

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA - CAMPUS CAMPINA GRANDE

COORDENAÇÃO DO CURSO SUPERIOR BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

DIEGO CARDOSO DOS SANTOS
IVYSSON FERNANDES DE QUEIROZ UCHÔA

SISTEMA AUTOMATIZADO DE IRRIGAÇÃO POR GOTEJAMENTO EM HIDROPONIA BASEADO EM VISÃO COMPUTACIONAL

Campina Grande - PB
2026

Diego Cardoso dos Santos
Ivysson Fernandes de Queiroz Uchôa

SISTEMA AUTOMATIZADO DE IRRIGAÇÃO POR GOTEJAMENTO EM HIDROPONIA BASEADO EM VISÃO COMPUTACIONAL

Monografia apresentada ao Curso de Bacharelado em Engenharia de Computação, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia da Computação.

Instituto Federal da Paraíba

Orientador: Alexandre Sales Vasconcelos
Coorientador: Breno Romero Mendes de Araujo

Campina Grande - PB
2026

Catalogação na fonte:
Ficha catalográfica elaborada por Gustavo César Nogueira da Costa – CRB 15/479

S237s Santos, Diego Cardoso dos.

Sistema automatizado de irrigação por gotejamento em hidroponia baseado em visão computacional / Diego Cardoso dos Santos, Ivysson Fernandes de Queiroz Uchôa. - Campina Grande, 2026.
100 f.: il.

Trabalho de Conclusão de Curso (Graduação em Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2026.
Orientador: Alexandre Sales Vasconcelos
Coorientador: Breno Romero Mendes de Araujo

1. Engenharia de computação. 2. Inteligência artificial - Agricultura de precisão. 3. Hidroponia. 3. Irrigação automatizada. I. Uchôa, Ivysson Fernandes de Queiroz. II. Vasconcelos, Alexandre Sales. III. Araujo, Breno Romero Mendes de. IV Título.

CDU 004.93

DIEGO CARDOSO DOS SANTOS
IVYSSON FERNANDES DE QUEIROZ UCHÔA

SISTEMA AUTOMATIZADO DE IRRIGAÇÃO POR GOTEJAMENTO EM HIDROPONIA BASEADO EM VISÃO COMPUTACIONAL

Monografia apresentada ao Curso de Bacharelado em Engenharia de Computação, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia da Computação.

Trabalho aprovado. Campina Grande - PB, _____ de _____ de 2026.

Prof. Dr. Alexandre Sales Vasconcelos
Orientador

Prof. Me. Breno Romero Mendes de Araujo
Coorientador

Prof. Dr. Jose Antonio Candido Borges da Silva
Membro da Banca

Prof. Dr. Marcelo Jose Siqueira Coutinho de Almeida
Membro da Banca

Campina Grande - PB
2026

AGRADECIMENTOS

Por Diego Cardoso dos Santos

À minha mãe, Eliene, pelo suporte, educação e amor.
Ao meu pai, Josimar, pelo apoio ao projeto, pela firmeza e pelo amor.
À minha avó, Zilda, por ser um anjo na Terra.
A toda a minha família, pelo carinho incondicional, pela fé e pela companhia.
Aos meus amigos, por me fazerem esquecer o que é a solidão.
Aos meus professores e à instituição, por contribuírem de forma indispensável para a minha formação profissional e pessoal.
À minha cadela Emma, por me lembrar da inocência que todos deveriam preservar.
Para todos que, direta ou indiretamente, contribuíram para o desenvolvimento deste trabalho.

Por Ivysson Fernandes de Queiroz Uchôa

Primeiramente, agradeço a Deus por possibilitar a realização deste trabalho.
À minha mãe, Neri, pelo incentivo e compreensão durante os momentos de dificuldade.
Ao meu irmão, Ícaro, pelo suporte e auxílio ao longo deste trabalho.
A Josimar, por compartilhar conosco seus conhecimentos e habilidades, fundamentais para a realização deste projeto.
A toda a minha família, pelo carinho, apoio e companhia durante essa trajetória.
Aos meus orientadores, Prof. Alexandre Sales e Prof. Breno Romero, pela dedicação, orientação e conhecimentos compartilhados ao longo do desenvolvimento deste trabalho.
Agradeço ao Instituto Federal da Paraíba e aos professores do curso, que contribuíram tanto para minha formação acadêmica e profissional.
Aos meus amigos e colegas, pelo apoio e pelos momentos compartilhados ao longo dessa caminhada.
Por fim, agradeço a todos que contribuíram, direta ou indiretamente, para a realização deste trabalho.

Dedicamos este trabalho a Deus, às nossas famílias, aos professores e aos amigos. Em meio às diferenças, encontramos uma interseção; no peso das mudanças de nossas vidas, seguimos. O sofrimento acompanha sempre uma inteligência elevada e um coração profundo. E cá estamos.

RESUMO

A irrigação inadequada constitui um dos principais desafios enfrentados por pequenos e médios produtores rurais, podendo comprometer o desenvolvimento das culturas e aumentar o desperdício de água e insumos. Neste trabalho, é apresentado o desenvolvimento de um sistema embarcado para automação da irrigação em cultivo hidropônico de coentro (*Coriandrum sativum*), utilizando técnicas de visão computacional para identificar automaticamente a fase fenológica das plantas e ajustar a frequência de irrigação de acordo com seu estágio de desenvolvimento. A arquitetura proposta integra uma câmera posicionada sobre a área de cultivo, um *Raspberry Pi* responsável pelo processamento das imagens e execução do modelo de inteligência artificial, um microcontrolador destinado ao acionamento da bomba de irrigação por meio de relé e uma plataforma para monitoramento remoto do sistema. A identificação das fases de crescimento foi realizada por meio do algoritmo YOLO, treinado com um dataset próprio composto por imagens obtidas durante diferentes ciclos de cultivo e condições de iluminação. Os resultados experimentais demonstraram a viabilidade da integração entre o modelo de visão computacional e a arquitetura embarcada proposta. O modelo final apresentou acurácia global de 66,85%, precisão de 91,90%, *recall* de 66,01% e F1-score de 70,77%, permitindo a identificação automática das fases de desenvolvimento utilizadas pelo sistema para definir a estratégia de irrigação. Além disso, o sistema permaneceu em operação durante um ciclo contínuo de cultivo de aproximadamente 25 dias, apresentando estabilidade operacional, recuperação automática após falhas de energia e conectividade, e reduzindo a necessidade de intervenção manual durante seu funcionamento. Os resultados obtidos demonstram a viabilidade técnica da solução proposta para aplicação em sistemas de irrigação hidropônica automatizados, constituindo uma alternativa de baixo custo para integrar visão computacional e sistemas embarcados em aplicações de agricultura de precisão.

Palavras-chave: sistema embarcado; hidroponia; visão computacional; coentro; agricultura de precisão.

ABSTRACT

Inadequate irrigation is one of the main challenges faced by small and medium-sized farmers, potentially compromising crop development while increasing water and input waste. This work presents the development of an embedded system for irrigation automation in hydroponic coriander (*Coriandrum sativum*) cultivation, using computer vision techniques to automatically identify the plants' phenological stage and adjust the irrigation frequency according to their growth stage. The proposed architecture integrates a camera positioned above the cultivation area, a *Raspberry Pi* responsible for image processing and artificial intelligence model execution, a microcontroller dedicated to controlling the irrigation pump through a relay module, and a remote monitoring platform. Plant growth stage identification was performed using the YOLO algorithm, trained on a custom dataset composed of images collected under different cultivation cycles and lighting conditions. The experimental results demonstrated the feasibility of integrating the computer vision model with the proposed embedded architecture. The final model achieved an overall accuracy of 66.85%, precision of 91.90%, *recall* of 66.01%, and F1-score of 70.77%, enabling the automatic identification of plant growth stages used by the system to determine the irrigation strategy. Furthermore, the system operated continuously throughout an approximately 25-day cultivation cycle, exhibiting operational stability, automatic recovery from power and network failures, and reduced manual intervention during operation. The obtained results demonstrate the technical feasibility of the proposed solution for automated hydroponic irrigation systems, providing a low-cost alternative for integrating computer vision and embedded systems into precision agriculture applications.

Keywords: Embedded Systems; hydroponics; computer vision; cilantro; precision agriculture.

LISTA DE FIGURAS

2.1 Representação do cultivo por hidroponia de gotejamento 6
2.2 Placa ESP32-S3-DevKitC-1 v1.1 10
2.3 Microcomputador Raspberry Pi 3B 11
2.4 Camadas do modelo TCP/IP e seus principais protocolos 13
2.5 Diagrama de uma Rede Neural Convolucional (CNN) 18
2.6 Processo de detecção do YOLO 19

3.1 Arquitetura do sistema 26
3.2 Ideia inicial para bancada de cultivo e reservatório de solução nutritiva . . 27
3.3 Bancada de cultivo sem as calhas de cultivo 28
3.4 Visão frontal da bancada com as calhas de cultivo 29
3.5 Visão traseira da bancada com estrutura hidráulica 30
3.6 Ideia inicial para cobertura para cultivo 31
3.7 Visão interna da cobertura 32
3.8 Uso do sombrite 50% junto a cobertura 32
3.9 Visão Geral da bancada e cobertura 33
3.10 Etapas de preparo para o cultivo do coentro 35
3.11 Suporte para captura de imagens com câmera de celular 37
3.12 Captura da bancada de cultivo com câmera de celular 37
3.13 Classificação da fase 1 no Make Sense 39
3.14 Classificação da fase 2 no Make Sense 39
3.15 Classificação da fase 3 no Make Sense 40
3.16 Exemplo de anotação no formato YOLO. 40
3.17 Captura da bancada de cultivo com ESP32-CAM 42
3.18 Instalação da ESP32-CAM no ambiente de cultivo 46
3.19 Posicionamento da câmera em relação ao cultivo hidropônico 47
3.20 Diagrama de blocos para módulo de captura 48
3.21 Bounding boxes resultantes de uma classificação YOLO 49
3.22 Diagrama de blocos para módulo de processamento 50
3.23 Lista com classes identificadas pelo ONNX. 50
3.24 Diagrama de blocos para módulo de controle de vazão 54
3.25 Esquemático do módulo de controle de vazão 54
3.26 Posicionamento do módulo de controle de vazão 55
3.27 Esquemático do Banco de Dados 57
3.28 Página de Status do Dashboard 58
3.29 Página de Logs do Dashboard 58

3.30 Visualização de captura da bancada de cultivo 59

4.1 Peças de impressão 3D utilizadas para válvula motorizada caseira 71
4.2 Válvula motorizada caseira de bronze 72
4.3 Nova válvula motorizada com impressão 3D 73
4.4 Sistema de irrigação com sensor e válvula motorizada 74
4.5 Sistema de irrigação com sensores de fluxo 75

LISTA DE ABREVIATURAS E SIGLAS

BaaS	Backend como Serviço (Backend as a Service)
CE	Condutividade Elétrica
CNN	Redes Neurais Convolucionais (Convolutional Neural Networks)
MLP	Perceptron Multicamadas (Multilayer Perceptron)
CPU	Unidade Central de Processamento (Central Processing Unit)
EEPROM	Memória Somente Leitura Programável Apagável Eletricamente (Electrically Erasable Programmable Read-Only Memory)
ESP32	Microcontrolador ESP32 da Espressif Systems
HTTP	Protocolo de Transferência de Hipertexto (Hypertext Transfer Protocol)
IaaS	Infraestrutura como Serviço (Infrastructure as a Service)
IC	Circuito Integrado (Integrated Circuit)
IDE	Ambiente de Desenvolvimento Integrado (Integrated Development Environment)
IoT	Internet das Coisas (Internet of Things)
IP	Protocolo de Internet (Internet Protocol)
MCU	Unidade de Microcontrolador (Microcontroller Unit)
MQTT	Protocolo de Telemetria para Enfileiramento de Mensagens (Message Queuing Telemetry Transport)
NFT	Técnica do Fluxo Laminar de Nutrientes (Nutrient Film Technique)
ONNX	Formato Aberto para Intercâmbio de Redes Neurais (Open Neural Network Exchange)
PaaS	Plataforma como Serviço (Platform as a Service)
PWM	Modulação por Largura de Pulso (Pulse-Width Modulation)
QoS	Qualidade de Serviço (Quality of Service)
RAM	Memória de Acesso Aleatório (Random Access Memory)
SaaS	Software como Serviço (Software as a Service)
TCP	Protocolo de Controle de Transmissão (Transmission Control Protocol)
TDS	Sólidos Totais Dissolvidos (Total Dissolved Solids)
TLS	Segurança da Camada de Transporte (Transport Layer Security)
URL	Localizador Uniforme de Recursos (Uniform Resource Locator)
YOLO	Você Só Olha Uma Vez (You Only Look Once)

SUMÁRIO

1 INTRODUÇÃO . . . 1
1.1 Objetivos . . . 1
1.1.1 Objetivo Geral . . . 1
1.1.2 Objetivos Específicos . . . 2
1.2 Estrutura e abordagem do documento . . . 2

2 FUNDAMENTAÇÃO TEÓRICA . . . 4
2.1 Hidroponia e Irrigação por Gotejamento . . . 4
2.2 *Coriandrum sativum* - Coentro . . . 6
2.3 Sistemas Embarcados e IoT . . . 8
2.3.1 Microcontroladores . . . 9
2.3.2 Microcomputadores . . . 11
2.4 Sensores, Atuadores e Periféricos . . . 12
2.5 Protocolos de rede e transferência de dados . . . 13
2.6 Computação em Nuvem . . . 15
2.7 Inteligência Artificial e Visão Computacional . . . 16

3 METODOLOGIA . . . 20
3.1 Metodologia para pesquisa bibliográfica . . . 21
3.2 Protótipo Proposto . . . 21
3.3 Requisitos do Protótipo . . . 22
3.4 Arquitetura e Integração do Sistema . . . 23
3.4.1 Arquitetura do Sistema . . . 23
3.4.1.1 Camada de Coleta . . . 23
3.4.1.2 Camada de Rede . . . 23
3.4.1.3 Camada de Processamento de Dados . . . 24
3.4.1.4 Camada de Aplicação/Serviço . . . 24
3.4.2 Fluxo Operacional do Sistema . . . 24
3.5 Construção da Bancada de Cultivo . . . 26
3.6 Cultivo do Coentro e Coleta de Dados . . . 33
3.7 Pré-processamento e Treinamento do Modelo . . . 38
3.8 Caracterização do Dataset . . . 43
3.9 Módulo de Captura de Imagens . . . 45
3.10 Módulo de Processamento e Controle . . . 48
3.11 Definição dos Parâmetros de Irrigação . . . 51
3.12 Controle de Vazão Automatizado . . . 53

3.13 Armazenamento em Nuvem . . . 55
3.14 Dashboard . . . 57
3.15 Procedimentos de Teste e Validação . . . 59
3.15.1 Teste do Módulo de Captura . . . 59
3.15.2 Teste do Módulo de Processamento . . . 60
3.15.3 Teste de Integração Parcial entre Captura e Processamento . . . 60
3.15.4 Teste de Integração Parcial entre Processamento e Irrigação . . . 60
3.15.5 Teste de Recuperação de Falhas . . . 61
3.15.6 Teste do Sistema Integrado . . . 61

4 RESULTADOS E DISCUSSÕES . . . 62
4.1 Desempenho do Modelo de Visão Computacional . . . 62
4.2 Resultados do Sistema Integrado . . . 66
4.3 Eficiência Operacional Alcançada com a Automatização . . . 68
4.4 Resultados do Dashboard . . . 69
4.5 Desafios Enfrentados no Desenvolvimento do Protótipo . . . 69

5 CONSIDERAÇÕES FINAIS . . . 76
5.1 Avaliação do Sistema de Irrigação Automatizado . . . 76
5.2 Lições Aprendidas . . . 77
5.3 Trabalhos Futuros . . . 77

REFERÊNCIAS BIBLIOGRÁFICAS . . . 79

APÊNDICE A – Materiais Necessários para Construção da Bancada Hidropônica . . . 85

APÊNDICE B – Materiais para Cultivo do Coentro . . . 86

APÊNDICE C – Componentes Eletrônicos Utilizados . . . 87

ANEXO A – Diagrama Esquemático da ESP32-CAM . . . 88

ANEXO B – Diagrama Esquemático do Raspberry Pi 3B . . . 89

1 INTRODUÇÃO

A agricultura é uma atividade essencial para a segurança alimentar e o desenvolvimento socioeconômico, especialmente em regiões rurais. No entanto, o uso ineficiente da água continua sendo um desafio, sobretudo para pequenos e médios produtores que utilizam métodos manuais ou temporizados de irrigação. Tais práticas desconsideram as reais necessidades hídricas das plantas em diferentes estágios de crescimento, resultando em desperdício de recursos, aumento de custos e redução da produtividade. (BRITO; BRITO et al., 2025)

No cultivo hidropônico por gotejamento, que utiliza uma solução nutritiva no lugar do solo, o controle da irrigação é ainda mais crítico, já que qualquer desequilíbrio pode comprometer o desenvolvimento das plantas. O coentro (Coriandrum sativum), muito presente na culinária brasileira, é particularmente sensível ao excesso ou à falta de água, sendo necessário ajustar a irrigação conforme cada fase fenológica: germinação, crescimento vegetativo e maturação.

Diante desse cenário, é proposto no presente trabalho o desenvolvimento de um sistema embarcado para o controle automatizado de irrigação em cultivo hidropônico, utilizando técnicas de visão computacional para identificar as fases de crescimento do coentro e ajustar automaticamente o fornecimento de água e solução nutritiva, bem como a criação de um dataset personalizado de imagens do cultivo de coentro em bancada hidropônica para o treinamento e a avaliação dos modelos de inteligência artificial. A arquitetura do sistema integra câmeras, microcontroladores, um microcomputador responsável pelo processamento central e um módulo de relé para acionamento da bomba de irrigação, operando de forma autônoma. O uso de redes neurais convolucionais (CNNs) e algoritmos como YOLO e ONNX permite uma classificação precisa das fases de desenvolvimento das plantas.

1.1 Objetivos

Para orientar o desenvolvimento dessa pesquisa e estabelecer as metas a serem alcançadas, foram definidos um objetivo geral e objetivos específicos, apresentados a seguir.

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo o desenvolvimento e a avaliação de um sistema embarcado para automação da irrigação em cultivos hidropônicos de coentro. Para

isso, é empregado um modelo de visão computacional responsável pela identificação da fase fenológica das plantas, permitindo que o acionamento da irrigação seja ajustado automaticamente de acordo com cada estágio de desenvolvimento.

1.1.2 Objetivos Específicos

- Apresentar o referencial teórico relacionado ao tema;
- Estruturar um sistema de controle de irrigação eletrônico, integrando microcontrolador e módulo de relé para o acionamento automatizado da bomba de irrigação;
- Desenvolver um *dataset* personalizado para o cultivo de coentro em bancada hidropônica;
- Desenvolver e treinar um modelo de visão computacional, utilizando o algoritmo YOLO, para identificar automaticamente as fases de desenvolvimento do coentro;
- Integrar o controle automatizado de irrigação com o classificador de visão computacional, formando um sistema completo de irrigação inteligente;
- Avaliar o desempenho do modelo de visão computacional e a estabilidade operacional do sistema integrado durante ciclos reais de cultivo.
- Realizar a avaliação da capacidade do sistema de operar de forma autônoma, integrando a captura de imagens, a identificação das fases fenológicas, o armazenamento de dados e o acionamento temporizado da irrigação.

1.2 Estrutura e abordagem do documento

Foi adotada uma abordagem aplicada e quali-quantitativa, direcionada ao desenvolvimento e à validação do sistema proposto. Utiliza-se o método experimental, com a construção de um protótipo que integra visão computacional e controle eletrônico de vazão. O sistema é testado em ambiente controlado, permitindo a análise da precisão do modelo de classificação. A metodologia combina medições quantitativas e análises qualitativas para avaliar o desempenho e a viabilidade da solução proposta.

Este trabalho está organizado em seis capítulos principais:

- Após esta introdução, no segundo capítulo é apresentado o referencial teórico, reunindo conhecimentos e estudos relacionados aos temas centrais: cultivo hidropônico, sistemas de irrigação por gotejamento, inteligência artificial e visão computacional.
- No terceiro capítulo, é apresentada a metodologia empregada, abordando os critérios técnicos utilizados, o processo de implantação do cultivo, os procedimentos de coleta e análise dos dados, além da realização de testes em ambiente controlado.

- No quarto capítulo, são apresentados os resultados obtidos após o desenvolvimento e implementação do sistema automatizado de irrigação por gotejamento baseado em visão computacional, incluindo as métricas de desempenho do modelo e o comportamento do sistema de irrigação.
- No quinto capítulo, é apresentada a discussão dos resultados, analisando e interpretando os dados obtidos, comparando o desempenho do sistema proposto com métodos manuais de irrigação e contextualizando os achados com a literatura existente.
- No sexto capítulo, são apresentadas as considerações finais do trabalho, sintetizando as principais conclusões, as limitações encontradas e as perspectivas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Conforme discutido anteriormente, o manejo inadequado da irrigação configura-se como um dos principais problemas enfrentados por pequenos e médios produtores rurais, ocasionando desperdício de recursos e impactos negativos na produtividade agrícola. Nesse contexto, o cultivo hidropônico de coentro apresenta potencial para a aplicação de tecnologias como visão computacional, inteligência artificial, sistemas embarcados e computação em nuvem, visando à automação e ao controle inteligente da irrigação. Dessa forma, torna-se fundamental compreender os principais conceitos relacionados ao tema abordado nesta pesquisa.

2.1 Hidroponia e Irrigação por Gotejamento

A hidroponia é uma técnica de cultivo que dispensa o uso do solo, sendo amplamente utilizada para a produção de hortaliças, frutas e verduras. Pode ser implementada tanto em escala doméstica quanto comercial, configurando-se como uma alternativa eficiente ao sistema agrícola tradicional.

Essa técnica baseia-se na manutenção das raízes das plantas imersas em uma solução nutritiva, a qual fornece todos os elementos necessários ao desenvolvimento vegetal. A composição dessa solução deve levar em consideração fatores como a espécie cultivada, o estágio de crescimento, a intensidade luminosa e a temperatura do ambiente (DNOCS, 2023).

De acordo com a Empresa Brasileira de Pesquisa Agropecuária (Embrapa), a solução nutritiva utilizada na hidroponia deve conter os principais nutrientes essenciais ao processo de germinação e crescimento das plantas, como nitrogênio (N), fósforo (P), potássio (K), cálcio (Ca), magnésio (Mg), enxofre (S), ferro (Fe), boro (B), manganês (Mn), cobre (Cu), zinco (Zn) e molibdênio (Mo) (CARRIJO; MAKISHIMA, 2000).

Uma solução nutritiva balanceada para o cultivo hidropônico depende de diversos fatores, como a temperatura da solução, o nível de oxigênio, o pH e a condutividade elétrica (CE). Entre esses parâmetros, o controle da CE, correspondente à concentração de íons presentes na solução, é fundamental para garantir a absorção adequada de água e nutrientes pelas plantas através do sistema radicular. Assim, é necessária a medição regular da CE com o auxílio de um condutivímetro, também conhecido como medidor de TDS (*Total Dissolved Solids*), que apresenta os resultados em p.p.m. (partes por milhão) (HIDROGOOD, 2019; COSTA et al., 2001).

Além da CE, o pH da solução nutritiva constitui outro parâmetro essencial para a

manutenção do cultivo hidropônico. O pH indica o grau de acidez ou alcalinidade da solução, variando de 0 a 14, sendo 7 o ponto neutro. A medição é realizada com um pHmetro, e a faixa ideal para soluções nutritivas situa-se entre 5 e 6, uma vez que este intervalo favorece a disponibilidade de nutrientes para as plantas (DUNN; SINGH, 2017).

Segundo Menezes, chefe da Divisão de Tecnologia e Operações Agrícolas do Departamento Nacional de Obras Contra as Secas (DNOCS), o cultivo hidropônico destaca-se por oferecer diversas vantagens aos produtores em relação à agricultura convencional. Os principais benefícios estão relacionados ao maior rendimento por área, o crescimento acelerado das culturas e a eliminação da necessidade de pesticidas ou agrotóxicos (DNOCS, 2023).

Para a implementação de um sistema hidropônico são necessários três elementos fundamentais: bancadas, sistema hidráulico e estufa.

As bancadas sustentam os perfis hidropônicos e devem ser construídas com materiais como madeira, metal ou plástico. Sua principal função é fornecer apoio à estrutura de cultivo por meio de calhas ou tubos, permitindo o escoamento da solução nutritiva. Essas bancadas devem ser montadas com um leve declive, de modo que a solução seja distribuída a partir do ponto mais alto e, ao escorrer, o excesso não absorvido pelas plantas seja coletado na extremidade inferior e recirculado ao reservatório (HIDROGOOD, 2019).

O sistema hidráulico é responsável pelo armazenamento, bombeamento e drenagem da solução nutritiva excedente. É composto por reservatórios, bombas e perfis hidropônicos. O reservatório concentra toda a solução e é o local onde são realizadas as medições e ajustes necessários. Geralmente, esse reservatório é instalado de forma subterrânea, permitindo que o bombeamento da solução para as bancadas seja realizado com eficiência, retornando por gravidade após o uso (HIDROGOOD, 2019).

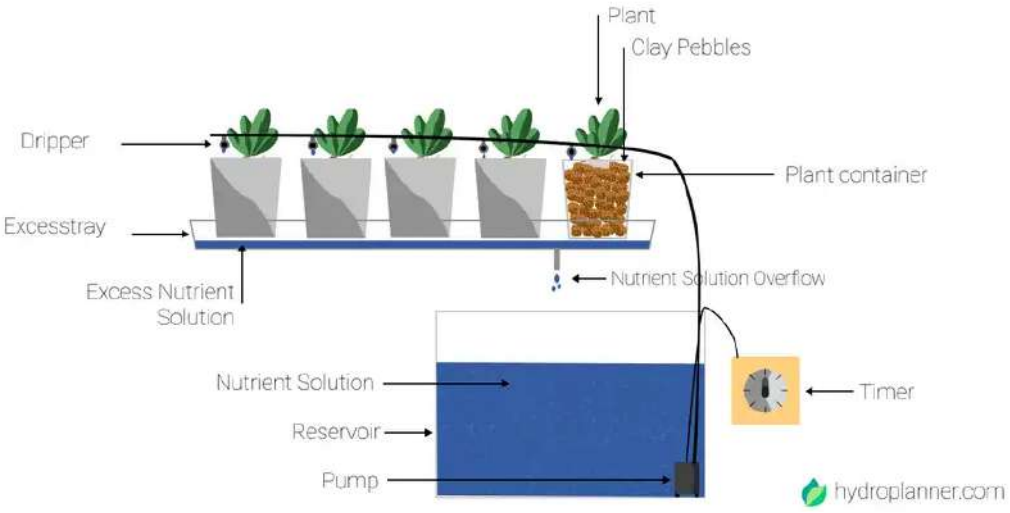
A estufa constitui a estrutura de proteção do cultivo hidropônico. Deve ser projetada para permitir a entrada adequada de luz solar e, ao mesmo tempo, proteger as plantas contra chuvas intensas e ventos fortes, que podem comprometer o desenvolvimento saudável do cultivo.

A partir dessa estrutura básica, o cultivo hidropônico pode ser subdividido em diferentes técnicas, definidas por variações na irrigação, na forma de fixação das plantas e na disposição das bancadas. Entre os métodos existentes, destaca-se a hidroponia por gotejamento, que permite alto controle sobre a irrigação e a dosagem de nutrientes (MCCANDLESS, 2022).

Essa técnica utiliza gotejadores, também chamados de emissores, que controlam o fluxo de água e nutrientes para a planta na forma de gotas. Esses dispositivos podem ser encontrados em uma grande variedade e são dimensionados pelo volume de solução nutritiva que liberam, a maioria dos gotejadores hidropônicos possuindo uma vazão de 0,5 a 7 L/h (FLORAFLEX, 2020; MCCANDLESS, 2022).

No sistema de gotejamento, cada calha ou recipiente possui ao menos um gotejador dedicado, localizado na extremidade das mangueiras ou linhas de irrigação, o qual pode ser ajustado manualmente para regular o volume de água aplicado a cada planta. A Figura 2.1 apresenta um exemplo de um sistema hidropônico por gotejamento. Essa flexibilidade torna o sistema versátil, permitindo que diferentes espécies vegetais recebam volumes distintos de irrigação, de acordo com suas necessidades específicas (TREES, 2022).

Figura 2.1: Representação do cultivo por hidroponia de gotejamento



Fonte: Hydroplanner (2025).

2.2 *Coriandrum sativum* - Coentro

O coentro *Coriandrum sativum L.*, pertencente à família *Apiaceae*, é uma planta anual que apresenta folhas alternas, pinadas e de coloração verde brilhante, caule ereto, flores de tonalidade rosada ou esbranquiçada, além de frutos secos, pequenos e que se separam em dois mericarpos quando maduros (SILVA et al., 2020).

No Brasil, o coentro é amplamente consumido e explorado por diversos produtores, sendo considerado uma cultura de grande relevância socioeconômica. Suas folhas são frequentemente utilizadas na composição e decoração de variados pratos regionais, especialmente no Nordeste do país (CERQUEIRA et al., 2016).

O coentro pode trazer uma série de benefícios para a saúde como: rico em polifenóis, fitoquímicos e carotenóides; ajuda no controle do açúcar no sangue; tem efeito anti-hiperglicemia por estimular a secreção de insulina; reduz o colesterol e triglicérides; tem efeito diurético; funciona como um detox do organismo; auxilia na remoção do mercúrio ingerido por água contaminada; tem função bactericida contra a salmonela e ação antifúngica. Rico em vitamina A, B1, B2, B3 e C, ele ainda conta com ácido fólico,

que é um forte aliado do cérebro, faz bem ao coração, pele, unhas e cabelos, previne o câncer e melhora a imunidade (GIFFONI, 2019).

Na região Nordeste, destaca-se o cultivo da variedade Verdão, adotada por um grande número de produtores ao longo de todo o ano, evidenciando sua importância social e econômica. Essa variedade apresenta características como precocidade, com ciclo de produção de folhas entre 30 e 40 dias, a depender da época do ano e da localidade de cultivo. Além disso, possui bastante vigor, folhas de coloração verde-escura, excelente rusticidade e boa resistência a doenças de folhagens (MELO et al., 2009).

Tais características permitem seu cultivo a partir de técnicas variadas, seja em campo aberto ou em estufas, em canteiros, vasos ou no sistema hidropônico. O sistema de cultivo por hidroponia vem crescendo por inúmeros motivos. Sem a necessidade de preparo de solo e capinas ao longo do ciclo, há melhor aproveitamento dos fertilizantes, que podem reduzir o ciclo em até 5 dias, antecipando colheitas e aumentando a produtividade. Além disso, o ambiente controlado também reduz a incidência de doenças comuns ao cultivo em solo (HORTICULTOR, 2023).

Outra vantagem relevante do cultivo em sistemas hidropônicos é a significativa economia de água. Em sistemas do tipo *Nutrient Film Technique* (NFT), nos quais uma fina lâmina de solução nutritiva circula continuamente pelas raízes das plantas, o consumo hídrico acumulado de um maço de coentro da variedade Verdão, definido como um conjunto de aproximadamente 12 plantas, varia entre 2,75 e 3,86 L ao longo de um ciclo de cultivo de 25 dias, considerando condições adequadas de desenvolvimento, sem ocorrência de estresse hídrico. Esse intervalo corresponde a um consumo médio diário de aproximadamente 130 mL por maço, evidenciando a elevada eficiência hídrica desse método de cultivo. (SILVA, 2019)

O coentro apresenta um ciclo de vida de aproximadamente quatro meses, sendo seu desenvolvimento dividido em etapas distintas, que vão desde a germinação até a floração e a produção de sementes. No contexto da produção agrícola, destacam-se como mais relevantes as fases de germinação, crescimento vegetativo e maturação (YWG, 2022).

O processo começa com o surgimento das primeiras folhas, conhecidas como cotilédones, na fase de germinação. Essas folhas são responsáveis pela nutrição inicial da muda. Após esta etapa, o coentro entra no crescimento vegetativo, caracterizado pelo alongamento do caule, crescimento das raízes e pelo aparecimento das folhas características do coentro, chamadas de folhas verdadeiras. Por fim, quando as folhas verdadeiras se tornam predominantes, a planta atinge o ponto adequado para colheita e consumo (YWG, 2022; TSHANAKAS, 2019).

2.3 Sistemas Embarcados e IoT

Um sistema é classificado como embarcado quando é dedicado a uma única tarefa e interage continuamente com o ambiente a sua volta por meio de sensores e atuadores. Diferentemente dos computadores de uso geral, esses sistemas são otimizados para funções específicas, operando frequentemente com restrições de recursos como memória, processamento e consumo de energia, o que os torna mais eficientes e adequados para aplicações de controle, monitoramento e automação (CHASE; ALMEIDA, 2007).

Junto a isso, o termo IoT (Internet of Things) refere-se a uma rede de dispositivos interconectados baseada em sensores, comunicação e processamento de informações, que permite a coleta e o compartilhamento de dados entre dispositivos e com a nuvem por meio da internet. Nesse contexto, objetos físicos e virtuais possuem identidades próprias, possibilitando seu reconhecimento e comunicação dentro da rede. Devido ao acesso contínuo a dados, a IoT tem sido amplamente utilizada em aplicações como monitoramento de equipamentos, automação de processos e apoio à tomada de decisões.(GOKHALE; BHAT; BHAT, 2018)

A crescente adoção de sistemas embarcados em projetos de Internet das Coisas (IoT) ampliou significativamente seu campo de aplicação. Nesses cenários, esses dispositivos atuam como nós de coleta, processamento e comunicação de dados, integrando sensores, atuadores e módulos de conectividade em soluções compactas e eficientes. Essa aplicação pode ser observada em trabalhos voltados ao monitoramento remoto de plantações, como o proposto por (VELOSO, 2023), cujo objetivo foi aplicar tecnologias de IoT para auxiliar agricultores na tomada de decisões mais eficientes, como a definição do momento ideal para a irrigação de suas lavouras.

Nesse contexto, foi utilizado o microcontrolador Arduino como elemento central para coleta e processamento dos dados, integrado a três sensores principais: o DHT11, responsável pela medição da temperatura e umidade do ar; o higrômetro, utilizado para monitoramento da umidade do solo; e um mini painel solar, empregado para mensurar a luminosidade da lavoura. Além disso, foi desenvolvido um aplicativo web para permitir ao usuário final a visualização, em tempo real, das informações coletadas pelo sistema.

Os sistemas embarcados podem ser implementados em diferentes categorias de hardware, sendo as principais os microcontroladores e os microcomputadores de placa única. A escolha entre essas abordagens depende dos requisitos de cada aplicação, considerando fatores como capacidade de processamento, consumo de energia, necessidade de sistema operacional e complexidade das tarefas a serem executadas.

2.3.1 Microcontroladores

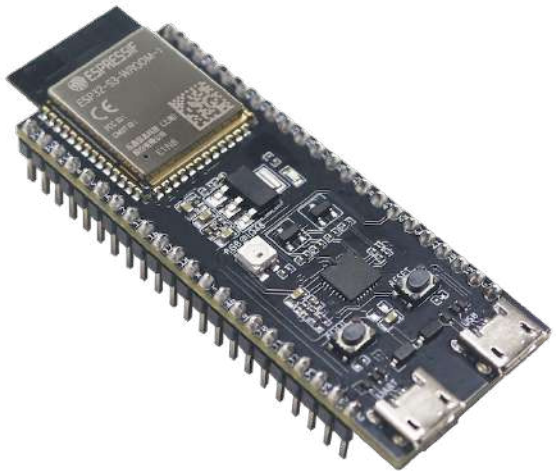
Os microcontroladores (MCUs) podem ser definidos como pequenos computadores integrados em um único chip, projetados para executar tarefas específicas em sistemas embarcados sem a necessidade de um sistema operacional complexo. Esses dispositivos compactos incluem, em um único circuito integrado (IC), o núcleo do processador, memória de acesso aleatório (RAM) e memória somente leitura (EEPROM), permitindo que os programas desenvolvidos para o microcontrolador sejam mantidos mesmo quando não há alimentação elétrica (SCHNEIDER; SMALLEY, 2024).

A crescente demanda por sistemas embarcados no mercado atual impulsiona a importância dos microcontroladores, uma vez que esses dispositivos possibilitam a construção de sistemas computacionais compactos, de baixo custo e com funcionalidades específicas. Atualmente, seu uso é amplamente aplicado em diferentes setores, sendo encontrados em centrais de alarme, teclados de computadores, monitores, discos rígidos, além de aplicações industriais, como controladores de processos, sensores inteligentes, inversores e interfaces homem-máquina (CARDOSO, 2020; KERSCHBAUMER, 2018).

Entre as principais famílias de microcontroladores disponíveis, destacam-se a linha PIC da Microchip, a Intel MCS da Intel e o Atmel AVR da Atmel, sendo esta última especialmente popular devido à utilização nas placas Arduino, uma plataforma amplamente adotada por estudantes e entusiastas de robótica, principalmente pela simplicidade e acessibilidade (CARDOSO, 2020).

Outro exemplo popular e mais avançado de microcontrolador é o ESP32, desenvolvido pela Espressif Systems e apresentado na Figura 2.2. Este dispositivo se destaca pelo processador dual-core de 32 bits, conectividade Wi-Fi e Bluetooth integradas, além de alto desempenho e versatilidade. O ESP32 suporta diversas linguagens de programação, como C/C++, Python, Lua, MicroPython e JavaScript, ampliando sua aplicabilidade em projetos que demandam maior poder de processamento e comunicação (RODRIGUES, 2025).

Figura 2.2: Placa ESP32-S3-DevKitC-1 v1.1



Fonte: Embarcados (2025).

A ampla adoção do ESP32 em diferentes tipos de projetos evidencia sua versatilidade, abrangendo desde aplicações locais e de pequeno porte até soluções de maior escala. Um exemplo de utilização desse microcontrolador é apresentado por (RODRIGUES et al., 2021), que propõe uma solução tecnológica para otimizar o processo de compostagem, prática essencial para a produção de adubo na agricultura orgânica. Nessa proposta, os autores desenvolveram um protótipo físico de uma composteira automatizada, utilizando o ESP32 como unidade central de processamento do sistema. O microcontrolador foi responsável pela coleta de dados de temperatura e umidade, pela análise dessas informações conforme os parâmetros estabelecidos e pelo controle dos atuadores, acionando automaticamente motores de passo acoplados ao protótipo para execução das ações necessárias ao processo.

De forma semelhante, no trabalho proposto por (ANCHESQUI, 2023), é apresentado o desenvolvimento de um sistema inteligente de irrigação baseado em IoT para agricultura de precisão, com foco no uso eficiente da água no setor agrícola. Nessa proposta, são utilizados três dispositivos físicos independentes, construídos com microcontroladores ESP32 e equipados com diferentes sensores para a coleta de dados locais, como variáveis ambientais, umidade do solo e nível de reservatório. Os dados coletados são então transmitidos para um gateway central, também baseado no ESP32, responsável pelo processamento dessas informações e pelo envio dos dados para exibição em uma aplicação web, permitindo o monitoramento remoto.

Esses exemplos demonstram a aplicabilidade e a robustez do ESP32, evidenciando sua capacidade computacional para leitura de sensores, processamento de dados e controle de periféricos. Além disso, destacam sua versatilidade, tornando-o adequado para diferentes

tipos de projetos e aplicações.

2.3.2 Microcomputadores

De forma semelhante, os microcomputadores são classificados como outra categoria de sistemas computacionais, sendo definidos como computadores completos em pequena escala, menores que um mainframe ou um minicomputador, e projetados para uso individual. Diferentemente dos microcontroladores, em um microcomputador são integrados, em uma única unidade denominada placa-mãe, a unidade central de processamento (CPU), a memória RAM, as portas de entrada e saída (E/S) e o sistema de interconexões (barramento) (WRIGHT; SHEA, 2024).

Uma característica fundamental dos microcomputadores é a possibilidade de suporte a diferentes sistemas operacionais (SOs), o que permite a gestão de múltiplos processos e a realização de tarefas simultâneas de forma mais fácil. Essa arquitetura possibilita ampla gama de aplicações, abrangendo desde o desenvolvimento de mini-servidores e testes de projetos computacionais até sistemas de multimídia e automação (MELLO, 2022).

Entre as plataformas mais reconhecidas no mercado, destaca-se o Raspberry Pi, desenvolvido pela Raspberry Pi Foundation. A Figura 2.3 apresenta um exemplo desse microcomputador. Trata-se de uma linha de microcomputadores de placa única, amplamente utilizada em ambientes educacionais e em projetos de ciência da computação. Casos de uso desse sistema incluem servidores de mídia, hubs para dispositivos IoT, sistemas de automação industrial, monitores cardíacos, emuladores de jogos, câmeras, robôs, entre outros (SHIMABUKURO; TOLEDO, 2025).

Figura 2.3: Microcomputador Raspberry Pi 3B



Fonte: Abreu (2025).

Dessa forma, a integração entre microcontroladores e microcomputadores possibilita o desenvolvimento de sistemas embarcados em maior escala, combinando as vantagens de cada abordagem de forma estratégica. Enquanto os microcontroladores oferecem simplicidade,

baixo custo e eficiência para tarefas específicas de controle e atuação, os microcomputadores proporcionam maior capacidade de processamento e versatilidade, sendo adequados para tarefas mais complexas como inferência de modelos de visão computacional.

2.4 Sensores, Atuadores e Periféricos

Utilizados em diversos tipos de aplicações, como na indústria, no comércio e até em ambientes residenciais, os sensores são dispositivos capazes de detectar estímulos físicos ou químicos, como calor, pressão, movimento ou luminosidade, e responder a eles por meio de sinais que podem ser processados e interpretados por outros sistemas (MATTEDE, 2018).

Por outro lado, os atuadores são dispositivos responsáveis por converter estímulos elétricos, hidráulicos ou pneumáticos em potência mecânica e movimento. Esse princípio de funcionamento é necessário para a execução de diversos processos industriais, uma vez que esses dispositivos participam diretamente da operação de máquinas e equipamentos, bem como do acionamento de mecanismos, como interruptores e válvulas.(INÁCIO, 2009; Equipe Nepin, 2025).

A escolha desses dispositivos está diretamente relacionada ao objetivo da aplicação, sendo essencial avaliar as condições do ambiente para a seleção mais adequada a cada necessidade. Esses dispositivos estão presentes em inúmeros setores, tanto industriais quanto cotidianos, destacando-se, entre eles, o agronegócio. Nesse setor, os sensores e atuadores têm papel fundamental em aumentar rendimentos e oferecer maior controle, permitindo, por exemplo, verificar o nível de umidade do solo, controlar o fluxo de água durante a irrigação e, por meio de câmeras especializadas, diferenciar culturas agrícolas de ervas daninhas em questão de milissegundos (MATTEDE, 2018; Bosch, 2021).

As tecnologias utilizadas na construção de sensores são diversas e podem variar de acordo com o tipo de equipamento e a complexidade da aplicação, podendo, em alguns casos, exigir a integração com outros sensores. Um exemplo importante é o sensor de imagem, presente em praticamente todas as câmeras digitais. Esse componente funciona capturando a luz e convertendo-a em um sinal elétrico, que é posteriormente processado para formar a imagem digital. O sensor contém milhões de receptores de luz, chamados fotossítios, que transformam a energia luminosa em uma carga elétrica proporcional à intensidade da luz recebida. Em seguida, os sinais elétricos de todos os fotossítios são enviados ao processador de imagem da câmera, que interpreta as informações e determina os valores de cor e brilho de cada pixel que compõe a imagem digital (HAWKINS; MEYER; SUMMERSBY, n.d.).

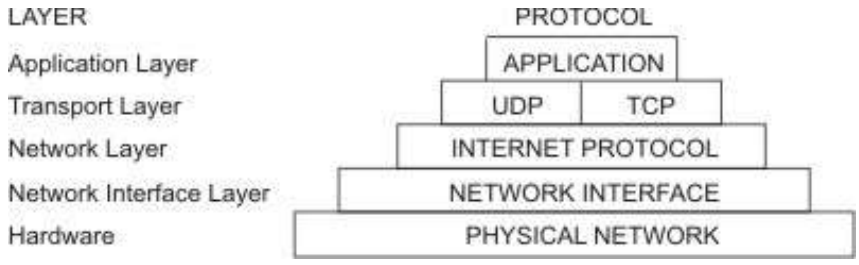
De forma semelhante, os atuadores apresentam grande variedade em sua construção e funcionamento, dependendo do tipo de energia utilizada como princípio, que pode ser

elétrica, hidráulica ou pneumática. Entre eles, destacam-se os atuadores elétricos, que oferecem resposta rápida e permitem acionamentos à distância. Um exemplo bastante utilizado é o relé. Trata-se de um dispositivo eletromecânico que funciona como um interruptor acionado eletricamente. Seu funcionamento baseia-se na aplicação de um sinal elétrico de baixa potência em uma bobina, que gera um campo magnético responsável por acionar os contatos internos. Dessa forma, é possível controlar dispositivos de maior tensão ou corrente por meio de um pequeno sinal elétrico, mantendo o isolamento entre o circuito de controle e o circuito de potência.(CRAVO, 2024; KOHMANN, 2008)

2.5 Protocolos de rede e transferência de dados

Os protocolos TCP/IP são conjuntos de normas que permitem a comunicação entre máquinas conectadas à internet, funcionando como uma linguagem comum entre computadores de diferentes fabricantes e sistemas operacionais. Além disso, organizam as etapas de estabelecimento, controle, transmissão e encerramento da comunicação (TEBALDI, 2019). Essa comunicação é estruturada em camadas, sendo atribuída a cada uma delas uma função específica no envio das informações entre remetente e receptor. São consideradas principais as camadas de Aplicação, Transporte, Rede, Interface de Rede e Hardware. A estrutura hierárquica desse modelo é apresentada no diagrama da Figura 2.4 (IBM, 2021)

Figura 2.4: Camadas do modelo TCP/IP e seus principais protocolos



Fonte: IBM (2021).

O processo de comunicação inicia-se na Camada de Aplicação, que envia os dados para a Camada de Transporte, como o TCP, onde as informações são segmentadas e recebem os endereços de destino. Em seguida, a Camada de Rede encapsula esses dados em datagramas IP e define a rota para o destino. A transmissão ocorre na Camada de Interface de Rede, que converte os datagramas em quadros compatíveis com o hardware utilizado, como Ethernet ou Wi-Fi. No destino, os dados percorrem as camadas no sentido inverso até serem reconstituídos na Camada de Aplicação. (IBM, 2021)

Além dos protocolos já citados, existem diversos outros tipos que podem ser utilizados durante a comunicação em rede, variando de acordo com o objetivo e o tipo de serviço oferecido. Entre os mais relevantes, destaca-se o HTTP, que é fundamental para o acesso e

a transferência de informações na World Wide Web (WWW), sendo amplamente utilizado na comunicação digital moderna (ALICE, 2024).

O HTTP é a base para que o cliente e um servidor web troquem informações. Devido à sua extensibilidade, é utilizado não apenas para acessar documentos de hipertexto, mas também para transferir imagens, vídeos e publicar conteúdo em servidores, tudo por meio de mensagens padronizadas. Nesse protocolo, a comunicação ocorre por meio de mensagens individuais, geralmente por meio de um navegador, sendo as requisições (requests) enviadas pelo cliente e as respostas (responses) retornadas pelo servidor (BEOCK et al., n.d.).

As requisições e respostas HTTP possuem formatos distintos. As requisições incluem o método HTTP (como GET, POST, PUT ou DELETE), o caminho do recurso em formato de URL, cabeçalhos opcionais e, em alguns casos, um corpo de dados. As respostas do servidor contêm um código de status, uma mensagem descritiva, cabeçalhos HTTP e, opcionalmente, o corpo com os dados do recurso solicitado. (MDN, 2024)

Dessa forma, a arquitetura cliente-servidor, combinada com uma comunicação padronizada e simples, permite que o HTTP avance com suas funcionalidades e utilizações, mas mantendo-se fácil de utilizar, possibilitando a criação de aplicações eficientes e seguras (DNC, 2023; ALICE, 2024).

Outro protocolo de comunicação amplamente utilizado em cenários de conectividade distribuída é o MQTT (Message Queuing Telemetry Transport). Esse é um protocolo de mensagens baseado em padrões, amplamente utilizado para comunicação entre dispositivos na Internet das Coisas (IoT). Seu principal objetivo é viabilizar a troca de dados em redes com recursos limitados, como baixa largura de banda e dispositivos de baixo consumo energético, a exemplo de microcontroladores e sensores inteligentes. Para isso, o MQTT utiliza o modelo de publicação/assinatura, no qual clientes publicadores enviam mensagens a um agente central denominado broker, responsável por filtrar e distribuir essas mensagens aos clientes assinantes de tópicos específicos. Essa estrutura garante escalabilidade e simplicidade, além de permitir que os dispositivos troquem informações de forma assíncrona, sem a necessidade de conhecerem a localização ou o estado de funcionamento uns dos outros (AWS, 2025).

Entre suas principais vantagens, destacam-se a leveza e eficiência, já que uma mensagem mínima pode conter apenas dois bytes, otimizando o uso da rede. Além disso, o protocolo suporta milhões de dispositivos conectados, fornece mecanismos de reconexão para ambientes com redes instáveis, define três níveis de Qualidade de Serviço (QoS) que garantem maior confiabilidade na entrega das mensagens e possibilita a implementação de medidas de segurança robustas, como criptografia via TLS e autenticação de clientes. Essas características tornam o MQTT uma escolha consolidada para aplicações que envolvem transmissão de dados entre dispositivos e a nuvem, como automação residencial,

monitoramento remoto e sistemas industriais (AWS, 2025).

Um exemplo da utilização desse protocolo em conjunto com a Internet das Coisas (IoT) é apresentado por (KELLER, 2016), que emprega o MQTT como base da comunicação em um ambiente de automação industrial. Nessa proposta, o protocolo é utilizado principalmente para integrar redes industriais heterogêneas e legadas, que operam com um padrão distinto de comunicação local, o Modbus RTU. Dessa forma, o sistema explora a flexibilidade de roteamento do MQTT ao estruturar os tópicos de publicação como uma ponte para comunicação de longa distância. Para isso, os comandos são capturados na rede local da fábrica, têm suas informações desnecessárias removidas, são convertidos para o formato MQTT e, em seguida, publicados em um servidor na nuvem, o broker. Nesse contexto, o uso do MQTT destacou-se pela confiabilidade exigida em aplicações industriais, uma vez que seus diferentes níveis de qualidade de serviço contribuíram para evitar a perda de dados e impedir a execução duplicada de comandos durante a transmissão pela internet.

2.6 Computação em Nuvem

A computação em nuvem é um modelo que permite disponibilizar recursos computacionais e infraestrutura de tecnologia da informação de forma remota por meio da internet. Nesse modelo, é possível acessar servidores físicos e virtuais, armazenamento de dados, redes, ferramentas de desenvolvimento, softwares, plataformas analíticas e serviços de inteligência artificial utilizando apenas um navegador ou a interface de um provedor de nuvem. Assim, empresas e usuários deixam de depender de infraestrutura própria instalada localmente. (SUSNJARA; SMALLEY, s.d.)

Apesar do termo “nuvem”, esses serviços funcionam por meio do acesso a servidores remotos e computadores de alto desempenho localizados em grandes centros de dados (data centers). Entre suas principais características estão a escalabilidade dos recursos, o acesso a partir de diferentes locais, a manutenção da infraestrutura realizada pelo provedor e a redundância de servidores para maior confiabilidade. Além disso, muitos provedores adotam o modelo de cobrança pay as you go, no qual o usuário paga apenas pelos recursos que utiliza. Essas características tornam a computação em nuvem uma solução atrativa, pois oferece flexibilidade, redução de custos e acesso a tecnologias avançadas sem a necessidade de gerenciar diretamente a infraestrutura. (CALIXTO, 2024)

Dentro desse contexto, existem diferentes modelos de serviço oferecidos pelos provedores de nuvem. Entre os principais estão o "Infrastructure as a Service"(IaaS), ou Infraestrutura como Serviço; o "Platform as a Service"(PaaS), ou Plataforma como Serviço; e o "Software as a Service"(SaaS), ou Software como Serviço. No modelo PaaS, o provedor da nuvem fornece todo o ambiente necessário para desenvolver e hospedar aplicações, permitindo ao desenvolvedor se concentrar no código da aplicação.(CALIXTO, 2024; SUSNJARA;

SMALLEY, s.d.)

Dentro do modelo de Plataforma como Serviço (PaaS), os serviços oferecidos permitem cobrir todo o ciclo de vida do desenvolvimento de uma aplicação. Entretanto, uma das categorias que se destaca é a de Backend as a Service (BaaS), que geralmente oferece diversas ferramentas prontas para o desenvolvimento de back-end, como gerenciamento de dados, autenticação, armazenamento em nuvem, funções em nuvem e integrações com mídias sociais. Entre as plataformas que mais se destacam estão o Firebase e o Supabase. (AYEZABU, 2022)

Enquanto o Firebase, criado pela Google, oferece uma série de ferramentas além dos serviços de backend, como envio de notificações push e hospedagem de sites estáticos, o Supabase, por ser uma plataforma open source, é mais focado em funcionalidades como banco de dados, autenticação e armazenamento em tempo real.

A escolha entre modelos BaaS de código aberto e proprietários representa uma decisão fundamental de arquitetura a longo prazo. Os modelos de código aberto oferecem transparência, potencial de personalização e menor dependência de fornecedores. Entretanto, podem impor maior complexidade na implantação e manutenção. Já os modelos proprietários oferecem infraestrutura gerenciada profissionalmente, suporte abrangente e ecossistemas de serviços integrados, mas criam relações de dependência com o fornecedor e podem impor limitações na flexibilidade de personalização ou migração. (YAZDANKHAH, 2025)

Em quesitos técnicos, uma das principais diferenças entre as duas soluções está na forma como o banco de dados é estruturado. O Supabase é baseado em PostgreSQL (relacional), permitindo maior organização dos dados e consultas mais complexas. Já o Firebase utiliza um modelo NoSQL, que oferece maior flexibilidade e agilidade para consultas menores e estruturas de dados mais dinâmicas. Dessa forma, a escolha entre as plataformas não depende apenas dos serviços oferecidos, mas também da arquitetura de dados que o desenvolvedor necessita. (SIQUEIRA, 2024)

2.7 Inteligência Artificial e Visão Computacional

A Inteligência Artificial (IA) refere-se à capacidade de um conjunto de tecnologias, como computadores e máquinas, de realizar tarefas comumente associadas a seres inteligentes, sendo frequentemente aplicada na simulação de habilidades humanas, como tomar decisões, atribuir significados, generalizar e aprender com experiências passadas (COPELAND, 2022).

Dentro do escopo da IA, destaca-se o campo do aprendizado de máquina (*machine learning*), que compreende o desenvolvimento de modelos a partir do treinamento de

algoritmos com dados. Esses modelos são capazes de realizar previsões e tomar decisões para tarefas específicas, baseando-se em padrões extraídos dos dados (CHEN, 2024).

Entre as diversas técnicas de aprendizado de máquina, uma das mais utilizadas atualmente é a rede neural artificial. Inspiradas na estrutura e no funcionamento do cérebro humano, as redes neurais são compostas por camadas de nós interconectados (análogos aos neurônios), que atuam de forma conjunta no processamento e análise de dados complexos. Essas redes são particularmente eficazes em tarefas que envolvem reconhecimento de padrões e identificação de relações em grandes volumes de dados (KAVLAKOGLU; STRYKER, 2024).

Uma das abordagens mais simples e amplamente aplicadas no aprendizado de máquina é a aprendizagem supervisionada. Nessa modalidade, utiliza-se um conjunto de dados rotulados para treinar algoritmos, associando cada exemplo de entrada a um rótulo de saída. O objetivo é que o modelo aprenda a relação entre essas entradas e saídas, tornando-se capaz de prever corretamente os rótulos de novos dados com base nos padrões identificados durante o treinamento (CANDIDO, 2023).

Entre as aplicações mais relevantes do aprendizado de máquina, especialmente de redes neurais profundas, destaca-se a visão computacional. Essa tecnologia permite que máquinas interpretem e compreendam imagens ou vídeos, simulando a capacidade de visão humana. Por meio dela, é possível não apenas classificar objetos em imagens, mas também identificar sua localização, o que permite diversas aplicações de automação, monitoramento e segurança. Essa funcionalidade é conhecida como detecção de objetos (GONZAGA; NEVEZ; NETO, 2012).

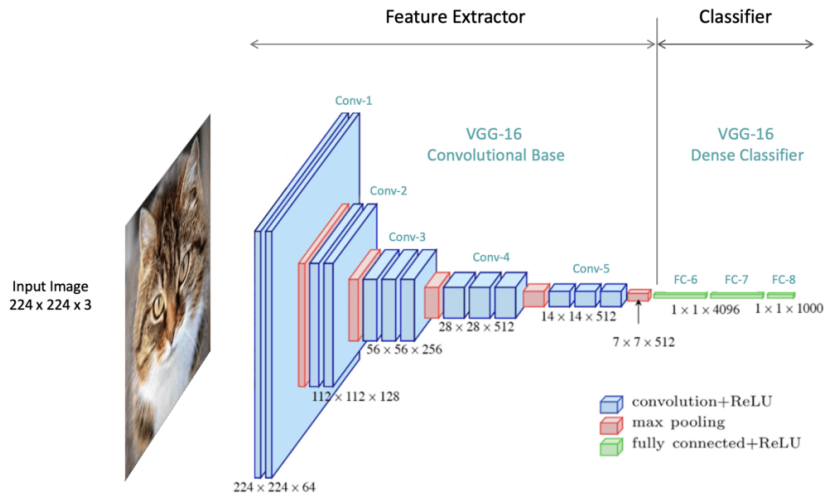
No processamento e na classificação de imagens digitais, limitações importantes são observadas nas redes neurais tradicionais, conhecidas como MLP (*Multi-Layer Perceptrons*). Nessas redes, a matriz de pixels de uma imagem precisa ser convertida em um vetor unidimensional antes de ser processada pela camada de entrada. Como consequência, a estrutura bidimensional da imagem é perdida, comprometendo as relações espaciais e a correlação visual entre pixels vizinhos (BROWNLEE, 2022).

Além da perda de informações espaciais, o processamento de imagens de alta resolução por MLPs resulta em um número elevado de parâmetros. Em uma imagem com resolução de 1280×1024 pixels, seriam necessários mais de 1,3 milhão de valores de entrada para a representação de uma imagem em escala de cinza. Caso uma única camada com 1.000 neurônios fosse conectada diretamente a essa entrada, seriam gerados mais de 1,3 bilhão de pesos. Esse elevado custo computacional aumenta a possibilidade de ocorrência de sobreajuste (*overfitting*) e dificulta a execução da inferência em dispositivos com recursos limitados de processamento e memória, como os microcomputadores de placa única (UNIQTECH, 2018).

Para que essas limitações fossem superadas, foram desenvolvidas as Redes Neurais CNN (*Convolutional Neural Networks*), projetadas para o processamento de dados organizados em estruturas de grade espacial, como as imagens. Diferentemente das MLPs, a estrutura espacial da entrada é preservada pelas CNNs. Em vez de serem realizadas conexões entre cada neurônio e todos os pixels da imagem, pequenos filtros computacionais são aplicados sobre regiões locais da imagem. Por meio da operação de convolução, mapas de características (*feature maps*) são gerados, permitindo a extração de padrões visuais, como bordas, contornos e texturas (IBM, 2025).

Em uma arquitetura típica de CNN, camadas convolucionais são combinadas sequencialmente com funções de ativação não lineares e camadas de agrupamento (*pooling*). Com isso, a dimensionalidade espacial dos mapas de características é gradualmente reduzida, enquanto as informações visuais mais relevantes são preservadas. Somente nas etapas finais, após a extração hierárquica dos padrões visuais, os dados são convertidos em um vetor (*flattening*) e processados por camadas totalmente conectadas (MLP), às quais é atribuída a tarefa de realizar a classificação final, conforme ilustrado na Figura 2.5 (LACERDA, 2025).

Figura 2.5: Diagrama de uma Rede Neural Convolucional (CNN)



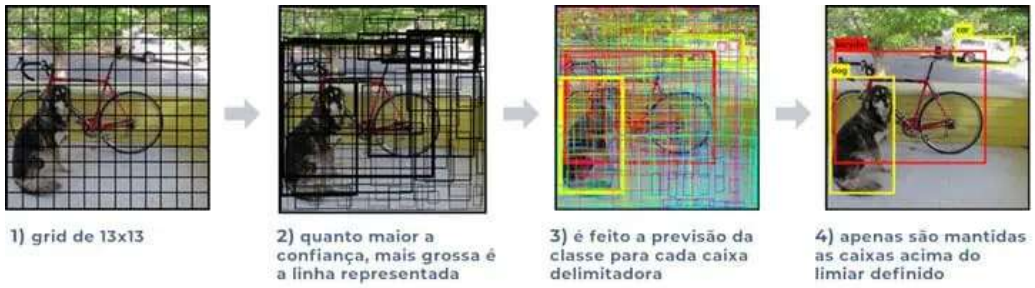
Fonte: Kromydas (2023).

Seguindo essa linha, uma das ferramentas de visão computacional que mais se destacou nos últimos anos é o YOLO (*You Only Look Once*). O YOLO é um método de detecção de objetos de passada única (*single pass*), ou seja, a imagem é analisada apenas uma vez antes de ser processada pela rede neural. Graças a isso e à utilização de uma rede neural convolucional como extrator de características (*features*), é possível realizar detecções com alta velocidade e precisão (ALVES, 2020).

O YOLO trata a detecção de objetos como um problema de regressão, dividindo

a imagem em uma grade composta por várias células. Cada célula é responsável por prever uma ou mais caixas delimitadoras (*bounding boxes*) caso haja objetos presentes naquela região. Para cada caixa, a célula também prevê a classe correspondente, como ilustrado pela Figura 2.6. Esse processo funciona como um classificador, atribuindo uma probabilidade a cada classe possível. O valor de confiança associado à caixa delimitadora é combinado com a probabilidade da classe, resultando em uma pontuação final que indica a chance de a caixa conter determinado objeto (ALVES, 2020).

Figura 2.6: Processo de detecção do YOLO



Fonte: Visão Computacional (2025).

Com o avanço das aplicações de visão computacional em dispositivos embarcados, surgiu a necessidade de formatos padronizados para a representação e portabilidade de modelos de aprendizado de máquina. Nesse contexto, foi desenvolvido o *Open Neural Network Exchange* (ONNX), um formato aberto criado para permitir a interoperabilidade entre diferentes frameworks de aprendizado de máquina, como PyTorch, TensorFlow e Keras. Com o ONNX, um modelo treinado em qualquer um desses frameworks pode ser exportado e executado em ambientes distintos, sem a necessidade de retraining ou adaptações complexas (ONNX Community, 2024).

A execução de modelos no formato ONNX é realizada por meio do ONNX Runtime, um motor de inferência de alto desempenho desenvolvido pela Microsoft, compatível com diversas plataformas de hardware, incluindo dispositivos embarcados com recursos computacionais limitados. O ONNX Runtime aplica otimizações automáticas no grafo computacional do modelo, reduzindo o tempo de inferência e o consumo de memória em comparação à execução do modelo no formato original do framework de treinamento (Microsoft, 2024).

Essas características tornam o ONNX especialmente adequado para sistemas embarcados, em que modelos treinados precisam ser executados em dispositivos com recursos de memória e processamento restritos. A conversão para este formato possibilita uma redução significativa no tempo de inferência, viabilizando aplicações de visão computacional em hardware embarcado sem comprometer a precisão do modelo original.

3 METODOLOGIA

A pesquisa científica consiste na aplicação sistemática de um conjunto de procedimentos objetivos para o desenvolvimento de um experimento e a produção de novos conhecimentos, integrando-os àqueles já existentes (CARDOSO, 2003).

Este trabalho caracteriza-se, por sua finalidade, como uma pesquisa aplicada, categoria que tem como objetivo produzir conhecimentos voltados à solução de problemas concretos e específicos da realidade, gerando inclusive novos processos tecnológicos e produtos (FONTELLES et al., 2009). Essa classificação se justifica pelo fato de o trabalho ter como meta solucionar um problema prático do setor agrícola por meio do desenvolvimento de um sistema automatizado de irrigação voltado para o cultivo hidropônico de coentro (*Coriandrum sativum*), com o objetivo de controlar dinamicamente a vazão de gotejadores utilizando técnicas de visão computacional para identificar as fases de desenvolvimento da planta.

No que diz respeito à abordagem metodológica adotada, a pesquisa pode ser classificada como qualitativa, quantitativa ou quali-quantitativa. A pesquisa qualitativa busca compreender fenômenos complexos sem foco em aspectos numéricos ou estatísticos, enquanto a quantitativa trabalha com variáveis mensuráveis e análise estatística dos dados. Já a abordagem quali-quantitativa integra características de ambas, permitindo combinar interpretação qualitativa e análise quantitativa (COELHO, 2019). Nesse contexto, a abordagem metodológica adotada neste trabalho é quali-quantitativa, combinando diferentes técnicas para garantir uma análise mais abrangente do sistema desenvolvido:

- Técnicas quantitativas, como a medição dos volumes irrigados, a avaliação da precisão do modelo de visão computacional e a coleta de dados sobre o desenvolvimento das plantas ao longo do tempo;
- Técnicas qualitativas, como a análise do desempenho do sistema em diferentes condições operacionais, a avaliação da viabilidade de aplicação prática em campo e a interpretação dos resultados obtidos sob a ótica de usabilidade e eficiência.

Essa combinação permite uma análise mais abrangente e integrada do problema, considerando tanto os aspectos técnicos mensuráveis quanto os contextuais e operacionais.

O método adotado é o experimental, uma vez que o estudo propõe, desenvolve e testa uma solução tecnológica em condições controladas, permitindo observar relações de causa e efeito entre variáveis como estágio de desenvolvimento da planta, tempo de irrigação e resposta do cultivo (SILVEIRA; CÓRDOVA, 2009). O experimento foi conduzido com um protótipo funcional do sistema, operando em ambiente simulado de cultivo, com o objetivo

de analisar o comportamento do sistema e a eficiência operacional alcançada.

3.1 Metodologia para pesquisa bibliográfica

Foram pesquisados trabalhos relacionados ao funcionamento do cultivo hidropônico, com o objetivo de validar a escolha e a viabilidade da temática proposta. Além disso, foram analisados os materiais comumente utilizados por agricultores de médio e grande porte que adotam essa técnica, visando realizar um levantamento prévio dos materiais e dos custos necessários, permitindo avaliar a viabilidade da construção de um sistema hidropônico em escala doméstica de forma acessível.

No que se refere ao uso de modelos de classificação, identificaram-se estudos relevantes na área de reconhecimento e classificação de imagens, que aplicam Redes Neurais Convolucionais (CNNs), com destaque para o uso do modelo YOLO (*You Only Look Once*). Esse modelo tem sido amplamente utilizado na detecção e identificação de hortaliças em sistemas de cultivo hidropônico, servindo como base teórica e técnica para o desenvolvimento da proposta apresentada neste trabalho.

3.2 Protótipo Proposto

Com o intuito de atender aos objetivos estabelecidos, foi concebido o projeto de um protótipo capaz de fornecer os resultados esperados. Esse protótipo é constituído por três módulos principais, interligados e interdependentes, responsáveis pelas funções de captura de imagens, processamento e controle de vazão.

O módulo de captura de imagens é composto por uma ESP32-CAM, ou seja, uma placa de desenvolvimento ESP32 integrada a um módulo de câmera de vídeo OV2640 de 2 MP. Esse dispositivo, que possui conectividade Wi-Fi integrada, atua como um servidor HTTP simples e tem como função capturar imagens da bancada de cultivo sempre que recebe uma requisição GET, transmitindo-as em seguida ao cliente solicitante.

O módulo de processamento e integração é composto por um microcomputador Raspberry Pi 3B, equipado com 1 GB de memória RAM. Esse módulo é responsável por solicitar à ESP32-CAM a captura de imagens da bancada de cultivo e, após recebê-las e armazená-las, aplica um modelo de visão computacional no formato ONNX para classificar a plantação de coentro em três fases: germinação (fase 1), crescimento vegetativo (fase 2) e maturação (fase 3).

Com base nas fases identificadas em cada calha de cultivo, o sistema determina o tempo ideal de irrigação, de modo a atender às necessidades específicas de cada estágio de desenvolvimento da planta. Em seguida, esse parâmetro é enviado ao módulo de controle

de vazão, que realiza os ajustes necessários no sistema de irrigação.

Por fim, o módulo de controle de vazão é formado por um módulo de relé mecânico de 5 V e um microcontrolador ESP32-S3. Ambos os componentes estão integrados à estrutura hidráulica de irrigação da bancada de cultivo. O microcontrolador, com base nos dados obtidos pela ESP32-CAM e no algoritmo de controle do Raspberry Pi 3B, regula a quantidade de solução nutritiva direcionada aos gotejadores ao ajustar os intervalos de irrigação de acordo com as necessidades da fase atual de crescimento da planta.

Em conjunto, os três módulos operam de forma cíclica: o sistema realiza a captura e a classificação das imagens em intervalos definidos, enquanto o monitoramento e o controle da vazão ocorrem de forma contínua, sendo atualizados conforme os resultados obtidos pelo processamento central. Dessa forma, o protótipo assegura um controle automatizado da irrigação na bancada de cultivo hidropônico.

3.3 Requisitos do Protótipo

Os requisitos funcionais do projeto foram definidos com base nas ações que o protótipo deve desempenhar. A seguir, são descritos os principais requisitos funcionais:

- Captura automatizada de imagens nítidas da bancada de cultivo hidropônico;
- Controle preciso dos intervalos de irrigação do sistema de gotejamento através do acionamento do módulo relé;
- Implementação de um algoritmo de visão computacional para a classificação do Coentro, com base em critérios visuais de suas fases de crescimento.

Os requisitos não funcionais do sistema foram definidos com base nas necessidades de desempenho, confiabilidade e usabilidade conforme descrito abaixo:

- O sistema deve realizar o controle de vazão com precisão;
- O modelo de visão computacional deve apresentar precisão e *recall* superior a 80%;
- O hardware (microcontroladores, microcomputador, câmera e relé) deve operar de forma consistente, suportando ciclos contínuos de operação;
- O microcomputador deve manter estabilidade e disponibilidade contínua, garantindo o registro em tempo real dos dados de classificação;
- O sistema deve ser simples de operar para qualquer usuário;

3.4 Arquitetura e Integração do Sistema

Durante a etapa de planejamento do sistema, optou-se pela adoção de uma arquitetura baseada em computação de borda (*Edge Computing*), na qual parte significativa do processamento de dados é realizada nos próprios dispositivos de coleta ou em *gateways* próximos à fonte. Essa abordagem foi escolhida principalmente para evitar o congestionamento da rede e o alto custo de transferência de dados brutos. Ao processar as informações na borda e enviar apenas os resultados ou alertas essenciais, o sistema garante respostas em tempo real, mantém a segurança das informações coletadas e assegura a continuidade das operações mesmo em cenários de conectividade limitada.

3.4.1 Arquitetura do Sistema

A arquitetura proposta é composta por quatro camadas principais, conforme detalhado a seguir:

3.4.1.1 Camada de Coleta

Essa camada é responsável pela aquisição de dados e imagens do cultivo hidropônico.

Dispositivo: ESP32-CAM (módulo ESP32 com câmera integrada OV2640).

Funções:

- Captura periódica de imagens da bancada de cultivo;
- Transmissão das imagens para o servidor local (Raspberry Pi);

Comunicação: Rede Wi-Fi local de 2,4 GHz estabelecida com o servidor Raspberry Pi 3B.

3.4.1.2 Camada de Rede

A camada de rede tem a função de garantir a comunicação confiável entre os dispositivos do sistema, assegurando a troca contínua de informações entre os módulos.

Tecnologia empregada: Rede Wi-Fi local de 2,4 GHz.

Protocolos de comunicação: HTTP (para transmissão de imagens) e MQTT (para controle de vazão).

Função: Estabelecer comunicação bidirecional entre a ESP32-CAM, o Raspberry Pi e o ESP32, permitindo tanto o envio de imagens quanto o recebimento de comandos de controle de vazão.

3.4.1.3 Camada de Processamento de Dados

O processamento central é executado em um servidor local hospedado no Raspberry Pi 3B, responsável pela análise das imagens, classificação das fases de crescimento das plantas e tomada de decisão quanto à irrigação.

Dispositivo: Raspberry Pi 3B com 1 GB de memória RAM.

Funções:

- Execução de um servidor para o recebimento das imagens enviadas pelos módulos ESP32-CAM;
- Armazenamento temporário das imagens para posterior processamento;
- Processamento das imagens por meio do modelo de visão computacional;
- Determinação do tempo ideal de irrigação com base na classificação obtida;
- Envio dos comandos de ajuste de vazão ao ESP32 responsável pelo controle hidráulico.

Comunicação: Rede Wi-Fi local de 2,4 GHz com os componentes ESP32-CAM e ESP32, utilizando os protocolos HTTP e MQTT.

3.4.1.4 Camada de Aplicação/Serviço

Esta camada atua diretamente sobre o sistema físico de irrigação, executando os comandos definidos pelo módulo de processamento.

Dispositivos: Módulo relé de 5 V controlado por um módulo ESP32.

Função: Regulagem da vazão da solução nutritiva por meio de intervalos de abertura e fechamento do relé.

Conexão: Interface de controle via portas GPIO do ESP32 presente no sistema hidráulico.

3.4.2 Fluxo Operacional do Sistema

O funcionamento integrado do sistema segue o fluxo descrito a seguir:

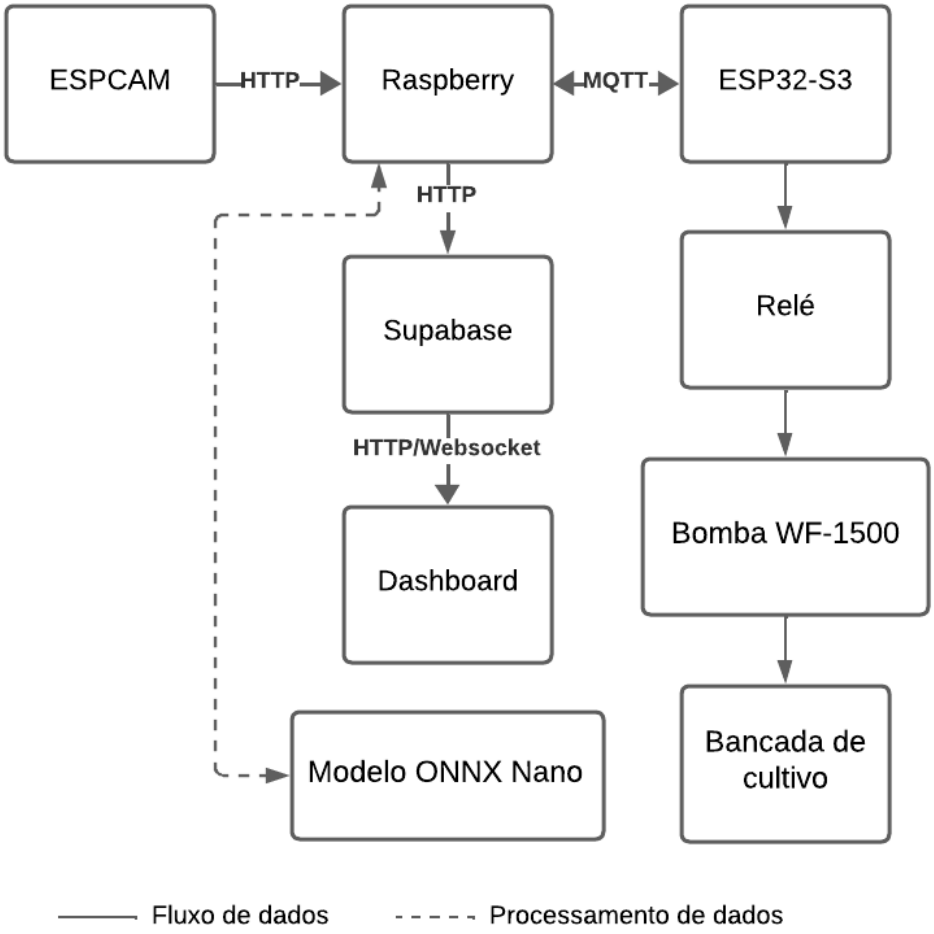
1. **Captura da imagem:** o módulo ESP32-CAM realiza a captura da imagem da bancada de cultivo ao receber uma requisição do servidor local (Raspberry Pi 3B).
2. **Envio da imagem:** a imagem capturada é transmitida via HTTP, utilizando o método *POST*, para o servidor local por meio da rede Wi-Fi.
3. **Recebimento e processamento no Raspberry Pi:** o servidor do Raspberry Pi recebe e armazena temporariamente a imagem. Em seguida, o modelo de visão

computacional processa o arquivo e classifica a fase de crescimento predominante das plantas. Com base nessa classificação, o sistema determina o tempo ideal de irrigação até a próxima captura de imagem.

4. **Envio do comando de vazão:** o Raspberry Pi envia, por meio do protocolo MQTT, o tempo de ativação do relé ao módulo ESP32 responsável pelo controle hidráulico.
5. **Recebimento e armazenamento do comando:** o ESP32 recebe o comando e armazena temporariamente o tempo de irrigação a ser aplicado no sistema.
6. **Controle de ativação do relé:** utilizando do comando recebido pelo servidor, o ESP32 realiza a ativação e desligamento do módulo relé em intervalos consistentes, mantendo o fluxo da solução nutritiva dentro do valor ideal para a fase atual da planta.
7. **Proteção contra falhas:** Em caso de perda de conexão com a internet, a irrigação passa a operar de forma contínua até o restabelecimento da conectividade, a fim de evitar falta de água nas plantas.
8. **Salvamento na Nuvem:** Após o processamento de cada captura, o servidor registra um log e armazena a imagem em um bucket no Supabase.
9. **Visualização em tempo real:** Por meio dos logs armazenados na nuvem, o dashboard web é atualizado continuamente com o status do sistema de irrigação.

Uma representação da arquitetura geral do sistema e do fluxo de comunicação entre os módulos é apresentada na Figura 3.1, permitindo uma visualização mais clara da estrutura proposta.

Figura 3.1: Arquitetura do sistema



Fonte: Autoria própria (2026).

3.5 Construção da Bancada de Cultivo

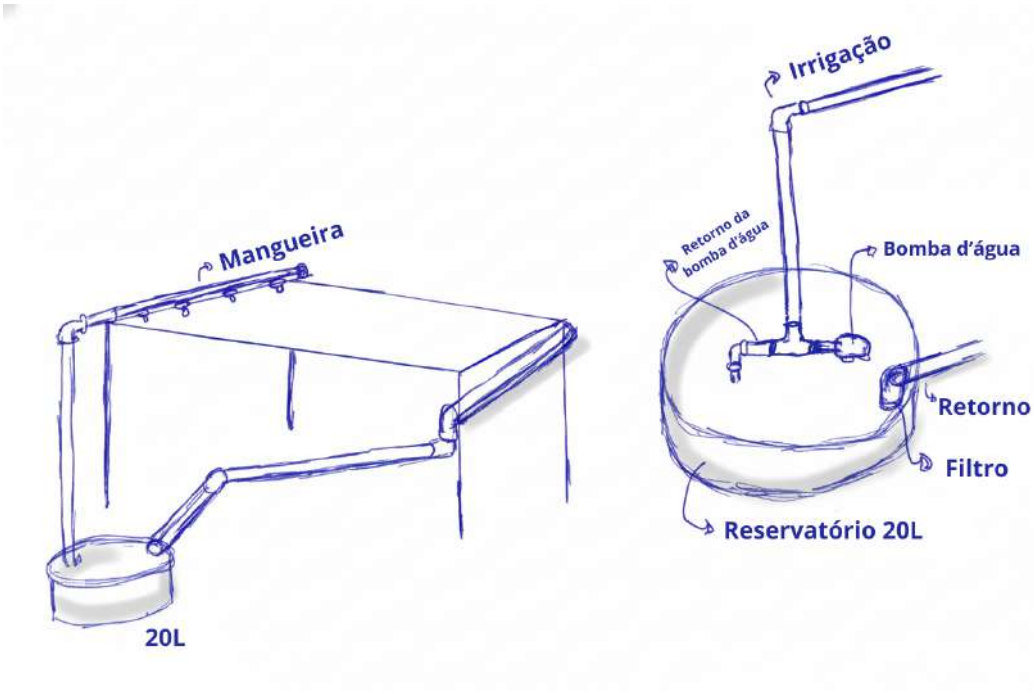
Após a definição dos objetivos e requisitos do projeto, iniciou-se o planejamento do protótipo. Para isso foi necessário primeiramente a elaboração e construção de uma bancada de cultivo hidropônico como principal meio para a coleta de dados e realização de testes.

O ponto de partida foi a busca por referências, realizada por meio de pesquisas em sites, blogs e plataformas de vídeos online, com conteúdos produzidos por agricultores que utilizam a técnica de cultivo hidropônico. Nessa etapa, a principal fonte de inspiração foi o perfil "Exuberante da Amazônia", da plataforma YouTube, que oferece conteúdos educativos voltados ao plantio de hortaliças utilizando hidroponia.

A etapa seguinte consistiu na elaboração do diagrama da bancada hidropônica, conside-

rando as proporções dos testes planejados e as limitações de recursos e espaço disponíveis. O diagrama elaborado é apresentado na Figura 3.2.

Figura 3.2: Ideia inicial para bancada de cultivo e reservatório de solução nutritiva



Fonte: Autoria própria (2025).

A estrutura da bancada foi projetada para ser compacta e funcional, adequada para pequenos espaços e com baixo consumo de recursos, mantendo a viabilidade para cultivo e coleta diversificada de dados, como anotações, observações e imagens.

Após a definição das métricas para a construção, iniciou-se a construção da bancada para o cultivo. Para essa etapa, contou-se com a colaboração de um marceneiro, que cedeu o espaço de sua marcenaria, bem como os equipamentos e apoio técnico necessários.

A bancada foi confeccionada em madeira de eucalipto, utilizando ripas com aproximadamente 2 m de comprimento. Após o processamento da madeira, que incluiu a remoção de rachaduras, o aplainamento e o lixamento, foram produzidas cerca de 11 peças, destinadas à base e aos pés da estrutura. As dimensões foram planejadas para acomodar calhas de 1 m de comprimento e 100 mm de diâmetro. A parte superior da estrutura, responsável pelo suporte das calhas, apresenta 1 m de comprimento por 90 cm de largura. A altura dos pés foi definida em 80 cm e 73 cm, respectivamente, proporcionando uma inclinação de aproximadamente 7%, a fim de favorecer o escoamento da solução nutritiva, conforme ilustrado na Figura 3.3.

Figura 3.3: Bancada de cultivo sem as calhas de cultivo



Fonte: Autoria própria (2025).

O sistema de suporte para o cultivo e irrigação das plantas é composto por quatro calhas de 1 m de comprimento, confeccionadas a partir de um tubo de PVC de 100 mm de diâmetro e 2 m de comprimento. Esse tubo foi cortado ao meio e, em seguida, dividido longitudinalmente, originando duas calhas semicirculares. Para evitar deformações, foi colada uma pequena "tampa" em uma das extremidades com um pedaço de madeira. Para o encaixe das calhas na bancada, foram feitas aberturas semicirculares de 10 cm de diâmetro, dispostas com espaçamento de 10 cm entre si ao longo da largura da estrutura. Essa configuração assegura espaço suficiente para o desenvolvimento das plantas, evitando sombreamento e competição por espaço.

O sistema de irrigação é alimentado por uma bomba submersível WF-1500, instalada em um reservatório. A solução nutritiva é conduzida por uma tubulação de 25 mm que percorre o topo das calhas, sendo distribuída por meio de gotejadores reguláveis do tipo

GA-2, alinhados com as canaletas. Na saída da bomba, foi acoplado um "T"de PVC, funcionando como extravasor (ou ladrão), responsável por aliviar a pressão e o excesso de solução que os gotejadores não conseguem escoar, além de permitir a entrada de ar por um tubo de venturi, promovendo a oxigenação e a constante agitação da mistura. Esses componentes podem ser observados na Figura 3.5.

Após percorrer o sistema, a solução retorna ao reservatório por meio de um sistema de drenagem construído com um tubo de 50 mm. Uma abertura retangular de aproximadamente 70 cm foi feita em um dos lados da bancada, coletando o excedente da solução e conduzindo-o de volta ao reservatório, conforme ilustrado nas Figuras 3.4 e 3.5.

Figura 3.4: Visão frontal da bancada com as calhas de cultivo



Fonte: Autoria própria (2025).

Figura 3.5: Visão traseira da bancada com estrutura hidráulica



Fonte: Autoria própria (2025).

Para garantir proteção contra possíveis chuvas e permitir o controle da luminosidade, foi planejada e construída uma cobertura com dimensões de 1,5 m de largura, 2 m de comprimento e 2 m de altura, espaço suficiente para acomodar o sistema de cultivo e possibilitar a realização de atividades em seu interior. O esboço para a estrutura da cobertura pode ser observado na Figura 3.6.

Figura 3.6: Ideia inicial para cobertura para cultivo



Fonte: Autoria própria (2025).

A cobertura, inspirada em modelos de estufas agrícolas, foi construída com apoio em um muro lateral. O telhado é composto por uma lona plástica transparente, facilmente encontrada em lojas de materiais de construção, esticada sobre uma estrutura retangular de madeira. Na primeira versão, a estrutura possuía apenas travas simples, conferindo certa rigidez, mas mantendo o telhado plano. Entretanto, a ausência de curvatura mostrou-se inadequada, pois impedia o escoamento eficiente da água da chuva, ocasionando seu acúmulo. Para solucionar esse problema, foram adicionadas, no interior da cobertura, travas de madeira em formato semicircular, proporcionando curvatura ao telhado e favorecendo o escoamento da água, evitando o acúmulo sobre a lona, como apresentado na Figura 3.7.

Figura 3.7: Visão interna da cobertura



Fonte: Autoria própria (2025).

Para o controle da intensidade luminosa no interior da estrutura, foi utilizada uma tela de sombreamento do tipo sombrite, que reduz a incidência da radiação solar em cerca de 50%. Essa tela foi instalada de forma removível, como demonstrado na Figura 3.8, permitindo ajustes conforme a necessidade de luminosidade para as plantas cultivadas

Figura 3.8: Uso do sombrite 50% junto a cobertura

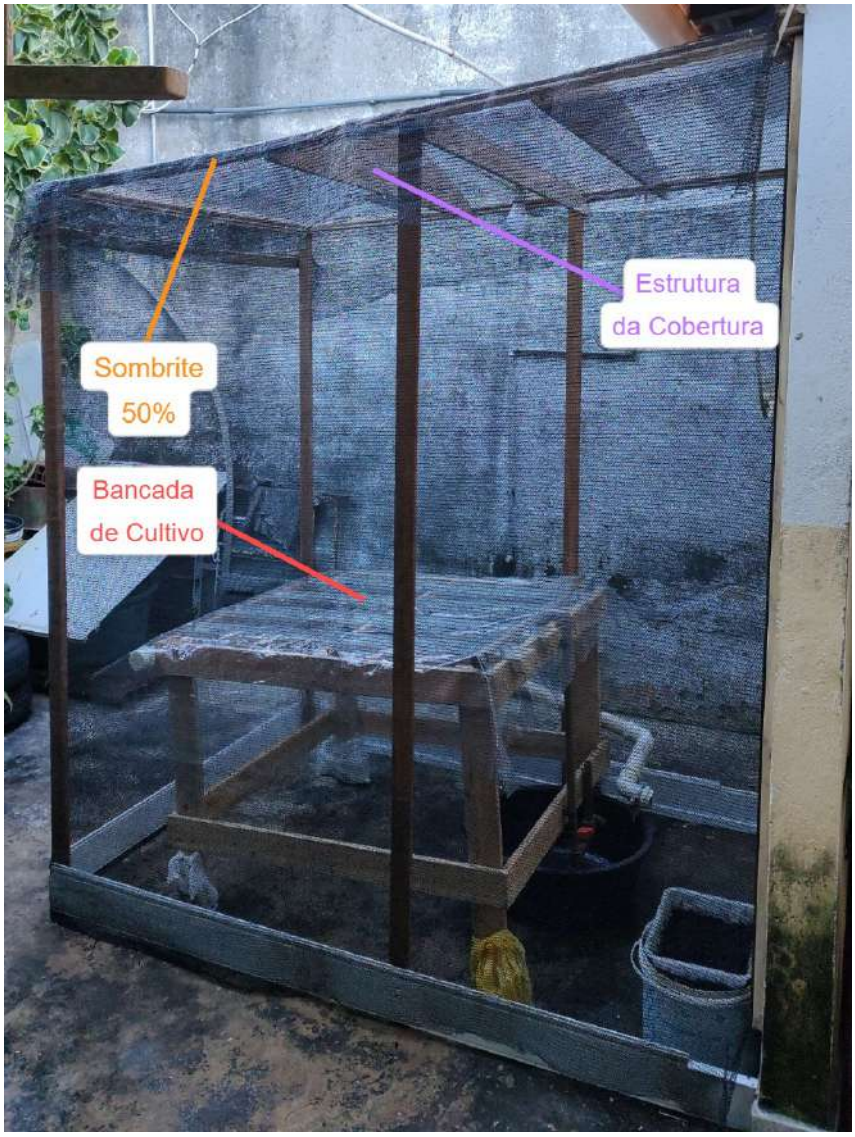


Fonte: Autoria própria (2025).

Com base nessas referências, foi realizado um levantamento inicial dos custos dos materiais necessários para a construção da estrutura, como apresentado no *Anexo 1*. Os preços e quantidades foram estimados a partir das fontes e medições mencionadas acima, complementados por pesquisas em lojas físicas locais e em marketplaces digitais como Amazon e Mercado Livre.

A configuração final da bancada de cultivo, incluindo a estrutura de cobertura e a tela de sombreamento, é apresentada na Figura 3.9.

Figura 3.9: Visão Geral da bancada e cobertura



Fonte: Autoria própria (2025).

3.6 Cultivo do Coentro e Coleta de Dados

Durante a etapa de planejamento do projeto, anterior à construção do conjunto de dados, foi realizada uma busca por *datasets* públicos relacionados ao cultivo de coentro e à identificação de suas fases fenológicas. A pesquisa foi conduzida nas etapas iniciais do desenvolvimento do trabalho, utilizando-se as principais plataformas de repositórios de dados para aprendizado de máquina, incluindo *Kaggle*, *Roboflow Universe* e *Google Dataset Search*.

Para essa busca, foram definidos critérios de inclusão que estabeleciam que as imagens deveriam conter plantas de coentro, apresentar anotações compatíveis com modelos de detecção de objetos, como *bounding boxes*, contemplar as diferentes fases de desenvolvimento

da cultura e estar inseridas em um contexto de cultivo hidropônico.

Entretanto, não foram identificados *datasets* que atendessem de forma satisfatória aos requisitos estabelecidos para este trabalho. Os conjuntos de dados encontrados foram descartados por apresentarem duas limitações principais: a maioria continha imagens de plantas de coentro isoladas e nenhuma das bases realizava a divisão nas três fases fenológicas consideradas essenciais para o controle temporal da irrigação proposto neste estudo. Dessa forma, a construção de um *dataset* próprio foi adotada como alternativa para garantir a padronização das imagens e a correta anotação dos dados utilizados no treinamento e na validação do modelo de visão computacional.

Após a finalização da construção da bancada de cultivo hidropônico, iniciou-se o cultivo do coentro com o objetivo de coletar dados e obter imagens para a composição do *dataset*. Durante esse processo, foram realizadas capturas periódicas das plantas em diferentes estágios de desenvolvimento, possibilitando a formação de uma base de dados representativa das três fases fenológicas consideradas neste trabalho.

Para viabilizar o cultivo e a coleta das imagens, foi necessária a aquisição dos principais insumos utilizados no experimento. O primeiro passo consistiu na escolha das sementes de coentro Verdão (*Coriandrum sativum*), variedade selecionada para este projeto. Além das sementes, foram adquiridos a palha de arroz carbonizada, utilizada como substrato e material para retenção de umidade, a brita zero (pedrita) e os nutrientes empregados na solução nutritiva, conforme detalhado no *Anexo 2*.

Os nutrientes essenciais ao desenvolvimento das plantas foram obtidos por meio da compra de um kit de solução hidropônica, que fornece os macro e micronutrientes necessários para o desenvolvimento das plantas cultivadas nesse tipo de sistema. Também foi adquirida uma pequena quantidade de brita zero (pedrita), muito utilizado na criação de estruturas de concreto e facilmente encontrada em lojas de materiais de construção.

Antes do uso, a brita passou por um processo de lavagem com água corrente para remover sujeiras e impurezas misturadas que poderiam comprometer a qualidade do sistema. Juntamente com isso, realizou-se a quebra da dormência das sementes, mecanismo natural que impede a semente de germinar em momentos inadequados (como períodos de seca ou frio extremo). Esse procedimento foi feito ao macerar levemente as sementes, a fim de criar pequenas fissuras na casca e facilitar a entrada de água.

Com os materiais preparados, iniciou-se a montagem do sistema de cultivo. Primeiramente, as britas foram distribuídas ao longo das calhas, formando uma camada de base. Sobre essa camada, as sementes de coentro foram distribuídas de maneira uniforme para, em seguida, serem cobertas com uma camada do substrato de palha de arroz, como ilustrado pela Figura 3.10.

Figura 3.10: Etapas de preparo para o cultivo do coentro



Fonte: Autoria própria (2025).

Essa disposição tem como objetivo permitir que a água e solução nutritiva dispensadas pelos gotejadores percorram a calha com relativa liberdade através da camada de brita, promovendo a umidificação do substrato e garantindo o contato adequado com as sementes. O substrato utilizado também possui como função atuar como meio de fixação para as raízes ao longo do crescimento da planta.

Após a montagem, as calhas foram umedecidas e cobertas com lona plástica para manter a umidade até o início da germinação, que ocorre entre 4 e 7 dias. Durante esse período, é fundamental manter a umidade constante no ambiente de cultivo, favorecendo a germinação e a quebra da dormência, processo em que a umidade amolece a casca externa resistente da semente, permitindo a entrada de água para a ativação do embrião.

Com o surgimento das primeiras folhas, a lona pôde ser removida e teve início a irrigação por gotejamento utilizando apenas água, por um período de aproximadamente sete dias. Nessa fase inicial, é recomendável que o gotejamento ocorra com baixa vazão, evitando o encharcamento do substrato e possíveis prejuízos ao desenvolvimento da planta.

Após a completa germinação das sementes, iniciou-se a aplicação da solução nutritiva. A preparação dessa solução foi realizada considerando a capacidade do reservatório utilizado para irrigação, bem como as recomendações fornecidas pelo kit de hidroponia utilizado, que podem variar de acordo com as especificações de cada marca. Para o reservatório adotado neste cultivo, foram utilizados 8,5 g do composto de macronutrientes e 8,5 g do

composto de micronutrientes, dissolvidos em 20 L de água.

Em seguida, foi realizada a aferição periódica do pH e da Condutividade Elétrica (CE) da solução, a fim de garantir que o pH permanecesse em níveis levemente ácidos (entre 5,3 e 6,3) e que a CE se mantivesse entre 600 e 700 ppm. A regulação desses parâmetros foi feita por meio de um medidor de pH e de um condutivímetro TDS EC, respectivamente. Para ajustar o pH, utilizou-se ácido fosfórico ou bicarbonato de sódio, conforme a necessidade de torná-lo mais ácido ou mais básico. Já a CE foi corrigida pela adição de nutrientes do kit de hidroponia, quando necessário aumentá-la, ou pela adição de água, quando era preciso reduzi-la.

Esse arranjo foi mantido até o término do ciclo de cultivo, quando o coentro atingiu o tamanho e o grau de maturidade adequados para a colheita. A vazão fornecida pelos gotejadores foi ajustada manualmente ao longo do desenvolvimento das plantas, conforme suas necessidades hídricas em cada fase de crescimento. Próximo ao final do ciclo, foi necessária a reposição do reservatório de solução nutritiva devido ao aumento no consumo pelas plantas.

Durante todo o período de cultivo do coentro, foi realizada a coleta diária de dados referentes às fases de crescimento das plantas, por meio de registros fotográficos da bancada. Para isso, foi instalado um suporte de madeira acima da estrutura de cultivo, próximo à altura da cobertura e centralizado em relação à bancada, como apresentado na Figura 3.11, permitindo o posicionamento de uma câmera ou de um telefone celular e garantindo a captura das imagens de forma padronizada e regular. As fotografias foram obtidas utilizando a câmera de um celular com resolução de 13 megapixels, em pelo menos três horários distintos ao longo do dia: manhã, tarde e noite.

Figura 3.11: Suporte para captura de imagens com câmera de celular



Fonte: Autoria própria (2025).

Ao término dessa etapa de coleta, cuja duração total foi de dois meses, foram conduzidos dois ciclos completos de cultivo, cada um com aproximadamente 21 dias, abrangendo desde a germinação até a colheita. Como resultado, foram obtidas 464 imagens em alta resolução da bancada de cultivo, nas quais foram registrados diferentes estágios de desenvolvimento das plantas e condições de iluminação. Um exemplo dessas imagens é apresentado na Figura 3.12

Figura 3.12: Captura da bancada de cultivo com câmera de celular



Fonte: Autoria própria (2025).

3.7 Pré-processamento e Treinamento do Modelo

Uma vez obtidas imagens suficientes do cultivo para a formação de um *dataset*, todo o material necessário ao treinamento de um modelo YOLO personalizado foi reunido. Antes do treinamento da rede neural, entretanto, é necessário realizar o pré-processamento das imagens coletadas. Essa etapa é fundamental no contexto do processamento de imagens digitais, pois garante que os dados estejam em um formato adequado para as etapas subsequentes de análise e processamento, contribuindo para resultados mais precisos e confiáveis.

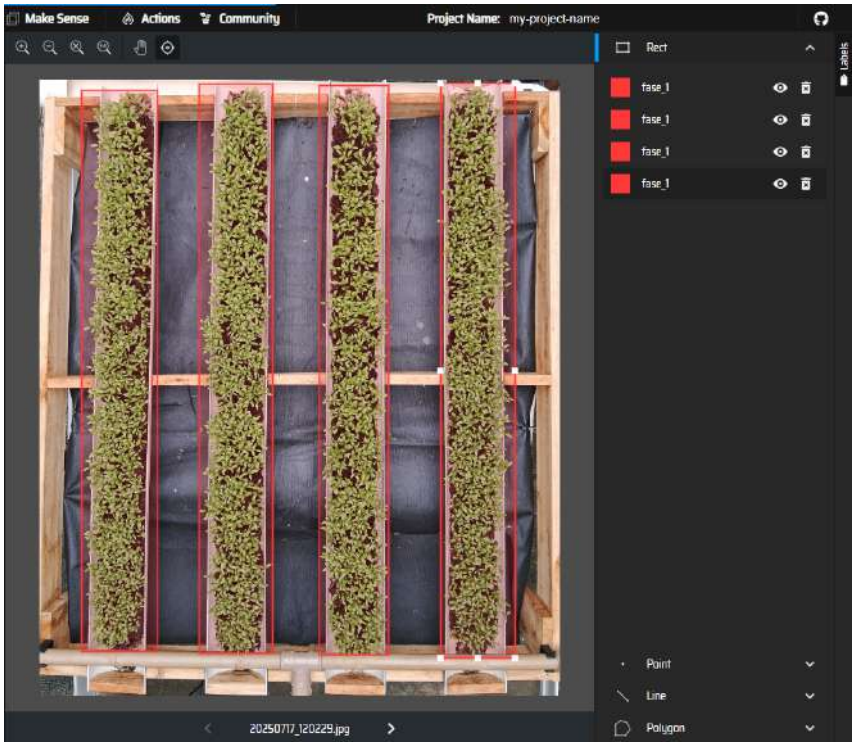
O pré-processamento tem início com o tratamento e edição das imagens. As principais ações realizadas nessa fase foram a remoção de objetos presentes nas bordas das fotografias e o reposicionamento da bancada de cultivo, de modo a centralizá-la nas imagens. Após essa etapa, as imagens foram padronizadas por meio de um script em Python, que reduziu sua resolução, originalmente muito alta, para 1280 pixels de altura por 1024 pixels de largura, valor escolhido por ser compatível com a resolução suportada pela ESP32-CAM, utilizada na construção do protótipo.

Em seguida, foi aplicada a técnica de aumento de dados (data augmentation), com o objetivo de expandir e diversificar o conjunto de treinamento. Essa técnica consistiu na aplicação de espelhamento em aproximadamente metade das imagens e da conversão para escala de cinza em todas elas. Além de duplicar o número de amostras disponíveis para o treinamento, essa abordagem aumenta a variabilidade dos dados, melhorando a capacidade do modelo de reconhecer e generalizar padrões visuais em diferentes condições. Como resultado, o tamanho total do *dataset* passou de 464 para 928 imagens.

O passo seguinte do pré-processamento consistiu na criação das anotações no formato YOLO, armazenadas em arquivos .txt correspondentes a cada imagem do conjunto de dados. Para facilitar esse processo, foi utilizada a plataforma Make Sense, uma ferramenta gratuita e de fácil utilização que permite a identificação manual dos objetos nas imagens e a geração automática dos arquivos em formato compatível com o YOLO.

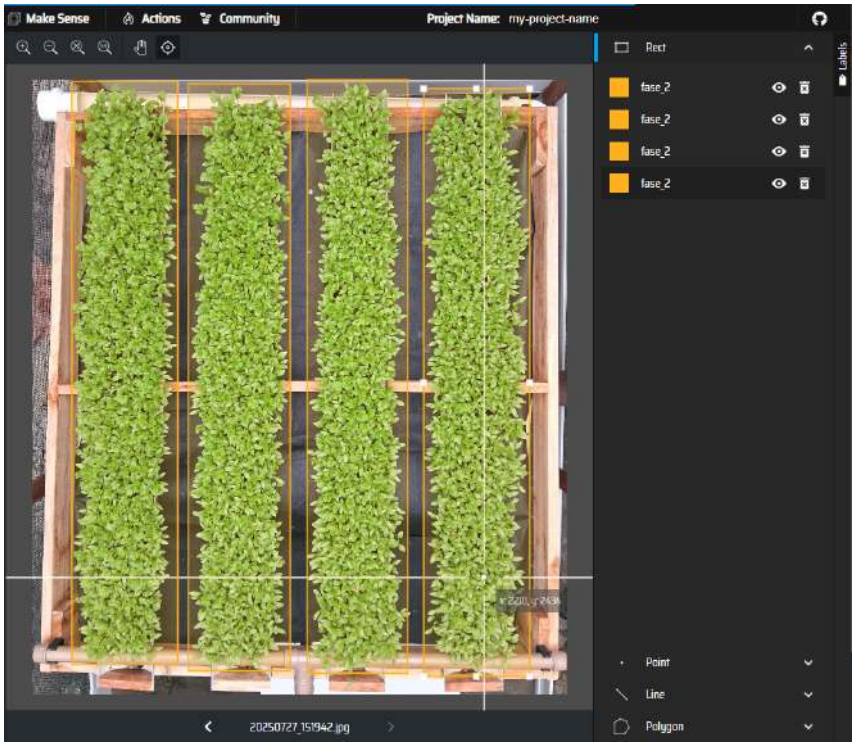
Para a identificação das fases de crescimento nas imagens capturadas da bancada de cultivo, adotou-se como critério o volume e o formato das folhas dos coentros presentes nas calhas. Dessa forma, definiu-se a fase de germinação, ou fase 1, como a presença das folhas iniciais do coentro (cotilédones), caracterizadas pelo tamanho reduzido e formato arredondado, conforme ilustrado na Figura 3.13. O aumento do volume da massa verde, aliado ao surgimento das primeiras folhas características da planta, foi utilizado para definir a fase 2, correspondente ao crescimento vegetativo, ilustrada na Figura 3.14. Por fim, a predominância das folhas verdadeiras e a elevada quantidade de folhas caracterizam a fase de maturação, ou fase 3, apresentada na Figura 3.15.

Figura 3.13: Classificação da fase 1 no Make Sense



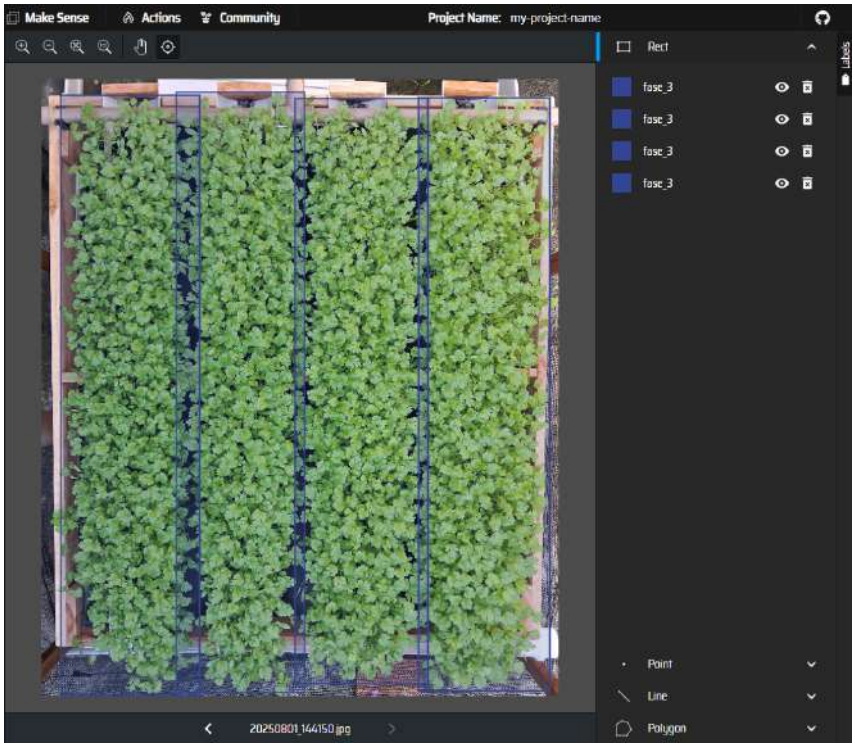
Fonte: Autoria própria (2026).

Figura 3.14: Classificação da fase 2 no Make Sense



Fonte: Autoria própria (2026).

Figura 3.15: Classificação da fase 3 no Make Sense



Fonte: Autoria própria (2026).

Ao final da classificação de todas as imagens, a plataforma Make Sense permite a geração simultânea de todos os arquivos de anotação. Em cada arquivo .txt, cada linha contém cinco informações que, em conjunto, descrevem um objeto identificado na imagem: o identificador da classe e as coordenadas do retângulo delimitador (*bounding box*), conforme ilustrado na Figura 3.16. Essa etapa é fundamental para que o modelo aprenda a localizar e reconhecer corretamente os objetos de interesse nas imagens.

Figura 3.16: Exemplo de anotação no formato YOLO.

```
1 0.154321 0.472612 0.145062 0.910313
1 0.378086 0.468700 0.141975 0.923354
0 0.604167 0.471308 0.134259 0.923354
0 0.842593 0.471960 0.141975 0.932484
```

Fonte: Autoria própria (2026).

Com a conclusão da etapa de pré-processamento das imagens, iniciou-se a preparação para o treinamento do modelo de visão computacional. O primeiro passo consistiu na configuração do ambiente de treinamento Tesla T4 da plataforma Google Colab, escolhida por disponibilizar gratuitamente GPUs (Unidades de Processamento Gráfico), recurso essencial para o treinamento eficiente de redes neurais e modelos de Inteligência Artificial.

O modelo selecionado para essa tarefa foi o YOLOv8, escolhido por sua eficiência no processamento de imagens e pela grande disponibilidade de documentação e suporte na internet. O YOLOv8 oferece cinco variações de arquitetura, que diferem em tamanho e desempenho computacional. Dentre essas, foram selecionadas as versões nano (YOLOv8n) e small (YOLOv8s), por serem mais adequadas para implementação em hardwares embarcados, como os utilizados no protótipo proposto.

Com o ambiente devidamente configurado, foi realizado o upload do *dataset*, já contendo as imagens pré-processadas e divididas em dois subconjuntos: 80% destinados ao treinamento e 20% à validação do modelo. Essa divisão é importante para garantir que o desempenho da rede neural seja avaliado de forma equilibrada, evitando o sobreajuste (overfitting).

Durante o treinamento dos modelos, definiu-se o tamanho da imagem de entrada em 1280x1024 pixels, mantendo a resolução original do dataset. O processo de treinamento foi executado por 100 épocas, utilizando o otimizador AdamW e lotes (*batches*) de 47 imagens por iteração. Além disso, foram aplicadas técnicas automáticas de aumento de dados (*data augmentation*) do próprio algoritmo, como variações de brilho, desfoque e espelhamento, para melhorar a capacidade de generalização da rede.

Ao término do treinamento, os modelos foram exportados para o formato ONNX utilizando funções disponibilizadas pela própria biblioteca ultralytics, o qual é mais leve que o modelo YOLO original, mas preservando sua precisão. A decisão de criação e comparação de modelos em diferentes arquiteturas e versões teve como objetivo identificar aquele com melhor precisão e desempenho, considerando as limitações do hardware embarcado utilizado no projeto.

Posteriormente, após a realização dos testes do sistema, foram obtidas 291 novas imagens do ciclo de cultivo, desta vez utilizando a ESP32-CAM. Essas imagens passaram novamente pela etapa de pré-processamento, resultando em um *dataset* ampliado, com um total de 1.219 imagens e maior diversidade de dados. A partir dessa nova base, foram treinados novos modelos YOLO e ONNX, mantendo-se os mesmos parâmetros adotados no treinamento inicial. Um exemplo dessas novas imagens é apresentado na Figura 3.17

Figura 3.17: Captura da bancada de cultivo com ESP32-CAM



Fonte: Autoria própria (2026).

Para fins de acompanhamento e validação da fase de aprendizagem da rede neural, os resultados obtidos logo ao final do treinamento computacional de ambos os modelos foram registrados. A Tabela 3.1 sintetiza as configurações de dados e as métricas de validação estática (Precisão, Recall, mAP50 e mAP50-95) atingidas ao final das 100 épocas para as duas versões do modelo.

Tabela 3.1: Dados e métricas da fase de aprendizagem dos modelos

Parâmetro / Métrica	**Modelo 1**	**Modelo 2**
Imagens utilizadas (Treino / Validação)	742 / 186	974 / 245
Tempo total de treinamento	1h 39 min	1h 56 min
Velocidade de inferência na GPU	8,1 ms	7,7 ms
Precisão Global (Validation)	99,9%	99,0%
Recall Global (Validation)	100,0%	99,5%
mAP50 Global	99,5%	99,2%
mAP50-95 Global	59,8%	68,1%

Fonte: Autoria própria (2026).

Os resultados obtidos durante o treinamento e a validação indicam que o Modelo 1 apresentou valores de Precisão e *Recall* quase perfeitos em condições controladas de laboratório. Por sua vez, ao Modelo 2, treinado com um conjunto de dados maior e mais diversificado, foram atribuídas métricas de validação ligeiramente mais realistas.

Entretanto, seu melhor desempenho foi observado na métrica mAP50-95, considerada mais rigorosa por avaliar a precisão das caixas delimitadoras (*bounding box*) sob diferentes limiares de interseção. O aumento de 59,8% para 68,1% nessa métrica representa uma melhoria na capacidade de generalização do segundo modelo, indicando maior adequação às condições de captura da bancada hidropônica.

Apesar disso, não se espera que os mesmos valores elevados de desempenho sejam reproduzidos durante os testes em ambiente real. Essa diferença pode ser atribuída à maior variabilidade das condições de operação, incluindo alterações na iluminação e no posicionamento das plantas que não foram completamente capturadas durante a composição do conjunto de dados. Dessa forma, os resultados obtidos durante a validação devem ser interpretados como uma referência do desempenho do modelo em condições controladas, enquanto a avaliação em ambiente real permitirá verificar sua capacidade real.

3.8 Caracterização do Dataset

Após as etapas de coleta, rotulagem, aplicação das técnicas de aumento de dados e divisão dos conjuntos de treinamento e validação, a caracterização do dataset produzido foi realizada. Essa etapa teve como objetivo a análise da composição do conjunto de dados quanto à quantidade de imagens, à distribuição das classes, à organização dos dados e à diversidade das condições de captura, de modo a verificar sua representatividade em relação ao problema proposto.

O dataset final foi composto por 1.219 imagens, das quais 755 corresponderam às imagens originais e 464 foram obtidas por meio das técnicas de aumento de dados descritas na Seção 3.7. As imagens foram rotuladas em três classes correspondentes às fases fenológicas do cultivo do coentro (Fase 1, Fase 2 e Fase 3), totalizando 4.858 objetos anotados por meio de *bounding boxes*. A média de 3,98 objetos por imagem foi diretamente influenciada pela configuração da bancada experimental, composta por quatro calhas hidropônicas cultivadas simultaneamente, o que fez com que aproximadamente uma anotação por calha fosse observada na maior parte das imagens. Por fim, um resumo das principais características do conjunto de dados é apresentado na Tabela 3.2.

Tabela 3.2: Caracterização geral do dataset

Característica	Valor
Imagens originais	755
Imagens após aumento de dados	1219
Número de classes	3
Objetos anotados (bounding boxes)	4.858
Média de objetos por imagem	3.98
Imagens para treinamento	975
Imagens para validação	244
Resolução das imagens	1280 × 1024 pixels

Fonte: Autoria própria (2026).

A distribuição das classes presentes no conjunto de dados também foi analisada. Foram identificadas 2.112 anotações pertencentes à Fase 1, 800 à Fase 2 e 1.946 à Fase 3, correspondendo, respectivamente, a 43%, 16% e 40% do total de objetos anotados.

Um desbalanceamento entre as classes foi observado, caracterizado pela menor quantidade de amostras da Fase 2. Entretanto, essa característica não foi decorrente de limitações durante as etapas de coleta ou rotulagem das imagens, mas das particularidades do próprio ciclo fenológico do coentro. A fase intermediária de desenvolvimento é caracterizada por uma duração significativamente inferior à das fases de germinação e maturação, fazendo com que a quantidade de imagens disponíveis para essa classe seja naturalmente reduzida durante o período experimental. Dessa forma, a distribuição observada pode ser considerada representativa das condições reais de desenvolvimento da cultura.

Além da distribuição das classes, a diversidade das condições de captura das imagens também foi considerada durante a caracterização do dataset. Ao longo dos experimentos, as fotografias foram obtidas em três períodos distintos do dia (manhã, tarde e noite), contemplando diferentes condições naturais de iluminação. No terceiro ciclo experimental, a captura das imagens passou a ser realizada automaticamente em intervalos regulares ao longo do dia, enquanto, nos dois primeiros ciclos, a aquisição foi realizada manualmente em horários previamente definidos. Como resultado, foram obtidas aproximadamente 1.620 imagens no período da manhã, 1.619 no período da tarde e 1.619 durante a noite, proporcionando maior variabilidade das condições de iluminação representadas no conjunto de dados.

Outra característica relevante do dataset refere-se à composição das imagens em relação às fases fenológicas. Em dois dos três ciclos de cultivo, as quatro calhas foram semeadas simultaneamente, fazendo com que todas as plantas presentes em uma mesma imagem

fossem pertencentes à mesma fase de desenvolvimento. No terceiro ciclo, entretanto, duas calhas foram cultivadas com um atraso aproximado de sete dias em relação às demais. Como consequência, foram obtidas imagens contendo simultaneamente plantas em diferentes fases fenológicas, aumentando a diversidade do conjunto de dados e aproximando-o das condições encontradas em aplicações reais.

Com base na caracterização realizada, verificou-se que o dataset é composto por uma quantidade adequada de amostras para o treinamento do modelo, além de apresentar diversidade nas condições de captura e representatividade das três fases fenológicas consideradas neste trabalho. Embora um desbalanceamento entre as classes tenha sido observado, essa característica é decorrente do comportamento natural do cultivo e deve ser considerada na interpretação dos resultados obtidos pelo modelo de classificação, os quais são apresentados no Capítulo 4.

3.9 Módulo de Captura de Imagens

Após a conclusão da etapa de coleta de dados para a construção do *dataset*, foi realizada a aquisição do módulo ESP32-CAM OV2640 por meio do marketplace digital Mercado Livre. Após a compra do dispositivo, optou-se pela utilização da Arduino IDE como ambiente de desenvolvimento, devido à sua simplicidade de uso e à grande disponibilidade de funções e bibliotecas que facilitam o processo de programação e integração do hardware.

O principal objetivo da escolha da ESP32-CAM neste projeto é sua utilização como um servidor web simplificado, permitindo a obtenção remota das imagens capturadas pelo módulo de câmera. Para isso, utilizou-se como base o script CameraWebServer, disponibilizado pela própria Arduino IDE. Esse código possibilita a transmissão de vídeo em tempo real para um navegador web, permitindo a visualização remota do fluxo de imagens gerado pela câmera.

A partir desse código-base, foram realizadas adaptações para melhorar a funcionalidade do sistema, incluindo a alteração da resolução de inicialização da câmera do padrão VGA para HD, a definição de um endereço IP fixo na conexão com a rede Wi-Fi designada, a adição de acesso a configurações extras e a criação de um endpoint para a transmissão de dados entre o servidor Raspberry Pi e a ESP32-CAM.

Durante o processo de desenvolvimento e testes, foram elaboradas duas versões do código, diferenciadas pela forma de transmissão das imagens capturadas. Na primeira versão, ao receber uma solicitação, a ESP32-CAM realizava a captura da imagem e a armazenava na memória antes de enviá-la ao cliente. Apesar de funcional, esse método apresentava falhas em resoluções mais altas devido à limitação de memória do dispositivo. Para solucionar esse problema, foi desenvolvida uma segunda versão, baseada na funciona-

lidade de streaming de vídeo do código original. Nessa abordagem, ao receber a solicitação, a ESP32-CAM inicia rapidamente a transmissão do stream de vídeo, ficando a cargo do cliente a captura e o armazenamento do frame. Essa solução reduziu significativamente os travamentos e problemas de memória da placa.

Com o hardware configurado e o código funcional, a câmera foi instalada no ambiente de cultivo para a captura das imagens, como pode ser observado nas Figuras 3.18 e 3.19. Nessa etapa, foram realizados testes de posicionamento e altura, buscando o melhor enquadramento da bancada de cultivo. Como resultado, foi necessário reduzir a altura do suporte anteriormente utilizado na captura do *dataset*, de modo a melhorar o enquadramento e o reconhecimento das características nas imagens, devido ao efeito de "olho de peixe" causado pela lente do módulo.

Figura 3.18: Instalação da ESP32-CAM no ambiente de cultivo



Fonte: Autoria própria (2025).

Figura 3.19: Posicionamento da câmera em relação ao cultivo hidropônico

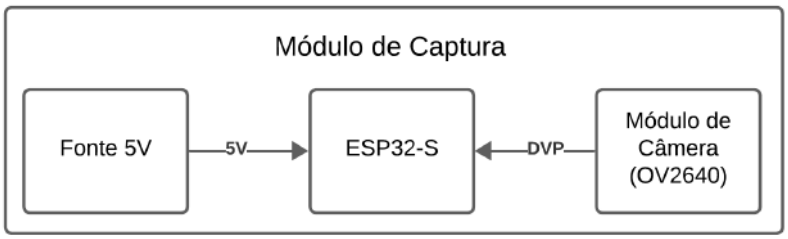


Fonte: Autoria própria (2026).

Outro aspecto relevante identificado ao verificar a qualidade das imagens capturadas foi o desempenho limitado em condições de baixa luminosidade, resultando em imagens escuras quando em ambientes fechados. Entretanto, no ambiente de cultivo, que possui ampla entrada de luz natural devido à estrutura transparente da cobertura, essa limitação não comprometeu o funcionamento do sistema, mesmo em dias mais nublados.

Por fim, a alimentação da ESP32-CAM foi realizada por meio de uma fonte de 5 V e 1 A, semelhante às utilizadas em carregadores de dispositivos móveis. Esses valores foram considerados adequados, garantindo o funcionamento estável do hardware durante todo o período de operação. A montagem é ilustrada pelo diagrama de blocos apresentado na Figura 3.20, enquanto as conexões elétricas realizadas são detalhadas no diagrama esquemático apresentado no *Anexo 4*.

Figura 3.20: Diagrama de blocos para módulo de captura



Fonte: Autoria própria (2026).

3.10 Módulo de Processamento e Controle

Para a elaboração do módulo central, responsável pelo processamento das imagens capturadas da bancada de cultivo e pelo controle dos demais módulos do sistema, foi inicialmente adquirido o BeagleBone Black (BBB), um microcomputador amplamente utilizado em projetos embarcados e que apresenta 32 GB de armazenamento e 512 MB de memória RAM.

Com o intuito de avaliar a viabilidade de sua utilização, especialmente para a execução de um modelo de visão computacional, foram empregadas inicialmente estratégias voltadas ao aumento do desempenho da placa. Entre essas estratégias, destacam-se o uso de sistemas operacionais mais leves, como o Linux Debian, e a configuração de memória swap no cartão de armazenamento, de forma a disponibilizar mais recursos para o processamento. A partir dessas configurações, foi elaborado um script experimental com o objetivo de realizar a classificação e identificação das classes presentes em imagens de teste, utilizando os modelos YOLO e ONNX previamente treinados.

Entretanto, o teste de classificação desenvolvido para os quatro modelos não conseguiu atingir a etapa de execução. Isso ocorreu devido à incapacidade do BeagleBone Black de realizar o download e a instalação das bibliotecas básicas necessárias para a execução dos modelos de visão computacional. Esse impedimento não pôde ser contornado, mesmo após diversas tentativas e buscas por alternativas, o que levou à inviabilidade de utilização do BBB como módulo central de processamento.

Diante dessa limitação, buscou-se um hardware com maior capacidade de processamento e memória, sendo então adquirido um Raspberry Pi 3B. Apesar de depender de um cartão de memória para armazenamento de dados, o dispositivo possui 1 GB de RAM e desempenho superior ao do BeagleBone Black.

O mesmo teste projetado para o BBB foi então executado no Raspberry Pi. A instalação das bibliotecas ocorreu sem erros ou atrasos, permitindo a classificação de três imagens aleatórias do *dataset* por cada modelo previamente treinado, com o objetivo de verificar a

precisão e o tempo de processamento.

Um exemplo do resultado obtido durante essa etapa é apresentado na Figura 3.21, que ilustra as *bounding boxes* produzidas pelo modelo YOLO durante a identificação das fases de crescimento do coentro.

Figura 3.21: Bounding boxes resultantes de uma classificação YOLO

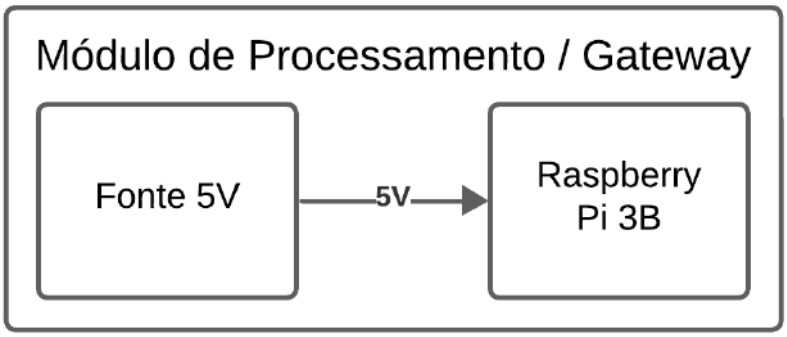


Fonte: Autoria própria (2025).

Com o resultado, optou-se pela utilização do modelo ONNX Nano, que apresentou tempo médio de processamento de pouco mais de 1 s, superando significativamente os demais modelos, cujos tempos foram de 22 s (ONNX Small), 30 s (YOLO Nano) e 62 s (YOLO Small), respectivamente.

Graças ao desempenho satisfatório do microcomputador, não foi necessário empregar técnicas adicionais de otimização de memória ou processamento, como as tentadas anteriormente no BeagleBone Black. A implementação do módulo de processamento é ilustrada pelo diagrama de blocos apresentado na Figura 3.22, enquanto o diagrama esquemático disponível no *Anexo 5* apresenta em detalhes as conexões elétricas do Raspberry Pi.

Figura 3.22: Diagrama de blocos para módulo de processamento



Fonte: Autoria própria (2026).

Definido o hardware a ser utilizado como módulo central do sistema, foi desenvolvido, em linguagem Python, o script responsável pelo gerenciamento do módulo. A estrutura do código inicia-se com a configuração das variáveis e parâmetros necessários ao funcionamento, incluindo a criação de um cliente MQTT para comunicação entre os módulos, e em seguida entra em um loop principal, responsável pela execução contínua das tarefas do sistema.

Dentro do *loop* principal, é realizada uma requisição GET à ESP32-CAM, a fim de obter uma nova imagem da bancada de cultivo. Após o recebimento, a imagem passa por uma etapa de pré-processamento, na qual seu tamanho e resolução são reduzidos para adequação ao modelo ONNX. Em seguida, o modelo de visão computacional é utilizado para classificar a fase de crescimento das plantas em cada calha de cultivo, retornando uma lista com as classes identificadas, conforme exemplificado na Figura 3.23.

Figura 3.23: Lista com classes identificadas pelo ONNX.

```
[
'fase 1',
'fase 1',
'fase 2',
'fase 2'
]
```

Fonte: Autoria própria (2026).

Com base nessas classificações, determina-se a fase de crescimento predominante, e um dicionário de referência contendo os intervalos de irrigação correspondentes a cada fase é consultado. A partir desse resultado, um log é enviado ao Supabase, juntamente com a captura atual da bancada de cultivo, para persistência em nuvem desses dados. Por fim, um comando com essa definição é transmitido via protocolo MQTT ao módulo ESP32 responsável pelo controle físico da vazão de água, indicando os tempos de acionamento e desligamento da bomba.

Adicionalmente, durante a execução do sistema e nos intervalos entre as aferições do cultivo, o servidor verifica, a cada 30 s, o estado de funcionamento do sistema hidráulico por meio do recebimento de mensagens de heartbeat do ESP32 via MQTT. Essa verificação tem como objetivo alertar o usuário caso seja identificado algum travamento ou falha que comprometa a comunicação entre os módulos.

Além disso, o servidor assegura que tanto a captura de imagens quanto o acionamento da bomba ocorram exclusivamente dentro da janela de operação estabelecida, compreendida entre 06:00 e 18:00. Fora desse intervalo, a execução dessas tarefas é automaticamente suspensa, sendo retomada apenas no horário previamente definido. Para garantir a inicialização automática do sistema, comandos em Bash foram configurados no agendador de tarefas crontab, dessa forma, o servidor permanece disponível mesmo após reinicializações inesperadas do microcomputador.

3.11 Definição dos Parâmetros de Irrigação

Por se tratar de um sistema de irrigação por gotejamento aplicado a um ambiente de cultivo reduzido, no qual se utiliza baixa vazão por minuto para cada calha individual, optou-se pela adoção de um método de controle baseado na ativação e desativação temporizada de um relé. Esse método é semelhante ao utilizado em sistemas hidropônicos do tipo NFT, nos quais a bomba de água é acionada em períodos pré-determinados para suprir as necessidades hídricas e nutritivas das plantas, sendo posteriormente desligada por tempo suficiente para manter o substrato úmido. Ao longo do ciclo de cultivo, a vazão total fornecida deve corresponder às exigências da planta em sua respectiva fase de desenvolvimento.

Diante disso, tornou-se necessário definir dois parâmetros fundamentais para o sistema proposto: o volume de solução nutritiva a ser circulado diariamente para cada fase de crescimento do coentro, considerando um período de operação de 12 h, e o intervalo entre as ativações do sistema de irrigação, de modo a evitar o ressecamento do substrato.

De acordo com (LUZ et al., 2017), intervalos de 15 min entre irrigações são comumente utilizados em cultivos na região Centro-Oeste e Sul do Brasil. Entretanto, a literatura aponta que o aumento das temperaturas e o consequente crescimento da evapotranspiração demandam ajustes na frequência de irrigação, tornando intervalos menores mais adequados para prevenir o estresse hídrico (ZANELLA et al., 2008). Considerando essa variação climática, foi realizado um experimento prático com o objetivo de determinar o intervalo mais adequado para o sistema proposto. Para isso, foi preparada uma calha com substrato e conduzidos testes com intervalos de 10 e 15 min entre as ativações da bomba, mantendo-se fixo o tempo de acionamento em 10 min. Após aproximadamente cinco ciclos para cada configuração, e com base na avaliação visual da umidade do substrato, verificou-se que

o intervalo de 10 min proporcionou melhores condições de umidade. Já os intervalos maiores resultaram em ressecamento excessivo, condição que pode comprometer o cultivo, especialmente durante períodos de temperaturas mais elevadas.

No que se refere ao volume de solução nutritiva, considerou-se que a demanda hídrica de um maço de coentro varia entre 2,75 e 3,86 L ao longo de um ciclo de cultivo de 25 dias, conforme indicado por (SILVA, 2019). Levando em conta a capacidade da bancada utilizada, estimada em aproximadamente 20 maços, obteve-se um consumo total entre 55 e 77 L por ciclo, o que corresponde a um consumo diário entre 2,2 e 3 L.

Com base nesses valores e adotando o intervalo de 10 min entre ativações, determinou-se que a vazão média necessária dos gotejadores deveria ser de aproximadamente 76 mL/min para o sistema como um todo, o que implicaria, inicialmente, em tempos de ativação entre 1 e 2 min. Contudo, observou-se que a absorção hídrica do substrato não ocorre de forma imediata, sendo necessária a formação prévia de uma lâmina de água uniforme ao longo da calha para garantir a distribuição adequada da solução nutritiva. Verificou-se experimentalmente que o tempo médio para formação dessa lâmina era em média de 2 min. Dessa forma, os tempos de ativação foram ajustados para um intervalo entre 3 e 5 min, considerando as fases de Germinação à Maturação das plantas.

Para a configuração da vazão do sistema, buscou-se inicialmente o ajuste individual dos gotejadores. Entretanto, uma das dificuldades encontradas durante o desenvolvimento do protótipo foi a determinação precisa do volume de solução efetivamente distribuído para cada calha. Em um primeiro momento, tentou-se utilizar um sensor de fluxo modelo YF-S401 instalado antes da distribuição da solução nutritiva. Contudo, verificou-se que a vazão do sistema era inferior à faixa de operação do sensor, o que resultava em leituras inconsistentes.

Como alternativa, adotou-se um método empírico para medição da vazão. Foram realizadas coletas individuais para cada gotejador em intervalos de 1 min, armazenando-se a água em um copo previamente pesado. Após isso, o volume foi determinado por meio da pesagem da água coletada, desconsiderando a massa do recipiente e assumindo a equivalência entre gramas e mililitros (densidade de 1 g/mL). Esse procedimento permitiu ajustar a vazão individual para aproximadamente 19 mL/min por gotejador, totalizando cerca de 76 mL/min no sistema completo.

Junto a isso, para validar a consistência da vazão e verificar possíveis variações do funcionamento da bomba, foram realizadas dez medições do volume total do sistema, considerando as quatro calhas em conjunto, para diferentes tempos de ativação (1, 3, 5, 10 e 15 min), mantendo-se intervalos de 10 min entre os ciclos. Os dados coletados foram organizados em planilha e analisados, sendo observada uma leve redução da vazão em períodos prolongados de funcionamento contínuo da bomba. Para quantificar esse

comportamento, foi ajustado um modelo de regressão linear aos dados experimentais, resultando na seguinte expressão:

$$Q(t) = -0.000518 * t + 1.553053$$

O modelo apresentou um erro médio absoluto de aproximadamente ±0,12 mL/s, valor considerado aceitável para os tempos de ativação reduzidos adotados neste projeto, não comprometendo, portanto, o desempenho geral do sistema de irrigação

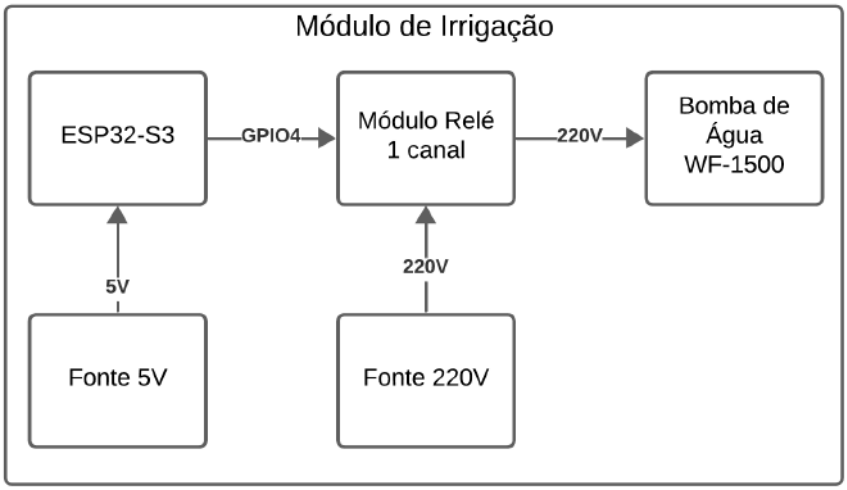
3.12 Controle de Vazão Automatizado

Com a definição dos parâmetros de irrigação, bem como a finalização dos processos de captura de imagens da bancada e de classificação das fases de crescimento do coentro, procedeu-se à construção do módulo de controle de vazão e à sua integração com os demais módulos previamente construídos.

Em termos de arquitetura, o módulo é implementado em uma protoboard e composto por um microcontrolador ESP32-S3, cuja saída de controle está conectada à entrada (IN) de um módulo relé de 5 V por meio de um transistor do tipo BC547, utilizado como elemento de chaveamento.

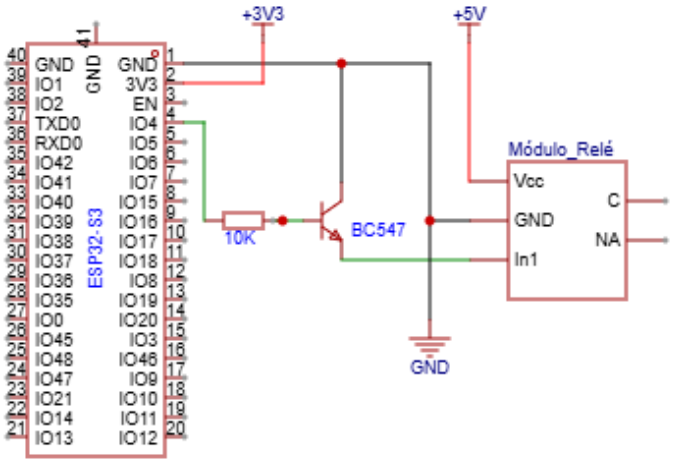
A alimentação do sistema é realizada por um módulo conversor de tensão, responsável por reduzir a tensão de 12 V para 5 V, fornecendo energia tanto ao microcontrolador quanto ao módulo relé. No circuito de potência, o relé é conectado com o cabo de alimentação da bomba de água WF-1500, atuando como elemento responsável pela interrupção ou liberação do fornecimento de energia elétrica ao dispositivo. Os diagramas apresentados nas Figuras 3.24 e 3.25 ilustram essa montagem.

Figura 3.24: Diagrama de blocos para módulo de controle de vazão



Fonte: Autoria própria (2026).

Figura 3.25: Esquemático do módulo de controle de vazão



Fonte: Autoria própria (2026).

Como medida de proteção física, o conjunto foi acomodado em um recipiente plástico com tampa, conforme apresentado na Figura 3.26, no qual foram realizados orifícios para a passagem dos cabos e para a ventilação do sistema, garantindo segurança operacional contra a umidade.

O funcionamento do módulo ocorre da seguinte forma: ao ser inicializado, o ESP32 estabelece conexão com a rede Wi-Fi local e realiza a subscrição em um tópico MQTT hospedado em um broker implementado no Raspberry Pi. Uma vez estabelecida a comunicação, o microcontrolador passa a trocar mensagens com o servidor no formato JSON, enviando, a cada 30 s, um sinal do tipo heartbeat. Essa mensagem contém informações sobre o tempo de execução do sistema desde sua inicialização. Além disso, a

cada alteração no estado do relé, uma mensagem de status é enviada ao servidor. Durante a operação, o ESP32 também permanece em escuta para o recebimento de mensagens provenientes do servidor, contendo comandos de ativação (pump_on) ou desativação (pump_off) da bomba, bem como a duração desses intervalos em segundos.

Figura 3.26: Posicionamento do módulo de controle de vazão



Fonte: Autoria própria (2026).

Considerando que o funcionamento do sistema depende da conectividade com a rede Wi-Fi, foi incorporado um mecanismo de segurança ao módulo de controle de vazão. Em caso de perda de conexão, o ESP32 realiza automaticamente a ativação do relé, mantendo a bomba de água em funcionamento contínuo. Essa estratégia permanece ativa enquanto o dispositivo tenta restabelecer a comunicação com a rede e com o servidor. Tal abordagem visa garantir a continuidade da irrigação, evitando a falta de solução nutritiva no cultivo em situações de longas falhas na comunicação.

3.13 Armazenamento em Nuvem

Para garantir a persistência dos dados e facilitar o acesso a logs sem a necessidade de conexão direta ao armazenamento local do servidor Raspberry Pi, optou-se pelo uso de armazenamento em nuvem por meio do BaaS Supabase. Essa solução permite não apenas a utilização de um banco de dados PostgreSQL com suporte a websockets para atualizações em tempo real, como também o armazenamento de arquivos, sendo especialmente adequada para o gerenciamento das imagens processadas pelo modelo de visão computacional.

A arquitetura de armazenamento no Supabase está estruturada em três componentes

principais: banco de dados, bucket de armazenamento e camada de comunicação com a aplicação.

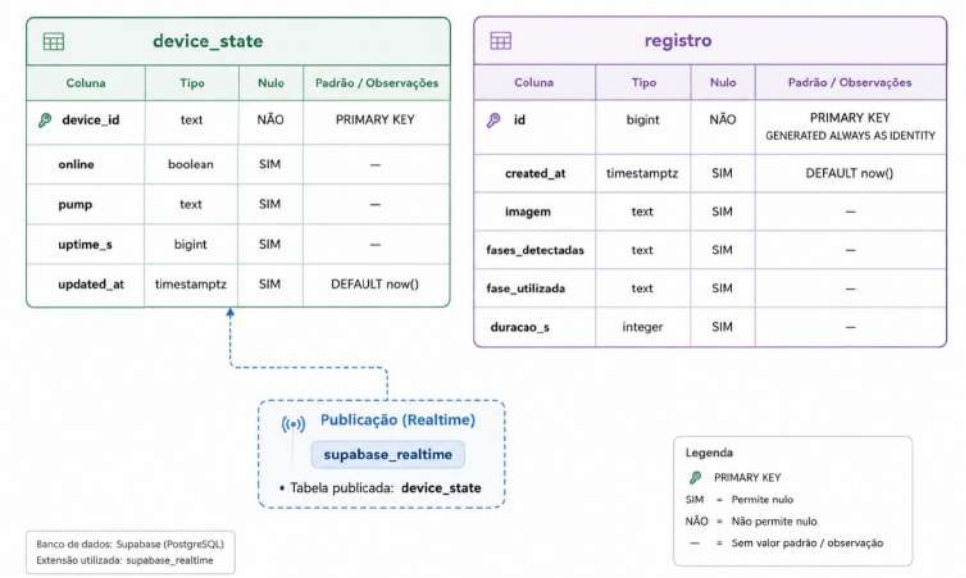
O banco de dados, baseado em PostgreSQL, possui duas tabelas principais, como pode ser observado na Figura 3.27. A primeira, denominada “registro”, é responsável pelo armazenamento de logs operacionais, registrando cada processamento de imagem e a decisão gerada pelo modelo ONNX. Essa tabela armazena informações como data e hora da captura, URL da imagem correspondente no bucket, resultado da classificação, fase predominante nas calhas e duração da ativação da bomba. Esses registros permitem rastreabilidade completa das operações, validação e acompanhamento do sistema ao longo do tempo.

A segunda tabela, denominada "device_state", é destinada ao monitoramento do estado do sistema de irrigação. Diferentemente da anterior, essa tabela mantém apenas um registro, atualizado continuamente via websocket. Nela são armazenados o status (online/offline) do ESP32 responsável pelo controle da bomba, o estado da bomba (ligada/desligada) e o tempo de atividade do dispositivo. Essas informações são atualizadas por meio de um mecanismo de heartbeat enviado pelo ESP32 ao servidor Raspberry Pi, permitindo monitoramento remoto e em tempo real do sistema de irrigação.

Complementarmente ao banco de dados, o bucket de armazenamento do Supabase é utilizado para guardar arquivos não estruturados, como imagens. Após a captura e processamento no servidor Raspberry Pi, cada imagem é enviada ao bucket, que retorna uma URL pública para acesso remoto. As imagens são armazenadas com base em suas data e hora de criação, facilitando a identificação, organização e posterior análise do desenvolvimento do cultivo.

Por fim, a comunicação entre a aplicação e o Supabase é realizada por meio de uma API e de uma ORM disponibilizadas pela própria plataforma. A API permite o envio direto de arquivos para o bucket, enquanto a ORM simplifica a interação com as tabelas do banco de dados, tornando as operações mais eficientes. Para autenticação e acesso ao projeto, o Supabase exige o uso de credenciais específicas, compostas por uma URL e uma chave (API Key) associadas à instância configurada.

Figura 3.27: Esquemático do Banco de Dados



Fonte: Autoria própria (2026).

3.14 Dashboard

A partir da disponibilização dos dados coletados e armazenados no banco de dados do Supabase, tornou-se necessário fornecer uma forma de acesso remoto mais simples e intuitiva. Isso se deve ao fato de que, até então, a visualização completa das informações exigia acesso direto à plataforma ou às credenciais do serviço.

Para solucionar essa limitação, foi desenvolvido um dashboard web em tempo real para o monitoramento do sistema automatizado de irrigação. No front-end, utilizou-se o framework Next.js em conjunto com a linguagem TypeScript. No back-end, a aplicação consome a API fornecida pelo Supabase, incluindo seus recursos de realtime para atualização contínua dos dados. Por fim, a hospedagem da aplicação foi realizada na plataforma Vercel, um serviço em nuvem que permite a publicação de aplicações web e sites estáticos de forma prática e escalável.

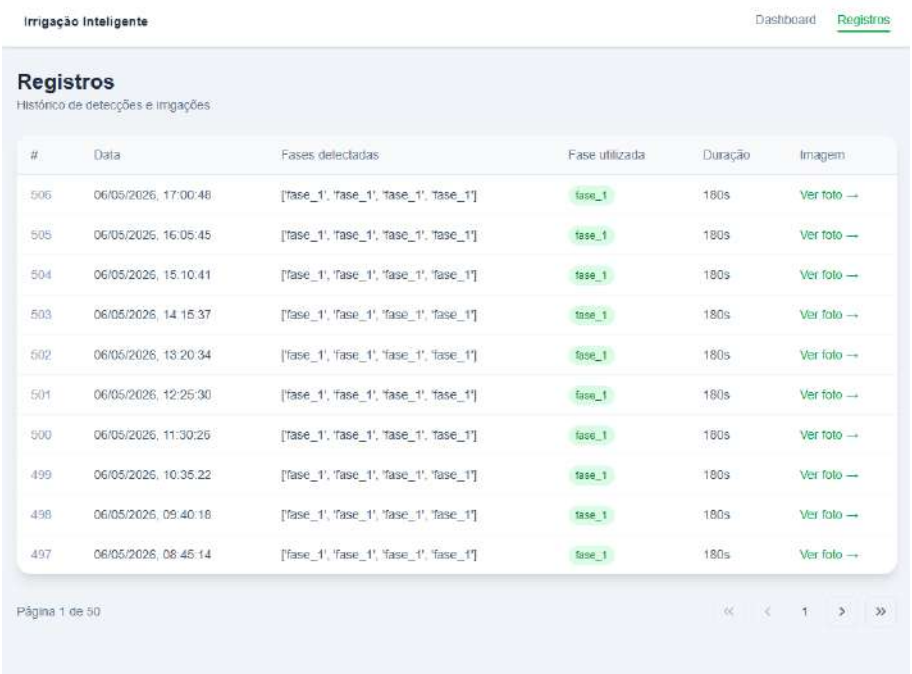
Essa interface possibilita ao usuário acessar os dados de maneira centralizada, sem a necessidade de interação direta com o ambiente do Supabase. O dashboard reúne, de forma visual e intuitiva, os principais status operacionais e os logs gerados pelo sistema, incluindo as imagens capturadas para o acompanhamento do crescimento do cultivo. As páginas de Status e de Logs do dashboard são apresentadas nas Figuras 3.28 e 3.29, respectivamente. Já a Figura 3.30 apresenta a interface utilizada para a visualização das imagens capturadas da bancada de cultivo.

Figura 3.28: Página de Status do Dashboard



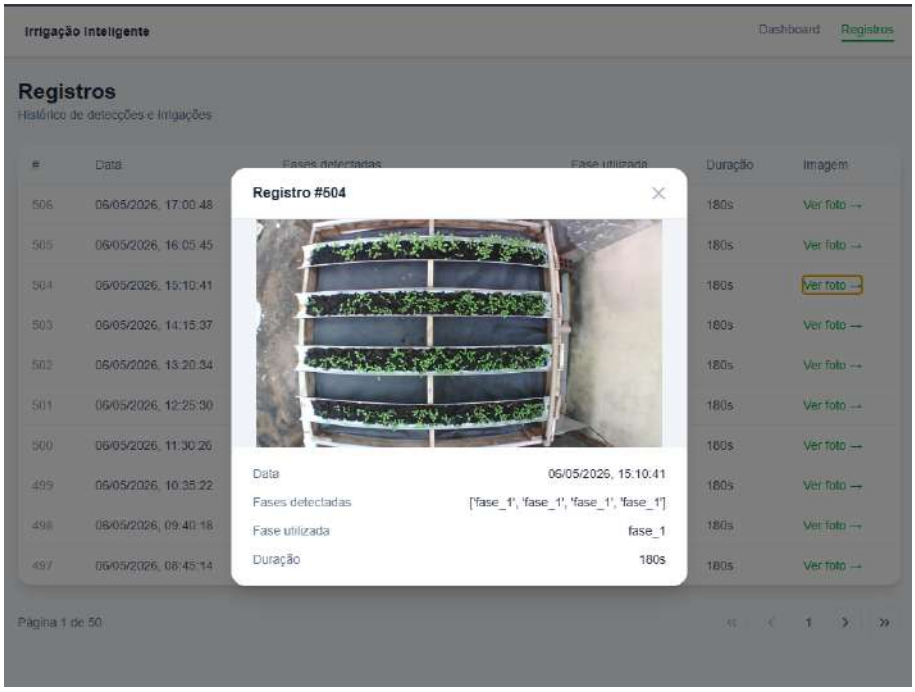
Fonte: Autoria própria (2026).

Figura 3.29: Página de Logs do Dashboard



Fonte: Autoria própria (2026).

Figura 3.30: Visualização de captura da bancada de cultivo



Fonte: Autoria própria (2026).

3.15 Procedimentos de Teste e Validação

Uma vez concluída a construção do protótipo, o sistema foi submetido a diferentes testes com o objetivo de garantir sua funcionalidade e identificar possíveis falhas ou melhorias a serem implementadas. Para uma análise mais organizada, essa etapa foi subdividida em dois níveis distintos: testes dos módulos individuais e testes do sistema integrado.

Os testes modulares foram planejados para validar os componentes essenciais do sistema. Seguindo a ordem definida pelo fluxo operacional do sistema, o primeiro módulo testado foi o responsável pela captura e envio das imagens da bancada de cultivo, seguido pelo módulo de processamento e, por fim, pelo módulo de controle de irrigação.

3.15.1 Teste do Módulo de Captura

No primeiro experimento, a ESP32-CAM foi posicionada em seu suporte definitivo e mantida em operação contínua durante um período de 12 h, das 07:00 às 19:00. Durante esse intervalo, comandos de captura e envio de imagens foram realizados de forma aleatória por meio de um script executado localmente em um notebook.

Esse procedimento teve como finalidade verificar o comportamento da câmera durante longos períodos de funcionamento, bem como avaliar a estabilidade da comunicação com o servidor HTTP configurado no dispositivo e a consistência das imagens capturadas ao

longo do período de funcionamento contínuo.

3.15.2 Teste do Módulo de Processamento

Para o módulo de processamento, foi elaborado um experimento destinado à análise do tempo de execução e do comportamento do modelo de visão computacional no formato ONNX Nano. O teste consistiu na execução, por meio do Raspberry Pi 3B, de um script em Python responsável por processar imagens aleatórias do conjunto de teste do *dataset* e registrar o tempo necessário para sua análise, assim como as fases identificadas em cada uma das calhas da bancada de cultivo.

Esse procedimento permitiu avaliar o desempenho do Raspberry Pi durante a execução do modelo de inferência, além de verificar a capacidade do sistema em realizar classificações em ambiente embarcado com recursos computacionais limitados.

3.15.3 Teste de Integração Parcial entre Captura e Processamento

Com o objetivo de examinar a estabilidade e a responsividade da comunicação entre os módulos de captura e processamento, foi realizado um novo cultivo para a execução de um teste de integração parcial. Utilizando o Raspberry Pi como elemento central do sistema, foi desenvolvido um script responsável pela realização de chamadas periódicas à ESP32-CAM por meio de requisições HTTP do tipo GET, permitindo capturar, tratar, processar, classificar e armazenar as imagens para validação posterior.

Esse teste teve duração equivalente a um ciclo completo de cultivo, aproximadamente 25 dias, com capturas realizadas a cada hora durante intervalos diários de 12 h, simulando as condições reais de operação do sistema. Além da comunicação entre os dispositivos, foram observados fatores como estabilidade do processamento contínuo, armazenamento das imagens capturadas no Supabase, possíveis falhas de sincronização entre os módulos e a acurácia do, até então, primeiro modelo de inferência.

3.15.4 Teste de Integração Parcial entre Processamento e Irrigação

O quarto experimento consistiu na verificação da comunicação entre o Raspberry Pi e o ESP32 responsável pelo módulo de controle de irrigação.

Para isso, foi executado um script no microcomputador capaz de estabelecer uma troca contínua de mensagens entre os dispositivos utilizando o protocolo MQTT. Após o estabelecimento da conexão, diferentes tempos de irrigação foram enviados ao ESP32 que, ao receber os comandos, retornava mensagens de heartbeat e realizava a ativação ou desativação do módulo relé, ainda desconectado da bomba hidráulica, de acordo com os

parâmetros recebidos.

Esse experimento teve como objetivo validar tanto a comunicação entre os módulos quanto o correto funcionamento dos ciclos de ativação e desativação do sistema de controle. Também foram observados o tempo de resposta entre envio e recebimento das mensagens e a estabilidade da comunicação durante execuções contínuas.

3.15.5 Teste de Recuperação de Falhas

Para avaliar a capacidade de recuperação do sistema diante de falhas comuns em ambientes reais, como perda de conectividade e interrupções no fornecimento de energia elétrica, foram simulados cenários de desconexão da rede e desligamento da alimentação elétrica individual e conjunta dos dispositivos.

O objetivo desse experimento foi verificar se o sistema seria capaz de retornar ao funcionamento normal após o restabelecimento das condições operacionais, assim como medir o tempo necessário para sua recuperação. Durante os testes, foram avaliados o comportamento do sistema após reconexão da rede, a retomada automática da comunicação MQTT e HTTP e a continuidade do funcionamento do sistema de irrigação.

3.15.6 Teste do Sistema Integrado

Por fim, os testes de sistema tiveram como objetivo validar a integração entre todos os componentes do protótipo e verificar seu funcionamento conjunto em ambiente real de cultivo.

Para isso, foi realizado um novo cultivo da variedade de coentro Verdão, utilizando o sistema completo em operação autônoma. O experimento teve duração aproximada de 25 dias, período durante o qual foram monitorados diariamente o controle adequado do tempo de irrigação, por meio de cronometragem manual, a correta classificação das fases de desenvolvimento das plantas com base nos logs de operação e o desenvolvimento saudável do cultivo.

Além disso, também foi validado o funcionamento do dashboard utilizado para acompanhamento remoto do sistema, comparando as informações exibidas em sua interface com os registros armazenados localmente no servidor Raspberry Pi.

Esse teste permitiu avaliar o comportamento geral do sistema em condições próximas às de uso real, verificando sua estabilidade operacional, sincronização entre módulos, acurácia das classificações do modelo mais recente e capacidade de funcionamento contínuo ao longo do ciclo de cultivo.

4 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados os resultados obtidos a partir do desenvolvimento e da implementação do protótipo proposto. Com base nos testes realizados em ambiente controlado de cultivo, foi possível avaliar o comportamento da solução sob diferentes perspectivas, abrangendo desde o desempenho do modelo de classificação até a estabilidade operacional do sistema ao longo dos ciclos experimentais. Dessa forma, buscou-se verificar em que medida a solução desenvolvida atende aos objetivos estabelecidos para este trabalho.

Para essa finalidade, a análise dos resultados foi estruturada com foco na resolução do problema central da pesquisa. Considerando que a automação proposta substitui o uso de temporizadores convencionais, responsáveis pela irrigação em intervalos fixos independentemente das condições da cultura, por um mecanismo de tomada de decisão baseado no estágio de desenvolvimento das plantas, o desempenho do sistema de irrigação passa a depender da confiabilidade do modelo responsável pela identificação das fases fenológicas.

Assim, os resultados foram organizados em duas etapas complementares. Inicialmente, é apresentada a avaliação da calibração e da precisão do modelo de visão computacional, empregado com o microcontrolador. Em seguida, são apresentados os resultados referentes ao desempenho eletromecânico, à estabilidade operacional e aos ganhos de eficiência do sistema de irrigação hidropônica integrado, obtidos durante sua operação em ambiente real de cultivo.

4.1 Desempenho do Modelo de Visão Computacional

Como destacado no Tópico 3.15, foram realizados dois testes principais com o objetivo de validar o desempenho e a acurácia do modelo de visão computacional implementado no sistema: o Teste de Integração Parcial entre os módulos de Captura e Processamento e o Teste do Sistema Integrado.

Durante o teste de integração parcial entre os módulos de captura e processamento, o módulo de processamento foi configurado com um script simplificado, responsável exclusivamente pelas etapas de captura, processamento e armazenamento dos dados. Buscando reproduzir as mesmas condições dos ciclos de cultivo realizados anteriormente, os resultados registrados nos logs de processamento foram comparados diariamente com o desenvolvimento real observado das plantas. Para auxiliar na validação visual, também foi considerada a contagem do tempo de cultivo.

Ao final de aproximadamente 25 dias de experimento, foram obtidos cerca de 291

registros de logs, contendo, em média, quatro classificações por imagem, correspondentes às calhas individuais de cultivo, totalizando 1.092 classificações. As predições geradas pelo modelo foram avaliadas manualmente e organizadas em uma matriz de confusão por classe, possibilitando o cálculo das métricas individuais e globais de desempenho. As métricas globais obtidas durante esse primeiro experimento são apresentadas na Tabela 4.1.

Tabela 4.1: Métricas globais do primeiro modelo

Métrica Global	Valor
Acurácia	73,63%
Precisão	73,99%
Recall	58,16%
F1-Score	57,97%

Fonte: Autoria própria (2026).

Inicialmente, as métricas globais indicam que o modelo apresentou acurácia de 73,63% e precisão média de 73,99%. Entretanto, ao analisar as métricas de *recall* e F1-Score, ambas próximas de 58%, observa-se que o modelo apresenta dificuldades em identificar corretamente todas as classes de forma equilibrada, ficando abaixo da meta de 80% esperada para essas métricas. Essa limitação torna-se mais evidente ao analisar as métricas individuais associadas a cada classe, apresentadas na Tabela 4.2.

Tabela 4.2: Métricas de classificação individual do primeiro modelo

Classe	Precisão	Recall	F1
Fase 1	71,03%	96,19%	81,70%
Fase 2	53,85%	4,83%	8,80%
Fase 3	97,10%	73,45%	83,40%

Fonte: Autoria própria (2026).

O primeiro ponto importante a ser destacado é que nem sempre as quatro calhas presentes na imagem foram detectadas e classificadas pela primeira versão do modelo de inferência, resultando em uma média de 3,75 calhas detectadas por imagem. Esse problema mostrou-se mais evidente na fase 3, contribuindo para sua taxa de *recall* mais baixa em comparação à fase 1, uma vez que essa métrica representa a capacidade do modelo de identificar corretamente os elementos pertencentes a determinada classe.

Em contrapartida, a fase 1 apresentou um alto *recall*, porém menor precisão. Isso ocorre devido à tendência do modelo em classificar incorretamente amostras da fase 2 como pertencentes à fase 1, aumentando a quantidade de falsos positivos dessa classe.

Por fim, a fase 2 apresentou o desempenho mais crítico entre todas as classes avaliadas. Conforme evidenciado pelo *recall* de apenas 4,83%, essa classe foi raramente identificada pelo modelo. Além disso, mesmo nas situações em que houve classificação como fase 2, observou-se baixa precisão, indicando alta taxa de classificações incorretas. Esses resultados mostraram a dificuldade do primeiro modelo em diferenciar adequadamente as características intermediárias do crescimento vegetativo em relação às demais fases do cultivo.

Com base nos resultados obtidos a partir dessas métricas, um novo modelo de inferência foi treinado utilizando as novas imagens coletadas, com o objetivo de aperfeiçoar principalmente a identificação da segunda fase de crescimento do coentro.

A partir disso, e após a finalização dos demais módulos do sistema, foi conduzido o Teste do Sistema Integrado, utilizando todos os módulos em funcionamento simultâneo. Para validação, o sistema foi submetido ao mesmo cenário e às mesmas condições dos cultivos anteriores, sendo realizado um novo ciclo experimental com duração aproximada de 25 dias. As métricas globais obtidas durante esse experimento são apresentadas na Tabela 4.3.

Tabela 4.3: Métricas globais do segundo modelo

Métrica Global	**Valor**
Acurácia	66,85%
Precisão	91,90%
Recall	66,01%
F1-Score	70,77%

Fonte: Autoria própria (2026).

As métricas globais calculadas demonstram que o novo modelo se tornou mais balanceado em seus resultados. A precisão, o *recall* e o score F1 apresentaram aumentos significativos. Entretanto, ao analisar a acurácia geral do modelo, observou-se uma queda de aproximadamente 6%, indicando que algumas classes se tornaram menos precisas em sua classificação. Esse comportamento é melhor compreendido a partir da análise individual das métricas de cada classe, apresentadas na Tabela 4.4.

Tabela 4.4: Métricas de classificação individual do segundo modelo

Classe	Precisão	Recall	F1-Score
Fase 1	94,55%	98,72%	96,59%
Fase 2	81,16%	76,71%	78,87%
Fase 3	98,01%	22,59%	36,85%

Fonte: Autoria própria (2026).

Comparando as métricas obtidas no teste anterior com as mais recentes, destaca-se a melhoria observada na fase 2, que apresentou aumento de 28% na precisão e 71% no *recall*. Esses resultados demonstram que o modelo passou a identificar essa classe com maior eficiência nas imagens, classificando-a corretamente e atingindo um dos principais objetivos do treinamento deste novo modelo. De forma semelhante, a fase 1 também apresentou evolução em seu desempenho, com aumento de 23% na precisão e 2% no *recall*.

Entretanto, de forma distinta do esperado, ao ser treinado e testado em ambiente real, o novo modelo apresentou queda no *recall* da fase 3, embora tenha mantido boa precisão. Diferentemente do comportamento anteriormente observado na fase 2, em que ocorria confusão com outras classes, neste caso a fase 3 deixou de ser identificada em parte das calhas presentes nas imagens, sendo simplesmente ignorada pelo modelo.

Ao utilizar novas imagens no treinamento do segundo modelo, o objetivo principal era ampliar sua capacidade de generalização, tornando-o mais robusto para identificar corretamente as classes mesmo com as distorções causadas pela câmera ESPCAM. Esse resultado foi alcançado para as fases 1 e 2. No entanto, a fase 3 apresentou sinais de overfitting durante esse teste, tornando-se menos generalizada para imagens diferentes daquelas utilizadas na etapa de treinamento. Esse comportamento não foi identificado durante a etapa de validação pós-treinamento, na qual apresentou resultados consistentes com o desempenho observado anteriormente.

Dessa forma, os resultados obtidos demonstram uma evolução significativa com a utilização do novo modelo. Embora a meta inicial de 80% tenha sido atingida apenas na métrica de precisão, que alcançou 91,90%, o objetivo estabelecido para o *recall* ainda não foi alcançado, evidenciando a necessidade de novos ajustes e aprimoramentos para aumentar a capacidade de detecção do modelo.

Entretanto, no contexto da automação do sistema de irrigação hidropônica desenvolvido neste trabalho, a elevada precisão mostrou-se o fator mais relevante para garantir a confiabilidade da tomada de decisão. Isso ocorre porque a ocorrência de falsos positivos pode resultar na classificação incorreta de uma fase fenológica mais avançada, ocasionando o acionamento da bomba por um período superior ao necessário e, consequentemente, o

fornecimento excessivo de água às plantas. Em contrapartida, uma eventual não detecção em um ciclo de inferência possui impacto reduzido, uma vez que as capturas e classificações são realizadas continuamente em intervalos regulares ao longo do dia, permitindo que a identificação da fase fenológica seja corrigida nas leituras subsequentes sem comprometer o desenvolvimento da cultura.

Assim, conclui-se que o modelo embarcado apresentou desempenho suficiente para atuar como subsistema de sensoriamento da solução proposta, atendendo aos requisitos necessários para o acionamento seguro do sistema de irrigação. Ainda assim, o aumento da métrica de *recall* permanece como uma oportunidade de aprimoramento, contribuindo para tornar o modelo mais robusto e ampliar sua confiabilidade em diferentes condições de operação, especialmente em aplicações futuras fora do ambiente experimental.

4.2 Resultados do Sistema Integrado

Para testar o funcionamento completo do sistema e validar a integração entre seus módulos, foi realizado um último cultivo experimental como etapa final de testes do protótipo.

Quanto à comunicação entre os módulos, o sistema apresentou boa estabilidade mesmo em momentos de lentidão da rede Wi-Fi local, mantendo o recebimento das capturas no módulo de processamento e o envio das atualizações de logs e status para o banco de dados do Supabase. Em testes controlados de desconexão da rede, todos os módulos demonstraram capacidade de reconexão conforme esperado, com tempos de recuperação variando em até aproximadamente 5 min para estabilização completa e retorno ao funcionamento normal, dependendo da qualidade da conexão Wi-Fi disponível.

Esses testes controlados também puderam ser observados em duas situações reais de indisponibilidade da conexão fornecida pelo provedor de internet. Embora ambas tenham ocorrido por curtos períodos, o sistema demonstrou capacidade de recuperar automaticamente sua operação após o restabelecimento da rede, sem necessidade de intervenção manual.

De forma semelhante, também foram realizados testes controlados de interrupção da alimentação elétrica, verificando a capacidade dos módulos de retomarem seu funcionamento automaticamente após a energização do sistema. Os módulos de captura e irrigação apresentaram reinicialização rápida, restabelecendo a conexão com a rede poucos instantes após o retorno da energia. Já o módulo de processamento apresentou tempo de recuperação superior, uma vez que depende da inicialização completa do sistema operacional do Raspberry Pi, do restabelecimento da conexão de rede e da execução automática do script responsável pelo funcionamento do sistema e do servidor local.

Essa tolerância a falhas também foi validada em uma situação real ocorrida durante o ciclo de cultivo, quando houve interrupção no fornecimento de energia elétrica no período entre o final da tarde e o início da noite. Nesse momento, o sistema já se encontrava próximo ao término da janela diária de irrigação, não sendo observados impactos relevantes na umidade do substrato ou no desenvolvimento das plantas. Após o restabelecimento da energia, o sistema reiniciou automaticamente e retornou ao funcionamento normal sem necessidade de intervenção manual.

De forma individual, cada módulo desempenhou corretamente sua função dentro da arquitetura proposta, não apresentando falhas operacionais relacionadas às suas funcionalidades específicas, mesmo após longos períodos de operação contínua e exposição ao ambiente externo. Esse comportamento pode ser observado, por exemplo, no módulo de captura, cuja disposição física é apresentada na Figura 3.19.

Uma observação importante relacionada ao ambiente de cultivo diz respeito à proteção da estrutura contra intempéries. Embora tenha sido construída uma cobertura em madeira com o objetivo de isolar o ambiente de fatores externos, como chuva e insetos, situações climáticas mais severas, especialmente durante períodos de chuvas intensas, ainda possibilitam a entrada parcial de umidade carregada pelo vento no interior da estrutura. Entretanto, mesmo diante dessas condições, a estrutura demonstrou ser capaz de resistir adequadamente, mantendo tanto a bancada de cultivo quanto os componentes eletrônicos protegidos e em funcionamento seguro ao longo dos testes.

Quanto ao cultivo, ao longo de todo o ciclo experimental, o desenvolvimento do coentro foi acompanhado diariamente, apresentando evolução compatível com a observada nos cultivos realizados anteriormente. Embora o novo modelo de inferência não tenha identificado a fase 3 em todas as calhas presentes nas imagens adquiridas pelo módulo de captura, devido ao seu *recall* abaixo do esperado, seu desempenho mostrou-se suficiente para que, na maior parte do tempo, o módulo de operação selecionasse tempos de irrigação compatíveis com essa etapa de desenvolvimento. Dessa forma, ainda que a vazão de irrigação definida para a fase de maturação tenha permanecido abaixo do ideal em determinados momentos, ela se mostrou adequada para evitar atrasos ou comprometimentos significativos no crescimento esperado das plantas.

Nesse contexto, os resultados obtidos permitem afirmar que o sistema integrado, de forma geral, cumpriu sua proposta inicial ao realizar capturas nítidas e padronizadas do cultivo hidropônico, além de executar o controle preciso dos intervalos de irrigação do sistema de gotejamento por meio do acionamento do módulo relé. Durante todo o período experimental, o sistema manteve estabilidade operacional e não apresentou falhas críticas de funcionamento, demonstrando robustez e confiabilidade suficientes para sua aplicação nas condições avaliadas.

4.3 Eficiência Operacional Alcançada com a Automatização

Uma das contribuições relevantes do sistema proposto está na redução do esforço operacional necessário para o acompanhamento do cultivo hidropônico. Antes da implementação do protótipo, o processo de cultivo demandava intervenções manuais diárias, exigindo a presença constante do operador para o acionamento da bomba de irrigação, verificação das condições de hidratação das plantas e monitoramento do nível do reservatório de solução nutritiva. Além disso, era necessário retornar ao ambiente de cultivo ao final do dia para desligar manualmente o sistema de irrigação. Dessa forma, a operação do cultivo exigia, no mínimo, duas visitas diárias à bancada, além do acompanhamento contínuo das condições das plantas ao longo do período de funcionamento.

Com a implementação do sistema proposto, essas tarefas foram automatizadas ou simplificadas. O acionamento e o desligamento da bomba passaram a ocorrer automaticamente dentro da janela de operação estabelecida entre 06:00 e 18:00, eliminando a necessidade de intervenção manual nesses momentos. A irrigação deixou de ser contínua, passando a funcionar em intervalos definidos com base na fase de crescimento identificada pelo modelo de visão computacional, evitando o funcionamento desnecessário da bomba ao longo do dia. Além disso, o sistema também passou a realizar capturas horárias da bancada de cultivo, classificar o estado das plantas e registrar os dados no Supabase, permitindo que o operador acompanhe remotamente o funcionamento do sistema por meio do dashboard, sem necessidade de deslocamento até o local de cultivo.

Dessa forma, a rotina de monitoramento presencial, que anteriormente precisava ser realizada diariamente, passou a ser necessária apenas semanalmente, permanecendo restrita à verificação de parâmetros que ainda exigem aferição manual, como o pH, a condutividade elétrica da solução nutritiva e o nível do reservatório. Esse resultado demonstra que, mesmo sem a utilização de instrumentação específica para a medição do consumo hídrico ou energético, foi possível reduzir de forma significativa o esforço operacional associado ao cultivo hidropônico, diminuindo a dependência da presença constante do operador e simplificando a operação do sistema.

Nesse contexto, considera-se que o problema que motivou o desenvolvimento deste trabalho, relacionado às limitações dos métodos manuais e temporizados de irrigação em sistemas hidropônicos, foi efetivamente abordado pela solução proposta. A inteligência artificial embarcada não foi empregada como um fim em si mesma, mas como o componente responsável por viabilizar a tomada de decisão baseada no estágio de desenvolvimento das plantas. Por meio dessa integração, foi possibilitada a operação autônoma do sistema de gotejamento durante 25 dias, com o fornecimento de água ajustado de acordo com as fases fenológicas do coentro. Dessa forma, foi demonstrado o potencial de aplicação de uma estratégia de irrigação automatizada e baseada em visão computacional para a redução da

necessidade de intervenção manual no cultivo hidropônico.

4.4 Resultados do Dashboard

Durante os testes do sistema integrado, o dashboard web desenvolvido permitiu o acompanhamento remoto do sistema de irrigação sem a necessidade de acesso direto ao Raspberry Pi ou às credenciais do Supabase. Além disso, a interface apresenta, em tempo real, informações sobre o estado de comunicação com o ESP32-S3 e o status de operação da bomba de irrigação, fornecendo ao operador um retorno visual imediato sobre o funcionamento do sistema.

Na página de status, foram exibidas corretamente informações referentes ao estado do ESP32-S3, estado da bomba, tempo de atividade do dispositivo e horário do último heartbeat recebido. As atualizações ocorreram continuamente por meio de websocket, refletindo de forma consistente as alterações registradas pelo módulo de controle de vazão. Durante os testes, o tempo médio de atualização das informações foi de aproximadamente 30 s, em conformidade com o intervalo de envio do heartbeat configurado no microcontrolador.

Já a página de registros apresentou o histórico completo das operações executadas pelo sistema, contendo, para cada entrada, a data e o horário da captura, as fases identificadas em cada calha de cultivo, a fase predominante utilizada para o controle da irrigação, o tempo de acionamento da bomba e o link para acesso à imagem armazenada no bucket do Supabase. Durante a validação do sistema integrado, verificou-se consistência entre os dados exibidos na interface e os registros armazenados localmente no Raspberry Pi, não sendo identificadas divergências entre as informações analisadas.

Além disso, o sistema permaneceu em operação contínua durante um ciclo completo de cultivo, com duração aproximada de 25 dias, sem ocorrência de falhas em seu funcionamento ou alimentação. Nesse período, foram mantidos de forma estável a comunicação entre os dispositivos, o funcionamento automatizado da irrigação e o registro contínuo e em tempo real das informações no dashboard.

4.5 Desafios Enfrentados no Desenvolvimento do Protótipo

Por se tratar de uma pesquisa de caráter exploratório, a ocorrência de testes e falhas é parte do processo de desenvolvimento. Ao longo da construção do protótipo atual, diversas modificações foram realizadas com o objetivo de atender aos requisitos propostos e aprimorar o desempenho do sistema.

Um dos principais desafios enfrentados foi a ausência de bases de dados públicas voltadas ao cultivo hidropônico de coentro, tornando necessária a criação de um *dataset*

próprio a partir de ciclos reais de cultivo. Além da coleta e anotação manual das imagens, também foram aplicadas técnicas de aumento de dados para ampliar a diversidade do conjunto utilizado no treinamento do modelo.

No que se refere especificamente ao controle de irrigação, foram desenvolvidas diferentes versões, baseadas em abordagens distintas para o controle e a medição da vazão destinada aos gotejadores. Inicialmente, partiu-se do princípio de restringir a passagem da solução nutritiva vinda da bomba submersa por meio de uma válvula solenóide, dispositivo que possui apenas as funções de abrir e fechar. Em conjunto com o uso de PWM, a proposta consistia em ajustar o nível de abertura da válvula para alcançar a vazão desejada. Associado a isso, o sensor de fluxo YF-S401 seria utilizado para medir a vazão, permitindo que o ESP32-S3 realizasse ajustes dinâmicos para manter o fluxo constante.

Para validação dessa abordagem, foi adquirida uma válvula solenóide de 12 V por meio do marketplace Mercado Livre. Embora trabalhos semelhantes indiquem a viabilidade dessa solução, o modelo testado demonstrou incapacidade de operar com sinais PWM, funcionando exclusivamente em dois estados: totalmente aberto ou totalmente fechado. Essa limitação inviabilizou sua aplicação no controle contínuo de vazão, levando ao descarte dessa abordagem ainda na fase inicial.

Diante desse cenário, buscou-se uma alternativa baseada em válvulas esfera motorizadas, capazes de proporcionar controle mais preciso da vazão por meio da variação do ângulo de abertura. Essa proposta foi motivada pela observação do sistema de irrigação já existente na bancada, no qual a regulagem manual do registro influenciava diretamente a frequência de gotejamento. Considerando o custo de soluções comerciais, optou-se pelo desenvolvimento de uma válvula motorizada de baixo custo.

A primeira versão consistiu na utilização de um registro esfera de PVC acoplado a um servomotor MG995. A conexão entre os componentes foi desenvolvida por meio de impressão 3D, utilizando um modelo disponibilizado pelo autor MarlonNW na plataforma Thingiverse. A Figura 4.1 apresenta as peças utilizadas na conexão entre o servomotor e o registro de esfera. Embora o protótipo tenha apresentado funcionamento inicial satisfatório, testes subsequentes evidenciaram a baixa resistência do material impresso, comprometendo sua confiabilidade para operação contínua.

Figura 4.1: Peças de impressão 3D utilizadas para válvula motorizada caseira



Fonte: Autoria própria (2025).

Com o objetivo de aumentar a robustez do sistema, uma segunda versão foi desenvolvida, substituindo o registro de PVC por um modelo em bronze, com menor atrito e maior resistência mecânica. Além disso, a estrutura de fixação foi reforçada com o uso de chapas de alumínio e parafusos. O resultado dessa nova versão da válvula é apresentado na Figura 4.2. Simultaneamente, foi realizada uma nova impressão 3D da versão inicial, com maior densidade de preenchimento e impressão mais lenta, permitindo a comparação entre as duas soluções

Figura 4.2: Válvula motorizada caseira de bronze



Fonte: Autoria própria (2025).

Nos testes realizados, verificou-se que o servomotor não foi capaz de acionar adequadamente a válvula de bronze, sendo danificado ainda no primeiro ensaio, possivelmente devido a falhas no acoplamento mecânico ou à exigência de torque superior à sua capacidade. Em contrapartida, a versão impressa em 3D, ilustrada na Figura 4.3, apresentou funcionamento satisfatório, permitindo o acionamento da válvula com menor esforço mecânico.

Figura 4.3: Nova válvula motorizada com impressão 3D

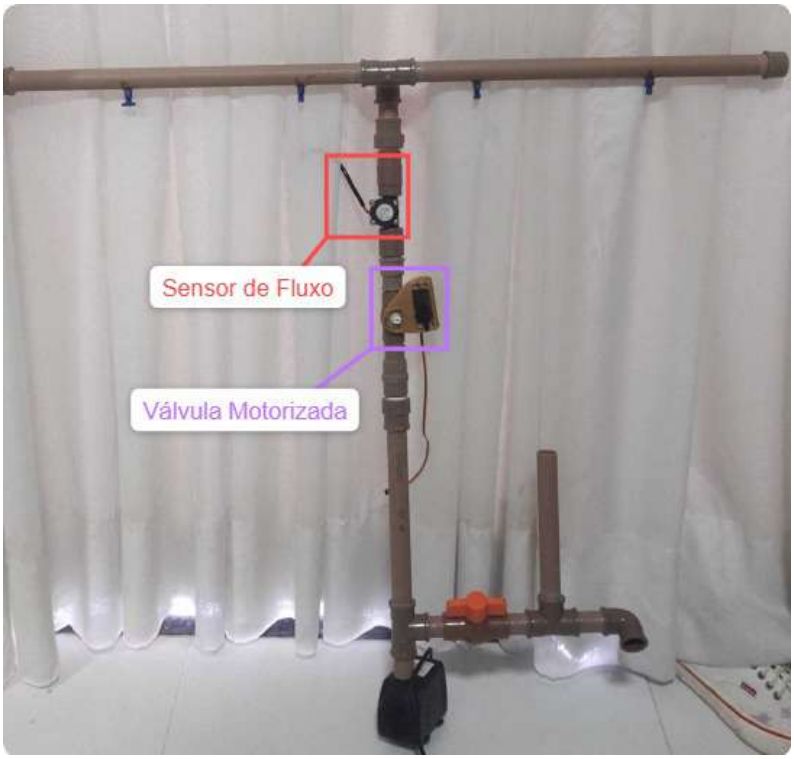


Fonte: Autoria própria (2025).

Na etapa seguinte, foi avaliada a capacidade da válvula impressa em 3D de controlar a vazão sob condições reais de operação, com a bomba de água conectada diretamente ao sistema. Inicialmente, o desempenho foi considerado adequado; no entanto, após ciclos repetidos de uso, observou-se aumento da rigidez do registro, dificultando o acionamento pelo servomotor e comprometendo a estabilidade do sistema.

Paralelamente ao desenvolvimento das válvulas, foram implementadas diferentes configurações do sistema de irrigação com o objetivo de medir a vazão efetivamente entregue aos gotejadores. Em uma segunda versão do sistema, propôs-se a utilização de um sensor de fluxo YF-S401 posicionado antes dos gotejadores, visando a medição direta da vazão e o ajuste em tempo real por meio do ESP32-S3, como apresentado pela Figura 4.4. Contudo, testes experimentais demonstraram que, devido à baixa vazão característica dos gotejadores, o sensor não apresentou sensibilidade suficiente para fornecer leituras confiáveis, levando ao descarte dessa configuração.

Figura 4.4: Sistema de irrigação com sensor e válvula motorizada



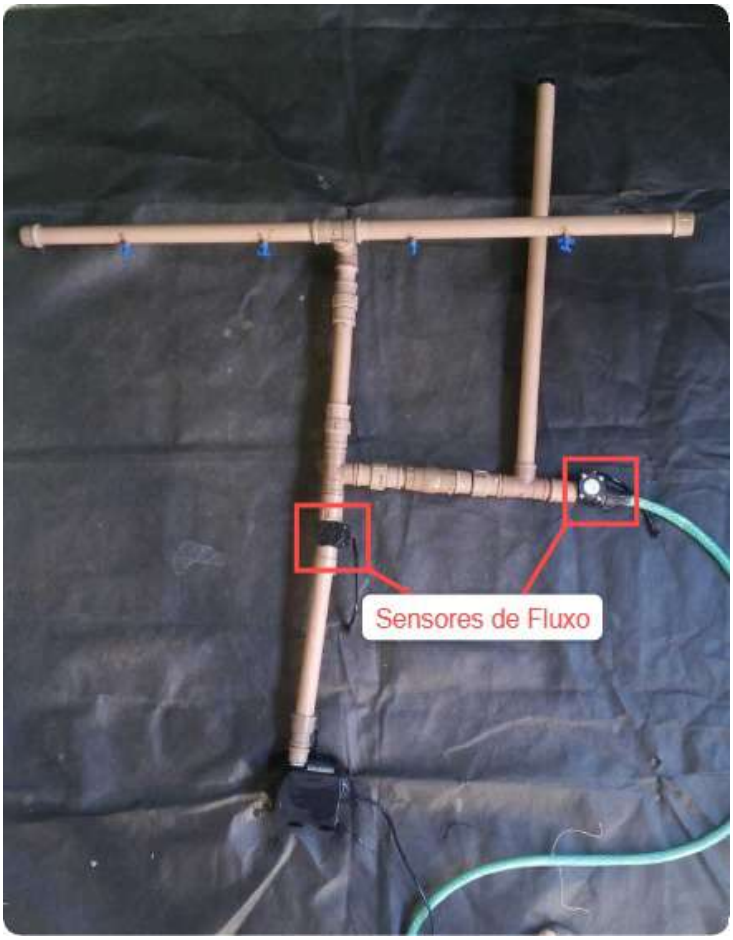
Fonte: Autoria própria (2025).

Na tentativa de contornar essa limitação, foi proposta uma terceira abordagem baseada no uso de dois sensores de fluxo: um instalado na saída da bomba, responsável pela medição da vazão total, e outro posicionado no retorno do extravasor, permitindo a estimativa da vazão destinada aos gotejadores por meio da diferença entre as medições. Nessa configuração, a válvula motorizada foi posicionada entre o primeiro sensor e o extravasor, com o objetivo de controlar a vazão ao longo do sistema.

Para viabilizar essa abordagem, foi desenvolvida uma nova estrutura hidráulica. No entanto, devido às falhas apresentadas pelas válvulas motorizadas nas etapas anteriores, optou-se pela substituição desse componente por um registro convencional durante a realização dos testes experimentais. A configuração hidráulica adotada é apresentada na Figura 4.5.

Apesar da adequação estrutural, os resultados obtidos não apresentaram confiabilidade satisfatória. Observou-se que os erros de medição associados a cada sensor se propagavam no cálculo da diferença entre as vazões, resultando em valores inconsistentes quando comparados às medições empíricas da vazão diretamente nos gotejadores.

Figura 4.5: Sistema de irrigação com sensores de fluxo



Fonte: Autoria própria (2026).

Diante das limitações observadas tanto no controle quanto na medição da vazão, optou-se por abandonar as abordagens baseadas em controle ativo contínuo. Considerando restrições de custo e viabilidade, adotou-se uma solução alternativa baseada no uso de um módulo relé para controle temporal da irrigação, associada à estimativa empírica da vazão dos gotejadores. Essa abordagem, embora mais simples, apresentou maior confiabilidade dentro do contexto do projeto.

Também foram enfrentadas limitações relacionadas ao hardware de processamento. Inicialmente, o sistema seria executado em um BeagleBone Black, porém o dispositivo não suportou adequadamente a instalação das bibliotecas necessárias para execução dos modelos de visão computacional. Em função disso, o Raspberry Pi 3B foi adotado como plataforma principal, atendendo satisfatoriamente aos requisitos de processamento e integração do sistema.

5 CONSIDERAÇÕES FINAIS

Foi apresentado, neste trabalho, o desenvolvimento de um sistema de irrigação automatizada para cultivo hidropônico de coentro (*Coriandrum sativum*), integrando visão computacional, inteligência artificial e sistemas embarcados com o propósito de automatizar o fornecimento de solução nutritiva de acordo com o estágio de crescimento das plantas. A proposta buscou oferecer uma alternativa de baixo custo e acessível para pequenos e médios produtores, contribuindo para a otimização do uso de recursos hídricos e para o aumento da eficiência no cultivo hidropônico.

Ao longo do desenvolvimento, foram aplicados conceitos relacionados à visão computacional, aprendizado de máquina, comunicação entre dispositivos embarcados e computação em nuvem, resultando em um sistema integrado capaz de monitorar, processar e disponibilizar informações em tempo real. Além disso, a realização de testes experimentais em ambiente controlado permitiu avaliar tanto o desempenho do modelo de classificação quanto a funcionalidade geral do sistema de irrigação.

5.1 Avaliação do Sistema de Irrigação Automatizado

O protótipo desenvolvido demonstrou a viabilidade da integração entre hardware e software para aplicações agrícolas inteligentes. A arquitetura baseada em Raspberry Pi 3B, ESP32-S3, ESP32-CAM e módulo relé possibilitou o monitoramento e acionamento automatizado da irrigação de maneira estável e funcional. O uso do Supabase como plataforma de armazenamento e sincronização de dados permitiu o gerenciamento das informações operacionais e das imagens coletadas em tempo real, enquanto o dashboard web desenvolvido em Next.js forneceu uma interface acessível para acompanhamento remoto do sistema.

Embora o modelo treinado não tenha apresentado o desempenho inicialmente desejado, atingindo apenas parcialmente as metas de precisão e recall, a utilização do modelo YOLOv8 convertido para o formato ONNX mostrou-se adequada ao ambiente embarcado. O modelo foi capaz de realizar inferências com tempo médio de processamento próximo de um segundo por imagem, evidenciando seu potencial para futuras melhorias e aplicação prática.

Além disso, a comunicação entre os dispositivos, realizada por meio dos protocolos HTTP e MQTT, apresentou funcionamento confiável durante os testes, garantindo a troca contínua de informações entre os módulos do sistema. Da mesma forma, a implementação de atualizações em tempo real por meio de API e websockets, integrando o módulo de

processamento, o banco de dados Supabase e o dashboard web, permitiu ao operador acompanhar continuamente os registros de classificação, o estado do sistema e as imagens capturadas ao longo do cultivo, proporcionando maior controle operacional e supervisão do processo automatizado.

5.2 Lições Aprendidas

O desenvolvimento deste trabalho proporcionou experiência prática em diferentes áreas da computação e da automação, especialmente em sistemas embarcados, visão computacional e inteligência artificial. A implementação do protótipo permitiu aprofundar conhecimentos relacionados ao treinamento de modelos, integração entre dispositivos eletrônicos e comunicação em tempo real entre módulos distribuídos.

Além disso, tornou-se evidente a relevância da integração entre tecnologias distintas para o desenvolvimento de sistemas automatizados mais completos e eficientes. A experiência também demonstrou que a construção de um *dataset* de qualidade é uma etapa crítica no desenvolvimento de soluções baseadas em aprendizado de máquina, demandando planejamento cuidadoso da coleta, padronização das imagens e consistência na anotação dos dados.

Outro aspecto importante observado foi a necessidade de projetar sistemas embarcados com mecanismos de tolerância a falhas, garantindo a continuidade da operação mesmo em situações de perda de conectividade ou falhas temporárias de comunicação, preservando assim a segurança do cultivo.

Por fim, ressalta-se que os testes realizados neste trabalho foram restritos à validação técnica do protótipo em ambiente controlado, não sendo conduzidas avaliações com agricultores ou outros usuários finais. Essa abordagem foi adotada em razão do escopo da pesquisa, que teve como foco o desenvolvimento e a validação do funcionamento do sistema embarcado e de seus componentes. Embora os resultados obtidos tenham demonstrado a viabilidade técnica da solução proposta, a participação de usuários externos poderá fornecer informações relevantes sobre aspectos relacionados à usabilidade, à adequação às necessidades práticas do cultivo e à aceitação da tecnologia. Dessa forma, a realização de avaliações com o público-alvo é identificada como uma etapa importante para a consolidação e o aperfeiçoamento de soluções voltadas à agricultura de precisão.

5.3 Trabalhos Futuros

Como possibilidades de continuidade deste trabalho, destaca-se inicialmente a expansão do *dataset* com novos ciclos de cultivo e diferentes condições ambientais, como variações

de iluminação e posicionamento da câmera, visando aumentar a robustez e a capacidade de generalização do modelo de classificação.

Também se mostra relevante investigar modelos mais recentes de visão computacional e técnicas de otimização, como quantização e compressão de modelos, buscando reduzir o tempo de inferência e ampliar o desempenho do sistema em hardware embarcado.

Outra perspectiva importante consiste no desenvolvimento de um mecanismo mais preciso de controle de vazão da solução nutritiva, utilizando sensores de fluxo mais sensíveis ou válvulas motorizadas comerciais, permitindo um controle mais refinado da irrigação de acordo com as necessidades específicas de cada fase fenológica da planta.

Além disso, futuras pesquisas podem explorar o treinamento de modelos de visão computacional voltados à identificação automática de anomalias no cultivo, como doenças nas plantas, sinais de desidratação e indícios de sobreirrigamento. A incorporação dessas funcionalidades permitiria que o sistema atuasse não apenas na automação da irrigação, mas também no monitoramento inteligente da saúde das plantas, auxiliando na tomada de decisão e na prevenção de perdas durante o cultivo.

Outra possibilidade relevante envolve a ampliação da automação relacionada à solução nutritiva utilizada no sistema hidropônico. Nesse contexto, o protótipo pode evoluir para realizar o monitoramento contínuo de parâmetros como pH, condutividade elétrica e nível do reservatório, executando automaticamente a correção desses parâmetros por meio da dosagem controlada de nutrientes, reguladores de pH e reposição de água sempre que necessário.

Também se pretende tornar o sistema mais escalável, permitindo que a plataforma seja capaz de monitorar e gerenciar múltiplos sistemas hidropônicos simultaneamente, incluindo diferentes bancadas e calhas de cultivo utilizando o protótipo proposto. Essa expansão possibilitaria a aplicação da solução em ambientes de produção de maior porte.

Além disso, futuras versões do dashboard podem incorporar mecanismos de notificações e alertas automáticos para identificação de falhas no sistema, como perda de comunicação com microcontroladores, falhas na irrigação ou inconsistências na correção automática dos parâmetros da solução nutritiva. Dessa forma, o operador poderia agir rapidamente diante de possíveis problemas operacionais.

Por fim, a integração dessas funcionalidades pode tornar o sistema mais autônomo, preciso e aplicável a diferentes cenários de produção hidropônica, contribuindo para a redução da intervenção manual, maior eficiência no uso de recursos e melhoria da qualidade do cultivo agrícola.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, L. *Kano: computador e kit de programação*. 2025. Acesso em: 21 set. 2025. Disponível em: <https://medium.com/@lciaabreu/kano-computador-e-kit-de-programa%C3%A7%C3%A3o-2776a7e386d0>.

ALICE, A. *HTTP: um guia completo sobre o que é e como funciona o protocolo da web*. Alura, 2024. Acesso em: 21 set. 2025. Disponível em: <https://www.alura.com.br/artigos/http>.

ALVES, G. *Detecção de Objetos com YOLO – Uma abordagem moderna*. IA Expert Academy, 2020. Acesso em: 20 mai. 2025. Disponível em: <https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/>.

ANCHESQUI, L. D. *Desenvolvimento de um sistema inteligente de irrigação baseado em IoT para agricultura de precisão*. Dissertação (Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica)) — Instituto Federal do Espírito Santo (IFES), Vitória, ES, 2023. Acesso em: 28 mai. 2026. Disponível em: <https://repositorio.ifes.edu.br/server/api/core/bitstreams/c2ae4bd4-75e6-444c-bf87-63bdde5969cb/content>.

AWS. *O que é MQTT?* Amazon Web Services, 2025. Acesso em: 28 set. 2025. Disponível em: <https://aws.amazon.com/pt/what-is/mqtt/>.

AYEZABU, A. Z. Supabase vs firebase: Evaluation of performance and development of progressive web apps. 2022. Acesso em: 03 mar. 2026. Disponível em: <https://medium.com/@babyzewdie/%supabase-vs-firebase-evaluation-of-performance-and-development-of-progressive-web-apps-%c43766c40ccb>.

BEOCK, R. P. et al. *Protocolo HTTP*. Colíder: Universidade do Estado de Mato Grosso – UNEMAT, n.d. Acesso em: 22 set. 2025. Disponível em: <https://www.vivaolinux.com.br/imagens/artigos/comunidade/Protocolo%20HTTP.pdf>.

Bosch. *Sensores: a relevância da tecnologia para um mundo conectado*. Notícias e Histórias, 2021. Acesso em: 27 set. 2025. Disponível em: <https://www.bosch.com.br/noticias-e-historias/aiot/sensores-em-um-mundo-conectado/>.

BRITO, A. da S.; BRITO, R. A. de C. et al. Sustentabilidade hídrica no agronegócio. *Revista Tópicos*, v. 3, n. 24, p. 1–12, 2025. Disponível em: <https://revistatopicos.com.br/artigos/sustentabilidade-hidrica-no-agronegocio>.

BROWNLEE, J. *When to Use MLP, CNN, and RNN Neural Networks*. Machine Learning Mastery, 2022. Acesso em: 19 jul. 2026. Disponível em: <https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>.

CALIXTO, G. M. *Computação em nuvem e tecnologias emergentes*. Editora Senac São Paulo, 2024. Acesso em: 06 mar. 2026. Disponível em: <https://books.google.com/books?id=9E0hEQAAQBAJ>.

CANDIDO, G. *Aprendizado supervisionado x não supervisionado*. Medium – Data Hackers, 2023. Acesso em: 13 jun. 2025. Disponível em: <https://medium.com/data-hackers/aprendizado-supervisionado-x-n%C3%A3o-supervisionado-ca79b522659d>.

CARDOSO, J. *Metodologia da Pesquisa*. [S. l.]: Pró-Reitoria de Pós-Graduação – PRPG, 2003. Acesso em: 14 jun. 2025. Disponível em: <https://www.academia.edu/download/34909124/MetodologiaPesquisa-Moresi2003.pdf>.

CARDOSO, M. *O Que É Um Microcontrolador?* IEEE RAS UFCG, 2020. Acesso em: 20 set. 2025. Disponível em: <https://edu.ieee.org/br-ufcgras/o-que-e-um-microcontrolador/>.

CARRIJO, O. A.; MAKISHIMA, N. *Princípios de hidroponia*. Brasília, DF: Embrapa Hortaliças, 2000. Acesso em: 13 jun. 2025. Disponível em: <https://www.embrapa.br/busca-de-publicacoes/-/publicacao/769981/principios-de-hidroponia>.

CERQUEIRA, F. B. et al. Doses de nitrogênio nas respostas morfológicas de coentro. *Global Science and Technology*, Rio Verde, v. 9, n. 1, p. 15–21, 2016. Acesso em: 15 jun. 2025. Disponível em: <https://www.researchgate.net/publication/312080744_Doses_de_Nitrogenio_nas_Respostas_Morfofisiologicas_de_Coentro_Coriandrum_sativum_L>.

CHASE, O.; ALMEIDA, F. Sistemas embarcados. *Mídia Eletrônica*, v. 10, n. 11, p. 3, 2007. Acesso em: 05 mar. 2026. Disponível em: <https://www.maxpezzin.com.br/aulas/6_EAC_Sistemas_Embarcados/1_SE_Introducao.pdf>.

CHEN, M. *O que é Machine Learning?* Oracle Brasil, 2024. Acesso em: 23 jun. 2025. Disponível em: <https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/>.

COELHO, B. *Tipos de pesquisa: Guia Completo para Classificação e Aplicação*. Blog da Mettzer, 2019. Acesso em: 14 jun. 2025. Disponível em: <https://blog.mettzer.com/tipos-de-pesquisa/>.

COPELAND, B. J. *Artificial intelligence*. Encyclopaedia Britannica, 2022. Acesso em: 23 jun. 2025. Disponível em: <https://www.britannica.com/technology/artificial-intelligence>.

COSTA, P. C. et al. Condutividade elétrica da solução nutritiva e produção de alface em hidroponia. *Scientia Agricola*, Piracicaba, v. 58, n. 3, p. 595–597, 2001. Acesso em: 21 set. 2025. Disponível em: <https://www.scielo.br/j/sa/a/kBJNL9GdzX8kF6yYqzSC9Hb/?format=html&lang=pt>.

CRAVO, E. *Atuadores: conheça todos os tipos e para que servem*. 2024. Acesso em: 05 mar. 2026. Disponível em: <https://blog.kalatec.com.br/tipos-de-atuadores-industriais/>.

DNC. *O que é Protocolo HTTP e Métodos de Requisição em APIs*. Blog Escola DNC, 2023. Acesso em: 27 set. 2025. Disponível em: <https://www.escoladnc.com.br/blog/o-que-e-protocolo-http-e-metodos-de-requisicao-em-apis>.

DNOCS. *Conheça o cultivo hidropônico: o plantio sem o uso do solo*. 2023. Acesso em: 17 maio 2025. Disponível em: <https://www.gov.br/dnocs/pt-br/assuntos/vem-conhecer/conheca-o-cultivo-hidroponico-o-plantio-sem-o-uso-do-solo>.

DUNN, B.; SINGH, H. *Electrical Conductivity and pH Guide for Hydroponics*. 2017. Acesso em: 21 set. 2025. Disponível em: <https://extension.okstate.edu/fact-sheets/electrical-conductivity-and-ph-guide-for-hydroponics.html>.

Embarcados. *ESP32-S3-DevKitC-1 v1.1*. 2025. Acesso em: 21 set. 2025. Disponível em: <https://embarcados.com.br/placa/esp32-s3-devkitc-1-v1-1/>.

Equipe Nepin. *Atuadores – O que são e para o que servem*. 2025. Acesso em: 05 mar. 2026. Disponível em: <https://www.nepin.com.br/blog/solucoes-industriais/atuadores-o-que-sao-e-para-o-que-servem/>.

FLORAFLEX. *The Hydroponic Drip System Explained*. 2020. Acesso em: 23 jun. 2025. Disponível em: <https://floraflex.com/default/blog/post/the-hydroponic-drip-system-explained?srsltid=AfmBOorQXGzdagpYo_yea1n946Nk5C__jDMrXv6USo6Tt-F3_Mk8KGjd>.

FONTELLES, M. J. et al. Metodologia da pesquisa científica: diretrizes para a elaboração de um protocolo de pesquisa. *Revista Paraense de Medicina*, 2009. Acesso em: 14 jun. 2025. Disponível em: <https://pesquisa.bvsalud.org/gim/resource/en,au:%22Martins%20Neto,%20Viviana%22/lil-588477>.

GIFFONI, A. *Entenda por que o coentro é um caso de amor e ódio*. 2019. Hospital IGESP. Acesso em: 21 set. 2025. Disponível em: <https://hospitaligesp.com.br/entenda-por-que-o-coentro-e-um-caso-de-amor-e-odio/>.

GOKHALE, P.; BHAT, O.; BHAT, S. Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, v. 5, n. 1, p. 41–44, 2018. Acesso em: 05 mar. 2026. Disponível em: <https://www.researchgate.net/publication/330114646_Introduction_to_IOT>.

GONZAGA, A.; NEVEZ, L. A. P.; NETO, H. V. *Avanços em visão computacional*. Omnipax Editora, 2012. Acesso em: 20 mai. 2025. Disponível em: <https://www.academia.edu/47374702/Avan%C3%A7os_em_vis%C3%A3o_computacional>.

HAWKINS, M.; MEYER, J.; SUMMERSBY, A. *Image sensors explained*. Infobank, n.d. Acesso em: 27 set. 2025. Disponível em: <https://www.canon-europe.com/pro/infobank/image-sensors-explained/>.

HIDROGOOD. *O que é Condutividade Elétrica e a sua importância no cultivo hidropônico*. 2019. Acesso em: 21 set. 2025. Disponível em: <https://hidrogood.com.br/artigos/fertilizantes/o-que-e-condutividade-eletrica-e-a-sua-importancia-no-cultivo-hidroponico/>.

HORTICULTOR, C. do. *Coentros Muqueca e Verdão: sucesso em diferentes métodos de cultivo*. 2023. Acesso em: 21 set. 2025. Disponível em: <https://canaldohorticultor.com.br/coentros-muqueca-e-verdao-sucesso-em-diferentes-metodos-de-cultivo/>.

Hydroplanner. *Hydroponics Drip System*. 2025. Acesso em: 23 jun. 2025. Disponível em: <https://hydroplanner.com/blog/hydroponics-drip-system-EN>.

IBM. *Protocolos TCP/IP*. IBM Documentation: AIX 7.3.0, 2021. Acesso em: 27 set. 2025. Disponível em: <https://www.ibm.com/docs/pt-br/aix/7.3.0?topic=protocol-tcpip-protocols>.

IBM. *O que são redes neurais convolucionais?* IBM, 2025. Acesso em: 19 jul. 2026. Disponível em: <https://www.ibm.com/br-pt/think/topics/convolutional-neural-networks>.

INÁCIO, M. J. Sensores e atuadores. *Faculdade de Ciências do Tocantins*, 2009. Acesso em: 05 mar. 2026.

KAVLAKOGLU, E.; STRYKER, C. *Inteligência Artificial*. IBM, 2024. Acesso em: 20 mai. 2025. Disponível em: <https://www.ibm.com/br-pt/think/topics/artificial-intelligence>.

KELLER, A. L. *Internet das Coisas aplicada à Indústria: Dispositivo para Interoperabilidade de Redes Industriais*. Dissertação (Dissertação de Mestrado) — Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS, 2016. Acesso em: 29 mai. 2026. Disponível em: <https://repositorio.jesuita.org.br/handle/UNISINOS/6233>.

KERSCHBAUMER, R. *Apostila Microcontroladores*. Luzerna, SC: Instituto Federal Catarinense – Campus Luzerna, 2018. Acesso em: 20 set. 2025. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2018/02/Apostila-Microcontroladores.pdf>.

KOHMANN, C. Rede sem fio de sensores e atuadores para uso agrícola. 2008. Acesso em: 05 mar. 2026. Disponível em: <https://lume.ufrgs.br/handle/10183/282985>.

KROMYDAS, B. *Convolutional Neural Network (CNN): A Complete Guide*. LearnOpenCV, 2023. Acesso em: 19 jul. 2026. Disponível em: <https://learnopencv.com/understanding-convolutional-neural-networks-cnn/>.

LACERDA, L. M. *Redes Neurais Convolucionais para Processamento de Imagens*. 62 p. Trabalho de Conclusão de Curso — Universidade Federal Fluminense, Niterói, RJ, jan. 2025. Acesso em: 19 jul. 2026. Disponível em: <https://estatistica.uff.br/wp-content/uploads/sites/33/2025/04/119054020BIB-Luana-Moreno-Lacerda.pdf>.

LUZ, G. L. da et al. Ação dos intervalos entre irrigações no sistema radicular da alface em hidroponia. *Interciencia*, v. 42, n. 6, p. 370–374, 2017. Acesso em: 29 mai. 2026. Disponível em: <https://www.redalyc.org/pdf/339/33951621006.pdf>.

MATTEDE, H. *O que são sensores e quais as suas aplicações*. Mundo da Elétrica, 2018. Acesso em: 27 set. 2025. Disponível em: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>.

MCCANDLESS, J. G. *6 Types of Hydroponic Systems That Every Hobbyist Should Know*. 2022. Acesso em: 23 jun. 2025. Disponível em: <https://ponicslife.com/6-types-of-hydroponic-systems-that-every-hobbyist-should-know/>.

MDN. *Visão geral do HTTP*. MDN Web Docs, 2024. Acesso em: 27 set. 2025. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Guides/Overview>.

MELLO, M. *O que é Raspberry Pi, para que serve e principais modelos no Brasil*. Victor Vision, 2022. Acesso em: 21 set. 2025. Disponível em: <https://victorvision.com.br/blog/o-que-e-raspberry-pi/>.

MELO, R. A. et al. Variabilidade genética em progênies de meios-irmãos de coentro. *Horticultura Brasileira*, Brasília, v. 27, n. 3, p. 325–329, 2009. Acesso em: 15 jun. 2025. Disponível em: <https://www.scielo.br/j/hb/a/LMtFQVmd5vLcsPT49gF4Ljp/?lang=pt>.

Microsoft. *ONNX Runtime: Cross-platform, high performance ML inferencing*. Microsoft, 2024. Acesso em: 05 mar. 2026. Disponível em: <https://onnxruntime.ai>.

ONNX Community. *ONNX: Open Neural Network Exchange*. ONNX Community, 2024. Acesso em: 05 mar. 2026. Disponível em: <https://onnx.ai>.

Raspberry Pi Ltd. *Raspberry Pi 3 Model B, Version 1.2 Schematic (Reduced)*. 2016. Acesso em: 30 mai. 2026. Disponível em: <https://github.com/SayanSeth/Raspberry-Pi-Documentation/blob/master/hardware/raspberrypi/schematics/RPI-3B-V1_2-SCHEMATIC-REDUCED.pdf>.

RODRIGUES, L. D. et al. Controle de temperatura e de umidade de uma composteira utilizando o esp32. In: *Anais do Simpósio Brasileiro de Agroinformática (SBIAGRO)*. [s.n.], 2021. Acesso em: 28 mai. 2026. Disponível em: <https://sol.sbc.org.br/index.php/sbiagro/article/view/18375>.

RODRIGUES, R. *O primo rico do Arduino: conheça o ESP32*. 3E Unicamp, 2025. Acesso em: 21 set. 2025. Disponível em: <https://www.3eunicamp.com/post/o-primo-rico-do-arduino-conhe%C3%A7a-o-esp32>.

SANTOS, R.; SANTOS, S. *ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained*. 2020. Acesso em: 30 mai. 2026. Disponível em: <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>.

SCHNEIDER, J.; SMALLEY, I. *O que é um microcontrolador?* IBM Think, 2024. Acesso em: 20 set. 2025. Disponível em: <https://www.ibm.com/br-pt/think/topics/microcontroller>.

SHIMABUKURO, I.; TOLEDO, V. *O que é o Raspberry Pi? Saiba para que serve e como funciona o minicomputador*. Tecnoblog, 2025. Acesso em: 21 set. 2025. Disponível em: <https://tecnoblog.net/responde/o-que-e-o-raspberry-pi/>.

SILVA, L. M. d. et al. Avaliação fenológica do coentro (*Coriandrum sativum L.*) em diferentes condições de luminosidade. *Diversitas Journal*, v. 5, n. 3, p. 1729–1737, 2020. Acesso em: 15 jun. 2025. Disponível em: <https://diversitasjournal.com.br/diversitas_journal/article/view/1123>.

SILVA, M. G. d. *Coentro hidropônico sob diferentes condições de cultivo relacionadas à solução nutritiva: temperatura, salinidade e recirculação*. Tese (Tese de Doutorado) — Universidade Federal do Recôncavo da Bahia, Cruz das Almas, Bahia, 2019. Acesso em: 10 mar. 2026. Disponível em: <https://www.ufrb.edu.br/pgea/images/Teses/MAIRTON_GOMES_DA_SILVA.pdf>.

SILVEIRA, D. T.; CÓRDOVA, F. P. A pesquisa científica. In: *Métodos de pesquisa*. Porto Alegre: Editora da UFRGS, 2009. p. 33–44. Acesso em: 14 jun. 2025. Disponível em: <https://lume.ufrgs.br/bitstream/handle/10183/213838/000728731.pdf>.

SIQUEIRA, V. C. A. Análise comparativa de ferramentas low-code para desenvolvimento de software. 2024. Acesso em: 03 mar. 2026. Disponível em: <https://repositorio.ufc.br/bitstream/riufc/78836/1/2024_tcc_vcasiqueira.pdf>.

SUSNJARA, S.; SMALLEY, I. *O que é computação em nuvem (cloud computing)?* s.d. Acesso em: 06 mar. 2026. Disponível em: <https://www.ibm.com/br-pt/think/topics/cloud-computing>.

TREES. *Hydroponic Drip System*. 2022. Acesso em: 23 jun. 2025. Disponível em: <https://www.trees.com/gardening-and-landscaping/hydroponic-drip-system>.

TSHANAKAS, G. *Understanding the difference between Cotyledon and True Leaves*. 2019. Gardening 4 Joy. Acesso em: 21 set. 2025. Disponível em: <https://www.gardening4joy.com/understanding-the-difference-between-cotyledon-and-true-leaves/>.

UNIQTECH. *Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning*. Medium, 2018. Acesso em: 19 jul. 2026. Disponível em: <https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1>.

VELOSO, E. M. C. Agricultura de precisão: sistema para monitoramento remoto de lavouras. 2023. Acesso em: 28 mai. 2026. Disponível em: <https://monografias.ufop.br/bitstream/35400000/6251/3/MONOGRAFIA_AgriculturaPrecis%c3%a3oSistema.pdf>.

Visão Computacional. *YOLO para Detecção de Objetos: Visão Geral*. 2025. Acesso em: 20 mai. 2025. Disponível em: <https://visaocomputacional.com.br/yolo-para-deteccao-de-objetos-visao-geral/>.

WRIGHT, G.; SHEA, S. *What is a microcomputer?* TechTarget, 2024. Acesso em: 21 set. 2025. Disponível em: <https://www.techtarget.com/iotagenda/definition/microcomputer>.

YAZDANKHAH, E. Comparative analysis of backend-as-a-service solutions for react native mobile applications: Firebase vs supabase. 2025. Acesso em: 03 mar. 2026. Disponível em: <https://www.theseus.fi/handle/10024/908059>.

YWG. *Guia detalhado de como plantar coentro*. 2022. Acesso em: 21 set. 2025. Disponível em: <https://blog.yeswegrow.com.br/guia-detalhado-de-como-plantar-coentro/>.

ZANELLA, F. et al. Crescimento de alface hidropônica sob diferentes intervalos de irrigação. *Ciência e Agrotecnologia*, SciELO Brasil, v. 32, n. 2, p. 366–370, 2008. Acesso em: 10 mar. 2026. Disponível em: <https://www.scielo.br/j/cagro/a/wndbWFDswrwjTQJpthYDsTn/?lang=pt>.

APÊNDICE A – MATERIAIS NECESSÁRIOS PARA CONSTRUÇÃO DA BANCADA HIDROPÔNICA

Componente	**Custo**	**Qtd. / Metragem**	**Subtotal**
Bomba Submersa	R$ 105,00 p/ unidade	1 unidade	R$ 105,00
Tubulação 100 mm de PVC	R$ 20,00 p/ metro	2,5 metros	R$ 50,00
Tubulação de 50 mm de PVC	R$ 15,00 p/ metro	2,5 metros	R$ 45,00
Tubulação 20 mm de PVC	R$ 20,00 p/ metro	2 metros	R$ 40,00
Reservatório de 20L	R$ 20,00 p/ unidade	1 unidade	R$ 20,00
Gotejadores GA2	R$ 0,70 p/ unidade	10 unidades	R$ 7,00
Sombrite 50%	R$ 13,00 p/ metro	5 metros	R$ 65,00
Conexões de PVC (caps, T, joelho, registro)	Variado	20 unidades	R$ 50,00
Plástico transparente para estufa	R$ 20,00 p/ metro	2 metros	R$ 40,00
Madeira para Bancada	R$ 8,00 p/ unidade	10 unidades	R$ 80,00
Total Estimado			**R$ 502,00**

Fonte: Autoria própria (2026).

APÊNDICE B – MATERIAIS PARA CULTIVO DO COENTRO

Componente	**Custo**	**Qtd. / Metragem**	**Subtotal**
Kit de Solução Nutritiva	R$ 55,00 p/ unidade	1 unidade	R$ 55,00
Palha de Arroz	R$ 15,00 o quilo	2 quilos	R$ 30,00
Sementes de Coentro Verdão	R$ 30,00 o quilo	0,5 quilo	R$ 15,00
Medidor de condutividade	R$ 25,00 p/ unidade	1 unidade	R$ 25,00
Medidor de Ph digital	R$ 40,00 p/ unidade	1 unidade	R$ 40,00
Total Estimado			**R$ 165,00**

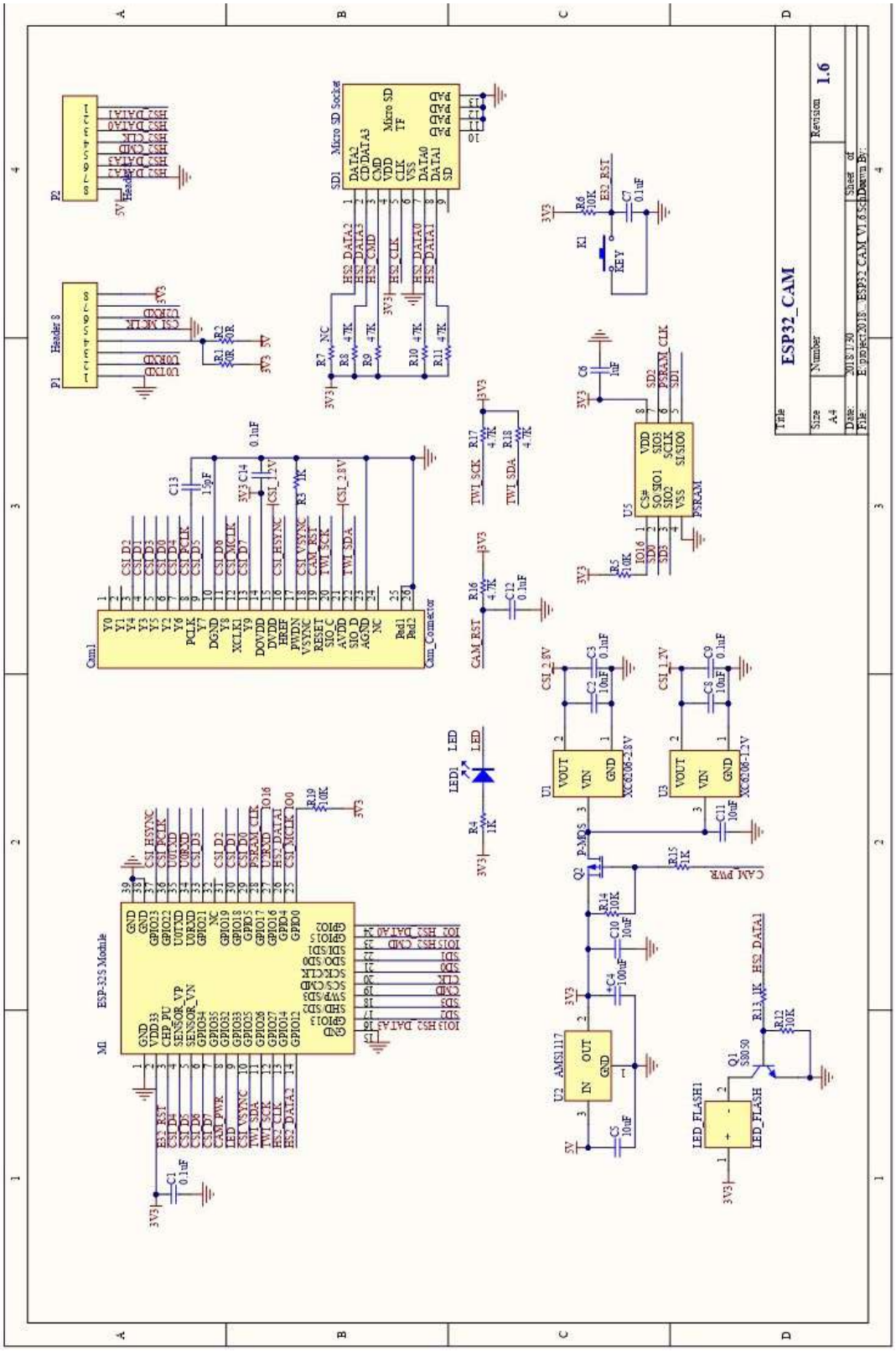
Fonte: Autoria própria (2026).

APÊNDICE C – COMPONENTES ELETRÔNICOS UTILIZADOS

Componente	**Custo**	**Qtd. / Metragem**	**Subtotal**
Esp32-S3	R$ 90,00 p/ unidade	1 unidade	R$ 90,00
Esp32-Cam	R$ 130,00 p/ unidade	1 unidade	R$ 130,00
Raspberry Pi 3B	R$ 288,00 p/ unidade	1 unidade	R$ 288,00
Módulo Relé de 1 Canal 5V	R$ 8,50 p/ unidade	1 unidade	R$ 8,50
Protoboard	R$ 12,00 p/ unidade	1 unidade	R$ 12,00
Fonte Ajustável p/ Protoboard	R$ 12,00 p/ unidade	1 unidade	R$ 12,00
Total Estimado			**R$ 540,50**

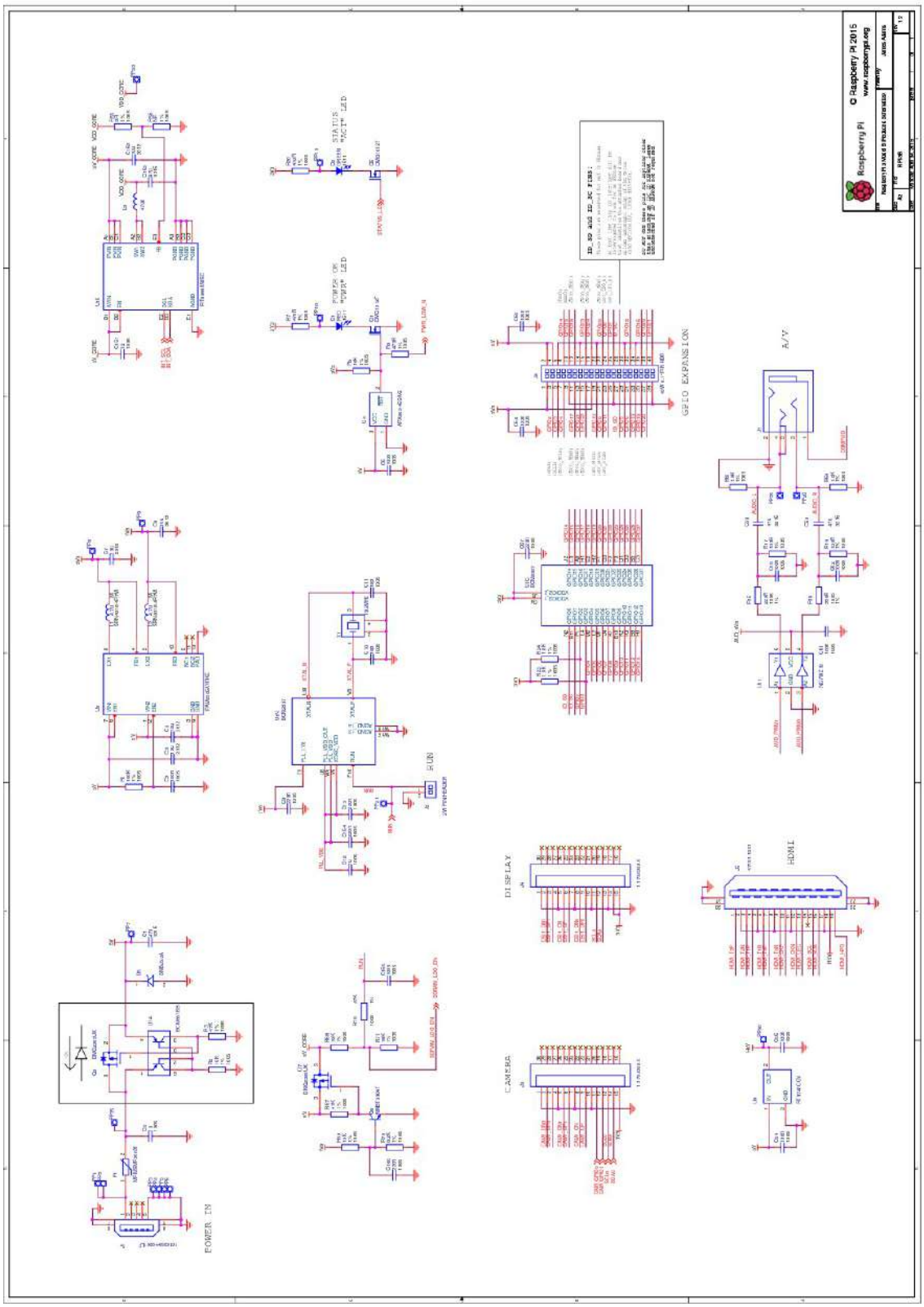
Fonte: Autoria própria (2026).

ANEXO A – DIAGRAMA ESQUEMÁTICO DA ESP32-CAM



Fonte: (SANTOS; SANTOS, 2020)

ANEXO B – DIAGRAMA ESQUEMÁTICO DO RASPBERRY PI 3B



Fonte: (Raspberry Pi Ltd, 2016)

Cópia de documento digital impresso por GUSTAVO NOGUEIRA (1653598) em 23/11/2026 11:15.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Campina Grande - Código INEP: 25137409
R. Tranqüilino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Trabalho

Assunto:	Trabalho
Assinado por:	Mirna Maia
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Documento Original

Documento assinado eletronicamente por:

- **Mirna Carelli Oliveira Maia**, **COORDENADOR(A) DE CURSO - FUC1 - CCEC-CG**, em 19/11/2026 15:25:48.

Este documento foi armazenado no SUAP em 19/11/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 2053415
Código de Autenticação: 6b4b5e9357