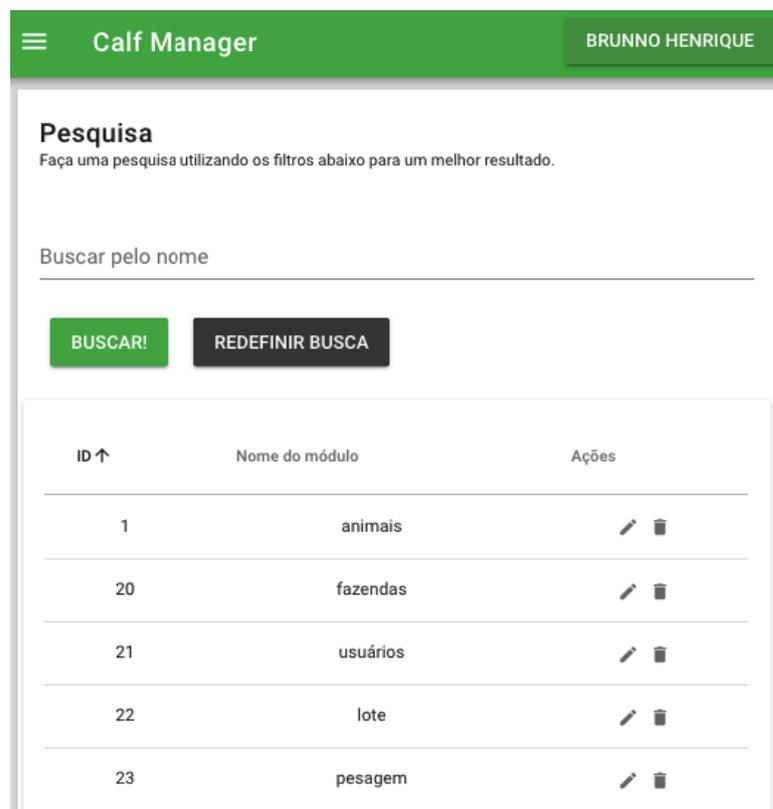


Figura 43 – Tela de listagem de permissões.

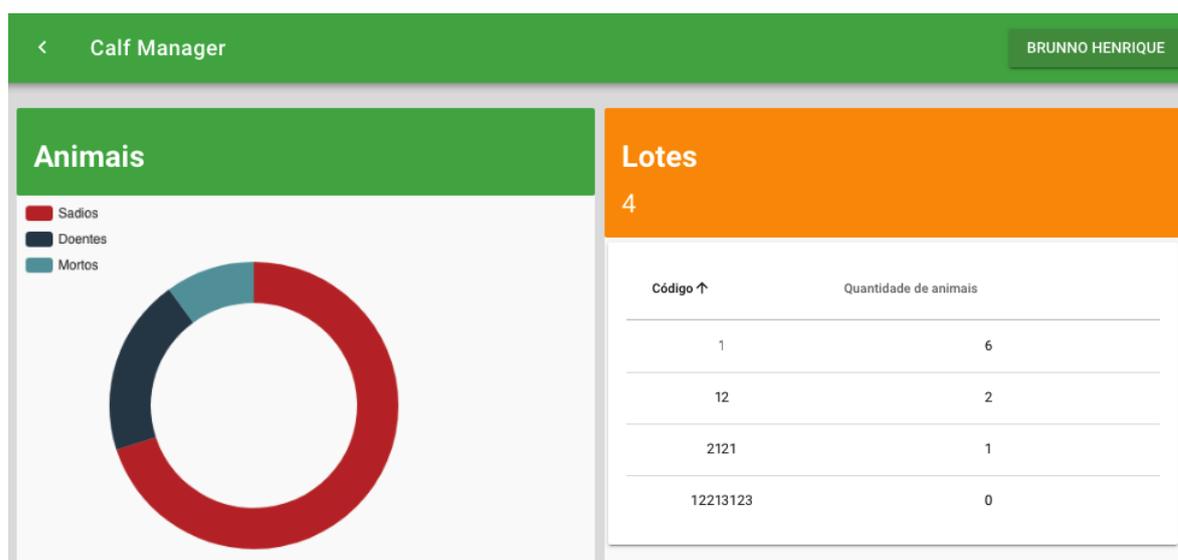
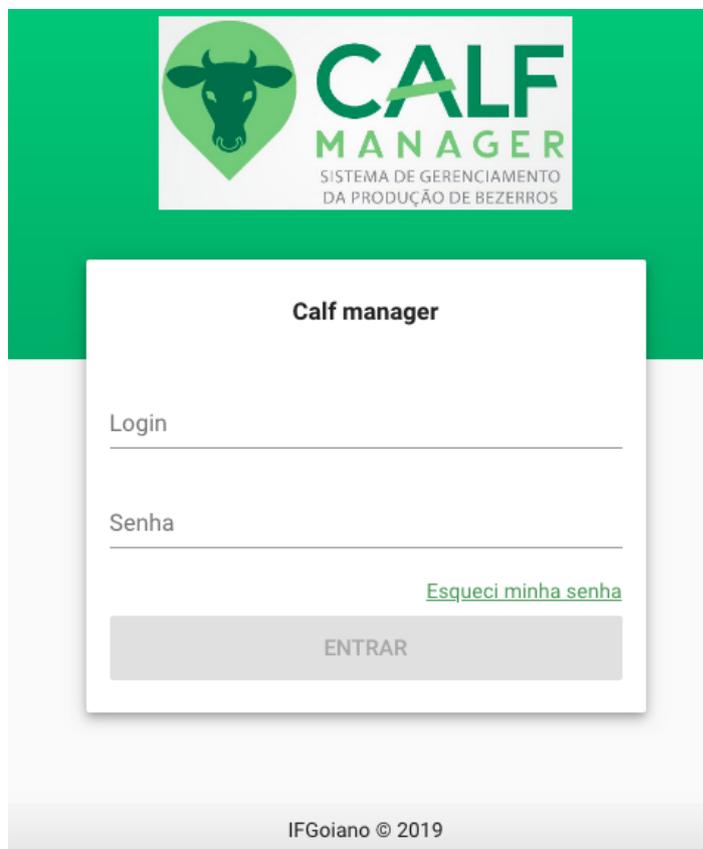


Figura 44 – Dashboard.

A tela de *login* apresentada na Figura 45 é responsável por validar as credenciais de *login* e senha do usuário. Caso as credências sejam válidas, o sistema verifica as permissões inseridas para o usuário e disponibiliza os links de navegação do menu de acordo com elas.



A imagem mostra a interface de login do sistema Calf Manager. No topo, há um cabeçalho verde com o logo de uma cabeça de boi verde e o texto "Calf MANAGER SISTEMA DE GERENCIAMENTO DA PRODUÇÃO DE BEZERROS". Abaixo, um formulário branco com o título "Calf manager" contém campos para "Login" e "Senha", um link "Esqueci minha senha" e um botão "ENTRAR". Na base do formulário, há o texto "IFGoiano © 2019".

Figura 45 – Tela de login.

Este capítulo apresentou a solução deste projeto, exibindo suas funcionalidades que, em conjunto, formam uma poderosa ferramenta no auxílio a administração e gerenciamento do manejo de bovinos. A seguir, no capítulo 7, será apresentadas as considerações finais.

## 6 Considerações finais

O desenvolvimento deste trabalho proporcionou analisar os efeitos positivos da aplicação das tecnologias voltadas para a construção de sistemas de gerenciamento e aplicá-las as necessidades do agronegócio buscando uma maior competitividade para o setor, dos índices de produção e da lucratividade.

O resultado final deste estudo foi a construção de uma aplicação *web* denominada Calf Manager destinada ao manejo da produção de bezerros, podendo ser gerenciados os animais em todos os seus ciclos de vida, podendo ser estipulado pelo administrador qual deverá ser o tempo de acompanhamento.

O Calf Manager fornece ao administrador e seus colaboradores funcionalidades para registrar as informações em um ponto único, substituindo os antigos procedimentos de registros que eram armazenados em folha de papel. Este novo método, facilita o acesso às informações e garante a integridade no armazenamento e controle de acesso as informações e auxiliando-o no processo de tomada de decisões relacionadas aos índices de produção e bem-estar animal.

Através do Calf Manager, além de eliminar os procedimentos antigos de registro e armazenamento de dados, permitimos aos gestores e colaboradores novas perspectivas de análise de dados, simplicidade no acompanhamento de históricos e uma infinidade de possibilidades manipulação e cruzamento que posteriormente poderão ser adicionadas como novas funcionalidades.

Além de funcionalidades para o controle das informações referentes ao manejo, o Calf Manager disponibiliza recursos para o gerenciamento de recursos humanos, com funcionalidades para controle e permissão de acesso aos colaboradores e gestores.



## 7 Trabalhos futuros

A área da tecnologia pode oferecer inúmeros recursos para o agronegócio, recomenda-se para trabalhos futuros a construção de um aplicativo *mobile* para oferecer aos colaboradores e gestores, uma ferramenta que possa ser acessada a partir de um dispositivo celular, abrindo-lhes os horizontes para que acessem suas informações sem a necessidade de um computador do tipo *desktop* ou *notebook*.

Outro benefício que esta solução atenderá, será a necessidade que um software *web* não consegue atender, pois sabemos que, no âmbito rural, o acesso a internet cabeada ou via-rádio podem não ser possíveis, principalmente nas regiões onde estão concentrados as atividades do manejo.

Para resolver, o aplicativo *mobile* deve conter uma tecnologia que permita o armazenamento das informações em memória local, para que sejam inseridas as informações no nele, independentemente se o dispositivo estiver conectado à internet, assim, uma vez que esteja novamente conectado a internet devem ser sincronizadas todas as informações que estavam armazenadas em memória local automaticamente.

Vislumbrando as possibilidades de recursos que este software pode oferecer, recomendamos a implementação das seguintes funcionalidades: cruzamento de informações, gerenciamento de custos financeiros e controle de validade dos medicamentos/vacinas.

A funcionalidade de cruzamento de dados, deve permitir ao usuário correlacionar as informações de doenças, com as alterações do PPT e hematócrito, suas pesagens e os dados de hereditariedade, apresentando suas informações em gráficos de doenças. Esta correlação de informações serão úteis para melhor identificação dos quadros de doenças que surgirem, oferecendo melhores possibilidades de análise de dados.

A funcionalidade de gerenciar os custos financeiros será um recurso importante para o acompanhamento dos custos de produção. Neste recurso, serão registrados e gerenciados as movimentações financeiras do controle de produção, incluindo aquisição e venda de produtos e serviços.

A funcionalidade de controle de validade dos medicamentos/vacinas, deve permitir ao usuário registrar e gerenciar no módulo de medicamentos/vacinas, a validade dos produtos adquiridos, necessitando notificar ao usuário de acordo com que as datas estiverem próximas de seu vencimento.

Munido desta tecnologia, o sistema pode oferecer um integral acompanhamento, proporcionando um gerenciamento refinado, permitindo coletar informações necessárias ao negócio em tempo real.



# Referências

- ADRIEL, W. Introdução ao framework php laravel. Disponível em: <<http://www.devmedia.com.br/introducao-ao-framework-php-laravel/33173>>. Acesso em: 20 Mai. 2016., 2015. Citado na página 33.
- ALEMIDA, B. R. D.; BANDEIRA, S. J. M. Desenvolvimento de um aplicativo para a plataforma android. Citado 2 vezes nas páginas 32 e 34.
- ALMEIDA, F. E. V. D. Campus quixadá bacharelado em engenharia de software. Citado na página 35.
- BAYLARI, A.; MONTAZER, G. A. Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications* 36(4), 8013-8021. <https://doi.org/10.1016/j.eswa.2008.10.080>, 2009. Citado na página 35.
- BELÉM, P. H. A. et al. Escolha de um framework para a linguagem de programação php através do método ahp clássico. In: *Anais X Congresso Nacional de Excelência em Gestão*. [S.l.: s.n.], 2014. v. 8. Citado na página 33.
- BENTO, E. J. *Desenvolvimento web com PHP e MySQL*. [S.l.]: Editora Casa do Código, 2014. Citado 2 vezes nas páginas 30 e 31.
- CARRIERI, A. d. P.; LIMA, J. Processos decisório e processos de decisão-ação em organizações rurais. *Encontro Anual da Associação Nacional dos Programas de Pós-Graduação em Administração (anpad)*, p. 235–247, 1992. Citado na página 23.
- CHEN, X. et al. Restful api architecture based on laravel framework. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2017. v. 910, n. 1, p. 012016. Citado 3 vezes nas páginas 32, 33 e 34.
- ELMASRI, R. et al. *Sistemas de banco de dados*. Pearson Addison Wesley São Paulo, 2005. Citado na página 24.
- FERREIRA, H. K.; ZUCHI, J. D. Análise comparativa entre frameworks frontend baseados em javascript para aplicações web. *Revista Interface Tecnológica*, v. 15, n. 2, p. 111–123, 2018. Citado na página 35.
- FIELDING, R. T.; TAYLOR, R. N. *Architectural styles and the design of network-based software architectures*. [S.l.]: University of California, Irvine Irvine, 2000. v. 7. Citado na página 32.
- INCAU, C. *Vue.js: Construa aplicações incríveis*. Casa do Código, 2017. Citado na página 35.
- JOBSTRAIBIZER, F. *Guia Profissional PHP*. [S.l.]: Digerati Books, 2009. Citado na página 33.
- JONES, M. B. The emerging json-based identity protocol suite. W3Cworkshop on identity in the browser. p. 1-3., 2011. Citado na página 36.

JONES M., B. J. S. N. Json web token (jwt) – rfc 7519. Internet Engineering Task Force, IETF. Disponível em: <<https://tools.ietf.org/pdf/rfc7519.pdf>>., 2015. Citado na página 36.

JWT. Json web token introduction. Disponível em: <<https://jwt.io/introduction/>>., 2017. Citado na página 36.

LACERDA, M. No mar de libs e frameworks: conhecendo o vue.js - parte i. BrazilJS. Disponível em: <<https://braziljs.org/blog/no-mar-de-libs-e-frameworks-conhecendo-o-vue-js-parte-i>>. Acesso em: 22 out. 2017., 2017. Citado na página 35.

LI, D. et al. Echarts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*, Elsevier, v. 2, n. 2, p. 136–146, 2018. Citado 2 vezes nas páginas 35 e 36.

LINGE N. PARSONS, D. Problem-based learning as an effective tool for teaching computer network design. *IEEE Transactions on Education*, 49(1), 5-10. <https://doi.org/10.1109/te.2005.852600>, 2006. Citado na página 35.

LUCAS, V. Valitron: validação fácil que não suga, github. Disponível em: <<https://github.com/vlucas/valitron>>. Acesso em 30/11/2019, 2014. Citado na página 34.

MACHADO, J. G.; NANTES, J. F.; ROCHA, C. E. Um estudo multicaso na pecuária de corte: o processo de informatização na produção da carne bovina. *Revista Brasileira de Agroinformática*, 2002. Citado na página 21.

MACHADO, J. G.; NANTES, J. F. D.; ROCHA, C. E. O processo de informatização das propriedades rurais: um estudo multicaso na pecuária de corte. *Revista Brasileira de Agroinformática*, 2002. Citado na página 21.

MELLO, D. A. P. d. *Solução para monitoramento ambiente utilizando Arduino*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2017. Citado 3 vezes nas páginas 30, 31 e 33.

MILANI, A. *MySQL-guia do programador*. [S.l.]: Novatec Editora, 2007. Citado na página 31.

MINETTO, E. L. Frameworks para desenvolvimento em php. *São Paulo: Novatec*, p. 34, 2007. Citado 2 vezes nas páginas 30 e 33.

MONTANHEIRO, L. S.; CARVALHO, A. M. M.; RODRIGUES, J. A. Utilização de json web token na autenticação de usuários em apis rest. XIII Encontro Anual de Computação. Anais do XIII Encontro Anual de Computação EnAComp, p. 186–193, 2017. Citado 2 vezes nas páginas 32 e 36.

MONTOYA, M. A.; ROSSETTO, C. R. *Abertura econômica e competitividade no agronegócio brasileiro: Impactos regionais e gestão estratégica*. [S.l.]: UPF Editora, 2002. v. 1. Citado na página 23.

- NETO, A. C. E.-M. Gestão de sistemas de produção de bovinos de corte: índices zootécnicos e econômicos como critérios para tomada de decisão. *Maringá, PR. Disponível em <http://www.ruralcentro.com.br/Artigos> (Acesso em 20/03/2009)*, 1999. Citado na página 24.
- NIEDERAUER, J. Integrando php 5 com mysql. *São Paulo. Novatec Editora Ltda*, 2005. Citado na página 31.
- OLIVEIRA, J. S. D.; ZANINE, A. D. M.; SANTOS, E. M. Fisiologia, manejo e alimentação de bezerros de corte. *Arquivos de Ciências Veterinárias e Zoologia da UNIPAR*, v. 10, n. 1, 2007. Citado na página 23.
- PORTA, H. d. S. L. Planejando viagens: desenvolvimento do sistema web para planejamento de viagens. 2018. Citado 2 vezes nas páginas 34 e 35.
- REYES, J. Wonderfully creative uses for php. Acesso em: 20 mai. 2016. Disponível em: <<http://code.tutsplus.com/tutorials/15-wonderfully-creative-uses-for-php-net-4714>>., 2009. Citado na página 30.
- RIBEIRO, M.; FRANCISCO, R. Web services rest: Conceitos, análise e implementação. *Educação, Tecnologia e Cultura-ETC*,(14), 2016. Citado na página 32.
- RICARDO, L. R. Google android, aprenda a criar aplicações para dispositivos móveis com o android sdk. *Pg-20 a*, v. 21, n. 3, 2015. Citado na página 32.
- RODRIGUES, J. Importância e aplicação dos sistemas de informação no agribusiness. AGROSOFT 99 – FEIRA E CONGRESSO DE INFORMÁTICA APLICADA À AGROPECUÁRIA E AGROINDÚSTRIA, Anais..., 6p., 1999. Citado na página 21.
- SCHNEIDER, A. H. Desenvolvimento web com client side rendering: combinando single page application e serviços de backend. Monografia, Universidade Federal do Rio Grande do Sul. Rio Grande do Sul, Brasil., 2016. Citado na página 36.
- SHAH, B. et al. Designing of website along with blog and forum using laravel. Oulun ammattikorkeakoulu, 2019. Citado 2 vezes nas páginas 34 e 35.
- SHELDON, A. O. R. Sql: Um guia para iniciantes. 3. ed. Rio de Janeiro: Editora ciência moderna Ltda., 2009. Citado na página 31.
- SILVA, G. A. S. d. *Desenvolvimento de um aplicativo móvel para mediação na Educação a Distância*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2019. Citado na página 35.
- SILVA, J. W. d. *FineClin: sistema de gerenciamento de uma clínica multidisciplinar*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2017. Citado na página 33.
- SILVA, V. M. da. Revisão sistemática da evolução mvc na base acm. 2012. Citado na página 31.
- SONG, J.; ZHANG, M.; XIE, H. Design and implementation of a vue. js-based college teaching system. *International Journal of Emerging Technologies in Learning (iJET)*, v. 14, n. 13, p. 59-69, 2019. Citado na página 35.

- SOUKI, G. Q.; XISTO, E. M. S.; SALAZAR, G. T. Reflexões sobre o custo de oportunidade da informação no setor agropecuário. CONGRESSO BRASILEIRO DE ECONOMIA E SOCIOLOGIA RURAL DA SOBER, 37,, 1999. Citado na página 21.
- SUNDARSHAN, S.; SILBERSCHATZ, A.; KORTH, F. H. *Sistemas de Banco de Dados*. 3ª. Edição. [S.l.]: Editora Makron Books, 1999. Citado na página 24.
- SUPPI, L. F. P. Desenvolvimento de aplicativos utilizando o framework zend. *REPOSITÓRIO DE RELATÓRIOS-Sistemas de Informação*, v. 1, n. 2, 2012. Citado na página 33.
- TABLELESS. O que é javascript? Disponível em: <<http://tableless.github.io/iniciantes/manual/js/>>. Acesso em: 14 nov, 2018. Citado na página 34.
- WEBER, W. Modification and use of nasap to aid in teaching network design. *IEEE Transactions on Education*, 12(4), 232-234. <https://doi.org/10.1109/te.1969.4320511>, 1969. Citado na página 35.
- YAMAGUCHI, L. C. T. et al. *Perspectivas da informatização rural no brasil*. 2016. Citado na página 21.