UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS SOBRAL

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

IAGO MAGALHÃES DE MESQUITA

SISTEMA WEB DE CONTROLE SELETIVO DE ANEMIA EM OVINOS

SOBRAL

2025

IAGO MAGALHÃES DE MESQUITA

SISTEMA WEB DE CONTROLE SELETIVO DE ANEMIA EM OVINOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica e da Computação do Programa de Pós-Graduação em Engenharia Elétrica e da Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e da Computação. Área de Concentração: Sistemas de Informação

Orientador: Dr. Iális Cavalcante de Paula Júnior

SOBRAL

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M544s Mesquita, Iago.
Sistema Web de controle seletivo de anemia em ovinos / Iago Mesquita. – 2026.
59 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Sobral, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Sobral, 2026.
Orientação: Prof. Dr. Iális Cavalcante de Paula Júnior.

1. Método FAMACHA. 2. Haemonchus Contortus. 3. Anemia. 4. Sistema de gerenciamento. 5. Aplicativo móvel. I. Título.

CDD 621.3

IAGO MAGALHÃES DE MESQUITA

SISTEMA WEB DE CONTROLE SELETIVO DE ANEMIA EM OVINOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica e da Computação do Programa de Pós-Graduação em Engenharia Elétrica e da Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e da Computação. Área de Concentração: Sistemas de Informação

Aprovada em:

BANCA EXAMINADORA

Dr. Iális Cavalcante de Paula Júnior (Orientador)
Universidade Federal do Ceará (UFC)

Dra. Larissa Ferreira Rodrigues Moreira
Universidade Federal de Viçosa (UFV)

Dr. Reuber Régis de Melo
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos meus pais e irmãos que sempre me apoiaram e se dedicaram no auxílio de meus estudos e por formar a pessoa que sou hoje.

AGRADECIMENTOS

Agradeço a Deus, por cada oportunidade de “tentar”. De levantar da cama e da um passo após o outro e superar meus medos. Assim como em Mateus 14:27, é necessário coragem para enfrentar as águas revoltas da vida e seguir adiante. E que mesmo em momentos mais difíceis e obscuros da minha vida me permitiu seguir em frente, pegar a cruz e continuar vivendo, superando as mais diversas crises de ansiedade e medo. Possibilitando realizar um grande sonho de mais uma conquista acadêmica.

À minha família, em especial minha mãe, Irislane Magalhães de Souza, por sempre me apoiar nos meus mais diversos projetos, por acreditar que cada sonho era possível mesmo que distante, por me apresentar a máquina que respondia qualquer perguntar, o computador, e despertar em uma criança que vivia no interior a curiosidade de uma vida inteira para compreender como ela funcionava. À meu pai, Marcos Heleno de Mesquita, por sempre esta comigo, correndo atrás do ônibus, acordando cedo e por me esperar todas as noites para voltar para casa. Aos meus irmãos, Antônio Thiago Magalhães de Mesquita e Maria Isabela Magalhães de Mesquita, por todo o apoio e por acreditarem que tudo daria certo.

Aos meus amigos de longa data, Gabriel Ferreira, Antônio Marcos e Igor Alves, pelos seus conselhos e conversas que alegravam meus dias na escola.

Aos meus professores do fundamental, Inaldo Pontes , Lindoneide Sousa e Tadeu Sousa, por acreditarem que um dia eu “iria longe”.

Ao meu querido amigo e professor Maquinele Furtado, pelos inúmeros conselhos, oportunidades e por acreditar em meu primeiro empreendimento. Pela oportunidade de ser voluntário em um projeto que leva vida e alegria a muitos. Que sua vontade e seus sonhos continuem fazendo parte da liga Ecoar e que de onde você esteja continue sendo feliz.

Aos meus amigos de ensino médio, Carlos Henrique, Bruno Rodrigues, Mariana Rodrigues, Nathalia Stelita e Pedro Henrique, pelas conversas, por estudarem comigo para os vestibulares e ENEM, por me ajudarem a ser uma pessoa melhor, compreender o mundo da tecnologia e a nunca desistir diante das dificuldades.

Aos meus professores de ensino médio, Francisco Das Chagas, Márcia Furtado, Marcelo Magalhães e Joelma Furtado, por todo tempo e dedicação. Pelas aulas extras em turnos de folga e dias de descanso. Por sempre acreditarem e ajudarem no processo.

Aos meus professores de curso técnico, Danilo Alves, Edivan Nunes, Jamilson Oliveira, Willian Ribeiro e Dilso, por me apresentarem o incrível mundo da tecnologia, desen-

volvimento e da automação. Por me mostrar que era possível mesmo em uma cidade pequena do interior desenvolver projetos e tecnologia, conhecer o mundo da robótica e programação. E me ajudarem a crescer tanto de forma pessoal como profissional. Em especial ao Willian Ribeiro, por me dizer que uma mente barulhenta e bagunçada um dia iria se ajeitar e conquistaria o que sonhava. Ao meu querido amigo Danilo Alves, por me incentivar e me apresentar os caminhos até aqui.

Aos meus queridos amigos de faculdade, Ana Karoline, Sarah Frota, Rhuan Nunes, Iranilson Nascimento, Humberto Pontes, Gabriel Nascimento, Francislane Teles, Felipe Monte, Edney Perreira, Wilson Frota e Cesar Augusto e muitos outros pelo caminho, pela companhia, conversas, trabalhos, tempo e me ajudaram a alcançar objetivos que só existiam em meus sonhos. Em especial a Ana Karoline e Sarah Frota, pelas longas conversas, projetos e reflexões sobre a vida e seus desafios.

Aos meus amigos de ônibus, Ariane Fernandes, Nonailton Silva, Maria Emanuele, Francisco Fernandes, José Neto e Izabel Veiga, por tornarem a longa viagem em conversas divertidas e alegria, por sempre me esperarem, perdirem ao motorista para esperar, guardarem lugar, pelos piqueniques e aventuras em Sobral.

À meus professores e amigos de graduação, Dr. Kleber César, Ms. Aldinei Perreira, Ms. Francilino Araújo, Ms. Fabiano Ribeiro e Ms. Leornado Tabosa, por acreditarem no meu potencial, pelo inúmeros conselhos, por serem mais que professores de sala de aula e me ajudarem em diversos desafios da vida. Por toda atenção em meus mais diversificados projetos extra aula, por sempre estarem disponíveis para indicar conteúdos, escreverem artigos e desenvolverem projetos, E por fim, pelos incentivos e conselhos na vida pessoal e profissional.

Aos meus amigos de UFC, João Vitor, Vanessa Carvalho e Francilândio Lima, por me mostrarem que computação é mais que código e por me acolherem em um ambiente novo. À Simone, da recepção por sempre me acolher todas as manhãs e por me dar todos os avisos que não recebia.

Aos meus amigos de mestrado, Acácio Salustiano, Kattiely Melo, Flávia Galvão, Allan Freire e Danillo Fernandes, pela amizade, companheirismo, boa vontade, pelos momentos nunca vividos de imensa alegria e surpresa, e por acreditarem nos projetos e trabalhos.

À professora Larissa Rodrigues, por acreditar em um desconhecido que chegou por e-mail. Por ajudar a melhorar a escrita, desenvolver artigos, a ter uma nova visão sobre pesquisa e pós-graduação. Ao professor Wendley Sousa por toda ajuda e confiança no desenvolvimento

de trabalhos.

À meu orientador Dr. Iális Cavalcante, por toda a orientação nesses dois anos tanto no projeto, em trabalhos diversos, cursos, disciplinas e na vida. Por me apresentar a UFC, por acreditar no meu potencial e por toda à paciência.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio e incentivo financeiro no desenvolvimento desta pesquisa.

Agradeço à minha namorada, Alice Sousa, por todo apoio, carinho, conversas e por acreditar no processo. Por me ensinar que a vida é repleta de coisas boas, que posso sentir felicidade nas coisas simples e que posso superar medos e desafios. Obrigado pela tranquilidade e por me fazer sentir paz.

E por fim, agradeço a todos que contribuíram ao longo dessa jornada. De alguma forma a sua ajuda foi importante e tornou essa realização possível. E como aprendi com o professor Kleber César, o conhecimento de uma pessoa é moldado por muitas mãos e cada pessoa que passou pela minha vida me ajudou de alguma forma. A todos, meus sinceros agradecimentos.

“Santo é um pecador que nunca desiste”

(São João Paulo II)

RESUMO

O *Haemonchus Contortus* é nematoide que ataca diversos animais, em especial pequenos ruminantes como os ovinos. Sendo um dos principais causadores de anemia nestes animais, doença essa que ocasiona perda de peso, de lã e redução da produção de leite. Trazendo prejuízos financeiros ao produtor e a redução de produtividade. Atualmente existem muitas formas de sanar essa problemática por meio da utilização de vermífugos que controle esse parasita de forma eficiente, porém, o seu uso deve ser controlado e aplicado corretamente para que o verme não crie resistência ao material. Infelizmente, para muitos casos não é o que acontece. Muitos dos criadores aplicam esses produtos de forma indiscriminada, aumentando a dose ou a reduzindo, causando o aumento da resistência do parasita. Muitas vezes à correta identificação de anemia no rebanho é fator primordial para o controle, visto que muitas vezes os animais só são diagnosticados quando estão perdendo peso, com perda de lã e bastante debilitados. Uma forma eficiente de identificar se o animal está anêmico é utilizando o cartão FAMACHA, que auxilia o criador ao identificar a se o animal está doente ou não através de uma escala de cores que ao ser comparada com a cor da mucosa ocular será possível definir o estado dele. Porém, o acesso a esse cartão possuí dificuldades como custo, uso e interpretação. Pensando nessas problemáticas e pelo fato de que a criação de ovinos é uma grande fonte de renda para o povo nordestino, o seguinte trabalho apresenta como proposta uma metodologia que une um sistema web para controle seletivo de anemia em ovinos com um aplicativo. Onde esses dois trabalham de forma unificada, onde o aplicativo permite capturar imagens da mucosa ocular dos ovinos e retorna se o animal está com anemia ou não. E o sistema web permite o gerenciamento mais completo e robusto, sendo possível analisar, controlar e compreender o comportamento da propagação da anemia causada pelo verme na região, gerar relatórios, utilizar modelos de *Machine Learning* e Visão Computacional para realizar a detecção da mucosa ocular e classificar-la em saudável ou não.

Palavras-chave: Método FAMACHA. Haemonchus Contortus. Anemia. Sistema de gerenciamento. Aplicativo móvel.

ABSTRACT

Haemonchus contortus is a nematode that infects various animal species, particularly small ruminants such as sheep. It is one of the main causative agents of anemia in these animals, a condition that leads to weight loss, wool reduction, and decreased milk production, thereby causing financial losses to farmers and a reduction in overall productivity. Currently, several strategies exist to mitigate this problem, mainly through the use of anthelmintics that effectively control this parasite. However, their administration must be carefully managed and correctly applied to prevent the development of resistance. Unfortunately, in many cases this is not observed. A considerable number of farmers apply these products indiscriminately, either by increasing or reducing the prescribed dose, which contributes to the selection of resistant parasite populations. Accurate identification of anemia within the herd is often a critical factor for effective control, as animals are frequently diagnosed only after exhibiting severe clinical signs such as significant weight loss, wool depletion, and general debilitation. One effective tool for early identification of anemia is the FAMACHA© card, which assists farmers in determining the health status of animals by comparing the color of the ocular mucosa with a standardized color scale. Nevertheless, access to this tool presents challenges related to cost, proper use, and correct interpretation. Considering these issues, and given that sheep farming represents an important source of income for the population of Northeastern Brazil, the present study proposes a methodology that integrates a web-based system for selective anemia control in sheep with a mobile application. These two components operate in a unified manner: the application enables the capture of ocular mucosa images and provides immediate feedback on whether the animal is anemic, while the web system offers more comprehensive and robust management capabilities. The platform allows for the analysis, monitoring, and understanding of the dynamics of anemia propagation caused by the parasite in the region, in addition to generating reports and incorporating Machine Learning and Computer Vision models to detect the ocular mucosa and classify it as healthy or anemic.

Keywords: AMACHA© method. Haemonchus contortus. Anemia. Management system. Mobile application.

LISTA DE FIGURAS

Figura 1 – *Haemonchus Contortus*. 21
Figura 2 – Cartão FAMACHA. 22
Figura 3 – Fluxograma de coleta de imagens. 30
Figura 4 – Coletas de imagens em campo. 31
Figura 5 – *Dataset* com imagens de mucosas oculares. 33
Figura 6 – Fluxograma de detecção de objetos. 34
Figura 7 – Fluxograma do algoritmo de classificação. 36
Figura 8 – Imagem da mucosa ocular com dimensões de 64 x 64 *pixels*. 40
Figura 9 – Fluxograma com funcionamento do aplicativo. 41
Figura 10 – Tela inicial do aplicativo *Stop Verme*. 42
Figura 11 – Tela de cadastro de propriedade. 43
Figura 12 – Tela de resultado. 44
Figura 13 – Tela *Home* do sistema de gerenciamento. 45
Figura 14 – Tela de *Login*. 46
Figura 15 – Tela de cadastro do sistema de gerenciamento. 46
Figura 16 – Tela de recuperação de senha. 47
Figura 17 – Tela de redefinição de senha. 47
Figura 18 – Tela de *dashboard*. 48
Figura 19 – Tela com *popup* no *dashboard*. 49
Figura 20 – Tela de relatório. 49
Figura 21 – Tela de análise de imagem. 51
Figura 22 – Tela de análise de imagem com resultados. 51
Figura 23 – Tela de relatório. 52
Figura 24 – Tela de página não encontrada. 53

LISTA DE TABELAS

Tabela 1 – Configuração da Arquitetura Proposta 35
Tabela 2 – Métricas para avaliação dos modelos 37
Tabela 3 – Resultados do detector de objetos *Yolo V5 nano*. 39
Tabela 4 – Resultados dos classificadores. 41
Tabela 5 – Dados coletados nas propriedades. 50
Tabela 6 – Valores das métricas de avaliação dos modelos. 52

LISTA DE SÍMBOLOS

API	*Application Programming Interface*
APP	*Aplicativo*
BPNN	*Backpropagation Neural Network*
CNN	*Convolutional Neural Network*
CSS	*Cascading Style Sheets*
DL	*Deep Learning*
DOM	*Document Object Model*
DWG	*Peso médio diário*
EMBRAPA	*Empresa Brasileira de Pesquisa Agropecuária*
GPS	*Global positioning system*
HC	*Haemonchus Contortus*
HTML	*HyperText Markup Language*
IA	*Inteligência Artificial*
IBGE	*Instituto Brasileiro de Geografia e Estatistíca*
INPI	*Instituto Nacional da Propriedade Industrial*
IOT	*Internet Of Things*
JS	*Java Script*
JSONB	*Java Script Object Notation*
ML	*Machine Learning*
NIRS	*Near Infrared Spectroscopy*
OPG	*Ovos por grama*
PCV	*Packed Cell Volume*
PDF	*Portable Document Format*
ROI	*Region of Interest*
SGBD	*Sistemas de Gerenciamento de Banco Dados*
SVM	*Support Vector Machine*

UFC	*Universidade Federal do Ceará*
UI	*User Interface*
WEB	*World Wide Web*
VC	*Visão Computacional*

SUMÁRIO

1 INTRODUÇÃO 16
1.1 Trabalhos relacionados 17
1.2 Objetivos 19
1.2.1 Objetivo Geral 19
1.2.2 Objetivos Específicos 19
1.3 Produção Científica 19
1.4 Estrutura do Trabalho 19
2 FUNDAMENTAÇÃO TEÓRICA 21
2.1 Haemonchus Contortus 21
2.2 Cartão FAMACHA 22
2.3 Front End 23
2.3.1 Java Script 23
2.3.2 React JS 24
2.4 Back End 24
2.4.1 Python 25
2.4.2 Postgre SQL 25
2.4.3 Flask 26
2.5 Sistemas de Machine Learning e Visão Computacional 26
2.5.1 Machine Learning 26
2.5.2 Visão Computacional 27
2.5.3 Dashboards 27
2.6 Aplicativo móvel 28
2.6.1 Kotlin 28
2.6.2 TensorFlow Lite 28
3 METODOLOGIA 30
3.1 Base de dados 30
3.1.1 Coletas em campo 31
3.1.2 Rotulação das imagens 32
3.1.3 Banco de imagens 33
3.2 Modelos de Machine Learning e Visão Computacional 33

3.2.1 *Modelos de detecção de objetos* 33
3.2.2 *Modelos de classificação* 35
3.2.3 *Métricas de avaliação* 36
3.3 **Plataforma de gerenciamento** 37
3.3.1 *Aplicativo móvel* 37
3.3.2 *Plataforma web* 37
4 **RESULTADOS** 39
4.1 **Resultados dos modelos de detecção de objetos** 39
4.2 **Resultados dos modelos de classificação** 40
4.3 **Resultados do aplicativo móvel** 41
4.4 **Resultados do sistema de gerenciamento web** 45
5 **CONCLUSÃO** 54
REFERÊNCIAS 56

1 INTRODUÇÃO

Anemia causada pelo ataque do nematoide *Haemonchus Cortortus* (HC) é uma das principais doenças que atinge pequenos ruminantes como ovinos e caprinos, gerando problemas como perda de peso, redução na produção de lã e leite (SILVA *et al.*, 2022). Segundo a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) e o Instituto Brasileiro de Geografia e Estatística (IBGE) a região nordeste concentra grande parte dos rebanhos de ovinos e caprinos do pais. Sendo cerca de 60% do rebanho de ovinos e 91.6% de caprinos (BORCHARTT; RIBEIRO, 2023). Muitos desses rebanhos são de pequenos e médios produtores, que sofrem com a falta de assistência técnica e ferramentas para identificação de doenças como à anemia. O que acarreta na aplicação de forma descontrolada de vermífugos, com uso de sub e sob dosagens, ou na sua negligência.

Apesar de ser uma doença já conhecida, à anemia ainda causa danos severos. Em muitos casos o uso incorreto do vermifugo é um dos fatores que dificultam o controle do HC nas propriedades, que muitas vezes é aplicado por repetidas vezes, com dosagens baixas ou muito altas, em animais doentes e saudáveis. Visto que ao avistar um animal doente, em geral, todo o rebanho é vermifugado. O que acarreta no aumento da resistência dos parasitas e a perda do poder de ação do vermifugo.

Para tentar sanar essa problemática e auxiliar na identificação do ataque de HC surgiu o cartão FAMACHA, que através de uma escala de cores é possível determinar se o animal esta doente ou não (MAIA *et al.*, 2014). Onde basta expor a mucosa ocular do animal e realizar a comparação com a escala de cores presente no cartão. Infelizmente o acesso ao cartão FAMACHA é inviável por boa parte dos pequenos produtores devido ao seu valor e informações de uso. Outros problemas como impressão, forma de visualização e posicionamento do mesmo leva a erros de interpretação que acabam dificultando o diagnóstico e tratamento.

Outras alternativas que tem se mostrado eficientes na identificação da presença de HC é o uso da tecnologia, desenvolvendo ferramentas com a capacidade de identificação de anemia em animais ou informando sobre o uso do cartão FAMACHA (BORCHARTT; RIBEIRO, 2023). Utilizando técnicas de *Machine Learning* (ML), *Deep Learning* (DL) e Visão Computacional (VC) que auxiliam na identificação de anemia de forma automática e mais rápida e precisa.

Em vista de tais fatores esse trabalho propõem desenvolver um sistema web que possibilite analisar dados e imagens obtidos por um aplicativo chamado *Stop Verme* (NóBREGA, 2025), que realiza captura de imagens de mucosas oculares de ovinos e as classifica, informando

aos usuários se o animal tem anemia ou não. A plataforma web trabalha em conjunto com o aplicativo (app) gerenciando dados, gerando relatórios e trazendo informações visuais como plotagem dos locais das propriedades em que as imagens foram coletadas. Além dessas funcionalidades, é possível gerar relatórios sobre as coletas realizadas, contendo *insights* como localização, quantidade animais, número de animais doentes e o vermífugo utilizado no local.

1.1 Trabalhos relacionados

Na literatura, pode-se encontrar diversos trabalhos com abordagem do uso de *Machine Learning*, Visão Computacional, *Deep Learning*, aplicações móveis e web para aplicações veterinárias. Tanto para detecção de doenças, quanto para estimar peso, altura e contagem de animais através de imagens. Para a detecção de anemia em ovinos existem muitas estratégias para o diagnóstico.

Em (SANTOS *et al.*, 2023), os autores propuseram predizer os valores de hematócrito (PCV) através da espectroscopia no infravermelho próximo (NIRS) e desenvolver um modelo de classificação e diagnóstico da infecção por *Haemonchus Contortus* em cordeiros através de valores de PCV, contagem de ovos por grama de fezes (OPG) e ganho de peso médio diário (DWG). Como resultado, o projeto desenvolveu dois modelos multivariados sendo um de regressão linear quantitativa e outro de classificação, o diagnóstico que utilizou dados de PCV, OPG e DWG para a classificação de animais doentes e saudáveis.

Em (SOUZA *et al.*, 2023), apresenta um método de prevenção de doenças gastrointestinais em pequenos ruminantes visando auxiliar à agricultura familiar. Visando sanar a dificuldade de pequenos produtores ao acesso a orientações técnicas sobre o uso do método FAMACHA. O trabalho propõe o uso de ML para realização da classificação de animais como anêmicos ou saudáveis. Os autores desenvolveram um aplicativo que coleta a imagem da mucosa do animal e utilizando técnicas como rede neural e *Support Vector Machine* (SVM), obtiveram acurácia de 83% e 87%, respectivamente.

Em (PESSANHA *et al.*, 2020), é desenvolvida uma metodologia utilizando *Convolutional Neural Network* (CNN) para identificação de dor em ovinos. O trabalho utiliza uma ampla gama de poses da cabeça do animal. Após a coleta de imagens, uma CNN ajustada é aplicada para detecção facial, seguida por uma estimativa de pose baseada em CNN e um método de localização de marcos informado pela pose. Em seguida, são extraídas características multimodais, combinando a aparência de regiões de interesse, descritas usando um Histograma

de Gradientes Orientados, com características geométricas e os valores de pose, resultando em um classificador SVM binário.

Em (ROCHA *et al.*, 2024) é proposto o uso de Visão Computacional, através do *Yolo V8 nano* para a segmentação de mucosas oculares de ovinos e caprinos. Após a segmentação é feita a classificação da região de interesse com o algoritmo *Random Forest*. Esses algoritmos foram treinados em uma base de dados com 450 imagens coletadas em campo. Durante os treinamentos os resultados obtidos apresentam acurácia de 64%, precisão de 66%, recall de 77% e F1 Score de 71%. Porém, a segmentação ficou limitada devido as variações de luminosidade, o que interfere no resultado final.

No trabalho de (SIDDIQUE *et al.*, 2025), foram utilizadas 4700 imagens de mucosas oculares de cabritos machos coletadas de forma semanal por um período de seis meses obtidas através de um *Smartphone*, para treinamentos de modelos de classificação para automatizar o processo do cartão FAMACHA. Os algoritmos analisados foram *Support Vector Machine*, *Backpropagation Neural Network* (BPNN) e *Convolutional Neural Network*. De forma comparativa foram utilizadas métricas de avaliação para selecionar o modelo com melhor desempenho. Dentre eles a CNN apresentou o valor de precisão de 97,8%.

Dentre os trabalhos analisados é possível perceber que muitos utilizam técnicas como *Machine Leaning* e *Deep Learning* alinhados com técnicas de Visão Computacional. Apresentando metodologias para segmentação da mucosa ocular ou de forma manual, extraindo a região de interesse afim de facilitar o uso da imagem por parte dos modelos de classificação e reduzir o poder de processamento. E por fim, classificando essa imagem de entrada com presença de anemia ou não. A metodologia empregada se assemelha e difere apenas em técnicas utilizadas para detecção de mucosa e classificações. Outro ponto importante é relação ao conjunto de dados que ainda apresenta uma quantidade de amostras muito baixa na maioria dos trabalhos. Neste trabalho, além de fornecer ferramentas para automatização do processo de identificação da anemia em ovinos é pensado no pós coleta de resultado, trazendo informações em forma de relatório para entender a disseminação do ataque de *Haemonchus Contortus* e quais vermífugos são utilizados e ainda surtem efeito. Possibilitando que seja geradas metodologias mais eficientes para controle desta doença e parasita.

1.2 Objetivos

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo geral desenvolver e analisar o uso de uma plataforma web para gerenciamento e controle de dados de animais, que tiveram imagens coletadas em campo por meio do aplicativo *Stop Verme* para identificação de anemia.

1.2.2 Objetivos Específicos

- Organizar dados de usuário e de coleta de mucosas de ovinos em uma plataforma web;
- Implementar técnicas de Sistemas de Gerenciamento de Banco Dados (SGBD) para gerenciar usuários e dados cadastrados;
- Aplicar ferramentas de geovisualização para compreensão de dados de localização de propriedades, uso de vermífugos e dispersão de animais com anemia na região;
- Analisar o desempenho de algoritmos de detecção de objetos e classificação de imagens de mucosas de ovinos em ambiente web;
- Gerar relatórios com base em *insights* obtidos através de dados obtidos pelo aplicativo *Stop Verme*.

1.3 Produção Científica

A partir dessa pesquisa e colaborações de demais pesquisadores da UFC e EMBRAPA foi realizado em outubro de 2025 o lançamento do aplicativo *Stop Verme* e se encontra disponível para uso da comunidade e demais pesquisadores. Além de obter no mesmo ano o seu registro de software junto ao Instituto Nacional da Propriedade Industrial (INPI) com o título: “Leitor de mucosas para identificação de anemia em pequenos ruminantes (*StopVerme*)”.

1.4 Estrutura do Trabalho

O restante do trabalho visa apresentar o estudo, ferramentas, metodologias utilizadas e análise dos resultados, dividindo-se em cinco capítulos. Sendo dividido da seguinte forma:

- **Capítulo 2:** Neste capítulo é apresentado o conjunto de ferramentas e tecnologias utilizadas para o desenvolvimento deste trabalho. Abrangendo conceitos essenciais para entender o agente causador da anemia em ovinos e como identificar à anemia causada por este.

Além de abordar as linguagens de programação e *frameworks* utilizados na plataforma de gerenciamento e aplicativo. Também é apresentado sobre sistemas que utilizam algoritmos de *Machine Learning* e Visão Computacional.

- **Capítulo 3:** Apresenta a metodologia empregada para o desenvolvimento da plataforma, detalhando o passo a passo e como cada ferramenta foi utilizada. Expondo como foi feita a coleta de imagens, seleção e montagem do banco de imagens utilizadas no treinamento dos modelos de classificação e detecção de objetos. Além de trazer como esses algoritmos foram avaliados. E por fim, é mostrado o passo a passo do desenvolvimento do aplicativo e da plataforma web.
- **Capítulo 4:** Descreve os resultados obtidos no desenvolvimento e implementação do projeto. Analisando o desempenho de modelos de classificação e de detecção de objetos com base nas principais métricas de avaliação disponíveis na literatura e com base no seu desempenho em uma aplicação *end to end*. Também é analisado o uso da plataforma por usuários e como é o seu desempenho fora do ambiente de produção.
- **Capítulo 5:** Por fim, é apresentado as conclusões sobre os experimentos e metodologias utilizadas. Trazendo as principais contribuições e propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

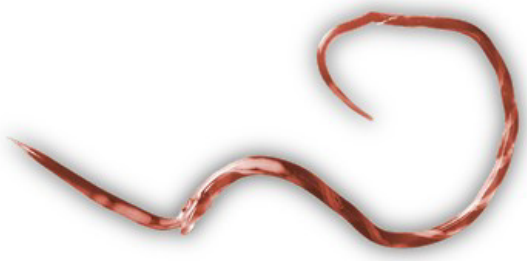
Nesta seção será apresentado uma visão detalhada sobre o verme *Haemonchus Contortus* e seu impacto em pequenos ruminantes, como os ovinos. Em seguida é demostrado o método de identificação de anemia causada pelo ataque do verme, através da utilização do cartão FAMACHA. Também é detalhado nessa seção conceitos sobre *Front End*, a linguagem de programação e *frameworks* utilizados para o desenvolvimento do projeto. Além disso, também é detalhado sobre *Back End* e as ferramentas empregadas. Ainda é apresentado sobre sistemas que utilizam *Machine Learning* e *Computer Vision*, buscando representar a utilização dessas técnicas por parte de usuários não especialistas. Por fim, é detalhado sobre as tecnologias empregadas no desenvolvimento do aplicativo.

2.1 Haemonchus Contortus

O parasita *Haemonchus Contortus* é um nematóide de grande impacto na ovinocultura (NIKOLAOU; GASSER, 2006). Ele ataca pequenos ruminantes como ovinos e caprinos (CAIN *et al.*, 2023). Se alimentando do sangue do animal, podendo consumir até 100mL por dia, dependendo da quantidade de vermes presentes. Tornando a presença desse parasita uma das principais causas de anemia nesses animais (CHAGAS *et al.*, 2007).

O HC chega a medir até 30mm de comprimento e seu ciclo de vida é rápido, podendo chegar até três semanas. A sua reprodução é por meio de ovos que são liberados pelas fêmeas e saem pelas fezes dos animais (BVMS PHD, 2021). Possibilitando assim que outros animais ao pastarem acabem sendo infectados e reiniciando o ciclo. A Figura 1 apresenta uma fotografia do verme.

Figura 1 – *Haemonchus Contortus*.



Fonte: Adaptado de (BOSQUES, 2022)

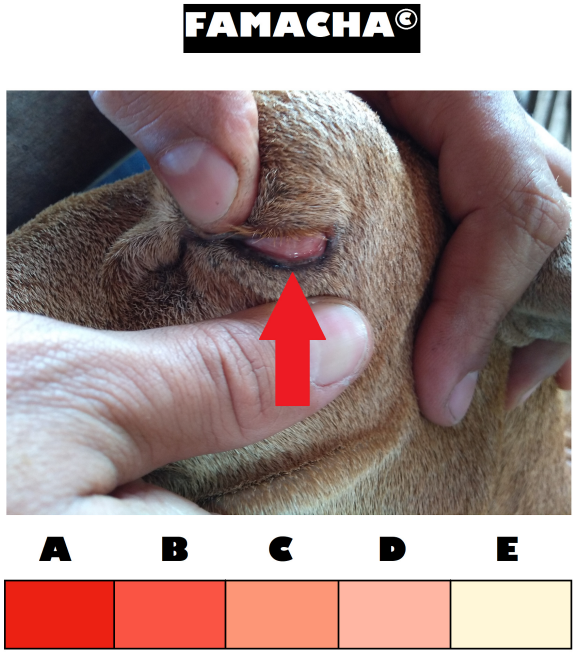
O impacto econômico para criadores de ovinos é evidente devido a perda de peso, redução de lã e produção de leite. Chegando a causar perdas de animais e redução da produção (BVMS PHD, 2021).

O tratamento para o controle do HC se dar por meio de anti-helmínticos químicos, tendo ação eficiente e rápida. Porém, o uso incorreto destes vermífugos causa o aumento da resistência do parasita (ADAMU *et al.*, 2020). Que acaba gerando imunidade e o difícil controle. Aspectos como subdosagens, sobredosagens e elevado números de aplicações estão entre as causas do aumento da resistência aos produtos registrados para o controle do nematoide.

2.2 Cartão FAMACHA

O método FAMACHA© foi criado por pesquisadores da África do Sul para auxiliar na identificação da anemia em pequenos ruminantes, como ovinos e caprinos, causada pelo nematoide HC (MAIA *et al.*, 2014). Essa metodologia se mostra eficaz na identificação desta doença através da cor da mucosa ocular. Utilizando uma escala de cores, que varia em cinco níveis. A Figura 2 apresenta uma representação do cartão FAMACHA, demonstrando sua escala de cores e a correta exposição da mucosa ocular do animal.

Figura 2 – Cartão FAMACHA.



Fonte: Adaptado de (BATH; WYK, 2001)

Para expor a mucosa do animal, deve-se utilizar as mãos para ajudar que a parte

inferior da mucosa ocular fique bem visível e exposta. Alguns cuidados devem ser tomados no processo, como o tempo de exposição e a força utilizada para isso. Um longo período para a realização do processo pode ocasionar a redução de circulação sanguínea e alterar a cor da mucosa, prejudicando o resultado do teste.

Para interpretar o cartão FAMACHA basta comparar a cor da mucosa exposta com os cinco níveis do cartão. Na Figura 2 os níveis são representados por A, B, C, D e E. Ao realizar a comparação das cores, os graus A e B estarão indicando que o animal analisado apresenta boas condições de saúde, sem anemia, com uma forte coloração vermelha e com presença de bastante de sangue. Já o nível C representa um estado utilizados por muitos criadores como estado de dúvida, em muitos casos já sendo aportado como fator para vermifugação. Nos níveis D e E já apontam anemia no ruminante com gravidade e ausência de sangue na mucosa devido sua coloração branca. Neste caso, o animal deve ser vermifugado imediatamente.

2.3 Front End

O *Front End* é a parte visual de um sistema computacional, onde o usuário pode interagir de forma prática e visual. Ele é responsável pelas telas, botões, formulários, gráficos e etc (EIS, 2015). Para isso, se utiliza de diversas linguagens, tanto de marcação, como *HyperText Markup Language* (HTML), as de estilo, como *Cascading Style Sheets* (CSS), e por fim, as de programação, como o *Java Script* (JS), *PHP* e etc. O trabalho em conjunto dessas linguagens permite a criação desses sistemas para aplicações web e *Mobile*.

2.3.1 Java Script

O *Java Script* é uma linguagem de programação leve e interpretada. Também baseada em objetos e conhecida como uma linguagem de *script* para navegadores (CONTRIBUTORS, 2025). Ela é popularmente utilizada para aplicações web, porém, a linguagem já consegue ser aplicada em ambientes fora dos navegadores.

O JS foi criado em 1995 pela *Netscape Communications Corporation*, inicialmente chamada de Mocha, posteriormente *LiveScript* e por fim *Java Script* (GRILLO; FORTES, 2008). A proposta era criar uma linguagem de processamento de modo cliente.

O JS é capaz de criar diversas aplicações, alinhados com *frameworks* como *React JS*, *React Native*, *Next JS* e entre outros. Seu principal ambiente de execução é na web, porém,

já é possível desenvolver aplicações *Mobile* e *Desktop* com ele. Além disso, essa linguagem é interpretada e não compilada. Para utilizar-la basta apenas digitar o código-fonte que o interpretador irá executar linha por linha do *script*. Como vantagem, se tem a facilidade para alterar o código, porém, existe uma desvantagem que é a perda de desempenho comparado as linguagens compiladas.

2.3.2 React JS

O *framework React JS* foi desenvolvida pelo *Facebook* visando a criação de interfaces de usuário (UI) voltados para aplicações web (FERREIRA *et al.*, 2024). Utilizando o conceito de componentes, permitindo seu reaproveitamento em diferentes partes do projeto e que ao serem combinados formam telas, páginas e aplicativos completos (META, 2025).

O *React JS* trabalha com o *Document Object Model* (DOM) para renderizar e atualizar os elementos de forma eficiente e rápida. Permitindo aos desenvolvedores utilizarem uma linguagem de domínio específico, o *Java Script XML* ou simplesmente JSX. O que possibilita o uso tanto das linguagens de marcação e estilo juntos com a parte lógica do sistema (FERREIRA *et al.*, 2024).

Além de trabalhar utilizando componentes é possível alinhar este *framework* com outras linguagens, como *Node*, *Python* e outras linguagens que rodem ao lado de servidores. Permitindo assim, que uma aplicação desenvolvida com *React Js* seja implementada e executada em um servidor.

Entre as vantagens deste *framework* temos a possibilidade da reutilização dos componentes, comunidade grande e ativa, DOM e compartibilidade com bibliotecas de terceiros. Já entre as desvantagens, as constantes atualizações dificultam o aprendizado, necessidade de bibliotecas de terceiros e dependência de conhecimento em *Java Script*.

2.4 Back End

O *Back End* é a parte do programa que fornece a lógica, gerenciando banco de dados, processamento dos dados, servidores e demais ações, fornecendo ao usuário tudo que ele precisa. Basicamente, ele é a parte invisível da aplicação, contudo, tudo passa por ele e depende do mesmo. Muitas linguagens de programação são utilizadas na parte lógica de um sistema, tais como *Python*, *Java Script*, *Java*, *PHP* e muitas outras.

Além de controlar o processamento de dados, gerenciar bancos de dados e servidores, o *Back End* ainda trabalha com *Application Programming Interface* (API) que permitem a comunicação entre diferentes sistemas e aplicações. Um exemplo prático é a comunicação do *Back End* com o *Front End*. Onde o *Back End* por meio da API fornece ao *front* dados essenciais, como informações de usuários, dados cadastrados em banco de dados, além de funcionalidades, como cadastros, autenticação e demais funcionalidades necessárias para o sistema.

2.4.1 Python

A linguagem de programação *Python* foi criada em 1980 por Guido Van Rossum, com o objetivo de ter comandos simples e fáceis de compreender (KALINOWSKI *et al.*, 2023). Ela é uma linguagem interpretada e possui uma curva de aprendizado pequena em relação a outras linguagens disponíveis.

Esta linguagem é do tipo dinâmica e fracamente tipada, ou seja, o próprio *Python* entende o tipo de dado a ser salvo em uma variável ao longo do programa sem precisar defini-la. Outro ponto importante salientar, é que esta linguagem é interpretada e de alto nível. Além, de ser orientada a objetos (ORG, 2025).

Suas aplicações são diversas, tanto para aplicações *Desktops*, web, *Mobile*, Microcontroladores, Ciência de dados e ML. Sendo bem popular para desenvolvimento de APIs utilizando diversos *frameworks* como *Flask* e *Django*. Mais o ponto que trás sua maior popularidade é por sua sintaxe fácil de aprender e compreender. Facilitando assim a compreensão de algoritmos mais complexos como em Ciência de Dados e ML, que ganham destaque com a utilização desta linguagem. Devido de se tratar de uma linguagem leve, rápida e por contar com uma infinidade de bibliotecas e ferramentas que auxiliam nestas tarefas.

2.4.2 Postgre SQL

O *Postgre SQL* é um banco de dados relacional de código aberto (GROUP, 2025b). Sendo um poderoso SGBD, desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley (GROUP, 2025c). Tendo evoluído do antigo sistema *Postgres*, o *Postgre SQL* foi lançado em definitivo em 1996 (GROUP, 2025a).

Sua aplicabilidade vão desde de campos como serviços financeiros, manufatura, varejo e logística, tendo como vantagens a capacidade de trabalhar com grande quantidade de dados e gerenciar-los. Possuindo aplicações com dados relacionais, Associação de *JavaScript*

Object Notation (JSONB) e geoespacial (AZURE, 2025). Sendo robusto, seguro e extensível.

Entre suas vantagens, está a capacidade de fornecer suporte a texto, imagens, sons e vídeos. O que aumenta sua popularidade entre os mais diversos públicos. Também é flexível e com compartibilidade com diversas linguagens de programação.

2.4.3 Flask

O *Flask* é um micro *framework* voltado para o desenvolvimento de aplicações web (FLASK'S, 2025). Visando possibilitar ao desenvolvedor criar ferramentas de formas fáceis e rápidas. Compatível com o *Python* e demais bibliotecas.

Com um grande número de exemplos, comunidade ativa e facilidade de utilização, este *framework* é ideal para projetos que necessitem de facilidade e rapidez no desenvolvimento. Sendo robusto e tendo uma curva de aprendizado rápida.

O *Flask* possibilita ao usuário desenvolver APIs, aplicações webs completas, com renderização de páginas, ferramentas de *login*, cadastro e autenticação. Além de possibilitar a conexão com diversos banco de dados. Sendo adequado para projetos de diversos tamanhos.

2.5 Sistemas de Machine Learning e Visão Computacional

Diversos sistemas utilizam ferramentas com ML e VC, muitas delas já no nosso dia a dia. Sendo muito comum em aplicações tanto web, móveis e até mesmos embarcados em dispositivos de hardware. Em muitos casos é possível a junção dessas áreas para a solução de uma determinada problemática, ou seja, junta algoritmos de ML e VC para solucionar um problema. Que vão desde classificação, detecção, rastreamento e etc.

2.5.1 Machine Learning

Machine Learning é área da computação que visa fazer com que as máquinas aprendam com dados, ou seja, é um campo que permite os computadores aprederem sem ser explicitamente programado (GéRON, 2019).

Diferentes das abordagens tradicionais da programação, que utiliza basicamente uma lista de regras para solucionar uma problemática, o ML trás a capacidade de adaptação ao programa. Podendo atualizar as respostas fornecidas através da obtenção de novos dados para treinamento, sem a necessidade de uma lista de regras complexas.

O Aprendizado de Máquina pode ser aplicado em diversos setores, do mais simples ao mais complexos, sendo empregado no dia a dia de muitas aplicações. Tanto na web, dispositivos móveis e Desktop, funcionando ao lado de servidores ou do próprio usuário.

Entre os problemas que podem ser solucionados com ML, existem os problemas de classificação, que são determinar a que grupo um determinado dado pertence. E os de regressão que englobam tarefas de previsão, como prever um preço através de um conjunto de informações, por exemplo.

Entre as suas vantagens pode-se citar a alta assertividade, segurança, robustez e confiança. Em contra partida, os pontos negativos são a necessidade de um grande número de dados, poder computacional e mão de obra qualificada para o seu desenvolvimento.

2.5.2 Visão Computacional

A Visão Computacional é um campo da ciências da computação que visa fazer com que máquinas consigam simular a visão humana (GONZALEZ R. C.; WOODS, 2009). Alinhada com técnicas clássicas e modernas, o uso de VC abrange diversos campos, como saúde, agricultura, indústria e muitos outros. Muitos dos algoritmos utilizados neste campo são compostos pela junção com ferramentas de ML. O que possibilita o desenvolvimento de uma abrangência maior de ferramentas, melhoria e confiabilidade nos resultados obtidos.

A VC moderna engloba desde aplicações em conjunto com ML para classificação de imagens ou vídeos, detecção de objetos, segmentação, rastreamento e etc. Muito desses avanços se devem a *Deep Learning* e as Redes Neurais Convoluncionais que unem os conceitos de VC e Redes Neurais para solucionar tarefas e problemáticas de forma automatizada.

2.5.3 Dashboards

Dashboards são interfaces que trazem informações de forma gráfica aos usuários, demonstrando dados e indicadores de negócios e processos ((EBAC), 2024). O que permite analisar, entender dados e gerar informações através destes de forma rápida, segura e eficiente. Sendo utilizado em diversos setores, principalmente em indústrias, negócios e gestão. Suas aplicações são desde analíticas, financeiras, operacionais, gerenciais, métricas, estratégicas e táticas.

Com essa ferramenta é possível monitorar o desempenho de uma fábrica, indústria ou processo, possibilitando monitorar e prever os ganhos e gastos. Resolver problemas de formas

mais rápidas e práticas, além de ser uma fonte para planejamento e melhoria das ações realizadas.

2.6 Aplicativo móvel

Aplicativos móveis possuem diversas aplicações, sendo empregados em muitas áreas. Muito comum para redes sociais, bancos, jogos, educação e etc, facilitando o acesso à informações e gerando praticidade.

Para realizar o desenvolvimento deste de tipo de programa podesse utilizar muitas linguagens de programação e *frameworks*. Entre as linguagens mais populares temos o *Java* para *Android* e o *Swift* para *IoS*, ambas as linguagens fornecem suporte ao desenvolvimento nativo de aplicações aos seus referentes sistemas operacionais. Com o rápido crescimento do uso e disseminação de aplicações móveis, surgiram novas tecnologias que possibilitam o desenvolvimento desse tipo de ferramenta de forma rápida e prática. Entre os mais populares temos o *React Native*, *Angular* e muitos outros.

2.6.1 Kotlin

O *Kotlin* é uma linguagem de programação multiplataforma desenvolvida pela empresa *JetBrains*, que utiliza a linguagem *Java* como referência (JETBRAINS, 2025). Ela pode ser utilizada tanto para desenvolver aplicativos multiplataforma, sendo para *Android* e *IoS*. Além disso, o *Kotlin* fornece suporte a desenvolvimento web e aplicações de Inteligência Artificial.

Entre suas vantagens está a sua segurança e eficiência, assim como o próprio *Java* tem em aplicações nativas para dispositivos móveis. Podendo acessar de forma mais rápida recursos do dispositivo, como sensores e atuadores. Outro ponto importante é o seu poder ao carregar bibliotecas nativas e de terceiros e arquivos pré-treinados de Inteligência Artificial (IA).

2.6.2 TensorFlow Lite

O *TensorFlow Lite* é uma biblioteca desenvolvida pelo *Google* em 2015, que une recursos de ML, computação númerica e muitas outras. Desenvolvido especificamente para operar em dispositivos móveis, incorporados e de *Internet of Things* (IoT) (ORG, 2021). Permitindo a melhor utilização do hardware disponível e otimizando modelos existentes.

O *TensorFlow Lite* pode ser utilizado com diversas linguagens, incluindo *Java*, *Python*, *JavaScript* e etc. A sua utilização é simplificada e ainda permite a conversão de modelos

já existentes no próprio *TensorFlow* para *TensorFlow Lite*. Reduzindo o poder de processamento necessário em grandes aplicações, como por exemplo, o *Yolo*, que exige grandes requisitos de hardware e com essa redução é possível ser executado em um dispositivo com menor poder de processamento.

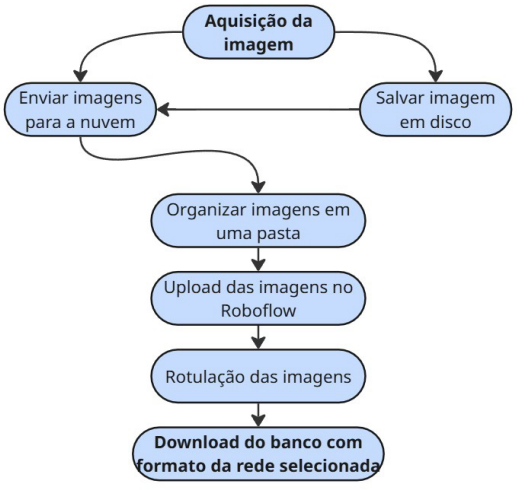
3 METODOLOGIA

Neste capítulo será abordado sobre a metodologia utilizada para o desenvolvimento do projeto. Apresentando os materiais e métodos utilizados, de forma sequencial e temporal. Descrevendo o processo de aquisição, seleção, rotulação e organização das imagens utilizadas para treino e teste dos modelos de ML e Detecção de Objetos. Também é informado quais as métricas de avaliação desses modelos. E por fim, é descrito o processo para o desenvolvimento do aplicativo móvel e da plataforma web.

3.1 Base de dados

A base de dados é uma parte essencial para o treinamento e validação de modelos de ML e VC. Sendo esta utilizada pelos algoritmos para extrair informações essenciais para o aprendizado da máquina e validação do que foi aprendido, assim como atualizações futuras. Porém, uma base de dados pode ser utilizada de diversas formas. Sendo possível sua utilização para analisar e compreender o problema estudado. Então, a montagem e organização de um banco de dados deve ser vista como uma ponte para um excelente desempenho das tecnologias empregadas na solução da problemática. A Figura 3 apresenta um breve fluxograma que representa como foi feita a coleta das imagens em campo, sua rotulação e disposição no banco de imagens.

Figura 3 – Fluxograma de coleta de imagens.



Fonte: Elaborado pelo autor.

3.1.1 Coletas em campo

A coleta de imagens de mucosas oculares de ovinos ocorreu em parceria com a EMBRAPA, no período de 2019 a 2025. Visitando diversas propriedades de pequenos e médios agricultores, assentamentos e fazendas experimentais em regiões do estado do Ceará, em especial a região norte. A Figura 4 mostra uma dessas visitas para a coleta de imagens.

Figura 4 – Coletas de imagens em campo.

Fonte: Elaborado pelo autor.

À aquisição destas imagens é primordial para o treinamento dos modelos de ML e Detecção de Objetos, visto que esses algoritmos necessitam de dados para que possam aprender e consigam ser eficientes nas tarefas de classificação e reconhecimento.

Para se obter as imagens foram utilizados dispositivos móveis, celulares, com o aplicativo *Stop Verme* instalado em sua versão ainda inicial que coletava imagens e as analisava,

armazenando as imagens de forma local. O aplicativo realizava captura das imagens e os salvava de forma local. Para garantir que a imagem era realmente salva, uma outra fotografia era feita, agora utilizando a câmera do próprio dispositivo.

Além da aquisição das imagens também era feito a coleta de sangue do animal para comprovação do seu estado de saúde, com ou sem anemia, em laboratório e aplicação de um questionário sobre informações da propriedade e qual vermifugo era utilizado. Essas informações obtidas por meio de questionário ajudava a equipe de coleta a entender qual princípio ativo dos vermífugos estava sendo utilizado na região e se havia o uso correto deste por meio dos criadores. Já que em muitos casos o mau uso desses produtos poderia ocasionar o aumento da resistência do parasita.

Para realizar a coleta das imagens também era necessário a seleção de animais, um fator que dificultou à aquisição de muitas amostras. Muitos dos locais visitados não apresentavam número suficiente de animais doentes, ou seja, com anemia em específico. Em muitos casos os animais haviam sido vermifugados de forma anterior a visita ou estes estavam em locais em que ter animais doentes iam muito além do prejuízo financeiro, como no caso das fazendas experimentais, que em caso de morte de algum animal é necessário todo um processo de licitação para aquisição de um novo. Devido a essa situação o banco de imagens ficou desbalanceado. E para sanar essa dificuldade, foi necessário avaliar novos locais e agendar visitas, o que ocasionou uma demora no período de coletas.

3.1.2 Rotulação das imagens

Para a utilização das imagens por algoritmos de ML é necessário informar aos modelos quais as características e regiões de interesse (ROI) serão utilizadas como entrada para o treinamento destes. Esse processo é muitas vezes feito de forma manual, rotulando as regiões e imagens. Para isso, muitas ferramentas foram desenvolvidas visando facilitar o processo de rotulação. Uma delas é *Roboflow* (ROBOFLOW, 2025) que fornece ao desenvolvedor diversas funções para recortar, selecionar e classificar as imagens desejadas. Tornando o processo mais fácil e ágil, tanto do ponto vista prático como na garantia da assertividade da tarefa.

O *Roboflow* foi utilizado para a rápida rotulação das imagens, tanto para os modelos de classificação e de detecção de objetos. Além das ferramentas disponíveis para essas tarefas ainda é possível exportar o *dataset* produzido e rotulado em diversos formatos que são aceitos em algoritmos clássicos da literatura. Como exemplo, temos as versões do *Yolo* e algoritmos

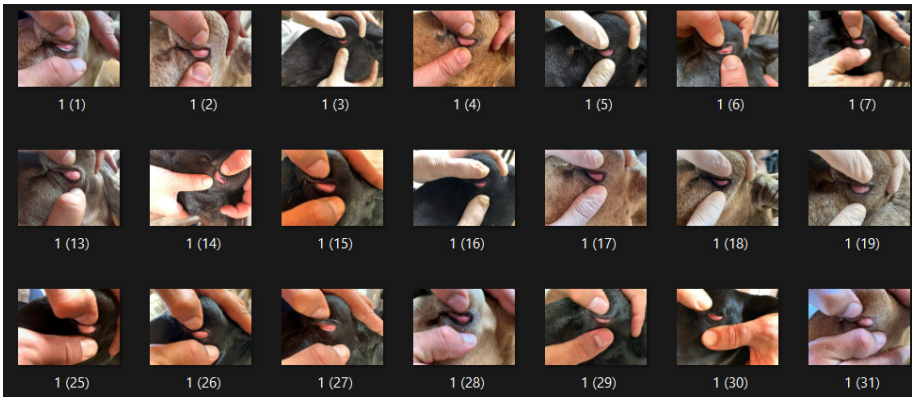
produzidos com as bibliotecas de ML, *PyTorch* e *TensorFlow*.

3.1.3 Banco de imagens

Após a coleta de imagens em campo e a rotulação destas, foi possível a montagem do banco de imagens. Para adaptar as imagens aos modelos de classificação e detecção de objetos foram criados diferentes conjuntos com as mesmas divisões de imagens. Porém, com formatos diferentes para que as arquiteturas analisadas podessem utilizar-los de forma correta.

Ao fim de todo o processo o *dataset* formado possui 165 imagens divididos em subconjuntos para treino e teste. A Figura 5 apresenta algumas amostras do *dataset* final. Devido ao número desbalanceado de classes, foi feita a remoção de algumas imagens de animais saudáveis para realizar o balanceamento do banco de imagens. Sendo ao final, divididas em saudáveis e não saudáveis. Incluindo 99 imagens saudáveis e 66 não saudáveis.

Figura 5 – *Dataset* com imagens de mucosas oculares.



Fonte: Elaborado pelo autor.

3.2 Modelos de Machine Learning e Visão Computacional

Neste trabalho foram utilizados dois algoritmos para classificação e detecção de objetos. Seguindo uma *pipeline* que junta as duas metodologias para a obtenção do resultado final, empregando o uso em conjunto da Visão Computacional com *Machine Learning*.

3.2.1 Modelos de detecção de objetos

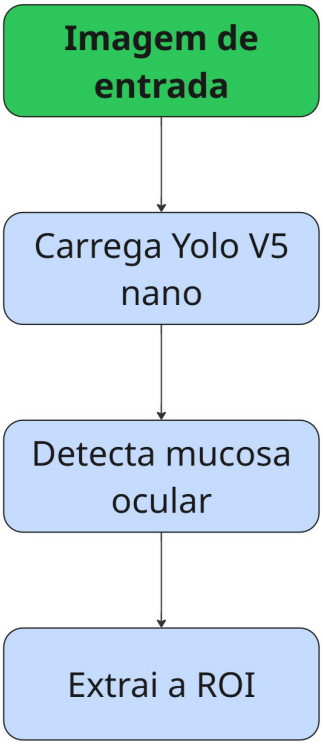
A fim de se obter uma determinada região de interesse da imagem coletada em campo com o uso do aplicativo, foi necessário a utilização de um modelo de detecção de objetos. Este modelo é capaz de analisar a imagem e retornar uma ROI, ou seja, apenas uma parte da

imagem onde o objeto de interesse se encontra. Esta etapa é feita para se selecionar a mucosa ocular do animal para ser inserida no algoritmo de classificação.

O modelo analisado foi o *Yolo V5 nano*, devido sua consolidação no mercado e por sua versão nano que é capaz de ser executada em dispositivos móveis. Ao todo 165 imagens foram utilizadas para o treinamento deste modelo, anotadas e rotuladas de forma manual como descrito na seção 3.1.2. Depois desta etapa, foram organizadas e divididas em pastas para treino, teste e validação.

A Figura 6 apresenta de forma visual o treinamento empregado ao detector de objetos e a representação da entrada e saída do modelo.

Figura 6 – Fluxograma de detecção de objetos.



Fonte: Elaborado pelo autor.

3.2.2 Modelos de classificação

A fim de aplicar o método FAMACHA em uma aplicação tanto web como *mobile* e automatizar o processo de identificação de animais anêmicos, decidiu-se por analisar o uso de algoritmos de ML para esta tarefa. No seguinte trabalho, foi utilizado modelos de classificação baseados em aprendizado supervisionado e treinados com uma base de dados própria. Desenvolvida e montada na própria região norte do estado do Ceará.

Os algoritmos implementados e testados foram a *EfficientNet-B0*, *MobileNet*, *NASNet Mobile* e um modelo de construído para este projeto. Sendo três estruturas já consolidadas pela comunidade de desenvolvedores e com bastante resultados positivos, além da arquitetura que foi criada especificamente para este problema.

Essa arquitetura proposta foi desenvolvida de forma empírica, selecionando e ajustando os parâmetros do modelo baseando-se em observações nos resultados obtidos pelo modelo e refinando com base em arquiteturas já existentes. Sendo criada e testada por (SILVA, 2025). A Tabela 1 apresenta as características do modelo proposto.

Tabela 1 – Configuração da Arquitetura Proposta

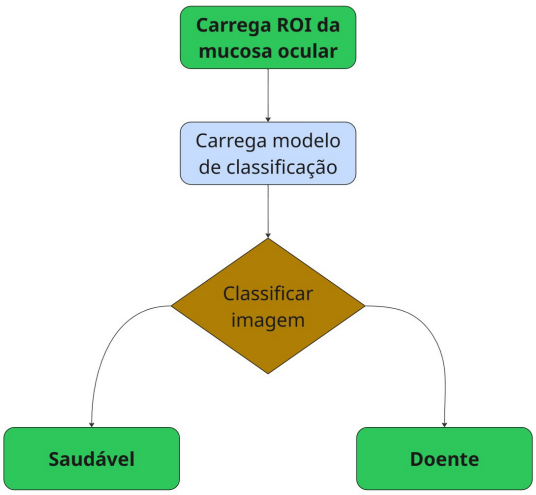
Camada	Configuração
Conv2D	32 filtros, (3x3), ReLU, L2 (= 0,01)
BatchNormalization	-
MaxPooling2D	(2x2)
Conv2D	64 filtros, (3x3), ReLU, L2 (= 0,01)
BatchNormalization	-
MaxPooling2D	(2x2)
Conv2D	128 filtros, (3x3), ReLU, L2 (= 0,01)
BatchNormalization	-
MaxPooling2D	(2x2)
Flatten	-
Dense	128 neurônios, ReLU, L2 (= 0,01)
Dropout	0,5
Dense (saída)	5 neurônios, softmax

Fonte: Adaptado de (SILVA, 2025).

Para o funcionamento desses algoritmos, as imagens coletadas de forma inicial passam por algoritmo de detecção de objetos e são recortadas, extraindo a ROI e sendo utilizadas como entrada nos modelos de classificação. Com as três redes implementadas, foi avaliado os valores de acurácia, *recall*, *F1-Score* e precisão. A fim de se obter o modelo que melhor performava sobre o conjunto de imagens obtida.

A Figura 7 apresenta a metodologia empregada para aplicação e utilização dos modelos de classificação.

Figura 7 – Fluxograma do algoritmo de classificação.



Fonte: Elaborado pelo autor.

Após a identificação de qual algoritmo se adaptava melhor ao problema, foi realizada a conversão dos pesos das redes para funcionamento em dispositivos móveis e web. Possibilitando assim, que usuários e técnicos da EMBRAPA utilizassem esses recursos sem burocracia e complicações.

3.2.3 Métricas de avaliação

Para avaliar os modelos desenvolvidos e sua aplicabilidade na plataforma web foi pensado em métricas que demonstrassem o seu desempenho em ambientes fora o de treinamento e teste clássicos. Para isto foi avaliado o tempo de processamento das imagens, consumo de memória e tempo de retorno ao usuário da plataforma. Essas informações ajudam a entender como os algoritmos se comportam em uma aplicação real e ao lado do usuário final. A Tabela 2 apresenta detalhes sobre as variáveis analisadas para compreender o desempenho dos modelos.

Tabela 2 – Métricas para avaliação dos modelos

Variáveis analisadas	Unidade
Tempo de processamento	s
Consumo de memória	Mb
Tempo de retorno	ms
Tempo de inferência	ms

Fonte: Elaborado pelo autor.

3.3 Plataforma de gerenciamento

A plataforma de gerenciamento web administra dados obtidos através do aplicativo móvel, que são imagens, geolocalização, dados de produtor e sobre os vermífugos utilizados. Nesse contexto, é apresentado o desenvolvimento do aplicativo e da plataforma de forma individual, já que empregam diferentes tecnologias em seu desenvolvimento e depois se ligam para análise dos dados.

3.3.1 Aplicativo móvel

O aplicativo móvel denominado *Stop Verme* foi desenvolvido utilizando a linguagem *Kotlin*, com essa linguagem é possível acessar requisitos nativos do sistema operacional *Android*. Esses requisitos incluem câmera, GPS e galeria, possibilitando adquirir imagens, posição de localização e salvar arquivos no dispositivo.

Para facilitar a utilização do aplicativo pelo homem do campo, primeiro foi pensado em como ele deveria funcionar e quais telas seriam necessárias. Buscando simplicidade e eficiência, foi proposto que app tivesse duas telas. A primeira tela com o recurso da câmera e a segunda com a apresentação do resultado do modelo de classificação.

3.3.2 Plataforma web

A plataforma web para gerenciamento de dados do aplicativo *Stop Verme* foi pensada com recursos clássicos de sistemas de administração, como *login*, cadastro e *dashboard* para obter segurança no acesso dos dados e ferramentas de forma visual para melhor análise das informações coletadas em campo.

Para viabilizar o desenvolvimento desse projeto, inicialmente foi feita uma coleta e análise de requisitos do sistema. Buscando entender como os técnicos e demais usuários do sistema precisavam acessar os dados obtidos com o aplicativo e como eles gostariam de gerar

resultados a partir deles.

O primeiro ponto analisado é que ela precisa ser acessível e possui interface amigável. Para sanar essa problemática foi pensado em desenvolver o *front end* do projeto utilizando o *framework React JS* para possibilitar tanto um desenvolvimento rápido devido sua facilidade e vasta documentação, como possibilitar que as telas fossem responsivas. Com a responsividade é possível acessar a plataforma de qualquer dispositivo sem perder qualidade da versão para Desktop e por fim, criar botões, formulários e apresentar o conteúdo de forma clara e objetiva para o usuário.

O segundo ponto analisado é a necessidade de segurança, percebendo a necessidade de funcionalidades de autenticação de usuários. Para isso foi pensado em incluir telas de *login* e cadastro de usuário no projeto. Verificando se o usuário possuía ligação com a EMBRAPA através do domínio de e-mail. Essas pessoas são cadastrados em um banco de dados e possuem o mesmo nível de acesso do sistema.

O terceiro ponto é como os dados poderiam ser analisados de forma visual e simplificada. Para isso foi pensando em plotar um mapa com pontos que representam as posições das fazendas visitadas. E que ao clicar, podessem gerar relatórios com dados cadastrados da propriedade. Fazendo com que fosse gerado um relatório em PDF para *download*.

4 RESULTADOS

Nesta seção será apresentado os resultados obtidos nos modelos de detecção de objetos, de classificação, do aplicativo e da plataforma de gerenciamento web. Ferramentas empregadas no projeto que auxiliam na identificação de anemia em ovinos, através da automatização do cartão FAMACHA. Possibilitando que criadores e técnicos identifiquem a presença do HC em propriedades e possam agir de forma rápida e eficiente no controle dos parasitas. E gerando informações através de coletas de imagens e dados diversos, como localização, informações da propriedades e remédios utilizados no controle do HC, criando uma base de dados robusta que possibilita a geração de relatórios e o entendimento da propagação da anemia e uso de vermífugos.

4.1 Resultados dos modelos de detecção de objetos

O modelo de detecção utilizado nesse projeto foi o *Yolo V5 nano*, uma versão mais compacta do robusto modelo *Yolo V5*. Muito utilizado em aplicações de detecção de objetos, rastreamento e segmentação, empregado para os mais diversos fins.

O modelo foi treinado utilizando 165 imagens dividas em cinco classes, assim como no cartão FAMACHA. Que tiveram a ROI recortada e rotulada com auxílio do *Roboflow*. Gerando uma nova base com 165 imagens, agora com apenas a parte da mucosa ocular dos animais. Necessárias para o treinamento e utilização dos modelos de classificação. Além disso, as amostras passaram a ter tamanho padrão de 320 x 320 *pixels*.

Durante o treinamento do modelo, foi realizado a implementação de técnicas de aumento de dados para expandir o *dataset* e fornecer mais amostras no treinamento. O período para o treino foi de 100 épocas e utilizado um *batch size* de 32.

A Tabela 3 apresenta os resultados obtidos pelo *Yolo V5 nano*, utilizando como métricas de avaliação a precisão, *recall*, *mAP0.5* e *mAP0.5:0.95*.

Tabela 3 – Resultados do detector de objetos *Yolo V5 nano*.

Modelo	Precisão	Recall	mAP0.5	mAP0.5:0.95
Yolo V5 nano	99.6%	100%	99.5%	59.8%

Fonte: Elaborado pelo autor.

Como pode-se observar, esta versão do *Yolo* performou bem sobre o conjunto de dados, com bastante precisão, chegando a quase 100%. E já o seu o *recall* obteve os 100%,

mostrando mais a sua eficiência. O *mAP0.5* e o *mAP0.5:0.95* também obtiveram resultados positivos e validam a utilização desta versão do *Yolo* para a detecção de mucosa ocular de ovinos.

4.2 Resultados dos modelos de classificação

Para os algoritmos de classificação foi utilizado a ROI extraída pelo *Yolo V5 nano*, focando apenas na mucosa ocular do ovino, como imagem de entrada. Estas agora com tamanho padrão de 64 x 64 *pixels*, a Figura 8 representa um exemplo da imagem de entrada dos modelos de classificação. As quatro redes analisadas foram submetidas as mesmas condições de treino e teste, sendo treinados com valor empírico de 700 épocas, *batch size* de 16 e implementação de validação cruzada com k = 5.

Figura 8 – Imagem da mucosa ocular com dimensões de 64 x 64 *pixels*.

Fonte: Elaborado pelo autor.

Os algoritmos foram treinados utilizando 165 imagens divididas em cinco classes, assim como no cartão FAMACHA. Sendo 33 imagens para cada classe, sendo 1 e 2 como saudável, 3 já indicando a possibilidade de vermifugação e as classes 4 e 5 já indicam que o animal está doente.

As redes analisadas foram *EfficientNet-B0*, *Mobile Net*, *NASNetMobile* e a rede proposta sem uso de transferência de aprendizado. Como métricas de avaliação foram utilizados à acurácia, precisão, *recall* e *F1-Score*.

A Tabela 4 apresenta as pontuações obtidas pelos classificadores no conjunto de dados montado para a tarefa. Como pode-se perceber a arquitetura proposta apresentou um melhor desempenho em relação as demais arquiteturas, com valores de todas as métricas sendo superiores a 90%. Juntamente com a *EfficientNet-B0* que obteve resultados acima de 80% para todas as métricas.

Tabela 4 – Resultados dos classificadores.

Classificador	Acurácia	Desvio Padrão Acurácia	Precisão	Recall	F1-Score
Arquitetura proposta	90%	±0.0521	92%	91%	90%
EfficientNet-B0	83%	±0.1159	88%	83%	83%
NASNetMobile	39%	±0.0691	32%	39%	30%
MobileNet	60%	±0.1767	61%	60%	56%

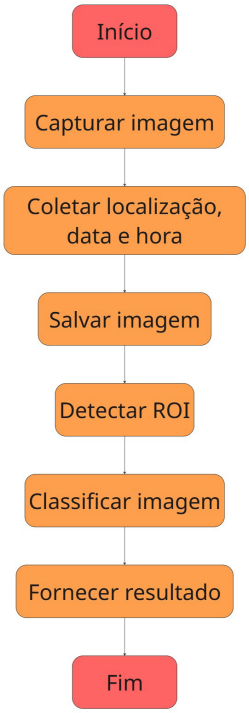
Fonte: Elaborado pelo autor.

4.3 Resultados do aplicativo móvel

Afim de facilitar e ajudar produtores e técnicos a identificarem o ataque do HC em ovinos, foi desenvolvido um aplicativo móvel para fornecer aos usuários uma ferramenta que analisa a imagem da mucosa ocular e fornece um diagnóstico de animal anêmico ou saudável. Possibilitando maior eficácia no processo de identificação, segurança no resultado por parte de interpretação, agilidade e rapidez.

Para isso, os modelos de detecção de objetos e classificação são inseridos no app e trabalham em conjunto para fornecer o diagnóstico ao produtor, rodando embarcados no dispositivo móvel. A Figura 9 apresenta o funcionamento básico do aplicativo. Mostrando a coleta da imagem por parte do usuário, seguido da detecção da mucosa ocular do animal, em seguida é feita a classificação e por fim, o resultado final é apresentado.

Figura 9 – Fluxograma com funcionamento do aplicativo.



Fonte: Elaborado pelo autor.

Após a implementação dos algoritmos de detecção e classificação, o aplicativo em sua versão final conta com duas telas. A primeira apresenta a funcionalidade da câmera para a captura de imagens e a segunda informa ao usuário o resultado final. No *Backend* da aplicação ocorre a captura da localização, data e hora em que as fotos foram coletadas. As Figuras 10, 11 e 12 apresentam as telas do aplicativo *Stop Verme*.

Figura 10 – Tela inicial do aplicativo *Stop Verme*.

Fonte: Elaborado pelo autor.

A Figura 10 apresenta a tela inicial do aplicativo *Stop Verme*, nela é possível observar

na barra superior do app a logo e um botão de configurações para cadastro da propriedade e o vermífugo utilizado. Na parte central, a câmera é utilizada para a captura de imagens em campo e abaixo temos o botão de galeria para acessar imagens salvas, o botão de captura da foto e um botão de dúvidas.

Figura 11 – Tela de cadastro de propriedade.

Fonte: Elaborado pelo autor.

A Figura 11 apresenta a tela de cadastro da propriedade avaliada, coletando informações como nome da fazenda, sigla, o vermífugo utilizado e qual seu princípio ativo.

Figura 12 – Tela de resultado.

Fonte: Elaborado pelo autor.

Na Figura 12 é apresentado o resultado obtido através da imagem capturada ou lida da galeria do dispositivo. A imagem de entrada passa pelo detector de mucosas e depois pelo classificador. Ao final do processo é informado ao usuário se o animal está anêmico ou não. E por fim, na parte inferior da tela é exibido um botão com a finalidade de salvar o resultado obtido. Que futuramente serão compartilhados com a plataforma web.

4.4 Resultados do sistema de gerenciamento web

Seguindo o fluxo de trabalho, visando analisar os dados coletados pelo aplicativo e analisar o desempenho dos modelos de detecção e classificação desenvolvidos foi criada uma plataforma web para gerenciamento desses dados. Facilitando a geração de relatórios e o entendimento do uso de vermífugos. Além de ajudar a compreender como a anemia tem se espalhado pela região.

O sistema leva ao usuário uma aplicação de controle seletivo de anemia em ovinos, desenvolvido utilizando o *framework React JS* para o *Front End* e a linguagem *Python* para *Back End* com auxílio do *framework Flask*. O banco utilizado para armazenar dados de usuários e informações coletadas pelo app como imagens e dados da propriedade foi *Postgree SQL*.

A Figura 13 apresenta a tela inicial do sistema de gerenciamento, nessa página é possível encontrar informações sobre o projeto, entendo sobre ele e quais suas aplicações. Tendo como foco uma breve apresentação sobre o aplicativo, equipe e instituições que trabalharam em conjunto no desenvolvimento do projeto. Ainda nesta tela é possível acessar a página de *Login*, através de um botão no cabeçalho.

Figura 13 – Tela *Home* do sistema de gerenciamento.

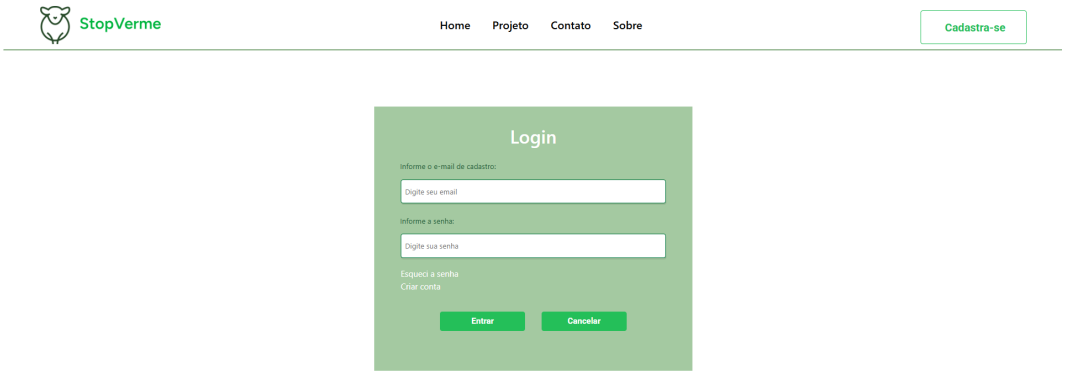
Fonte: Elaborado pelo autor.

Na Figura 14 mostra a tela de *Login* da aplicação, nela é possível realizar *login* no sistema e acessar a tela de *dashboard*. Para isso é necessário informar o *e-mail* e senha cadastrados. O *e-mail* deve ser do domínio da EMBRAPA, devido a plataforma está disponível inicialmente apenas para técnicos da instituição. Logo abaixo dos campos de *e-mail* e senha, existem dois links que levam a funcionalidades de recuperação de senha e cadastro de usuário.

Ao clicar no botão de “Entrar”, é realizado a verificação se o usuário está cadastrado.

Caso positivo, é liberado o acesso ao sistema. Gerando um *token* de autenticação para isso, garantindo segurança nas demais rotas.

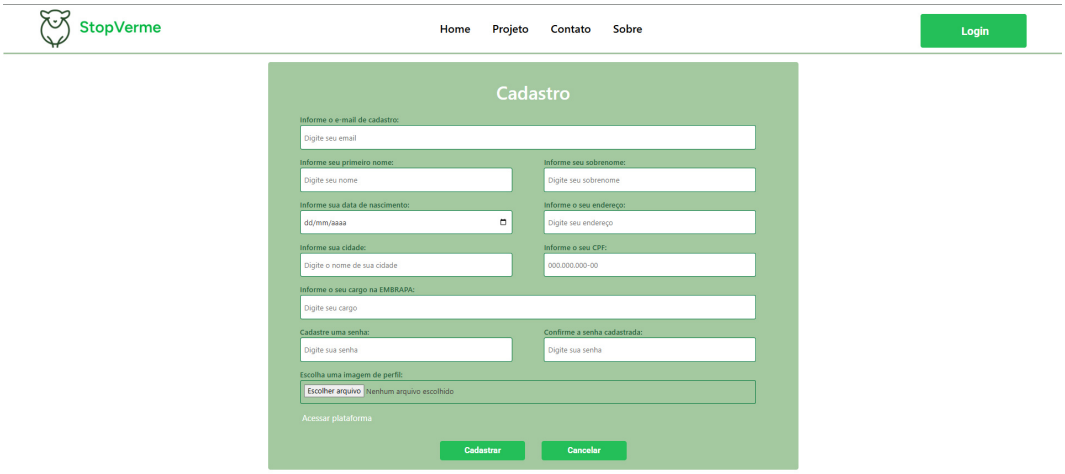
Figura 14 – Tela de *Login*.



Fonte: Elaborado pelo autor.

Para realizar o cadastro de novos usuários, foi pensado em coletar informações relevantes para identificação e comprovação de ligação com a EMBRAPA. Após uma análise de requisitos, chegou-se a conclusão que ao utilizar o *e-mail* com domínio da própria instituição e o cargo já seria o suficiente para esta comprovação. Outras informações relevantes são, o nome, sobrenome, data de nascimento, enderenço, Cidade, CPF, foto de perfil e a senha para segurança da plataforma. A Figura 15 apresenta o resultado final desta tela.

Figura 15 – Tela de cadastro do sistema de gerenciamento.



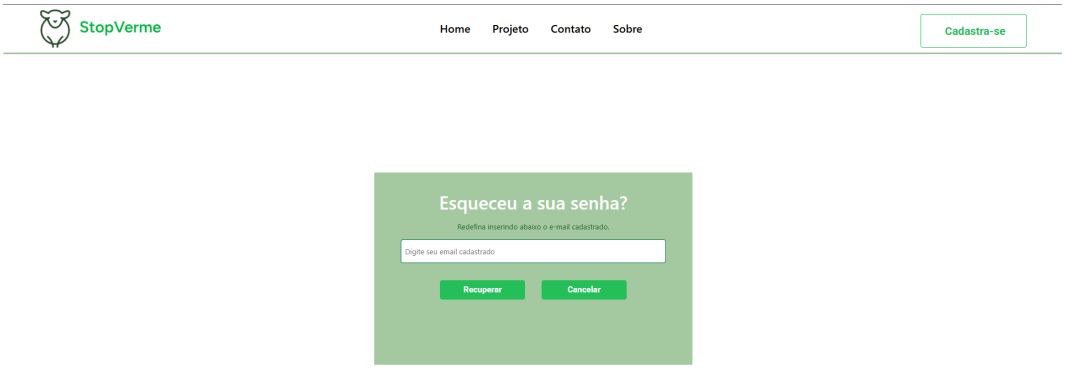
Fonte: Elaborado pelo autor.

Quando o usuário realiza o cadastro, é acionado o *Backend* para isenção dos dados no banco *Postgree SQL* por meio da API. Ao ser cadastrado, também é gerado um *token* para

autenticação de usuário e possibilitar o acesso rotas protegidas, como *dashboard* e demais funcionalidades.

Em muitas situações é possível que os usuários esqueçam a senha e pedam o acesso ao sistema. Pensando nessa possibilidade, foi desenvolvido uma funcionalidade para recuperação de senha. A Figura 16 apresenta a tela para recuperação de senha. Onde o usuário pode informar o *e-mail* cadastrado e solicitar a recuperação.

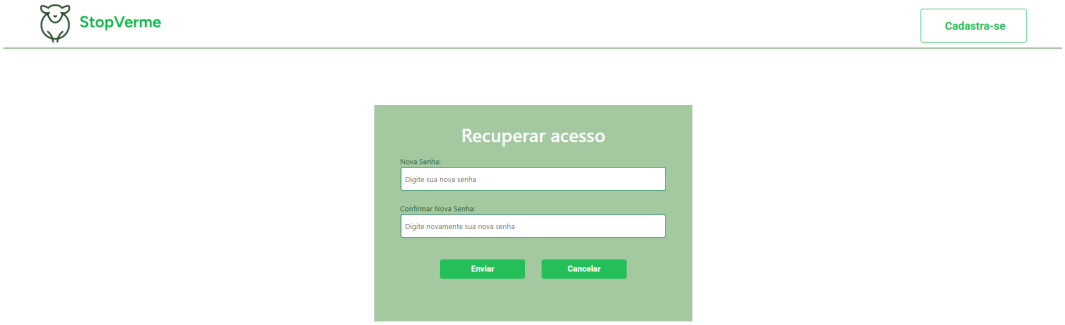
Figura 16 – Tela de recuperação de senha.



Fonte: Elaborado pelo autor.

Após solicitar esta recuperação é enviado por *e-mail* um link para atualização da senha. Este link chega no *e-mail* informado na tela de recuperação e pode ser acessado apenas clicando nele. A Figura 17 apresenta a tela de redefinição de senha, onde é possível cadastrar uma nova.

Figura 17 – Tela de redefinição de senha.

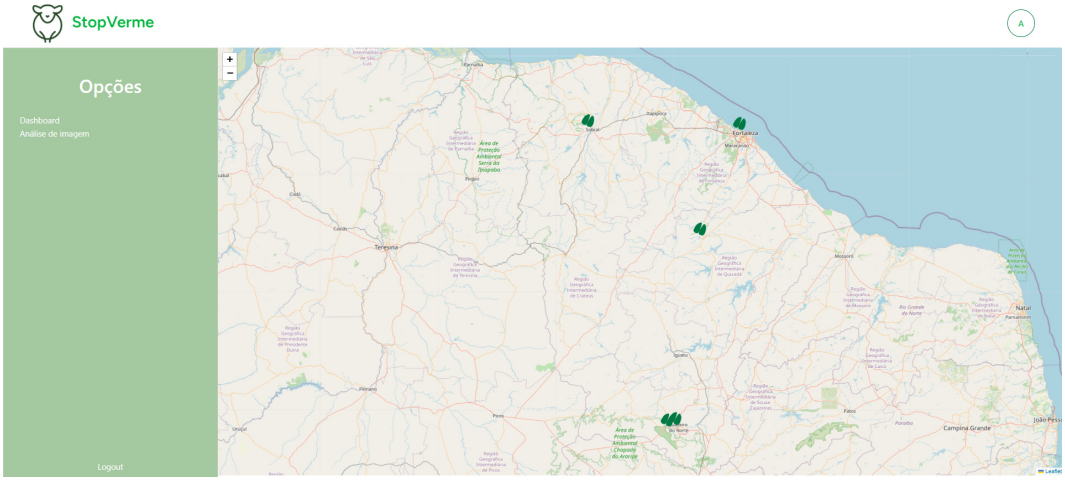


Fonte: Elaborado pelo autor.

Após realizar o cadastro ou *login* no sistema, o usuário será levado ao *dashboard* da aplicação. Nesta tela é apresentado um mapa com foco no estado do Ceará, esse foco se da devido que o projeto ainda é aplicado apenas no estado. Além disso, existe um menu lateral no lado esquerdo com links para navegação entre as telas de análise de imagem e o próprio *dashboard*. Já na parte inferior do lado esquerdo também tem a opção de *logout*, sendo possível sair da aplicação e retornar casa tela inicial. No lado direito, no cabeçalho tem-se o botão para a tela de perfil. A Figura 18 mostra o resultado final dessa tela.

No mapa existem pontos que são identificados com ícones de patas de ovelhas que representam os locais das propriedades visitadas para coletas de imagens com o aplicativo *Stop Verme*. Os pontos apresentados na Figura 18 ainda são exemplos fictícios e apresentam locais em cidades aleatórias.

Figura 18 – Tela de *dashboard*.

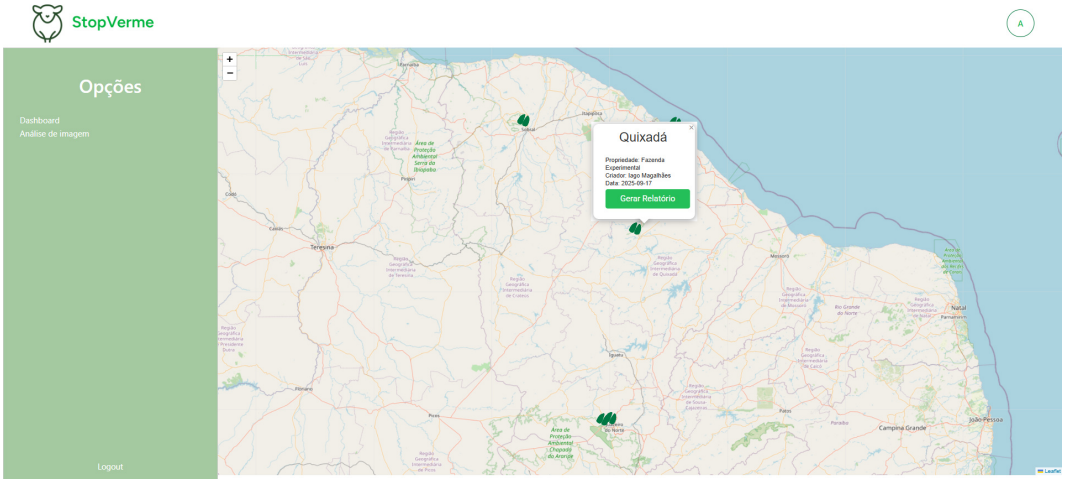


Fonte: Elaborado pelo autor.

A partir da tela de *dashboard* e os demais links presentes possuem a necessidade do *token* de autenticação para navegação. Por padrão, a duração em que um usuário pode ficar logado é de 6 horas. Após esse período uma mensagem é exibida na tela e o usuário é redirecionado a tela de *login* de forma automática.

Ao clicar em um dos ícones de patas de ovelhas, um *popup* é aberto apresentando um breve resumo sobre a propriedade. Informações como a cidade, nome da fazenda, proprietário e data da coleta são exibidas ao usuário. Neste *popup* ainda possuí um botão para gerar relatório, que fornece a opção de gerar um arquivo PDF com informações da propriedade e sobre os dados coletados. O *popup* é representado na Figura 19 apresenta melhor como é possível gerar relatórios específicos sobre as fazendas visitadas e atendidas.

Figura 19 – Tela com *popup* no *dashboard*.

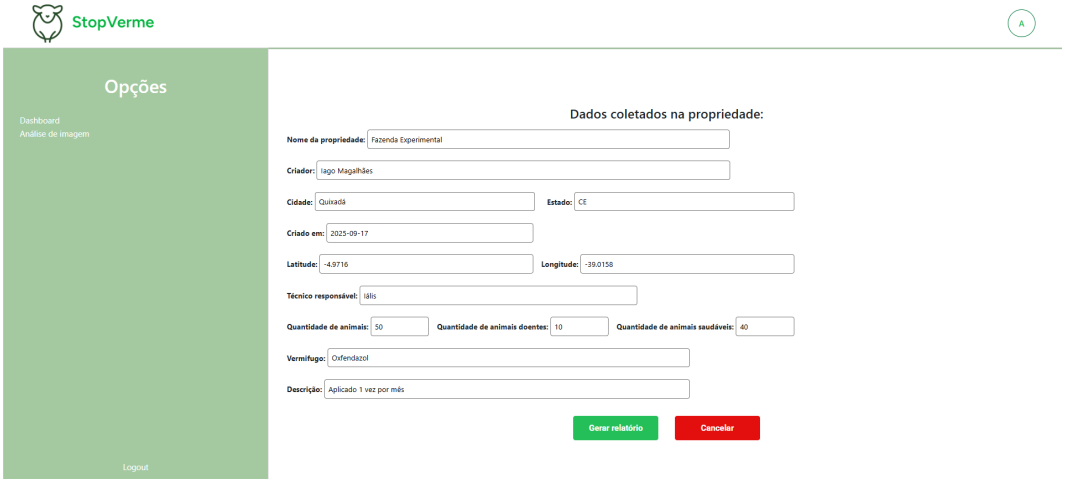


Fonte: Elaborado pelo autor.

Selecionando a opção de gerar relatório o usuário será redirecionado a uma nova tela. A Figura 20 apresenta a tela de relatório, onde é possível analisar as informações coletadas na propriedade com mais detalhes. Esses dados são obtidos por meio de entrevista com os produtores, ainda preenchendo formulários de papel e posteriormente inseridos no sistema.

Ao optar por gerar relatório clicando no botão verde na parte inferior da tela é gerado um PDF com as informações da propriedade visualizados anteriormente. Ainda existe a opção de cancelar que leva a tela de *dashboard* e possibilita a seleção de outras fazendas.

Figura 20 – Tela de relatório.



Fonte: Elaborado pelo autor.

Os dados utilizados para gerar os relatórios podem ser visualizados na Tabela 5, essas informações são importantes para compreender sobre a localização da propriedade, número de animais, entender como está sendo feito o controle do verme HC, quais produtos são utilizados e como está sendo feita sua aplicação. Possibilitando entender como o vermífugo se comporta e se

há ou não perda do poder de eficiência do princípio ativo.

Tabela 5 – Dados coletados nas propriedades.

Dados coletados	Tipo de dado
Nome da propriedade	Texto
Criador	Texto
Cidade	Texto
Estado	Texto
Data de coleta	Data
Latitude	Numérico
Longitude	Numérico
Técnico responsável	Texto
Quantidade de animais	Numérico
Animais doentes	Numérico
Animais saudáveis	Numérico
Vermífugo	Texto
Descrições adicionais	Texto

Fonte: Elaborado pelo autor.

Além da funcionalidade de gerar relatório, também é possível realizar à análise de imagem com os mesmos modelos empregados no app. Tanto o detector de objetos e o classificador estão disponíveis na versão web para uma análise mais minuciosa e detalhada. Possibilitando que o usuário selecione imagens específicas, realize a detecção da mucosa ocular do animal e por fim, classifique-a.

O modelo do *Yolo V5 nano* para detecção da mucosa pode funcionar de forma independe do classificador, sendo possível extrair apenas a mucosa do animal e salvar uma imagem com a ROI. Já o classificador depende do detector para que apenas a ROI seja inserida como entrada no modelo de classificação, assim como no app.

A Figura 21 representa a tela de análise de imagem do sistema, nesta tela o menu lateral à esquerda permanece como na tela de *dashboard* possibilitando a navegação entre elas. Na parte central e à direita é apresentado as funcionalidades para utilização dos modelos de detecção de mucosas, classificação e salvar a imagem resultante por meio de botões. Logo abaixo dos botões, existe uma campo para carregar à imagem desejada e ao lado uma área reservada para apresentação dos resultados.

O objetivo dessas funcionalidades é trazer aos usuários a experiência de compreender melhor o funcionamento da VC e ML por trás do aplicativo e na extração dos dados. Mostrando como um algoritmo de detecção recorta apenas à área da mucosa ocular de forma automática sem auxílio de seres humanos e como o algoritmo de classificação entrega um resultado preciso

de se o animal está anêmico ou não através da região recortada anteriormente pelo detector. Tudo isso sem a necessidade prévia de um exame de hematócrito ou do cartão FAMACHA.

Figura 21 – Tela de análise de imagem.

Fonte: Elaborado pelo autor.

Já à Figura 22 representa o uso desta funcionalidade. Uma imagem de ovino é carregada e feita à classificação, sendo possível visualizar a forma correta de exposição da mucosa ocular. Ao carregar uma imagem clicando no botão "Faça *upload*", o arquivo selecionado passa a ser exibido na tela e um novo botão aparece, permitindo remover essa imagem ao clicar em "Remover".

Ao realizar à classificação os dados da detecção são exibidos, como o resultado, apresentando se o animal está saudável ou não. Além do tempo necessário para realizar a classificação e o consumo de memória.

Figura 22 – Tela de análise de imagem com resultados.

Fonte: Elaborado pelo autor.

A Tabela 6 apresenta resultados com a utilização do detector e classificador em uma aplicação web, analisando informações indicadas na Tabela 2. O recurso de hardware utilizado foi um notebook com 8Gb de memória RAM, SSD com 254Gb e processador *Ryzen 5*. Em média, à aplicação se mostrou muito promissora tanto no consumo de memória como no tempo de processamento. Devido aos testes ainda se darem de forma local, o tempo de retorno não foi avaliado com total precisão devido que no funcionamento em servidor real haveria tempo de latência e não foi possível simular em ambientes de teste. Já o tempo de inferência permaneceu similar ao tempo do aplicativo na casa do ms. Os valores apresentados são médias obtidas em testes locais.

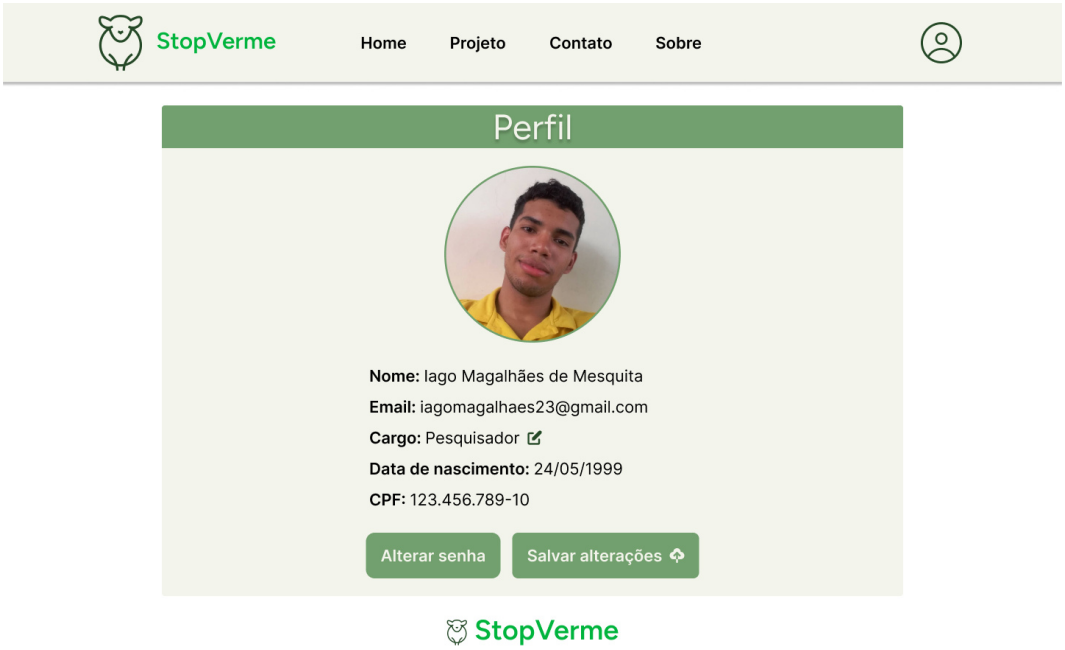
Tabela 6 – Valores das métricas de avaliação dos modelos.

Variável analisada	Valor
Tempo de processamento	2.1s
Consumo de memória	0.1Mb
Tempo de retorno	–
Tempo de inferência	103ms

Fonte: Elaborado pelo autor.

A tela de perfil é representada na Figura 23 e trás informações que foram informadas pelo usuário no ato do cadastro. Nela é possível alterar senha, foto de perfil e cargo. Os demais dados não podem ser modificados apenas visualizados.

Figura 23 – Tela de relatório.



Fonte: Elaborado pelo autor.

Por fim, a última tela presente no sistema pode ser visualizada na Figura 24 que representa uma mensagem de aviso em caso de tentativa de acesso de rotas não existentes ou não encontradas.

Figura 24 – Tela de página não encontrada.

Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Neste trabalho é apresentado o desenvolvimento de um sistema web de controle seletivo de anemia em ovinos, que irá analisar dados vindos do aplicativo *Stop Verme* e imagens que podem ser carregadas de forma local para à aplicação. O projeto une um sistema web, um aplicativo móvel e o uso de detector de objetos e classificadores. Integrando sistemas web e *mobile* com ferramentas de Visão Computacional e *Machine Learning*. Apresentando uma metodologia robusta para sanar o objetivo principal, que é a identificação de anemia em ovinos simulando a metodologia empregada no cartão FAMACHA através da análise da cor da mucosa ocular do animal.

Para isso foi seguido uma metodologia de que, primeiro se deveria montar uma base de dados com imagens de mucosas oculares obtidas em campo para treinamento do modelo de detecção e classificação. Sendo um ponto importante por ser um *dataset* com informações únicas e de conteúdo pouco disponível para pesquisadores. Apresentando uma contribuição importante para a comunidade e possibilitando novos estudos no segmento. O segundo passo foi treinar um detector de objetos para identificar a mucosa ocular do ovino. Para isso foi utilizado o *Yolo V5 nano* devido sua compartibilidade com dispositivos de baixo processamento, apresentando uma precisão de 99.6% e *mAP0.5* de 99.5%. Após a seleção e treinamento do modelo de detecção foi analisado algoritmos de classificação baseados em CNNs. Dentre os testes realizados, à arquitetura proposta nessa pesquisa performou melhor sobre o conjunto de imagens e atigiu resultados de acurácia de 90%, precisão de 92%, *Recall* de 91% e *F1-Score* de 90%.

Depois de selecionados e treinados os algoritmos que seriam utilizados para detecção e classificação, foi desenvolvido um aplicativo móvel para facilitar a utilização desses modelos por usuários em campo. Facilitando sua utilização e levando agilidade e precisão na identificação de animais com anemia causada pelo verme HC.

Por fim, afim de analisar os dados obtidos pelo aplicativo e entender melhor o funcionamento dos modelos foi desenvolvido um sistema web que possibilita não só testar os modelos desenvolvidos, mais administrar os dados obtidos nas coletas em campo. Como informações sobre a propriedade, número de animais, se há presença ou não de ovinos com anemia e sobre o uso de vermífugos no local.

Como contribuição, tanto o aplicativo e a plataforma trazem agilidade no processo de detecção de animais anêmicos. Evitando erros de interpretação e utilização, como o de métodos já conhecidos na comunidade. Além de fornecer além do diagnóstico uma análise posterior,

ajudando a entender onde esses animais doentes estão presentes, tendo um maior controle de zoonoses e compreendendo sua propagação na região. Outro ponto é plotar os dados em um mapa e ver onde os vermífugos utilizados pelos produtores estão fazendo efeito ou não.

Por fim, a metodologia empregada apresenta resultados que irão auxiliar produtores e técnicos no processo de identificação de animais doentes. Fornecendo maior segurança nesse processo e ajudando no controle mais efetivo e preciso, visto que o uso indiscriminado de vermífugos é umas causas do aumentado da resistência do verme HC a eles.

Durante a pesquisa surgiram propostas como trabalhos futuros que podem auxiliar ainda mais no processo de identificação de animais anêmicos e melhorar sua assertividade:

- Realizar testes com mais arquiteturas para detecção de objetos e avaliar o seu desempenho, consumo de memória e empregabilidade tanto na plataforma como no aplicativo;
- Analisar o uso de outros classificadores buscando maximizar o estudo comparativo e utilizar o que melhor se adapta aos dados;
- Realizar mais coletas de imagens, tornando a base de dados mais abrangente e robusta a nível do número de amostras por classe;
- Analisar o uso de técnicas de *Machine Learning* e *Deep Learning* para geração de novas amostras de imagens;
- Desenvolver *pipelines* que empreguem uso de técnicas de Processamento Digital de Imagens para correção e melhoria de imagens obtidas, removendo ruídos, distorções e melhorando a qualidade da amostra;
- Empregar algoritmos de apresentem explicabilidade de algoritmos de de detecção e classificação de imagens, afim de se mostrar o que os modelos empregados estão utilizando como parâmetros para chegarem aos resultados;
- Submeter a plataforma desenvolvida a testes com usuários, visando analisar usabilidade e coletar informações para futuras melhorias.

REFERÊNCIAS

ADAMU, M.; DZEVER, P.; IKURIOR, S.; ODE, S.; WILLIAMS, J. Validation of the famacha© system for detecting anaemia and helminthosis in west african dwarf sheep in makurdi, benue state, nigeria. **Comparative Clinical Pathology**, v. 29, p. 15, 2020.

AZURE, M. **O que é PostgreSQL?** 2025. Acesso em: 2025-09-05. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-postgresql>.

BATH; WYK, V. Using the famacha system on commercial sheep farms in south africa. Proceedings of the 5th International Sheep Veterinary Congress, p. 17, 2001.

BORCHARTT, T. B.; RIBEIRO, W. F. Automated detection of anemia in small ruminants using non-invasive visual analysis based on bic descriptor. 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), p. 5, 2023.

BOSQUES, J. **Monitoring gastrointestinal parasite loads in Florida sheep and goat operations**. 2022. Acesso em: 2025-08-22. Disponível em: <https://blogs.ifas.ufl.edu/hardeeco/2022/07/26/monitoring-gastrointestinal-parasites-in-florida-sheep-and-goat-operations/>.

BVMS PHD, D. H. D. H. I. R. T. Vaccines against parasites. In: ____. **Vaccines for Veterinarians**. [S.l.]: Elsevier, 2021. p. 293–300.

CAIN, J. L.; GIANECHINI, L. S.; VETTER, A. L.; DAVIS, S. M.; BRITTON, L. N.; MYKA, J. L. Rapid, automated quantificarion of haemonchus contortus ova in sheep faecal samples. **International Journal for Parasitology**, 2023.

CHAGAS, A. C. d. S.; OLIVEIRA, M. C. d. S.; CARVALHO, C. O. d. Método famacha©:um recurso para o controle da verminose em ovinos. **Circular Técnica**, v. 52, p. 8, 2007.

CONTRIBUTORS, M. **JavaScript**. 2025. Acesso em: 2025-09-03. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>.

(EBAC), E. B. de artes criativas tecnologia. **O que é um Dashboard e como fazer um?** 2024. Acesso em: 2025-09-10. Disponível em: <https://ebaconline.com.br/blog/o-que-e-um-dashboard-e-como-fazer-um#:~:text=Um%20dashboard%2C%20tamb%C3%A9m%20conhecido%20como,na%20tomada%20de%20decis%C3%B5es%20estrat%C3%A9gicas.>

EIS, D. **Guia Front-End**. [S.l.]: Casa do código, 2015. v. 1.

FERREIRA, F.; BORGES, H. S.; VALENTE, M. T. Refactoring react-based web apps. **Journal of Systems and Software**, v. 215, 2024.

FLASK'S. **Flask**. 2025. Acesso em: 2025-09-10. Disponível em: <https://flask.palletsprojects.com/en/stable/>.

GONZALEZ R. C.; WOODS, R. C. **Processamento digital de imagens**. [S.l.]: Pearson Educación, 2009. v. 1.

GRILLO, F. D. N.; FORTES, R. P. de M. **Aprendendo JavaScript**. [S.l.: s.n.], 2008. v. 1.

GROUP, T. P. G. D. **A Brief History of PostgreSQL**. 2025. Acesso em: 2025-09-04. Disponível em: <https://www.postgresql.org/docs/current/history.html>.

GROUP, T. P. G. D. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2025. Acesso em: 2025-09-04. Disponível em: <https://www.postgresql.org/>.

GROUP, T. P. G. D. **What Is PostgreSQL?** 2025. Acesso em: 2025-09-04. Disponível em: <https://www.postgresql.org/docs/current/intro-whatis.html>.

GéRON, A. **Mãos à obra: Aprendizado de Máquina com Scikit-Learn TensorFlow**. [S.l.]: Alta Books, 2019. v. 1.

JETBRAINS. **Kotlin**. 2025. Acesso em: 2025-09-10. Disponível em: <https://kotlinlang.org/>.

KALINOWSKI, M.; ESCOVEDO, T.; VILLAMIZAR, H.; LOPES, H. **Engenharia de Software para Ciências de Dados**. [S.l.]: Casa do Código, 2023. v. 1.

MAIA, D.; ROSALINSKI-MORAES, F.; WYK, S. W. Jan Aucamp van; SOTOMAIOR, C. S. Assessment of a hands-on method for famacha© system training. **Veterinary Parasitology**, v. 200, p. 165–171, 2014.

META. **React**. 2025. Acesso em: 2025-08-04. Disponível em: <https://react.dev/>.

NIKOLAOU, S.; GASSER, R. Prospects for exploring molecular developmental processes in haemonchus contortus. **International Journal for Parasitology**, v. 36, p. 859–868, 2006.

NóBREGA, A. **Aplicativo usa inteligência artificial para controle de verminose em ovinos**. 2025. Acesso em: 2025-11-25. Disponível em: <https://www.embrapa.br/busca-de-noticias/-/noticia/103612192/aplicativo-usa-inteligencia-artificial-para-controle-de-verminose-em-ovinos>.

ORG, P. **Python**. 2025. Acesso em: 2025-09-04. Disponível em: <https://www.python.org/>.

ORG, T. **TensorFlow Lite**. 2021. Acesso em: 2025-09-10. Disponível em: <https://www.tensorflow.org/lite/guide?hl=pt-br>.

PESSANHA, F.; MCLENNAN, K.; MAHMOUD, M. Towards automatic monitoring of disease progression in sheep: A hierarchical model for sheep facial expressions analysis from video. 15th IEEE International Conference on Automatic Face and Gesture Recognition, p. 7, 2020.

ROBOFLOW. **Roboflow: Computer vision tools for developers and enterprises**. 2025. Acesso em: 2025-09-18. Disponível em: <https://roboflow.com/>.

ROCHA, M. B. d.; SARMENTO JOSé LINDENBERG ROCHA; SANTOSA, N.; RABELO, R. A. L.; SILVAO, R. R. V. E.; FILHO, A. O. d. C.; ARAÚJO, F. H. D. d. Método computacional para automação do famacha em cabras e ovelhas. Anais da Escola Regional de Computação do Ceará, Maranhão e Piauí (ERCEMAPI), v. 12, p. 31–40, 2024.

SANTOS, I. B.; FERREIRA, A. U. C.; RABELO, M. D.; ANHOLETO, L. A.; SOUSA, G. A.; GAíNZA, Y. A.; FIGUEIREDO, A.; ESTEVES, S. N.; CHAGAS, A. C. S. Portable near-infrared spectroscopy: A rapid and accurate blood test for diagnosis of haemonchus contortus infection and for targeted selective treatment of sheep. **International Journal for Parasitology**, v. 53, p. 8, 2023.

SIDDIQUE, A.; KHAN, S.; TERRILL, T. H.; MAHAPTRA, A. K.; PANDA, S. S.; MORGAN, E. R.; PECH-CERVANTES, A. A.; RANDALL, R.; SINGH, A.; BATCHU, P.; GURRAPU, P.; WYK, J. A. van. Smart farming with ai: Enhancing anemia detection in small ruminants. **Veterinary Parasitology**, v. 338, 2025.

SILVA, J. V. de A. Aplicativo para detecção de anemia em ovinos. 2025.

SILVA, S. R.; SACARRãO-BIRRENTO, L.; ALMEIDA, M.; RIBEIRO, D. M.; GUEDES, C.; MONTAñA, J. R. G.; PEREIRA, A. F.; ZARALIS, K.; GERALDO, A.; TZAMALOUKAS, O.; CABRERA, M. G.; CASTRO, N.; ARGüELLO, A.; HERNáNDEZ-CASTELLANO, L. E.; ALONSO-DIEZ, J.; MARTíN, M. J.; CAL-PEREYRA, L. G.; STILWELL G., . d. A. A. M. Extensive sheep and goat production: The role of novel technologies towards sustainability and animal welfare. **Animals**, v. 12, n. 7, 2022.

SOUZA, L. F.; COSTA, M. H.; RIET-CORREA, B. Mobile app for targeted selective treatment of haemonchosis in sheep. **Veterinary Parasitology**, v. 316, 2023.